DOCTOR OF PHILOSOPHY

Twitter sentiment analysis on health services in Arabic

Alayba, Abdulaziz

Award date:
2019

Awarding institution:
Coventry University

Twitter Sentiment Analysis on Health Services in Arabic

By

Abdulaziz Mohammad Alayba

September 2018

A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy

Coventry University

Certificate of Ethical Approval

Applicant:

Abdulaziz Alayba

Project Title:

Twitter Sentiment Analysis on Health Services in Arabic Language

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

05 September 2018

Project Reference Number:

P75809

Please add a copy of this section within the first few pages of your thesis, after your title page. Refer to 'Thesis Information Guidance' for more information.

Section 3 Submission Declaration

	Yes	No
Have materials contained in your thesis/submission been used for any other submission for an academic award?	□	☒
If you have answered Yes to above please state award and awarding body and list the material:		
I am aware of no health reasons that will prevent me from undertaking and completing assessment and will undertake to notifying my Director of Studies and the Doctoral College and Centre as soon as any change in these circumstances occurs.	☒	□
Ethical Declaration: I declare that my research has full University Ethical approval and evidence of this has been included within my thesis/submission. Please also insert ethics reference number below Project Reference: **P 75809**	☒	□

Freedom of Information:

Freedom of Information Act 2000 (FOIA) ensures access to any information held by Coventry University, including theses, unless an exception or exceptional circumstances apply.

In the interest of scholarship, theses of the University are normally made freely available online in CURVE, the Institutions Repository, immediately on deposit. You may wish to restrict access to your thesis for a period of three years. Reasons for restricting access to the electronic thesis should be derived from exemptions under FOIA. (Please also refer to the University Regulations Section 8.12.5)

Do you wish to restrict access to thesis/submission: No

If Yes please specify reason for restriction:

Does any organisation, other than Coventry University, have an interest in the Intellectual Property Rights to your work?

No

If Yes please specify Organisation:

Please specify the nature of their interest:

Candidates Signature:	**Date:**

إهـــــــــداء

الحمد لله الذي بعزته وجلاله تتم الصالحات.. يا رب لك الحمد كما ينبغي لجلال وجهك وعظيم سلطانك.. حمداً كثيراً أقصى مبلغ الحمد.. والشكر له سبحانه من قبل ومن بعد.

والصلاة والسلام على من بلغ الرسالة وأدى الأمانة.. ونصح الأمة.. الرحمة المهداة والنعمة المسداة.. سيدنا محمد صلى الله عليه وسلم.

أما بعد، فإني أهدي هذا العمل:

إلى معلمي الأول في الحياة، من كان معززاً لي، من كان يرغب أن يرى أبنائه حاصلين على الشهادات العليا، كان ذلك دافعاً لي لتحقيق أمنيته، سمع باجتيازي لمناقشة هذه الرسالة، ودعني قائلاً "شوفتكم تسوى الدنيا"، ولم يلبث طويلاً بعد اجتياز المناقشة، والدي محمد اللعيباء رحمه الله..

إلى من غرست أهمية العلم في صغري، من أنعمت عليّ بصادق دعواتها التي أنارت طريقي، من أقف عاجزاً عن رد جميلها، أمي سعدى الخوير، اللهم أمدد لها بالعمر والعافية..

إلى رفيقتي في دربي، إلى من بدّدت ألم غربتي، إلى من سارت معي لتحقيق هذا الحلم، ومن شاركتني لحظةً بلحظة، بدأناه معاً وأنجزناه معاً وسنبقى معاً بإذن الله، زوجتي العزيزة إمتنـان الضبعان.. وإلى من غمرني بالسعادة والحياة في الغربة ابني وحبيبي محمد..

إلى إخواني الأعزاء أحمد، بندر، بدر، زياد وإلى أخواتي الغاليات إبتسام، حنان، أفراح، أسماء، نادية وإلى أبنائهم، منكم أستمد العزيمة والصمود، فشكراً لكم..

إلى جدتي الغالية، وإلى الأقرباء والأصدقاء وأخص بذلك كل من أهداني دعوة في ظهر الغيب، وإلى كل من علمني حرفاً في جميع مراحل دراستي التعليمية السابقة..

Acknowledgements

I would like to express my gratitude to my supervision team, Dr. Vasile Palade, Dr. Matthew England, and Dr. Rahat Iqbal, for their support and guidance throughout my PhD journey. In particular, I appreciate all of the effort and patience from my director of study Dr. Vasile Palade, and express my special thanks for his help. He has been available to advise, guide, support, and encourage me to overcome all challenges throughout this research. I am very blessed to be a PhD student under his supervision and I have learned a lot from him. I am also grateful to Dr. Matthew England for his technical help.

I would like to show my appreciation to the annotators for their time and effort in labelling the dataset. The annotators are: Dr. Abdullah M. Al-Homayan, Dr. Mohammad T. Alshammari, and Mr. Tariq F. Aldhamadi. They were very helpful and patient in reading and annotating 2026 tweets. Also, I appreciate all the efforts of Dr. Thamir Alomaim and Mr. Yasser Albakry in translating the words from Arabic to English in the tables in Appendix B.

I am very thankful to my sponsor, the University of Hail (UOH), in Saudi Arabia, for my scholarship. They have provided support throughout all my studies in the UK since 2010 to 2012 and since 2015 untill now. Also, I am grateful to the Royal Embassy of Saudi Arabia Cultural Bureau in London for their cooperation.

I would like to thank all of the friends and staff at Coventry University for their help. Also, I am extremely grateful to Coventry University for making available their research facilities and resources. A special thanks to Dr. Alex Pedcenko for his help in using the Zeus HPC.

Abstract

In the past decade, people have directed their interests toward using different social media platforms. These are where they share their experiences and life activities with other people. Twitter is one of the popular examples of social media, where users can write or post short messages. The data in Twitter is easily accessible and, thus, it is a good source to be analysed. Sentiment analysis is the task of extracting and classifying the opinions or the feeling from text.

This thesis first develops an Arabic Health Services (AHS) dataset for sentiment classification purposes. The dataset has been collected from Twitter, filtered, and annotated manually by three people. Then, three word embedding techniques which are Word2vec, GloVe, and fastText, were trained using two different Arabic corpora. These models obtain vectors to be used as input for the sentiment classification. This thesis also studies the effectiveness of different features for classifying Arabic text. The effectiveness of using word embedding models for sentiment analysis for short Arabic text was also investigated in this thesis.

Different sentiment analysis levels were proposed in order to deal with the complexities of the Arabic language in morphology and orthography. Additionally, several deep neural network models and machine learning classifiers were trained to conduct a sentiment analysis on the newly developed AHS dataset. In particular, a model that combines a Convolutional Neural Network combined with Long Short Term Memory has been used to analyse an Arabic dataset for the first time. The purpose CNN and LSTM model achieved good sentiment classification performance.

Table of Contents

List of Figures **xxi**

List of Tables **xxv**

List of Acronyms **xxix**

1 Introduction **1**

1.1 Introduction 2

1.2 Research Motivation 4

1.3 Aims and Objectives 6

1.4 Scope and Limitation 7

1.5 Thesis Contributions 8

1.6 Thesis Structure 9

1.7 Publications 10

2 Literature Review and Context **13**

2.1 Social Networks 14

2.2 Social Network Data Analysis 16

2.3 Sentiment Analysis 22

2.3.1 Arabic Sentiment Analysis Challenges 25

2.4 Related Work 28

2.4.1 General Sentiment Analysis 29
2.4.2 Health Sentiment Analysis 32
2.4.3 Arabic Sentiment Analysis 33
2.5 Summary 38

3 Methodology 39

3.1 Overview 40
3.2 Data Collection 42
3.3 Data Filtering 44
3.3.1 Stop Words 44
3.3.2 Repeated Letters or Repeated Words 45
3.3.3 Normalisation 45
3.3.4 Words with Similar Meanings 46
3.3.5 Arabic Homonyms Words 46
3.3.6 Spelling Mistakes 48
3.3.7 Stemming and Lemmatising 48
3.4 Text Features 49
3.4.1 N-grams 50
3.4.2 Word Embedding 51
3.4.3 Part of Speech Tagging (POS tagging) 51
3.4.4 Bag of Words 51
3.4.5 TF and TF-IDF 52
3.4.6 Stanford Tokenizer 53
3.5 Annotating the Corpus 54
3.6 Machine Learning Algorithms 54
3.7 Measuring Classifier Performance 58
3.8 Summary 59

4 Collecting and Annotating a Health Sentiment Dataset in Arabic 61

4.1 Data Collection Overview 62

4.2 Data Collection Process 62

4.3 Automatic Data Filtering 67

4.3.1 Removing Retweeted Tweets 68

4.3.2 Removing Overlapping Tweets 68

4.3.3 Filtering the Tweets’ Text 69

4.3.4 Normalising some Arabic Characters 69

4.3.5 Removing Short Vowels (Diacritics) 70

4.3.6 Removing Tatweel 70

4.4 Manual Data Filtering 70

4.4.1 Spam Tweets 71

4.4.2 Tweets with No Opinions 73

4.4.3 Removing any Opinion Irrelevant to Health 74

4.4.4 Removing #Hashtag_Topics, #Hash and Other Symbols 75

4.4.5 Combining Tweets 75

4.4.6 Editing Compressed Text 77

4.4.7 Repeated Letters 78

4.4.8 Compound Words 78

4.4.9 Words from Other Languages 79

4.4.10 Removing Special Letters 80

4.4.11 Correcting Spelling Mistakes 80

4.5 Data Annotating Process 82

4.6 Data Collection and Pre-processing Challenges 85

4.7 Summary 88

5 Word Embedding Models for Arabic 89

5.1 Overview 90

5.2 Word Embedding Techniques 91

5.2.1 Word2Vec Model 91

5.2.2 Global Vector Model (GloVe) 94

5.2.3 FastText Model 95

5.3 Arabic Language Corpora 96

5.3.1 Abu El-khair Corpus 96

5.3.2 Pre-processing the Abu El-khair Corpus 97

5.3.3 Arabic Twitter Corpus 98

5.3.4 Pre-processing the Arabic Twitter Corpus 99

5.4 Building and Evaluating Word Representation Models 100

5.5 Summary 102

6 Feature Sets and Classifiers for Sentiment Analysis 103

6.1 Overview 104

6.2 Sentiment Analysis Levels 105

6.2.1 Character Level (Char-Level) 105

6.2.2 Character 3-Gram Level (Ch3gram-Level) 105

6.2.3 Character 5-Gram Level (Ch5gram-level) 106

6.2.4 Stanford Tokenization Level (StanfordToken-Level) 107

6.2.5 Word Level (Word-Level) 108

6.2.6 Word Bigram Level (Word-Bigram-Level) 108

6.3 Using Different Machine Learning Algorithms 108

6.3.1 Different Machine Learning Algorithms 108

6.3.2 Evaluation and Discussion on Using Different Machine Learning Algorithms 111

6.4 Convolutional Neural Network (CNN) Models 118
6.4.1 CNN Model 120
6.4.2 Evaluation and Discussion of the CNN Model 122
6.4.3 Lexicon Integrated CNN Model 124
6.4.4 Evaluation and Discussion of the Lexicon Integrated CNN model . 126
6.5 A Combined CNN and LSTM Network Model 129
6.5.1 CNN and LSTM Model 129
6.5.2 Evaluation and Discussion of the Combined CNN and LSTM Network Model 132
6.6 Summary 134

7 Conclusions 137
7.1 Overview 138
7.2 Main Contributions 138
7.3 Future Work 140

References 143

Appendix A Summary of the Results for All Sentiment Classification Models 159

Appendix B The Most Similar Words to the Words "*good*" and "*bad*" in Arabic Using Word2Vec 163

Appendix C The Details of Filtering the Abu El-khair Corpus and the *Python* Code Used 169

List of Figures

2.1 Examples of density measurements for directed and undirected networks . . 17

2.2 An example of clustering, bonding and bridging techniques 18

2.3 An example of core and periphery nodes 19

3.1 The workflow of this project and the four main steps 41

4.1 An example of a spam tweet 72

4.2 An example of an advertisement tweet 73

4.3 An example of a news tweet about topic one 73

4.4 An example of a non-opinion tweet 73

4.5 An example of an irrelevant oninion in a tweet 74

4.6 An example of a tweet containing more than one hashtag and one of the hashtags contains opinion words 75

4.7 An example of a tweet that contains special letters 80

4.8 Visualising the number of positive and negative tweets in the dataset based on the three annotators 85

4.9 Visualising the number of positive and negative tweets in the dataset based on the four different topics 86

5.1 Word2vec model, continuous bag of words (CBOW) and skip-gram (SG) . 92

5.2 The Word2Vec CBOW neural network architecture 93

5.3 The Word2Vec SG neural network architecture 94

5.4 The fastText SG neural network architecture 96

5.5 An example of one article in one of the XML files 98

6.1 An example of the *Part of Speech Tagging* on an Arabic tweet 111

6.2 The accuracy for the Main-AHS dataset using different machine learning models, with the three used features based on different sentiment analysis levels 116

6.3 The accuracy for the Sub-AHS dataset using different machine learning models, with the three used features based on different sentiment analysis levels 117

6.4 CNN model architecture with an example tweet to classify it as either *Positive* or *Negative*. 120

6.5 The accuracy of the CNN model on the test set for the Main-AHS dataset using different sentiment analysis levels 123

6.6 The accuracy of the CNN model on the test set for the Sub-AHS dataset using different sentiment analysis levels 123

6.7 The Lexicon Integrated CNN model architecture, the lexicon embeddings are concatenated to the word embeddings. 125

6.8 The accuracy of the Lexicon Integrated CNN Model on the test set for the Main-AHS dataset using different word representation models 128

6.9 The accuracy of the Lexicon Integrated CNN Model on the test set for the Sub-AHS dataset using different word representation models 128

6.10 A combination of the CNN and LSTM model architecture for sentiment analysis, with an example of an Arabic tweet. 130

6.11 The accuracy of the combined CNN and LSTM network model on the test set for the Main-AHS dataset using different sentiment analysis levels . . . 133

6.12 The accuracy of the combined CNN and LSTM network model on the test set for the Sub-AHS dataset using different sentiment analysis levels 133

C.1 An example of one article in one of the XML files after runing the Python code 170

C.2 The First part of the Python code to do the further text pre-processing . . . 171

C.3 The second part of the Python code to do the further text pre-processing . . 172

List of Tables

2.1 Examples of direct one-way, direct two-way and indirect links 19

2.2 Some examples of the multiple forms of a single Arabic verb 26

3.1 Examples of words related to health in English, Arabic and the UTF-8 encoding 43

3.2 The multiple meanings of one Arabic word 47

3.3 An example of an Arabic word that has different meanings in two tweets . . 47

3.4 An example of an Arabic word with prefix and suffex 49

3.5 The confusion matrix 58

4.1 The hashtags of the Arabic Health Services (AHS) dataset 64

4.2 Number of retrieved CSV files 66

4.3 The changes in the number of tweets for each topic before and after filtering the dataset and the percentage change 67

4.4 The changes in the number of tweets for each topic before and after removing retweeted tweets in the dataset and the percentage change 68

4.5 The changes in the number of tweets before and after removing duplicated tweets for each topic 69

4.6 The changes in the number of tweets before and after filtering the tweets manually 71

4.7 The changes in the number of tweets before and after removing the spam tweets . . . 72

4.8 The changes in the number of tweets before and after removing the tweets with no opinions . . . 74

4.9 An example of multiple tweets from one user about one idea . . . 76

4.10 The changes in the number of tweets before and after combining the tweets 77

4.11 Examples of words merged as one word, the number of merged words and the correction . . . 78

4.12 Examples of compound words (highlighted and underlined for each tweet) . 79

4.13 Examples of compound nouns before and after merging . . . 79

4.14 Two examples of non-Arabic words used in the dataset (the words are highlighted and underlined) . . . 80

4.15 Four examples of different spelling mistakes . . . 81

4.16 Four examples of different majority voting (P = Positive & N = Negative) The first two examples show that all the annotators agree either positive or negative. However, in the last two examples one annotator has a different judgment from the other two annotators. . . . 83

4.17 Summary of all the different states, the number of occurrences of each state, the majority voting of each state (Final Sentiment) and the total number of positive and negative tweets. The Sub-(AHS) is labelled using the blue line 84

4.18 Two examples of Arabic tweets which were not decoded to Arabic characters 86

6.1 Examples of words in a tweet and their stem and root forms using Tashsphyne Arabic Light Stemmer . . . 104

6.2 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Char-Level . . . 112

6.3 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Ch3gram-Level 113

6.4 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Ch5gram-Level 113

6.5 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the StanfordToken-Level 114

6.6 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Word-Level 115

6.7 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Word-Bigram-Level 115

6.8 The size of the longest tweets in AHS dataset based on different levels . . . 119

6.9 The accuracy of the CNN model for the Main-AHS and the Sub-AHS datasets based on the test set using different sentiment levels 122

6.10 The accuracy of the Lexicon Integrated CNN model for the Main-AHS and Sub-AHS datasets using different word embedding models based on the test set126

6.11 The accuracy results of the Combined CNN and LSTM network model for the Main-AHS and Sub-AHS datasets based on the test set using different sentiment levels 132

A.1 Summary of all the accuracies for ML classifiers with the three feature selections (TF, TF-IDF, POS) for both Main-AHS and Sub-AHS 160

A.2 Summary of all the accuracies for CNN model and CNN and LSTM model with different sentiment levels for both Main-AHS and Sub-AHS 161

A.3 All the accuracies for Lexicon Integrated CNN model with different word embedding models for both Main-AHS and Sub-AHS 161

B.1 The most similar words to the word *good* in Arabic using the Word2Vec technique *CBOW* for different dimensionalities: 10, 50, 100, 200, and 300 . 164

B.2 The most similar words to the word *bad* in Arabic using the Word2Vec technique *CBOW* for different dimensionalities: 10, 50, 100, 200, and 300 . 165

B.3 The most similar words to the word *good* in Arabic using the Word2Vec technique *SG* for different dimensionalities: 10, 50, 100, 200, and 300 . . . 166

B.4 The most similar words to the word *bad* in Arabic using the Word2Vec technique *SG* for different dimensionalities: 10, 50, 100, 200, and 300 . . . 167

List of Acronyms

Roman Symbols

AHS dataset	Arabic Health Services dataset that we collected from Twitter
ANEW	Affective Norms for English Words
ANLP	Arabic Natural Language Processing
API	Application Program Interface
ASTD	Arabic Sentiment Tweets Dataset
BBS	Bulletin Board System
BNB	Bernoulli Naive Bayes
CBOW	Continuous Bag of Words
Ch3gram-Level	Character 3-Gram Level
Ch5gram-Level	Character 5-Gram Level
Char-Level	Character Level
CNN	Convolutional Neural Network
ConvNets	Convolutional Neural Network

CSV	Comma Separated Values
DAE	Deep Auto Encoders
DBN	Deep Belief Networks
DNN	Deep Neural Networks
FN	False Negative
FP	False Positive
GloVe	Global Vectors for Word Representation
GPS	Global Positioning System
HCR	Health Care Reform
KNN	K-Nearest Neighbours
LABR	Large Scale Arabic Book Reviews Dataset
LIWC	Linguistic Inquiry and Word Count
LR	Logistic Regression
LSTM	Long Short Term Memory
LSVC	Linear Support Vector Classification
Main-AHS	is the Arabic Health Services dataset that we collected it from Twitter, and it contains 2026 tweets
Max-Ent	Maximum Entropy
ML	Machine Learning

MNB	Multinomial Naive Bayes
MSA	Modern Standard Arabic
NB	Naïve Bayes
NLP	Natural Language Processing
NN	Neural Networks
NSVC	Nu-Support Vector Classification
OMD	Obama-McCain debate
POS	Part of Speech
RAE	Recursive Auto Encoder
RDG	Ridge Classifier
RecNN	Recursive Neural Network
ReLU	Rectified Linear Unit
REST APIs	Representational State Transfer-Application Programming Interface
RNN	Recurrent Neural Network
SG	Skip-gram
SGD	Stochastic gradient descent
SGML	Standard Generalised Markup Language
SNA	Social Network Analysis

StanfordToken-Level	Stanford Tokenization Level
STS	Stanford Twitter Sentiment
Sub-AHS	is the Arabic Health Services dataset which is a sub set of the Main-AHS, and it contains 1732 tweets
SVM	Support Vector Machines
TF	Term Frequency
TF-IDF	Term Frequency Inverse Document Frequency
TN	True Negative
TP	True Positive
TXT	Plain text
URL	Universal Resource Locator
UTF-8	Unicode Transformation Format that uses 8-bit
Word-Bigram-Level	Word Bigram Level
Word-Level	Word Level
WordNet	A Large Lexical Database for the English Language
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Introduction

In the past fifteen years, the number of social networks and users of them has increased rapidly. Seman (2014) confirmed that social networking on the Internet started in 1978 with the creation of the Bulletin Board System (BBS), which was the first online site that allowed users to sign up and communicate with other users. Boyd and Ellison (2007) mentioned that the first social network site, as it is defined in this thesis in Chapter 2, appeared on the web in 1997.

Today, there are a vast number of social media sites, many with an enormous number of users who share their life experiences with their friends in different forms, such as texts, emoticons, photos, videos, GPS locations and others. This growth has led to a flood of data: Gantz and Reinsel (2012) estimate that the amount of data from 2005 to 2020 will increase by a factor of 300. Social media sites in 2005 contained 130 exabytes of data; by 2020, that figure will reach approximately 40,000 exabytes, or 40 trillion gigabytes. Also, Ularu et al. (2012) stated that Facebook only receives about 100 terabytes of data every day. At this level, data is called "Big Data" and Roger Magoulas from O'Reilly Media introduced this term in 2005. Olshannikova et al. (2017) noted "social big data" as one of the Big Data types. Examples include Facebook posts, events, photos, links, video, relationships, etc; Twitter tweets, followers, direct messages, GPS locations, etc; and Instagram data, such as photos, GPS locations, friendships, etc. Furthermore, customer feedback or reviews on some applications like Yelp.com, TripAdvisor.com and Foursquare.com all count as social data. The data needs to be analysed to be of value: there are several different kinds of social networking data analytic tasks, the one of interest here is sentiment analysis.

Dey et al. (2016) define sentiment analysis as a combination of text mining and natural language processing and offer two classifications: polarity classification and subjectivity classification. The polarity classification classifies the text based on the opinion or emotional state whether positive, negative, or neutral. Subjectivity based classification divides the text into either subjective or objective. In the subjective, the text describes the feelings or orientations of the user, while objective phrases or sentences contain only factual information. There are different levels at which to apply the sentiment analysis, which are: document level, sentence level, phrase level (Balaji et al., 2017), entity and aspect level (Feldman, 2013), word level, character level (Lakomkin et al., 2017) and sub-word level (Joshi et al., 2016).

- **Document level :** classifies the entire document into its class. Articles are classified into their categories, such as politics, finance, sports, etc. (Pang et al., 2002).
- **Sentence level :** labels each sentence according to different classes such as, positive, negative, or neutral (Wiebe et al., 1999).
- **Phrase level :** detects whether an expression or a phrase has a positive, negative, or neutral opinion (Wilson et al., 2005).
- **Entity and aspect level :** considers the entity only, such as service, product, etc. and the aspect of this entity such as the quality, price, etc. In the following example, "*The car's engine is very good, but the colour is bad*", the entity is *car* and there are two aspects, which are the *engine* and the *colour* (Liu, 2012).
- **Word level :** represents each word in the sentiment dataset as a single feature (Neviarouskaya et al., 2007).
- **Sub-word level :** focuses on sub-words instead of using entire words or using characters only, such as in the work of Joshi et al. (2016) using the Long Short Term Memory network.
- **Character level :** uses the characters as features in a row of text. It has been used in a lot of research, such as in Zhang et al. (2015) and Golub and He (2016).

1.2 Research Motivation

Sentiment analysis is an approach of classifying text that contains any sentiment, opinion, state, attitude, evaluation, recommendation, and emotion of a person to a sentiment class (Liu, 2012). The sentiment classes can be emotional ones like *anger, disgust, fear, joy, love, sadness and surprise*, as presented by Roberts et al. (2012). Also, they might be a position on a scale, such as *very positive, positive, neutral, negative and very negative*, as was presented by Le and Mikolov (2014). A common framework in sentiment analysis is to have three classes that are *positive, negative and neutral*, like in Pak and Paroubek (2010); or only two classes that are *positive and negative*, as in Socher et al. (2013). This field can be named as either opinion mining or sentiment analysis, and it is a subset of text classification with a combination of computational linguistics and natural language processing.

There are a series of steps that need to be followed in order to measure the sentiment in the text. The steps start by reading the text and mapping the labels to it. After that, the text needs to be tokenized, which can be sentence tokenization, word tokenization, or character tokenization. Word tokenization is the most common choice in sentiment analysis and text classification. Then, there are some optional steps, which can be implemented, such as filtering stop words, stemming or lemmatising the words and dealing with negation words. Next, converting the text based on the tokenization into numeric data using different techniques of feature selection or feature extraction. After that, training the machine learning classifiers using the converted numeric data on the training set and mapping each document with the label. Finally, using the trained model to predict a class of text not used in the training and measure the efficiency of the classifier using different measurements techniques, such as a confusion matrix, precision, recall, and accuracy.

The difficulties of this task are that many attributes are involved, which can affect the prediction and the analytics. These involve the type of the machine learning classifiers used, the feature selection used, and the nature of the unstructured data (the text in our case). In addition to that, the Arabic language is complex, because there is a standard or a formal Arabic language, and then there are twenty-two Arab countries that use Arabic as a first language, and each country has one or more dialects. Also, short vowels or diacritics can change the meaning in many words that have the same spelling. Moreover, Arabic words are either masculine or feminine; nouns can be either singular, dual, or plural; verbs in Arabic, based on the tense can be categorised as past, present, and imperative, and so there are many forms for a single word. The masculine or feminine forms in the Arabic language are not always referring to gender. For example, the words sun "شمس", fire "نار", tree "شجرة", and car "سىارة" are treated as feminine nouns. In contrast, the Arabic words moon "قمر", finger "إصبع", pen " قلم", and book "كتاب" are classified as masculine nouns. For the plural nouns, the Arabic language has three categories that are: the masculine plural, the feminine plural, and the broken plural. Therefore, dealing with Arabic text and short messages (tweets) makes the sentiment analysis a very challenging task. Twitter users usually write a tweet and express their opinion using their own dialects and their own style of writing.

Sentiment analysis has been considered by many researchers in different applications, such as movie reviews (Pang et al., 2002; Ghorbel and Jacot, 2011), political elections (Bermingham and Smeaton, 2011; Tunggawan and Soelistio, 2016), Amazon reviews (Jindal and Liu, 2008), emotions classification (Hogenboom et al., 2013). Health care has been considered in many researches, such as (Speriosu et al., 2011). Health services from a topic of interest everywhere in the world due to the importance of health care. Many patients or some relatives of them may share on the social media their opinions about their experiences in the hospitals or clinics. Twitter is one of the most attractive platform and there is very large number of active Twitter users in Saudi Arabic (Alasem, 2015). Some examples of sharing opinions about health services in Arabic in Twitter are:

شئ مؤسف تردى حال الحدمات الطبة..الحدمة الطبة ڡى الأردں ممتارة ودات جودة وبسعر
اقل بكثر مں السعودىة..لىس هىاك ما ىبهج ڡى الصحة ڡى السعودىة,

الحدمات الصحة ڡى شمال المملكة مں باحة المىشاءات جده ولكں تنقصها الكادرالطبى , المتحصص لتعطة احتاجات المطقه الطبه,

السعودنة قامت بعمل ممتار ڡى التحرى عں ڡاروس كوروىا والحلولة دوں ادلاعه بشكل وبائى.

Hence, we are targeting Twitter as the source to collect the tweets about health services. Due to the lack of sentiment analysis research in the Arabic language, an Arabic sentiment dataset will be introduced by the present thesis. Gathering tweets, which contain reviews or opinions on a specific topic from Twitter, is a challenging task because of the unrestrained nature of the Twitter platform. Therefore, we need to launch a Twitter hashtag and ask Twitter's users to share their opinions about their experiences with health services. However, it is arduous to encourage and involve a large number of people to write their opinions about their experiences on any topic, e.g., health services in our case. So an alternative approach to collect the tweets about health services is also used by observing the Twitter trending hashtags in Saudi Arabia and collecting any hashtag about health. Another difficulty is filtering the retrieved tweets and only keeping the tweets that contain opinions about health services, and labelling the dataset. Word representation and deep neural networks will be used to improve the sentiment classification. Additionally, the effect of reducing the forms of some Arabic words by using an Arabic segmenter will be measured.

1.3 Aims and Objectives

This thesis introduces an Arabic Health Services (AHS) dataset for sentiment classification from Twitter. Also, the thesis presents a study on sentiment analysis on Arabic scripts from social media and explores the role of using different feature selection methods and different machine learning methods. It also addresses the role of using deep neural networks and word embedding to represent the vocabulary. The following explicit objectives were identified in order to achieve the previous aims. The first objective is collecting the data from Twitter about health services and developing our own dataset by retrieving relevant Arabic tweets. Then, filtering the dataset from unwanted data, i.e. unrelated tweets to health services, repeated tweets, tweets that do not contain an opinion, etc. Filtering and pre-processing the text of the corpus, e.g. removing short vowels, Tatweel, repeated letters, etc is the next objective. The fourth objective is annotating and building a health sentiment analysis dataset from Twitter by three Arabic native speakers followed by identifying the effectiveness of extracted features, such as unigram, bigram, part of speech tagging (POS), word embedding, etc. The sixth objective is implementing various machine learning methods, and especially deep neural network models, to measure the sentiment on health services from Twitter messages in Arabic. The final objective is investigating different machine learning algorithms, including neural networks, and using different features in order to improve the accuracy of sentiment analysis.

1.4 Scope and Limitation

- This thesis will perform sentiment analysis on short messages (tweets) that contain a limited number of characters: only 140. The dataset was collected from Twitter before the increase in allowed characters to 280.

- Twitter users ordinarily write tweets using their own style of writing, which is not the Standard Arabic Language or the Formal Arabic Language. The tweets are locally collected based on Saudi Arabian Twitter users.

- The number of tweets in the dataset is limited due to the dataset being only focused on health service reviews, and it was collected from Twitter which is not just a platform for opinions.

- The dataset is unbalanced, presenting positive tweets at 69.01% and negative tweets at 30.99%. The limited number of tweets meant it was not appropriate to make the dataset balanced.
- The classes in the dataset are binary, positive and negative. The neutral class has been eliminated because the number of tweets in this class would overwhelm the others.
- There is no large Arabic Twitter corpus available to be used for building word embedding models.
- The used features in the experiments are not the same for all models.
- The experiments focus on using deep neural networks compared with other machine learning algorithms.
- The dimensionalities for GloVe and fastText models are 200 and it is based on the best Word2Vec models which obtained in our study.

1.5 Thesis Contributions

The main contributions of this thesis are:

- Developing a new healthcare dataset for sentiment analysis in Arabic (there is no such dataset already available). The dataset has been collected from Twitter and it has been annotated by three human annotators. It is freely available online at Alayba et al. (2018b). This contribution has also been reported in Alayba et al. (2017).
- Identifying and comparing between a set of feature representations (namely Word2vec, GloVe, and fastText) suitable for the novel AHS dataset. Comparing between different deep learning models to classify users' opinions in the AHS dataset (Alayba et al., 2018c).

- Using novel effective features for the Arabic language to be used for sentiment analysis classification and other information retrieval tasks in Arabic, e.g. security, marketing, etc. This contribution has also been presented in Alayba et al. (2018c) and Alayba et al. (2018a).
- Implementing and training different neural network models and machine learning methods. Some of which were used for the first time in the context of Arabic sentiment analysis, in order to classify the sentiment on health services from Twitter messages in the Arabic language. This contribution has also been stated in Alayba et al. (2017), Alayba et al. (2018c), and Alayba et al. (2018a).
- Describing a novel manual tuning procedure to identify the best dimensionality for a word embedding model for binary sentiment analysis (Alayba et al., 2018c).
- Attempting to address the issues of Arabic morphology through the usage of n-grams and Stanford Tokenizer (Alayba et al., 2018a).

1.6 Thesis Structure

The rest of the thesis is organised as follows: Chapter 2 presents the literature review and the context of this thesis. Social networks and sentiment analysis are introduced, and the challenges of the sentiment analysis are explained in this chapter. Some related works will be described briefly. The related works are based on the general sentiment analysis, Arabic sentiment analysis and health sentiment analysis respectively.

Chapter 3 describes the methodological flow and other background in this research. It defines the different ways of collecting, filtering and annotating the sentiment data. Moreover, it clarifies some terms related to the sentiment analysis or text classification in general. For example, stemming, lemmatisation, normalisation, n-gram, part of speech tagging, etc. Also, it explains some machine learning classifiers used and the way of measuring the performance of the classifiers.

Chapter 4 details all of the steps for collecting, labelling and building an Arabic sentiment dataset. It explains the tools used to retrieve Arabic tweets and the way of filtering and pre-processing the sentiment tweets. For example, removing retweeted tweets, removing spam tweets, removing no opinion tweets, etc. Also, the process of annotating the dataset is illustrated. Finally, some data collection and pre-processing challenges are mentioned.

Chapter 5 introduces the term ‘word embedding’ and the procedures of building Arabic word embedding models. There are three word representation techniques used, which are; Word2vec, GloVe, and fastText. In addition, the different word embedding models and Arabic language corpora will be clarified. The process of building and filtering the Arabic corpora will be introduced too. Finally, the ways of evaluating the word embedding models will be demonstrated.

Chapter 6 explains all of the sentiment analysis methods applied using different feature selection and extraction methods. In addition, it illustrates the used machine learning algorithms. Also, it will introduce different sentiment analysis levels in order to have different tokenization of the text and generate different features. Furthermore, it will describe the architecture of the sentiment analysis models used. Finally, the way of evaluating the performance of different classifiers or sentiment analysis models will be discussed as well.

Chapter 7 concludes the thesis, clarifies the contributions of this study, and introduces possible future applications and improvements for this research.

1.7 Publications

Parts of this thesis have been published in the following list.

1. Alayba, A., Palade, V., England, M., and Iqbal, R. (2017). Arabic Language Sentiment Analysis on Health Services. In *1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 114–118, Nancy, France. IEEE.

2. Alayba, A. M., Palade, V., England, M., and Iqbal, R. (2018). Improving Sentiment Analysis in Arabic Using Word Representation. In 2*nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, pages 13–18, London, UK. IEEE.

3. Alayba A.M., Palade V., England M., Iqbal R. (2018) A Combined CNN and LSTM Model for Arabic Sentiment Analysis. In: *Machine Learning and Knowledge Extraction. CD-MAKE 2018*. Lecture Notes in Computer Science, pages 179-191, vol 11015. Springer, Cham

4. A journal paper is planned for submission soon.

Chapter 2

Literature Review and Context

2.1 Social Networks

There are many social network web applications available on the internet and most of them are free to use. Marin and Wellman (2011) defined the term social network where a node represents a member or user of a social network, and a given link is the relation or connection between any two users. Moreover, Aggarwal (2011) offers a similar perspective, describing social networks as interactions or a connection network between nodes (users) through links (relationships). Also, Boyd and Ellison (2007) introduced a definition of a social network site as a service that provides social benefits through the web. The three main actions that a user is able to undertake on social networks are:

- creating a public or semi-public profile or page under the rules of the site;
- listing connections with other users within the system;
- observing and interacting with one's own list of connections and the connections of other users in the system.

Mislove et al. (2007) divide social networks into three components:

- **Users:** a user must sign up for the system to obtain the benefit of the services that it provides. Users can edit profile information, such as adding photos, their birthday, education, work experiences, location(s), interests, etc.
- **Links:** the relationship or connection between two users is described as a link. The creation of a link between two users can be achieved in two ways. Under the first method, some social network websites allow a user to send a request to another user, and the connection is made if that user accepts it. The second method used by social network sites establishes links immediately, without waiting for approval from the other user. Some systems, like Facebook and Instagram, offer hybrid options that allow users to tailor the degree of permission required.

- **Groups:** most social network websites offer the option of creating a group. Different users with similar interests can join the group, which permits them to interact within the group, regardless of whether they are otherwise linked.

We define "social network" as a platform which allows the users to create their own profiles, create their connections with other users based on their interests, and share different forms of data with other users. Different platforms have different functionalities, such as, in Twitter, users can favourite or retweet other tweets, or post a new tweet that contains text, photo, video, link, etc. Also, there are many actions that users can do in Twitter and other social networks such as, post, like, comment, follow, unfollow, subscribe, block other users, message other users, etc.

Most social network sites in Arabic are the same as the existing English social network sites, but they are able to support the Arabic language. The number of users on social network sites in Arab countries is huge. The numbers are different depending on the different social network sites or different countries. TNS (2015) reported that the most used platforms in Arab countries were Facebook and WhatsApp which had over 80% of users. While, the users of YouTube, Instagram and Twitter were 39%, 34%, and 32% respectively. The user can register and exploit the available services and access most of the social network features. Different social networks have different features, such as sharing photos or videos, blogs, microbloggings, movies, music, businesses, academics and researchers, books, travel, games, general topics, etc. There are many examples of social network sites, but the most common sites are Facebook, Twitter, LinkedIn, Google+, Instagram, WhatsApp and Snapchat.

Twitter will be introduced briefly because the data in this study will be collected from it. Kwak et al. (2010) defined Twitter as a micro-blogging service established in July 2006 by inventors Jack Dorsey, Noah Glass, Biz Stone, and Evan Williams. Twitter users can traditionally write up to 140 characters in each tweet, although in November 2017 this limit increased to 280 characters. The followers of a given user will receive each of that user's tweets on their timelines. A user's unique identity is based on a username, which means that no two usernames can be identical to each other: all usernames start with the "@" character. A tweet's receivers can perform three main actions on the tweet: retweet, reply to the writer of the tweet, or mark the tweet as a favourite. There are other actions such as copy tweet link, report tweet, add to moments, etc. The hashtag ("#") symbol is used in Twitter in front of a word (or more, but without a space) and it can categorise the tweets into a subject or a topic. Clicking on this hashtag generates all the tweets containing this hashtag. This is to make it easier to find all the tweets that have the same subject or topic. Ahmed et al. (2017) reported some reasons for the common use of Twitter in academic research, such as the availability and accessibility of Twitter API to the public, the advantage of collecting data using the hashtags, and the wide usage of Twitter by many people. According to Twitter (2018), the platform has 335 million active users monthly and 80% of active users use it with mobile devices; and about 34 languages are supported.

2.2 Social Network Data Analysis

Social Network Analysis (SNA) is an attractive topic for several disciplines, such as sociology, mathematics, communications, physics and computer science. The purpose of SNA, as Ehrlich and Carboni (2005) introduced it, is to reveal the structures and relationships between people by focusing on a group and identifying their social connections. Hoppe and Reinelt (2010) reported that people, events, ideas and objects can all be represented as the nodes of the network, with the link between any two nodes constituting a relationship. There are several mathematical techniques to measure networks:

- **Density**

  Hoppe and Reinelt (2010) indicated that the density of a social network denotes the general level of connectedness within it. It can be measured by dividing the actual number of existing links by the number of all possible links between nodes in the network. Let N be the number of nodes and L the number of links. All possible links in a directed network can be measured by the formula $N(N-1)$ and all possible links in an undirected network can be measured using $\frac{1}{2}(N)(N-1)$. Figure 2.1 explains the measurements of the density for directed and undirected networks.

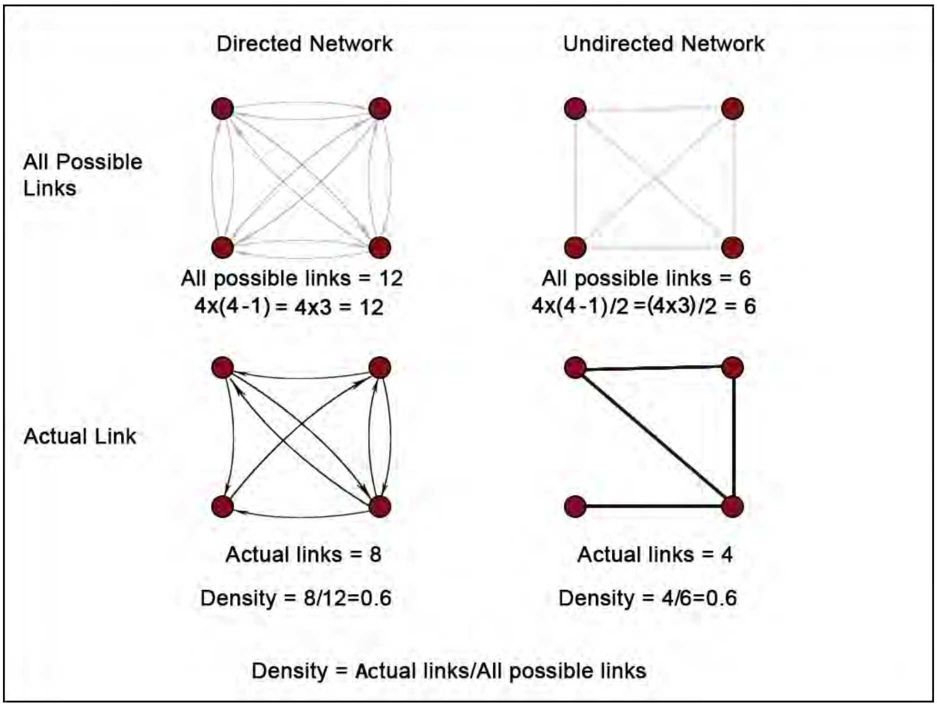


Fig. 2.1 Examples of density measurements for directed and undirected networks

- **Clustering**

  Clustering is one method that can be used to analyse social networks. Hoppe and Reinelt (2010) clarified that clustering in terms of SNA involves gathering nodes in subgroups that share similar attributes. Figure 2.2 illustrates the idea of clustering nodes.

- **Bonding and Bridging**

  Geys and Murdoch (2010) state that bonding and bridging are terms used in SNA to classify networks. Bonding involves tying together nodes that are highly connected to each other, while bridging is linking two different groups that have varied connections via a separate node. Figure 2.2 shows the difference between bonding and bridging.

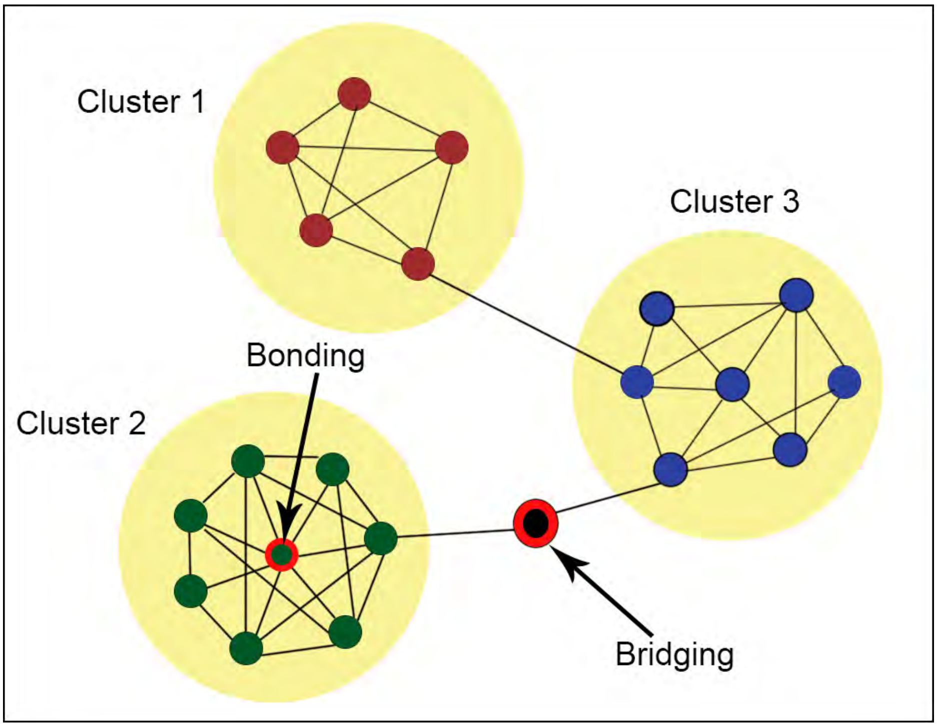


Fig. 2.2 An example of clustering, bonding and bridging techniques

- **Core and Periphery**

  Hoppe and Reinelt (2010) explained that core and periphery are the structural features of a network. The core is a node that has the densest connections with other nodes, while peripheries have the fewest links to other nodes. In Figure 2.3, node 16 is the core node and node 14 is a periphery node.

- **Direct and Indirect Links**

  Otte and Rousseau (2002) illustrated that the links in a network can be direct or indirect. A direct link has initial and final points and it can be either a one-way or two-way link. The former, also known as an unreciprocated link, moves in only one direction, such as a link beginning at node (A) and ending at node (B). A two-way or reciprocated link has connections that move in both directions. An indirect link, meanwhile, involves linking two nodes when the direction has no significance. Table 2.1 clarifies the differences between direct and indirect links.

Table 2.1 Examples of direct one-way, direct two-way and indirect links

Link Type	Shape	Description
Direct (One-Way)	Ⓐ→Ⓑ	(A) sends a message to (B)
Direct (Two-Way)	Ⓐ↔Ⓑ	(A) sends a message to (B) and (B) replies to (A)
Indirect	Ⓐ—Ⓑ	(A) and (B) know each other

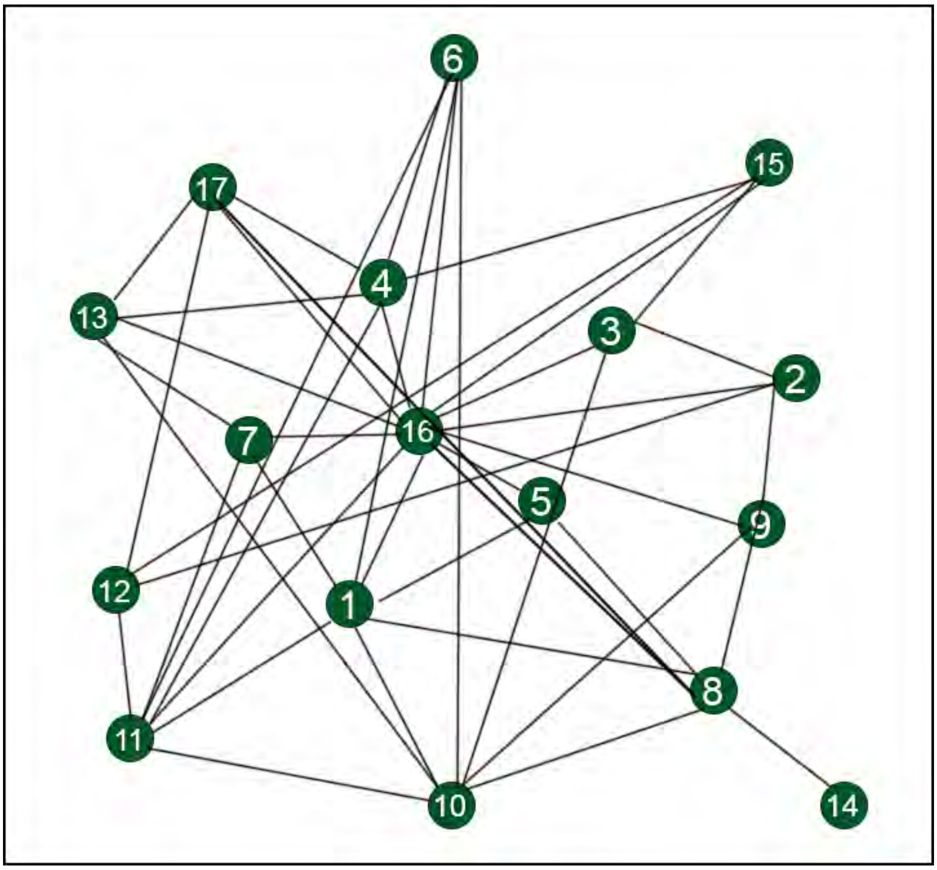


Fig. 2.3 An example of core and periphery nodes

- **Centrality**

  Otte and Rousseau (2002) explained that the three most significant measurements to calculate the centrality are the degree centrality, the closeness centrality and the betweenness centrality.

  Degree centrality refers to the number of links that connect to the nodes and is measured by Equation 2.1:

$$d_c(n) = \frac{d(n)}{N-1} \tag{2.1}$$

  where $d(n)$ is the total number of links that are connected to a node n, and N is the total number of the nodes in the network (Otte and Rousseau, 2002).

  Closeness centrality is the total distance from all other nodes in the network to a given node; it is measured by Equation:

$$c_c(n) = \frac{N-1}{c(n)} \tag{2.2}$$

  where $c(n)$ is the total number of shortest paths from node m to node n (Otte and Rousseau, 2002).

  Betweenness centrality is the number of shortest routes between different nodes passing via this node as Equation 2.3:

$$b(n) = \sum_{i,j} \frac{r_{inj}}{r_{ij}} \tag{2.3}$$

  where r_{inj} is the number of shortest routes or paths from node i to node j passing through node n. $(i \neq n \neq j)$; and r_{ij} is the total number of shortest routes or paths from node i to node j (Otte and Rousseau, 2002).

In a network containing N nodes, the maximum value of $b(n)$ is $\frac{1}{2}(N^2-3N+2)$, so betweenness centrality $(b_c(n))$ is represented by Equation 2.4:

$$b_c(n) = \frac{2b(n)}{N^2-3N+2}. \tag{2.4}$$

All the previous measurements may be used to analyse the relations between users in social networks. Adedoyin-Olowe et al. (2014) presented the following five approaches to social networks data analytics in a survey of data mining techniques.

1. Graph theory tools that analyse the main features of networks like nodes and links that represent the friendships between users or followers and the one who is followed. There are also topics related to graph theory, like community detection, recommender systems and the semantic web.

2. Opinion analysis on social networks, which focuses on classifying users' opinions about a particular subject as expressed on social media.

3. Supervised, semi-supervised and unsupervised classification of social network data.

4. Topic detection and tracking, which concentrates on unearthing new topics or events through social media.

5. Sentiment analysis of social networks, which classifies users' opinions into positive, negative, or neutral. There are subtypes of sentiment analysis, such as sentiment orientation, product ratings and reviews, aspect rating analysis and sentiment lexicon.

In addition, there are many different techniques to analyse the content of social networks and the impact of users on other users. There is a rise in the number of social networking sites that provide some services, and users can write their opinions and reviews about the products or the services, (e.g. Booking.com, Tripadvisor.com, Yelp.com, Amazon.com, etc).

The reviews are usually in a text format and sentiment analysis (opinion mining) is a task to study people's opinions or feelings using machine learning algorithms.

2.3 Sentiment Analysis

Sentiment analysis aims to reveal the opinions, attitudes, reactions, recommendations or emotions through the text, such as reviews, states, survey responses, social media, and healthcare (Liu, 2012). The advent in the web applications especially the growth of the social media web sites drives people to share their opinions on them. There are many web sites that allow users to write their opinions, such as imdb.com for movies, amazon.com for products, books, etc., booking.com for hotels, tripadvisor.com for holidays experiences, etc. Twitter is also one of the important resources that users engaged to write their opinions in different topics. In addition, it provides an enormous real-time data in short text, which called a tweet. Sentiment analysis can be applied to many areas, such as marketing, product or service reviews, emotional detections, overall contextual polarity, etc. It uses natural language processing, computational linguistics, feature selection or extraction and other machine learning algorithms.

Sentiment analysis has been covered by many researchers in the past twenty years and various approaches have been applied on the sentiment classification. The task has been studied using different machine learning methods both supervised and unsupervised algorithms alongside using different features. In the supervised learning technique, the dataset is labelled into classes, such as in our proposed dataset there are two classes which are positive and negative. It is highly recommended to filter the text from punctuations, numbers, etc. if they do not affect the meaning. Filtering the text reduces the noise in the dataset and improves the classification (Haddi et al., 2013; Saif et al., 2014; Singh and Kumari, 2016).

In order to do the sentiment classification on a text, the text need to be tokenized into entities such as words entities, characters entities, etc. (Webster and Kit, 1992). Then, filtering and pre-processing the text is an optional step, e.g., removing stop words, removing numbers, stemming and lemmatising the text which will be detailed in Section 3.3.7, etc. After that, selecting or extracting the feature from the tokens using n-grams, part of speech tagging, bag of words and these terms will explain in Sections 3.4.1, 3.4.3, and 3.4.4. Then, these features will be converted to numeric values using some techniques, such as, Term Frequency (TF), Term Frequency Inverse Document Frequency (TF-IDF), etc. and these techniques describe in Section 3.4.5. Finally, these numeric features become inputs for the supervised machine learning methods. The second technique is unsupervised learning and it applies on unannotated dataset. It estimates the sentiment scores using expert knowledge in the format of a lexicon or a dictionary, in which the sentiment words or phrases have a sentiment values. Moreover, there are widely used lexicon for sentiment analysis, such as Affective Norms for English Words (ANEW) (Bradley et al., 1999) which were adapted for measuring the sentiment happiness from the text (Dodds and Danforth, 2011). An alternative method is an application that called Linguistic Inquiry and Word Count (LIWC) and it was firstly launched in 2001, then it developed in 2007 and 2015 (Pennebaker et al., 2015). Also, appraisal groups were used for sentiment analysis (Whitelaw et al., 2005), using syntactic or grammatical structure to detect the sentiment (Sayeed et al., 2012).

Then, many researchers in NLP tasks employed different deep learning models which are advanced machine learning approaches to achieve better classification results. The deep learning approach uses Neural Networks (NN) with extensive layers to solve complex problems. The basic NN consists of input layer, hidden layer, and output layer. Each layer in the NN consists of many numbers of neurons that are connected to each other. The connection between them holds a weighted value to control the learning rate between neurons. The Deep NN takes a raw of vectors as inputs which represent a row of data. In this study, in order to represent the text into vectors, some word embedding techniques are required and the word embedding will be illustrated in Section 3.4.2 and Chapter 5. The word embedding is an approach that used in NLP tasks and it models the features of text into fixed-dimension of vectors. The vectors contain a fixed-size of float numbers, i.e., the vector of the word "Coventry" is [0.56, 0.97, 0.13, 0.83, ..., 0.44]. There are different techniques to build the word embedding, such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014a), RAND-WALK (Arora et al., 2015), fastText (Bojanowski et al., 2017), StarSpace (Wu et al., 2017), etc. Then, the inputs move into the hidden layers and each one computes Equation (2.5):

$$f(x) = f(\sum_{i=1}^{m} w_i.x_i + b) \tag{2.5}$$

where i is number of current input, m is the maximum number of the inputs, w_i is the weight of the current input, x_i is the value of the current input, b is the Bias, and f is non-linear activation function, such as sigmoid (Han and Moraga, 1995), Hyperbolic Tangent Function (tanah) (Kalman and Kwasny, 1992), Rectified Liner Unit (ReLU) (Nair and Hinton, 2010), and Leaky ReLU (Maas et al., 2013), etc. In the output layer, a linear function is used to measure the regression problems because of the unbounded values. In the classification tasks, the softmax function is applied because it measures the probabilities of each input and classify it.

There are enormous numbers of sentiment analysis researches that applied different type of deep learning neural networks. For instance, Convolutional Neural Network (CNN) (Kim, 2014), Recurrent Neural Network (RNN) (Arras et al., 2017), Long Short-Term Memory network (LSTM) (Li and Qian, 2016), Recursive Neural Network (RecNN) (Timmaraju and Khanna, 2015), etc. In addition, applying different sentiment analysis levels and different feature extraction techniques in order to improve the learning results. Kim (2014) proposed a CNN model for sentences classification using unsupervised pre-training word2vec vectors. dos Santos and Gatti (2014) presented a sentiment analysis approach using deep CNN. It considers different representation levels that are character, word, and sentence to implement the sentiment classification. Poria et al. (2016) introduced a deep CNN model based on aspect level to extract the opinions from the text.

There are many complications in this type of analysis, because it deals with text and unstructured data. Furthermore, there are difficulties to detect or identify the effective sentiment words in the text and different context can change the meaning. There are many challenges in this task in general, which will be explained next. This is true especially for Arabic texts, which will be detailed as well in the following section.

2.3.1 Arabic Sentiment Analysis Challenges

There are many different challenges in this task and they might be theoretical, technical or both. Liu (2010) introduced some issues in the sentiment analysis task, such as identifying the objects, extracting the features, grouping the synonyms, determining the opinion words and classifying them, and integrating the previous components. Also, some opinions on social networks are hardly recognisable by a machine search, especially when people disagree with others (Ahmed et al., 2013). Furthermore, subjectivity in sentiment analysis is sensitive and can be positive or negative based on different researchers' interpretations (Ahmed et al., 2013).

There are some specific challenges to sentiment analysis in Arabic. Ahmed et al. (2013) noted some difficulties of sentiment analysis using Arabic text from social media:

- Opinion keywords can be changed according to context and word order, which might indicate contrary opinions (Ahmed et al., 2013).
- There are different forms of a single root in an Arabic word: the verb "play" in English has only four forms ("play, played, plays, playing"), whereas the same verb in Arabic has many forms, like (لعب، لعبت، يلعب، تلعب، يلعبان، تلعبان، يلعبون، يلعبن، العب، العبوا، لاعب... الخ). Table 2.2 clarifies the differences in the forms of a single

verb in Arabic. The second column (Buckwalter Arabic Encoding (Smrž, 2016)) is a pronunciation translation of the Arabic word.

Table 2.2 Some examples of the multiple forms of a single Arabic verb

Arabic word	Buckwalter transliteration	Word type
لَعِبَ	laEiba	Masculine verb - past tense for singular
لَعِبَت	laEibat	Feminine verb - past tense for singular
ىلعَبُ	ylEabu	Masculine verb - present tense for singular
تلعَبُ	tlEabu	Feminine verb - present tense for singular
ىلعباں	ylEbAn	Masculine verb - present tense for dual
تلعَباں	tlEabAn	Feminine verb - present tense for dual
ىلعبوں	ylEbwn	Masculine verb - present tense for plural (three or more)
ىلعبں	ylEbn	Feminine verb - present tense for plural (three or more)
العب	AlEb	Masculine verb - imperative
لاعب	lAEb	Noun - the subject form of the verb

- There are difficulties with sentiment analysis of social media because of the following issues:
  - Unstructured text;
  - Flexibility in word order for Arabic;
  - Slang words and different dialects;
  - Character repetition like "Sooo goooood" in English, which in Arabic is "جمىىىل جداااا".
- There are some special characters included in systems like Twitter:

  - "RT" means retweeted tweet;
  - "#" means hashtag, used to categorise a tweet;
  - "@username" refers to a specific username and it appears when replying to a particular Twitter user or to mention a Twitter user in a tweet;
  - Emoticons or emoji like ☺, ☹, >:, ❤️, 😂 , etc.;
  - Links to a web page, such as http://goo.gl/MJ3v7r.

- Some Arabic users might write Arabic words or sentences on social media using the Latin alphabet, a technique known as Arabizi, Moaarab or Araby. For instance, writing the name Mohammad "محمد" can be transliterated as "m7md".

Alhumoud et al. (2015) adds three other issues specific to analysing Arabic text:

- There are different dialects in Arabic, with widely diverse vocabularies.
- Punctuation and diacritics or short vowels in Arabic text can change the meaning of a word with the same spelling. For example,
  - (يَدرُس) is pronounced in Buckwalter's transliteration as (yadrus) (Smrž, 2016). The meaning of the word is "to learn"
  - (يُدرِّس) is pronounced in Buckwalter's transliteration as (yudri∼s) (Smrž, 2016). The meaning of the word is "to teach."
- There are some conjunctions of contrast such as "but" in English. The phrase or clause can hold two opinions, and there are many in Arabic such as the words "لكن ، بينما ، ... الخ." that in Buckwalter's transliteration are (lkn, bynmA, etc.).

Farghaly and Shaalan (2009) presented several challenges and solutions related to Arabic Natural Language Processing (ANLP). The first challenge is that the translated and transliterated name entity from other language such as "google" can be written as "قوقل" or "جوجل". Another challenge is normalising the Arabic letter Alef with Hamza in the words "إن" and "أن" to Alef without Hamza "ان" that leads to a meaningless word. The third challenge is the homograph, where a word can be classified to more than one part of speech. For example, the word " علم" can be a noun means "a flag", or a noun means "science", or a verb mean "to know". The fourth challenge is the ambiguity in some sentence i.e. "مدير البنك الجديد.", which can mean either "the new manager of the bank"" or "the manager of the new bank".

Abdul-Mageed and Diab (2012) explained the difficulties in annotating many Arabic sentences even by Arabic linguistics experts, because of some words occurring in unfamiliar contexts. Abdulla et al. (2013) mentioned the challenges of dealing and building a lexicon that contains various of negation words in Modern Stander Arabic MSA, such as "لا", " لم", "لن", "ما", and "ليس. Ibrahim et al. (2015) presented the challenge of dealing with old sayings, idioms, or old expressions in the text, such as "افتكرناه موسى طلع فرعون", which hold a negative meaning. Al Sallab et al. (2015) added other challenges to automatically apply the sentiment analysis to the Arabic text, which are the ambiguity and the rich morphology in the Arabic language. Also, there is a lack in the tools that deal with Arabic text in order to pre-process the text (Alwakid et al., 2017).

2.4 Related Work

In this section, there are many terms that used in the sentiment classification. They will be defined in Chapter 3.

2.4.1 General Sentiment Analysis

Roberts et al. (2012) studied the way of detecting the emotions through micro-blogging on Twitter. The emotional corpus has seven different types of emotions: anger, disgust, fear, joy, love, sadness and surprise. The corpus was collected from 14 topics on Twitter: Valentine's Day, Lindsay Lohan, September 11th, the 2012 U.S. Election, Palestinian statehood, the Egyptian riots, the Super Bowl XLV, the 2010 World Cup, Christmas, the DC/NY earthquake, the Emmys, Eminem, the stock market, and the Greek bailout. A baseline method was used for discovering the specified seven emotions on Twitter. A Support Vector Machine (SVM) classifier was used to classify the text into classes. The classification features used were: unigrams, bigrams, trigrams, WordNet synsets, WordNet hypernyms, topic scores and significant words. A 10-fold cross validation was used to validate the classifier model. It is an advantage in this study that has a large number of tweets to identify different sentiment words. However, the diversity of topics leads to a variety of non-sentiment words and this makes more noise in the features and the dataset.

Go et al. (2009) used word feature extractions such as unigrams, bigrams, combinations of unigrams and bigrams and combinations of unigrams and parts of speech tags. Then, three machine learning algorithms were applied to the emotional datasets that were collected from Twitter. Usernames, links and repeated letters in tweets were removed so the size of the corpus was reduced to 45.85% of the original one. A mapping technique was used to map the emotions such as, :), :-), :), :D, and =) being all mapped to :) . The best classifier's accuracy was 83.0%, achieved using unigram and bigram features and the maximum entropy method. The author of the paper removed any tweets that contains both positive and negative emoticons in one tweet. It is better if there are many mixed opinion tweets to create a third class for these tweets and name it i.e. "mixed" class.

Speriosu et al. (2011) used three different available datasets, which were the health care reform (HCR), the Obama-McCain debate (OMD) and the Stanford Twitter Sentiment (STS).

The HCR dataset was divided into training and testing sets. Three different classification processes were used: lexicon-based baseline, maximum entropy and label propagation. The experiment's results improved from 58.1% using the lexicon-based baseline method to 71.2% using the label propagation method with feature-edges and noisy-seed for HCR dataset. The weakness in this paper is that the author used different features selections for different classification process. The lexicon feature was used with the baseline method, the unigram and bigram features were used with Maximum Entropy classifier, and various seed distributed techniques with Label Propagation.

Kumar and Sebastian (2012) presented a new approach to sentiment analysing Twitter data, by extracting opinion words focusing on only adjectives, verbs and adverbs. The approaches chosen were the corpus-based method and the dictionary-based method to find the semantic orientation of adjectives, verbs and adverbs. After finding the opinion words, scoring modules were applied for the adjective group and the verb group. Finally, a linear equation was used to calculate the tweets' sentiment score. The limitation in this paper that the author only considers the adjectives, verbs, and adverb only. However, some nouns and other type of words might have sentiment meaning, i.e., the words "quality", "badness", etc. Also, most of the Twitter users used to not write some words in the tweet in a correct spelling e.g. the word "wuz" which means was, "besty" which means "best", etc. Thus, classifying and recognising these words correctly for sentiment analysis purpose is hard.

Agarwal et al. (2015) used three different datasets: restaurant reviews, software reviews and movie reviews. The datasets were divided into 90% for training and 10% for testing. There were five methods that were used in the experiments: baseline, domain-specific ontology, the importance of the feature, contextual information and a combined method of context information and the importance of the feature. The best method was the combined method and the accuracies were 80.1% for the software dataset, 78.9% for the movie dataset and 79.4% for the restaurant dataset. The strength in this paper is identifying the entities and the aspects in the tweets using ontology. Both ConceptNet and WordNet are used in order to build and expand the features for the sentiment classification. However, the experiments in this paper only rely on the values of the words in the lexicons and the ontologies' relation between these words. Therefore, the values of the words in the lexicons can impact the sentiment score.

Zhang et al. (2011) proposed a method using a SVM to classify four different datasets: Obama, Harry Potter, Tangled, iPad and Packers. The average accuracy of the method for the four datasets was 85.4% and the average F-measure = 74.9%, precision = 68.7% and recall = 82.7%. The author of this paper only used unigram binary feature values with attention to negations. However, other feature extraction techniques should be considered to show the strength of their model.

Jianqiang and Xiaolin (2017) studied the effect of pre-processing the text on the sentiment classification performance. Six methods of pre-processing the text were applied: removing URLs, removing numbers, removing stop words, normalising repeated letters, normalising acronyms to their original, and normalising negative mentions. These methods were applied on five datasets and they evaluated using four classifiers. The study indicated that removing numbers, stop words, and URLs reduce the noise in the datasets. However, normalising negative words and acronyms improve the classification performance. The author of the paper applied the sentiment classifications using three classes only, which are "positive", "neutral", and "negative". In the normalisation phase, the author did not normalise the repeated letters to the original forms i.e. the word "goooooood" is normalised to "goood". This approach is good to discriminate between the class "positive" and "very positive". Thus, the word "goood" has emphatic meaning and it can be classified as "very positive" whereas, the word "good" can be classified "positive" but there is no "very positive" class in this study.

2.4.2 Health Sentiment Analysis

Yadav et al. (2018) collected a dataset that contains opinions about health conditions from the "patient.info" website. Some particular health domains have been considered in this study such as allergy, asthma, anxiety, and depression. There were two classification strategies that were followed in the study which are medical condition and medication. Each one has different labels: for medical condition these are exist, recover, and deteriorate and for medication strategy these are effective, ineffective, and serious adverse effects. The study showed a significant sentiment classification performance for the Convolutional Neural Network model comparing with SVM, Random Forest, and Multi-layer perceptron.

Saif et al. (2012) classified users' opinions on Twitter using semantic features. The experiments used three available Twitter datasets: the Stanford Twitter Sentiment Corpus (STS), the Health Care Reform (HCR) and the Obama-McCain Debate (OMD). Replacement, augmentation and interpolation were the three methods used to incorporate semantic features for sentiment classification purposes. The best result in the experiments came from using the interpolation approach, unigrams as features and Naïve Bayes as a classifier. The report results showed that large datasets are best analysed by semantic methods, while sentiment-topic is the best method for small datasets or limited topics. The author evaluated three semantic concept extraction tools that are AlchemyAPI, OpenCalais, and Zemanta and AlchemyAPI had the best evaluation result. It will be better, if the author combined them together because the paper stated that using other tools have good results.

Sadilek et al. (2012) observed the illnesses of Twitter users who described their conditions by some words related to sickness and health. Data collection, using the Twitter Application Program Interface (API) and the Python programming language, was confined to New York City over a month-long period beginning on 18 May 2010. The SVM algorithm was used to classify the data into two classes of tweets (sick and other). The algorithm was trained using a dataset of 5,128 tweets, after which the classifiers were used to differentiate a set of 1.6

million tweets into either the sick class or the other class. There were three features: unigram, bigram and trigram. Some positive and negative feature weights were also applied to some words, with the weights differing according to the value of a word. This paper presented a prediction technique for illnesses transmission from Twitter data in one geographic location. It will be beneficial to link these data with the data from hospitals and clinics to reduce the spread of some disease using social data.

Aramaki et al. (2011) detected flu in Japan by building a data of flu corpus from Twitter. The data was collected in four seasons (winter 2008, summer 2009, winter 2009 and summer 2010). The data was divided into positive and negative based on two conditions: the tweet's writer or a relative had the flu and the time of the tweet should not exceed the event by more than 24 hours. Several machine learning methods were applied to the corpus, such as Decision Tree, Logistic Regression, Naïve Bayes, SVM, etc. The best results for F-measure was SVM. The approach of this paper is helpful to detect the flu by observing the social media to identify the approximate number of patients. Thus, it helps to provide good treatments and prepare earlier based on the seasons of the flu. On the other hand, the author should have eliminated the tweets that contain news about the flu to improve the performance of detecting the flu, or removed the tweets that have the same news.

2.4.3 Arabic Sentiment Analysis

Ahmed et al. (2013) mentioned several challenges in undertaking subjectivity and sentiment analysis on Arabic text and offered several solutions. The datasets were collected from the Twitter stream API. Four Arabic keywords on politics, sport, technology and religion were considered and the size of the datasets based on the keywords are: 604, 668, 935, and 654 respectively. There were three classes that are positive, negative, and neutral for all datasets. A 10-fold cross validation was used for testing the model. The author of the paper measured the effect of using n-gram features, text pre-processing and normalisation on the classification. Five algorithms used in the experiments were SVM, Naïve Bayes, J48 Decision Tree, Bayesian network and Maximum Entropy. All datasets had the same best results by employing unigram features with normalisation the test dataset. In this paper, the multiple forms of an Arabic word have been treated using a lexicon but the size of the lexicon will be very large to cover most of the words in Arabic. Also, there are two issues: replacing many words and the number of the negative tweets in the sport dataset is very low comparing to the other classes.

Refaee and Rieser (2014) collected an Arabic twitter corpus dataset containing 8,868 tweets. There were five classes that are positive, negative, neutral, polar, and mixed. The dataset contains six different topics about products, sports, internet, social issues, presidents, and organizations. The dataset splits into training and test sets. The training set had 7,503 tweets, whereas the test set had 1,365 tweets. There are only two annotators who labelled the dataset. However, the author should have employed more than two annotators because there are five classes in dataset. The classes are "polar", "positive", "negative", "neutral", and "mixed". Hence, the majority voting can be measured to achieve better classification.

Ibrahim et al. (2015) created a corpus with different types of data (49% tweets, 16% comments on hotels, 18% comments on television programmes and 17% product reviews). The negative dataset had 971 documents and the positive dataset had 1029 documents. The experiments were divided into two parts: in the first part, 80% of the data was used for training, 10% for validation and the remaining 10% for testing. SVM methods were applied on the dataset four times and the results in general improved from the first through to the fourth iterations. The final results were: total accuracy = 95.12%, total precision = 93.15%, total recall 98.55% and total F-measure = 95.77%. This paper considered old saying or idioms which might not contains adjectives or clear sentiment. However, the way of performing that is by translating the phrases using the sentiment meaning instead of the original meaning of the phrase.

Dahou et al. (2016) proposed a web-crawled corpus to build several word representation models for Arabic using the Word2Vec technique. The pre-trained Arabic models were used with Convolutional Neural Network (CNN) for sentiment classification. The sentiment classification results of this models were evaluated using several available Arabic datasets and they were compared with other models. The quality of the data in the pre-trained models increases the classification performance. Also, the performance of high dimension vectors was shown to be more effective on a large corpus. The proposed approach is interesting because it combined the word embedding technique to produce vectors along with Convolutional Neural Networks (CNN).

Hamouda and Akaichi (2013) analysed 260 Facebook posts about the Tunisian Revolution from 01/01/2011 until 01/06/2011. The data was only taken from Tunisian users. Three lexicons were built (acronyms, emotions and interjections) and selected features extraction was used: bag of words, n-grams and parts of speech tagging. SVM and Naïve Bayes methods were used to classify the data, with the accuracy are between 71.33% and 75.35%. The size of the dataset is small and it would be better to apply sentiment classification to other large datasets.

Salamah and Elkhlifi (2014) created a corpus from Twitter that contains of 340,000 tweets about political topic in Kuwait for two years. There were four main steps for this work: building the dataset from Twitter, pre-processing the tweets, extracting opinion words based on Kuwaiti dialect, classifying the dataset. There were only two classes that are positive and negative and the classifiers were decision tree and SVM only. The average results of the classification using precision and recall were 76% and 61% respectively. The author employed three native speakers to annotate 340,000 tweets manually, which is a very large number. It may be better to use some machine learning to predict unlabelled tweets and to measure the efficiency of these techniques.

Abdul-Mageed and Diab (2012) proposed a subjectivity and sentiment analysis corpus for Modern Standard Arabic (MSA), which is called AWATIF. It was collected from three different sources. First, Penn Arabic Treebank, which is an available collection of news documents in various topic, i.e., economic, sports, etc. Second, Wikipedia Talk Pages, which was extracted from Wikipedia editors talk pages. Third, Web Forums, which contains conversations from different seven sites. This paper used two interesting ways of annotating the corpus. The first is a basic way by training the annotators that there are three classes (positive, negative, and neutral). The second is labelling the sentences using annotators that are experts in the linguistic field. This paper presented the importance of labelling the corpus and how that can affect the sentiment classification. The variety of topics and different sources that the corpus was collected from might affect the sentiment classification without using a lexicon.

Nabil et al. (2015) introduced an Arabic Sentiment Tweets Dataset ASTD and applied different machine learning classifier to do the sentiment classification. The author of this paper collected over 84,000 tweets using two methods that are: collecting the most recent tweets from the most active 30 Egyptian accounts, and collecting tweets from Twitter trending hashtag in Egypt. About 36,000 tweets were gathered using the first technique and over 48,000 tweets were obtained using the second approach. Around 10,000 tweets were labelled manually using the Amazon Mechanical Turk service through Boto API. The dataset has four classes that are subjective positive, subjective negative, subjective mixed, and objective. The sentiment classification applied on the dataset using different n-gram tokens and different ML classifier e.g. Naïve Bayes, SVM, Logistic Regression, etc. The best classification results were using Multinomial Naïve Bayes for balanced dataset and SVM for unbalanced dataset. The author translated the Arabic tweets to English in order to annotate the tweets manually using Amazon Mechanical Turk service. Thus, there was an issue in translating the tweets from the Egyptian dialects to English because this could affect the accuracy of the translation process.

Abdulla et al. (2013) created an Arabic Twitter dataset for sentiment analysis on multiple topics, such as politics, economic, etc. The dataset contains 2000 positive and negative tweets that divided into 50% for each class. These tweets contain opinions written in MSA and the Jordanian dialect. Supervised and unsupervised sentiment analysis approaches were used in this paper. In the supervised experiment, four machine learning classifiers were used: SVM, NB, Decision-Tree, and KNN. Using Arabic light stemmer with SVM and NB leads better results compared with root stemmer and original forms of words. In the unsupervised technique, the author used three different sizes of lexicons with the classification and the large lexicon indicates a high accuracy in the classification performance. Microsoft Word was used to correct the spelling mistakes by choosing the first option presented. Note that, this may not be the correct option as there are words that were written in the Jordanian dialects. The author dealt with repeated letters when it occurs more than five times only; it is better to reduce the number to three.

Aly and Atiya (2013) built LABR, the Large Scale Arabic Book Reviews dataset for sentiment classification and it was collected from goodreads.com website. The reviewers in the website can rate the book from one to five stars and to write their reviews. Over 220,000 reviews were collected from the top 2143 Arabic books in the website. The reviews were filtered and the final number of reviews in the dataset was about 63,000. There were two strategies for labelling the reviews. The first is rating classification based on the number of the stars for each review. The second is sentiment polarity classification which converts the number of stars into positive, neutral, and negative. The positive class involves four and five stars reviews, the neutral represents three stars only, while the negative indicates one or two stars rating. The paper presented different experiments using balanced and unbalanced dataset. The weighted feature techniques were using TF-IDF or binary with different n-grams.

The classifiers that were used are Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), and Support Vector Machines (SVM). In the sentiment polarity classification and the rating classification, the best classification technique were MNB using TF-IDF with 1-gram + 2-gram + 3-gram for balanced dataset and SVM for unbalanced dataset. It can be recommended that the author should collect the reviews from the top, middle, and bottom books in the list of the best Arabic books. Hence, the variations between the classes will be less than the presented dataset.

Al Sallab et al. (2015) applied four different deep learning techniques for Arabic sentiment analysis. The models are: Deep Neural Networks (DNN), Deep Belief Networks (DBN), Deep Auto Encoders (DAE), and Recursive Auto Encoder (RAE). The inputs for DNN, DBN, and DAE are bag of words based on the score of Arabic sentiment lexicon, while the inputs for RAE are word embedding vectors. The study confirmed that the RAE model has the highest sentiment classification performance compared to other models. The issue in this paper is that the inputs are different and the comparison is unfair. As a result, the reason of improving the sentiment classification is not clear since it does show whether the role of using word embedding technique as input or the RAE model itself. The author should have applied the word embedding to all other models and justify the results.

2.5 Summary

This chapter defines the term of social networks in general and Twitter in particular. Then, it describes various techniques to analyse the social networks, i.e., Density, Clustering, etc. Also, it introduces and reviews the sentiment analysis and it provides Arabic sentiment analysis challenges. Finally, it describes related works to sentiment analysis in general, about health sentiment, and Arabic sentiment analysis. In the next chapter, a methodology is presented in detail.

Chapter 3

Methodology

3.1 Overview

Twitter is an accessible microblogging platform for public users, where the users can write their tweets about any topic in a limited number of characters. Lee et al. (2011) classified Twitter trending hashtags into 18 different topics such as, politics, business, sports, food & drink, health, etc. Here are some examples of tweets related to health: "@itsmacotaco: stomach flu is the worst", "@thorpeal: i [sic] hate cancer so much" and "@raeanne_genth: Winter is bad for my health i [sic] need warmth".

Twitter's millions of users can provide real-time information about their health. Therefore, investigating this information is easier, cheaper and potentially more powerful than collecting information from clinical resources. It is a valuable resource for research. For example, Lampos and Cristianini (2010) presented a method for tracking the spread of the flu epidemic in the UK using Twitter and compared the data with data from the Health Protection Agency. Ritterman et al. (2009) predicted the potential occurrence of a swine flu pandemic using context features from tweets. Aramaki et al. (2011) proposed a novel method for discovering influenza epidemics from Twitter by using Natural Language Processing (NLP).

Figure 3.1 shows the main four stages in this research, which are collecting the tweets from Twitter, filtering and pre-processing the dataset, annotating the dataset, and applying different feature selection and machine learning algorithms for classification. The proposed dataset has only two classes that are positive and negative. Therefore, the sentiment classification experiments in this thesis are binary classifications.

Firstly, this chapter briefly explains all the possibilities for retrieving tweets from Twitter. Then, the dataset filtering and pre-processing steps will be briefly explained. Also, several terms regarding the feature selections and data pre-processing in a sentiment analysis will be clarified briefly. Lastly, the dataset labelling and measuring the machine learning performance will be outlined.

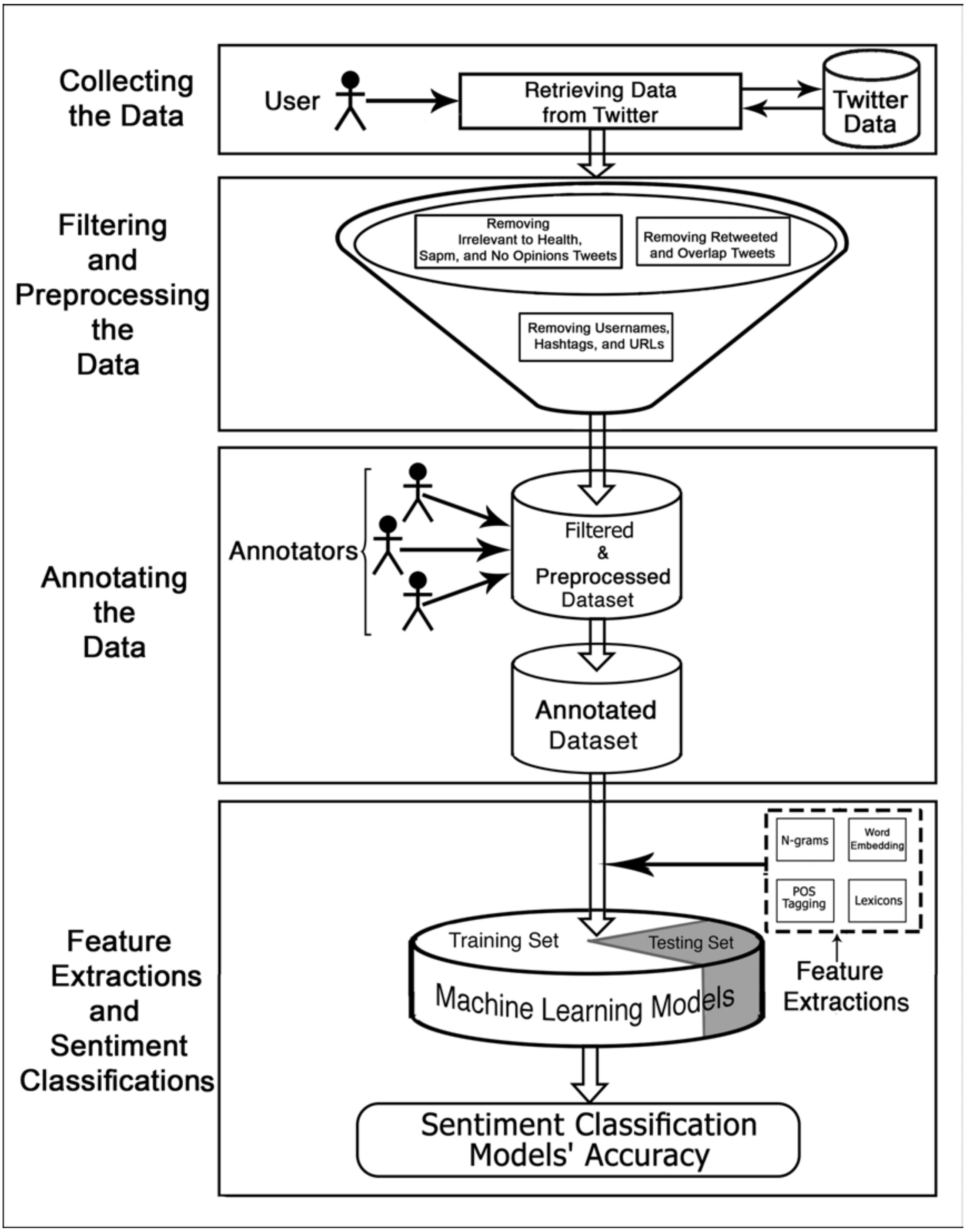

Collecting the Data
User
Retrieving Data from Twitter
Twitter Data
Filtering and Preprocessing the Data
Removing Irrelevant to Health, Sapm, and No Opinions Tweets
Removing Retweeted and Overlap Tweets
Removing Usernames, Hashtags, and URLs
Annotating the Data
Annotators
Filtered & Preprocessed Dataset
Annotated Dataset
Feature Extractions and Sentiment Classifications
N-grams
Word Embedding
POS Tagging
Lexicons
Feature Extractions
Training Set
Testing Set
Machine Learning Models
Sentiment Classification Models' Accuracy

Fig. 3.1 The workflow of this project and the four main steps

3.2 Data Collection

There are two different ways of retrieving the Twitter data: Crawler and Twitter API. In this study the tweets will be collected using Twitter API, which can be accessed from Twitter developers. The reasons for using Twitter APIs are the ease of use, the availability, and the quality of the results. APIs in Twitter are of three types:

- **REST APIs**

  Representational State Transfer-Application Programming Interface (REST APIs) enable the ability to read data from Twitter or to write data in Twitter by using programming codes. These APIs allow the user to perform only a single task such as writing a new tweet, reading a user's profile or discovering the data of a user's followers.

- **Streaming APIs**

  Streaming APIs allow the user to retrieve continuous data from Twitter. Streaming data can be tailored to a search term or a specific user. There are three types of streaming APIs: public streams, user streams, and site streams.

- **Ads APIs**

  There are two different types of Twitter accounts: advertising accounts and user accounts. The Ads API is an approach to connect to an advertising account by programming. The benefit is that it allows marketers to promote their advertising tweets to Twitter users, with appropriate levels of targeting.

Twitter APIs have functions such as GET, POST and DELETE and several objects like users, tweets and places. In order to use Twitter APIs, a user must register an application to obtain these four information: *consumer key, consumer secret, access token* and *access secret*, which are the parameters of *OAuth* functions. In this research, Twitter Streaming

APIs will be used. Twitter only allows the retrieval of tweets that have been written within the past seven days. However, after collecting the Arabic Health Services dataset, the Twitter Premuim API has been introduced to retrieve historical tweets from Twitter without any time limtits. The Python 3.4 programming language and the tweepy package version 3.1.0 can connect to Twitter APIs to retrieve the tweets. In order to collect tweets on a particular topic, it is necessary to specify all words relevant to the topic to collect all tweets that contain those words. As the focus of this study is health services, the key terms will include "health", "hospital", "clinic", "disease", "medicine", and other such terms. Moreover, to retrieve only Arabic tweets requires encoding words to UTF-8 characters and the UTF-8 unicode codes were taken from Wood (2015). Table 3.1 shows the words in English, Arabic and the UTF-8 encoding.

Table 3.1 Examples of words related to health in English, Arabic and the UTF-8 encoding

English Word	Arabic Word	UTF-8 Encoding
Health	صحه / صحة	u'\u0635\u062d\u0629' / u'\u0635\u062d\u0629'
Hospital	مستشفى	u'\u0645\u0633\u062a\u0634\u0641\u0649'
Clinic	مستوصف	u'\u0645\u0633\u062a\u0648\u0635\u0641'
Disease	مرض	u'\u0645 \u0631\u0636'
Medicine	علاج	u'\u0639 \u0644\u0627\u062b'

Not all the collected tweets had opinions. Most of them were news tweets which are not relevant to this study. Subsequently, in order to avoid the complexity of converting the Arabic characters to UTF-8 encoding, the R Programming language and "twitteR" package has been used because of its support in using Arabic characters (Gentry, 2016). The way of collecting tweets with opinions in health was by observing the trending hashtags related to health; all the details will be explained in Chapter 4 (Collecting and Annotating a Health Sentiment Dataset in Arabic).

3.3 Data Filtering

After retrieving the tweets from Twitter to build the dataset, the tweets need to be pre-processed by ridding the content of undesirable words, for example, username, unrelated hashtags, URLs, photos and non-Arabic words. Following previous examples, words are replaced with special tags. For instance, all different usernames which appear after the symbol @ are replaced by the word "username", and all the words that follow the symbol # are replaced by the word "hashtag". The URL links are similarly identified by finding the tag "http:" and the URLs are changed to the word "URL". In addition to pre-processing, the text might need to be amended for spelling mistakes or repeated letters or words and the text might require filtering by removing unrelated tweets and conducting normalisation.

The filtering or pre-processing phase is essential in order to improve the classification. Also, reducing the forms to a single word is helpful, because if there is any misspelling in a word the classifier will treat it as a different word. Therefore, several common pre-processing steps will be detailed in the following sub sections.

3.3.1 Stop Words

Stop words are words that do not have any impact on the sentimental meaning of the sentence. Examples of English language stop words include prepositions (on, in, at, about, etc.), forms of the verb to be (am, is, are, was, etc.), auxiliary verbs, pronouns, etc. Furthermore, the Arabic language has different stop words such as (مں ، ڡى ، على ، إلى ، حول ، الدى ، التى ، الدس ، اللاتى ، الخ). These types of words can occur in any the sentence and they increase the size of the vocabulary in the dataset, and thus create more computational difficulties, but they do not affect the sentiment.

3.3.2 Repeated Letters or Repeated Words

The maximum number of characters in a tweet is 140 including spaces, so users face a limited text length in writing a tweet. Despite this fact, users may well repeat some letters or words. In particular, some users emphasise their opinions by repeating letters or words. For instance, the word "ممتااااار" means "excellent" and one letter is repeated four times (alif "ا"). The extra four letters should be eliminated to obtain the original word "ممتار". However, some words like "ملل" ("Malal"), which means "boring", has the same two letters and at the end of the word and they can be replicated as "مللللل"; in this case there are four extra letters and there are in total six of the same letter. In the first case, the word can be corrected automatically, whereas the second example needs extra care to be edited automatically. Other users might duplicate words, and they might use some sentiment words to emphasise or to indicate the depth of feeling of a repeated word, such as "جميل جميل جميل", which is the word "beautiful" repeated three times. Users might also iterate adverbs like "very" or "much", which in Arabic is "جداً جداً جداً".

3.3.3 Normalisation

Ahmed et al. (2013) defined normalisation as converting a word to its original character by deleting all non-letters, punctuation, diacritics, etc. Different normalisation techniques for the Arabic language will be detailed below:

1. Delete punctuation from the text, such as “” , : ; .

2. Delete short vowels (diacritics) from the Arabic text, which are the marks above and below the Arabic letter (ا) in these examples اَ اً اِ اٍ اُ اٌ اْ اّ .

3. Delete special characters from the text, such as § ! ? > < $ %.

4. Delete any tatweel which extends the length between two linked Arabic letters. For example, the original form of the word “صحه” with tatweel becomes "صـــحــــه" or "صــــحه" or "صحـــــه".

5. Change the "alif" letter from the three forms “أ”, “إ” and “آ” to just one form “ا” .

6. Change the "waw" letter forms “وء” and “ؤ” to “و” .

7. Change the "yea" letter forms “ى ”, “ ئ” and “ ىء” to “ى” .

8. Change the "tea marbota" letter “ ة ” to “ ه ” .

3.3.4 Words with Similar Meanings

All the words in the group "رائع، مدهل، جمل، بديع ... ألخ" are similar in meaning ("beautiful, marvellous, gorgeous, etc."). These words are positive in meaning. However, if a rating system like a scale from one to five is used, it is difficult to give each word a value. The value of one in the scale means the worst or most negative sentiment, while the value of five is the best or most positive opinion. Abdulla et al. (2013) state that building a lexicon that contains all the root words, both positive and negative, is helpful and was employed with an unsupervised machine learning approach.

3.3.5 Arabic Homonyms Words

Homonyms mean words that have similar or the same spellings but the meaning is different, and there are many examples of such words in Arabic. For example, the word “ عقد ”, which in Buckwalter transliteration is "Eqd" (Smrž, 2016). It consists of the three Arabic letters "Ain", "Qaf", and "Dal". It has many meanings and Table 3.2 shows the differences. Similarly, “ صحيح ” means “true” or "correct", but can also mean integer in mathematics. In this project, many hospitals and clinics are named after personal names, geographic names or other words that are widely shared. Moreover, the word “health” in Arabic is “صحه / صحة”, which can have a different meaning when it links with another word like “news”; in that case it means “incorrect or inaccurate news”. Table 3.3 shows the different meanings of the Arabic word صحة. Where the first example means health but the second example means incorrect news.

Table 3.2 The multiple meanings of one Arabic word

Word in Arabic	Part of Speech	Buckwalter transliteration	The translation
عَقْد	Noun	Eaqod	Agreement, Lease, or Contract
عَقْد	Noun	Eaqod	Fastening, Tying, or Knitting
عَقْد	Noun	Eaqod	Decade, or 10 years
عَقَد	Verb	Eaqad	Held, or Make
عَقَّدَ	Verb	Eaqa~da	Snarl, or Make Difficult
عِقْد	Noun	Eiqod	Necklaces

Table 3.3 An example of an Arabic word that has different meanings in two tweets

The Tweet in Arabic	Related or Not
"@drfaisalalmalki: حطر بطاطس الشبس على # صحة الأطفال تحتوى على مادة أكريلامايد و إحتمالةالسرطنة و صار بالجهارالعصبى #توعية # صحتك _ تهمنا https://t.co/ckU5SQWtA8"	Related to Health
"@msdmrrsk: كلمة رسميا فى التغريده تقلل من مصداقة صاحب الحساب فى حالة عدم صحة الخبر فأحذروا ياقوووم من كتابة رسميا قبل التأكد من صحة الخبر	Unrelated to Health

3.3.6 Spelling Mistakes

Each tweet should be checked manually for any misspellings due to fast typing or the users' spelling weaknesses. Correcting spelling mistakes is helpful for the analysis of words that indicate positive or negative opinions.

3.3.7 Stemming and Lemmatising

Jivani (2011) states that both stemming and lemmatising have the aim of minimising a word's forms into a short or root form. There is a slight difference between the two in their manner of shortening words. Stemming involves removing the suffixes or prefixes of the words in some cases, and cutting the last letters of the words in others. Stemming does not consider the part of speech or position of a word in a sentence. For instance, the stemming of ("stemmer, stemming and stemmed") is "stem"; ("argue, argued, arguer and arguing") is "argu" and ("arguments") is "argument". Lemmatisation, however, means removing suffixes or grouping different forms of a word into the root. A word's part of speech is important in lemmatisation. Examples of lemmatising are (walking, walk), (better, good) and (meeting, if it is a noun, will remain "meeting" and if it is a verb, it will be "meet").

These techniques can be applied to the text in Arabic as well. The Arabic language morphologically and orthographically are rich. There are many forms of a single Arabic word that can be derived from its root. According to Al-Kabi et al. (2015) there are two approaches of stemming Arabic text that are light stemming and heavy or root-based stemming. The light stemming is to eliminate the affixes only from the word. The heavy stemming is to convert a word to its root. Al-Nashashibi et al. (2010) suggested the use of rule-based light stemmer and a pattern-based infix remover to deal with geminated words, hamzated, and eliminated-long-vowel in Arabic. Al-Shammari and Lin (2008) proposed an Arabic light lemmatisation method to remove the prefixes, suffixes, and vowels characters from Arabic words. El-Shishtawy and El-Ghannam (2012) introduced a root-based lemmatisation technique for Arabic and it generates the POS tagging of the word to produce the word lemmatisation. For example, the word "سينتظرونهم" has more than one prefix and suffix and it means "they will wait them". Table 3.4 illustrates the Arabic word with all types of affix.

Table 3.4 An example of an Arabic word with prefix and suffex

Prefix		**Core**	**Suffix**	
Antefix	**Prefix**		**Suffix**	**Postfix**
سـ	يـ	نتظر	ـونـ	ـهم
A letter means “will”	A letter indicates the present tense and the person of conjugation	It is not canonical form of Arabic word whereas the canonical form is انتظر which means “waited” or “to wait”	Termination of conjugation for plural	A pronoun meaning “them”

The light stemming of the word is "ينتظرون" and the root-based stemming is the core word "نتظر" which is meaningless in Arabic. On the other hand, the light lemmatization is similar to the light stemming form "ينتظرون" but the root-based lemmatisation is different than the root-based stemming which is "انتظر".

3.4 Text Features

There are several feature selection methods used for NLP. The features in text classification can be a sentence, a word, a character, or vectors based on these. The tokenisation technique can split the text into sentences, words and characters, and there are different techniques to select the features. Some of these feature selection approaches are summarised in the following subsections.

3.4.1 N-grams

Song and Croft (1999) define the n-gram technique, which measures the probabilities of a word's occurrence in a sentence, as used in the information retrieval field. It is similar in the text classification field, and it groups a number of words next to each other in a sentence. It has several sub-types:

- **Unigram (1-gram)**

  This technique assumes that the occurrence of the word is independent and does not consider other words in the context. Also, this technique can consider a single character only at the character level. Taking this sentence as an example "I study at Coventry University"; the 1-gram for word level is ('I', 'study', 'at', 'Coventry', 'University'). The 1-gram for character level is ('I', ' ', 's', 't', 'u', 'd', 'y', ' ', 'a', 't', ' ', 'C', 'o', 'v', 'e', 'n', 't', 'r', 'y', ' ', 'U', 'n', 'i', 'v', 'e', 'r', 's', 'i', 't', 'y')

- **Bigram (2-grams)**

  This technique considers the sequence of two words together as one feature. The word level 2-gram example based on the previous sentence is ('I study', 'study at', 'at Coventry', 'Coventry University'). Also, the example of 2-gram based on character level is ('I ', ' s', 'st', 'tu', 'ud', 'dy', 'y ', ' a', 'at', 't ', ' C', 'Co', 'ov', 've', 'en', 'nt', 'tr', 'ry', 'y ', ' U', 'Un', 'ni', 'iv', 've', 'er', 'rs', 'si', 'it', 'ty').

- **Trigram (3-grams)**

  This technique is similar to the bigram model, but the trigram uses three words next to each other as a single feature. The examples of 3-grams using the same sentence are for word level ('I study at', 'study at Coventry', 'at Coventry University') and for character level ('I s', ' st', 'stu', 'tud', 'udy', 'dy ', 'y a', ' at', 'at ', 't C', ' Co', 'Cov', 'ove', 'ven', 'ent', 'ntr', 'try', 'ry ', 'y U', ' Un', 'Uni', 'niv', 'ive', 'ver', 'ers', 'rsi', 'sit', 'ity').

4-grams, 5-grams, etc. can also be used for this study. They use the same techniques of grouping the words or characters together as explained previously, but the differences are in the number of words or characters.

3.4.2 Word Embedding

Yu et al. (2017) defined the term word embedding as a technique to measure the semantic and syntactic meaning of a word by leveraging information from a large text. This method has been widely used for NLP tasks. Also, it is known as word representation, because all the words in the large corpora are represented in different dimensions as vectors. There are some prominent examples of word embedding, like Word2Vec introduced in Mikolov et al. (2013), GloVe presented in Pennington et al. (2014a), and fastText Bojanowski et al. (2017). There will be more details on word embedding provided in Chapter 5 (Word Embedding Models for the Arabic Language).

3.4.3 Part of Speech Tagging (POS tagging)

Owoputi et al. (2013) defined POS tagging as a model that arranges and tags chains of words based on the structure of the sentences. It categorises a word by its class such as noun, verb, adjective, adverb, conjunction, pronoun, etc. There are many different types of softwares that are able to perform the POS tagging automatically for English. However, there are few tools to do the POS tagging for Arabic text. We found in our experiments that the *Stanford CoreNLP – Natural language software* (Manning et al., 2014) is the most accurate one.

3.4.4 Bag of Words

Lebanon et al. (2007) state that the bag of words model generates random words to represent them as features for the machine learning classifiers. It is a simple approach and it does not consider any specific type of word, all the words are treated the same. This technique has shown great results in NLP tasks and information retrieval, such as in Argamon et al. (2007), Yang et al. (2007), etc.

3.4.5 TF and TF-IDF

TF is an abbreviation of Term Frequency. Manning et al. (2008) define TF as a numeric score measurement based on the frequency of a specific word in a document. TF is denoted by $tf_{w,d}$ and is measured as in Equation (3.1).

$$tf_{w,d} = \frac{T_w}{T_d} \tag{3.1}$$

where T_w is the total occurrence of a word w in a document d and T_d is the total number of words in the document d. In this technique, some stop words and propositions will have a large value because these words commonly occur more than other words. In contrast, TF-IDF represents Term Frequency Inverse Document Frequency, which is a score of frequent occurrences of a word w in a document d affected negatively by the numeric score of the word in all the documents. The Inverse Document Frequency (*idf*) of a word w is calculated using Equation (3.2).

$$idf_w = \log \frac{N}{df_w} \tag{3.2}$$

where N is the total number of documents in a corpus and df_w is the total number of documents in a corpus that contain the word w. Therefore, the formula of TF-IDF is represented by Equation (3.3).

$$tf - idf_{w,d} = tf_{w,d} \times idf_w \tag{3.3}$$

For example, let us assume the occurrence of the word w in the document d is 2 and the total number of words in the document d is 20. The collection of documents contains 1000 documents and only 10 documents that have the word w. Therefore, the TF = $\frac{2}{20}$ = 0.1, IDF = $\log\frac{1000}{10}$ = 2, and the TF-IDF = 0.1 × 2 = 0.2.

3.4.6 Stanford Tokenizer

Stanford CoreNLP is a natural language processing tool that deals with several languages, e.g., Arabic, English, Chinese, French, etc. This tool provides many functionalities to deal with text i.e. *Text Tokenization*, *Lemmas*, *Word Dependency*, *Part of Speech*, etc. These functions are built in *Java*, but it can be used in other programming languages, such as *Perl*, *PHP*, *C#*, *R*, and *Python*. We will use Python to run these functions and in order to do that we will connect Python with the Stanford CoreNLP server using some packages e.g. *py-corenlp*, *pynlp*, etc.

Text tokenization is one of the main pre-processing steps for many NLP tasks and sentiment analysis is one of them. A text tokenizer divides the text into a series of tokens, such as sentences, words, etc. The Stanford Tokenizer is capable to work well with text in several languages even with a language that does not require a space between two words. It works with English, French, and Spanish languages as word tokenizer which divides the text into words, while it works with Arabic and Chinese like a word segmentation. The Arabic tokenizer processes the Arabic text according to the segmentation techniques (Monroe et al., 2014). For example, the following sentence which contains 11 words

ستكون حدماتنا الصحة فى أفصل مستوياتها حسما يلترم كل الموطفس بمهامهم.

will be tokenized using the Stanford Tokenizer as the next line:

س تكون حدمات نا الصحة فى أفصل مستويات ها حسما يلترم كل الموطفس ب مهام هم

In the tokenized sentence, the number of tokens becomes 16 and there are seven words that have no changes. Whereas, there are four words that were affected by splitting the prefix, suffix, or both from them. The words ستكون becomes (سـ and تكون), حدماتنا becomes (حدمات and نا), مستواباتها becomes (مستو بات and ها), and بمهامهم becomes (بـ, مهام and هم).

3.5 Annotating the Corpus

The dataset will contain many tweets that have opinions about health services. In order to classify the dataset using different machine learning algorithms, the dataset should be labelled first. Therefore, there will be three human annotators to classify all the tweets and all of the annotators will judge each tweet based on their opinion. The annotators will classify each tweet as positive or negative only, in order to apply several supervised machine learning algorithms later. More details are given in Chapter 4 (Collecting and Annotating a Health Sentiment Dataset in Arabic).

3.6 Machine Learning Algorithms

Murphy (2012) defined machine learning as a set of algorithms that can automatically build a pattern from the dataset features. It then checks the association between them to predict future data, classify the data, help in making decision, etc.

- **Supervised Approach:**

  This approach aims to discover the linkage between input attributes and a target class in the dataset. It trains classifiers or algorithms with a training dataset, and then applies the trained classifiers to another dataset, which is called the test set. For instance, Naïve Bayes (NB), Support Vector Machine (SVM), Maximum Entropy (MaxEnt), etc. (Maimon and Rokach, 2005)

- **Unsupervised Approach:**

This approach aims to group the dataset into different categories based on similarities between the dataset elements. This approach clusters data without explicitly training or learning. Examples of this approach are hierarchical clustering, K-means, mixture models, etc. (Ghahramani, 2004).

- **Semi-supervised Approach:**

  This approach is a combination of supervised and unsupervised learning approaches. The algorithms in this approach are applied on both labelled and unlabelled data. Labelled data is supervised information that has associations between inputs and a target. In contrast, unlabelled data has unknown classes or unsupervised information. Examples of this approach include generative models, Low-Density Separation, Graph-Based Methods, etc. (Chapelle et al., 2010).

There are many machine learning algorithms, the algorithms used in this study are described next.

- **Naïve Bayes (NB):**

  NB is a generative method that uses Bayes' theorem (McNamara et al., 2006) to measure the independence between pairs of features (Kirk, 2014). It is a probabilistic classifier. The class that has the highest probability is the most probable class. This is also known as Maximum A Posteriori (MAP) (Zhang et al., 2009). NB has different types of methods: Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes. NB has been used in many sentiment analysis studies, such as Bifet and Frank (2010), Narayanan et al. (2013), and Gamallo and Garcia (2014).

- **Support Vector Machine (SVM):**

  SVM was introduced in Boser et al. (1992) and it is a separating hyperplane in the space of features of the data. It is a powerful discriminative classifier for binary classification and it is used for both classification or regression issues (Cristianini and Shawe-Taylor, 2000). After training the classifier, the model's goal is to build an optimal hyperplane to classify the test examples. The best decision boundary is the one that has the maximum distance (margin) from both classes. Support Vector Machine models have different kernels, which are linear, polynomial, radial basis function (RBF), and sigmoid. SVM has been widely used in sentiment analysis, such as Go et al. (2009), Pak and Paroubek (2010), Jiang et al. (2011), and Bermingham and Smeaton (2011).

- **Logistic Regression (LR):**

  LR is a classification algorithm where the class or the target is categorical. It measures the probability of a given example where it belongs to any class. It uses the logistic function and the shape of it is like the curve of the letter "S". The curve is between the values 0 and 1 (James et al., 2013).

- **Stochastic Gradient Descent (SGD):**

  Pedregosa et al. (2011) stated that SGD is an efficient algorithm for discriminative learning of linear classifiers. It is an iterative algorithm that optimises the objective function. It approximates the gradient descent optimisation to a global or a local minimum (Bottou, 1998).

- **Ridge Classifier (RDG):**

  RDG is a classifier that uses Ridge regression (Pedregosa et al., 2011). The Ridge regression is a technique for analysing a dataset that has multicollinearity (a high-dimensional matrix). This classifier regulates the weights to reduce them to a very small value in order to avoid over-fitting (Owen, 2006).

- **Convolutional Neural Networks (ConvNets / CNNs):**

  CNNs consist of neurons and each neuron has input, weight, and bias. Also, the whole network has a single score function and a loss function e.g. Adam/Softmax on the last layer (fully-connected). However, the architecture CNNs are different to the regular Neural Networks. There are five main layers to build the CNNs which are; Input, Convolutional, ReLU, Pooling, and Fully-Connected (Aghdam and Heravi, 2017).

  The input layer holds the dataset values. The convolutional layer calculates the output of each neuron that is linked to a small region (filter) in the input layer. Each filter reduces the size of the region. ReLU is an abbreviation of Rectified Linear Units and it applies the element wise function to keep the volume's size. The pooling layer applies a downsampling operation. It implements nonlinear functions such as the *max* which is the most common pooling operation. The fully-connected layer computes a single score in order to predict the corresponding class (Aghdam and Heravi, 2017).

- **Long Short-Term Memory Networks (LSTM Networks):**

  A LSTM unit consists of a cell memory and three gates in the LSTMs: the input gate, the forget gate, and the output gate. The input gate is to write the input to the cell, the forget gate is to reset the old cell value, and the output gate is to read the output from the cell (Gers and Schmidhuber, 2001). The main idea of LSTM is to maintain its state over time (Greff et al., 2017). A network that consists LSTM units is called a LSTM network. It advances state-of-the-art techniques for sequential data problems, such as machine translation (Luong et al., 2015), speech recognition (Graves et al., 2013), and visual recognition (Donahue et al., 2015).

3.7 Measuring Classifier Performance

The evaluation of the performance of each classifier using different feature selection methods will be undertaken with the standard measurements of precision, recall, accuracy and F-measure (Manning et al., 2008) . These concepts will be clarified by using the equations and confusion matrix in Table 3.5. True Positive (TP) is the collection of positive data examples that were correctly predicted as positive. False Positive (FP) is the collection of negative data examples that were incorrectly predicted as positive. False Negative (FN) is the collection of positive data examples that were incorrectly predicted as negative, and True Negative (TN) is the collection of negative data examples that were correctly predicted as negative.

Table 3.5 The confusion matrix

	Actual Positive Class	**Actual Negative Class**
Predicted Positive Class	True Positives (TP)	False Positives (FP)
Predicted Negative Class	False Negatives (FN)	True Negatives (TN)

There are different standard measurements for evaluating the performance of a machine learning algorithm (Powers, 2011). They all use some or all the values in the confusion matrix in Table 3.5 to compute the performance of the classifier. The most common measurements are the following.

- **Precision :** is the proportion of correctly predicted examples as positive (*TP*) to the total of all predicted positive examples.

$$Precision = \frac{TP}{(TP+FP)} \tag{3.4}$$

- **Recall :** is the proportion of correctly predicted examples as positive (*TP*) and the total of all of the acutal positive class.

$$Recall = \frac{TP}{(TP+FN)} \tag{3.5}$$

- **Accuracy :** is the proportion of correctly predicted (*TP and TN*) in the total of all the values in the confusion matrix.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \tag{3.6}$$

- **F-measure :** this can also be called (F1 score) and it is the weighted average of Precision and Recall.

$$F - measure = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \tag{3.7}$$

3.8 Summary

This chapter presents the methodological workflow of this research, the methods of collecting Arabic tweets, and some challenges related to collecting the data from Twitter. Then, it describes the steps of pre-processing the Arabic text and the method of labelling the dataset. Also, it describes some Machine Learning Algorithms. Finally, it reports different measurements of the sentiment classification performance. In the next chapter, data collections, filtering, and annotating process are presented in detail.

Chapter 4

Collecting and Annotating a Health Sentiment Dataset in Arabic

4.1 Data Collection Overview

Twitter is an important social media platform, which contains valuable data written daily. This study focuses on tweets that express opinions about health services and are written in Arabic. This chapter will explain in detail all the tools and techniques used for collecting the sentiment dataset from Twitter.

Four main tools were used for collecting, filtering, pre-processing and building the sentiment dataset: the programming languages **R** and **Python** and the software packages **Microsoft Excel** and **LibreOffice Vanilla**. **R** was used to retrieve Arabic tweets and to filter and pre-process the dataset. **Python** was used to do more filtering and pre-processing of the dataset. The retrieved data was exported and saved to CSV files. The data was in Arabic and because of this there was a decoding issue opening the CSV files using **MS-Excel**. **LibreOffice Vanilla** is an application similar to Microsoft Office applications and is available for Mac OS. It can correctly read the Arabic data in the CSV files and it has been used to correct the data formats as Arabic. This chapter details all the procedures that have been taken to collect, filter, pre-process and annotate the dataset. Also, the challenges found at each step will be explained.

4.2 Data Collection Process

The collected sentiment dataset is about health services in Saudi Arabia and is collected from Twitter (Alayba et al., 2017). It was first published in Alayba et al. (2018a) and it is freely available online at Alayba et al. (2018b). The data was collected from 01/02/2016 to 31/07/2016. The dataset contains 2026 Arabic tweets, of which 1398 expressed negative sentiments and 628 expressed positive sentiments. **R** was used to collect Arabic tweets by writing key search words in Arabic without encoding the Arabic character to UTF-8.

In this study, we aimed to collect tweets that contain opinions about health services to do binary sentiment classification. At the beginning of the data collection phase, we used general Arabic words related to health in general, i.e., "hospital", "health", "patient", etc. However, all the retrieved tweets did not have any opinions. In addition, we tried to specify the search by combining these words with some names of the cities in Saudi Arabia, such as "hospital + Riyadh" but the results were the same as the previous approach. Then, we changed the keywords to "health services", "medical services", but there were no retrieved tweets because the Twitter APIs allow to retrieved tweets within seven days only. Alternatively, we modified the keywords to include some names of the hospitals in Saudi Arabia, e.g., "King Fahad Medical City", "Maternity and Children's Hospital", "Al Noor Hospital", etc. However, the retrieved tweets contained only news and non-opinions tweets.

Furthermore, we tend to collect the Twitter data about health from one hashtag only. We launched a hashtag topic especially for this study, asking Twitter users to share their opinions about health services (Opinions about Health) #رأيك _ بالخدمات _ الصحة. However, the number of Twitter users, who got involved in this topic, was very low. The number of tweets, which were collected from this topic, was 3,033 tweets and after filtering them it became 285 tweets. There were many retweeted and duplicated tweets in this topic and the number of deleted tweets was 2,748 tweets. As a result, the number of collected tweets was not appropriate to do the sentiment classification.

Due to the lack of the number in the collected tweets from the previous hashtag, we needed to find another way to collect more tweets about health services in Arabic. The best alternative method is to observe the trending Twitter hashtags where many users involve and write tweets about these topics. We checked the trending hashtags in Saudi Arabia for the period between 01/02/2016 and 31/07/2016 in order to find topics about health. There were three hashtags that raised as trending hashtags in Saudi Arabian Twitter and they are:

- #الصحة _ تعلق _ مستشفى

This topic is about closing a private hospital (Closing Hospital).

- #مں _ ٮعالج _ الصحة

  This hashtag is asking who will resolve the problems in the health services (Resolving Health).

- #ٮٮتطر _ تحسں _ الصحة

  This topic means that people are waiting for improvements in health services (Improving Health).

Table 4.1 summarises the hashtags that we collected the Arabic Health Services AHS dataset from Twitter and the given names for each one.

Table 4.1 The hashtags of the Arabic Health Services (AHS) dataset

The order	The hashtags in Arabic	The giving names
First topic	#الصحة _ تعلق _ مستشفى	Closing Hospital
Second topic	#مں _ ٮعالج _ الصحة	Resolving Health
Third topic	#رأٮك _ بالحدمات _ الصحة	Opinions about Health
Fourth topic	#ٮٮتطر _ تحسں _ الصحة	Improving Health

In order to retrieve tweets from Twitter using Twitter APIs, both a Twitter account and a Twitter developer application are needed. After creating the Twitter application, a consumer key, a consumer secret, an access token and an access secret will be generated. The generated information is used in **R** with a **“twitteR”** package (Gentry, 2016) to set up the API connection using the `"oauth"` function. Also, this package contains the function `"searchTwitter"` to retrieve all the tweets based on the specified search strings (keyword). The keywords in this study were all the four hashtags that mentiond previously. Then, the type of retrieved data is compiled into a list and needs to be converted to a data frame using the `"twListToDF"` function to export the data to a file. The three functions that have been used in the “**twitteR**” package are detailed below:

- `setup_twitter_oauth` has four parameters which are (consumer key, consumer secret, access token and access secret) and it configures the authentication to Twitter servers.
- `searchTwitter` is a function where the key search words or query can be supplied and it might be a concatenation of two key words or more using the "+" sign between words such as “First Word” + “Second Word”. It has many attributes such as "n" for the maximum number of retrieved tweets set to 15,000, and "lang" for the language of the key words. In this case, the four hashtag topics were used as a search key_word.
- `twListToDF` is a transformational function from twitteR lists to data frames which allows the data to be stored as TXT or CSV files.

The retrieved data contains 17 columns, which are: No., text, favourited, favouriteCount, replyToSN, created, truncated, replyToSID, id, replyToUID, statusSource, screenName, retweetCount, isRetweet, retweeted, longitude, latitude, as detailed below.

- **No.:** Numbers of retrieved tweets;
- **Text:** The tweet’s text message;
- **Favourited:** It is a Boolean column which contains true or false. It indicates if the tweet has been marked as favourite or not;
- **FavouriteCount:** The number of times that the tweet has been marked as favourite ;
- **ReplyToSN:** If the tweet was a reply to another tweet, then the name of the user who was replied to will be in this column;
- **Created:** The date and the time of the tweet creation;
- **Truncated:** It contains either true or false only and it will be true if the tweet has been quoted, otherwise it will be false;
- **replyToSID:** It is similar to ReplyToSN, but it contains the id number of the users in this case;
- **id:** It contains the unique number of the tweet;

- **replyToUID:** It contains the unique ids of the twitter user who received the tweeted reply;
- **statusSource:** This column has the information of the tweet source, such as the device, the operation system, etc;
- **screenName:** It contains the username of who wrote the tweet;
- **retweetCount:** It shows the number of times that the tweet has been retweeted;
- **isRetweet:** It is a Boolean column. False if the tweet has not been retweeted and true if it has been retweeted.

All the columns were eliminated except the text column, which contains the tweet's text in order to build the sentiment dataset from the text column only. There is a need to run the code at different times in order to gather all the tweets under the hashtags. Also, any tweets that were created over 7 days before the collection time would not be retrieved. Thus, the process of the collection of each individual topic needed to be run more than once. The reasons for that were either the number of retrieved tweets reached over 15000 (the maximum limit of a single run) or there were new tweets after the last run. After each retrieval of tweets for each topic, the data was stored in a CSV file, and Table 4.2 illustrates the number of saved CSV files for each topic.

Table 4.2 Number of retrieved CSV files

Topic	Number of CSV files
First topic (Closing Hospital)	8
Second topic (Resolving Health)	2
Third topic (Opinions about Health)	7
Fourth topic (Improving Health)	2

Only the first topic (Closing Hospital) reached the maximum number of tweets which is 15,000 in 7 files, whereas other topics did not achieve this number in a single file. The second, third and fourth topics have more than one file because of retrieving the data at different times.

Table 4.3 clarifies the number of tweets for each topic before and after filtering and pre-processing the data.

Table 4.3 The changes in the number of tweets for each topic before and after filtering the dataset and the percentage change

Topic	Number of tweets before filtering	Number of tweets after filtering	Percentage change
Closing Hospital	105,275 tweets	1,009 tweets	-99.04%
Resolving Health	11,624 tweets	492 tweets	-95.77%
Opinions about Health	3,033 tweets	285 tweets	-90.60%
Improving Health	7,027 tweets	240 tweets	-96.58%
TOTAL	126,959 tweets	2,026 tweets	-98.40%

As Table 4.3 shows, there is a huge drop in the number of tweets after filtering them. This is due to the following reasons:

- The tweets of each topic were retrieved more than once at different times, thus many tweets overlapped in different files.
- Retweeted tweets are copies of the original tweets and they start with the two characters “RT”.
- Many tweets were irrelevant to the health topics such as, spam tweets, advertisements, etc.
- There were a lot of tweets that did not have any opinions like neutral tweets, health news tweets, etc.

4.3 Automatic Data Filtering

There are some steps that were followed to filter the tweets automatically. These steps will be explained in the next subsections.

4.3.1 Removing Retweeted Tweets

As has been explained previously, the "RT" in the tweets indicates a retweeted tweet. Python has been used to remove all the retweeted tweets via the following four steps:

1. Reading data from the CSV file.
2. Iterating through each line of data.
3. If the line contains “RT”, then delete this line.
4. Saving updated data to the CSV file.

Table 4.4 indicates the differences in the number of tweets for each topic before and after removing retweeted tweets.

Table 4.4 The changes in the number of tweets for each topic before and after removing retweeted tweets in the dataset and the percentage change

Topic	**Number of tweets before removing “RT” tweets**	**Number of tweets after removing “RT” tweets**	**Percentage change**
Closing Hospital	105,275 tweets	15,736 tweets	-85.05%
Resolving Health	11,624 tweets	3,274 tweets	-71.83%
Opinions about Health	3,033 tweets	769 tweets	-74.65%
Improving Health	7,027 tweets	3,650 tweets	-48.06%
TOTAL	126,959 tweets	23,429 tweets	-81.55%

4.3.2 Removing Overlapping Tweets

There are many of the same tweets that were generated more than once because of collecting each topic more than once, so these duplicated tweets need to be removed. MS-Excel has a function called Removed Duplicates, which can delete any similar cells. Table 4.5 shows the differences in the tweet numbers before and after removing duplicated tweets.

Table 4.5 The changes in the number of tweets before and after removing duplicated tweets for each topic

Topic	Number of tweets before removing duplicate tweets	Number of tweets After removing duplicate tweets	Percentage change
Closing Hospital	15,736 tweets	4,717 tweets	-70.02%
Resolving Health	3,274 tweets	1,930 tweets	-41.05%
Opinions about Health	769 tweets	333 tweets	-56.70%
Improving Health	3,650 tweets	1,831 tweets	-49.84%
TOTAL	23,429 tweets	8,811 tweets	-62.39%

4.3.3 Filtering the Tweets’ Text

Feinerer et al. (2015) introduced the **Text Mining Package “tm”**, which is a natural language processing package for **R**. It has been used to eliminate some words in Arabic Health Services (AHS) dataset such as:

1. Twitter usernames which start with the "@" character and remove any characters following this special character until the next space such as **"@user_name"**.
2. URLs were eliminated, which started with **"http://"** until the next space, which indicates the end of the URL.
3. Some special words that appear in many tweets, such as “**available**”, “**via**”, and some punctuation.

4.3.4 Normalising some Arabic Characters

There are letters in the Arabic language that can have different forms. The Alef letter is a such letter, but it has four different shapes “ أ ، إ ، آ ، ا ”, or the Tea Marbota letter “ ة ، ه ”and many tweets have words containing these letters. We normalised all the Alef letter forms to the form “ ا ”, and the Tea Marbota forms to the form “ ه ”. There are many tweets in the AHS dataset that contain these letters.

4.3.5 Removing Short Vowels (Diacritics)

There are several diacritics in the Arabic language, which can change the meanings of some words even if they consist of the same letters. However, in the corpus, some words have diacritics and they have only one meaning, so the diacritics have been deleted. There are a lot of tweets in the AHS dataset that contain the diacritics. In the following example, the words with diacritics have been underlined.

لآ حآةً لمّن تنآديّ!!صاعت ارواحّ عالىّ وتقهرت قلوب اهلهمّ ولآ شمعآ الا من أسوء الى أرداء
حايل عباره عن فساد اداري

4.3.6 Removing Tatweel

Tatweel is this character “ـ” which is commonly used in the Arabic font art. It can be added between two linked letters so that it does not change the meaning of the word. In contrast, if it exists in a word, the word without Tatweel will be tokenized differently from the word with it. For example, the word “صحه” without the Tatweel character, and “صحـــــه” with the Tatweel character indicate the same meaning denoting health in Arabic. There are some tweets in the AHS dataset that contain the Tatweel. In the following example, the words with Tatweel have been highlighted and underlined.

ليس ب صحيح ان الصحه اعلقته بل المواطن الوزير توفيق الربيعه هو من اعلقـه بِحُسن
إدارته فهنئاً لـا وللوطن وجود أمثالـه # الصحه _ تعلق _ مستشفى

4.4 Manual Data Filtering

It is hard to code a process to eliminate all the tweets that do not have any opinions, such as spam tweets, news tweets, advertisement tweets, etc. Thus, at this stage, the tweets were checked manually to see if they contained an opinion or not. Moreover, at the same time the remaining tweets were checked for spelling mistakes. Table 4.6 shows the changes in the number of tweets before and after filtering them manually.

Table 4.6 The changes in the number of tweets before and after filtering the tweets manually

Topic	**Number of tweets before manual filtering**	**Number of tweets after manual filtering**	**Percentage change**
Closing Hospital	4,717 tweets	1,009 tweets	-78.61%
Resolving Health	1,930 tweets	492 tweets	-74.51%
Opinions about Health	333 tweets	285 tweets	-14.41%
Improving Health	1,831 tweets	240 tweets	-86.89%
TOTAL	8,811 tweets	2,026 tweets	-77.01%

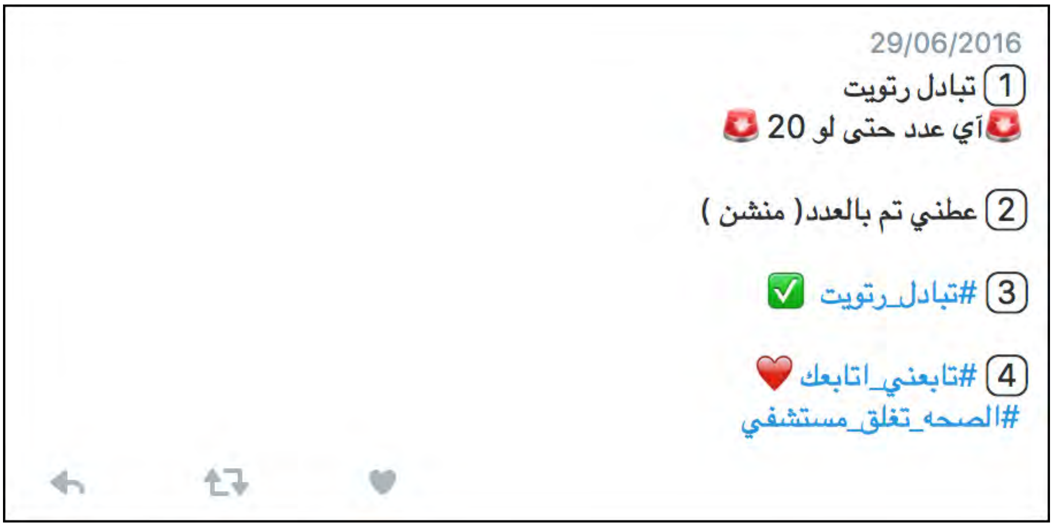
The dataset contains a lot of unwanted tweets like spam tweets, no opinions tweet, unrelated to health tweets, etc. The manual filtering is important to filter the dataset from unwanted data, no opinions tweets, spam tweets, news tweets, etc. These tweets can be classified as neutral, however, in this study, this class was not used. This will increase the quality of the data and help the annotators to label the AHS dataset either positive or negative. Therefore, the filtered data will provide accurate sentiment classifications. This data will negatively affect the results of analysis using machine learning algorithms with binary classification (positive and negative). Some examples of manual filtering are given in the following subsections.

4.4.1 Spam Tweets

Spam tweets are defined as unsolicited, deceptive or repeated tweets that annoy other users (Twitter 2016). They target the trending hashtags which are popular topics to make the spreading of spam tweets easier and quicker. They are hard to detect automatically because there are no specific structures for spam tweets. There are many kinds of spam tweets and here are two examples on a health topic. Figure 4.1 shows a spam tweet that asks for retweets. Figure 4.2 is a tweet that has an advertisement for selling gifts for special occasions. All spam tweets were eliminated manually from the dataset. Table 4.7 presents the differences in the tweet numbers before and after removing the spam tweets.

Table 4.7 The changes in the number of tweets before and after removing the spam tweets

Topic	**Number of tweets before removing spam**	**Number of tweets after removing spam**	**Percentage change**
Closing Hospital	4,717 tweets	1,534 tweets	-67.48%
Resolving Health	1,930 tweets	873 tweets	-54.77%
Opinions about Health	333 tweets	322 tweets	-03.30%
Improving Health	1,831 tweets	454 tweets	-75.20%
TOTAL	8,811 tweets	3,183 tweets	-63.87%



29/06/2016

1 تبادل رتويت
🚨آي عدد حتى لو 20 🚨

2 عطني تم بالعدد(منشن)

3 #تبادل_رتويت ✅

4 #تابعني_اتابعك ❤️
#الصحه_تغلق_مستشفي

Fig. 4.1 An example of a spam tweet

Fig. 4.2 An example of an advertisement tweet

4.4.2 Tweets with No Opinions

The collected dataset had a lot of tweets that did not have any opinions, such as news tweets like in Figure 4.3, or neutral tweets, such as Figure 4.4. Therefore, these tweets were removed from the corpus. Table 4.8 demonstrates the changes in the number of tweets before and after deleteing any tweet without opinions.

Fig. 4.3 An example of a news tweet about topic one

Fig. 4.4 An example of a non-opinion tweet

Table 4.8 The changes in the number of tweets before and after removing the tweets with no opinions

Topic	Number of tweets before removing tweets without opinions	Number of tweets after removing tweets without opinions	Percentage change
Closing Hospital	1,534 tweets	1,022 tweets	-33.38%
Resolving Health	873 tweets	507 tweets	-41.92%
Opinions about Health	322 tweets	298 tweets	-07.45%
Improving Health	454 tweets	257 tweets	-43.39%
TOTAL	3,183 tweets	2,048 tweets	-34.53%

4.4.3 Removing any Opinion Irrelevant to Health

There are several tweets that contain opinions on health services and also opinions on other services, such as education, roads, housing, etc. In this situation, any health opinions have been kept and any non health opinions have been removed. In Figure 4.5, there are two negative opinions, the first one about health, whereas the second opinion, which is highlighted, is about the need to improve roads. There are only six tweets in this dataset that contain opinions irrelevant to health.

Fig. 4.5 An example of an irrelevant oninion in a tweet

4.4.4 Removing #Hashtag_Topics, #Hash and Other Symbols

The first attempt of filtering the dataset was by removing all the hashtags in all of the tweets. However, that led to misunderstandings as some users write positive or negative words inside the hashtag. For example, Figure 4.6 contains the hashtag “#شكرا_توفيق”, which means #thanks_Tawfiq (Thanks to the health minister in Saudi Arabia) and this expressed a positive sentiment in the word "thanks". Therefore, removing all hashtags might remove some sentiment words. However, in this case only the "#" symbol and all the four health hashtags were removed. All the tweets in the dataset have at least one hashtag topic because we collected the AHS dataset using a keyword containing one of the four hashtags in Table 4.1.

Fig. 4.6 An example of a tweet containing more than one hashtag and one of the hashtags contains opinion words

4.4.5 Combining Tweets

When we collected the tweets in 2016, the maximum length of a tweet message was 140 characters. Thus, there are some users who write more than one tweet about one idea. Table 4.9 is an example of several tweets from one user and all of them are about one opinion in the fourth topic, which is #ننتظر_تحسن_الصحة (Improving Health). The user indicated that the tweets are linked to each other by the word “يتبع”, which can mean continue to the next tweet or follow the next tweets. All the tweets were combined into a single tweet and the number of words were reduced by removing all the undesirable words, such as hashtag words, no opinion words, etc. The length of the input layer or matrix for the neural networks will be based on the longest tweet in the dataset. Therefore, other tweets will have the same length of vector as the longest tweet. Thus, the dimensionality of the array's vectors will affect the running time of neural networks (Vanhoucke et al., 2011). Table 4.10 shows the changes in the number of tweets before and after combining the tweets.

Table 4.9 An example of multiple tweets from one user about one idea

No.	Examples of Multiple Tweets in One Idea
1	#يتنطر _ تحسس _ الصحة # صحة _ حائل # حائل # ورارة _ الصحة #توفق _ الربعة دهب العمار سئ وجاء مدر اقبح رمں لمستشفى الملك حالد وقت ادارته ..يتبع
2	#يتنطر _ تحسس _ الصحة # صحة _ حائل # حائل # ورارة _ الصحة #توفق _ الربعة كف أعتلى مصب مدرشؤون صحةمطقة وهوعر قادر على ادرارة مستشفى .. يتبع
3	#يتنطر _ تحسس _ الصحة # صحة _ حائل # حائل # ورارة _ الصحة #توفق _ الربعة مشكلتا لست بتعىر مدر وتتهى برد علاج الرأس حتى يصلح الجسد ... يتبع
4	#يتنطر _ تحسس _ الصحة # صحة _ حائل # حائل # ورارة _ الصحة #توفق _ الربعة تعىر جمع المساعدس ومدراء الادارات والاقسام بالمدرية ... يتبع
5	#يتنطر _ تحسس _ الصحة # صحة _ حائل # حائل # ورارة _ الصحة #توفق _ الربعة احراب وعصرية وتسلط لابد مں فكها ، وبث دماء شابه اداريا ً يتبع ...
6	#يتنطر _ تحسس _ الصحة # صحة _ حائل # حائل # ورارة _ الصحة #توفق _ الربعة لاربد مدر لـ قسم او اداره امصى اكثر مں ه سوات ! يوجد الافصل .. يتبع
7	#يتنطر _ تحسس _ الصحة # صحة _ حائل # حائل # ورارة _ الصحة #توفق _ الربعة ولآ يامعالى الورر ادمج صحة حائل مع القصم شرف لنا مں اما حاعلة مں سو

Table 4.10 The changes in the number of tweets before and after combining the tweets

Topic	Number of tweets before combining tweets	Number of tweets after combining tweets	Percentage change
Closing Hospital	1,022 tweets	1,009 tweets	-01.27%
Resolving Health	507 tweets	492 tweets	-02.96%
Opinions about Health	298 tweets	285 tweets	-04.36%
Improving Health	257 tweets	240 tweets	-06.61%
TOTAL	2,084 tweets	2,026 tweets	-02.5378%

4.4.6 Editing Compressed Text

When we collected the tweets in 2016, the maximum number of characters in one tweet was only 140. Therefore, some Twitter users tended to compress the text. The Arabic language contains 28 letters and most Arabic characters can be linked to the previous and next characters. However, there are several letters in Arabic that cannot be linked to the following letter which are (ا ، ء ، د ، د ، ر ، ر ، و ، ة). When any one of those letters occurs at the end of the word, it will not be linked to the next word without a separated space. As a result, some Twitter users might write a word that ends with any letter from the list, and then they do not split the words using a space to exploit more characters in the tweet. This will cause a word tokenization error because the two or more words will be tokenized as one word, as the tokenization techniques split a word based on the space. This issue was addressed by pre-processing manually. There are 29 combined tokens in 21 different tweets in the AHS dataset. Here is an example of a tweet that had a compressed text.

عدمانصبح هداالوررحدث الناس بالانجاب فهومس نحتاجه ى كل الورارات انتم شهداءالله
ى الارص اللهم وفقه وبارك فه

The highlighted and underlined words are compressed words and Table 4.11 shows the compressed words in the example, the word count and uncompressed words.

Table 4.11 Examples of words merged as one word, the number of merged words and the correction

Compressed words	Number of words	Words after splitting
عدماىصبح	2	عدما ىصبح
هداالورىرحدث	3	هدا الورىر حدث
فهومس	2	فهو مس
شهداءالله	2	شهداء الله

4.4.7 Repeated Letters

There are 14 tweets in the dataset that contain words with repeated letters. These words were normalised to the original form. For instance, this tweet below

كبىىىىىىىىىىىر ىاورىر الصحه الله ىكثر من امثالك ىارب وىرىدك تقدم افصل وافصل
#الصحه _ تعلق _ مستشفى.

In the example, the first word, which is highlighted and underlined كبىىىىىىىىىىىر, has the letter Yia ى that has been repeated many times in the word. The repeated letter has been removed and the word has been normalised to the original form which is "كبىر".

4.4.8 Compound Words

There are a lot of compound words in Arabic and these words might consist of two or three words together where these words are separated by a space. For example, المملكة العربىة السعودىة, which contains three words and مكة المكرمة which consists of two words. The AHS dataset has five examples of compound words. Table 4.12 has some examples of tweets which have compound words, which are highlighted and underlined.

These words will always occur together and, in the tokenization step each word will be tokenized individually. This will not make any sense for the individual word, so the space between these words is removed to combine the words together and, in the tokenization step

Table 4.12 Examples of compound words (highlighted and underlined for each tweet)

No.	Examples of compound words in the tweets
1	ٮأمل من ورىر الصحه رىاره ل مستشفى الملك حالد ڡى حڡر الباطں و المراكر الصحه ٮشوف التسٮب # الصحه _ تعلق _ مستشفى
2	معالى الورىر ٮالت تبدا ڡى مستشڡمات وراره الصحه اول مستشفى الملك ڡهد ڡى المدىٮه المىوره ڡى ٮتطارك # الصحه _ تعلق _ مستشفى
3	لو راروا وادى الدواسر وشاهدوا »المستشفى العام« لتعجبوا أىه لا ىرال شعال حتى هده اللحطة! # الصحه _ تعلق _ مستشفى

will be tokenized as one word. Table 4.13 shows the way each compound word in Table 4.12 was dealt with.

Table 4.13 Examples of compound nouns before and after merging

Original compound words	Compound words after combining
حڡر الباطں	حڡرالباطں
المدىٮه المىوره	المدىٮهالمىوره
وادى الدواسر	وادٮالدواسر

4.4.9 Words from Other Languages

Most Twitter users do not use the modern standard Arabic or formal Arabic. Instead they write their tweets using their dialects and there are many. In addition to that, some Twitter users might use some words that do not originally refer to the Arabic language; there are only two examples of that in the AHS dataset. In Table 4.14, the non Arabic words were highlighted and underlined. The first tweet contains the word “Bravo براڡو” and the second tweet has the word “Check شىكو” and these words have been written using Arabic letters. Both words have been kept because they are commonly used on social media.

Table 4.14 Two examples of non-Arabic words used in the dataset (the words are highlighted and underlined)

No.	Examples of words from other languages in the tweets
1	ابتداء شعل التطعڡ براڡوا علںك بالربعة والله ںوفقك # الصحه _ تعلق _ مستشفى
2	البلا مو مں المستشڡات البا بالموصڡص بالمستشڡى اهمال عںر طبىعى شکو على الباقى وحاصه سکاکا # الصحه _ تعلق _ مستشفى

4.4.10 Removing Special Letters

The special letters are characters that have the same shape as the Arabic alphabet, but it is from other languages such as Urdu, Punjabi or Farsi. There is only one tweet in the AHS dataset that contains words with special letters as shown in Figure 4.7. In Figure 4.7 there are two highlighted words contain special letters which is "ے" and it is originally an Urdu letter.

Fig. 4.7 An example of a tweet that contains special letters

4.4.11 Correcting Spelling Mistakes

The dataset has 37 words which were corrected from the spelling mistakes and there were different common types of mistakes. In Table 4.15, an example of each common mistake is marked as highlighted and underlined. The types of mistakes are:

- **Missing a letter**

  It commonly happens due to typing the tweet quickly. It might occur without the consideration of the writer. The example of this type is the first tweet in Table 4.15.

- **Writing a letter that is next to the correct letter on the keyboard by mistake**

  It can occur because of fast typing and an example of this is the second tweet in Table 4.15 (the letter ع next to the letter هـ).

- **Writing the word incorrectly as it is pronounced**

  Arabic has three short vowels that have the sound as the three vowel letters ا , و , ى . Many Twitter users write some words with short vowels incorrectly using vowel letters. Also, many Twitter users post a tweet using Arabic dialects and the spelling of a word can be different. The third tweet in Table 4.15 is an example of this type.

- **Spelling weakness of the writer**

  There are many words in Arabic that are hard to spell correctly, such as the letter Hamzah " ء ". It can be in four forms based on the sound of the word and the forms are "أ ، ئ ، ؤ ، ء". The complexity of it is that the sound of it is similar to short vowels and the letter vowels. Thus, many users may write any word containing this letter incorrectly and an example of this is the fourth tweet in Table 4.15.

Table 4.15 Four examples of different spelling mistakes

No.	Examples of different spelling mistakes in Tweets
1	كل الشكر والتقدىر للورىر المثالى المتفانى ڡى العمل الدى ىرى الصب تكلىف ولىس تشرىف # الصحه _ تعلق _ مستشفى
2	طب مستشفمات وراره الصحه فعا ىفس المحالفات مدرطبى اجبى وىعطى ع الاطباءىشتعلو مں عىترحص # الصحه _ تعلق _ مستشفى
3	ما ىلام و الله اجل ڡى هبو ىقول اشربى ببسى و عاربات و بعدس اكتشف ابو معطى حلب بودرة ربد الورں رچمى حرررب
4	كفو ىا توفق الربعه متفاءلىں فىك كثر و ىفحر بك الله ىقوىك،، لىتنا ىقدر ىستسحك لكل الورارات ما شاء الله

4.5 Data Annotating Process

There are a few Arabic datasets for sentiment classification in which each author has used different annotating techniques. For example, Aly and Atiya (2013) proposed the LABR reviews dataset from www.goodreads.com. There are two different labelling techniques for this dataset which are: *rating classification* and textitsentiment polarity classification. The former encompasses the classes from 1-star to 5-stars. The latter characterises the classes as positive, neutral, and negative. In the *rating classification*, the website itself allows the users to write a review about a book and rate it in a scale of five stars. In the *sentiment polarity classification*, the author converted the reviews with four and five stars to a positive class, the reviews with one and two stars to negative class, and any reviews with three stars to neutral class. Also, Abdulla et al. (2013) introduced the Arabic Twitter dataset and employed two Arabic native speakers to classify the tweets to either positive or negative. If the two annotators disagree about a review, a third annotator employed to classify the review. Nabil et al. (2015) collected the ASTD from Twitter and translated them to English. Then, the tweets were annotated manually using Amazon Mechanical Turk through the API Boto.

We collected the AHS dataset from Twitter and the tweets were written using Saudi dialects or Modern Standard Arabic MSA. There are challenges to translate the tweets from the Saudi dialects in order to annotate them by Amazon Mechanical Turk. Moreover, there was no publicly available sentiment analysis dataset for Saudi dialects in order to automatically annotate the dataset using Machine Learning algorithms. Therefore, it is better to classify the dataset manually by human interference because it provides a high-quality annotation. There are three annotators, who are from Saudi Arabia, judged each tweet as expressing either a positive or a negative sentiment. The reasons for choosing three judges is to avoid the dataset bias, people have different views and judgments, and a majority vote can be calculated for each tweet. The three annotators are from Saudi Arabia who are experts in the Saudi Arabian dialects and they are:

1. **Dr. Abdullah M. Al-Homayan (Annotator 1)**

   PhD in Human Resources Management/Health Management, Dean of the Public Health & Health Informatics College at the University of Hail – Saudi Arabia.

2. **Dr. Mohammad T. Alshammari (Annotator 2)**

   PhD in Computer Science, Vice Dean of Academic Affairs at the College of Computer Science and Engineering at the University of Hail – Saudi Arabia.

3. **Mr. Tariq F. Aldhamadi (Annotator 3)**

   Master's in Management/ HR, Lecturer at the College of Business Administration at the University of Hail – Saudi Arabia.

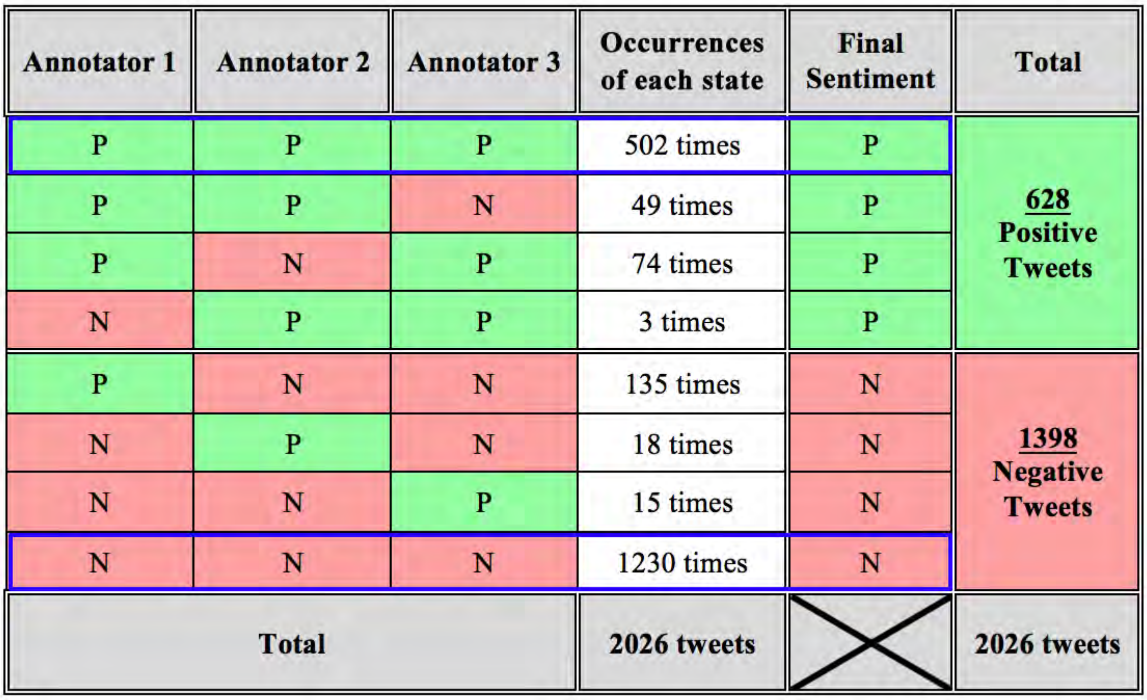
Table 4.16 Four examples of different majority voting (P = Positive & N = Negative) The first two examples show that all the annotators agree either positive or negative. However, in the last two examples one annotator has a different judgment from the other two annotators.

Text	**1**	**2**	**3**	**Sentiment (Mode)**
ما نترك الربعه الوراره الا و نحں نثق ب المستشفات الحكومه ان شاء الله	P	P	P	P
مستشفات كثره لو تعلق افضل	N	N	N	N
اتمى ان الربعه ىبقى ورر الى ان ىشوف وصع كل مستشفات المملكه و ىصع حلول ل تطورها	P	P	N	P
عقبال ما تم اعلاق جميع المستشفات المهمله	P	N	N	N

The dataset was sent to the three annotators and they have judged each tweet based on their impressions. Many sentences were judged the same by all the annotators, either positive or negative, whereas other sentences had two annotators the same and the third one different. All annotated datasets were combined into one and the final sentiment class was calculated based on the majority vote. See an example in Table 4.16.

As there are three annotators and only two classes, (Positive = P or Negative = N), there are only eight states that can occur. Table 4.17, illustrates the number of times that each

Table 4.17 Summary of all the different states, the number of occurrences of each state, the majority voting of each state (Final Sentiment) and the total number of positive and negative tweets. The Sub-(AHS) is labelled using the blue line



Annotator 1	Annotator 2	Annotator 3	Occurrences of each state	Final Sentiment	Total
P	P	P	502 times	P	**628** Positive Tweets
P	P	N	49 times	P	
P	N	P	74 times	P	
N	P	P	3 times	P	
P	N	N	135 times	N	**1398** Negative Tweets
N	P	N	18 times	N	
N	N	P	15 times	N	
N	N	N	1230 times	N	
Total			2026 tweets	☒	2026 tweets

state happened in the dataset and the final sentiment which is the result of majority voting and the total number of positive and negative tweets in the dataset. This is the Main Arabic Health Services (Main-(AHS)) dataset and it contains 628 positive tweets and 1398 negative tweets. Also, we made a (Sub-(AHS)) dataset where all three annotators agree either positive or negative. The Sub-(AHS) contains 502 positve tweets and 1230 negative tweets.

The final step in the annotating is mapping the class of each tweet with their text. This is followed by creating two text files, one for all the positive tweets and another one for all the negative tweets for both datasets.

Figure 4.8 shows the three annotators and the number of positive and negative tweets based on each annotator. The classification results for Annotators 2 and 3 are almost similar, whereas Annotator 1 is slightly different. Overall, the negative sentiments are more prevalent than the positive sentiments.

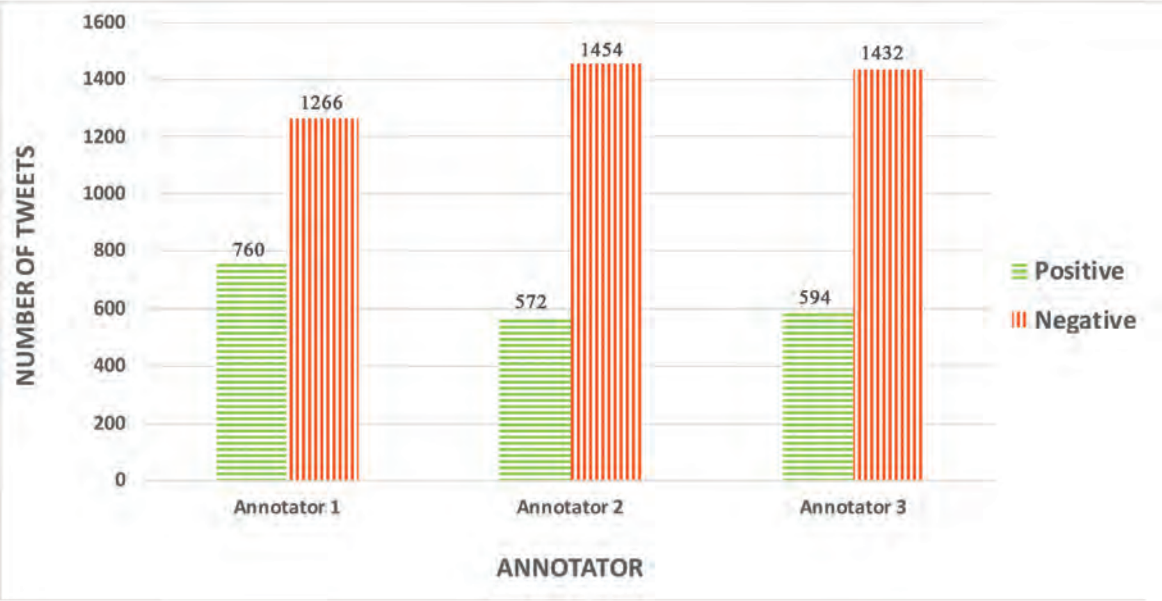


Fig. 4.8 Visualising the number of positive and negative tweets in the dataset based on the three annotators

Figure 4.9 illustrates the number of positive and negative tweets in the dataset within the four different topics and the whole dataset. The topics are: first topic (closing hospital), second topic (solving health), third topic (opinions about health) and fourth topic (improving health). The total number of negative tweets in the corpus is 1398 tweets, whereas the number of positive tweets is 628. The majority of tweets are negative and there is a huge difference between the number of positive and negative tweets in all the topics, except the first one. The first topic has slightly similar numbers of positive and negative tweets, which are 505 and 504 respectively.

4.6 Data Collection and Pre-processing Challenges

The process of collecting tweets from Twitter is effortless, but in the case of collecting tweets on a specific topic it is arduous. There were many challenges when creating an Arabic dataset about health services, and in this section the challenges will be summarised.

1. **Encoding and decoding Arabic text**

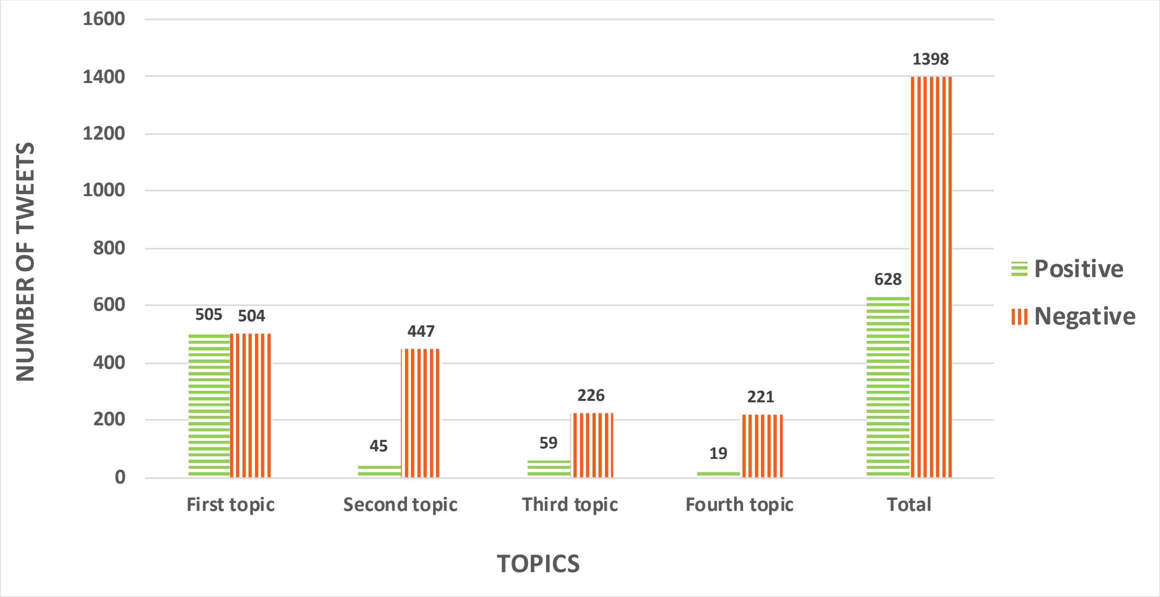


Fig. 4.9 Visualising the number of positive and negative tweets in the dataset based on the four different topics

At the beginning of the data collection stage, Python 3.5 was used to collect data, but it could not support Arabic text without encoding the query text and decoding the retrieving data. Consequently, it was very slow to retrieve data because of this process. In Table 4.18, there are two examples of Arabic tweets without decoding the text. However, after using R, which provides support to retrieve Arabic tweets using Arabic text, this problem was resolved.

Table 4.18 Two examples of Arabic tweets which were not decoded to Arabic characters

No.	Example of Arabic tweets that were not decoded
1	"text:" "âœ‹âœ‹âœ‹Ø¨Ø´Ø±Ù‰■ Ù„Ù…Ù† Ù_Ù…Ù„ÙƒØ§Ø_Ù‡Ù… Ø¨Ù†ÙƒØ§Ù„Ø±Ù_Ø§Ù† Ø§Ù„Ù‚Ø·Ø±Ù_ (Ù†Ø¨Ù_Ø_Ù‡Ø§Ù^Ù†Ø_Ø¶Ø±Ø£ Ø±Ø¨Ø§Ø_Ùƒ) Ø§Ù„Ø±Ù_Ø§Ø¶Ø_Ø§Ù_Ù„ Ø_ÙÅØ±Ø§Ù„Ø¨Ø§Ø·Ù† ØªØ¨Ù^Ùƒ Ø§Ù„Ù‡Ù„Ø§Ù„"
2	"text":"(\u0648\u0625\u0630\u0642\u0627\u0644\u062a\u0623\u0645\u0629 \u0645\u0646\u0647 \u0645\u0644\u0645\u062a\u0639\u0638\u0648\u0646 \u0642\u0648\u0645\u0627\u0627\u0644\u0644\u0647\u0645\u0647\u0644 \u0643\u0647\u0645\u0623\u0648\u0645\u0639\u0630\u0628\u0647\u0645 \u0639\u0630\u0627\u0628\u0627\u0634\u062f\u064a\u062f\u0627\u0642 \u0627\u0645\u0639\u0630\u0631\u0629\u0625\u0644\u0649\u0631\u0628

2. **Specifying words related to health**

   There were many attempts to collect tweets about health services. This was done by specifying some words related to health, such as “hospital مستشفى”, “clinic مستوصف”, “health الصحه”. Many tweets were retrieved, but it was hard to find tweets that contained opinions about health services. This process was repeated many times at different times, but it was not an effective way to collect sentiment tweets on a specific topic.

3. **Encouraging Twitter users to write tweets about health**

   The third topic in the dataset, which is “opinions about health”, was launched for this study. A smaller number of Twitter users were involved in this topic and shared their opinions about health services, even though many Twitter celebrities retweeted some tweets on this topic. However, the other topics were trending topics on Twitter and many Twitter users easily engaged with any trending hashtags.

4. **Arabic dialects**

   The Modern Standard Arabic, or formal Arabic, is called Al-Fusha الفصحى in Arabic. The modern standard Arabic is rarely used on Twitter or social networks. The main source of the formal Arabic language is the Holy Quran. Also, the formal Arabic language is used in official speeches, newspapers, etc., but most Twitter users use their dialect in their tweets.

5. **Combining the tweets increases the size of the sentences**

   After combining two or more tweets, the number of characters increased to more than 140, which was the limit when the tweets were retrieved. In the dataset, there are 21 tweets with over 140 characters because some tweets were combined into one tweet in the filtering phase. The maximum number of characters for a single tweet is 241 and for words, 52. As mentioned previously, some Twitter users wrote more than one tweet about health services. The advantage of that might be, that some tweets do not have enough positive or negative features in a single sentence. On the other hand, the size of the input layer in deep neural network models will be bigger and it will increase the cost of time.

4.7 Summary

This chapter describes in detail all the used techniques in collecting, filtering, and labelling the dataset. It provides all the details for collecting the tweets using **"R"**. The dataset focuses on Arabic tweets which contain opinions about health services only in the period of six months. These were the reasons for collecting the tweets from four Twitter trending hashtags. Then, it details the procedures of filtering the dataset automatically and manually. Additionally, it presents the process of annotating the dataset and it shows different statistical data about the AHS dataset. Finally, it states different challenges experienced when collecting the tweets about health services and pre-processing them. In the next chapter, word embedding models are presented in detail.

Chapter 5

Word Embedding Models for Arabic

5.1 Overview

Word embedding is also known as word representation and word distribution. It is a powerful approach in terms of gathering semantic or syntactic meaning. Bian et al. (2014) defined this term as the NLP technique to compute continuous vectors of distributed word representation. In addition, Chen et al. (2015) introduced the word representation approach as when each word is mapped to a vector in a space; with a huge number of word vectors allowing researchers to gain word similarity based on a large corpus of text. Moreover, the word embedding concept is illustrated in De Boom et al. (2016) as representing words in distributed vectors as real-numbers in fixed-dimensional space.

There are different methods to generate the word embeddings from a large row of text data: neural networks (Mikolov et al., 2013), word co-occurence statistics (Lin and Hovy, 2003), dimensionality reduction (Lebret and Collobert, 2014), and other methods. There are unsupervised learning techniques. The algorithms cluster the words semantically or syntactically based on the availability of abundant and diverse words in a corpora and associate each word with other words based on the context. The large row of text is fed to the word embedding algorithms in order to map a word to a vector. Many researchers have discussed these approaches and have shown the impact of this task in different linguistic areas such as text classification, sentiment analysis, document clustering for information retrieval, paraphrasing, etc. Zahran et al. (2015) trained three word embedding models for Arabic that are: Word2Vec (Continuous Bag of Words (CBOW) and Skip-gram (SG)) and Global Vector Model (GloVe). The vector dimension is 300 for all models and the window sizes are five for CBOW and ten for SG and GloVe. Word analogy questions were used to evaluate the models. Dahou et al. (2016) collected an Arabic corpus from the web and it was used to train Word2Vec (CBOW and SG) models. Three dimensionalities were used for both models that are: 100, 200, and 300. A window size of five was applied to the CBOW model and a window size of 10 was used for the SG model. The trained model outputs were used as input for the CNN to apply the sentiment classification on five Arabic datasets.

There are different techniques to create word embedding models, such as Word2Vec (Mikolov et al., 2013), Global Vector (GloVe) (Pennington et al., 2014a), fastText (Bojanowski et al., 2017), and Poincaré Embeddings (Nickel and Kiela, 2017). In this chapter, three different techniques for Arabic word embedding will be used to build the models using two different Arabic Corpuses. The techniques are Word2Vec, GloVe and fastText, and the Arabic corpora are the Abu El-khair Corpus (Abu El-khair, 2016) and the Arabic Twitter Corpus created for this study and it is available in (Alayba et al., 2019). We will use these pre-trained word embedding models with the sentiment classification in Section 6.4.2.

5.2 Word Embedding Techniques

As has been described in the overview section, there are different approches for word representation. In this section, the three techniques that will be used in our experiments will be described in detail.

5.2.1 Word2Vec Model

Mikolov et al. (2013) proposed the Word2Vec approach to produce word representations using neural networks. It is an efficient approach to group the words that are semantically and syntactically similar from the context. The input of this model is a vocabulary-size vector that represents the words presented in the sentence by placing 1 in their respective index. There is a fixed sized window that slides over all the input text and applies the probabilistic computation to predict similar words. Harris (1954) stated that the similarity in the meaning of the words will increase if they occur in similar contexts. In order to measure the similar meaning between the words, there is a fixed-size of a window to specify the size of the context. Therefore, the prediction is measured based on the surrounding words within the fixed-size of the window. There are two architecture models for Word2Vec: continuous bag of words (CBOW) and skip-gram (SG), and Figure 5.1 illustrates the differences between the two models. The window size in Figure 5.1 is three and the dimensions of each word are the same.

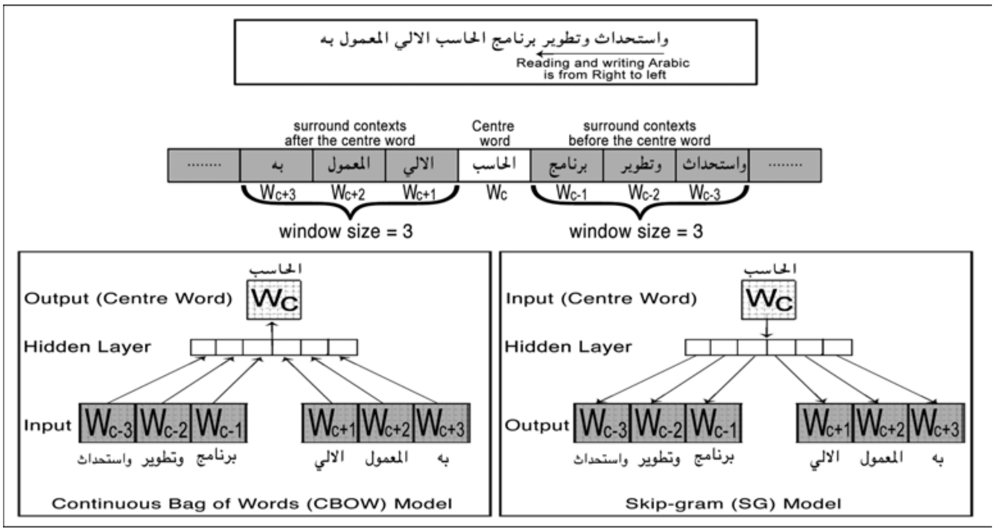


Fig. 5.1 Word2vec model, continuous bag of words (CBOW) and skip-gram (SG)

- **CBOW Model:** Continuous Bag of Words (CBOW) predicts the target (centre) word based on a fixed-size window from the context. For example, for "I am a PhD student at Coventry", the input in this model is ("I", "am", "a", "student", "at", "Coventry") and the outputs or the predicted word is "PhD" in the CBOW model. In Figure 5.2, we expand the information in Figure 5.1, and we represent the architecture of CBOW model in a neural network architecture. The inputs are one-hot vectors of the context words within the window size of three. The total number of inputs in Figure 5.2 is six, which are the three words on the left and the three words on the right of the centre word. Each word in the input layer is represented by a one-hot vector where 1 is placed in the respective index of the word. The number of neurons in the hidden layer is 200 and it represents the dimensions of the output vector. The output consists of neurons that are equal to the vocabulary-size, but only the neuron of the centre word is updating its values.

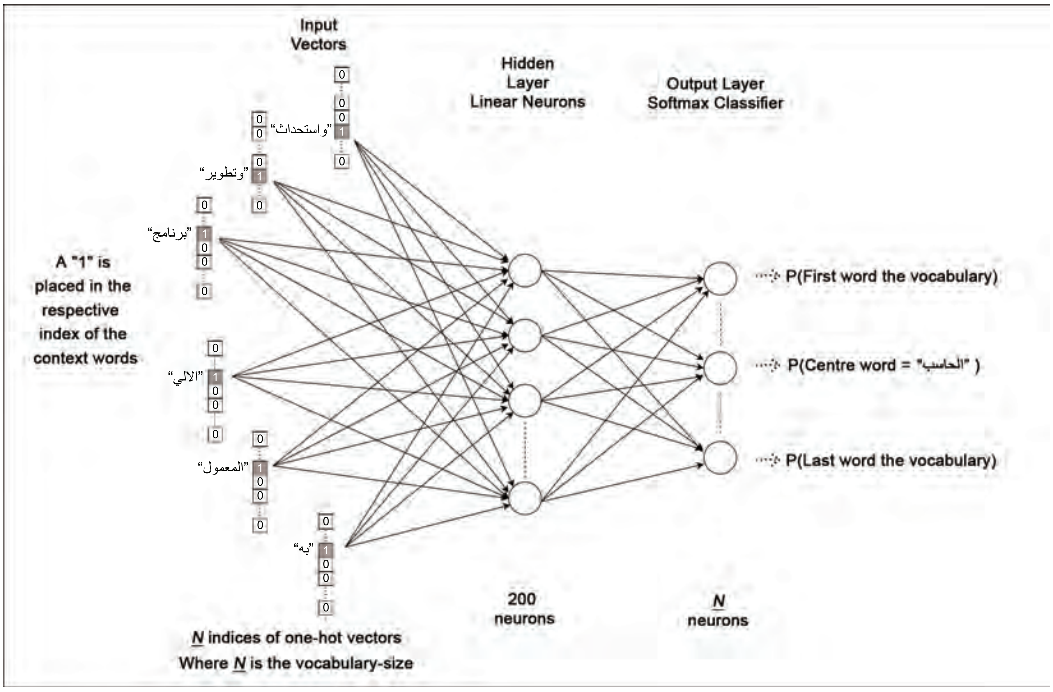

Input
Vectors
Hidden
Layer
Linear Neurons
Output Layer
Softmax Classifier
"واستحداث"
"وتطوير"
"برنامج"
"الالي"
"المعمول"
"به"
A "1" is placed in the respective index of the context words
P(First word the vocabulary)
P(Centre word = "الحاسب")
P(Last word the vocabulary)
200
neurons
N
neurons
N indices of one-hot vectors
Where N is the vocabulary-size

Fig. 5.2 The Word2Vec CBOW neural network architecture

- **SG Model:** Skip-gram (SG) predicts the context or surrounding words, which are the output from the centre or input word within the window size which is also three for the following exaple. This is an opposite model to CBOW. For example, based on the previous sentence example, the input is "PhD" and the predicted words are ("I", "am", "a", "student", "at", "Coventry"). In Figure 5.3, we clarify the information in Figure 5.1 and we explain the architecture of SG model in a neural network architecture. The input is only the centre word in the window size of three. The centre word in the input layer is represented by one-hot vector, where 1 is placed in the respective index of the word. There are 200 neurons in the hidden layer of the network which are the dimensions of the output vector. The output contains neurons that equal to the vocabulary-size. However, only the six neurons of the surrounding words to the centre words are updating their values.

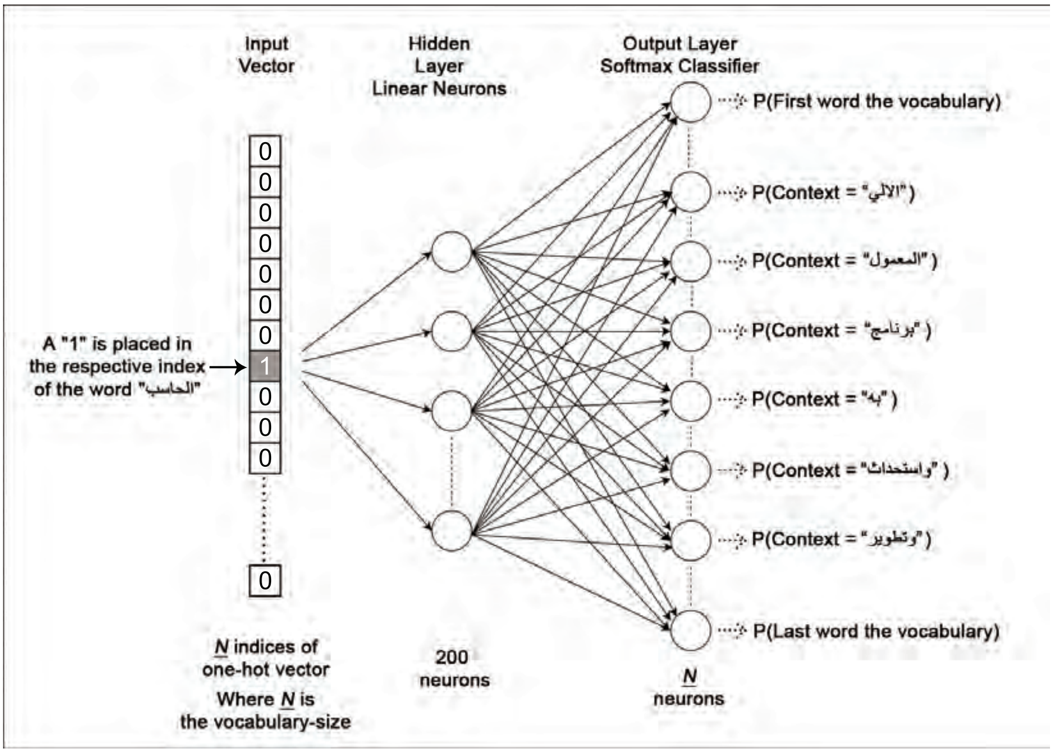


Fig. 5.3 The Word2Vec SG neural network architecture

5.2.2 Global Vector Model (GloVe)

Global Vectors for Word Representation (GloVe) is an extended model for representing a word in a vector. Pennington et al. (2014a) introduced this count-based and prediction-based method at Stanford University. It is an unsupervised approach to capture similar word embedding by observing three main components, which are the corpus, the co-occurrence count, and the word frequency. This technique uses a fixed window size to measure the syntactic and semantic similarities between the words within the size of the window. Also, it considers the global co-occurrence counts of a word from the co-occurrence matrix to identify the general topics.

Pennington et al. (2014a) explained the steps of building the GloVe model from a text corpus. The first step is the construction of a word to word co-occurrence matrix, which contains values that correspond with the number of times the two words occur in a fixed context. Then, it measures the probabilities that the first word occurs in the context of the second one.

5.2.3 FastText Model

Facebook researchers produced an unsupervised learning method to obtain a word embedding model called fastText (Bojanowski et al., 2017). There are two fastText models that are skip-gram and Continuous-Bag-of-Words. We will use Skip-gram in this study because it provides better results, as noted by the Facebook researchers (Facebook AI Research Team, 2016). The fastText skip-gram is an extension to the Word2Vec skip-gram model and it considers sub-words by breaking a word into character n-grams and representing them in vectors. By default, the sub-words take from three to six sequence of characters from the original word. The subwords are attached to the original words in a hash list. The sum of the vector representation of the sub-words is equal to the vector representation of the word itself. It shows strength in morphological word representations. Boundary symbols are added before and after the word, such as <Coventry> to determine the beginning and the end of the word. For instance, with $n = 5$ and the word <Coventry>, the 5-grams for the word are:

`<Cove, Coven, ovent, ventr, entry, ntry>`

Note that the sequence `<entry>`, which refers to the word *entry* is different from the 5-gram `entry` for the word *Coventry*. The sum of vectors for the 5-gram `<Cove, Coven, ovent, ventr, entry, ntry>` will be equal to the sequence `<Coventry>`. Figure 5.4 shows the neural network architecture of fastText skip-gram model. It is similar to Word2Vec skip-gram model and it has hash lists for sub-words to each word that contains over three characters (by default). The updating in the outputs are only for the vectors of the context words and the vectors of the sub-words in hash lists.

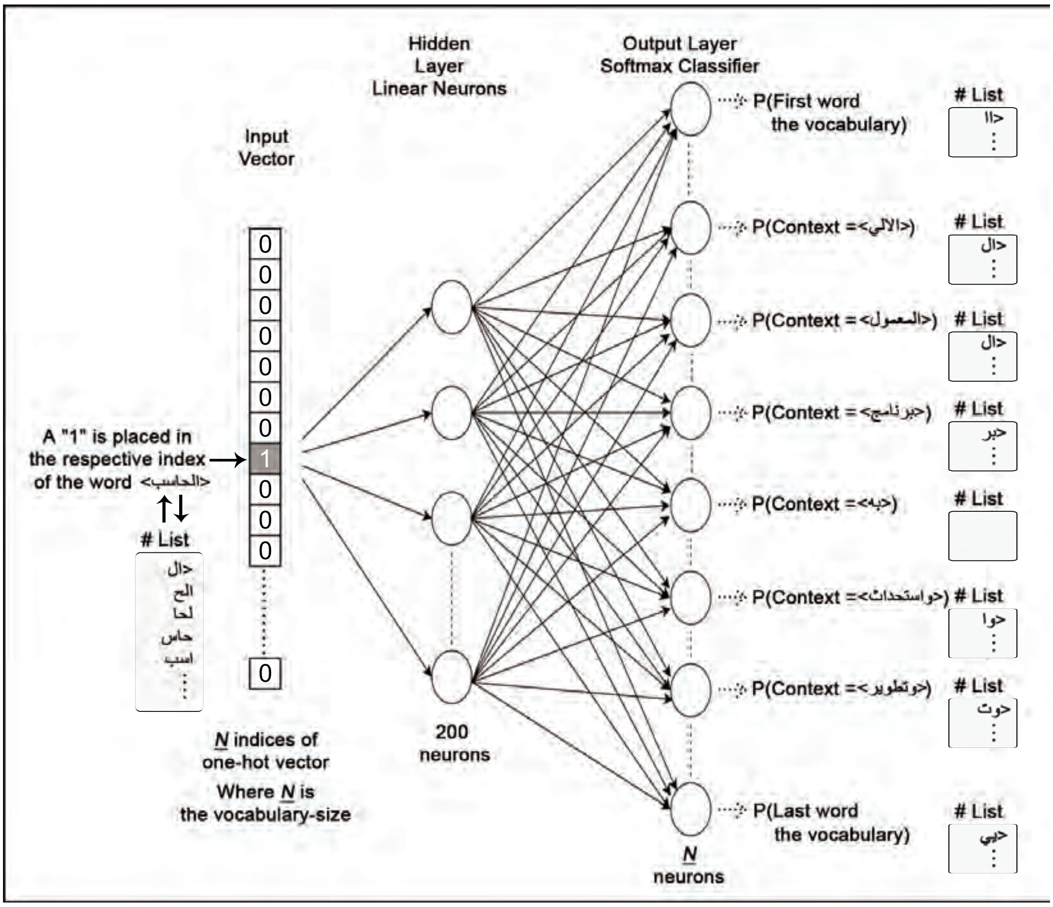


Fig. 5.4 The fastText SG neural network architecture

5.3 Arabic Language Corpora

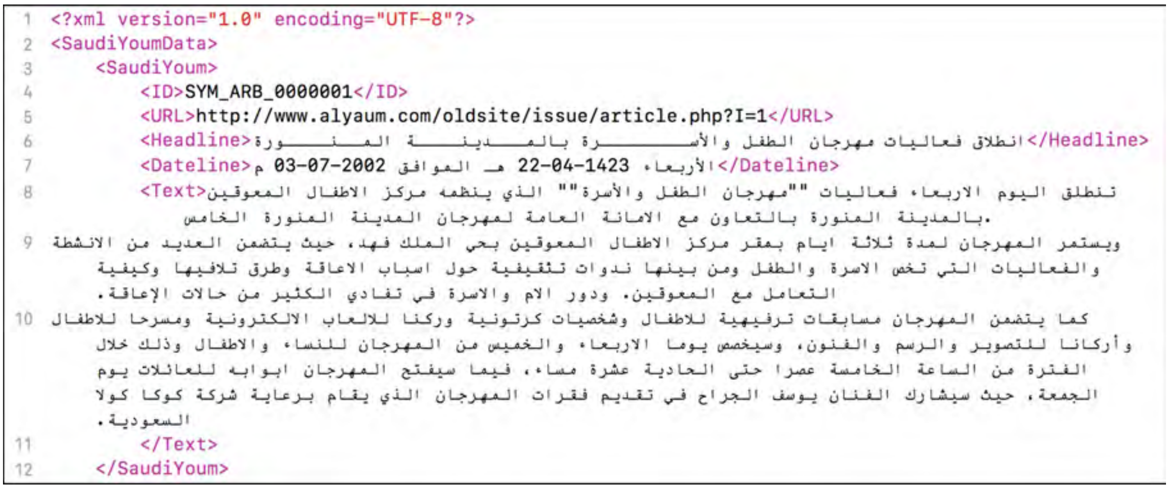
One of the requirements to construct a word representation model is a large number of text data. The Abu El-khair Corpus (Abu El-khair, 2016) will be used to build the word embedding models because it contains 1.5 billion Arabic words. Also, the Twitter Corpus, which was built for this study, will be used as well. The structure of a tweet is different from generic Arabic text and therefore, the Arabic Twitter corpus is needed.

5.3.1 Abu El-khair Corpus

There are many Arabic Corpora freely available, such as the International Corpus of Arabic (Alansary and Nagi, 2014), the Word Count of Modern Standard Arabic (Attia et al., 2011) as well as many other corpora mentioned in Zaghouani (2014). The targeted corpus, in order to build the Arabic word embedding models, should cover a huge number of tokens and Arabic vocabulary entries. The Abu El-khair Corpus (Abu El-khair, 2016) is a free and recent Arabic corpus and it contains over 1.5 billion words and over 3 million vocabulary entries. It has been collected from over five million articles for various subjects from ten different newspapers. The newspapers are from eight Arab countries and it took more than fourteen years to collect. Although, the corpus is from newspaper articles, it covers a large number of Arabic words in different Arabic dialects. The corpus is presented in two different tagging schemes, which are SGML (Standard Generalised Markup Language) and XML (Extensible Markup Language). Also, it has been encoded using two types of Arabic encoding techniques, which are Windows-1256 and UTF-8. In this research, the corpus in XML tagging schemes and UTF-8 encoding has been used and it was stored in ten files based on different newspapers.

5.3.2 Pre-processing the Abu El-khair Corpus

The Abu El-khair Corpus (Abu El-khair, 2016) is structured in XML format using UTF-8 encoding for Arabic. The corpus contains ten different XML files. Figure 5.5 shows an example of one article structured in XML tags. In order to use the corpus for word embedding, there is a need to eliminate all the XML tags and unwanted data. The aim is to keep only the headline and the text (the body of the articles). There were some challenges in parsing the XML files and extracting the text because of either the large size of the files or storing Arabic text with some encoding errors. We tried to parse and filter the XML files using the *Python* package, which called *Beautiful Soup* (Leonard Richardson, 2015), but we could not parse them because of the large XML files and limited memory. Alternatively, we used another *Python* tool called *The ElementTree XML* (Python Software Foundation, 2019) because it can filter large XML files. However, the XML files contain non-UTF-8 characters since they are not valid for this package. As a result, we converted the XML files to TXT files and filtered unwanted data as well as XML tags based on the lines. Then, we pre-processed the text itself from unwanted characters and remaining XML tags. There are further details on the pre-processing the Abu El-khair Corpus and *Python* code in Appendix C.



```
<?xml version="1.0" encoding="UTF-8"?>
<SaudiYoumData>
    <SaudiYoum>
        <ID>SYM_ARB_0000001</ID>
        <URL>http://www.alyaum.com/oldsite/issue/article.php?I=1</URL>
        <Headline>انطلاق فعاليات مهرجان الطفل والأســـــــرة بالمـــــدينـــــة المـــنـــــورة</Headline>
        <Dateline>الأربعاء 22-04-1423 هـ الموافق 03-07-2002 م</Dateline>
        <Text>تنطلق اليوم الاربعاء فعاليات ""مهرجان الطفل والأسرة"" الذي ينظمه مركز الاطفال المعوقين بالمدينة المنورة بالتعاون مع الامانة العامة لمهرجان المدينة المنورة الخامس.
ويستمر المهرجان لمدة ثلاثة ايام بمقر مركز الاطفال المعوقين بحي الملك فهد، حيث يتضمن العديد من الانشطة والفعاليات التي تخص الاسرة والطفل ومن بينها ندوات تثقيفية حول اسباب الاعاقة وطرق تلافيها وكيفية التعامل مع المعوقين. ودور الام والاسرة في تفادي الكثير من حالات الإعاقة.
كما يتضمن المهرجان مسابقات ترفيهية للاطفال وشخصيات كرتونية وركنا للالعاب الالكترونية ومسرحا للاطفال وأركانا للتصوير والرسم والفنون، وسيخصص يوما الاربعاء والخميس من المهرجان للنساء، والاطفال وذلك خلال الفترة من الساعة الخامسة عصرا حتى الحادية عشرة مساء، فيما سيفتح المهرجان ابوابه للعائلات يوم الجمعة، حيث سيشارك الفنان يوسف الجراح في تقديم فقرات المهرجان الذي يقام برعاية شركة كوكا كولا السعودية.
        </Text>
    </SaudiYoum>
```

Fig. 5.5 An example of one article in one of the XML files

5.3.3 Arabic Twitter Corpus

This corpus has been collected for this study and has been used to build the word representation models for sentiment analysis purposes. There are many resources available online to collect Arabic corpuses. Also, it is easy to collect many Arabic texts from the web. However, Twitter is the source for this corpus, as it will be used for sentiment classification on the Arabic Health Services (AHS) dataset which was also collected from Twitter. The reason for using Twitter as a resource is because the way of writing a tweet in Twitter is different from text on other platforms. For instance, the tweet has a limited length of characters and the language used has its own style. All the vocabularies in the AHS dataset is used as keywords to retrieve tweets that contain these words to build this corpus. The vocabulary size is 6900 words. The reason for that is to ensure that all words in the AHS sentiment dataset have vectors represented in a pre-trained word embbeding model.

There are two optional ways to retrieve the tweets from Twitter, which are either using Python or R. Python takes a longer time to retrieve uncommon words that have been written on Twitter, thus R has been used to retrieve the tweets automatically. The R package called "twitteR" was used for tweet collection and saved the retrieved results of each word in a single CSV file. The maximum number of retrieved tweets for each word is 1000 tweets and the minimum number is 5 tweets. Using 5 tweets as a minimum is because any word that occurs less than 5 times for word embedding models will not be considered. There were 46 words that did not have any retrieved results for them, due to the word being written incorrectly, or it not being a common word on Twitter, or the seven day limitation access to the Twitter API.

5.3.4 Pre-processing the Arabic Twitter Corpus

The retrieved tweets were stored in many CSV files, but the Arabic text may be lost if using tools that do not support Arabic. Several methods were used in order to parse the tweets without losing the Arabic text. Below, is a listed series of steps that were followed in order to filter the text and combine them in a single file:

1. Each CSV file contains 17 columns as mentioned in Chapter 3, and the tweets are in a column called *text*. Only the *text* column is parsed.

2. After parsing the text column, any retweeted tweets were removed to have a variety of context next to the keywords.

3. For any file containing less than five tweets, the keyword will be returned to collect more tweets for the returned keyword. There were 55 words that had less than five tweets.

4. Collecting tweets manually for 46 words that did not have any retrieved results and 55 words that had less than the minimum number of tweets.

5. After collecting between 5 and 1000 tweets for each word in the AHS dataset, the text needs to be filtered by removing non Arabic letters and special characters.

5.4 Building and Evaluating Word Representation Models

Three techniques are used in this thesis, *word2vec*, *GloVe*, and *fastText*, to build the word representation models. The two different corpora (Abu El-khair Corpus and Arabic Twitter Corpus) are used to feed the models. Word2vec showed good results in many previous studies, such as (Tang et al., 2014; Levy et al., 2015; Nalisnick et al., 2016). Therefore, a word2vec representation model was built first using the Abu El-khair Corpus. The gensim 3.2.0 tool (Řehůřek and Sojka, 2010) for Python 3.6.1 was used to generate the word2vec models. A fixed window size of five was used in this study as this is a default option. Also, the words that have at least a frequency of five times in the corpus were considered. Five different models were built for each technique (CBOW and SG) and the differences between them were based on the dimensionality length. The dimensionality is 10, 50, 100, 200 and 300, so, as a result, there are ten different Word2vec models.

There is a need to evaluate the models in order to choose the appropriate model for this sentiment analysis study. The most common way to evaluate the vectors in word representation models is by measuring the cosine similarity which is in Equation (5.1). It is a standard measure in Vector Space Modelling to calculate the similarity of two words' vectors (Frome et al., 2013).

$$similarity = cos(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{||\boldsymbol{x}||\cdot||\boldsymbol{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{5.1}$$

It considers the proportional analogies, which is the similarity of the word *France* to the word *Paris* which in turn is like the word *England* to the word *London*. It can be calculated as *London* = *Paris* - *France* + *England*. This evaluation method was used in much research, such as (Mikolov et al., 2013; Drozd et al., 2016; Ghannay et al., 2016), and for Arabic (Zahran et al., 2015). Different types of word relations were used, such as *countries* to *capitals* and *currency*, *adjective* to *adverb*, *masculine* to *feminine*, etc.

On the other hand, the evaluation of the word embedding models in this study uses the `most_similar` function from gensim (Řehůřek and Sojka, 2010) to find the similar 10 entities to the target word. This study concentrates on sentiment analysis and the most popular two words for positive and negative sentiment are the word "*good*", which in Arabic is "جيد" and the word "*bad*", which in Arabic is "سيئ". These two words were used as a target, and then the ten most similar words to them were retrieved. Finally, the model which had the most similar words was selected.

Appendix B, illustrates the results of the ten most similar words to the words *good* and *bad* "جيد و سيئ" based on the ten different word2vec models. From the tables in Appendix B, the best model was evaluated which could get the best results for the sentiment analysis. The two models SG and CBOW with 10 dimensions are not appropriate because similar words do not have the relevant meaning to the words *good* and *bad*. For example, the word *missing* "افتقاده" appears as a related word to *good* "جيد", and the word *confused* "مرتبك" has a similarity to the word *bad*. Also, the two models SG and CBOW with 300 dimensions are not good choices as well, because the CBOW has words of opposite meaning within the top ten words. The SG is suitable for the word *good* "جيد", but the similar words to *bad* "سيئ" are only the word *bad* with different spellings. In the 100 and 200 dimensions of the SG model, the opposite word *bad* "سيء" occurred within the list of similar words to the word *good* "جيد". The CBOW model with 50 dimensions is not an appropriate option because the word *natural* "طبيعي" occurs as a similar word to both the words *good* "جيد" and *bad* "سيء". The word *natural* "طبيعي" can be classified as either positive or negative, so any models having this word in the similar words list are not considered, which are the SG with 50 dimensions and the CBOW with 100 dimensions. As a result of this analysis, the most convenient model to be selected for this study is the *CBOW with 200 dimensions*. A very low dimension indicates not enough trained data to represent similar words, but a very high dimension leads to overtrained models.

Building the *GloVe* models for the Arabic language is implemented using the available code in Pennington et al. (2014b). The Abu El-khair Corpus was used to represent the Arabic words in vectors. The same attributes of the best model in *word2vec* were applied to create the *GloVe* model. These attributes are: not considering words that occur less than five times in the corpus, the window size of five, and the dimension of 200. Also, the same attributes were used to build the *fastText* model, and the code available in Bojanowski et al. (2017) was used to build the model based on the Abu El-khair Corpus. After building the word embedding models based on the Abu El-khair Corpus, three models that have the same attributes as the previous models were built using the Arabic Twitter Corpus. However, the window size was changed to three instead of five because the length of the tweets was small.

5.5 Summary

This chapter reports some definitions of the word embedding and describes some techniques of them which are Word2vec, GloVe, and fastText. Then, it describes the Abu El-khair corpus, the process of parsing, and extracting the text from it. Also, it proposes the Arabic Twitter corpus and explains the pre-processing of the corpus. It details the techniques of training different word embedding models using the two corpora. Finally, it evaluates different Word2vec models to be used for the sentiment classification. In the next chapter, sentiment classifications are presented in detail.

Chapter 6

Feature Sets and Classifiers for Sentiment Analysis

6.1 Overview

There is a lack of tools that preprocess and deal with Arabic text. In Subsection 2.3.1, it was mentioned, that there is complexity in Arabic, because of the multiple forms for a single word. Al-Nashashibi (2014) addressed different methods for extracting Arabic roots. We tried Tashsphyne Arabic Light Stemmer (Zerrouki, 2012) as a stemmer and roots generator for Arabic words in the datasets. However, as we see in Table 6.1, it was not an accurate tool to do the stemming and root generating. This tool might be excellent for text that is written in Modern Standard Arabic but it is not appropriate for our dataset. Table 6.1 shows an example of a tweet and clarifies the stem and root forms of each word in the tweet. In Table 6.1, the word المستشفات has the stem and root form شف while the stem form should be مستشفى and the root استشفى . Moreover, the word ممتارة has the stem form متار and the root form مر , while the stem form should be ممتار and the root form should be امتار .

Table 6.1 Examples of words in a tweet and their stem and root forms using Tashsphyne Arabic Light Stemmer

Original word	Stemmed word	Root of the word
الحدمات	حدما	حدم
الصحه	صح	صح
التى	-	-
تقدمها	قدم	قدم
المستشفات	شف	شف
ممتارة	متار	مر

This chapter explains an alternative way to prevent the multiple forms of a word in Arabic using different sentiment analysis levels. In addtion, all the sentiment analysis experiments using various features and machine learning methods are described in this chapter. Finally, the sentiment analysis experiments results will be explored using the proposed classifiers and features.

6.2 Sentiment Analysis Levels

This study applies sentiment classification on short messages (tweets) in Arabic. As the tweets contain a limited number of words, there is a need to expand the number of features, by considering each single character in a tweet. Moreover, some words are split into sub-words in order to generate more features, and some words that have many forms are normalised. This is an example of a tweet in the AHS dataset: "الخدمات الصحه التى تقدمها المستشفيات ممتاره" and the translation in English is: "*The health services that are provided by hospitals are excellent*". This section describes the different sentiment analysis levels that were used in this study, with examples for this tweet.

6.2.1 Character Level (Char-Level)

This level splits a tweet in both datasets into character rather than words and each charchter in the short message becomes a single feature. For example, the previously mentioned tweet becomes in **Char-Level** [*’T’, ’h’, ’e’, ’ ’, ’h’, ’e’, ’a’, ’l’, ’t’, ’h’, ’ ’, ’s’, ’e’, ’r’, ’v’, ’i’, ’c’, ’e’, ’s’, ’ ’, ’t’, ’h’, ’a’, ’t’, ’ ’, ’a’, ’r’, ’e’, ’ ’, ’p’, ’r’, ’o’, ’v’, ’i’, ’d’, ’e’, ’d’, ’ ’, ’b’, ’y’, ’ ’, ’h’, ’o’, ’s’, ’p’, ’i’, ’t’, ’a’, ’l’, ’s’, ’ ’, ’a’, ’r’, ’e’, ’ ’, ’e’, ’x’, ’c’, ’e’, ’l’, ’l’, ’e’, ’n’, ’t’*]. For the Arabic text, this is [’ا’, ’ل’, ’خ’, ’د’, ’م’, ’ا’, ’ت’, ’’, ’ا’, ’ل’, ’ص’, ’ح’, ’ى’, ’ه’, ’’, ’ا’, ’ل’, ’ت’, ’ى’, ’’, ’ت’, ’ق’, ’د’, ’م’, ’ه’, ’ا’, ’’, ’ا’, ’ل’, ’م’, ’س’, ’ت’, ’ش’, ’ف’, ’ى’, ’ا’, ’ت’, ’’, ’م’, ’م’, ’ت’, ’ا’, ’ر’, ’ه’]. In the Arabic example, there are six words and 39 characters.

6.2.2 Character 3-Gram Level (Ch3gram-Level)

Arabic is a complicated language in terms of the variety of words that are generated from the root word. There are many forms as has been explained in Chapter 3. Most of the words in Arabic are derived from the root, which consists of three consonant letters and forms a *verb*. When some affixes are added to the root, it transforms the word into different forms.

There are fifteen possible forms known as "الأوزان Alawzaan" (Alsaad and Abbod, 2014). Each form has its own basic meaning and the meaning is linked to the root. For instance, the word كتب *ktb* means *write*. When the letter م *meem* is added as a prefix to the word, and the letter ة *teh marbuta* is added as a suffix to the word, it becomes مكتبة *mktbp* which means *library*. Also, if the two letters س *seen* and ي *yeh* are added as prefixes to the root and the two letters و *waw* and ن *noon* are added as a suffix to the root, the word becomes سيكتبون *syktbwn*, which means "*they will write*".

Due to the lack of an accurate tool to do the pre-processing for the Arabic text, the **Ch3gram-Level** will be used, which will divide any word that has more than three letters into 3-grams. For example, the aforementioned tweet in the English translation becomes in **Ch3gram-Level** [*"The", "hea", "eal", "alt", "lth", "ser", "erv", "rvi", "vic", "ice", "ces", "tha", "hat", "are", "pro", "rov", "ovi", "vid", "ide", "ded", "by", "hos", "osp", "spi", "pit", "ita", "tal", "als", "are", "exc", "xce", "cel", "ell", "lle", "len", "ent"*]. Also, the **Ch3gram-Level** in Arabic is ['الخ', 'لحد', 'حدم', 'دما', 'مات', '', 'الص', 'لصح', 'صحى', 'حيه', '', 'الت', 'لتى', '', 'تقد', 'قدم', 'دمه', 'مها', '', 'الم', 'لمس', 'مست', 'ستش', 'تشف', 'شفى', 'فيا', 'يات', '', 'ممت', 'متا', 'تار', 'اره']. The number of features in this Arabic example is 27.

6.2.3 Character 5-Gram Level (Ch5gram-level)

The number of characters is extended and the 5-gram is used on any words that have more than five lettters. The reason for choosing the number five is because the average length of all the words in the AHS dataset equals five. For example, the **Ch5gram-Level** for the English translation is [*'The', 'healt', 'ealth', 'servi', 'ervic', 'rvice', 'vices', 'that', 'are', 'provi', 'rovid', 'ovide', 'vided', 'by', 'hospi', 'ospit', 'spita', 'pital', 'itals', 'are', 'excel', 'xcell', 'celle', 'ellen', 'llent'*], and the **Ch5gram-Level** in Arabic is ['الحدم', 'لحدما', 'حدمات', '', 'الصحى', 'لصحيه', '', 'التى', '', 'تقدمه', 'قدمها', '', 'المست', 'لمستش', 'مستشف', 'ستشفى',

’تشما’, ’شمات’, ’’, ’ممتار’, ’متاره’]. The number of distinctive features for this Arabic example is 16.

6.2.4 Stanford Tokenization Level (StanfordToken-Level)

StanfordToken-Level is implemented using Stanford CoreNLP natural language software (Manning et al., 2014). It has several functions and one of them is tokenizing the text. This tokenizer is beneficial when using it with Arabic text, because of its ability to split some Arabic linked object pronouns that are linked to nouns, verbs, adjectives, etc. In order to tokenize the Arabic text using Stanford Tokenizer, we need to download the Arabic *CoreNLP Server* and run it on the computer. Then, we used the *pycorenlp* Python package to connect to the server using the class *StanfordCoreNLP*. This class has a function called `annotate`, which takes two inputs; the text and properties. In the properties, we can specify the required *Stanford CoreNLP* functions, such as POS, Lemmas, Tokenization, etc. The challenge in this method is extracting the tokens from the output which is a Python dictionary.

After tokenizing the tweets, many tokens contain only one character and they will be removed. For example, the word تقدمها is a feminine present verb with an object pronoun at the end of the word. The root and the past tense of the word is قدم. When adding the letter ت to the beginning of the root word, it will become تقدم, which is the feminine present verb form. The object pronoun ها means *her* linked to the end of the feminine present verb, to make the word تقدمها. When this word is tokenized based on the **StanfordToken-Level**, there will be two tokens which are the feminine present verb form تقدم and the object pronoun ها. The **StanfordToken-Level** for the example tweet in Arabic is [’الحدمات’, ’الصحى’, ’ه’, ’التى’, ’تقدم’, ’ها’, ’المستشمات’, ’ممتاره’]. However, when we use this tool to tokenize English text, the text will be tokenized based on the words only.

6.2.5 Word Level (Word-Level)

Word-Level is the most commonly used level in sentiment analysis and text classification, even for short messages. Each word in a tweet is a single feature, and the tweet is split by spaces " ". For instance, [*'The', 'health', 'services', 'that', 'are', 'provided', 'by', 'hospitals', 'are', 'excellent'*]. Also, the Arabic example is ['الحدمات', 'الصحه', 'التى', 'تقدمها', 'المستشڡات', 'ممتاره']. The number of features in the Arabic example is 6.

6.2.6 Word Bigram Level (Word-Bigram-Level)

Word-Bigram-Level focuses on combining two words in a tweet as one feature using the *2-gram* or *bigram* method. The number of features is reduced in this level compared to the **Word-Level**. Applying **Word-Bigram-Level** on the example tweet gives [*'The health', 'health services', 'services that', 'that are', 'are provided', 'provided by', 'by hospitals', 'hospitals are', 'are excellent'*]. Also, the Arabic example is ['الحدمات الصحه', 'الصحه التى', 'التى تقدمها', 'تقدمها المستشڡات', 'المستشڡات ممتاره']. The number of features in the Arabic example is 5.

6.3 Using Different Machine Learning Algorithms

The following subsections describe all the used features and different ML models for sentiment classification. Also, the evaluations of these models will be presented as well.

6.3.1 Different Machine Learning Algorithms

Various machine learning classifiers are used to do the sentiment analysis. *Scikit-learn*, also known as *sklearn*, is used in this thesis (Pedregosa et al., 2011). It is a Python tool for data analysis. This package contains a range of machine learning methods to implement supervised and unsupervised learning, and other functions such as preprocessing, dimensionality reduction, model selection, etc. It is built on *NumPy* (Van der Walt et al., 2011), *SciPy* (Jones et al., 2001), and *matplotlib* (Hunter, 2007) to simplify the data distribution and the data visualisation. In this section, we will use three different feature selection techniques for the sentiment classification, which are TF, TF-IDF, and Part of Speech (POS) tagging. Each text token will be presented by a numeric value using TF and TF-IDF techniques, explained in Section 3.4.5. In the POS technique, the required POS tagging classes of the words will be represented by numeric values. There will be more details about feature selection in Subsection 6.3.1. Also, we will use seven machine learning methods from the *sklearn* package that are:

1. **Multinomial Naive Bayes (MNB)**
2. **Bernoulli Naive Bayes (BNB)**
3. **Nu-Support Vector Classification (NSVC)**
4. **Linear Support Vector Classification (LSVC)**
5. **Logistic Regression (LR)**
6. **Stochastic Gradient Descent (SGD)**
7. **Ridge Classifier (RDG)**

We will apply these algorithms on the proposed sentiment classification problem to predict the class (positive and negative). We will use different features and different sentiment analysis levels to improve the classification. The used features with the previous Machine Learning algorithms are:

- **Term Frequency (TF):** *sklearn* has a class called `TfidfVectorizer` which transforms documents to a matrix that contains TF-IDF weighted features using the `fit_transform` method. It has many parameters, such as `analyzer` which is used for specifying the kinds of features and whether they are words or characters. Also, `ngram_range` builds the n-gram using minimum and maximum values. Another used class is `TfidfTransformer`, to convert a count matrix to weighted TF or TF-IDF using the `transform` method, and one of its parameters is `use_idf`. When the value of the parameter is *False*, the terms will be weighted using TF only.

- **Term Frequency Inverse Document Frequency (TF-IDF):** has the same steps as TF, but the only difference is using the default value in the `use_idf` parameter, which is True. This parameter is to calculate the inverse-document-frequency of the terms.

- **Part of Speech Tagging (POS):** is applied using the *Stanford CoreNLP – natural language software* (Manning et al., 2014) for the Arabic language (Green and Manning, 2010) and the `CountVectorizer` class which is one of the classes in *scikit-learn*. *Stanford CoreNLP* has tools for human language technology and it has packaged models for several languages, Arabic being one of them. It has different functionalities to generate *Part of Speech*, *Lemmas*, *Constituency Parse*, *Words Dependency*, *Sentiment*, etc.

  There is a Python package, that is called *py-corenlp* for Stanford CoreNLP. It provides an API to connect to the Stanford CoreNLP server and do some text processing using the available functionalities in the *Stanford CoreNLP* (Milli, 2016). The *Part of Speech Tagging* annotation is used to classify each word into its type, *Verbs Adjectives, Comparatives*, and *Superlatives* are considered. A list was built that contains *Verbs* and *Adjectives* in order to utilise them as features for the sentiment classification. Figure 6.1 shows the *Part of Speech tagging* for the example tweet, which is "الحدمات الصحه التى تقدمها المستشفمات ممتاره".

  In order to use various machine learning classifiers, the words need to be converted to numeric features; and `CountVectorizer` converts a text to a matrix of token counts

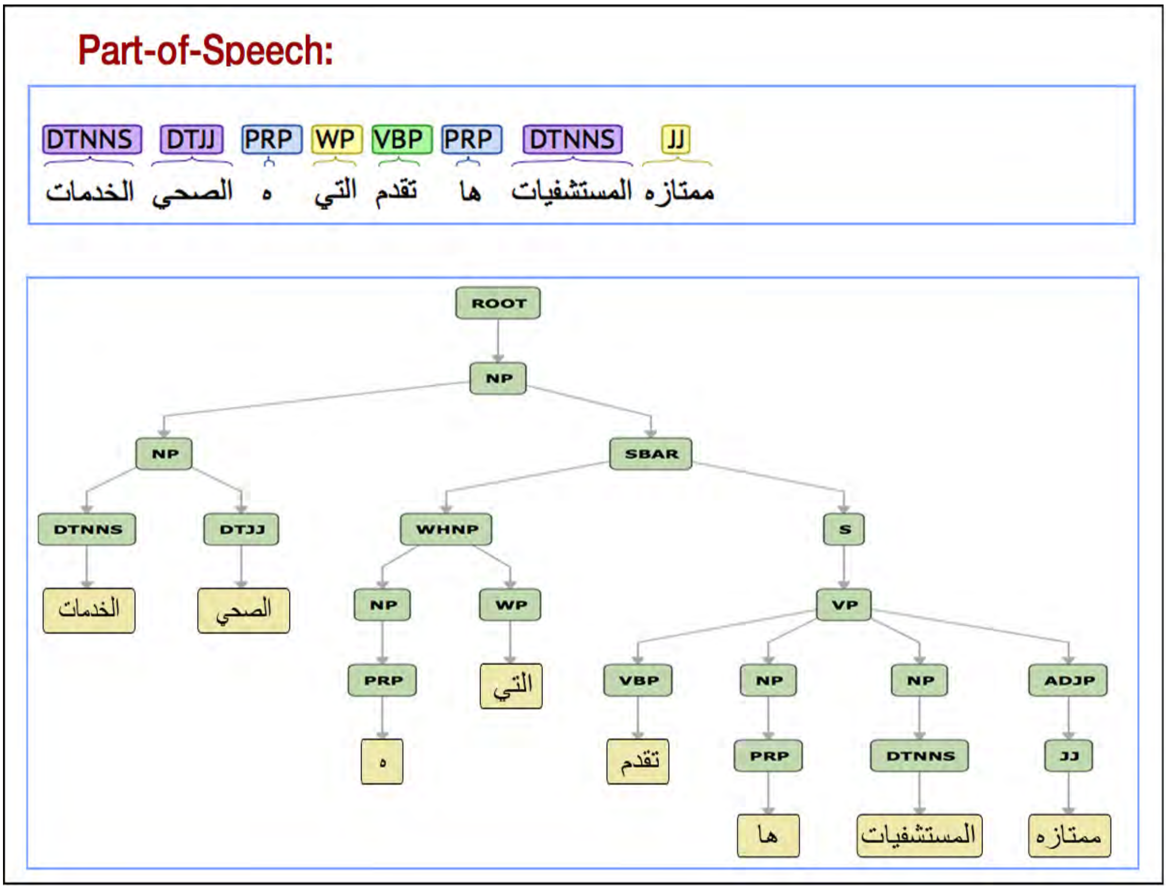


Fig. 6.1 An example of the *Part of Speech Tagging* on an Arabic tweet

using both the tokenization and its frequency counting in this class. It has the same parameters as `TfidfVectorizer`, and one of them is `vocabulary` used either to map a Python dictionary or iterate over terms. The created *Part of Speech Tagging* list is iterated using the `vocabulary` parameter.

6.3.2 Evaluation and Discussion on Using Different Machine Learning Algorithms

We will use the accuracy to evaluate the performance of the classifiers. We use 10-fold Cross Validation to evaluate each model with the three different feature sets. Also, the variations of 10-fold Cross Validation are calculated. There are six different tables from Table 6.2 to

Table 6.7 that contain the accuracy and the variation results for different sentiment analysis levels. In each table the highest accuracy for each dataset has been emboldened. Table 6.2 shows the accuracy of the seven used ML models using the features based on the Main-AHS and Sub-AHS datasets. Using the part of speech tagging, as a feature selection, is not applicable in the Char-Level. The accuracy for tha Main-AHS are between 0.7700 and 0.8199 and the accuracy for tha Sub-AHS are between 0.7985 and 0.8459.

Table 6.2 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Char-Level

	Main-AHS			Sub-AHS		
	TF	**TF-IDF**	**POS**	**TF**	**TF-IDF**	**POS**
MNB	0.7904 (+/- 0.11)	0.7983 (+/- 0.13)	N/A	0.8026 (+/- 0.27)	0.8076 (+/- 0.25)	N/A
BNB	0.7700 (+/- 0.20)	0.7745 (+/- 0.24)	N/A	0.7985 (+/- 0.24)	0.7997 (+/- 0.20)	N/A
NSVC	**0.8199** (+/- 0.18)	0.8130 (+/- 0.20)	N/A	0.8429 (+/- 0.25)	0.8424 (+/- 0.16)	N/A
LSVC	0.8198 (+/- 0.20)	0.8164 (+/- 0.19)	N/A	0.8452 (+/- 0.24)	0.8430 (+/- 0.19)	N/A
LR	0.8189 (+/- 0.16))	0.8164 (+/- 0.17)	N/A	0.8441 (+/- 0.21)	**0.8459** (+/- 0.18)	N/A
SGDC	0.8070 (+/- 0.20)	0.8031 (+/- 0.27)	N/A	0.8262 (+/- 0.32)	0.8251 (+/- 0.33)	N/A
RDG	0.8139 (+/- 0.21)	0.8149 (+/- 0.21)	N/A	0.8435 (+/- 0.23)	0.8447 (+/- 0.19)	N/A

Table 6.3 illustrates the accuracy performance of the different machine learning algorithms with the TF, TF-IDF and POS as features. The results in the Ch3gram-Level improve in comparison to the results in the Char-Level. They are between 0.8618 and 0.9072 for the Main-AHS, and for the Sub-AHS the results are in a range between 0.8631 and 0.9307. The POS feature is not suitable for this level, due to the majority of the verbs and adjectives in Arabic having a length over three characters.

Table 6.3 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Ch3gram-Level

	Main-AHS			Sub-AHS		
	TF	**TF-IDF**	**POS**	**TF**	**TF-IDF**	**POS**
MNB	0.8914 (+/- 0.21)	0.8855 (+/- 0.20)	N/A	0.9215 (+/- 0.13)	0.9221 (+/- 0.11)	N/A
BNB	0.8974 (+/- 0.19)	0.8958 (+/- 0.23)	N/A	0.9284 (+/- 0.13)	0.9284 (+/- 0.19)	N/A
NSVC	0.8618 (+/- 0.25)	0.8894 (+/- 0.30)	N/A	0.8666 (+/- 0.13)	0.8632 (+/- 0.14)	N/A
LSVC	0.9018 (+/- 0.18)	**0.9072** (+/- 0.20)	N/A	0.9296 (+/- 0.15)	0.9261 (+/- 0.13)	N/A
LR	0.8885 (+/- 0.20)	0.8722 (+/- 0.25)	N/A	0.9128 (+/- 0.12)	0.8961 (+/- 0.20)	N/A
SGDC	0.8929 (+/- 0.24)	0.8973 (+/- 0.23)	N/A	0.9267 (+/- 0.13)	0.9232 (+/- 0.15)	N/A
RDG	0.8998 (+/- 0.19)	0.8998 (+/- 0.21)	N/A	**0.9307** (+/- 0.10)	0.9284 (+/- 0.14)	N/A

Table 6.4 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Ch5gram-Level

	Main-AHS			Sub-AHS		
	TF	**TF-IDF**	**POS**	**TF**	**TF-IDF**	**POS**
MNB	0.8914 (+/- 0.19)	0.8924 (+/- 0.23)	0.8879 (+/- 0.19)	0.9209 (+/- 0.20)	0.9220 (+/- 0.18)	0.9191 (+/- 0.20)
BNB	0.8815 (+/- 0.18)	0.8835 (+/- 0.25)	0.8933 (+/- 0.20)	0.9140 (+/- 0.15)	0.9128 (+/- 0.18)	0.9278 (+/- 0.15)
NSVC	0.8569 (+/- 0.28)	0.8919 (+/- 0.24)	0.8825 (+/- 0.27)	0.8707 (+/- 0.28)	0.8944 (+/- 0.35)	0.8949 (+/- 0.23)
LSVC	0.9008 (+/- 0.27)	0.9017 (+/- 0.24)	**0.9032** (+/- 0.17)	0.9307 (+/- 0.18)	0.9272 (+/- 0.10)	0.9324 (+/- 0.11)
LR	0.8761 (+/- 0.26)	0.8504 (+/- 0.20)	0.8845 (+/- 0.20)	0.9007 (+/- 0.21)	0.8649 (+/- 0.11)	0.9070 (+/- 0.15)
SGDC	0.8954 (+/- 0.26)	0.8918 (+/- 0.30)	0.8954 (+/- 0.18)	0.9302 (+/- 0.19)	0.9272 (+/- 0.09)	0.9307 (+/- 0.07)
RDG	0.8993 (+/- 0.24)	0.8948 (+/- 0.25)	0.9022 (+/- 0.18)	0.9273 (+/- 0.18)	0.9180 (+/- 0.13)	**0.9347** (+/- 0.17)

Table 6.4, Table 6.5 and Table 6.6 present the accuracy results for the ML models using the three different sentiment levels that are the Ch5gram-Level, the StanfordToken-Level and the Word-Level. The three experiments' results are similar and they are between the range of 0.8455 to 0.9132 for the Main-AHS and for the Sub-AHS, they are in the range of 0.8568 to 0.9406. Table 6.5 provides the results using the StanfordToken-Level and it has the best accuracies, 0.9132 for the Main-AHS using LSVC with the POS feature and 0.9406 for the Sub-AHS using SGDC with the POS feature as well. Overall, the POS feature selection has the best accuracy results. In the Ch5gram-Level, the POS feature is the best for all classifiers except MNB and NSVC as well as in the StanfordToken-Level, the POS has the best accuracy performance for all classifiers except the NSVC.

Table 6.5 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the StanfordToken-Level

	Main-AHS			Sub-AHS		
	TF	**TF-IDF**	**POS**	**TF**	**TF-IDF**	**POS**
MNB	0.8914 (+/- 0.16)	0.8904 (+/- 0.16)	0.8890 (+/- 0.23)	0.9180 (+/- 0.20)	0.9186 (+/- 0.13)	0.9192 (+/- 0.13)
BNB	0.8968 (+/- 0.14)	0.9002 (+/- 0.24)	0.9018 (+/- 0.24)	0.9296 (+/- 0.24)	0.9301 (+/- 0.13)	0.9302 (+/- 0.23)
NSVC	0.8707 (+/- 0.14)	0.8988 (+/- 0.22)	0.8860 (+/- 0.20)	0.8903 (+/- 0.24)	0.9030 (+/- 0.27)	0.8984 (+/- 0.1930)
LSVC	0.9121 (+/- 0.11)	0.9091 (+/- 0.16)	**0.9132** (+/- 0.21)	0.9405 (+/- 0.23)	0.9319 (+/- 0.18)	0.9400 (+/- 0.24)
LR	0.8855 (+/- 0.21)	0.8568 (+/- 0.18)	0.8904 (+/- 0.22)	0.9053 (+/- 0.21)	0.8735 (+/- 0.18)	0.9174 (+/- 0.14)
SGDC	0.9102 (+/- 0.14)	0.9047 (+/- 0.18)	0.9063 (+/- 0.26)	0.9347 (+/- 0.22)	0.9359 (+/- 0.14)	**0.9406** (+/- 0.23)
RDG	0.9043 (+/- 0.13)	0.8953 (+/- 0.16)	0.9097 (+/- 0.24)	0.9353 (+/- 0.20)	0.9221 (+/- 0.12)	0.9405 (+/- 0.24)

Table 6.6 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Word-Level

	Main-AHS			Sub-AHS		
	TF	**TF-IDF**	**POS**	**TF**	**TF-IDF**	**POS**
MNB	0.8919 (+/- 0.22)	0.8865 (+/- 0.21)	0.8889 (+/- 0.10)	0.9255 (+/- 0.19)	0.9146 (+/- 0.17)	0.9220 (+/- 0.19)
BNB	0.8914 (+/- 0.20)	0.8870 (+/- 0.18)	0.8884 (+/- 0.17)	0.9186 (+/- 0.24)	0.9169 (+/- 0.12)	0.9226 (+/- 0.12)
NSVC	0.8736 (+/- 0.24)	0.8963 (+/- 0.19)	0.8667 (+/- 0.08)	0.8886 (+/- 0.20)	0.9019 (+/- 0.39)	0.8995 (+/- 0.21)
LSVC	0.9018 (+/- 0.18)	0.9013 (+/- 0.16)	**0.9037** (+/- 0.14)	0.9388 (+/- 0.1547)	0.9324 (+/- 0.07)	**0.9394** (+/- 0.17)
LR	0.8722 (+/- 0.25)	0.8455 (+/- 0.15)	0.8850 (+/- 0.12)	0.8938 (+/- 0.21)	0.8568 (+/- 0.1467)	0.9117 (+/- 0.1597)
SGDC	0.8973 (+/- 0.17)	0.8983 (+/- 0.18)	0.8934 (+/- 0.20)	0.9313 (+/- 0.13)	0.9348 (+/- 0.13	0.9330 (+/- 0.17)
RDG	0.9018 (+/- 0.20)	0.8905 (+/- 0.19)	0.9028 (+/- 0.11)	0.9302 (+/- 0.12)	0.9296 (+/- 0.16)	**0.9394** (+/- 0.19)

Table 6.7 The accuracy and the standard deviation of all the ML classifiers with the three text feature selection (TF, TF-IDF, POS), on the Main-AHS and the Sub-AHS datasets, based on the Word-Bigram-Level

	Main-AHS			Sub-AHS		
	TF	**TF-IDF**	**POS**	**TF**	**TF-IDF**	**POS**
MNB	0.8352 (+/- 0.10)	0.8006 (+/- 0.14)	N/A	0.8522 (+/- 0.12)	0.8101 (+/- 0.18)	N/A
BNB	0.7414 (+/- 0.13)	0.7408 (+/- 0.12)	N/A	0.7558 (+/- 0.09)	0.7535 (+/- 0.13)	N/A
NSVC	0.8201 (+/- 0.58)	0.7929 (+/- 0.22)	N/A	0.8396 (+/- 0.53)	0.8151 (+/- 0.48)	N/A
LSVC	0.8376 (+/- 0.11)	0.8169 (+/- 0.24)	N/A	0.8528 (+/- 0.15)	0.8308 (+/- 0.17)	N/A
LR	0.7730 (+/- 0.18)	0.7182 (+/- 0.09)	N/A	0.7812 (+/- 0.14)	0.7304 (+/- 0.10)	N/A
SGDC	**0.8435** (+/- 0.11)	0.8223 (+/- 0.20)	N/A	**0.8568** (+/- 0.18)	0.8407 (+/- 0.18)	N/A
RDG	0.8223 (+/- 0.14)	0.7942 (+/- 0.24)	N/A	0.8349 (+/- 0.14)	0.8014 (+/- 0.18)	N/A

Finally, Table 6.7 shows the accuracy results for different ML methods using the Word-Bigram-Level and they are the worst results in comparison to the other levels. The results are between 0.7182 and 0.8435 for the Main-AHS and they are from 0.7304 to 0.8568 for the Sub-AHS.

Both Figure 6.2 and Figure 6.3, show the accuracy in histogram for both datasets: Main-AHS and Sub-AHS. They summarise the results for all of the different sentiment levels using different machine learning algorithms with different features. We do not use the POS tagging feature selection in three levels: Char-Level, Ch3gram-Level, and Word-Bigram-Level. All of these levels are not appropriate to extract the POS tagging because the features in these levels are either: characters, three characters (not a meaningful word), or a combination of two words.

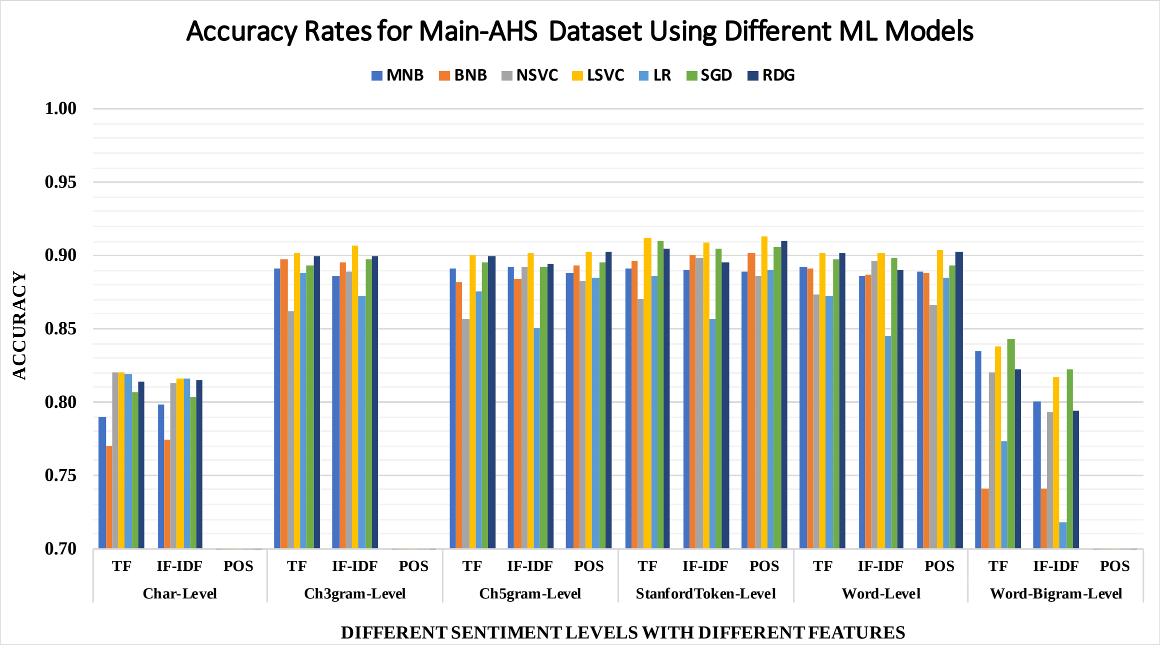


Fig. 6.2 The accuracy for the Main-AHS dataset using different machine learning models, with the three used features based on different sentiment analysis levels

It is clear from both figures that the Char-Level and the Word-Bigram-Level have lower accuracies compared to the other levels. In the Char-Level, the BNB classifier has the lowest accuracies for both datasets indicating 0.77 for the Main-AHS and 0.79 for the Sub-AHS. On

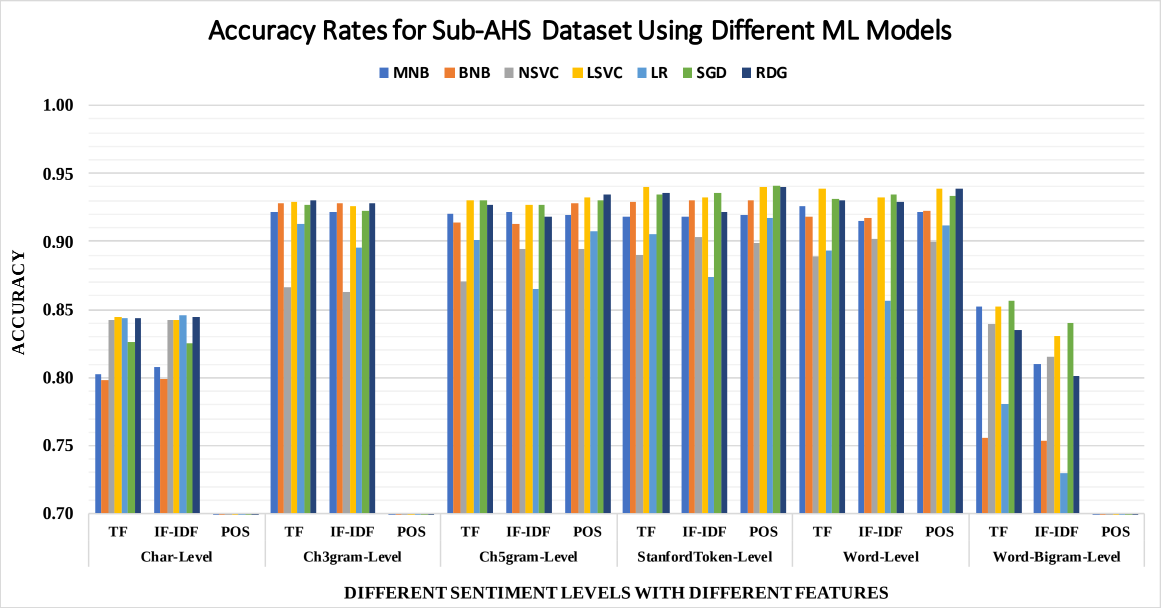


Fig. 6.3 The accuracy for the Sub-AHS dataset using different machine learning models, with the three used features based on different sentiment analysis levels

the other hand, the NSVC, LSVC, LR, and RDG classifiers have the highest classification results that are around 0.81 for Main-AHS and 0.84 for the Sub-AHS. In general, all the used ML algorithms have similar results using both TF and TF-IDF features. In the Word-Bigram-Level, the lowest accuracy is about 0.71 for Main-AHS using LR classifier and TF-IDF feature selection and 0.73 for Sub-AHS. The BNB method is the second lowest results for both feature selections (TF and TF-IDF) that are about 0.74 for the Main-AHS and around 0.75 for the Sub-AHS. In contrast, the powerful ML classifier is SGDC using TF feature selection and the results are around 0.84 for the Main-AHS and about 0.85 for the Sub-AHS. The TF have better results compared with the TF-IDF in all used ML classifiers in Word-Bigram-Level. The variations between the two features are slightly large except in the BNB method because the features are converted to binary numbers either (0 and 1) instead of float number. There is a `binarize` parameter in `naive_bayes.BernoulliNB` class in sklearn package with a threshold equals 0.5 to convert the feature over the threshold to 1, otherwise to 0.

The results in the Ch3gram-Level, the Ch5gram-Level, the StanfordToken-Level, and the Word-Level are generally almost similar in both graphs. The range of the results in these levels are between 0.84 and 0.91 for the Main-AHS and between 0.85 and 0.94 for the Sub-AHS. In all the four different levels, the LSVC shows the greatest classification results for the Main-AHS and the RDG, SGDC, and LSVC are the best used algorithms in the Sub-AHS. The StanfordToken-Level has the highest accuracy, which is over 0.91 in the Main-AHS, and it is the only level that reaches 0.94 for the Sub-AHS. In addition, in the Ch5gram-Level, the StanfordToken-Level, and the Word-Level where we used POS tagging as a feature selection, it almost presents better features extractions compared with TF and TF-IDF.

As a result, the length of the documents in the Char-Level is large compared with other levels and the number of features are limited to Arabic characters only. These caused the low classification results as we used TF and TF-IDF feature selection. In the Word-Bigram-Level, the features are large and the term frequency of one feature is rare due to the flexibility in the word order in Arabic. Therefore, the classification results in this level are low using the different ML methods.

6.4 Convolutional Neural Network (CNN) Models

CNN are feed-forward neural network and they consist of different layers that are: the input layer, the convolutional layer, the pooling layer, the fully connected layer, and the output predictions layer.

- **Input Layer:** it contains the vectors of the input data and in our case are the tokens of the text based on different levels. This layer must have a fixed number of vectores and dimensions. The number of vectors in the text data will be based on the longest document in our case (tweet). Table 6.8 shows the number of vectors, where each vectore represents a token based on different sentiment analysis levels.

Table 6.8 The size of the longest tweets in AHS dataset based on different levels

Sentiment analysis levels	**The size of the longest tweet**
Char-Level	241
Ch3gram-Level	148
Ch5gram-Level	87
StanfordToken-Level	67
Word-Level	52
Word-Bigram-Level	51

- **Convolutional Layer:** there are different window sizes in this layer that slide or convolve over a matrix in order to generate the features. It uses nonlinear activation functions like *ReLU* or *tanh*. There is a stride window size that can have an effect on generating the features, which is the size of shifting the windows over the matrix. The windows can also be called kernels, filters, or feature detectors.

- **Pooling Layer:** also called the *subsampling* or *downsampling layer* and is used after the *Convolutional Layer*. It generates subsamples from the Convolutional Layer using a `max` function for each filter. It reduces the dimensions and generates the important features. The `max` function is the most commonly used function in the pooling layer, where some users might use other functions, such as `average`, *sum*.

- **Fully Connected Layer:** this can be called the *affine layer*, and it contains neurons that have full connections to the previous layer. It takes the outputs' high-level features from the convolutional and pooling layers as input in order to classify them based on classes of the dataset. It applies the softmax activation function.

Each layer might be repeated depending on the structure of the network and the complexity of the problem that needs to be solved. CNN shows robust learning in computer vision systems, such as in Li et al. (2015) and Lin et al. (2015).

Kim (2014) introduced a text classification model using CNN and the model performance obtained good classification accuracy. This thesis proposes two CNN sentiment classification models, which are in next two subsections.

6.4.1 CNN Model

This is slightly modified from Kim (2014) in order to apply it on both the Main-AHS and Sub-AHS datasets, using different sentiment analysis levels to do the text classification. Figure 6.4 clarifies the structure of the CNN model and the different layers. Also, it shows an example of representing and padding an Arabic tweet in order to classify it as either positive or negative.

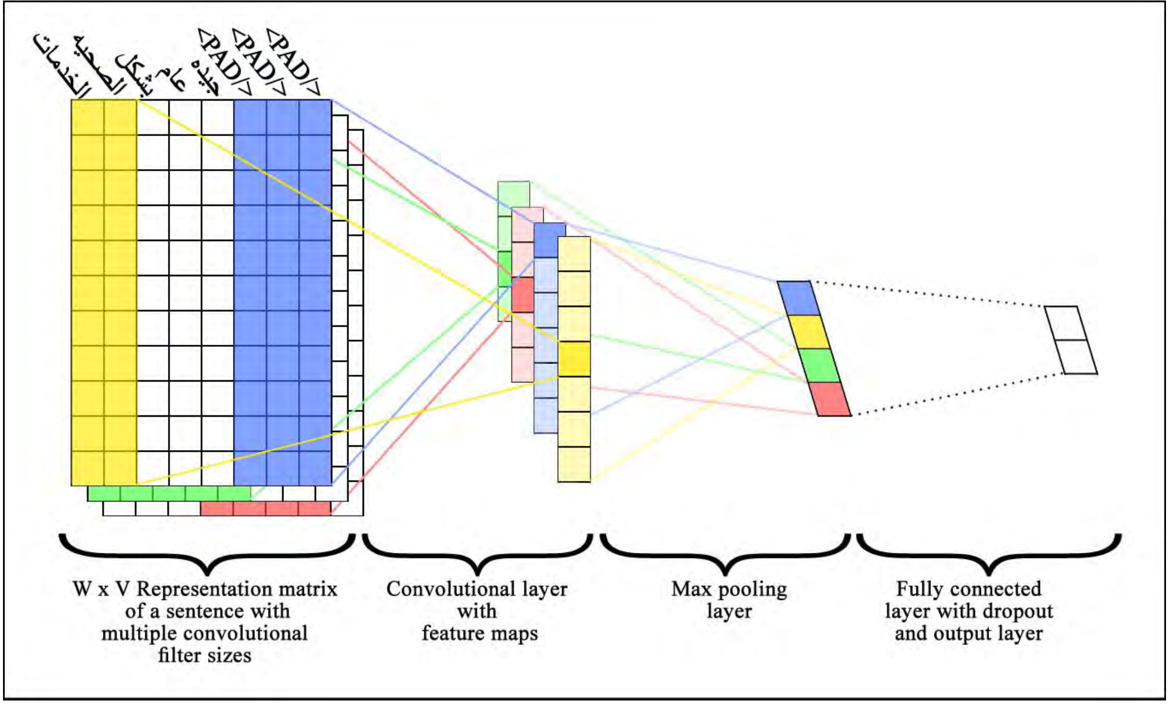

الخدمات
الصحيه
بشكل
عام
جيده
<PAD/>
<PAD/>
<PAD/>
W x V Representation matrix of a sentence with multiple convolutional filter sizes
Convolutional layer with feature maps
Max pooling layer
Fully connected layer with dropout and output layer

Fig. 6.4 CNN model architecture with an example tweet to classify it as either *Positive* or *Negative*.

The steps taken to do the classification process in the CNN model are:

1. Loading the Main-AHS or Sub-AHS dataset files as text files.

2. Filtering the text from special characters, numbers, punctuation, etc.

3. Generating different sentiment levels, as mentioned earlier in this chapter.

4. Calculating the length of the tweets and finding a tweet that has the maximum length.

5. Padding each tweet to the maximum tweet length such as 52 tokens as the maximum length of the tweets when using **Word-Level**. The <*PAD*> token is commonly used in NLP tasks to pad the sentences to the maximum sentence length and here it is the same. The reason for that is to efficiently batch the data as it must have the same length to do the batching.

6. Building a vocabulary index based on all of the vocabularies that occur in the used dataset. Mapping each word in the used dataset to an integer index between 0 and the vocabulary size based on different datasets and sentiment levels, such as 6900 words when using the **Word-Level** for the Main-AHS. Each tweet is represented as a vector of float.

7. Representing each token in the tweet by its word embedding vectors. The word embedding models were trained using the sentiment dataset with the 200 dimension. Each tweet is represented as a $W \times V$ matrix where W is the maximum length of the tweet, such as W = 52 for **Word-Level**, and V is the word embedding dimension (V = 200).

8. Splitting the datasets randomly to 80% for the training set and 20% for the test set.

9. Sliding four filters of different sizes (2, 3, 4, and 5) over the input matrix to generate new features. Also, applying the Rectified Linear Unit (*ReLU*) as a non-linear activation function to obtain the same shape as the feature map.

10. Reducing the dimensionality by using the `max` function in the **Pooling Layer**. It applied to each window result by extracting the maximum element in the feature.

11. Computing the class scores in the **Fully Connected Layer:** using the *softmax* activation function in order to predict the class.

6.4.2 Evaluation and Discussion of the CNN Model

We will evaluate **CNN model** by measuring the accuracy on the test set which is 20% of the whole dataset and it is randomly divided. We measure the performance of the model over 100 epochs using the five different sentiment levels. Table 6.9 compares the accuracy performances of the **CNN Model** using different sentiment analysis levels for both the Main-AHS and the Sub-AHS datasets. The StanfordToken-Level and the Word-Level have the highest results in both datasets compared with other levels.

Table 6.9 The accuracy of the CNN model for the Main-AHS and the Sub-AHS datasets based on the test set using different sentiment levels

Sentiment Analysis Levels	Main-AHS	Sub-AHS
Char-Level	0.8374	0.8576
Ch3gram-Level	0.8892	0.9108
Ch5gram-Level	0.9015	0.9251
StanfordToken-Level	**0.9154**	**0.9395**
Word-Level	0.9113	0.9337
Word-Bigram-Level	0.8596	0.8905

The features in this model are represented by vectors with 200 dimensions. There are four different filters that convolve over the vectors of the tweets, which are represented in different sentiment levels. The sizes of filters are 2, 3, 4, and 5. These filters consider the features that occur next to each other in different sizes of filters. The benefit in this model is that different filter sizes generate different features and then reduce the size of the features. The output of each filter maps the results to the features map. Figure 6.5 and Figure 6.6,

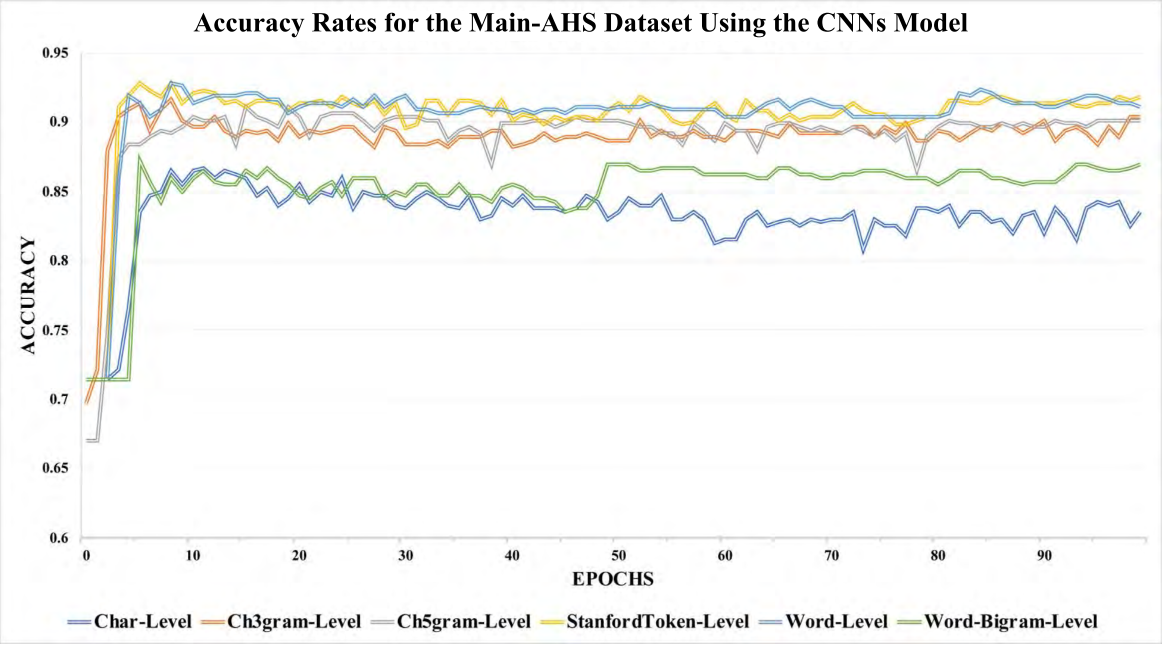


Fig. 6.5 The accuracy of the CNN model on the test set for the Main-AHS dataset using different sentiment analysis levels

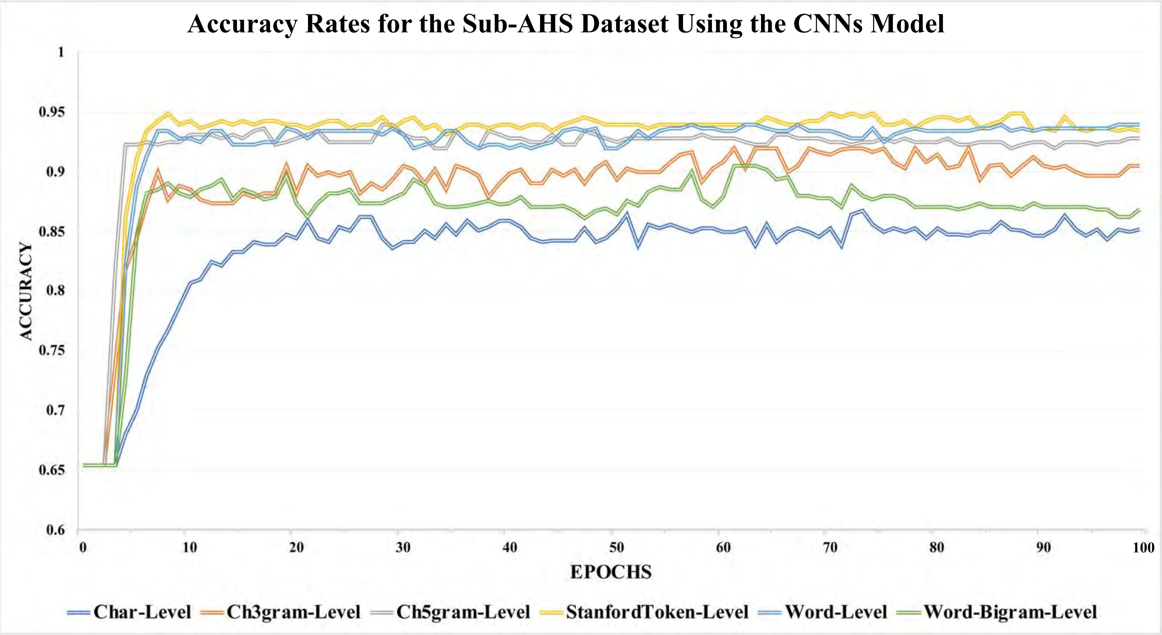


Fig. 6.6 The accuracy of the CNN model on the test set for the Sub-AHS dataset using different sentiment analysis levels

present the accuracy scores for the CNN Model over 100 epochs for both datasets using all different levels.

The Char-Level has the lowest accuracy in both datasets. In this level, each character in the tweets is represented by a vector and this leads to increase the length of the input layer. Also, the size of convolving filters, which capture the features, are small (from 2 to 5) vectors representing (2 to 5) characters only. Therefore, the classification results are low for this level. It might be useful to increase the size of the sliding filters and also to use more than one convolution and pooling layers to achieve better results. However, we used these sizes of filters in order to make appropriate comparison. The Word-Bigram-Level has the second lowest accuracy performance in the Main-AHS and the Sub-AHS. This is due to the flexibility of the word order in Arabic and the vector in this level represents a combination of two words. Consequently, similar combination of two word in the AHS dataset is rarely to occur.

In the Main-AHS dataset, the other levels have very similar results to each other which are between 0.88 and 0.91. In Sub-AHS, the Ch5gram-Level, the StanfordToken-Level, and the Word-Level have very similar scores. The highest accuracy for the CNN Models in the Main-AHS dataset is the StanfordToken-Level which is 0.9154. Also, the best accuracy for the CNN Models in the Sub-AHS dataset is the StanfordToken-Level which is 0.9395.

6.4.3 Lexicon Integrated CNN Model

This is adapted from the *Naive Concatenation model* in Shin et al. (2017) to be used for Arabic sentiment analysis. It has a similar structure to the previous model (CNN model). The only difference is that the lexicon embedding matrix is appended to the end of the of the tweet embedding matrix. Figure 6.7 illustrates the architecture of the **Lexicon Integrated CNN Model**. The same configuration as in the CNN model is used apart from considering and comparing different pre-trained word embedding models that have been explained in Chapter 5.

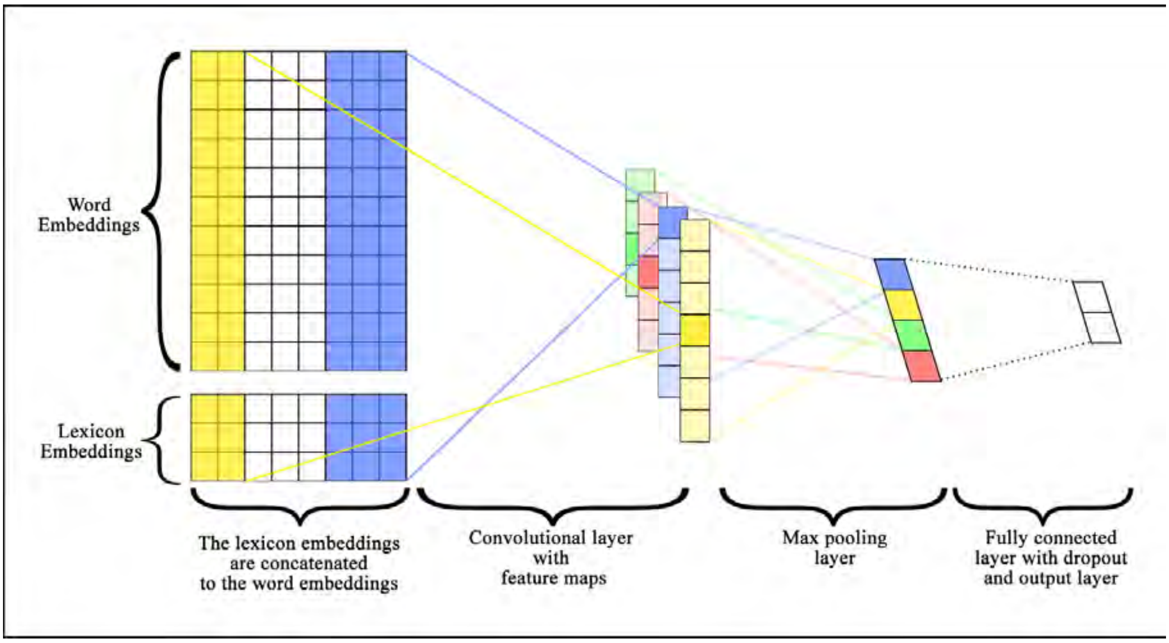

Word Embeddings
Lexicon Embeddings
The lexicon embeddings are concatenated to the word embeddings
Convolutional layer with feature maps
Max pooling layer
Fully connected layer with dropout and output layer

Fig. 6.7 The Lexicon Integrated CNN model architecture, the lexicon embeddings are concatenated to the word embeddings.

Moreover, two different Arabic lexicons were used in this study. First, the *SemEval-2016 Arabic Twitter Lexicon* (Kiritchenko et al., 2016). It contains sentiment words and phrases and it has 1,168 single words, 198 phrases, with the total at 1,366 terms. The sentiment scores are numbers between -1 and 1, where any score indicates the degree of positive and negative sentiment. The second lexicon is the *Arabic Health Twitter Lexicon*, which was built manually for this study and it is based on the Main-AHS dataset. It contains only two values which are -1 for negative terms and 1 for positive terms. The lexicon embedding for this model is built by concatenating both lexicon scores for the word *w*, and if *w* does not occur in one lexicon, 0 is placed as a value.

6.4.4 Evaluation and Discussion of the Lexicon Integrated CNN model

We will measure **Lexicon Integrated CNN model** based on the accuracy of the test set which is 20% of the whole dataset and it is randomly divided. We will use the performance of the model over 100 epochs by considering only the Word-Level because of using the lexicons of words. Also,we will compare the pre-trained word embedding models that were built in Chapter 5 which are; Word2Vec, GloVe, and fastText using the two different Arabic corpora.

Table 6.10 illustrates the accuracy results using the **Lexicon Integrated CNN model** for both datasets. The Word-Level was only used in this experiment because of using lexicons of words in this model. The word vectors in this model is generated from the outputs of proposed pre-trained word embedding models in Chapter 5. The accuracy results that used the word embedding models based on the Twitter Corpus are slightly less than the results of the word embedding models based on the Abu El-khair Corpus. The accuracy scores are between 0.8938 and 0.9284 for the Main-AHS and for the Sub-AHS, the results of the accuracy start from 0.8994 to 0.9502. Overall, the Word2Vec technique using the Abu El-khair Corpus is a superior word embedding option. On the other hand, the fastText word representation model using the Twitter Corpus has inferior results.

Table 6.10 The accuracy of the Lexicon Integrated CNN model for the Main-AHS and Sub-AHS datasets using different word embedding models based on the test set

Word Embedding Models	Main-AHS	Sub-AHS
fastText-Abu El-khair	0.9126	0.9407
fastText-Twitter	0.8938	0.8994
GloVe-Abu El-khair	0.9160	0.9259
GloVe-Twitter	0.9012	0.9037
Word2Vec-Abu El-khair	**0.9284**	**0.9502**
Word2Vec-Twitter	0.9111	0.9175

Figure 6.8 and Figure 6.9, visualise the accuracy performances of the Lexicon Integrated CNN Model over 100 epochs and the length of input vectors is 200 dimensions. The vectors in this model are used from the six pre-trained word representation models in Chapter 5. The

models are Word2Vec-Abu El-khair, Word2Vec-Twitter, GloVe-Abu El-khair, GloVe-Twitter, fastText-Abu El-khair, and fastText-Twitter. Then, we used four sizes of filters for the CNN that are 2, 3, 4, and 5 to extract the features from the input vectors. In this model, we used only word-level because the lexicons contain words and the scores of each word. Thus, it is not appropriate to score characters, sub words, etc. Hence, in this model, we tended to compare the results of using different word embedding techniques with different corpora.

In Figure 6.8, the variation in the lines graphed are very close to each other in the Main-AHS. The Word2Vec word representation technique using the Abu El-khair Corpus has the best sentiment classification performance which is 0.9284. Then, the second best classification models is using GloVe-Abu El-khair, followed by using fastText-Abu El-khair, Word2Vec-Twitter, GloVe-Twitter, and fastText-Twitter. On the other hand, the variation in Figure 6.9 for the Sub-AHS dataset is bigger compared to Figure 6.8. Also, the highest classification accuracy for the Sub-AHS is Word2Vec technique using the Abu El-khair Corpus, which is 0.9502. Likewise, the order of the best used word representation in Sub-AHS is similar to it in Main-AHS except the second best one is fastText-Abu El-khair and the third best used model is GloVe-Abu El-khair.

The performance of sentiment classification using the word embedding models with the Twitter Corpus is slightly lower than Abu El-khair Corpus because of the small size of the corpus. There is a need to collect more tweets and increase the maximum and minimum number of collected tweets to improve the word representation. Another reason might be because of the size of the sliding window over the corpus which is three only because of the short text. We trained different Word2Vec techniques using Abu El-khair Corpus and the fixed-window size was five. There were different dimensions for each Wor2vec models and we determined the appropriate model for the sentiment classification in Section 5.4, which was 200 dimensions. Based on the attributes of the best Word2Vec model we trained the GloVe and fastText models using the same attributes. This might be the reason that Word2Vec has the best sentiment analysis results using the Lexicon Integrated CNN Model.

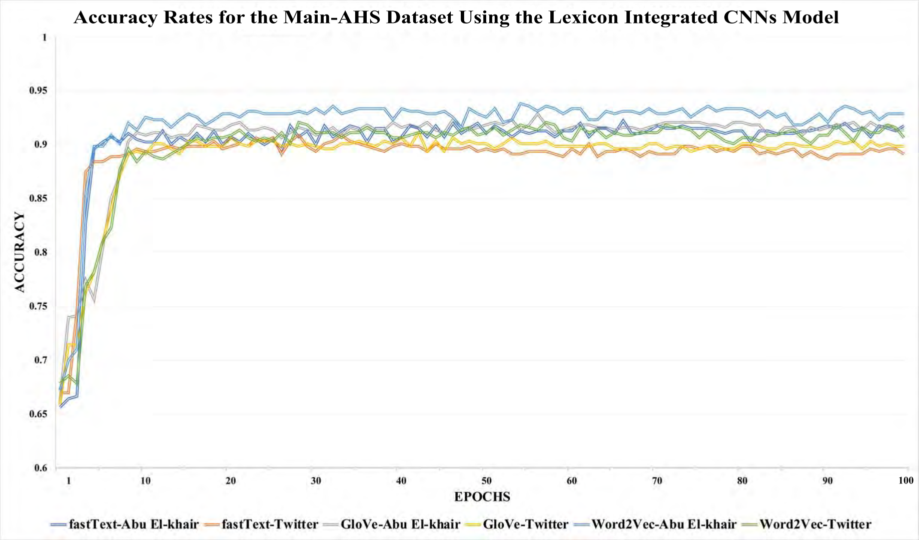


Fig. 6.8 The accuracy of the Lexicon Integrated CNN Model on the test set for the Main-AHS dataset using different word representation models

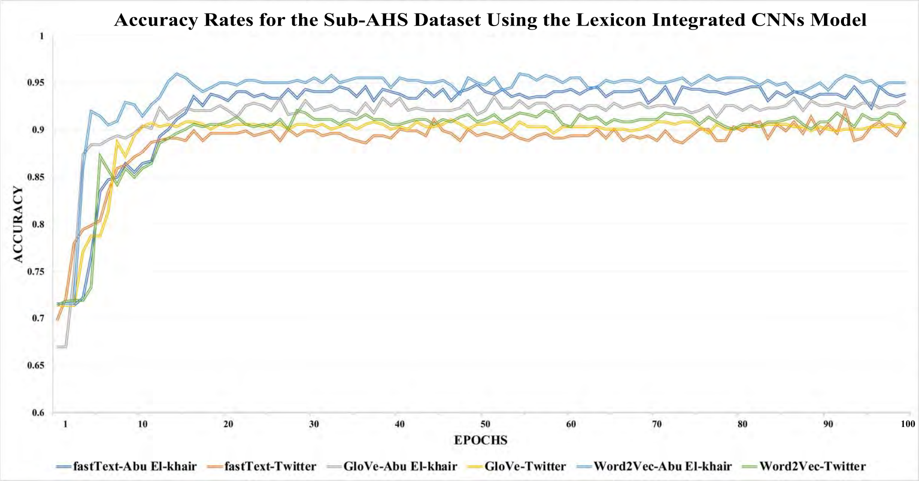


Fig. 6.9 The accuracy of the Lexicon Integrated CNN Model on the test set for the Sub-AHS dataset using different word representation models

6.5 A Combined CNN and LSTM Network Model

6.5.1 CNN and LSTM Model

Wang et al. (2016) proposed a CNN-LSTM model which has two parts: regional CNN and LSTM for prediction. Tang et al. (2015) introduced a model to do the sentiment classification for document level. The model composes the sentence representation from word embedding using CNN or LSTM models. Then, a Backward Gated Neural Network does the document composition from the sentence representation. Finally, the sentiment classification is applied based on the document representation. Zhou et al. (2015) proposed a combined CNN and LSTM model for text classification. A dropout (Srivastava et al., 2014) were used either before feeding the vectors into the CNN or to the output of LSTM to regulate the network and to prevent it from the overfitting. Also, the pooling layer was not applied after the convolutional layer in order to prevent breaking the sequence of vectors. However, in our model, we used a pooling layer to generate the maximum value using a fixed-size of window to keep the sequence of the feature. As a consequence, we will apply a dropout layer to avoid the overfitting and feed the output sequence of the pooling layer to the LSTM.

The combined CNN and LSTM network model in this thesis consists of CNN model and LSTM model. The CNN generates and reduces the feature from the input layer. There are many fixed-size of filters that slide over the input layer to generate the features. The generated features are placed in a feature map in order to select the maximum value within a fixed-size of a window by the Max-pooling. The fixed-size in the Convolution layer and Max-pooling layer is to a fixed-size of features as input to the LSTM. The outputs of the LSTM layer are connected to a fully connected layer to classify the input data. In this model, we consider different sentiment analysis levels to improve the sentiment classification. In Convolution layer, the filter size is changed based on the different sentiment levels. For instance, this thesis used the filter size of 2 for the Word-Bigram-Level, 3 for the Word-Level, and 4 for the StanfordToken-Level. In addition, we used the window sizes 10, 15, and 20 for the Ch5gram-Level, the Ch3gram-Level, and the Char-Level, respectively. As a result, the size of the filters are raised due to the increase in the length for the row of data in each level Alayba et al. (2018c).

Another neural network used in this model is the Long Short Term Memory Network (LSTM), which is one type of the Recurrent Neural Network (RNN). RNNs are intended to learn data using feedback connections and it is ideal for time series and sequential data (Medsker and Jain, 1999). Hochreiter and Schmidhuber (1997) proposed the LSTM and derived it from RNNs by extending the memory of the network. Therefore, it is an appropriate algorithm to learn from data, in which the order of data is important Alayba et al. (2018c).

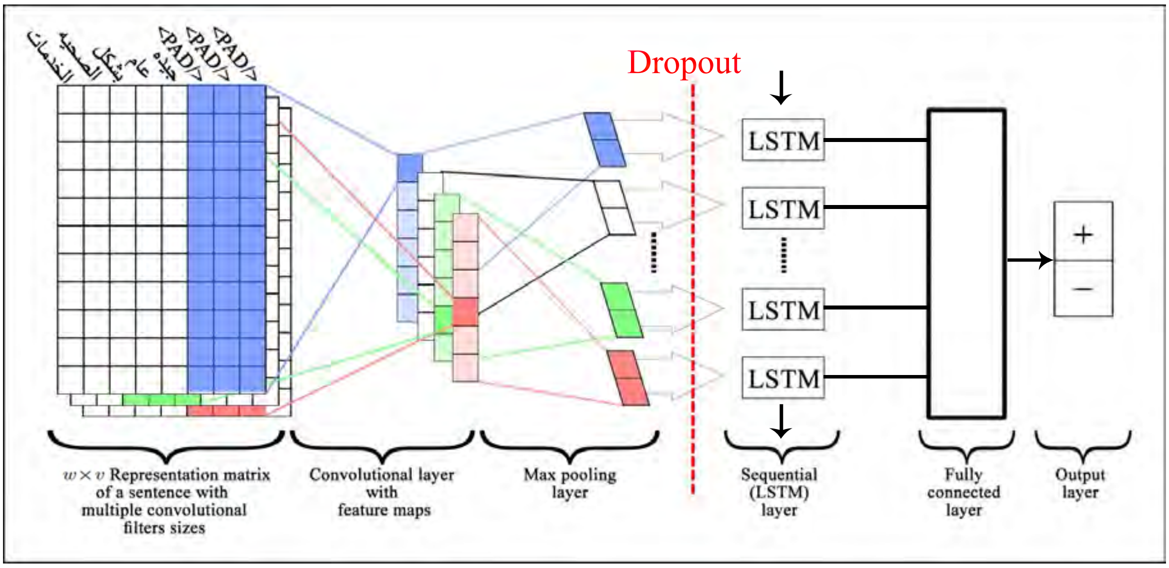


Fig. 6.10 A combination of the CNN and LSTM model architecture for sentiment analysis, with an example of an Arabic tweet.

CNN has an advantage in being able to generate features from the data (Athiwaratkun and Kang, 2015) and LSTM are good for sequential data (Medsker and Jain, 1999). In this model, we propose a combination of two neural networks, which are CNN to generate the features and LSTM to consider the sequential nature of the text. Figure 6.10 presents the combined CNN and LSTM model to classify the Arabic tweets.

The steps to do the sentiment analysis using this model are:

1. Loading the dataset, either the Main-AHS or the Sub-AHS and filtering the text.

2. Applying the different sentiment analysis levels to the dataset.

3. Identifying the maximum row of text in the used dataset. Each level used has a different maximum length. The maximum lengths are 241 for the Char-Level, 148 for the Ch3gram-Level, 87 for the Ch5gram-Level, 67 for the StanfordToken-Level, 52 for the Word-Level, and 51 for the Word-Bigram-Level.

4. Padding the length of all the tweets in the dataset to the maximum tweet length. For instance, the maximum length of a row of text in the dataset using the Ch3ngram-Level is 148. Thus, the word <*PAD*> is added to the end of any tweet that has less than 148.

5. Creating a vocabulary index list and mapping each token to its corresponding vector as we built the vectors from the beginning using either the Main-AHS or the Sub-AHS.

6. Dividing the used dataset into 80% for training and 20% for testing.

7. Convolving a fixed-size of filter and a fixed-number of filters over the input matrix to capture the features. Each sentiment level has different sizes and numbers of filters. The filter sizes are 2, 3, 4, 10, 15, and 20 for the Word-Bigram-Level, the Word-Level, the StanfordToken-Level, the Ch5gram-Level, the Ch3gram-Level, and the Char-Level, respectively. Also, the number of filters are changed based on the different sentiment levels, according to the formula $(L-S)+1$ where L is the length of the input layer and S is the size of the filter.

8. Applying the `max` function in the pooling layer to generate the maximum value in the feature map. The window size of the max pooling is 4.

9. Using the dropout layer to avoid overfitting in the network.

10. Feeding each LSTM unit by the output of each max pooling layer. The number of LSTM units equals the length of the input layer to ensure there are enough units.

11. Merging and concatenating all the LSTM units output to a fully connected layer, which has a single output. Then, applying the sigmoid function (Han and Moraga, 1995) to measure the output between 0 and 1, which is either a negative or positive class.

6.5.2 Evaluation and Discussion of the Combined CNN and LSTM Network Model

In this model, we use the accuracy to measure the sentiment classification. We use also the same size of training and test set as in the previous models. The accuracy results of this model are in Table 6.11 for both the Main-AHS and the Sub-AHS datasets. Using the Word-Level for the Main-AHS dataset achieved the best classification performance at 0.9408. Whereas, the StanfordToken-Level has the most powerful level in the Sub-AHS dataset which is 0.9626.

Table 6.11 The accuracy results of the Combined CNN and LSTM network model for the Main-AHS and Sub-AHS datasets based on the test set using different sentiment levels

Sentiment Analysis Levels	Main-AHS	Sub-AHS
Char-Level	0.8941	0.9164
Ch3gram-Level	0.9015	0.9481
Ch5gram-Level	0.9113	0.9568
StanfordToken-Level	0.9366	**0.9625**
Word-Level	**0.9408**	0.9510
Word-Bigram-Level	0.8621	0.8876

Figure 6.11 and Figure 6.12, present the accuracy over 50 epochs using the combined CNN and LSTM network model for both datasets. We used only 50 epochs in this model because the accuracy results reach almost a steady state after epoch 10 and, so there is no need to have 100 epochs. The Word-Bigram-Level has the lowest scores for the Main-AHS

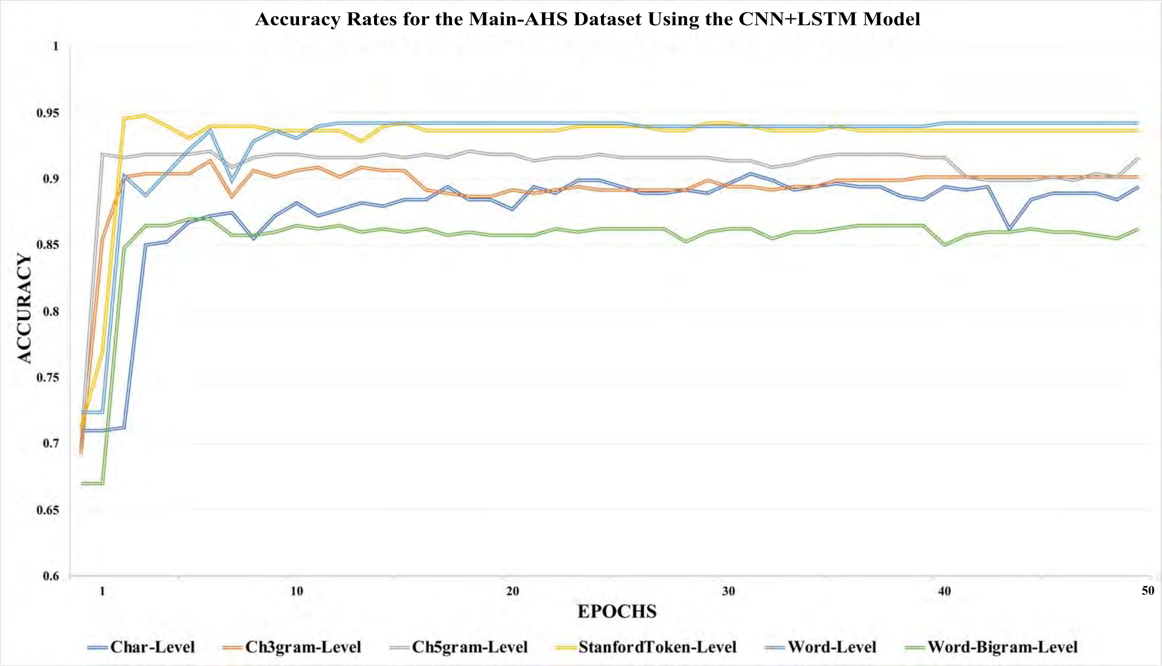


Fig. 6.11 The accuracy of the combined CNN and LSTM network model on the test set for the Main-AHS dataset using different sentiment analysis levels

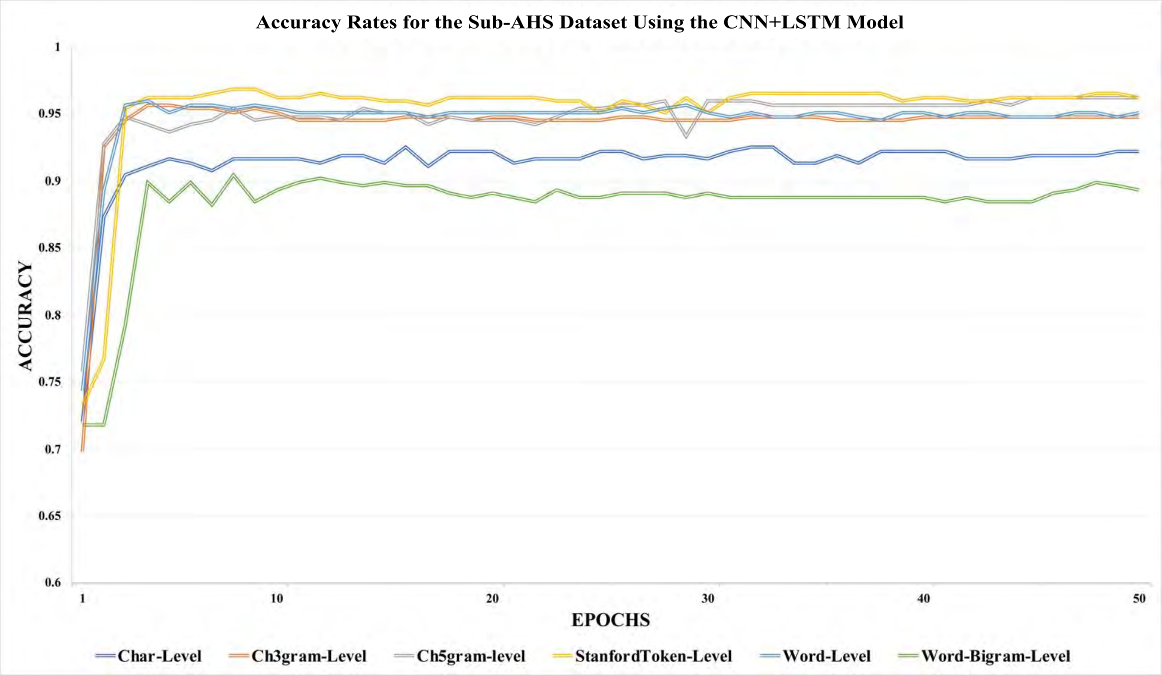


Fig. 6.12 The accuracy of the combined CNN and LSTM network model on the test set for the Sub-AHS dataset using different sentiment analysis levels

and Sub-AHS datasets. The lines charted in Figure 6.11, shows that the StanfordToken-Level and the Word-Level have very similar results which are the best sentiment classification accuracy using the combined CNN and LSTM network model. However, there are four lines that have very similar accuracy scores to each other in Figure 6.12. The lines are: the Ch3gram-Level, the Ch5gram-Level, the StanfordToken-Level, and the Word-Level.

In the combined CNN and LSTM model, the Word-Bigram-Leve has the lowest accuracy, because of the flexibility of the word order in Arabic as we stated in the discussion of previous models. The second lowest accuracy is the Char-Level and this is due to the limited number of vectors that represent the 28 Arabic characters only. Another reason is because the tweets represented in characters level and the size of the tweets is large. We used four small sizes of convolving filters (2, 3, 4, and 5) for the CNN. However, the results of this level improved from the lowest result in CNN Model to be the second lowest results. This improvements are due to the usage of LSTM with CNN for time service data, where it considers the sequence of characters. In the Ch3gram-Level, the accuracy is the third lowest results for both datasets because the number of features increased, which leads to make noise in the features. The Word-Level has the best sentiment classification performance for the Main-AHS and the StanfordToken-Level has the best result for the Sub-AHS. As a result, the StanfordToken-Level is the best used sentiment level in the combined CNN and LSTM network model for sentiment analysis.

6.6 Summary

This chapter introduces different sentiment analysis levels and feature selections that we used in the sentiment classification. Then, it describes different deep neural network models and other machine learning models that we used in this study. Also, it evaluates and discusses the results of all the sentiment analysis models. In Appendix A, we provide all the obtained results for all the used models in three tables. In Char-Level, the reason for having low classification results are that the length of the tweets is large. Also, the number of features is limited to 28 Arabic letters only, and there is much noise in this level compared with others. In Ch3gram-Level and Ch5gram-Level, we tried to expand the number of features with some words to get the root or the main letters of the words. This leads to a rise in the noise in the dataset especially when using Ch3gram-Level. In the StanfordToken-Level, the text segmentation is very good, where it splits the word roots from the prefixes and suffixes without adding more noise to the dataset. Also, this level reduced the multiple forms of some Arabic words and that led to a very good sentiment classification. In Word-Bigram-Level, due to the flexibility of the word order in Arabic, a combination of two words in the dataset is rare to occur.

In word embedding models, the large corpus (Abu El-khair) shows better sentiment classification results compared with the small corpus (Arabic Twitter corpus). Also, using a small window size to train the word embedding models affect the sentiment analysis. We checked different dimensions and models for Word2Vec and we determined the best one, which was CBOW with 200 dimensions. However, we trained the GloVe and fastText using the same window size and dimensions, and this might be the reason for better classification for Word2Vec.

Chapter 7

Conclusions

7.1 Overview

This thesis addresses the sentiment classification task for Arabic text collected from social networks. Sentiment analysis can be useful for other NLP tasks, such as machine translation (Hutchins, 2003), question answering (Ravichandran and Hovy, 2002), automatic summarisation (Hutchins, 2003), text segmentation (Choi, 2000), etc. Additionally, sentiment analysis has an essential role in predicting sales performance (Liu et al., 2007), spam detection (Jindal and Liu, 2007), predicting elections (Tumasjan et al., 2010), ranking products and merchants based on reviews (McGlohon et al., 2010). Many researchers are encouraged to investigate more in social data and develop many intelligent systems. The literature covers various languages. English is the most popular one, while there is a more limited number of research studies and available datasets in Arabic.

Arabic is one of the most complicated languages due to its richness in morphology, difficulty of orthography, complexity of grammar, number of dialects, etc. Also, there is a lack of tools that can deal with Arabic text for NLP tasks. These limitations lead this study to investigate deeper in this area and identify new ways to improve research.

7.2 Main Contributions

The main contributions of this research are:

1. **Building the Arabic Health Services (AHS) dataset.**

   In order to do that, some tasks were performed to produce the sentiment dataset, which are collecting, filtering, and annotating the tweets. Also, there are several challenges to create and prepare the dataset, such as Arabic text encoding, filtering the tweets and the text, etc. At the beginning, there were difficulties writing Arabic script in Unicode form, especially to create more than one word. Then, it was very hard to retrieve many tweets that contained sentiments in a specific topic, unless there are many trending hashtags on this topic. After filtering the dataset, it was very challenging to judge whether the tweets contain an opinion or not. Twitter does not have a review scale like on Amazon, Booking.come, etc., which makes annotating the tweets a very complicated task. Therefore, the tweets were annotated based on the view of each annotator, and the final sentiment was measured based on majority voting. Some tweets were not agreed on by all three annotators, as being either positive or negative Alayba et al. (2017) .

2. **Training different word representation models.**

   We trained different word embedding models, using three techniques on two different Arabic corpuses. The techniques are Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014a), and fastText (Bojanowski et al., 2017). The corpuses are the Abu El-khair Corpus (Abu El-khair, 2016), which is collected from ten newspapers and the Arabic Twitter Corpus that was built for this study. We examined the effectiveness of using a very large corpus on the sentiment classification task. Also, we considered sentiment analysis based on the way of writing the tweets using the Arabic Twitter Corpus. We applied all different pre-trained word embedding models as a part of the sentiment classification, using the Word-Level only. The variation of the sentiment classification is small when using the Main-AHS dataset, while it is large when using the Sub-AHS dataset. The word embedding model that had the highest sentiment classification accuracy was Word2Vec using the Abu El-khair Corpus Alayba et al. (2018c) .

3. **Utilising different sentiment analysis levels.**

   Due to the lack of NLP tools for the Arabic language, different sentiment analysis levels were proposed. Besides that, we expand the number of features in some levels, as this study focuses on tweets sentiment classification. All the different levels were compared using different features and classifiers. As a result from the sentiment analysis experiments, the Char-Level and the Word-Bigram-Level have shown the lowest accuracy scores in all the sentiment classification models. On the other hand, both the StanfordToken-Level and the Word-Level shared the highest sentiment analysis results, and especially the StanfordToken-Level had the top scores in most of the models Alayba et al. (2018a) .

4. **Developing effective sentiment classifiers**

   This thesis presented three primary types of sentiment classification models. The models were: different machine learning algorithms, the CNN, and the combined CNN and LSTM model. It is a very intricate process to compare the results of the classifiers in different machine learning algorithm sections because of using different features and using different levels. However, the LSVC, RDG, and SGDC have shown the best performance for both datasets in most of the different sentiment levels. The results of the CNN model are almost similar to the results when using different machine learning algorithms, where the top results are 0.91 for the Main-AHS and 0.94 for the Sub-AHS Alayba et al. (2017). There is a slight improvement for the sentiment classification using the Lexicon Integrated CNN model, which increased to 0.9284 for the Main-AHS and to 0.9502 for the Sub-AHS Alayba et al. (2018c). Finally, the last model, which is a combined CNN and LSTM model, achieves the best sentiment classification accuracy compared with all other classifiers. The accuracy results increased to 0.94 for the Main-AHS and 0.96 for the Sub-AHS Alayba et al. (2018a).

7.3 Future Work

This study has detailed the process and the challenges of building an Arabic sentiment dataset. Also, it has investigated the feasibility of using different sentiment analysis level with different classifiers and feature selections. However, the work is continuous and there is a need to explore new areas in Arabic sentiment analysis and other NLP tasks. Moreover, there are few Arabic sentiment datasets available and the amount of data in these datasets is small in comparison to other languages.

It will be interesting to obtain many datasets from different social media platforms. In addition, gathering different datasets for different Arabic dialects and different topics would be very helpful for many reasons. Firstly, it would be very good to train the machine on the various examples of Arabic sentences, because Arabic has a flexible word order. Secondly, it can capture different words from different Arabic dialects, so it would be easy for clustering the synonyms, sentiment words and building Arabic lexicons. Also, the advantage of this can be to build an automatic Arabic system for collecting and classifying the data.

There are few accurate Arabic NLP tools to do the preprocessing on Arabic text. It would be useful to investigate this task more, and provide some improvements. There are some tools discussed in some studies, such as Hadni et al. (2013), Althobaiti et al. (2014), Al-Kabi et al. (2015), etc. As mentioned in Chapter 3, Arabic words have multiple forms, such as the word مستشفى "hospital", which occurs in 42 different forms in the Main-AHS. All these forms are represented differently before the classification, thus the classifier deals with them as different words. It is necessary to rely on convenient NLP tools for Arabic to normalise or reduce the number of forms.

Twitter is an easy source to collect data from. Moreover, the length of a tweet has been expanded to 280 characters recently, which provides the users with more characters to write their tweets. However, there are many users that use this platform to post their advertisements for products, services, etc. Some studies consider this issue in detecting spam tweets for the Arabic language, such as Abozinadah et al. (2015), El-Mawass and Alaboodi (2016), and Abozinadah and Jones (2017). It is worth improving Arabic spam tweet detection techniques, which will help Twitter to prevent the spread of spam accounts, and for filtering the data as well.

It would be a good idea to expand our Arabic Health Services dataset. As health is an important topic everywhere, and feedback about health services is required to attend to patients' needs. It would be interesting to collect the tweets about health from different Arab countries to cover many Arabic dialects, share experiences and improve decision making. This will lead to an increase in the number of classes to three classes: positive, negative and neutral. Also, it might be advantageous to have five classes that are very positive, positive, neutral, negative, very negative. This may help to improve the measurement of the sentiment score for the words in the dataset.

References

Abdul-Mageed, M. and Diab, M. (2012). AWATIF: A Multi-Genre Corpus for Modern Standard Arabic Subjectivity and Sentiment Analysis. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA).

Abdulla, N. A., Ahmed, N. A., Shehab, M. A., and Al-Ayyoub, M. (2013). Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2013*, pages 1–6. IEEE.

Abozinadah, E. A. and Jones, Jr., J. H. (2017). A statistical learning approach to detect abusive twitter accounts. In *Proceedings of the International Conference on Compute and Data Analysis*, ICCDA '17, pages 6–13, New York, NY, USA. ACM.

Abozinadah, E. A., Mbaziira, A. V., and Jones, J. H. J. (2015). Detection of Abusive Accounts with Arabic Tweets. *International Journal of Knowledge Engineering-IACSIT*, 1(2):113–119.

Abu El-khair, I. (2016). 1.5 billion words Arabic corpus. *arXiv:1611.04033[cs]*.

Abu El-khair, I. (2019). *Abu El-Khair Corpus*. Arabic Portal, http://abuelkhair.net/index.php/en/arabic/abu-el-khair-corpus. accessed 21/01/2019.

Adedoyin-Olowe, M., Gaber, M. M., and Stahl, F. (2014). A Survey of Data Mining Techniques for Social Media Analysis. *Journal of Data Mining & Digital Humanities*.

Agarwal, B., Mittal, N., Bansal, P., and Garg, S. (2015). Sentiment Analysis Using Common-Sense and Context Information. *Computational Intelligence and Neuroscience*, 2015(30):1–9.

Aggarwal, C. C. (2011). *Social Network Data Analytics*. Springer Publishing Company, Incorporated, 1st edition.

Aghdam, H. H. and Heravi, E. J. (2017). *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*. Springer Publishing Company, Incorporated, 1st edition.

Ahmed, S., Pasquier, M., and Qadah, G. (2013). Key issues in conducting sentiment analysis on Arabic social media text. In *2013 9th International Conference on Innovations in Information Technology (IIT)*, pages 72–77. IEEE.

Ahmed, W., Bath, P. A., and Gianluca, D. (2017). Chapter 4 Using Twitter as a Data Source: An Overview of Ethical, Legal, and Methodological Challenges. In Woodfield, K., editor, *The Ethics of Online Research. Advances in Research Ethics and Integrity (2)*, pages 79–107. Emerald.

Al-Kabi, M. N., Kazakzeh, S. A., Abu Ata, B. M., Al-Rababah, S. A., and Alsmadi, I. M. (2015). A Novel Root Based Arabic stemmer. *Journal of King Saud University - Computer and Information Sciences*, 27(2):94–103.

Al-Nashashibi, M. Y. (2014). *Arabic Language Processing for Text Classification. Contributions to Arabic Root Extraction Techniques, Building An Arabic Corpus, and to Arabic Text Classification Techniques*. Ph.D. University of Bradford.

Al-Nashashibi, M. Y., Neagu, D., and Yaghi, A. A. (2010). An improved root extraction technique for Arabic words. In *2nd International Conference on Computer Technology and Development*, pages 264–269, Cairo, Egypt.

Al Sallab, A., Hajj, H. M., Badaro, G., Baly, R., El-Hajj, W., and Shaban, K. B. (2015). Deep Learning Models for Sentiment Analysis in Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 9–17, Beijing, China. ACL.

Al-Shammari, E. and Lin, J. (2008). A Novel Arabic Lemmatization Algorithm. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data*, pages 113–118, Singapore. ACM.

Alansary, S. and Nagi, M. (2014). The International Corpus of Arabic: Compilation, Analysis and Evaluation. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Langauge Processing (ANLP)*, pages 8–17, Doha, Qatar. Association for Computational Linguistics.

Alasem, A. (2015). egovernment on twitter: The use of twitter by the saudi authorities. *The Electronic Journal of e-Government*, 13(1):67–74.

Alayba, A., Palade, V., England, M., and Iqbal, R. (2017). Arabic Language Sentiment Analysis on Health Services. In *1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pages 114–118, Nancy, France. IEEE.

Alayba, A., Palade, V., England, M., and Iqbal, R. (2018a). A Combined CNN and LSTM Model for Arabic Sentiment Analysis. In *Machine Learning and Knowledge Extraction. CD-MAKE 2018. Lecture Notes in Computer Science*, pages 179–191, Hamburg, Germany. Springer, Cham.

Alayba, A., Palade, V., England, M., and Iqbal, R. (2018b). Arabic Health Services (AHS) dataset. https://bitbucket.org/a_alayba/arabic-health-services-ahs-dataset/src. accessed 10/01/2019.

Alayba, A., Palade, V., England, M., and Iqbal, R. (2019). Arabic-twitter-corpus. https://github.com/alaybaa/Arabic-Twitter-Corpus.git. accessed 22/01/2019.

Alayba, A. M., Palade, V., England, M., and Iqbal, R. (2018c). Improving Sentiment Analysis in Arabic Using Word Representation. In *2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, pages 13–18, London, UK. IEEE.

Alhumoud, S. O., Altuwaijri, M. I., Albuhairi, T. M., and Alohaideb, W. M. (2015). Survey on Arabic Sentiment Analysis in Twitter. *International Journal of Social, Behavioral, Educational, Economic and Management Engineering*, 9(1):364–368.

Alsaad, A. and Abbod, M. (2014). Arabic Text Root Extraction via Morphological Analysis and Linguistic Constraints. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pages 125–130. IEEE.

Althobaiti, M., Kruschwitz, U., and Poesio, M. (2014). AraNLP: a Java-based Library for the Processing of Arabic Text. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Alwakid, G., Osman, T., and Hughes-Roberts, T. (2017). Challenges in Sentiment Analysis for Arabic Social Networks. *Procedia Computer Science*, 117:89 – 100. Arabic Computational Linguistics.

Aly, M. and Atiya, A. (2013). LABR: A Large Scale Arabic Book Reviews Dataset. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 494–498, Sofia, Bulgaria. ACL.

Aramaki, E., Maskawa, S., and Morita, M. (2011). Twitter catches the flu: Detecting influenza epidemics using twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1568–1576, Stroudsburg, PA, USA. Association for Computational Linguistics.

Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., and Levitan, S. (2007). Stylistic Text Classification Using Functional Lexical Features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2015). RAND-WALK: A Latent Variable Model Approach to Word Embeddings. *arXiv:1502.03520[cs].*

Arras, L., Montavon, G., Müller, K., and Samek, W. (2017). Explaining Recurrent Neural Network Predictions in Sentiment Analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168, Copenhagen, Denmark. Association for Computational Linguistics.

Athiwaratkun, B. and Kang, K. (2015). Feature Representation in Convolutional Neural Networks. *arXiv:1507.02313[cs.CV].*

Attia, M., Pecina, P., Toral, A., Tounsi, L., and van Genabith, J. (2011). A Lexical Database for Modern Standard Arabic Interoperable with a Finite State Morphological Transducer. In Mahlow C. and Piotrowski M., editors, *Systems and Frameworks for Computational Morphology. SFCM 2011. Communications in Computer and Information Science*, volume 100, pages 98–118. Springer, Berlin, Heidelberg.

Balaji, P., Nagaraju, O., and Haritha, D. (2017). Levels of sentiment analysis and its challenges: A literature review. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, volume 6, pages 436–439. IEEE.

Bermingham, A. and Smeaton, A. F. (2011). On Using Twitter to Monitor Political Sentiment and Predict Election Results. In *the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP) at the International Joint Conference for Natural Language Processing (IJCNLP)*, pages 2–10, Chiang Mai, Thailand.

Bian, J., Gao, B., and Liu, T.-Y. (2014). Knowledge-Powered Deep Learning for Word Embedding. In *Machine Learning and Knowledge Discovery in Databases*, volume 5211, pages 132–148.

Bifet, A. and Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data. In Pfahringer B., Holmes G., and Hoffmann A., editors, *Discovery Science. DS 2010. Lecture Notes in Computer Science*, volume 6332. Springer, Berlin, Heidelberg.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Bottou, L. (1998). On-line Learning and Stochastic Approximations. In Saad, D., editor, *On-Line Learning in Neural Networks*, pages 9–42. Cambridge University Press, Cambridge.

Boyd, D. M. and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230.

Bradley, M. M., Lang, P. J., Bradley, M. M., and Lang, P. J. (1999). Affective Norms for English Words (ANEW): Instruction manual and affective ratings . Technical report, The Center for Research in Psychophysiology, University of Florida, Gainesville, FL.

Chapelle, O., Schlkopf, B., and Zien, A. (2010). *Semi-Supervised Learning*. The MIT Press, Cambridge, MA, 1st edition.

Chen, J., Tandon, N., and de Melo, G. (2015). Neural Word Representations from Large-Scale Commonsense Knowledge. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 225–228. IEEE.

Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 26–33, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Dahou, A., Xiong, S., Zhou, J., Haddoud, M. H., and Duan, P. (2016). Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification. In *Proceedings of the 26th International Conference on Computational Linguistics, COLING 2016*, pages 2418–2427, Osaka, Japan. The COLING 2016 Organizing Committee.

De Boom, C., Van Canneyt, S., Demeester, T., and Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recogn. Lett.*, 80:150–156.

Dey, L., Chakraborty, S., Biswas, A., Bose, B., and Tiwari, S. (2016). Sentiment Analysis of Review Datasets Using Naïve Bayes‘ and K-NN Classifier. *International Journal of Information Engineering and Electronic Business*, 8(4):54–62.

Dodds, P. and Danforth, C. (2011). Measuring the happiness of large-scale written expression: Songs, blogs, and presidents. *Journal of Happiness Studies*, 11(4):441 – 456.

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Darrell, T., and Saenko, K. (2015). Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 39, pages 2625–2634. IEEE.

dos Santos, C. N. and Gatti, M. (2014). Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, Dublin, Ireland.

Drozd, A., Gladkova, A., and Matsuoka, S. (2016). Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530. The COLING 2016 Organizing Committee.

Ehrlich, K. and Carboni, I. (2005). Inside social network analysis. Technical report, IBM.

El-Mawass, N. and Alaboodi, S. (2016). Detecting Arabic spammers and content polluters on Twitter. In *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, pages 53–58, Beirut, Lebanon. IEEE.

El-Shishtawy, T. and El-Ghannam, F. (2012). An Accurate Arabic Root-Based Lemmatizer for Information Retrieval Purposes. *International Journal of Computer Science Issues*, 9(3).

Facebook AI Research Team (2016). *fastText Word representations.* fastText v0.1.0, https://fasttext.cc/docs/en/unsupervised-tutorial.html. accessed 21/01/2019.

Farghaly, A. and Shaalan, K. (2009). Arabic natural language processing : Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):1–22.

Feinerer, G., Hornik, K., and Software, A. (2015). *Package ‘tm’: Text Mining Package. R package version 0.2-8.* R package version 0.2-8, https://CRAN.R-project.org/package=tm. accessed 14/09/2016.

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS’13, pages 2121–2129, Lake Tahoe, Nevada, USA. Curran Associates Inc.

Gamallo, P. and Garcia, M. (2014). Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 171–175. Association for Computational Linguistics.

Gantz, J. and Reinsel, D. (2012). The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. *IDC: Analyze the Future*, 2007:1–16.

Gentry, J. (2016). *Package 'twitteR': R Based Twitter Client.* R package version 1.1.9, https://cran.r-project.org/package=twitteR. accessed 01/09/2016.

Gers, F. and Schmidhuber, E. (2001). LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340.

Geys, B. and Murdoch, Z. (2010). Measuring the 'Bridging' versus 'Bonding' Nature of Social Networks: A Proposal for Integrating Existing Measures. *Sociology*, 44(3):523–540.

Ghahramani, Z. (2004). Unsupervised Learning. In Bousquet O., von Luxburg U., and Rätsch G., editors, *Advanced Lectures on Machine Learning. ML 2003. Lecture Notes in Computer Science*, volume 3176, pages 72–112. Springer, Berlin, Heidelberg.

Ghannay, S., Favre, B., Estève, Y., and Camelin, N. (2016). Word embedding evaluation and combination. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 300–305, Paris, France. European Language Resources Association (ELRA).

Ghorbel, H. and Jacot, D. (2011). Sentiment Analysis of French Movie Reviews. In *Advances in Distributed Agent-Based Retrieval Tools (DART 2010)*, pages 97–108, Geneva, Switzerland. Springer, Berlin, Heidelberg.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. Technical report, Stanford.

Golub, D. and He, X. (2016). Character-Level Question Answering with Attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1598–1607, Stroudsburg, PA, USA. Association for Computational Linguistics.

Graves, A., Jaitly, N., and Mohamed, A. (2013). Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278. IEEE.

Green, S. and Manning, C. D. (2010). Better Arabic Parsing: Baselines, Evaluations, and Analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 394–402. Association for Computational Linguistics.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2017). Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232.

Haddi, E., Liu, X., and Shi, Y. (2013). The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*, 17:26 – 32. First International Conference on Information Technology and Quantitative Management.

Hadni, M., Ouatik, S. A., and Lachkar, A. (2013). Effective Arabic Stemmer Based Hybrid Approach for Arabic Text Categorization. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, 3(4):1–14.

Hamouda, S. B. and Akaichi, J. (2013). Social Networks' Text Mining for Sentiment Classification : The case of Facebook' statuses updates in the "Arabic Spring" Era. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, 2(5):470–478.

Han, J. and Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In *Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, IWANN '96, pages 195–201, London, UK. Springer-Verlag.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hogenboom, A., Bal, D., Frasincar, F., Bal, M., de Jong, F., and Kaymak, U. (2013). Exploiting emoticons in sentiment analysis. In *the 28th Annual ACM Symposium on Applied Computing SAC '13*, pages 703–710, Coimbra, Portugal. ACM.

Hoppe, B. and Reinelt, C. (2010). Social network analysis and the evaluation of leadership networks. *The Leadership Quarterly*, 21(4):600 – 619.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science Engineering*, 9(3):90–95.

Hutchins, J. (2003). Machine Translation: General Overview. In Mitkov, R., editor, *The Oxford Handbook of Computational Linguistics*, pages 501–511. Oxford University Press, Oxford.

Ibrahim, H. S., Abdou, S. M., and Gheith, M. (2015). Sentiment Analysis for Modern Standard Arabic and Colloquial. *International Journal on Natural Language Computing (IJNLC)*, 4(2):95–109.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*, volume 103 of *Springer Texts in Statistics*. Springer New York, New York, NY, USA.

Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 151–160. Association for Computational Linguistics.

Jianqiang, Z. and Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, 5:2870–2879.

Jindal, N. and Liu, B. (2007). Review spam detection. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1189–1190, New York, NY, USA. ACM.

Jindal, N. and Liu, B. (2008). Opinion spam and analysis. In *the 2008 International Conference on Web Search and Data Mining WSDM '08*, pages 219–230, Palo Alto, California, USA. ACM.

Jivani, M. A. G. (2011). A Comparative Study of Stemming Algorithms. *International Journal of Computer Technology and Applications*, 2(6):1930–1938.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. http://www.scipy.org/. accessed 14/09/2016.

Joshi, A., Prabhu, A., Shrivastava, M., and Varma, V. (2016). Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491. The COLING 2016 Organizing Committee.

Kalman, B. L. and Kwasny, S. C. (1992). Why tanh: choosing a sigmoidal function. In *Proceedings 1992 IJCNN International Joint Conference on Neural Networks*, volume 4, pages 578–581, Baltimore, MD, USA.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Kiritchenko, S., Mohammad, S., and Salameh, M. (2016). SemEval-2016 Task 7: Determining Sentiment Intensity of English and Arabic Phrases. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 42–51. Association for Computational Linguistics.

Kirk, M. (2014). *Thoughtful Machine Learning: A Test-Driven Approach*. O'Reilly Media, Sebastopol, CA, USA.

Kumar, A. and Sebastian, T. M. (2012). Sentiment Analysis on Twitter. *International Journal of Computer Science Issues IJCSI*, 9(4):372–378.

Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web - WWW '10*, page 591, New York, New York, USA. ACM Press.

Lakomkin, E., Bothe, C., and Wermter, S. (2017). GradAscent at EmoInt-2017: Character and Word Level Recurrent Neural Network Models for Tweet Emotion Intensity Detection. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 169–174, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lampos, V. and Cristianini, N. (2010). Tracking the flu pandemic by monitoring the social web. In *2010 2nd International Workshop on Cognitive Information Processing*, pages 411–416, Elba, Italy. IEEE.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Lebanon, G., Mao, Y., and Dillon, J. (2007). The locally weighted bag of words framework for document representation. *Journal of Machine Learning Research*, 8:2405–2441.

Lebret, R. and Collobert, R. (2014). Word embeddings through hellinger pca. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–490. Association for Computational Linguistics.

Lee, K., Palsetia, D., Narayanan, R., Patwary, M. M. A., Agrawal, A., and Choudhary, A. (2011). Twitter trending topic classification. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 251–258, Washington, DC, USA. IEEE Computer Society.

Leonard Richardson (2004-2015). *Beautiful Soup Documentation.* Beautiful Soup v4.4.0, https://www.crummy.com/software/BeautifulSoup/bs4/doc/. accessed 21/01/2019.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Li, D. and Qian, J. (2016). Text sentiment analysis based on long short-term memory. In *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, pages 471–475. IEEE.

Li, H., Lin, Z., Shen, X., Brandt, J., and Hua, G. (2015). A convolutional neural network cascade for face detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5325–5334, Boston, MA, USA. IEEE.

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78. Association for Computational Linguistics.

Lin, T., RoyChowdhury, A., and Maji, S. (2015). Bilinear cnn models for fine-grained visual recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1457, Santiago, Chile. IEEE.

Liu, B. (2010). Sentiment analysis: A multifaceted problem. *IEEE Intelligent Systems*, 25(3):76–80.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining.* Morgan & Claypool Publishers.

Liu, Y., Huang, X., An, A., and Yu, X. (2007). Arsa: A sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 607–614, New York, NY, USA. ACM.

Luong, T., Sutskever, I., Le, Q., Vinyals, O., and Zaremba, W. (2015). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19. Association for Computational Linguistics.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning ICML*, volume 28, Atlanta, Georgia, USA.

Maimon, O. and Rokach, L. (2005). *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag, Berlin, Heidelberg.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Marin, A. and Wellman, B. (2011). Social Network Analysis: An Introduction. In *The SAGE Handbook of Social Network Analysis*, pages 11–25. SAGE Publications Ltd, London, United Kingdom.

McGlohon, M., Glance, N., and Reiter, Z. (2010). Star quality: Aggregating reviews to rank products and merchants. In *Proceedings of Fourth International AAAI Conference on Weblogs and Social Media*, Washington, D.C., USA. Association for the Advancement of Artificial Intelligence AAAI Press.

McNamara, J. M., Green, R. F., and Olsson, O. (2006). Bayes' theorem and its applications in animal behaviour. *OIKOS*, 112(2):243–251.

Medsker, L. R. and Jain, L. C. (1999). *Recurrent Neural Networks: Design and Applications*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv:1301.3781[cs.CL]*.

Milli, S. (2016). Py-corenlp. https://pypi.org/project/stanfordcorenlp/. accessed 13/07/2016.

Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B. (2007). Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement - IMC '07*, page 29, New York, New York, USA. ACM Press.

Monroe, W., Green, S., and Manning, C. D. (2014). Word Segmentation of Informal Arabic with Domain Adaptation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 206–211, Baltimore, Maryland, USA. Association for Computational Linguistics.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press, 1st edition.

Nabil, M., Aly, M., and Atiya, A. (2015). ASTD: Arabic Sentiment Tweets Dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519, Lisbon, Portugal. Association for Computational Linguistics.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, pages 807–814.

Nalisnick, E., Mitra, B., Craswell, N., and Caruana, R. (2016). Improving Document Ranking with Dual Word Embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*, pages 83–84, New York, New York, USA. ACM Press.

Narayanan, V., Arora, I., and Bhatia, A. (2013). Fast and accurate sentiment classification using an enhanced naive bayes model. In *Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning - IDEAL 2013*, pages 194–201. Springer-Verlag New York, Inc.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2007). Textual Affect Sensing for Sociable and Expressive Online Communication. In Paiva A.C.R., Prada R., and Picard R.W., editors, *Affective Computing and Intelligent Interaction. ACII 2007*, volume 4738, pages 218–229. Springer, Berlin, Heidelberg.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'17, pages 6341–6350, Long Beach, CA, USA. Curran Associates Inc.

Olshannikova, E., Olsson, T., Huhtamäki, J., and Kärkkäinen, H. (2017). Conceptualizing Big Social Data. *Journal of Big Data*, 4(3):1–19.

Otte, E. and Rousseau, R. (2002). Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6):441–453.

Owen, A. B. (2006). A robust hybrid of lasso and ridge regression. Technical report, Stanford University.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. a. (2013). Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of NAACL-HLT 2013*, pages 380–390. Association for Computational Linguistics.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 1320–1326. European Languages Resources Association (ELRA).

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, volume 10, pages 79–86, Morristown, NJ, USA. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennebaker, J., Boyd, R., Jordan, K., and Blackburn, K. (2015). The development and psychometric properties of LIWC2015. Technical report, University of Texas at Austin.

Pennington, J., Socher, R., and Manning, C. (2014a). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014b). GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/. accessed 26/04/2017.

Poria, S., Cambria, E., and Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

Powers, D. (2011). Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Python Software Foundation (1990-2019). *The ElementTree XML API (xml.etree.ElementTree).* xml.etree.ElementTree v2.5, https://docs.python.org/2/library/markup.html. accessed 21/01/2019.

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 41–47. Association for Computational Linguistics.

Refaee, E. and Rieser, V. (2014). An Arabic Twitter Corpus for Subjectivity and Sentiment Analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2268–2273, Reykjavik, Iceland. European Language Resources Association (ELRA).

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. European Language Resources Association (ELRA).

Ritterman, J., Osborne, M., and Klein, E. (2009). Using prediction markets and twitter to predict a swine flu pandemic. In *Proceedings of the 1st International Workshop of Mining Social Media*, pages 9–17.

Roberts, K., Roach, M. A., Johnson, J., Guthrie, J., and Harabagiu, S. M. (2012). Empatweet: Annotating and detecting emotions on twitter. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3806–3813. European Language Resources Association (ELRA).

Sadilek, A., Kautz, H., and Silenzio, V. (2012). Predicting disease transmission from geotagged micro-blog data. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 136–142. Association for the Advancement of Artificial Intelligence AAAI Press.

Saif, H., Fernandez, M., He, Y., and Alani, H. (2014). On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, (i):810–817.

Saif, H., He, Y., and Alani, H. (2012). Semantic sentiment analysis of twitter. In *Proceedings of the 11th International Conference on The Semantic Web*, ISWC'12, pages 508–524. Springer-Verlag.

Salamah, J. B. and Elkhlifi, A. (2014). Microblogging Opinion Mining Approach for Kuwaiti Dialect. In *The International Conference on Computing Technology and Information Management (ICCTIM2014)*, pages 388–396. SDIWC Digital Library.

Sayeed, A. B., Boyd-Graber, J., Rusk, B., and Weinberg, A. (2012). Grammatical structures for word-level sentiment detection. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 667–676, Montreal, Canada.

Seman, S. (2014). Organizational Member Use of Social Networking Sites and Work Productivity. *International Journal of Innovation, Management and Technology*, 5(1):30–34.

Shin, B., Lee, T., and Choi, J. D. (2017). Lexicon Integrated CNN Models with Attention for Sentiment Analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 149–158, Copenhagen, Denmark. Association for Computational Linguistics.

Singh, T. and Kumari, M. (2016). Role of text pre-processing in twitter sentiment analysis. *Procedia Computer Science*, 89:549 – 554.

Smrž, O. (2016). Encode Arabic Online Interface. http://quest.ms.mff.cuni.cz/cgi-bin/encode/index.fcgi. accessed 03/01/2017.

Socher, R., Perelygin, A., and Wu, J. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Song, F. and Croft, W. B. (1999). A General Language Model for Information Retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 316–321, New York, NY, USA. ACM.

Speriosu, M., Sudan, N., Upadhyay, S., and Baldridge, J. (2011). Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 53–63, Stroudsburg, PA, USA. Association for Computational Linguistics.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal. Association for Computational Linguistics.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning Sentiment-Specific Word Embedding. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565, Baltimore, Maryland, USA. Association for Computational Linguistics.

Timmaraju, A. and Khanna, V. (2015). Sentiment analysis on movie reviews using recursive and recurrent neural network architectures. *Semantic Scholar*.

TNS (2015). Arab Social media Influencers Summit. Technical report, The Arab Social Media Influencers Summit.

Tumasjan, A., Sprenger, T., Sandner, P., and Welpe, I. (2010). Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185, Washington, D.C., USA. Association for the Advancement of Artificial Intelligence AAAI Press.

Tunggawan, E. and Soelistio, Y. E. (2016). And the winner is...: Bayesian twitter-based prediction on 2016 u.s. presidential election. In *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pages 33–37, Tangerang, Indonesia. IEEE.

Twitter (2018). Twitter Developers. https://developer.twitter.com/. accessed 20/08/2018.

Ularu, E. G., Puican, F. C., Apostu, A., and Velicanu, M. (2012). Perspectives on Big Data and Big Data Analytics. *Database Systems Journal*, III(4):3–14.

Van der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, 13(2):22–30.

Vanhoucke, V., Senior, A., and Mao, M. Z. (2011). Improving the speed of neural networks on CPUs. In *Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*, pages 1–8, Granada, Spain.

Wang, J., Yu, L.-C., Lai, K. R., and Zhang, X. (2016). Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 225–230, Berlin, Germany. Association for Computational Linguistics.

Webster, J. J. and Kit, C. (1992). Tokenization as the initial phase in nlp. In *Proceedings of the 14th Conference on Computational Linguistics*, COLING '92, pages 1106–1110, Stroudsburg, PA, USA. Association for Computational Linguistics.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 625–631, Bremen, Germany. ACM.

Wiebe, J. M., Bruce, R. F., and O'Hara, T. P. (1999). Development and Use of a Gold-Standard Data Set for Subjectivity Classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics -*, pages 246–253, Morristown, NJ, USA. Association for Computational Linguistics.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wood, A. (2015). Alan Wood's Unicode Resources. http://www.alanwood.net/unicode/. accessed 01/12/2015.

Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., and Weston, J. (2017). StarSpace: Embed All The Things! *arXiv:1709.03856 [cs]*, abs/1709.03856.

Yadav, S., Ekbal, A., Saha, S., and Bhattacharyya, P. (2018). Medical Sentiment Analysis using Social Media: Towards building a Patient Assisted System. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2790–2797, Miyazaki, Japan. European Language Resources Association (ELRA).

Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*, pages 197–206, New York, NY, USA. ACM.

Yu, L.-C., Wang, J., Lai, K. R., and Zhang, X. (2017). Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 534–539. Association for Computational Linguistics.

Zaghouani, W. (2014). Critical Survey of the Freely Available Arabic Corpora. In *Proceedings of the International Conference on Language Resources and Evaluation*, (LREC'2014), pages 26–31, Reykjavik, Iceland. European Language Resources Association.

Zahran, M. A., Magooda, A., Mahgoub, A. Y., Raafat, H., Rashwan, M., and Atyia, A. (2015). Word Representations in Vector Space and their Applications for Arabic. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing. CICLing 2015*, volume 9041, pages 430–443. Springer, Cham.

Zerrouki, T. (2012). Tashaphyne, Arabic light stemmer. https://pypi.python.org/pypi/Tashaphyne/0.2. accessed 06/03/2017.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., and Liu, B. (2011). Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis. Technical report, HP Laboratories.

Zhang, M.-L., Peña, J. M., and Robles, V. (2009). Feature Selection for Multi-label Naive Bayes Classification. *Information Sciences*, 179(19):3218–3229.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 649–657, Montreal, Canada. MIT Press.

Zhou, C., Sun, C., Liu, Z., and Lau, F. C. (2015). A C-LSTM Neural Network for Text Classification. *arXiv:1511.08630[cs]*.

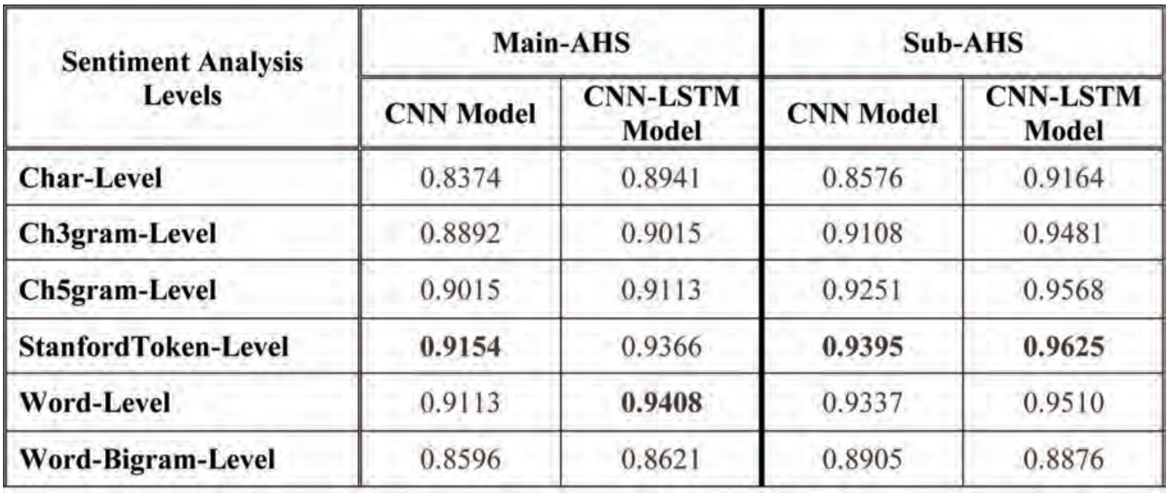
Appendix A

Summary of the Results for All Sentiment Classification Models

Table A.1 Summary of all the accuracies for ML classifiers with the three feature selections (TF, TF-IDF, POS) for both Main-AHS and Sub-AHS

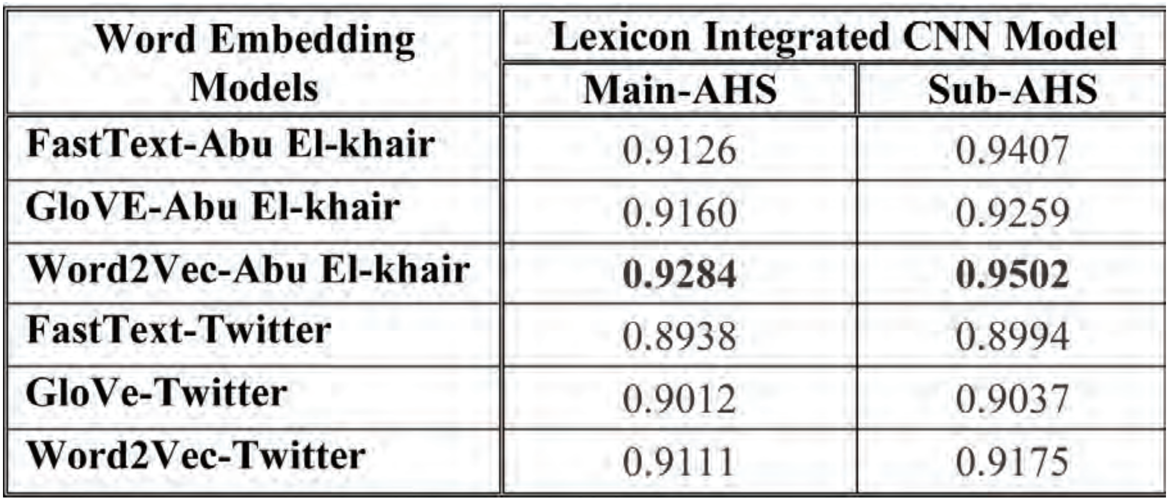
The Dataset	The MLs	The Features	Char-Level	Ch3 gram-Level	Ch5 gram-Level	Stanford Token-Level	Word-Level	Word-Bigram-Level
Main-AHS	MNB	TF	0.7904	0.8914	0.8914	0.8914	0.8919	0.8352
		TF-IDF	0.7983	0.8855	0.8924	0.8904	0.8865	0.8006
		POS	N/A	N/A	0.8879	0.8890	0.8889	N/A
	BNB	TF	0.7700	0.8974	0.8815	0.8968	0.8914	0.7414
		TF-IDF	0.7745	0.8958	0.8835	0.9002	0.8870	0.7408
		POS	N/A	N/A	0.8933	0.9018	0.8884	N/A
	NSVC	TF	**0.8199**	0.8618	0.8569	0.8707	0.8736	0.8201
		TF-IDF	0.8130	0.8894	0.8919	0.8988	0.8963	0.7929
		POS	N/A	N/A	0.8825	0.8860	0.8667	N/A
	LSVC	TF	0.8198	0.9018	0.9008	0.9121	0.9018	0.8376
		TF-IDF	0.8164	**0.9072**	0.9017	0.9091	0.9013	0.8169
		POS	N/A	N/A	**0.9032**	**0.9132**	**0.9037**	N/A
	LR	TF	0.8189	0.8885	0.8761	0.8855	0.8722	0.7730
		TF-IDF	0.8164	0.8722	0.8504	0.8568	0.8455	0.7182
		POS	N/A	N/A	0.8845	0.8904	0.8850	N/A
	SGDC	TF	0.8070	0.8929	0.8954	0.9102	0.8973	**0.8435**
		TF-IDF	0.8031	0.8973	0.8918	0.9047	0.8983	0.8223
		POS	N/A	N/A	0.8954	0.9063	0.8934	N/A
	RDG	TF	0.8139	0.8998	0.8993	0.9043	0.9018	**0.8223**
		TF-IDF	0.8149	0.8998	0.8948	0.8953	0.8905	0.7942
		POS	N/A	N/A	0.9022	0.9097	0.9028	N/A
Sub-AHS	MNB	TF	0.8026	0.9215	0.9209	0.9180	0.9255	0.8522
		TF-IDF	0.8076	0.9221	0.9220	0.9186	0.9146	0.8101
		POS	N/A	N/A	0.9191	0.9192	0.9220	N/A
	BNB	TF	0.7985	0.9284	0.9140	0.9296	0.9186	0.7558
		TF-IDF	0.7997	0.9284	0.9128	0.9301	0.9169	0.7535
		POS	N/A	N/A	0.9278	0.9302	0.9226	N/A
	NSVC	TF	0.8429	0.8666	0.8707	0.8903	0.8886	0.8396
		TF-IDF	0.8429	0.8632	0.8944	0.9030	0.9019	0.8151
		POS	N/A	N/A	0.8949	0.8984	0.8995	N/A
	LSVC	TF	0.8452	0.9296	0.9307	0.9405	0.9388	0.8528
		TF-IDF	0.8430	0.9261	0.9272	0.9319	0.9324	0.8308
		POS	N/A	N/A	0.9324	0.9400	**0.9394**	N/A
	LR	TF	0.8441	0.9128	0.9007	0.9053	0.8938	0.7812
		TF-IDF	**0.8459**	0.8961	0.8649	0.8735	0.8568	0.7304
		POS	N/A	N/A	0.9070	0.9174	0.9117	N/A
	SGDC	TF	0.8262	0.9267	0.9302	0.9347	0.9313	**0.8568**
		TF-IDF	0.8251	0.9232	0.9272	0.9359	0.9348	0.8407
		POS	N/A	N/A	0.9307	**0.9406**	0.9330	N/A
	RDG	TF	0.8435	**0.9307**	0.9273	0.9353	0.9302	0.8349
		TF-IDF	0.8447	0.9284	0.9180	0.9221	0.9296	0.8014
		POS	N/A	N/A	**0.9347**	0.9405	**0.9394**	N/A

Table A.2 Summary of all the accuracies for CNN model and CNN and LSTM model with different sentiment levels for both Main-AHS and Sub-AHS



Sentiment Analysis Levels	Main-AHS		Sub-AHS	
	CNN Model	CNN-LSTM Model	CNN Model	CNN-LSTM Model
Char-Level	0.8374	0.8941	0.8576	0.9164
Ch3gram-Level	0.8892	0.9015	0.9108	0.9481
Ch5gram-Level	0.9015	0.9113	0.9251	0.9568
StanfordToken-Level	**0.9154**	0.9366	**0.9395**	**0.9625**
Word-Level	0.9113	**0.9408**	0.9337	0.9510
Word-Bigram-Level	0.8596	0.8621	0.8905	0.8876

Table A.3 All the accuracies for Lexicon Integrated CNN model with different word embedding models for both Main-AHS and Sub-AHS



Word Embedding Models	Lexicon Integrated CNN Model	
	Main-AHS	Sub-AHS
FastText-Abu El-khair	0.9126	0.9407
GloVE-Abu El-khair	0.9160	0.9259
Word2Vec-Abu El-khair	**0.9284**	**0.9502**
FastText-Twitter	0.8938	0.8994
GloVe-Twitter	0.9012	0.9037
Word2Vec-Twitter	0.9111	0.9175

Appendix B

The Most Similar Words to the Words "*good*" and "*bad*" in Arabic Using Word2Vec

Table B.1 The most similar words to the word *good* in Arabic using the Word2Vec technique *CBOW* for different dimensionalities: 10, 50, 100, 200, and 300

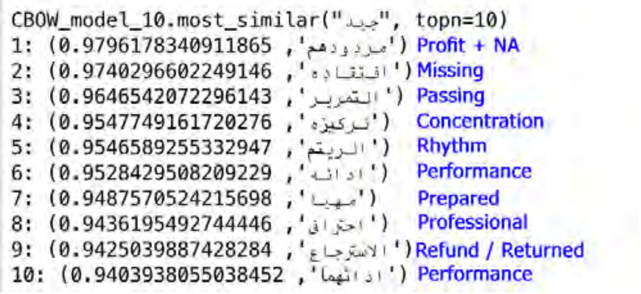
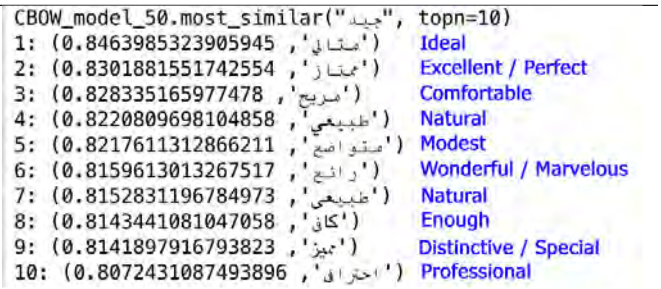
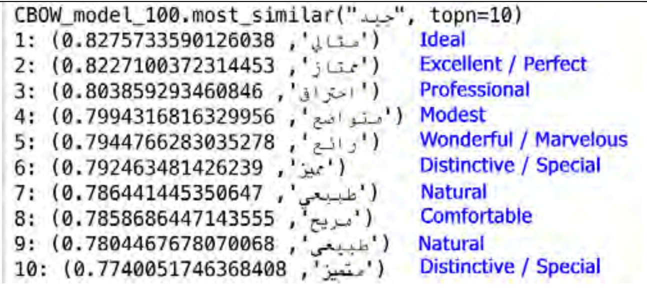
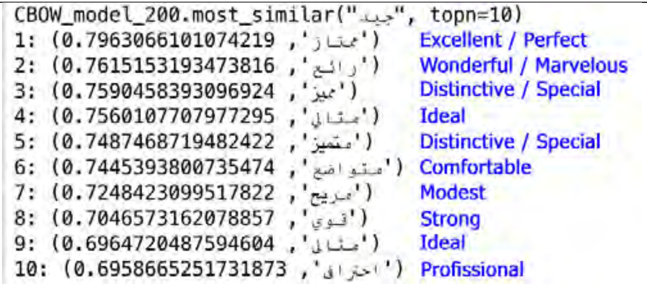
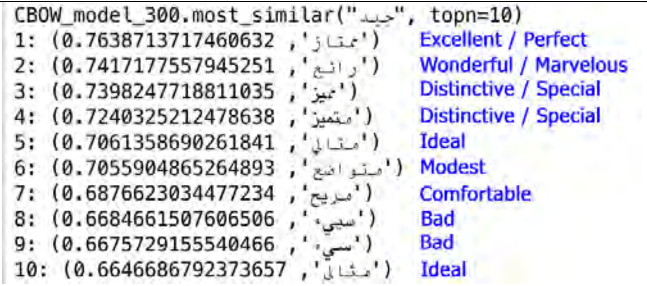
dimensionality	**Good "جد"**
10	 CBOW_model_10.most_similar("جيد", topn=10) 1: (0.9796178340911865 ,'مردودهم') Profit + NA 2: (0.9740296602249146 ,'افتقاده') Missing 3: (0.9646542072296143 ,'التمرير') Passing 4: (0.9547749161720276 ,'تركيزه') Concentration 5: (0.9546589255332947 ,'الريتم') Rhythm 6: (0.9528429508209229 ,'اداءه') Performance 7: (0.9487570524215698 ,'مهيا') Prepared 8: (0.9436195492744446 ,'احتراق') Professional 9: (0.9425039887428284 ,'الاسترجاع') Refund / Returned 10: (0.9403938055038452 ,'اداءهما') Performance
50	 CBOW_model_50.most_similar("جيد", topn=10) 1: (0.8463985323905945 ,'مثالي') Ideal 2: (0.8301881551742554 ,'ممتاز') Excellent / Perfect 3: (0.828335165977478 ,'مريح') Comfortable 4: (0.8220809698104858 ,'طبيعي') Natural 5: (0.8217611312866211 ,'متواضع') Modest 6: (0.8159613013267517 ,'رائع') Wonderful / Marvelous 7: (0.8152831196784973 ,'طبيعى') Natural 8: (0.8143441081047058 ,'كاف') Enough 9: (0.8141897916793823 ,'مميز') Distinctive / Special 10: (0.8072431087493896 ,'احتراق') Professional
100	 CBOW_model_100.most_similar("جيد", topn=10) 1: (0.8275733590126038 ,'مثالي') Ideal 2: (0.8227100372314453 ,'ممتاز') Excellent / Perfect 3: (0.803859293460846 ,'احتراق') Professional 4: (0.7994316816329956 ,'متواضع') Modest 5: (0.7944766283035278 ,'رائع') Wonderful / Marvelous 6: (0.792463481426239 ,'مميز') Distinctive / Special 7: (0.786441445350647 ,'طبيعي') Natural 8: (0.7858686447143555 ,'مريح') Comfortable 9: (0.7804467678070068 ,'طبيعى') Natural 10: (0.7740051746368408 ,'متميز') Distinctive / Special
200	 CBOW_model_200.most_similar("جيد", topn=10) 1: (0.7963066101074219 ,'ممتاز') Excellent / Perfect 2: (0.7615153193473816 ,'رائع') Wonderful / Marvelous 3: (0.7590458393096924 ,'مميز') Distinctive / Special 4: (0.7560107707977295 ,'مثالي') Ideal 5: (0.7487468719482422 ,'متميز') Distinctive / Special 6: (0.7445393800735474 ,'متواضع') Comfortable 7: (0.7248423099517822 ,'مريح') Modest 8: (0.7046573162078857 ,'قوي') Strong 9: (0.6964720487594604 ,'مثالى') Ideal 10: (0.6958665251731873 ,'احتراق') Profissional
300	 CBOW_model_300.most_similar("جيد", topn=10) 1: (0.7638713717460632 ,'ممتاز') Excellent / Perfect 2: (0.7417177557945251 ,'رائع') Wonderful / Marvelous 3: (0.7398247718811035 ,'مميز') Distinctive / Special 4: (0.7240325212478638 ,'متميز') Distinctive / Special 5: (0.7061358690261841 ,'مثالي') Ideal 6: (0.7055904865264893 ,'متواضع') Modest 7: (0.6876623034477234 ,'مريح') Comfortable 8: (0.6684661507606506 ,'سيىء') Bad 9: (0.6675729155540466 ,'سيء') Bad 10: (0.6646686792373657 ,'مثالى') Ideal

Table B.2 The most similar words to the word *bad* in Arabic using the Word2Vec technique *CBOW* for different dimensionalities: 10, 50, 100, 200, and 300

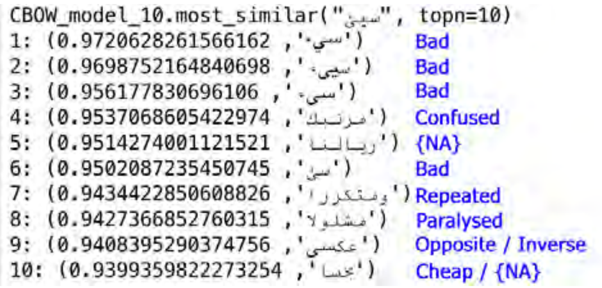
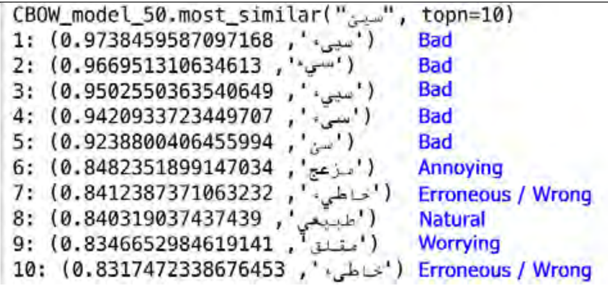
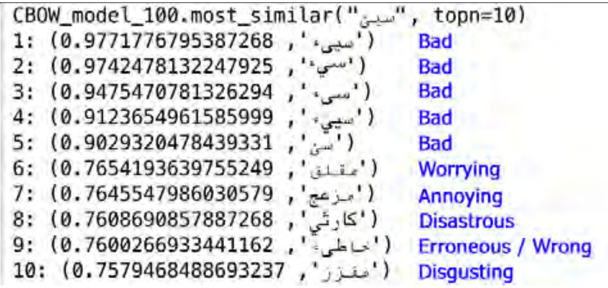
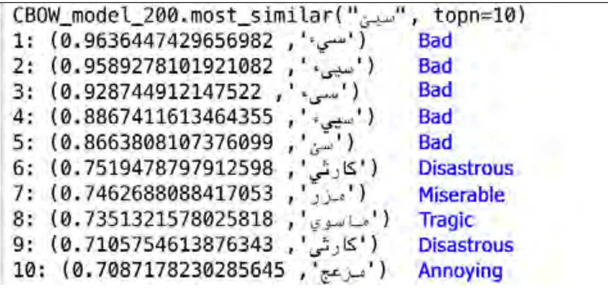
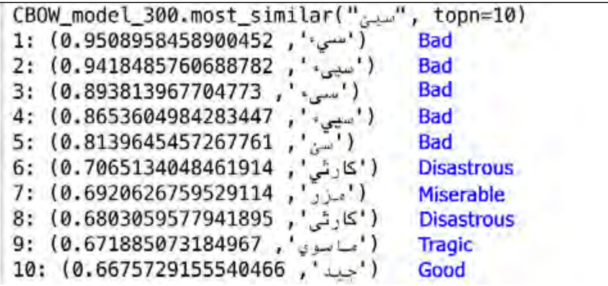
dimensionality	**Bad "سئ"**
10	 CBOW_model_10.most_similar("سيئ", topn=10) 1: (0.9720628261566162 ,'سيء') Bad 2: (0.9698752164840698 ,'سيىء') Bad 3: (0.956177830696106 ,'سىء') Bad 4: (0.9537068605422974 ,'مرتبك') Confused 5: (0.9514274001121521 ,'ريالنا') {NA} 6: (0.9502087235450745 ,'سئ') Bad 7: (0.9434422850608826 ,'ومتكررا') Repeated 8: (0.9427366852760315 ,'مشلولا') Paralysed 9: (0.9408395290374756 ,'عكسي') Opposite / Inverse 10: (0.9399359822273254 ,'بخسا') Cheap / {NA}
50	 CBOW_model_50.most_similar("سيئ", topn=10) 1: (0.9738459587097168 ,'سيىء') Bad 2: (0.966951310634613 ,'سيء') Bad 3: (0.9502550363540649 ,'سييء') Bad 4: (0.9420933723449707 ,'سىء') Bad 5: (0.9238800406455994 ,'سئ') Bad 6: (0.8482351899147034 ,'مزعج') Annoying 7: (0.8412387371063232 ,'خاطيء') Erroneous / Wrong 8: (0.840319037437439 ,'طبيعي') Natural 9: (0.8346652984619141 ,'مقلق') Worrying 10: (0.8317472338676453 ,'خاطىء') Erroneous / Wrong
100	 CBOW_model_100.most_similar("سيئ", topn=10) 1: (0.9771776795387268 ,'سيىء') Bad 2: (0.9742478132247925 ,'سيء') Bad 3: (0.9475470781326294 ,'سىء') Bad 4: (0.9123654961585999 ,'سييء') Bad 5: (0.9029320478439331 ,'سئ') Bad 6: (0.7654193639755249 ,'مقلق') Worrying 7: (0.7645547986030579 ,'مزعج') Annoying 8: (0.7608690857887268 ,'كارثي') Disastrous 9: (0.7600266933441162 ,'خاطىء') Erroneous / Wrong 10: (0.7579468488693237 ,'مقزز') Disgusting
200	 CBOW_model_200.most_similar("سيئ", topn=10) 1: (0.9636447429656982 ,'سيء') Bad 2: (0.9589278101921082 ,'سيىء') Bad 3: (0.928744912147522 ,'سىء') Bad 4: (0.8867411613464355 ,'سييء') Bad 5: (0.8663808107376099 ,'سئ') Bad 6: (0.7519478797912598 ,'كارثي') Disastrous 7: (0.7462688088417053 ,'مزر') Miserable 8: (0.7351321578025818 ,'ماسوي') Tragic 9: (0.7105754613876343 ,'كارثى') Disastrous 10: (0.7087178230285645 ,'مزعج') Annoying
300	 CBOW_model_300.most_similar("سيئ", topn=10) 1: (0.9508958458900452 ,'سيء') Bad 2: (0.9418485760688782 ,'سيىء') Bad 3: (0.893813967704773 ,'سىء') Bad 4: (0.8653604984283447 ,'سييء') Bad 5: (0.8139645457267761 ,'سئ') Bad 6: (0.7065134048461914 ,'كارثي') Disastrous 7: (0.6920626759529114 ,'مزر') Miserable 8: (0.6803059577941895 ,'كارثى') Disastrous 9: (0.671885073184967 ,'ماسوي') Tragic 10: (0.6675729155540466 ,'جيد') Good

Table B.3 The most similar words to the word *good* in Arabic using the Word2Vec technique *SG* for different dimensionalities: 10, 50, 100, 200, and 300

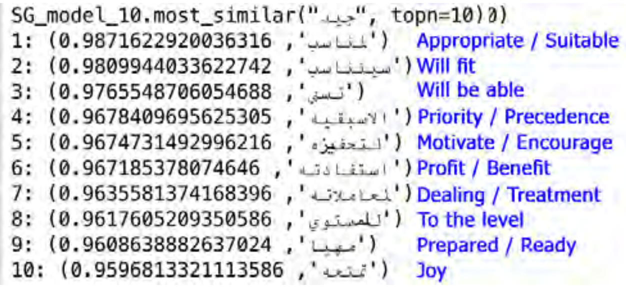
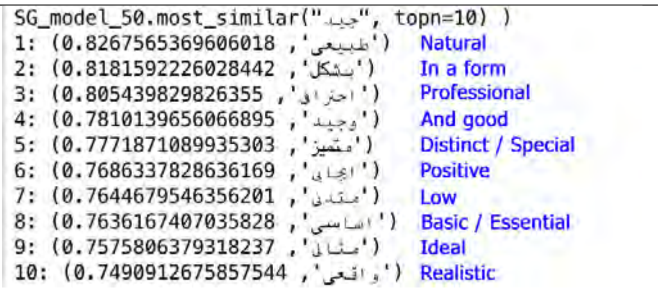
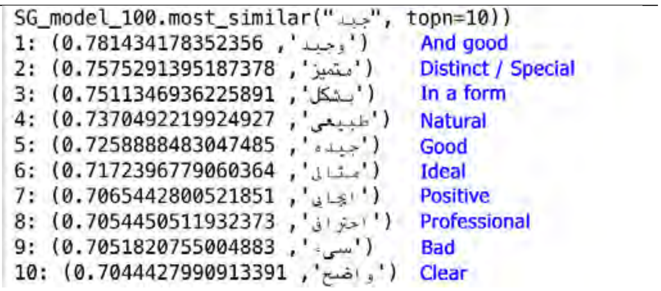
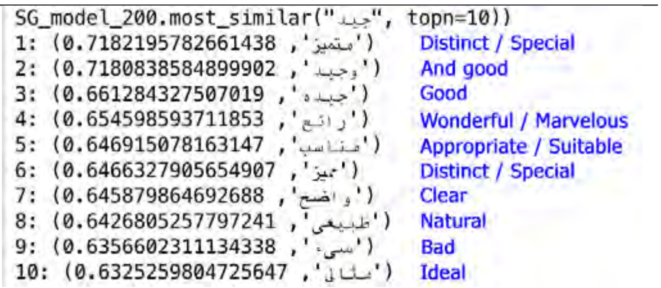
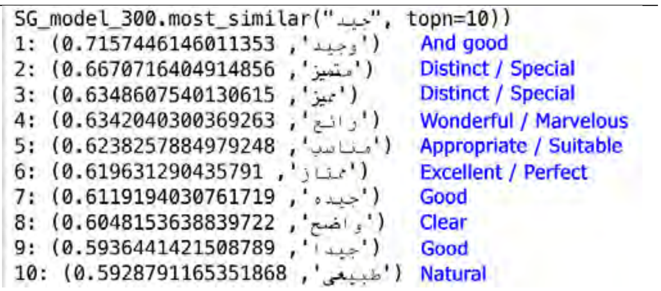
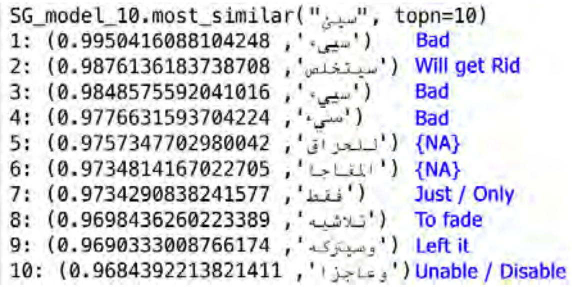
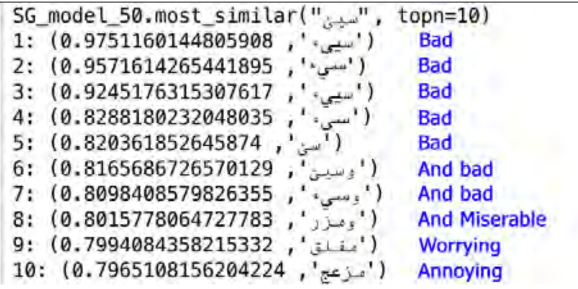
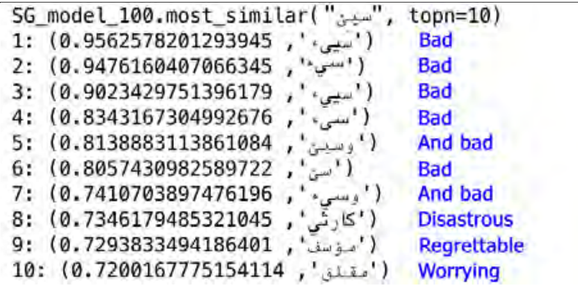
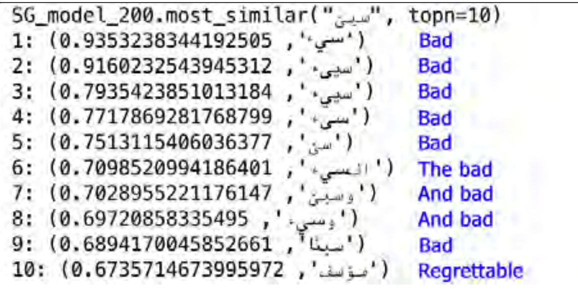
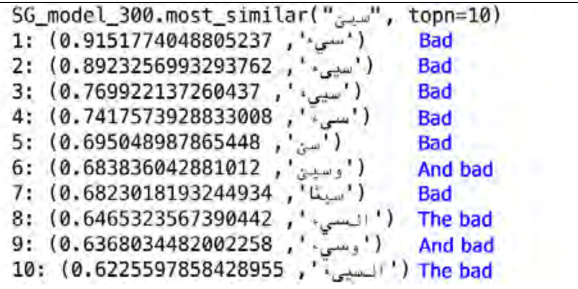
dimensionality	**Good "جد"**
10	 SG_model_10.most_similar("جيد", topn=10)∂) 1: (0.9871622920036316 ,'لمناسب') Appropriate / Suitable 2: (0.9809944033622742 ,'سيتناسب') Will fit 3: (0.9765548706054688 ,'تسي') Will be able 4: (0.9678409695625305 ,'الاسبقيه') Priority / Precedence 5: (0.9674731492996216 ,'التحفيزه') Motivate / Encourage 6: (0.967185378074646 ,'استفادته') Profit / Benefit 7: (0.9635581374168396 ,'لمعاملاته') Dealing / Treatment 8: (0.9617605209350586 ,'للمستوي') To the level 9: (0.9608638882637024 ,'مهيا') Prepared / Ready 10: (0.9596813321113586 ,'تمتعه') Joy
50	 SG_model_50.most_similar("جيد", topn=10)) 1: (0.8267565369606018 ,'طبيعي') Natural 2: (0.8181592226028442 ,'بشكل') In a form 3: (0.805439829826355 ,'احتراف') Professional 4: (0.7810139656066895 ,'وجيد') And good 5: (0.7771871089935303 ,'متميز') Distinct / Special 6: (0.7686337828636169 ,'ايجابي') Positive 7: (0.7644679546356201 ,'متدني') Low 8: (0.7636167407035828 ,'اساسي') Basic / Essential 9: (0.7575806379318237 ,'مثالي') Ideal 10: (0.7490912675857544 ,'واقعي') Realistic
100	 SG_model_100.most_similar("جيد", topn=10)) 1: (0.781434178352356 ,'وجيد') And good 2: (0.7575291395187378 ,'متميز') Distinct / Special 3: (0.7511346936225891 ,'بشكل') In a form 4: (0.7370492219924927 ,'طبيعي') Natural 5: (0.7258888483047485 ,'جيده') Good 6: (0.7172396779060364 ,'مثالي') Ideal 7: (0.7065442800521851 ,'ايجابي') Positive 8: (0.7054450511932373 ,'احتراف') Professional 9: (0.7051820755004883 ,'سيء') Bad 10: (0.7044427990913391 ,'واضح') Clear
200	 SG_model_200.most_similar("جيد", topn=10)) 1: (0.7182195782661438 ,'متميز') Distinct / Special 2: (0.7180838584899902 ,'وجيد') And good 3: (0.661284327507019 ,'جيده') Good 4: (0.654598593711853 ,'رائع') Wonderful / Marvelous 5: (0.646915078163147 ,'مناسب') Appropriate / Suitable 6: (0.6466327905654907 ,'مميز') Distinct / Special 7: (0.645879864692688 ,'واضح') Clear 8: (0.6426805257797241 ,'طبيعي') Natural 9: (0.6356602311134338 ,'سيء') Bad 10: (0.6325259804725647 ,'مثالي') Ideal
300	 SG_model_300.most_similar("جيد", topn=10)) 1: (0.7157446146011353 ,'وجيد') And good 2: (0.6670716404914856 ,'متميز') Distinct / Special 3: (0.6348607540130615 ,'مميز') Distinct / Special 4: (0.6342040300369263 ,'رائع') Wonderful / Marvelous 5: (0.6238257884979248 ,'مناسب') Appropriate / Suitable 6: (0.619631290435791 ,'ممتاز') Excellent / Perfect 7: (0.6119194030761719 ,'جيده') Good 8: (0.6048153638839722 ,'واضح') Clear 9: (0.5936441421508789 ,'جيدا') Good 10: (0.5928791165351868 ,'طبيعي') Natural

Table B.4 The most similar words to the word *bad* in Arabic using the Word2Vec technique *SG* for different dimensionalities: 10, 50, 100, 200, and 300

dimensionality	Bad "سئ"
10	 SG_model_10.most_similar("سيئ", topn=10) 1: (0.9950416088104248 ,'سيء') Bad 2: (0.9876136183738708 ,'سيتخلص') Will get Rid 3: (0.9848575592041016 ,'سيء') Bad 4: (0.9776631593704224 ,'سيء') Bad 5: (0.9757347702980042 ,'للحراق') {NA} 6: (0.9734814167022705 ,'المفاجا') {NA} 7: (0.9734290838241577 ,'فقط') Just / Only 8: (0.9698436260223389 ,'تلاشيه') To fade 9: (0.9690333008766174 ,'وسيتركه') Left it 10: (0.9684392213821411 ,'وعاجزا') Unable / Disable
50	 SG_model_50.most_similar("سيئ", topn=10) 1: (0.9751160144805908 ,'سيء') Bad 2: (0.9571614265441895 ,'سيء') Bad 3: (0.9245176315307617 ,'سيء') Bad 4: (0.8288180232048035 ,'سيء') Bad 5: (0.820361852645874 ,'سئ') Bad 6: (0.8165686726570129 ,'وسيئ') And bad 7: (0.8098408579826355 ,'وسيء') And bad 8: (0.8015778064727783 ,'ومزر') And Miserable 9: (0.7994084358215332 ,'مقلق') Worrying 10: (0.7965108156204224 ,'مزعج') Annoying
100	 SG_model_100.most_similar("سيئ", topn=10) 1: (0.9562578201293945 ,'سيء') Bad 2: (0.9476160407066345 ,'سيء') Bad 3: (0.9023429751396179 ,'سيء') Bad 4: (0.8343167304992676 ,'سيء') Bad 5: (0.8138883113861084 ,'وسيئ') And bad 6: (0.8057430982589722 ,'سئ') Bad 7: (0.7410703897476196 ,'وسيء') And bad 8: (0.7346179485321045 ,'كارثي') Disastrous 9: (0.7293833494186401 ,'مؤسف') Regrettable 10: (0.7200167775154114 ,'مقلق') Worrying
200	 SG_model_200.most_similar("سيئ", topn=10) 1: (0.9353238344192505 ,'سيء') Bad 2: (0.9160232543945312 ,'سيء') Bad 3: (0.7935423851013184 ,'سيء') Bad 4: (0.7717869281768799 ,'سيء') Bad 5: (0.7513115406036377 ,'سئ') Bad 6: (0.7098520994186401 ,'السيء') The bad 7: (0.7028955221176147 ,'وسيئ') And bad 8: (0.69720858335495 ,'وسيء') And bad 9: (0.6894170045852661 ,'سيئا') Bad 10: (0.6735714673995972 ,'مؤسف') Regrettable
300	 SG_model_300.most_similar("سيئ", topn=10) 1: (0.9151774048805237 ,'سيء') Bad 2: (0.8923256993293762 ,'سيء') Bad 3: (0.769922137260437 ,'سيء') Bad 4: (0.7417573928833008 ,'سيء') Bad 5: (0.695048987865448 ,'سئ') Bad 6: (0.683836042881012 ,'وسيئ') And bad 7: (0.6823018193244934 ,'سيئا') Bad 8: (0.6465323567390442 ,'السيء') The bad 9: (0.6368034482002258 ,'وسيء') And bad 10: (0.6225597858428955 ,'السيئ') The bad

Appendix C

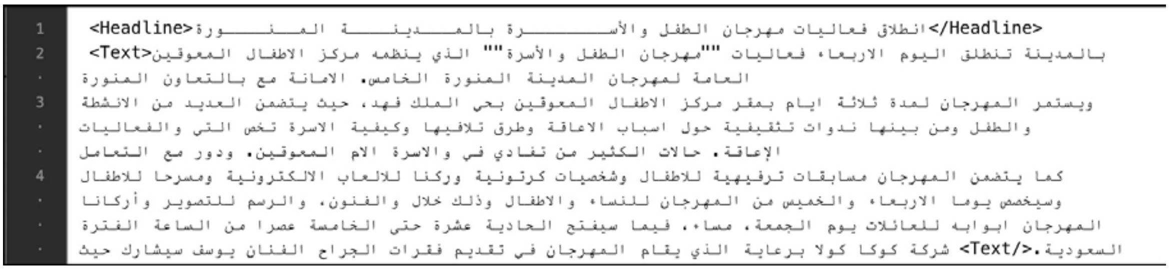
The Details of Filtering the Abu El-khair Corpus and the *Python* Code Used

Due to the large size of the XML files in the Abu El-khair Corpus, we separately did the text pre-processing for each file. The corpus contains ten XML files and the files names are based on the names of the newspapers, which were collected from. The names of the files, as we downloaded them from (Abu El-khair, 2019), are: Alittihad.xml, Almasryalyoum.xml, Almustaqbal, Alqabas.xml, Echoroukonline.xml, Ryiadh.xml, Sabanews.xml, SaudiYoum.xml, Techreen.xml, and Youm7.xml. We converted all the files to TXT format instead of XML. Then, we ran the following code to keep only the titles and the bodies or the text of the articles only. The following code is to filter the SaudiYoum file and if we want to filter another file, we changed the IN_file, OUT_file, and some elements of XMLtags list based on the unwanted XML tags.

```python
__author__ = 'aalayba'
# Filtering *** SaudiYoum.txt ***

# Specifying the input file
IN_file = 'SaudiYoum.txt'
```

```

# Specifying the name of the output file
OUT_file = 'SaudiYoum_Del_Tags.txt'

# Opening the output file to write each line of the text in it
with open(OUT_file, 'w') as outfile:
# Opening the input file to read the text from it
with open(IN_file, 'r') as txt:
# A loop to read each line from the input text
for line in txt:
# Specifying a list of unwanted data and XML tags in the SaudiYoum.txt file
XMLtags = ("<?xml","<SaudiYoumData>","<SaudiYoum>"," <ID>", " <Dateline>"," <URL>","</SaudiYoum>","</SaudiYoumData>")
# If condition to ignore any line that contains any element in the XMLtags list
if all(tag not in line for tag in XMLtags):
# Writing the line that not contains any element in the output file
outfile.write(line)
```

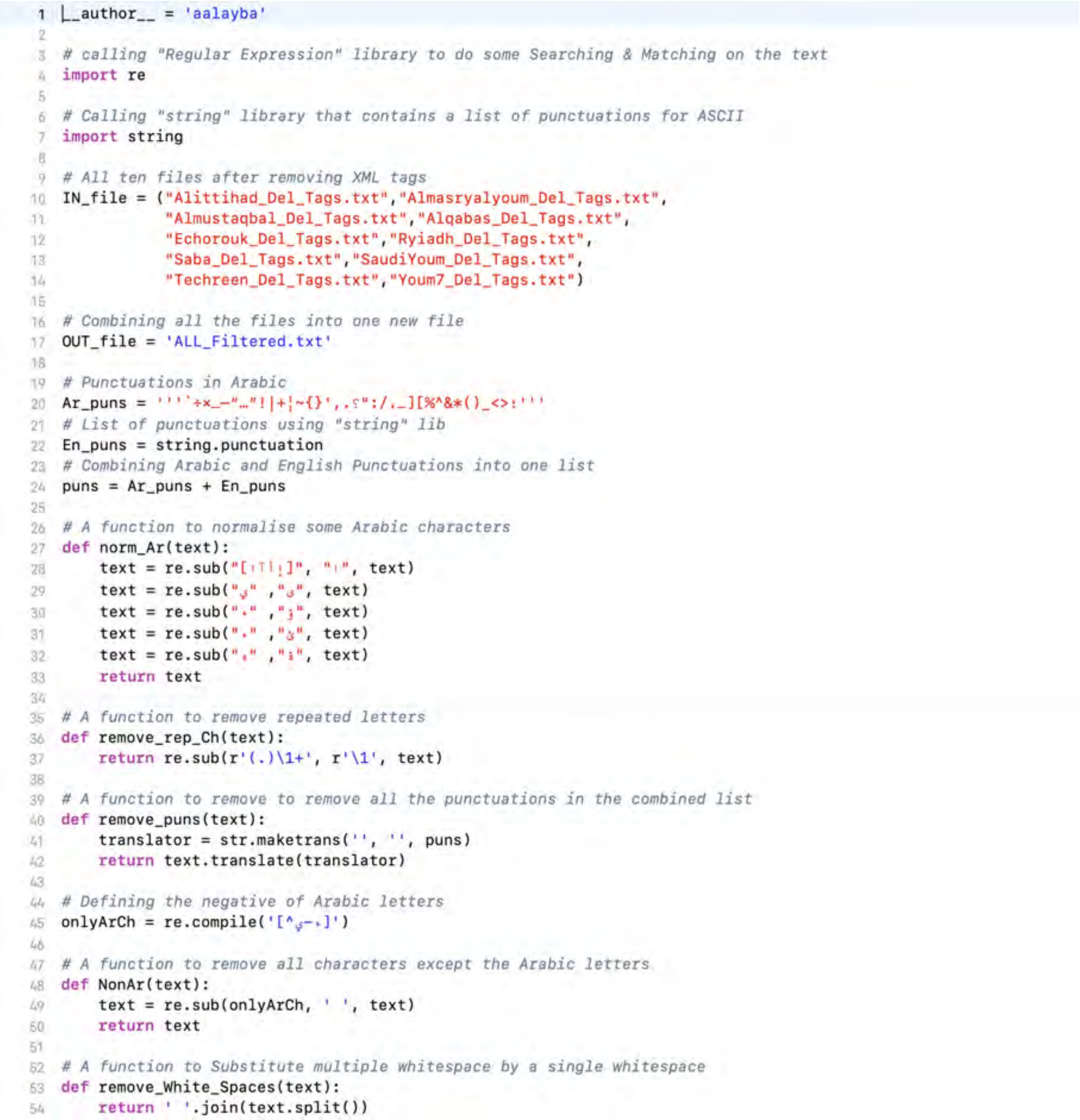
Listing C.1 A Python code to remove unwanted data and XML tags



```
<Headline>انطلاق فعاليات مهرجان الطفل والأسـرة بالمـديـنـة المـنـورة</Headline>
<Text>بالمدينة تنطلق اليوم الاربعاء، فعاليات ""مهرجان الطفل والأسرة"" الذي ينظمه مركز الاطفال المعوقين
العامة لمهرجان المدينة المنورة الخامس. الامانة مع بالتعاون المنورة
ويستمر المهرجان لمدة ثلاثة ايام بمقر مركز الاطفال المعوقين بحي الملك فهد، حيث يتضمن العديد من الانشطة
والطفل ومن بينها ندوات تثقيفية حول اسباب الاعاقة وطرق تلافيها وكيفية الاسرة تخص التي والفعاليات
الإعاقة. حالات الكثير من تفادي في والاسرة الام المعوقين. ودور مع التعامل
كما يتضمن المهرجان مسابقات ترفيهية للاطفال وشخصيات كرتونية وركنا للالعاب الالكترونية ومسرحا للاطفال
وسيخصص يوما الاربعاء، والخميس من المهرجان للنساء، والاطفال وذلك خلال والفنون، والرسم للتصوير وأركانا
المهرجان ابوابه للعائلات يوم الجمعة، مساء، فيما سيفتح الحادية عشرة حتى الخامسة عصرا من الساعة الفترة
السعودية.</Text> شركة كوكا كولا برعاية الذي يقام المهرجان في تقديم فقرات الجراح الفنان يوسف سيشارك حيث
```

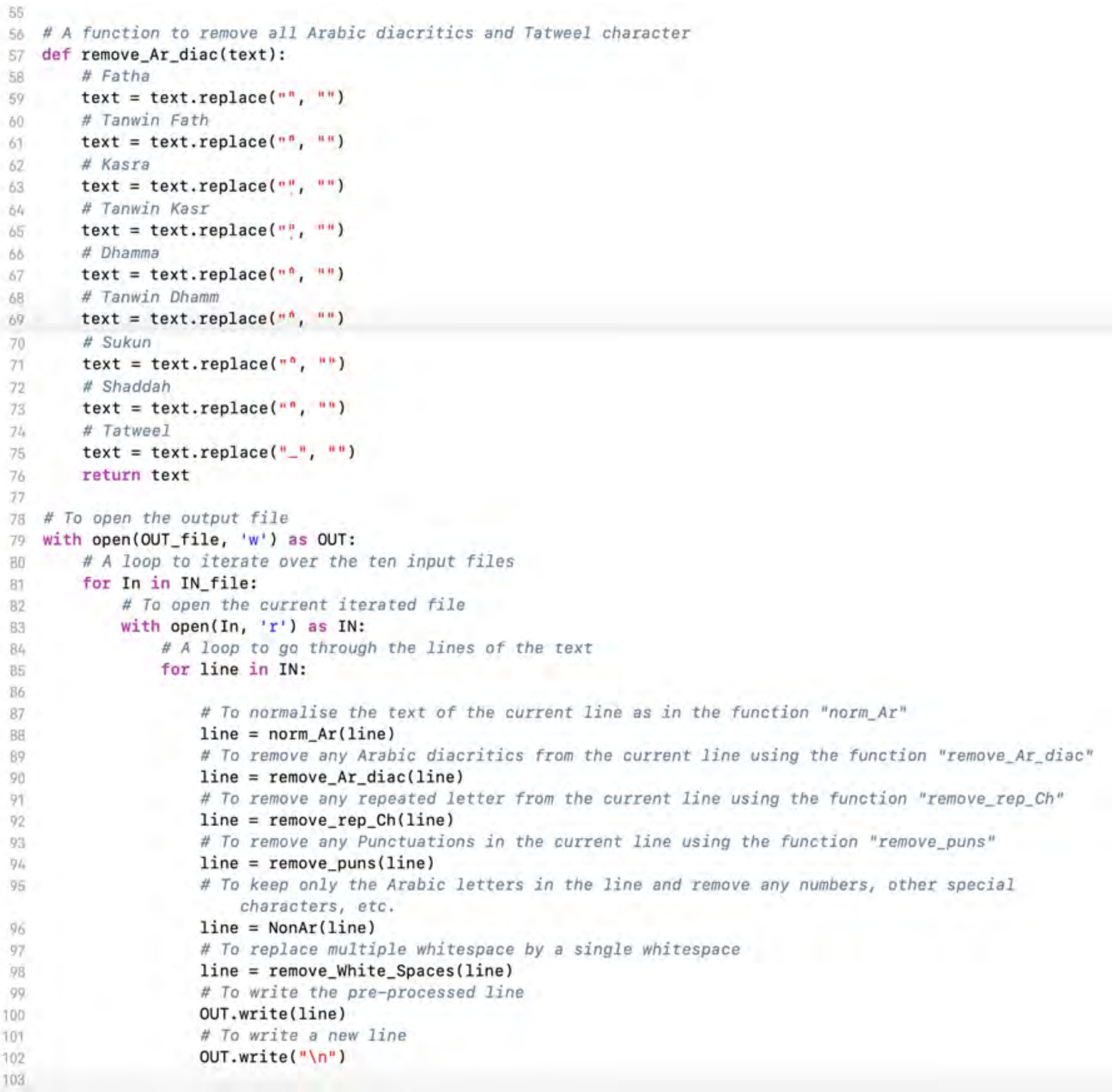
Fig. C.1 An example of one article in one of the XML files after runing the Python code

After running the previous python code, the output files will be similar to Figure C.1. They have four XML tags remaining that are "<Headline>", "</Headline>", "<Text>", and "</Text>". Also, the text itself have unwanted characters, such as, punctuations, numbers, special characters, letters from other languages, normalising some Arabic letters, etc. Therefore, the text still needs further pre-processing. For example, to normalise some Arabic letters, remove Arabic diacritics, remove Tatweel, remove punctuations, remove any non-Arabic letters, and replace multiple whitespaces by single whitespace. We used the Python code which is split into Figure C.2 and Figure C.3, to do the further text pre-processing.



```python
__author__ = 'aalayba'

# calling "Regular Expression" library to do some Searching & Matching on the text
import re

# Calling "string" library that contains a list of punctuations for ASCII
import string

# All ten files after removing XML tags
IN_file = ("Alittihad_Del_Tags.txt","Almasryalyoum_Del_Tags.txt",
           "Almustaqbal_Del_Tags.txt","Alqabas_Del_Tags.txt",
           "Echorouk_Del_Tags.txt","Ryiadh_Del_Tags.txt",
           "Saba_Del_Tags.txt","SaudiYoum_Del_Tags.txt",
           "Techreen_Del_Tags.txt","Youm7_Del_Tags.txt")

# Combining all the files into one new file
OUT_file = 'ALL_Filtered.txt'

# Punctuations in Arabic
Ar_puns = '''`÷×_−"…"!|+¦~{}',.؟":/،_][%^&*()_<>؛'''
# List of punctuations using "string" lib
En_puns = string.punctuation
# Combining Arabic and English Punctuations into one list
puns = Ar_puns + En_puns

# A function to normalise some Arabic characters
def norm_Ar(text):
    text = re.sub("[إأآا]", "ا", text)
    text = re.sub("ي" ,"ى", text)
    text = re.sub("ء" ,"ؤ", text)
    text = re.sub("ء" ,"ئ", text)
    text = re.sub("ه" ,"ة", text)
    return text

# A function to remove repeated letters
def remove_rep_Ch(text):
    return re.sub(r'(.)\1+', r'\1', text)

# A function to remove to remove all the punctuations in the combined list
def remove_puns(text):
    translator = str.maketrans('', '', puns)
    return text.translate(translator)

# Defining the negative of Arabic letters
onlyArCh = re.compile('[^ء-ي]')

# A function to remove all characters except the Arabic letters
def NonAr(text):
    text = re.sub(onlyArCh, ' ', text)
    return text

# A function to Substitute multiple whitespace by a single whitespace
def remove_White_Spaces(text):
    return ' '.join(text.split())
```

Fig. C.2 The First part of the Python code to do the further text pre-processing



```python

# A function to remove all Arabic diacritics and Tatweel character
def remove_Ar_diac(text):
    # Fatha
    text = text.replace("َ", "")
    # Tanwin Fath
    text = text.replace("ً", "")
    # Kasra
    text = text.replace("ِ", "")
    # Tanwin Kasr
    text = text.replace("ٍ", "")
    # Dhamma
    text = text.replace("ُ", "")
    # Tanwin Dhamm
    text = text.replace("ٌ", "")
    # Sukun
    text = text.replace("ْ", "")
    # Shaddah
    text = text.replace("ّ", "")
    # Tatweel
    text = text.replace("ـ", "")
    return text

# To open the output file
with open(OUT_file, 'w') as OUT:
    # A loop to iterate over the ten input files
    for In in IN_file:
        # To open the current iterated file
        with open(In, 'r') as IN:
            # A loop to go through the lines of the text
            for line in IN:

                # To normalise the text of the current line as in the function "norm_Ar"
                line = norm_Ar(line)
                # To remove any Arabic diacritics from the current line using the function "remove_Ar_diac"
                line = remove_Ar_diac(line)
                # To remove any repeated letter from the current line using the function "remove_rep_Ch"
                line = remove_rep_Ch(line)
                # To remove any Punctuations in the current line using the function "remove_puns"
                line = remove_puns(line)
                # To keep only the Arabic letters in the line and remove any numbers, other special
                    characters, etc.
                line = NonAr(line)
                # To replace multiple whitespace by a single whitespace
                line = remove_White_Spaces(line)
                # To write the pre-processed line
                OUT.write(line)
                # To write a new line
                OUT.write("\n")

```

Fig. C.3 The second part of the Python code to do the further text pre-processing