Palestine Polytechnic University
Deanship of Graduate Studies and Scientific Research
Master of informatics

Sentiment Analysis of News Headlines on Middle East in Arabic Media

Submitted by:

Fedaa Hassan Amro

Supervised by:

Dr.Mohammed Aldasht

Thesis submitted in partial fulfillment of requirements of degree "Master of Science in Informatics"
2019

The undersigned hereby certify that they have read, examined and recommended to the Deanship of Graduate Studies and Scientific Research at Palestine Polytechnic University the approval of a thesis entitled: **Sentiment Analysis of News Headlines on Middle East in Arabic Media**, submitted by **Fedaa Hassan Amro** in partial fulfillment of the requirements for the degree of Master in Informatics.

Graduate Advisory Committee:

Dr. Mohammed Aldasht(Supervisor), Palestine Polytechnic University.

Signature:_____ Date:_____

Dr. Diya Abu Zeina (Internal committee member), Palestine Polytechnic University.

Signature:_____ Date:_____

Dr. Radi Jarrar (External committee member), Birzeit University.

Signature:_____ Date:_____

Thesis Approved

Dr. Murad Abu Subaih

Dean of Graduate Studies and Scientific Research

Palestine Polytechnic University

Signature:_____ Date:_____

DECLARATION

I declare that the Master Thesis entitled **"Sentiment Analysis of News Headlines on Middle East in Arabic Media"** is my original work, and herby certify that unless stated, all work contained within this thesis is my own independent research and has not been submitted for the award of any other degree at any institution, except where due acknowledgement is made in the text.

Fedaa Hassan Amro

Signature:_____ Date:_____

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for the master degree in Informatics at Palestine Polytechnic University, I agree that the library shall make it available to borrowers under rules of the library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from, reproduction, or publication of this thesis may be granted by my main supervisor, or in his absence, by the Dean of Graduate Studies and Scientific Research when, in the opinion of either, the proposed use of the material is for scholarly purposes.

Any copying or use of the material of this thesis for financial gain shall not be allowed without my written permission.

Fedaa Hassan Amro

Signature:_____ Date:_____

DEDICATION

To the spirits of deceased Muslims

To my dear mother, Fatima

To my dear father, Hassan

To my sisters and brothers

To my family

To my friends

To my professors

To Palestine Polytechnic University

To the Martyrs, Wounded, Prisoners from the Islamic nations

And finally, To my home land, Palestine

ACKNOWLEDGEMENT

I would like to acknowledge Palestine Polytechnic University for giving us the opportunity to complete the Master Degree in Informatics.
Acknowledgements to Dr. Motaz Saad, a lecturer at the Islamic University in Gaza, who provided us with a wide range of news headlines from different sources. Acknowledgement also goes to all the instructors in the Deanship of Graduate Studies and Research Units for their great impact in our education, especially my distinguished supervisor Dr. Mohammed Aldasht, and every body who lighten my education path.

Furthermore, I will not forget to acknowledge my family and friends.

الملخص

يمكن تحقيق تحليل المشاعر عن طريق النصوص باستخدام المعنى اللغوي أوطرق التعلم الآلي للتعرف على معاني المحتوى واستخراج الآراء الموجودة في النص. مع وجود كمية كبيرة من الأخبار التي يتم إنشاؤها في الوقت الحاضر من خلال مواقع الأخبار المختلفة، فمن الممكن تطبيق تقنيات التنقيب عن النص بغرض استخراج المشاعر العامة لأخبار معينة. ينصب التركيز في هذه الحالة على استخدام تطبيق تحليل المشاعر لاستخراج الأراء من عناوين الأخبار. في هذه الأطروحة، اقترحنا نموذجًا مخصصًا لتقييم المشاعر لقياس مستوى التوتر (نحو السالب أو الموجب) لكل يوم من خلال عناوين أخبار الشرق الأوسط في وسائل الإعلام العربية. يتم جمع البيانات من مواقع الأخبارالعربية مثل الجزيرة، ثم يتم تطبيق خطوات المعالجة لتنقية وتصفية البيانات مثل إزالة الكلمات غير المهمة وعلامات الترقيم من أجل الحصول على مجموعة بيانات خالصة كمدخل لنموذج تعلم التعلم الآلي. في هذه الأطروحة، استخدمنا عناوين الأخبار مع تواريخها التي تم جمعها منذ عدة سنوات من خمسة مواقع مختلفة. أيضًا، ابتكرنا طريقة لجمع عناوين الأخبار مع تواريخها وفئتها ووصفها تلقائيًا من مواقع الأخبار لاستخدامها في الأعمال المستقبلية. ثم تمت معالجة البيانات ومراجعتها بواسطة العديد من الأدوات المهمة في بايثون بالاضافة على ذلك ، استخدمنا خدمة الحوسبة السحابية في جوجل بطريقة مبتكرة لترجمة العناوين تلقائيًا. بعد ذلك، ابتكرنا طريقة لعنونة العناوين لإعطاء درجة لكل منها سواء نحو السالب أم الموجب، يتم استخدام صيغة عشرية كمقياس للجودة (درجة التوتر) لكل عنوان بواسطة معاجم الورد نت والسينتي ورد نت. لقد قمنا بتدريب نموذج الانحدار الخطي المتعدد استنادًا إلى عنصرين هامين لكل يوم هما مجموع الدرجات الإيجابية ومجموع الدرجات السلبية في ذلك اليوم، بحيث يقيس النموذج درجة التوتر في ذلك اليوم المحدد. لقد قمنا باختبار النموذج من خلال مقاييس مهمة ، وحصلنا على درجة تباين تساوي أربعة وتسعون بالمائة ونسبة خطأ تساوي أربعة بالمائة من خلال بيانات تدريب لمدة عام كامل. أخيرًا ، قمنا بتوصيل النموذج بقاعدة بيانات وخادم لتنفيذ النموذج على أرض الواقع.

Abstract

Sentiment Analysis can be achieved using lexicons and machine learning methods to identify the sentiment of a content and opinion mining of a text. With the large amount of news being generated nowadays through various news websites, it is possible to apply text mining techniques with the purpose of extracting general sentiment of particular news. The emphasis in this case is on using a Sentiment Analysis application to extract sentiment from news headlines. In this thesis, we have proposed a customized model for sentiment evaluation to measure tensions level (using negative and positive scores) for every day on Middle East news headlines in the Arabic media. The data are collected from Arabic media websites like Aljazeera, then the required pre-processing steps are applied. Steps such as stop words and punctuation marks removal, in order to get a pure dataset as an input for the regression learning model. In this thesis, we have used the news headlines with their dates collected over many years ago from five different sites. Also, we have devised a method for collecting the news headlines automatically with their dates, category and description from RSS feed for news websites to use them for future works. In addition, the data were processed and revised by several important tools in Python. Moreover, We have used Google Cloud Translation API in an innovative way to translate headlines automatically. Then, we devised a method for headline labeling to give a score for each one, the Decibel formula is used as a quality measure (sentiment score) for every headline, based on two main lixicons, namely WordNet and SentiWordNet. We have trained a multiple linear regression model based on two important entries for every day, the sum of positive scores and the sum of negative scores on that day, so that the model will measure the sentiment score for that particular day. We have tested the model with important measures, we have obtained 0.937 Explained Variance Score, 0.94 R^2 Score and 0.04 MSE through a full year training data. Finally, we have connected the model with Database and Flask server to achieve real time measurement.

Table of Contents

1 Introduction **2**

1.1 Sentiment Analysis Importance and Definition 3

1.2 Motivation of Thesis 3

1.3 Problem Statement 4

1.4 Objectives 4

1.5 Scope and limitations of Thesis 4

1.6 Thesis Structure 5

2 Background and Literature Review **6**

2.1 Natural Language Processing 7

2.1.1 Sentiment Analysis 7

2.1.2 Natural Language Toolkit(NLTK) 8

2.2 Machine Learning 9

2.3 Literature Review 12

3 Data and Methods **20**

3.1 Development Environment 21

3.2 Thesis Methodology 22

3.3 Data Collection 24

3.4 Data pre-processing 24

3.4.1 News headline Translation 25

3.4.2 Stop Words Removal 27

3.4.3 Punctuation Removal 28

3.4.4 WordNet Lexicon 29

3.4.5 SentiWordNet Lexicon 29

3.4.6 Sentiment Score calculation 29
3.5 Regression Model 31
3.6 Implementation 36

4 Experiments and Results **41**
4.1 Experiments 42
4.1.1 Experiments Environment 42
4.1.2 Experiment A: Training Sentiment Model using Arabic headlines . . . 42
4.1.3 Experiment B: Training Sentiment Model using translated arabic headlines (English headlines) 44
4.1.4 Experiment C: Training Sentiment Model using translated arabic headlines (English headlines) grouped per day(Daily headlines) . . . 45
4.2 Discussions and Results 47
4.2.1 Results 47
4.2.2 Discussion 74
4.3 Real Time Analysis Experiment 75

5 Conclusions and Future Works **80**
5.1 Conclusions 81
5.2 Future works 82

List of Figures

2.1 Machine Learning process 10

3.1 Workflow of Thesis 23
3.2 Workflow of Data Pre-processing 25
3.3 Stop Words in English 28
3.4 Stop Words in Arabic 28
3.5 Punctuation Symbols 28
3.6 Multiple Linear Regression equation 33
3.7 ML Software Architecture 40

4.1 The difference between Y_Test and Y_Predict for the experiment A 48
4.2 Relationship between SumPS, SumNS and Sentiment Score in the experiment A 49
4.3 Sentiment Score Frequency in the experiment A 50
4.4 The difference between Y_Test and Y_Predict for the experiment B(5000 headlines) 51
4.5 Relationship between SumPS, SumNS and Sentiment Score in the experiment B(5000 headlines) 52
4.6 Sentiment Score Frequency in the experiment B(5000 headlines) 53
4.7 The difference between Y_Test and Y_Predict for the experiment B(15000 headlines) 54
4.8 Relationship between SumPS, SumNS and Sentiment Score in the experiment B(15000 headlines) 55
4.9 Sentiment Score Frequency in the experiment B(15000 headlines) 56
4.10 The difference between Y_Test and Y_Predict for the experiment B(30000 headlines) 57

4.11 Relationship between SumPS, SumNS and Sentiment Score in the experiment B(30000 headlines) 58
4.12 Sentiment Score Frequency in the experiment B(30000 headlines) 59
4.13 The difference between Y_Test and Y_Predict for the experiment B(80000 headlines) 60
4.14 Relationship between SumPS, SumNS and Sentiment Score in the experiment B(80000 headlines) 61
4.15 Sentiment Score Frequency in the experiment B(80000 headlines) 62
4.16 The difference between Y_Test and Y_Predict for the experiment C(3 month) 63
4.17 Relationship between SumPS, SumNS and Sentiment Score in the experiment C(3 month) 64
4.18 Sentiment Score Frequency in the experiment C(3 month) 65
4.19 The difference between Y_Test and Y_Predict for the experiment C(6 month) 66
4.20 Relationship between SumPS, SumNS and Sentiment Score in the experiment C(6 month) 67
4.21 Sentiment Score Frequency in the experiment C(6 month) 68
4.22 The difference between Y_Test and Y_Predict for the experiment C(9 month) 69
4.23 Relationship between SumPS, SumNS and Sentiment Score in the experiment C(9 month) 70
4.24 Sentiment Score Frequency in the experiment C(9 month) 71
4.25 The difference between Y_Test and Y_Predict for the experiment C(12 month) 72
4.26 Relationship between SumPS, SumNS and Sentiment Score in the experiment C(12 month) 73
4.27 Sentiment Score Frequency in the experiment C(12 month) 74
4.28 SQLite3 Database Screenshot of Distribution Data 76
4.29 Predicted News Headlines per day in year 2014 77
4.30 Predicted News Headlines per Week in year 2014 77
4.31 Predicted News Headlines per month in year 2014 78
4.32 Predicted News Headlines per day in year 2015 78
4.33 Predicted News Headlines per week in year 2015 79
4.34 Predicted News Headlines per month in year 2015 79

List of Tables

2.1 Arabic Domain Literature Review 15

2.2 Pros and Cons of Literature Review 17

4.1 Arabic headlines Dataset 43

4.2 Dataset After All pre-processing Phase 43

4.3 Arabic Headlines Dataset 44

4.4 Translated Arabic Headlines 44

4.5 Dataset After All Pre-processing Phase 45

4.6 Training headline per Day Dataset 45

4.7 Arabic headlines connected period Dataset 46

4.8 Translated Arabic Headlines per Day 46

4.9 Dataset with Sum of Positive Score and Sum of Negative Score For each headline per day 46

4.10 Dataset After All Pre-processing Phase 47

4.11 Experiment A metrics results 47

4.12 Experiment B metrics results 50

4.13 Experiment C metrics results 62

List of Algorithms

1 Google Cloud Translate API Call 26

2 headline Translations 27

3 Prepare headline Scores Training 30

4 Training and Testing 35

5 Filter the dataset per time period and group the dataset per daily functions 36

6 Reading any news DataSource File 36

7 Inserting the Predicted headlines Inside DB 37

8 Prediction News headlines On The Fly 38

9 Real Time 40

10 Arabic Version of SentiWordNet 43

List of Abbreviations

PosScore	Positive Score
NegScore	Negative Score
OM	Opinion Mining
NLP	Natural Language Processing
SA	Sentiment Analysis
ML	Machine Learning
AI	Artificial Intelligence
CL	Computational Linguistics
NLTK	Natural Language Toolkit
MLR	Multiple Linear Regression
MSE	Mean Squared Error
GCP	Google Cloud Platform
RSS	Rich Site Summary
SumPS	Sum Of Positive Score
ME	Maximum entropy
KNN	k-Nearest Neighbor
NB	Naïve Bayes
SVM	Support Vector Machine
DT	Decision Tree
LF	Lexical Features
CRF	Conditional Random Field

Chapter 1

Introduction

1. Chapter Outline:

    1.1. Sentiment Analysis Importance and Definition

    1.2. Motivation of Thesis

    1.3. Problem Statement

    1.4. Objectives

    1.5. Scope and limitations of Thesis

    1.6. Thesis Structure

1.1 Sentiment Analysis Importance and Definition

The sizeable use of social networks, formal websites, news websites, boards and personal blogs enabled tens of millions of human beings to post and proportion their feedback or reviews on the internet. these reviews can cover several topics including merchandise, movies and others. This truth endorsed many companies, governments, customers and other events to make analysis of these evaluations.

"Sentiment" is associated with emotions, attitudes, feelings and critiques which are not facts but subjective impressions. Sentiment Analysis is sometimes referred to as Opinion Mining additionally pursuits to discover, extract or represent the sentiment of a given textual content via the use of Natural Language Processing (NLP), statistics or Machine Learning (ML) techniques. Sentiment Analysis (SA) is an effort of portraying the polarity of opinions, feelings or emotions. Notwithstanding, supposition examination will in general automatically derive a person's attitude towards a subjects [49]

The expression of Sentiment Analysis and Opinion Mining (OM) are equivalents [46]. Preparing the massive number of reviews, headlines, comments, articles, and statements is a challenge that confronted analysts in the fields of Text Mining and information recovery. This approach mission is considered within the Sentiment Analysis or Opinion Mining field; at the same time, it is a sub-undertaking of Text Mining[53].

1.2 Motivation of Thesis

Physically gathering individuals' sentiments through tremendous measures of reviews is tedious and it could be incomprehensible, particularly with the fast development of different fields. Hence, the successful answer to this issue is Sentiment Analysis.

There are various applications that use Sentiment Analysis in several areas, such as review-related websites,recommendation systems, flame detection, question answering systems, summarizing and citation analysis, business and government intelligence, politics, sociology etc [49]. In the midst of the widespread increase of these applications areas and domains, our motivation is to use Sentiment Analysis in news domain.

Our motivation for applying Sentiment Analysis to news headline data from formal websites is the lack of previous work conducted in this area. Most of the studies use reviews (movie reviews, product reviews, Twitter reviews etc.), since the writers often summarize their sentiments in their reviews. Besides, it becomes easy to annotate training data as positive or negative since most of the websites accept user reviews also provide a rating system, i.e. a 0-5 scale star system. Moreover, reviews are short and relatively easy to analyze.

1.3 Problem Statement

A customized model for sentiment evaluation to measure tensions level (using negative and positive scores) for every day on Middle East news headlines in the Arabic media.

1.4 Objectives

Our main objective is to build we have proposed a customized model for sentiment evaluation to measure tensions level (using negative and positive scores) for every day on Middle East news headlines in the Arabic media. The data will be collected from the Arabic headlines of the news websites, then the required pre-processing will be performed in order to get the pure data ready as an input for the learning model, and training data will be labelled. finally, the model will be trained to determine the tensions level as a score in the area.

1.5 Scope and limitations of Thesis

This study will focus on predicting the tension level in the Middle East through applying Sentiment Analysis on the Arabic news headlines. And In order to achieve that, this study uses lexicon-based approach to calculate the positive and negative scores for each headline beside using linear regression ML model approach to predict the sentiment score for the headline or what we call it the tension level. This study use over than 250K of Arabic news headlines that have been collected from different resource between (2000 - 2019) to

be used to train and test the ML model. The rest is used to find the tension level using the prediction. Also, developing a simple website to display the tension level that has been found in the predicted dataset using an elegant meaningful charts. In addition, in this study Arabic news headlines are translated into English. The English translation is used by the tool to achieve the study goal, because of the lack of finding good and mature Arabic lexicon resource that contains sufficient scores to rely on in this study.

1.6 Thesis Structure

This thesis' five chapters are organized as follows:

- Chapter 1
  - Chapter one introduces the study problem of the thesis. The chapter provides an introduction to the problem, purpose, deliverable, thesis contributions , objectives, scope and the thesis limitation.
- Chapter 2
  - Chapter two presents the basic concepts within Sentiment Analysis and Machine Learning. It covers the background of Sentiment Analysis and Machine Learning. In addition, it sheds light on the related technology used in this thesis. also, views the related literature that contributes to the study.
- Chapter 3
  - Chapter Three introduces the research data and methodology used in this study.
- Chapter 4
  - Chapter Four shows the Experiments and Results of this study.
- Chapter 5
  - Chapter Five is a discussion of conclusions and Future Works

Chapter 2

Background and Literature Review

2. Chapter Outline:

    2.1. Natural Language Processing

        2.1.1. Sentiment Analysis

            2.1.1.1. Machine Learning approach

            2.1.1.2. Lexicon-Based approach

            2.1.1.3. Hybrid approach

        2.1.2. Natural Language Toolkit(NLTK)

    2.2. Machine Learning

    2.3. Literature Review

2.1 Natural Language Processing

Natural Language Processing (NLP) [22] is a field in Computer Science that was improved from Artificial Intelligence (AI) and Computational Linguistics (CL). It includes the comprehension of human languages and therefore empowering computers to infer. This which means from human or natural language inter just as to create natural language. Also, to translate between various languages.

Natural Language Processing is an innovation that manages human language showing up in numerous sources like, site pages, web-based life, messages, paper articles for a large number of languages and their combinations.

There is a wide scope of NLP applications such as automatic question answering, email spam detection, grammar correction, machine translation and spelling. Sentiment Analysis means to extract opinion from a given book is a use of NLP, as well.

Statistical NLP, dealing with Machine Learning and data mining techniques, by using statistical, probabilistic and stochastic methods makes thousands or millions of believable analyses of the texts easier.

2.1.1 Sentiment Analysis

Sentiment Analysis has many other names such as Opinion Extraction, Opinion Mining, Sentiment Mining and Subjectivity Analysis. Sentiment Analysis is the automated extraction of attitudes, values, and emotions from text, speech, and database sources through Natural Language Processing (NLP). Sentiment Analysis involves the grouping of views in text into categories such as “positive”, “negative” or “neutral” [49]. For Sentiment Analysis, there are two primary approaches: Machine-learning based and lexicon-based. In addition, few research studies have combined these two approaches and achieved relatively better results, the hybrid approach is the combined approach from the two approaches.

2.1.1.1 Machine Learning approach

Mostly, this approach deals with Sentiment Analysis of supervised classifications. A Machine Learning approach [57] needs two set of documents: the first one is training set that is used by automatic classifier to learn the differentiating characteristics of documents, the second one is a test set that is used to check how the classifier performs. In Sentiment Analysis, Machine Learning approach have achieved very good success like Naïve Bayes, Maximum Entropy and Support Vector Machines. The first step in Machine Learning-based approach is collecting training dataset then training a classifier on the training data. An important decision in this approach is feature selection that can tell us how documents are represented.

2.1.1.2 Lexicon based approach

Lexicon-based approach [57] compares the features of a given text against sentiment lexicons to make classifications. Sentiment lexicon can express peoples feelings and opinions by using lists of words and expressions. The lexicon based approach to Sentiment Analysis does not require prior training in order to classify the data.

2.1.1.3 Hybrid approach

Hybrid approach combines both the Machine Learning approach and the lexicon based approach [57] to improve the sentiment classification performance. The main advantage of using the combination of lexicon and learning is attaining then best of both worlds, readability from a carefully designed lexicon and high accuracy from a powerful supervised learning algorithm.

2.1.2 Natural Language Toolkit(NLTK)

Natural Language Toolkit (NLTK) [9] is a platform for building Python programs to figure with human language data. It provides easy-to-use interfaces to over fifty corpora and lexical resources like WordNet, in conjunction with a set of text process libraries for classification, tokenization, stemming, tagging, parsing, and linguistics reasoning, etc. This toolbox acts

as a key role in transforming the text data into a format that can be used to extract sentiment from them. NLTK provides various functions which are used in pre-processing of data so that data available. NLTK supports various Machine Learning algorithms which are used for training classifiers and to calculate the accuracy of different classifiers.

In our thesis, we use Python as our base programming language. NLTK is a library of Python which acts a very important role in converting natural language text to a sentiment. So, NLTK is used for processing news headline, covering the Sentiment Tokenizer, Word Tokenizer, Stop words removal, WordNet lexicon, and SentiWordNet lexicon.

WordNet [47] is a large lexical database. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are inter linked by means of conceptual-semantic and lexical relations [14]. SentiWordNet is a lexical resource for opinion mining that assigns to each synset of WordNet three sentiment scores: positivity, negativity, and objectivity [13]. It has a Web-based graphical user interface, and it is freely available for research purposes. The development of the resource is based on the quantitative analysis of the glosses associated to synsets, and on the use of the resulting vectoral term representations for semi-supervised synset classification. Positivity, negativity, and objectivity are derived by combining the results produced by a committee of eight ternary classifiers [29].

2.2 Machine Learning

Machine Learning is one of the applications for provides systems and machines the ability to learn and make decisions without being explicitly programmed. Machine Learning (ML) focuses on computer programs and tools that can access data and information in order to use this data to train computer machines [55].

Figure 2.1 shows flow diagram of ML process. The process begins with data or observations like instructions, some characters, sounds or may be direct experience. This data will be used to extract patterns after training and testing some learning models. A good model outcome means good examples and data provided to training model.

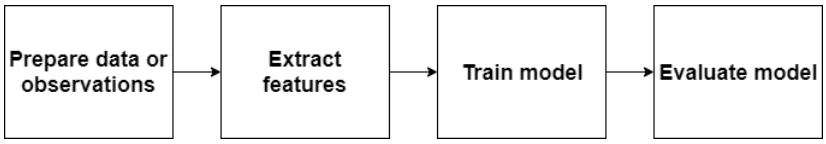

Figure 2.1: Machine Learning process

Machine Learning could be categorized to supervised and unsupervised algorithms, and semi-supervised Machine Learning. Each type is explained as follow:

- Supervised Machine Learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compares its output with the correct, intended output and find errors in order to modify the model accordingly.

- Unsupervised Machine Learning algorithms, are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

- Semi-supervised Machine Learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training, typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring labeled data generally doesn't require additional resources.

- Reinforcement Machine Learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement

learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance.

Machine Learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining Machine Learning with AI and cognitive technologies can make it even more effective in processing large volumes of information [54].

Regression is a statistical process falls under supervised learning, it estimates the relation between two variables. Dependent variable which is called outcome, and independent variable called predictor or feature. The most known types of regression are:

- Linear regression.
- Nonlinear regression.

In linear regression [19], the line or some complex linear function will closely fit the data points. This type allows researchers and scientific estimates expectations for dependent value when they give some value to independent variable. Regression analysis is used for two main purposes:

- Its widely used for prediction and forecasting as a field of Machine Learning.
- It can be used to infer causal relationships between independent and dependent variables.

In nonlinear regression, the relation between the independent variables is not linear. Parameters appears as function and algorithms used to find the solution of nonlinear problems requires initial values for parameters. Analytical expressions for the partial derivatives can be complicated. If analytical expressions are impossible to obtain either the partial derivatives must be calculated by numerical approximation or an estimate must be made of the "Jacobian", often via finite differences. Also, nonlinear least squares are non-convergence and solving them is usually iterative process that terminates when convergence is satisfies. There may be multiple minima in the sum of squares.

2.3 Literature Review

The importance of Sentiment Analysis have been explained above in the introduction section. So, in this section of the dissertation, the past research work done in the area of Sentiment Analysis by different researchers or authors have been viewed. Hu and Liu [37] have summarised the list of positive and negative words after analysing the customer feedback. They have summarised 2006 positive words and 4783 negative words. Sentiment Analysis on the movie reviews data have been done by Lina Zhou et al [20]in 2005 using supervised Machine Learning technique . Due to text classification, the supervised learning approach was used. In 2008 Ahmed Abbas et al [15] have used Sentiment Analysis method for opinion mining of multi lingual web forum. Many other application-oriented research have been done and published too other than academic research in this area. In 2007 Liu et al [44] made a model after Sentiment Analysis to predict the sales performance . Machine Learning technique was used for the Sentiment Analysis by Bo Pang et al [50] to observe the classification efficiency by analysing overall sentiments. It is shown in a work done by Jeonghee Yi et al [58] that using Machine Learning techniques that Machine Learning experiments are giving efficient results on reviews data. Xiaowen Ding et al [23] have proposed a work using natural language expressions to find the semantic orientation of opinions. The product review dataset was used for this experiment and the algorithm was using linguistic pattern to solve special words or phrases and finally this experiment results in efficient result.

In one of the work product review, data (For example product review data from amazon.com) have been used by Anindya et al [32] for the ranking of different products based on the customer reviews which helps customers in better decision making and based upon the customer orientation . Econometric analysis and text mining methods were used for this experiment. Machine Learning and clustering technique-based sentiment classifier was proposed by Michael et al [31]. It was a prototype model which was used for mining topics based on different sentiments that was taken from customer review data. Miniqing Hu et al [36] used three different process on the customer review data for mining and

summarization process.

- Natural language process and data mining
- Opinion mining
- WordNet

Data mining and NLP were used for mining the reviews and converting the data into structured or semi structured format. The opinion mining was used for classification of review based on the sentiments like positive and negative etc. and finally WordNet was used to identify the semantic orientation. The results was summarized after all the three processes. The idea behind this work was to show summary based on features for the big amount of products that are being sold online.

Sentiment Analysis in Twitter: During the past few years the communications on the social network websites like Twitter in the form of tweets (short text) have emerged and ubiquitous. Sentiments and opinions about any product/ movies or things of surrounding worlds are widely expressed through the tweets or messages [33].

Suchdev et al [56] present Twitter Sentiment Analysis using Machine Learning and knowledge-based approach. Twitter Sentiment Analysis is a difficult task due to the presence of slang's and misspellings. Moreover, Twitter restricts the length of a tweet to 140 characters. So, extracting valuable information from short texts is another challenge [56] [18] . In this paper [56], authors analyze peoples sentiments in their tweets about certain companies. Computing a basic sentiment score and then classifying the tweets as positive, negative or neutral will help serve . Feature extraction is done in two phases: First phase : the extraction and acquiring of the dataset of data from Twitter. Then, Pre-processing tweets. Second phase: feature extraction is performed again on this text to get more feature vector.

Llombart et al [52] use different Machine Learning techniques that have different ways to work. Then, they compare the performance of this methods and different types of data. In this paper, authors mention the effect of applying transformations on the data to improve the performance of the classication methods, but the type of transformations

depends on the dataset and the type of the language it has. A special mention to the Recurrent Neural gives better results on all data that can be seen on this document. Chopra et al [21], The objective of this work is to provide a platform for serving good news and create a positive environment. For the analysis of news headlines, dataset gathered news articles from RSS feed. For this study a sample of 103 news, dating from January 1, 2015 to February 28, 2015, was obtained from online Indian newspaper namely The Times of India. The collected text is noisy. So, strings are converted into words using some filtration techniques and pre-processing method to transform or tokenize the text stream to words. After that , Sentiment Analysis stage, this stage handles the polarity measurement and sentiment by employing Machine Learning methods. Finally, Machine Learning-based sentiment classification is applied. In this paper they have discussed about the polarity of news articles in terms of positivity, negativity and neutrality using Naïve Bayes algorithm. Loureiro et al [45], This project presents a method to assign valence ratings to entities, using information from their Wikipedia page, and considering user preferences gathered from the users Facebook profile. Furthermore, a new affective lexicon is compiled entirely from existing corpora, without any intervention from the coders. The two main novelties introduced in this paper are the following: (1) the automatic generation of an objective affective lexicon, and (2) a user centric emotion evaluation technique that considers personal preferences towards entities.

Kalyani et al [39] Assume that news articles have impact on stock market, so they attempt to study relationship between news and stock trend. They created three different classification models which depict polarity of news articles being positive or negative. Observations show that Random Forest and Support Vector Machine perform well in all types of testing. Naïve Bayes gives good result to some extent two.

The following table 2.1 shows some previous work in the field of Arabic language and the difference between them in terms of the mechanism of work, scope of work and results.

Table 2.1: Arabic Domain Literature Review

Auther	Classifier	Feature Selection	Datatype	Accuracy
El-Halees 2011 [26]	ME, KNN, NB, and SVM	Lexicon	Multi-domains	80
Elarnaoty et al. 2012 [27]	Semi-supervised, CRF (Conditional Random Field)	English paper	News	85.52
Abbasi et al. 2008 [15]	SVM	Stylistic Lexicon	Movie Reviews, Web Forums	93
Almas & Ahmad 2007 [17]	-	Domain specific Lexical features	Financial News	-
Elhawary & Elfeky 2010 [28]	-	Lexicon feature	Business Reviews	-
Farra et al. 2010 [30]	SVM	Syntactic, LF (Lexical Feature)	News	87

Table 2.1 continued from previous page

Abdul-Mageed & Diab 2012 [16]	**-**	**LF, Syntactic, Genre Specific Social Media Features**	**News Social Media**	**-**
Rafea & Mostafa 2013 [51]	**Bisecting k-mean Clustering**	**-**	**Tweets**	**72.5**
Wahsheh et al. 2013 [41]	**SVM**	**-**	**Collection of Dataset**	**-**
Hamouda & El-taher 2013 [34]	**SVM NB DT**	**Manual Features**	**Facebook News**	**73.4**
Khoufi & Boudokhane 2013 [42]	**Statistical Approach**	**-**	**-**	**89.01**
El-Beltagy & Ali 2013 [25]	**Lexicon**		**Twitter**	**83.8**
Khalifa & Omar 2014 [40]	**Semi-Supervised, CRF**	**-**	**Arabic News Text**	**54.03**
Duwairi et al. 2014 [24]	**NB SVM KNN**	**-**	**Tweets**	**76.78 71.68 59.99**
Oraby et al. 2013 [48]	**Kouja POS**	**Root-Based Method**	**Movie**	**93.2**

The following table 2.2 shows Pros and cons of previous work that are mentioned in the

previous table.

Table 2.2: Pros and Cons of Literature Review

Auther	Advantages	Disadvantages
El-Halees 2011 [26]	**-Combines lexical and ML** **-Multi-domain**	**-No Arabic-specific features**
Elarnaoty et al. 2012 [27]	**-Publically released An annotated Arabic corpus for opinion holder and an Arabic subjectivity lexicon resource for Arabic NLP researchers** **-Proposed semantic field and named entities Features**	**-Using feature from English paper**
Abbasi et al. 2008 [15]	**- Language independence** **- Effective feature selection**	**- High computational cost**
Almas & Ahmad 2007 [17]	**- Simple method** **- Language independence**	**- No sentiment classification (only phrase extraction)**

Table 2.2 continued from previous page

Elhawary & Elfeky 2010 [28]	**-Builds large-scale lexicon -Demonstrated a system for mining Arabic business reviews from the web**	**- No Arabic-specific features**
Farra et al. 2010 [30]	**-Combines LF and syntactic**	**-Evaluated on small dataset**
Abdul-Mageed & Diab 2012 [16]	**- Combines language-independent and Arabic-specific features - Incorporates dialectal Arabic - Employs a wide-coverage polarity lexicon**	**Some genre and social media features are costly to acquire**
Rafea & Mostafa 2013 [51]	**-Ability to identify the best feature.**	**-Small dataset**
Wahsheh et al. 2013 [41]	**- Build a SPAR system for spam detection based on some criteria**	**- Limited criteria**
Hamouda & El-taher 2013 [34]	**-Analysis the polarity**	**-Not evaluated with previous methods.**

Table 2.2 continued from previous page

Khoufi & Boudokhane 2013 [42]	**- Proposed method for the Morphological annotation of Arabic (Arabic Morphological Annotation System)**	
El-Beltagy & Ali 2013 [25]	**-An Egyptian Dialect sentiment lexicon**	**-There is no classifier**
Khalifa & Omar 2014 [40]	**-Release research corpus and lexicon for opinion mining community to encourage further research**	**-Still, the possibility to enhance the Arabic opinion holder extraction task performance while utilizing a robust Arabic lexical or dependency parser constituents.**
Duwairi et al. 2014 [24]	**-Novel aspects such as handling Arabic dialects, Arabizi and Emoticons. Also, crowdsourcing was utilized to collect large Dataset of tweets**	**- Need to expand the dataset and dictionary**
Oraby et al. 2013 [48]	**-Enhance the root for Sentiment analysis task**	

Chapter 3

Data and Methods

3. Chapter Outline:

    3.1. Development Environment

    3.2. Thesis Methodology

    3.3. Data Collection

    3.4. Data pre-processing

        3.4.1. News headline Translation

        3.4.2. Stop Words Removal

        3.4.3. Punctuation Removal

        3.4.4. WordNet Lexicon

        3.4.5. SentiWordNet Lexicon

        3.4.6. Sentiment Score calculation

    3.5. Regression Model

    3.6. Implementation

This thesis presents a customized model for sentiment evaluation to measure tensions level (using negative and positive scores) for every day on Middle East news headlines in the Arabic media. The data have collected from the Arabic headlines of the news websites, then the required pre-processing and labelled have performed in order to get the pure data ready as an input for the learning model. This chapter explains the steps of thesis, supported by figures and algorithms. For more, you can see the appendix I where there are some important codes.

3.1 Development Environment

In our thesis, we are using Anaconda[1] [2] which is a completely free Python distribution (including for commercial use and redistribution). It includes more than 300 of the most popular Python packages for science, math, engineering, and data analysis. It is available across platforms and installable through a binary. Anaconda is one of the many open source platforms that facilitate the use of open source programming languages (R, Python) for largescale data processing, predictive analytics, and scientific computing[38].

We are installing Jupyter Notebook [2] [10] inside Anaconda Environment. Jupyter Notebook is a non-profit, open-source project, born out of the IPython Project in 2014 as it evolved to support interactive data science and scientific computing across all programming languages. Jupyter will always be 100% open-source software, free for all to use and released under the liberal terms of the modified BSD license [43]. Jupyter notebooks, a document format for publishing code, results and explanations in a form that is both readable and executable.

Python [3] [1] is a programming language that created in 1990 by Guido van Rossum at Stichting Mathematisch Centrum. In this thesis I used version of Python 3.0 (Python 3000) or Py3K, there are many changes applied on this version compared with Python 2.0 let you work quickly and integrate systems effectively. All of these changes are important for Python users since a lot of old cruft had been removed. Changes include the following:

[1] `https://www.anaconda.com/`
[2] `https://jupyter.org/`
[3] `https://www.python.org/download/releases/3.0/`

- Print is now a function, print function does not support soft space feature like old version.
- Views and Iterators instead of lists. Some APIs will no longer returns a list.
- Ordering comparisons, Python makes comparisons simple.
- Long integers renamed to int, and other integers expressed as different way.
- Text vs. data instead of Unicode vs. 8-bit, Python uses text and binary data instead of Unicode and 8-bit strings. Text is typed as (str) and data is (bytes) so any attempt to mix these two types in Python 3.0 will make a type error. File names are passed to and return from applications and functions as (Unicode) strings.

Python is a strong and powerful programming language used in many science fields like machine learning and data science. Python environment provide tools that develop skills for data analysis and stunning data visualization with matplotlib, folium and seaborn, it also gets a practical introduction to many of machine learning applications. Python provides many libraries like pandas and numpy that used to analysis data, and to build machine learning models using scipy, scikitlearn. Real life data science problems will be solved efficiently with Python 3.0. Because of the previous features of Python 3.0 and its importance in data science, we have chosen it as the primary language for our master thesis.

3.2 Thesis Methodology

To achieve this objective of thesis that discussed before, we have built a customized model for sentiment evaluation to measure tensions level(score) in the middle east. The data collected from the Arabic headlines of the news sites, then the required pre-processing and labelled have performed in order to get the clean data ready as an input for the learning model. finally, the model have learnt to determine the tensions level in the area.

In this thesis, a new model was built for the evaluation of tensions level in the middle east. The data collected from the Arabic news headlines from the news sites on the web. In workflow of thesis, the first thing was to find suitable Arabic news data for the experiment

and then convert the arabic news data to English using Google Cloud Translate API in the Python code. The second most important thing after collecting and translating the news headlines data for this project is the use of different Python libraries for the pre-processing of the data. Here we have worked on the regression problem so we have assigned weight/score to each headline based on the different sentiments before using any machine learning algorithm for training the data set.

Figure 3.1 shows the work flow of thesis.

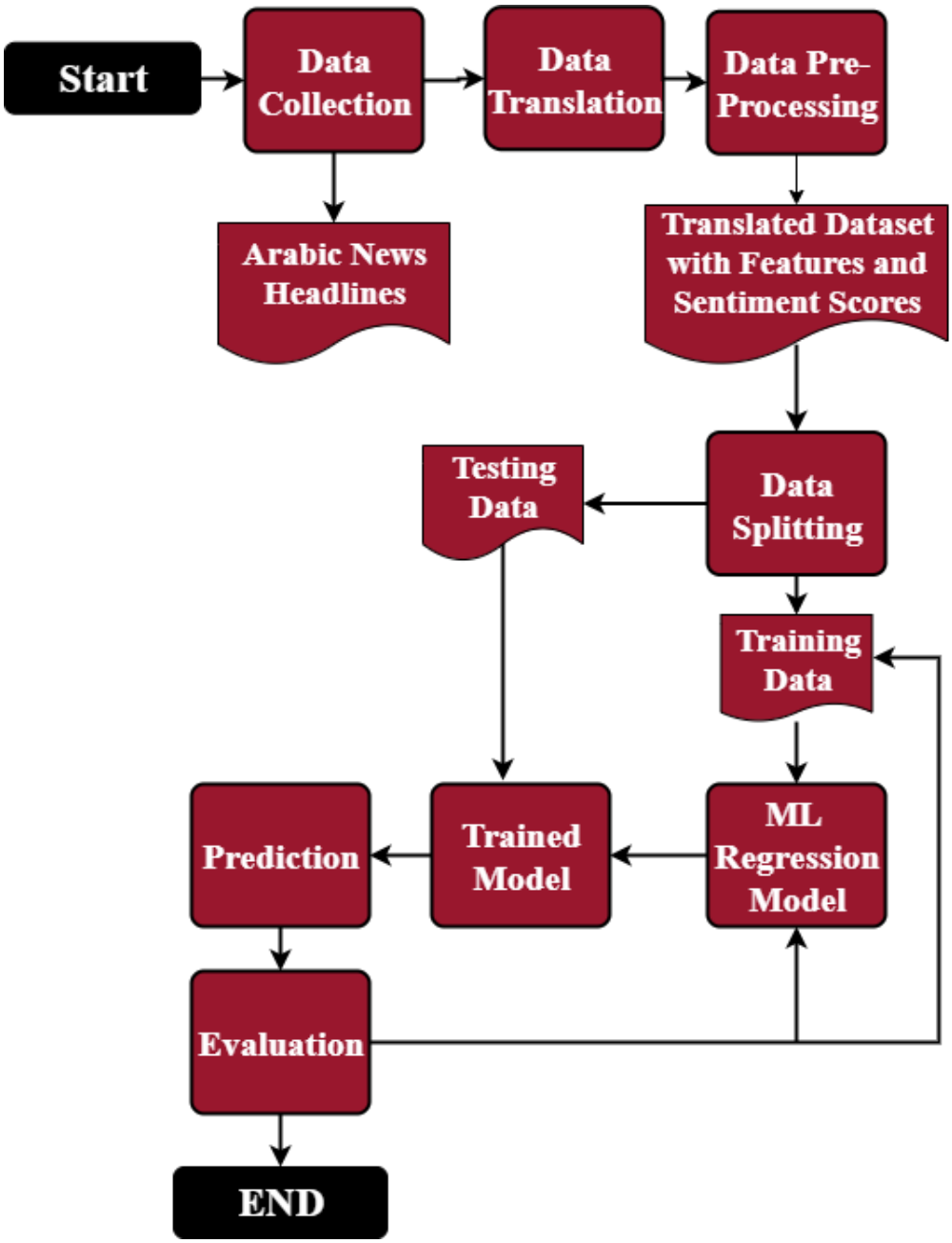

Figure 3.1: Workflow of Thesis

3.3 Data Collection

The collected data is Arabic News headlines for NLP collected from these news websites BBC Arabic[4], EuroNews [5], Aljazeera[6], CNN Arabic[7] and RT Arabic[8] by Dr Motaz Saad [8]. He crawled data in Crawl Date: 19-04-2019, these news are collected by news-please Python library. To extract news and headlines from articles we are using json methods from Python "json2corpus.py" because this tool store collected articles in JSON format, so one can split article's information easily (headline, date, text, images, ...etc.). Here [5] we found all corpus information, also here [3] We Found all dataset. Also, there are tables show the most and least common words in each corpus, and the vocabulary size and the number of terms in each corpus. more over we found tables show headline dates distribution for each corpus.

3.4 Data pre-processing

NLTK (Natural Language Toolkit) have been used to learn the concept of text mining and data pre-processing before doing classification. NLTK is a Python library and we know that Python is simple and very powerful programming language. For purposes of scientific, educational or industrial research Python is being used greatly around the world. It is open source and highly readable. Its syntax is not hard to understand. As an object oriented language Python allow data and method to be encapsulated and re-used easily. For natural language processing NLTK provides basic classes for the data representations. It is easy to do pre-processing on data by writing simple and small codes we can do things like tokenization, part of speech tagging, stemming, stop word removal etc. Which will clean the data and make it more useful for the classification or regression problem.

Figure 3.2 shows the workflow of Data Pre-processing steps.

[4] `https://www.bbc.com/arabic`
[5] `https://arabic.euronews.com`
[6] `https://www.aljazeera.net`
[7] `https://arabic.cnn.com`
[8] `https://arabic.rt.com`

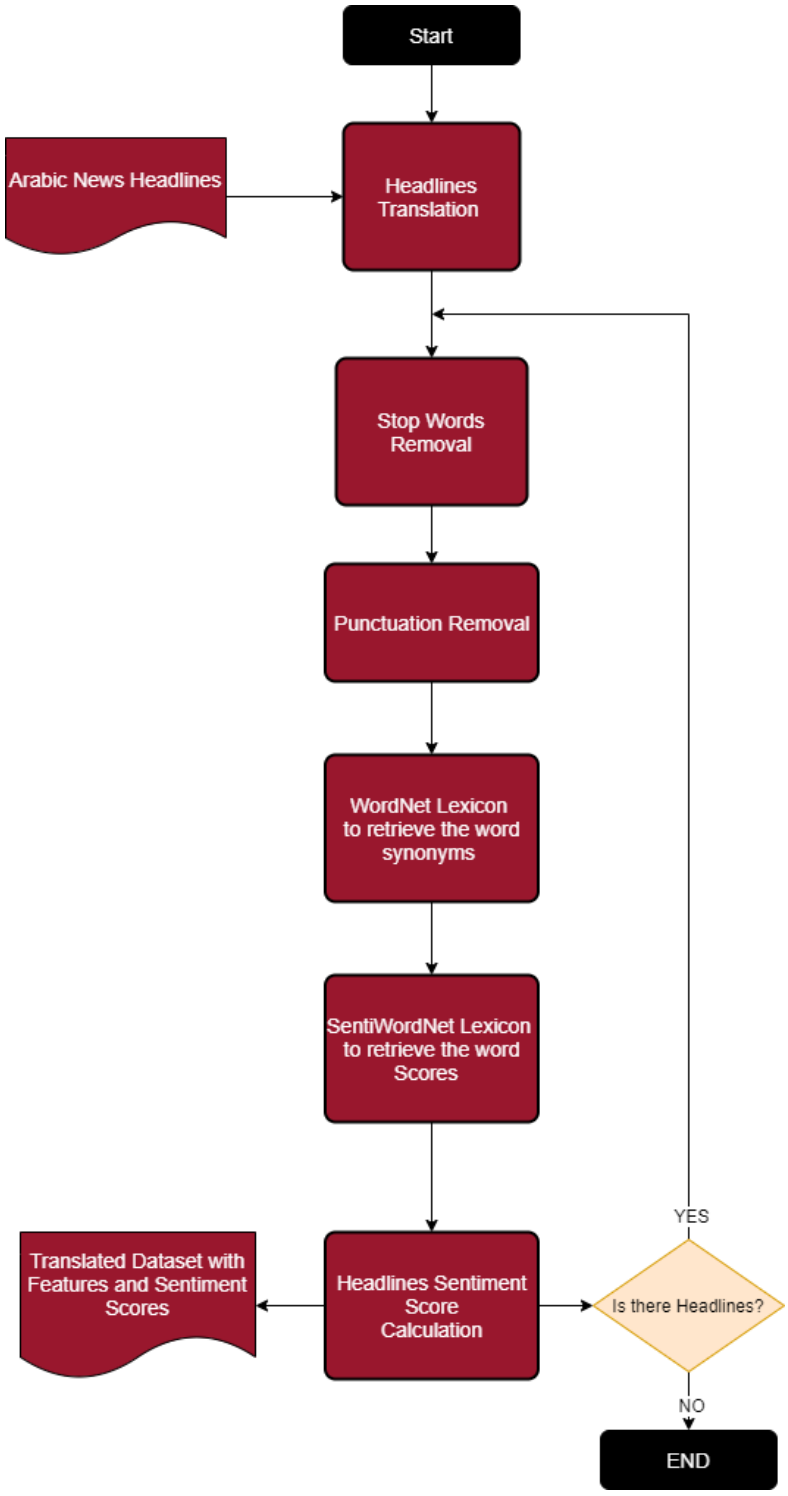

Figure 3.2: Workflow of Data Pre-processing

3.4.1 News headline Translation

In this step we have translated the news headlines data-set to English using Google Cloud Translation API [9][4]. This step came due to the strength of WordNet and SentiWordNet

[9] `https://cloud.google.com/translate`

lexicon for supporting English language. while, their Arabic lexicon wasn't contains sufficient supporting and scores for Arabic language. So it's better to use English in sentiment analysis while using lexicon based approach. SentiWordNet distinguish the score of positive and negative words based on their context, meanings and synonyms and gives each one of them a different score such as killed, killer, kill etc.

We did two experiments in this aspect. In the first experiment we have translated the headlines by translating the words separately from Arabic into English. The second experiment was to translate the entire headline in the context where it was the best because it preserves the meaning and context of the entire headline and increases the accuracy of prediction. We have presented the results of this experiment in the next chapter.

We had start with Google Translate API [6], but there was a big limit on the number of localized requests if they were up to 60 request per day, so the best solution is Google cloud Translate API. Algorithm 1 shows Google cloud translation API call function for translating a single word or a headline.

Algorithm 1 Google Cloud Translate API Call

translation ← translateClint.translate(word, source-language='ar',
target-language='en')

Return translation['translationtext']

Algorithm 2 shows the implementation of bulk translation of Arabic news headline represented as Pandas dataframe and translate each headline by calling the Algorithm 1 for executing the actual GCP translate API call. Also in Algorithm 2 we re-create new GCP authentication object every 400 request in order to avoid blocking the current IP from accessing the GCP.

Algorithm 2 headline Translations

translationClient ← gcpAuthenticate( )

counter ← 1

dataset ← [ ]

try

for *index, row in arabicheadlinesDF.interrows( ):* **do**

  if *(counter % 400)=0* **then**

    translationClient ← gcpAuthenticate( )

    OUTPUT 'create new authenticationobject ::' counter

  end

  englishheadline ← googleCloudTranslateAPICall(row.headline, translateClient)

  counter += 1

  dataset.append([row.headline.replace('\n',''), englishheadline, 0, 0, 0, row.date])

end

df ← pd.DataFrame(dataset, columns=['Arabicheadline', 'Englishheadline', 'PositiveScore', 'NegativeScore', 'Sentiment', 'Date'], dtype=float)

Return df

except Expectation as e

df ← pd.DataFrame(dataset, columns=['Arabicheadline', 'Englishheadline', 'PositiveScore', 'NegativeScore', 'Sentiment', 'Date'], dtype=float)

OUTPUT str(e)

Return df

3.4.2 Stop Words Removal

Filtering useless data is one of the major forms of pre-processing. In the processing of natural language, useless words (data) are called stop words. A stop word is a commonly used word (such as "the", "a", "an", "in") to be missed by a search engine, both when indexing search entries and when retrieving them as a result of a search query. We wouldn't want these terms to take up our server storage or take up valuable processing time. We can

easily remove them by saving a list of words you find to be stop words for this. In Python, NLTK(Natural Language Toolkit) has a list of stop words [35] stored in 16 languages. figure 3.3 and 3.4 shows the commands in the Python to check the list of stop words in Arabic and English.

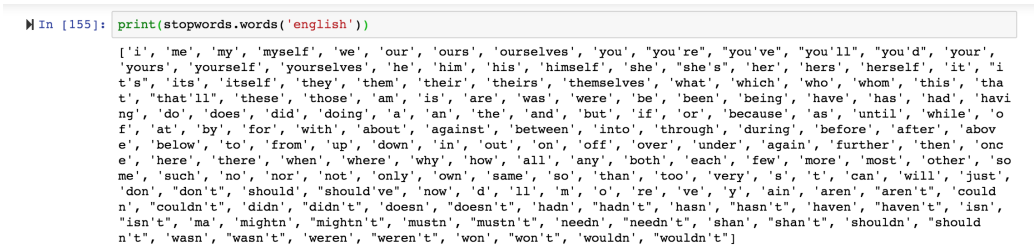

Figure 3.3: Stop Words in English

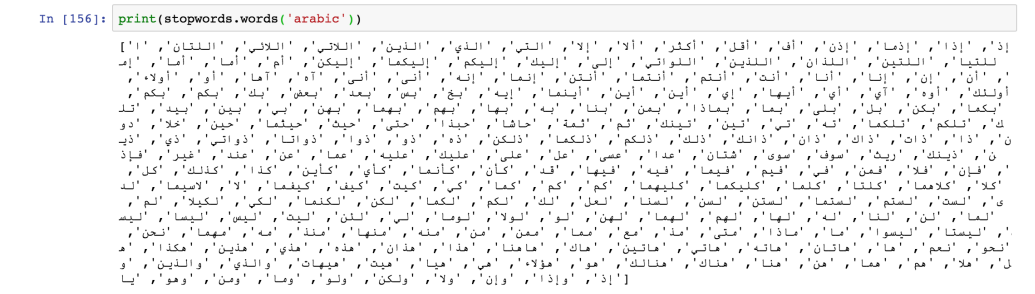

Figure 3.4: Stop Words in Arabic

We did two experiments in this aspect. In the first experiment we remove Arabic stop words before translating headline from Arabic into English. The second experiments is Remove English stop words after translating the headline.

3.4.3 Punctuation Removal

Punctuation can provide an understanding-friendly grammatical context. Punctuation of sentiment analysis does not add value to the bag of words. We wouldn't want these mark to take up storage or processing time. We can easily remove them by updating a list of stop words to include the Punctuation. figure 3.5 shows the commands in the Python for adding Punctuation removal.

Figure 3.5: Punctuation Symbols

3.4.4 WordNet Lexicon

WordNet[47] is a large lexical database. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations[14].

3.4.5 SentiWordNet Lexicon

SentiWordNet is a lexical resource for opinion mining that assigns to each synset of WordNet three sentiment scores: positivity, negativity, and objectivity [13]. It has a Web-based graphical user interface, and it is freely available for research purposes. The development of the resource is based on the quantitative analysis of the glosses associated to synsets, and on the use of the resulting vectoral term representations for semi-supervised synset classification. Positivity, negativity, and objectivity are derived by combining the results produced by a committee of eight ternary classifiers[29].

3.4.6 Sentiment Score calculation

At this stage we have calculated the output value for each headline by applying the quality equation to the two input variables (PosScore, NegScore). This step comes in order to get the training data ready for the stage of regression. In the equation we have calculated the positive sums totals for all the words and the negative sums totals for all the words in each headline, then we took the logarithm of the result of their division. This is the equation of sentiment score.

$$SentimentScore = 10 \log \left(\frac{\sum PosScore}{\sum NegScore} \right)$$

We calculated a value indicating the positive or negative effects of the entire headline. Through the application of this equation all the data is labeled and ready to train in the machine learning model. Algorithm 3 shows the preparing news headlines for training the model which means calculating the positive, negative and sentiment scores for each headline.

Algorithm 3 Prepare headline Scores Training

```
Begin
    try
    englishheadlineScoreDataset ← [ ]
    for index, row in arabicheadlinesDF.interrows( ): do
        englishheadline ← row.Englishheadline
        words ← word-tokenize(Englishheadline)
        filteredWords ← [w for w in words If w.isalpha( ) AND not w in englishStopWords]
        headlineTotalNegativeScore ← 0, headlineTotalPositiveScore ← 0
        scoreCountPerheadline ← 0, headlineSentimentScore ← 0
        for word in filteredWords: do
            synsets ← wn.synsets(word)
            if not synsets then
                Continue
            end
            selectedSynset ← None, maxNetagiveScore ← 0
            for synset-word in synsets: do
                senti-synset ← swn.senti-synset(synset-word.name( ))
                negativeScore ← senti-synset.neg-score( )
                if negativeScore > maxnegativeScore: then
                    maxNetagiveScore ← NegativeScore
                    selectedSynset ← synset-word
                end
            end
            if maxnetafgiveScore=0: then
                selectedSynset ← synsets[0]
            end
        end
    end
```

```
for index, row in arabicheadlinesDF.interrows( ): do
    for word in filteredWords: do
        swn-synset ← swn.senti-synset(selectedSynset.name( ))
        headlineTotalPositiveScore +=swn-synset.pos-score( )
        headlineTotalNegativeScore +=swn-synset.neg-score( )
        headlinePreScore ← (1+ headlineTotalPositiveScore)/(1+ headlineTotalNega-
        tiveScore)
        headlineSentimentScore ← 10*math.log10(headlinePreScore)
        scoreCounterPerheadline += 1
    englishheadlineScoreDataset.append([row.Arabicheadline.replace('\n',''), english-
    headline, scoreCountPerheadline, headlineTotalPositiveScore, headlineTotalNega-
    tiveScore, headlineSentimentScore, row.Date])
df ← pd.DataFrame(englishheadlineScoreDataset, columns=['Arabicheadline',
'Englishheadline', 'NumberOfWords', 'SumPositiveScores', 'SumNegativeScores',
'headlineSentimentScore', 'Date'], dtype=float)
Return df
except Exception as e
OUTPUT str(e)
```

3.5 Regression Model

Machine learning is a data-driven process which uses statistical methods. Machine learning means learning from the past data to predict the future outcomes. After a survey Pang and Lee [49] stated that, the increase in the amount of labelled sentiment relevant data was an important contribution factor to activity in both supervised and unsupervised learning. Supervised machine learning technique is applied to the labelled training data, to train the

classifier for the decision making. We have used regression supervised machine learning algorithms for our research work. There were several reasons for choosing a regression model in this study. The main and important reason is to facilitate the process of expressing tension with a number so that we can plot the results in ascending and descending curves. In addition to, our need to express more accurately the tension away from the classification of groups.

The regression algorithms starts by trying to estimate the mapping function (f) from the input variables (x) to the numerical or continuous output variables (y). Now, a real value could be the output variable, which could be an integer or a floating point value. The regression prediction issues are therefore usually quantities or sizes. For example, if you are presented with a house dataset and are asked to predict their prices, this is a job of regression as the price is going to be a continuous production.

Linear Regression is generally classified into two types:

1. Simple Linear Regression:

   Simple Linear Regression, this is one of the most common and interesting types of regression technique. Here we predict a target variable Y based on the input variable X. A linear relationship should exist between target variable and predictor and so comes the name Linear Regression. The hypothesis of linear regression is $y = a + bx$.

2. Multiple Linear Regression (MLR):

   We try to find a relationship between two or more independent variables (inputs) and the corresponding dependent variable (output) in Multiple Linear Regression. The independent variables can be categorical or continuous. Figure 3.6 show the formula that explains how the expected values of y are related to the independent variables of p is called the equation of multiple linear regression.

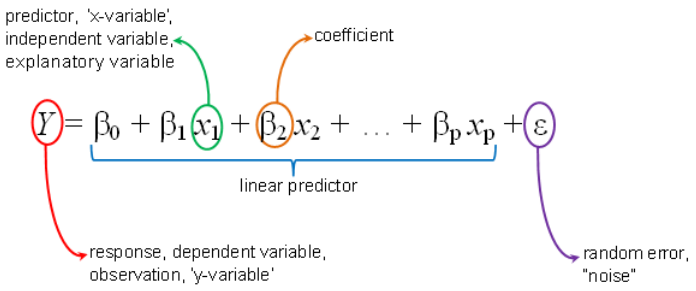

Figure 3.6: Multiple Linear Regression equation

In our thesis, We have implemented this type of regression because we introduce two variables into the model(PosScore, NegScore)in order to predict one variable (Sentiment Score). We implemented the regression model using the Python Scikit-Learn library, which is one of the most popular machine learning libraries for Python[11].

Perhaps Scikit-learn [10][12] is the most useful machine learning library in Python. It's on NumPy,Pandas, SciPy and matplotlib, this library contains many effective machine learning and statistical modeling tools including classification, regression, clustering and reduction of dimensionality. Think of any supervised learning algorithm you've heard of, and there's a very high chance it's part of scikit-learning. Starting with general linear models (e.g. linear regression).

To train the model after the pre-processing phase we execute We have split our data into training and testing sets using train test split function. Then, implementing linear regression models with Scikit-Learn is extremely straightforward, as all you need to do is import the LinearRegression class, instantiate it, and call the fit () method together with our training data. This is about as simple as having to train on your data when using a machine learning library.

Now that we have trained our model, some predictions have to be made. To do this, we have used our test data and see how accurately the percentage score is predicted by our model, to make predictions about the test data. Finally, we have evaluated the performance of model by finding the values for Explained variance score, Mean squared error(MSE), R^2 Score from sklearn metrics [7].

[10] `https://scikit-learn.org/stable/`

Explained variance score calculates the explained variance regression score. $\hat{y}$ is the predicted output, y the correct testing output, and V_{ar} is Variance, the square of the standard deviation, then the explained variance is calculated as follow:

$$explainedvariance\left(y,\hat{y}\right)=1-\frac{V_{ar\{y-\hat{y}\}}}{V_{ar}\left\{y\right\}}$$

The best possible score of explained variance regression score is 1.0, lower values are worse.

Mean squared error (MSE) is threat measure equivalent to the calculated square (quadratic) error or loss cost. $\hat{y}_i$ is the predicted output of i-th sample, and y the correct testing output, then the mean squared error (MSE) calculated over n_{sample} as follow:

$$MSE\left(y,\hat{y}\right)=\frac{1}{n_{sample}}\sum_{i=0}^{n_{sample}-1}\left(y_i-\hat{y}_i\right)^2$$

R^2 Score is the coefficient of determination. It represents the proportion of variance (of y) that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance. As such variance is dataset dependent, R^2 may not be meaningfully comparable across different datasets, best possible score is 1.0. $\hat{y}_i$ is the predicted output of the i-th sample and y the correct testing output for total n samples, the estimated R^2 calculated as follow:

$$R^2\left(y,\hat{y}\right)=1-\frac{\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2}$$

where $\bar{y}=\frac{1}{n}\sum_{i=1}^{n}y_i$.

In summary, we import the packages and classes needed, we provide data to work with and eventually do appropriate transformations, create a regression model and fit it with existing data, check the results of model fitting to know whether the model is satisfactory, apply the model for predictions. Algorithm 4 shows the functions taking care of splitting the news dataset into training and testing and use them to train and

test the model, also showing the model accuracy percent.

Algorithm 4 Training and Testing

X ← np.asarray(englishheadlinesWithScore[[‘SumPositiveScores’, ‘SumOfNegativeScores’]])

Y ← np.asarray(englishheadlinesWithScore[‘headlineSentim

entScore’])

X-train, X-test,Y-train,Y-test ← train-test-split(X, Y, test-size=0.30, train-size=0.70)

linearRegressionModel ← LinearRegression( )

linearRegressionModel.fit(X-train, Y-train)

Y-predict ← linearRegressionModel.predict(X-test)

OUTPUT ‘R^2 The Coefficient of Determination of the Prediction’ r^2-score(Y-test, Y-predict)

OUTPUT ‘The Coefficient;’, linearRegressionModel.coef-

OUTPUT ‘Intercept-:’, linearRegressionModel.inercept-

OUTPUT ‘Explained Variance Regression Score:’, explained-variance-score(Y-test, Y-predict)

OUTPUT ‘Mean Squared Error’, mean-squared-error(Y-test, Y-predict)

In addition to the previous steps, the code was created to receive the element of time and training it to make other experiments and link it to time, where the following algorithm 5 explains how to group the news headlines and there scores per day.

Algorithm 5 Filter the dataset per time period and group the dataset per daily functions

filterDatasetByTimePeriod(dataframe, start-date, end-date))

filteredDataset = dataframe[(dataframe[‘Date’] ≥ start-date) & (dataframe[‘Date’] ≤ end-date)]

Return filteredDatasetdef groupDatasetPerDay(dataframe)

groupedDataByDay = dataframe.groupby(‘Date’).sum().reset-index()

groupedDataByDay[‘TitleSentimentScore’]=(1+groupedDataByDay[‘SumPositiveScores’]) / (1 + groupedDataByDay[‘SumOfNegativeScores’])

groupedDataByDay[‘TitleSentimentScore’]=groupedDataByDay[‘TitleSentimentScore’].apply(lambda x : 10 * math.log10(x))

groupedDataByDay[‘Week-Number’] = groupedDataByDay.Date.dt.week

Return groupedDataByDay

3.6 Implementation

Algorithm 6 shows the functions responsible for reading any news datasource file and return it as pandas dataframe

Algorithm 6 Reading any news DataSource File

newsDF ← pd.read-csv(arabicEuroNews, sep=‘\t’, lineterminator=‘\n’, header=None)

newsDF.columns ← [‘headline’, ‘date’]

Return newsDF

Algorithm 7 shows functions responsible for inserting the predicted headlines inside DB real time analysis displayed by Charts.

Algorithm 7 Inserting the Predicted headlines Inside DB

try

conn← create-connection( )

cur← conn.cursor( )

month← int(str(row.Date).split('-')[1])

Newsheadline←(row.Englishheadline,row.SumPositiveScores,
row.SumNegativeScores,row.Date,selectedYear,month,row.headlineSentimentScore)

sql←INSERT INTO news_sentiment (headline, pos_score,neg_score, date, year, month, sentiment_score) VALUES(?,?,?,?,?,?,?)

cur.execute(sql, Newsheadline)

conn.commit( )

Return cur.lastrowid

except Error as e:

OUTPUT e

Algorithm 8 show functions responsible for inserting the predicted headlines inside DB real time analysis displayed by Charts.

Algorithm 8 Prediction News headlines On The Fly

```
Begin
    try
    emptyDataIntoDB( ), englishheadlineScoreDataset ← [ ]
    TranslateClient ← gcpAuthenticate( ), counter ← 1
    for index, row in filteredDFPerYear.interrows( ): do
        if (counter % 400)=0 then
            translationClient ← gcpAuthenticate( )
            OUTPUT 'create new authenticationobject ::', counter
        end
        englishheadline ← googleCloudTranslateAPICall(row.headline, translateClient)
        counter += 1
        words ← word-tokenize(englishheadline)
        filteredWords ← [w for w in words If w.isalpha( ) AND not w in englishStopWords]
        SumPositiveScore ← 0, SumNegativeScore ← 0
        scoreCountPerheadline ← 0, headlineSentimentScore ← 0
        for word in filteredWords: do
            synsets ← wn.synsets(word)
            if not synsets then
                Continue
            end
            selectedSynset ← None, maxNetagiveScore ← 0
            for synset-word in synsets: do
                senti-synset ← swn.senti-synset(synset-word.name( ))
                negativeScore ← senti-synset.neg-score( )
                if negativeScore > maxNetagiveScore: then
                    maxNetagiveScore ← NegativeScore
                    selectedSynset ← synset-word
                end
            end
        end
    end
```

```
for index, row in filteredDFPerYear.interrows( ): do
    for word in filteredWords: do
        if maxNetagiveScore=0: then
            selectedSynset ← synsets[0]
        swn-synset ← swn.senti-synset(selectedSynset.name( ))
        SumPositiveScore +=swn-synset.pos-score( )
        SumNegativeScore +=swn-synset.neg-score( )
        scoreCounterPerheadline += 1
    X-score ← np.asarray([[SumPositiveScore, SumNegativeScore]])
    predicted-Score ← linearRegressionModel.predict(X-Scores)
    sentimentScore ← predicted-Score[0]
    insertNewRecord(englishheadline, SumPositiveScores, SumNegativeScores,
    SentimentScore, row.Date, selectedYear)
except Exception as e
OUTPUT str(e)
```

Algorithm 9 shows main part of code for calling and starting the predicting process and showing the analysis in real time after choosing the data source.

Algorithm 9 Real Time

```
1 arabicEuroNews ← 'headlines-dates/arabic.euronews.com_20190409_date_headlines.txt' ;
/* Pointing to Arabic euronews datasource */
2 arabicCNN ← 'headlines-dates/arabic.cnn.com_20190419_date_headlines.txt' ; /* Pointing
 to Arabic CNN news datasource */
3 selectedYear ← '2018' ;     /* Determine the selected year to perform our prediction */
4 newsDF ← readNewsData(arabicCNN) ;  /* Call reading Arabic news dataset function
 */
5 filteredDF ← newsDF[newsDF['date'].str.contains(selectedYear) = True ] ;   /* Filter
 the Arabic news by selected date */
6 filteredDF ← filteredDF.head(100) ;              /* Take first 100 rows as a sample  */
7 predictNewsheadlinesOnTheFly(filteredDF, selectedYear) ;     /* Call predict function
 with given sample to start processing */
```

Figure 3.7 indicates an overview of our model to really predict. We get news headlines from different sources, whether cloud, drive or file system. Then, data have entered to ML prediction function Which in turn sends the prediction results for the DataBase to draw and represent it directly in the Dashboard.

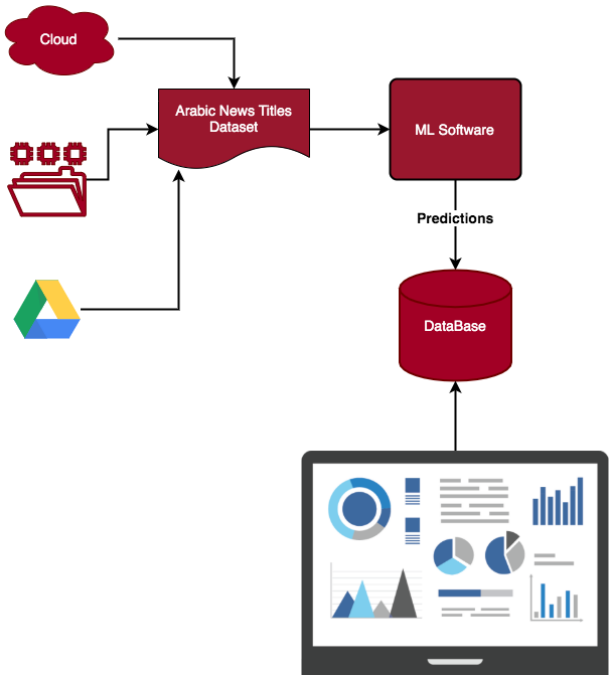

Figure 3.7: ML Software Architecture

Chapter 4

Experiments and Results

4. Chapter Outline:

    4.1. Experiments

        4.1.1. Experiments Environment

        4.1.2. Experiment A:Training Sentiment Model using Arabic headlines.

        4.1.3. Experiment B:Training Sentiment Model using translated arabic headlines (English headlines).

        4.1.4. Experiment C:Training Sentiment Model using translated arabic headlines (English headlines) group per day (Daily headlines).

    4.2. Discussions And Results

        4.2.1. Results

        4.2.2. Discussions

    4.3. Real Time Analysis Experiment

In this chapter, we have described the experiments that we have conducted in order to evaluate our proposed model. We have discussed the results of these experiments.

4.1 Experiments

4.1.1 Experiments Environment

The experimental environment used for all experiments was CPU/Intel Core Pentium i7 processor, Memory of 8 GB RAM, System type 64-bit operating system, Windows 10. In addition, we used anaconda-navigator 1.9.7[1], Jupyter Notebook[2] and Python 3 [3]for preparing our data by implementing the steps of learning regression model. Also, we used flask[4] framework with SQL3 lite to make real time prediction.

4.1.2 Experiment A: Training Sentiment Model using Arabic headlines

For this experiment, we used the data that was prepared beforehand as described in Section 3.3, which represents headlines collected from five news websites. We took a sample AlJazeera website file as a dataset. We will explain the steps of the experiment with realistic examples from the data and then we will discuss the results in the next section of this chapter. In this experiment we did not translate Arabic, but we used WordNet and SentiWordNet lexicons that support the Arabic language, the following algorithm 10 explains the changes on algorithm 3, which illustrates here the use of the Arabic version of the lixicons. As a result, small percentage of the headlines have been calculated and have a sentiment score due to the lack of words and weak support in Arabic. So, as we will see in the results in the next section, most of scores value centered around zero, where data is a fake.

[1]https://www.anaconda.com/
[2]https://jupyter.org/
[3]https://www.python.org/download/releases/3.0/
[4]https://flask.palletsprojects.com/en/1.1.x/

Algorithm 10 Arabic Version of SentiWordNet

Calculate the positive and negative score for Arabic words function

```
for word in filteredWords do
    st = ISRIStemmer()
    stemWord = st.stem(word)
    synsets = wn.synsets(stemWord, lang='arb')
    if not synsets then
        continue
    end
end
```

The following table 4.1 shows sample of the Arabic headlines data format before the pre-processing.

Table 4.1: Arabic headlines Dataset

NO.	Arabic News headlines	Date
1	قرار أممي يدعو الأسد للتنحي	2012-02-19
2	العراق يلغي صفقة سلاح مع روسيا	2012-11-10
3	جدل بمصر حول قرارات مرسي	2012-11-22
4	ثلاثة وسبعون شهيدا في غزة ومجزرة بخان يونس	2014-07-23
5	حلب.. ستة أيام من القصف الدامي	2016-09-26
...	...	...

The following table 4.2 shows the sample of headlines in DataFrame after pre-processing where the values of the score is calculated.

Table 4.2: Dataset After All pre-processing Phase

NO.	Arabic headline	NoOfWords	SumPS	SumNS	Sentiment Score	Date
1	قرار أممي يدعو الأسد للتنحي	1.0	0.0	0.25	-0.969100130080564	2012-02-19
2	العراق يلغي صفقة سلاح مع روسيا	2.0	0.25	0.0	0.9691001300805642	2012-11-10
3	جدل بمصر حول قرارات مرسي	3.0	0.125	0.375	-1.3830269816628142	2012-11-22
4	ثلاثة وسبعون شهيدا في غزة ومجزرة بخان يونس	1.0	0.0	0.0	0.0	2014-07-23
5	حلب.. ستة أيام من القصف الدامي	1.0	0.0	0.0	0.0	2016-09-26
...	...	...	...	...	...	...

4.1.3 Experiment B: Training Sentiment Model using translated arabic headlines (English headlines)

For this experiment, we used the data that was prepared beforehand as described in Section 3.3, which represents headlines collected from five news websites. We took a sample from the Al Jazeera website file, we used four different size of dataset for this experiment as follows, 5000 headlines, 15000 headlines, 30000 headlines and 80000 headlines as a dataset. We will explain the steps of the experiment with realistic examples from the data and then we will discuss the results in the next section of this chapter.

The following table 4.3 shows sample of the original data format before the pre-processing.

Table 4.3: Arabic Headlines Dataset

NO.	Arabic News headlines	Date
1	قرار أممي يدعو الأسد للتنحي	2012-02-19
2	العراق يلغي صفقة سلاح مع روسيا	2012-11-10
3	جدل بمصر حول قرارات مرسي	2012-11-22
4	ثلاثة وسبعون شهيدا في غزة ومجزرة بخان يونس	2014-07-23
5	حلب.. ستة أيام من القصف الدامي	2016-09-26
...	...	...

The following table 4.4 shows the sample of headlines in DataFrame after sending them to Google Cloud Translation API to translate the news headlines from Arabic to English before pre-processing where the values of the score is still zero and not calculated.

Table 4.4: Translated Arabic Headlines

NO.	Arabic headline	English headline	SumPS	SumNS	Sentiment Score	Date
1	قرار أممي يدعو الأسد للتنحي	UN resolution calls on Assad to step down	0.0	0.0	0.0	2012-02-19
2	العراق يلغي صفقة سلاح مع روسيا	Iraq cancels arms deal with Russia	0.0	0.0	0.0	2012-11-10
3	جدل بمصر حول قرارات مرسي	Controversy in Egypt over Morsi decisions	0.0	0.0	0.0	2012-11-22
4	ثلاثة وسبعون شهيدا في غزة ومجزرة بخان يونس	73 martyrs in Gaza and the massacre of Khan Younis	0.0	0.0	0.0	2014-07-23
5	حلب.. ستة أيام من القصف الدامي	Aleppo .. Six days of bloody bombing	0.0	0.0	0.0	2016-09-26
...	...	...	...	...	...	...

The following table 4.5 shows the sample of headlines in DataFrame after pre-processing

where the values of the score is calculated.

Table 4.5: Dataset After All Pre-processing Phase

NO.	Arabic headline	English headline	NoOfWords	SumPS	SumNS	Sentiment Score	Date
1	قرار أممي يدعو الأسد للتنحي	UN resolution calls on Assad to step down	4.0	0.125	2.0	-4.259687322722812	2012-02-19
2	العراق يلغي صفقة سلاح مع روسيا	Iraq cancels arms deal with Russia	5.0	0.0	0.125	-0.5115252244738131	2012-11-10
3	جدل بمصر حول قرارات مرسي	Controversy in Egypt over Morsi decisions	3.0	0.125	0.0	0.5115252244738129	2012-11-22
4	ثلاثة وسبعون شهيدا في غزة ومجزرة بخان يونس	73 martyrs in Gaza and the massacre of Khan Younis	4.0	0.0	0.75	-2.4303804868629446	2014-07-23
5	حلب.. ستة أيام من القصف الدامي	Aleppo .. Six days of bloody bombing	5.0	0.0	0.375	-1.3830269816628142	2016-09-26
...	...	...	...	...	...	...	...

4.1.4 Experiment C: Training Sentiment Model using translated arabic headlines (English headlines) grouped per day(Daily headlines)

For this experiment, we used the data that was prepared beforehand as described in Section 3.3, which represents headlines collected from five news websites. We took a sample of the RT file, we used four different size of dataset for this experiment as Shown in 4.6. This experiment takes day by day input and not the headline by headline like previous experiment. We will explain the steps of the experiment with realistic examples from the dataset and then we will discuss the results in the next section of this chapter.

The following table 4.6 shows Four Dataset in different size information of this experiment.

Table 4.6: Training headline per Day Dataset

3 month	6 month	9 month	12 month
headlines count = 3378	headlines count = 6604	headlines count = 10100	headline count = 14442
Days = 90	Days = 180	Days = 273	Days = 365

The following table 4.7 shows the sample of the original data format before the pre-processing which is like previous experiment but is an Connected period from date to date.

The following table 4.8 shows the sample of connected period headlines in DataFrame after sending them to Google Cloud Translation API to translate the news headlines from Arabic to English before pre-processing where the values of the score per day is still zero and not calculated.

Table 4.7: Arabic headlines connected period Dataset

Date	Arabic News headlines
2014-07-07	قتلى وجرحى في تفجيرين انتحاريين ببغداد
2014-07-07	فيديو للغارات الإسرائيلية على قطاع غزة مصَور من الطائرة
2014-07-08	مقتل ١١ مسلحا و ٤ جنود بهجومين للقاعدة في اليمن
...	...
2014-07-29	مقتل ١٠ جنود يرفع حصيلة خسائر الجيش الإسرائيلي إلى ٥٣
2014-07-29	رئيسة البرازيل تصف العملية الإسرائيلية على غزة بـلْمجزرةْ
2014-07-29	القصف الإسرائيلي يوقف محطة كهرباء غزة
...	...

Table 4.8: Translated Arabic Headlines per Day

Date	Arabic headline	English headline	SumPS	SumNS	Sentiment Score
2014-07-07	قتلى وجرحى في تفجيرين انتحاريين ببغداد	Suicide bombers killed and wounded in Baghdad	0.0	0.0	0.0
2014-07-07	فيديو للغارات الإسرائيلية على قطاع غزة مصَور من الطائرة	Video of the Israeli raids on the Gaza Strip	0.0	0.0	0.0
2014-07-08	مقتل ١١ مسلحا و ٤ جنود بهجومين للقاعدة في اليمن	11 insurgents and 4 soldiers killed in al-Qaeda attacks in Yemen	0.0	0.0	0.0
...	...	...	...	...	...
2014-07-29	مقتل ١٠ جنود يرفع حصيلة خسائر الجيش الإسرائيلي إلى ٥٣	Killing of 10 soldiers raises the number of Israeli army casualties to 53	0.0	0.0	0.0
2014-07-29	رئيسة البرازيل تصف العملية الإسرائيلية على غزة بـلْمجزرةْ	Brazilian president calls Israeli operation on Gaza “massacre”	0.0	0.0	0.0
2014-07-29	القصف الإسرائيلي يوقف محطة كهرباء غزة	Israeli shelling stops Gaza power plant	0.0	0.0	0.0
...	...	...	...	...	...

The following table 4.9 shows the sample of headlines in DataFrame after pre-processing where the values of the sum of positive score and sum of negative score is calculated for each headline, as shown the sentiment score still not calculated. Table 4.10 shows the all calculated sum of the positive sums and negative sums of all headlines for each day in 2014, also the sentiment score calculated for each day using pre-processing steps.

Table 4.9: Dataset with Sum of Positive Score and Sum of Negative Score For each headline per day

Date	Arabic headline	English headline	NOofWords	SumPS	SumNS	Sentiment Score
2014-07-07	قتلى وجرحى في تفجيرين انتحاريين ببغداد	Suicide bombers killed and wounded in Baghdad	5	0.25	1.5	-3.010299957
2014-07-07	فيديو للغارات الإسرائيلية على قطاع غزة مصَور من الطائرة	Video of the Israeli raids on the Gaza Strip	5	0.0	0.75	-2.430380487
2014-07-08	مقتل ١١ مسلحا و ٤ جنود بهجومين للقاعدة في اليمن	11 insurgents and 4 soldiers killed in al-Qaeda attacks in Yemen	5	0.375	1.75	-3.010299957
...	...	...	...	...	...	...
2014-07-29	مقتل ١٠ جنود يرفع حصيلة خسائر الجيش الإسرائيلي إلى ٥٣	Killing of 10 soldiers raises the number of Israeli army casualties to 53	7	0.375	2.125	-3.565473235
2014-07-29	رئيسة البرازيل تصف العملية الإسرائيلية على غزة بـلْمجزرةْ	Brazilian president calls Israeli operation on Gaza “massacre”	7	0.125	0.75	-1.918855262
2014-07-29	القصف الإسرائيلي يوقف محطة كهرباء غزة	Israeli shelling stops Gaza power plant	6	0.375	1.25	-2.138798199
...	...	...	...	...	...	...

Table 4.10: Dataset After All Pre-processing Phase

Date	NOofWords	SumPS	SumNS	Sentiment Score
2014-07-06	219	7.875	26.5	-4.911643321
2014-07-07	186	9.625	34.625	-5.254259343
2014-07-08	259	12.625	40.625	-4.850177356
...	...	...	...	...
2014-07-29	244	9.75	40.375	-5.853295425
2014-07-30	232	10.763	36.737	-5.062492748
2014-07-31	267	10.667	42.833	-5.748419976
...	...	...	...	...

4.2 Discussions and Results

4.2.1 Results

The following table 4.11 shows 80000 Arabic headlines Dataset results of regression model metrics of experiment A. As we have noticed in the previous table the result in this

Table 4.11: Experiment A metrics results

Metrics	80000 headline
Explained Variance Score	0.976
Mean Secure Error	0.03
R^2 Score	0.98

experiment is promising and deceptive at the same time. We did an experiment where number of headlines Dataset is 109129 but the number of headlines that contains Sentiment Scores is 12278. Most of headlines had Sentiment Scores of Zero. The following figures show what we mean which shows 80000 Arabic headlines dataset training results of experiment A. Figure 4.1 shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict). The x-axis represents the first 250 of the news headlines out of 30% of the testing data. We notice here the y-axis represents the value of the Sentiment Score where the blue color represents the real output value of the

Score, and the orange color represents the value that was measured by the model.

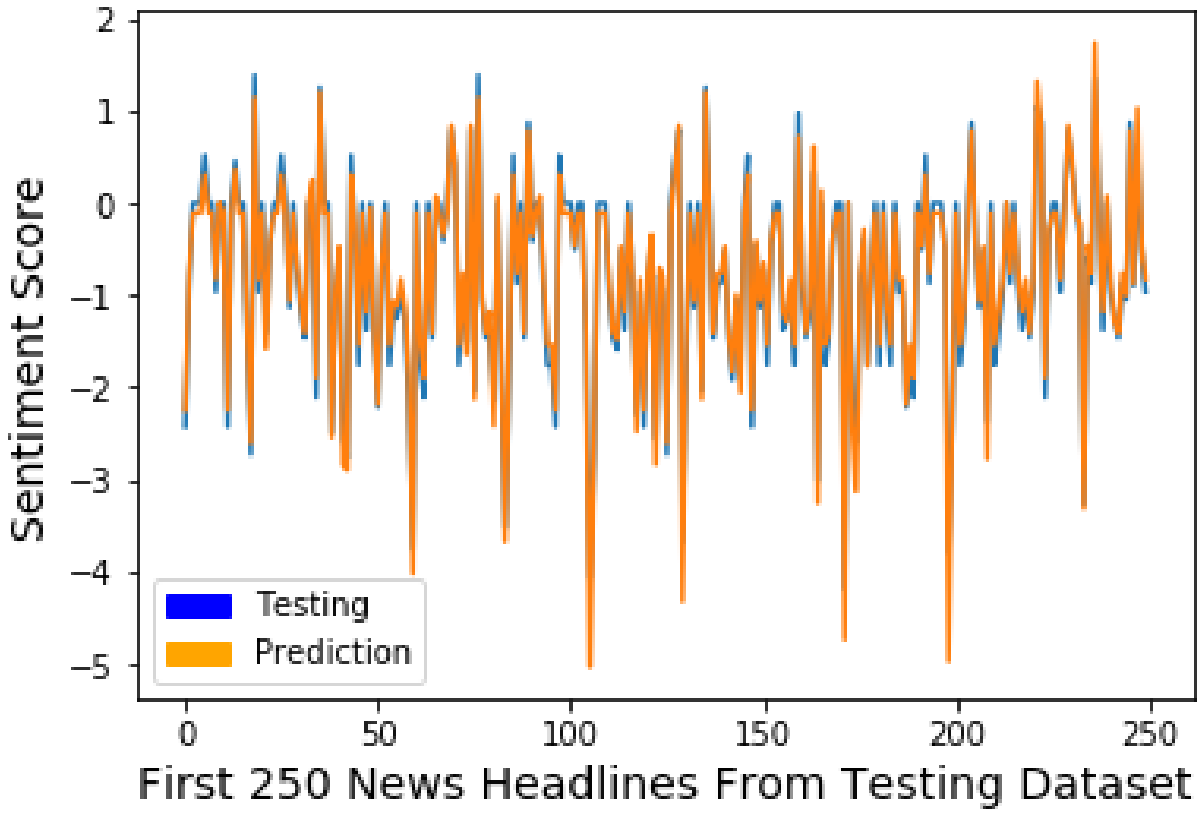

Figure 4.1: The difference between Y_Test and Y_Predict for the experiment A

Figure 4.2 shows relationship between the summation of positive Score of headline and sentiment Score. Where we note that the more total positive increased sentiment Score for the better, ie less tension. Also, Figure 4.2 shows relationship between the summation of negative Score of headline and sentiment Score. Where we note that the less total negative decreased sentiment Score for the worst, ie high tension. The blue color indicates the relationship between sum of positive score for each headline in 80000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each headline in 80000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the inverse proportions between the two axes.

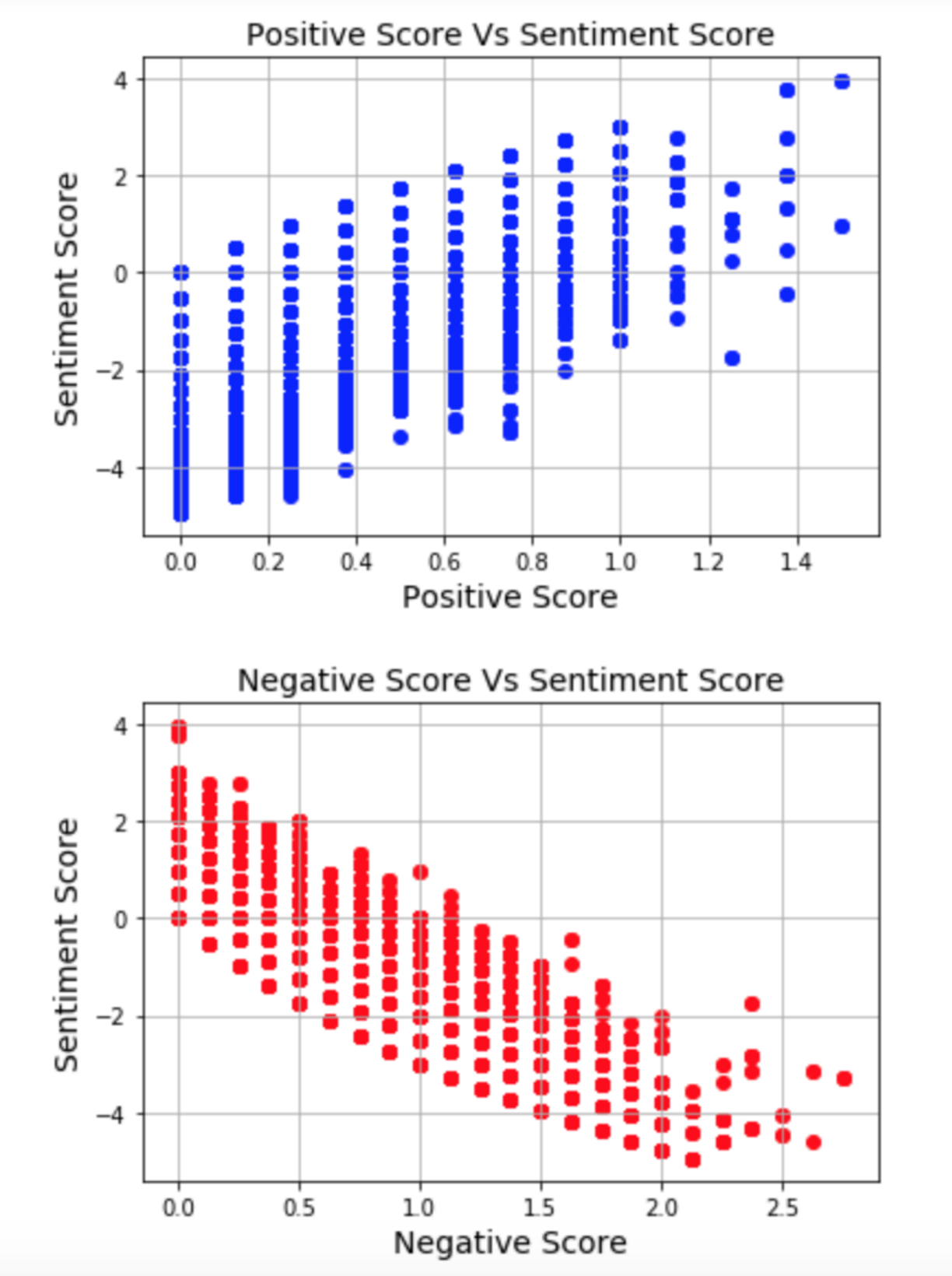

Figure 4.2: Relationship between SumPS, SumNS and Sentiment Score in the experiment A

Figure 4.3 shows sentiment score frequency in relation to the news headlines for all

datasets taken in this experiment, where we notice here the majority of the score centered around zero, and this is what we talked about previously. The figure shows the x-axis that represents the sentiment score for 80000 news headlines divided in periods with the y-axis that represents the number of score frequency within these periods. We notice in this experiment the majority of score around the zero because there are no scores for many titles when using Arab sources directly.

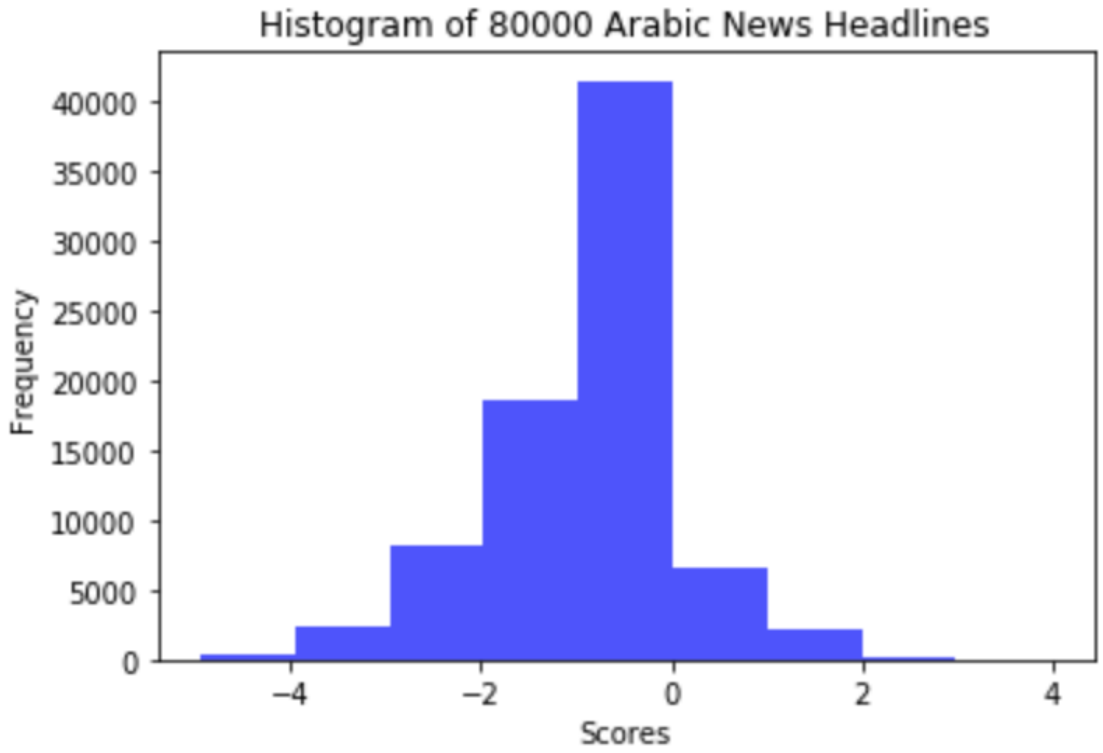

Figure 4.3: Sentiment Score Frequency in the experiment A

The following table 4.12 shows Four Dataset results of regression model metrics in different size information of experiment B.

Table 4.12: Experiment B metrics results

Metrics	5000 headline	15000 headline	30000 headline	80000 headline
Explained Variance Score	**0.962**	**0.968**	**0.969**	**0.968**
Mean Secure Error	**0.07**	**0.06**	**0.05**	**0.06**
R^2 **Score**	**0.96**	**0.97**	**0.97**	**0.97**

Figure 4.4 shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict) when training and testing data is

5000 headlines. The x-axis represents the first 250 of the news headlines out of 30% of the testing data. We notice here the y-axis represents the value of the Sentiment Score where the blue color represents the real output value of the Score, and the orange color represents the value that was measured by the model.

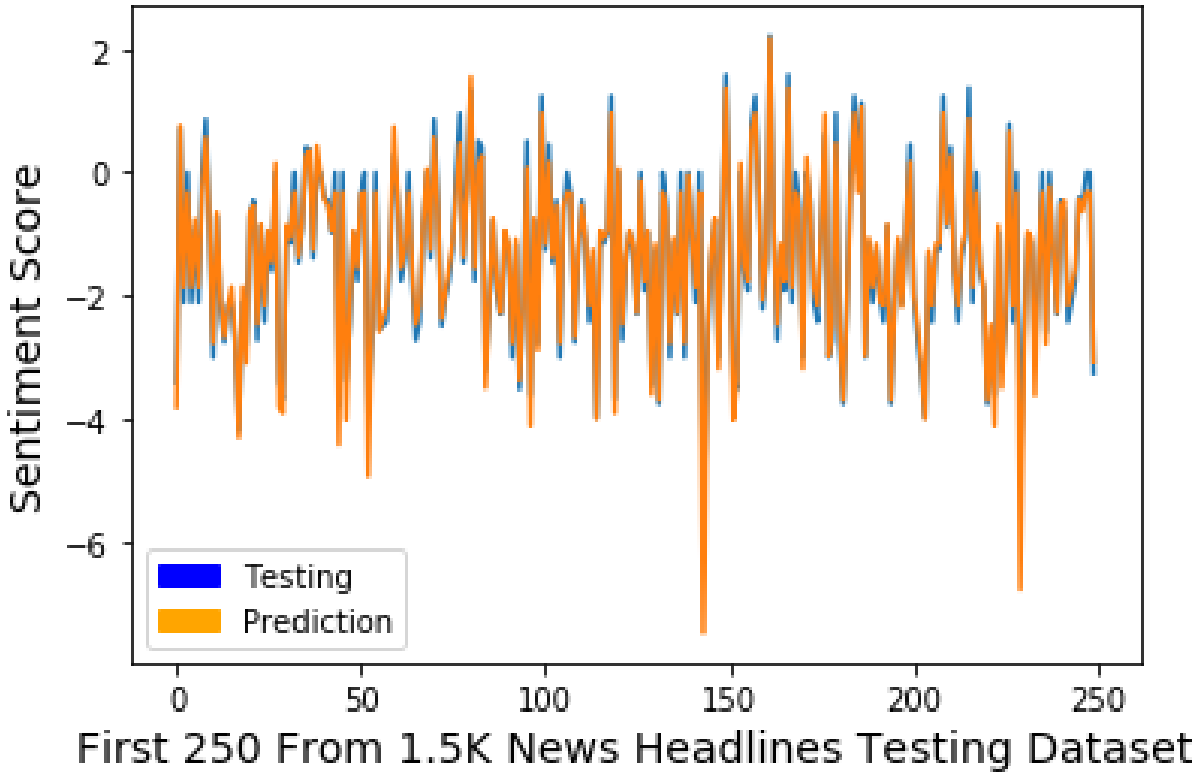

Figure 4.4: The difference between Y_Test and Y_Predict for the experiment B(5000 headlines)

Figure 4.5 shows relationship between the summation of positive Score of headline and sentiment Score when training and testing data is 5000 headlines. Where we note that the more total positive increased sentiment Score for the better, i.e. less tension. Also, Figure 4.5 shows relationship between the summation of negative Score of headline and sentiment Score. Where we note that the less total negative decreased sentiment Score for the worst, i.e. high tension. The blue color indicates the relationship between sum of positive score for each headline in 5000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each headline in 50000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the

figure shows the inverse proportions between the two axes.

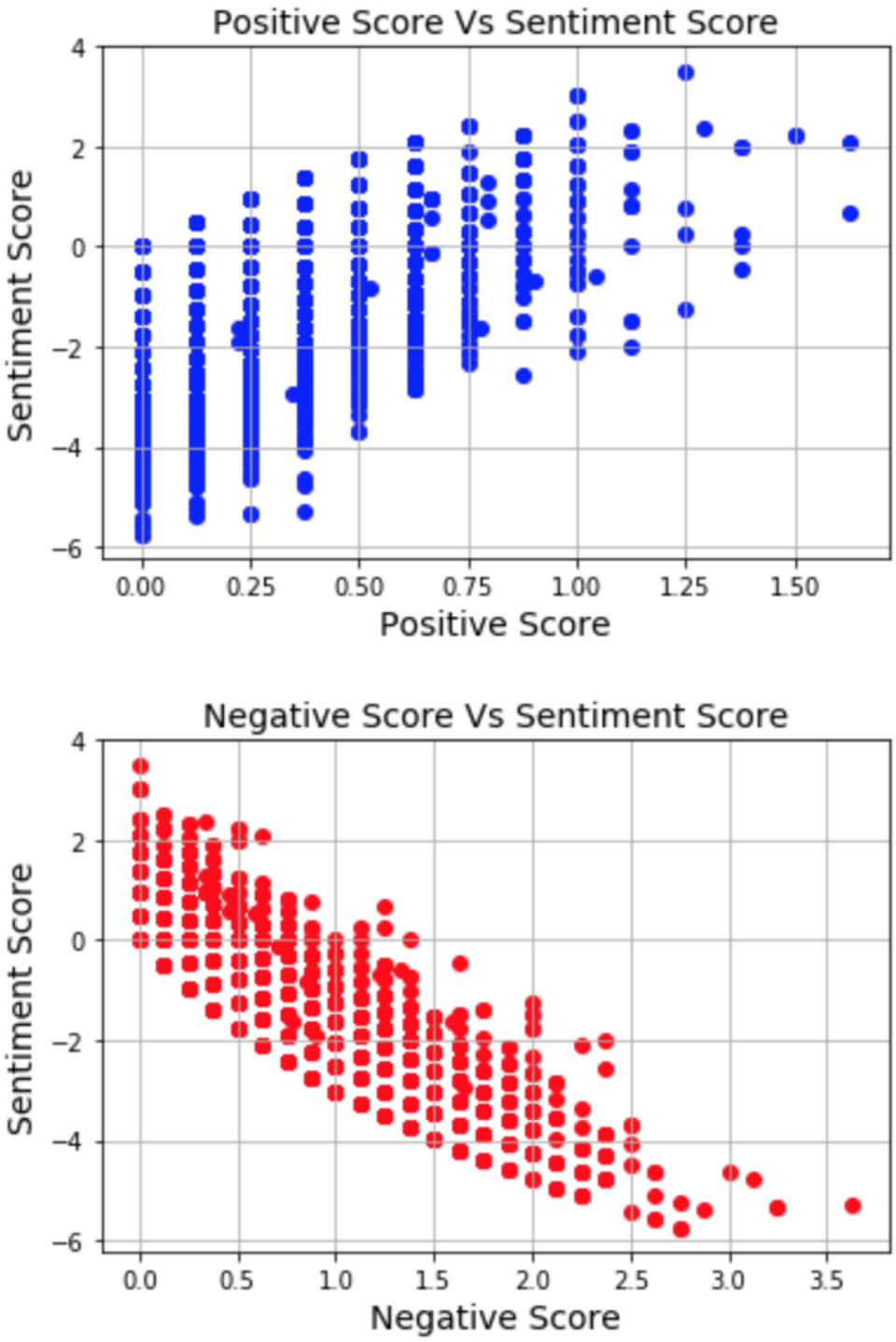

Figure 4.5: Relationship between SumPS, SumNS and Sentiment Score in the experiment B(5000 headlines)

Figure 4.6 shows sentiment score frequency in relation to the number of headlines when training and testing data is 5000 headlines. The figure shows the x-axis that represents the

sentiment score for 5000 news headlines divided in periods with the y-axis that represents the number of score frequency within these periods.

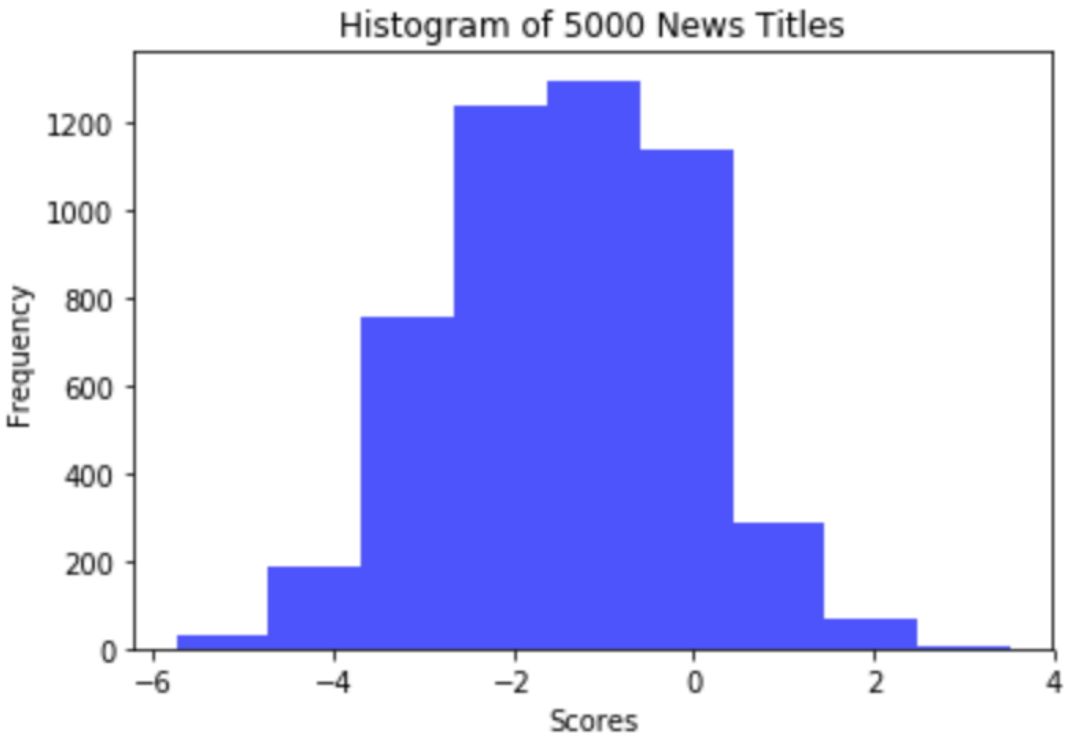

Figure 4.6: Sentiment Score Frequency in the experiment B(5000 headlines)

Figure 4.7 shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict) when training and testing data is 15000 headlines. The x-axis represents the first 250 of the news headlines out of 30% of the testing data. We notice here the y-axis represents the value of the Sentiment Score where the blue color represents the real output value of the Score, and the orange color represents the value that was measured by the model.

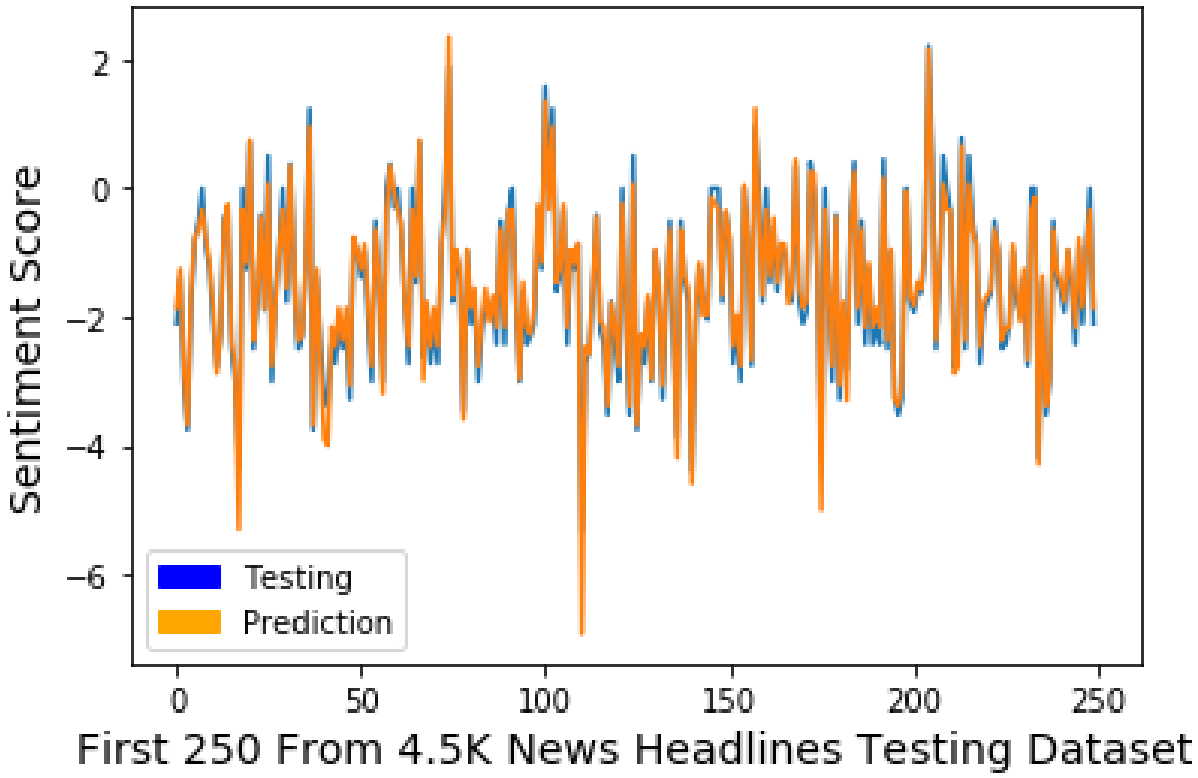

Figure 4.7: The difference between Y_Test and Y_Predict for the experiment B(15000 headlines)

Figure 4.8 shows relationship between the summation of positive Score of headline and sentiment Score when training and testing data is 15000 headlines. Where we note that the more total positive increased sentiment Score for the better, ie less tension. Also, Figure 4.8 shows relationship between the summation of negative Score of headline and sentiment Score. Where we note that the less total negative decreased sentiment Score for the worst, ie high tension. The blue color indicates the relationship between sum of positive score for each headline in 150000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each headline in 150000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the inverse proportions between the two axes.

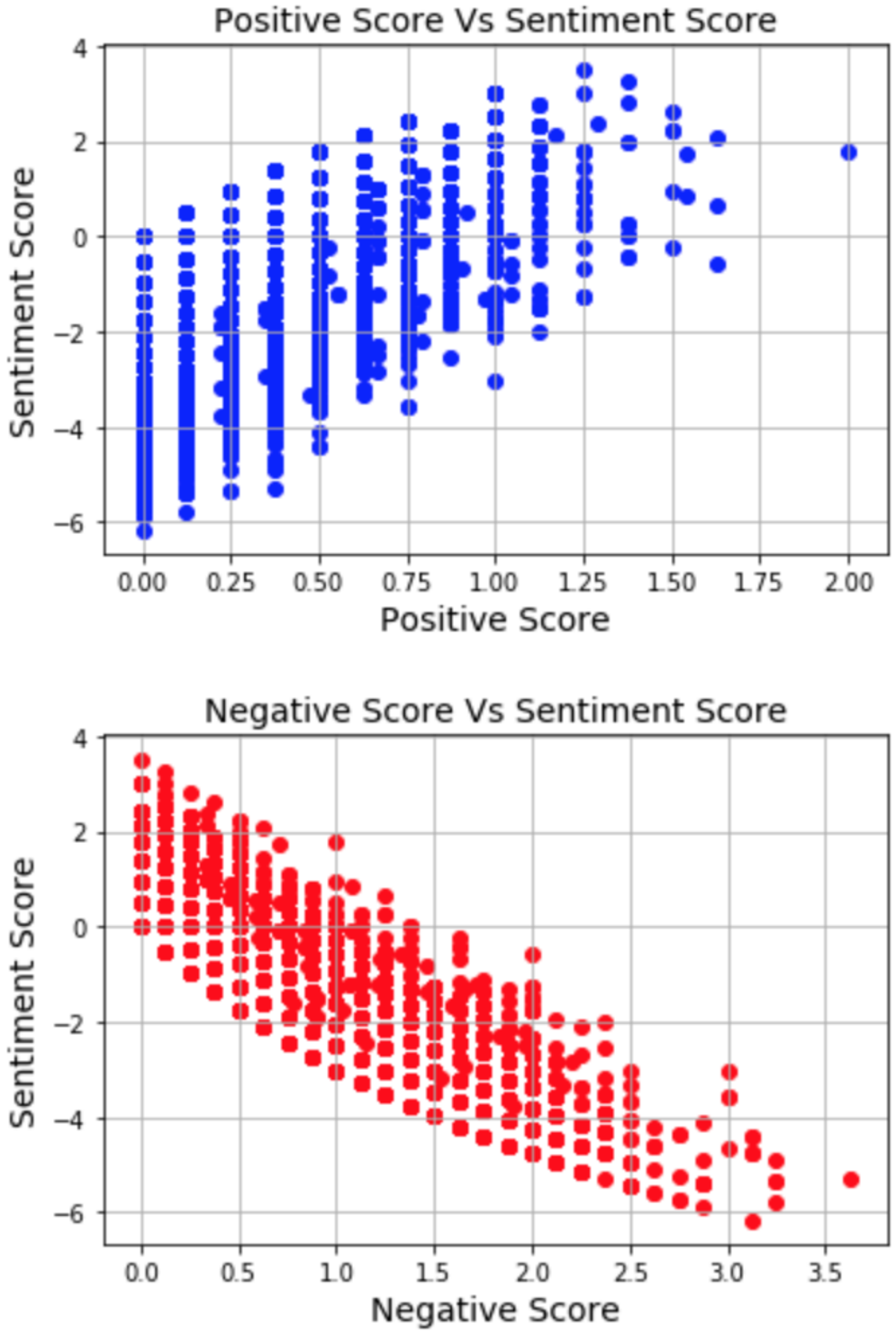

Figure 4.8: Relationship between SumPS, SumNS and Sentiment Score in the experiment B(15000 headlines)

Figure 4.9 shows sentiment score frequency in relation to the number of headlines when training and testing data is 15000 headlines. The figure shows the x-axis that represents the sentiment score for 15000 news headlines divided in periods with the y-axis that represents

the number of score frequency within these periods.

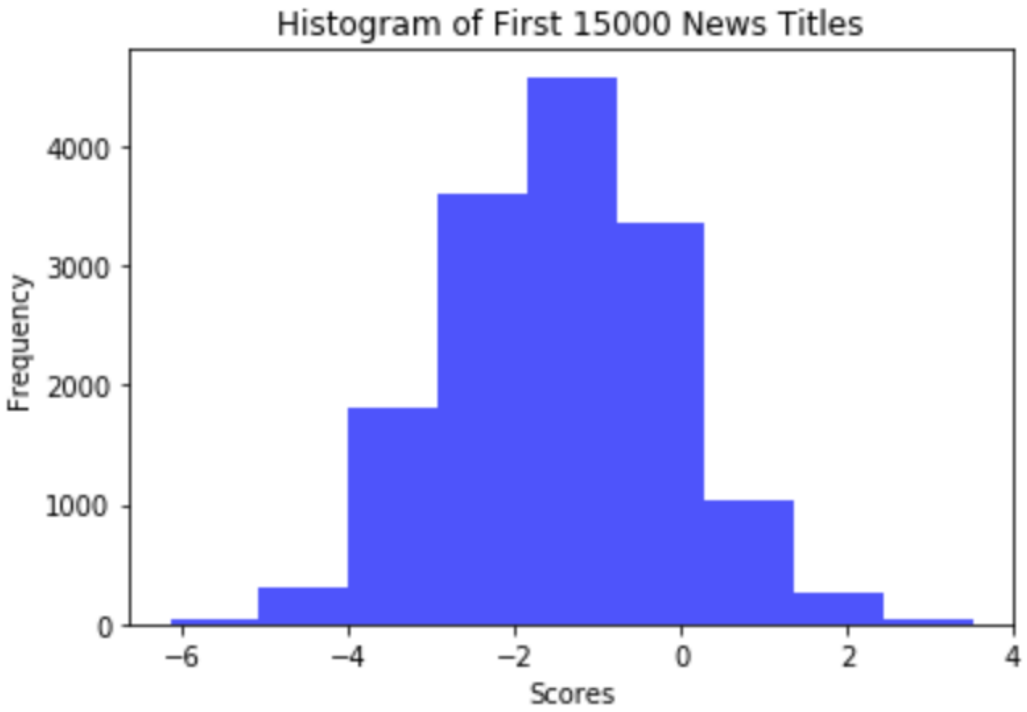

Figure 4.9: Sentiment Score Frequency in the experiment B(15000 headlines)

Figure 4.10 shows shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict) when training and testing data is 30000 headline. The x-axis represents the first 250 of the news headlines out of 30% of the testing data. We notice here the y-axis represents the value of the Sentiment Score where the blue color represents the real output value of the Score, and the orange color represents the value that was measured by the model.

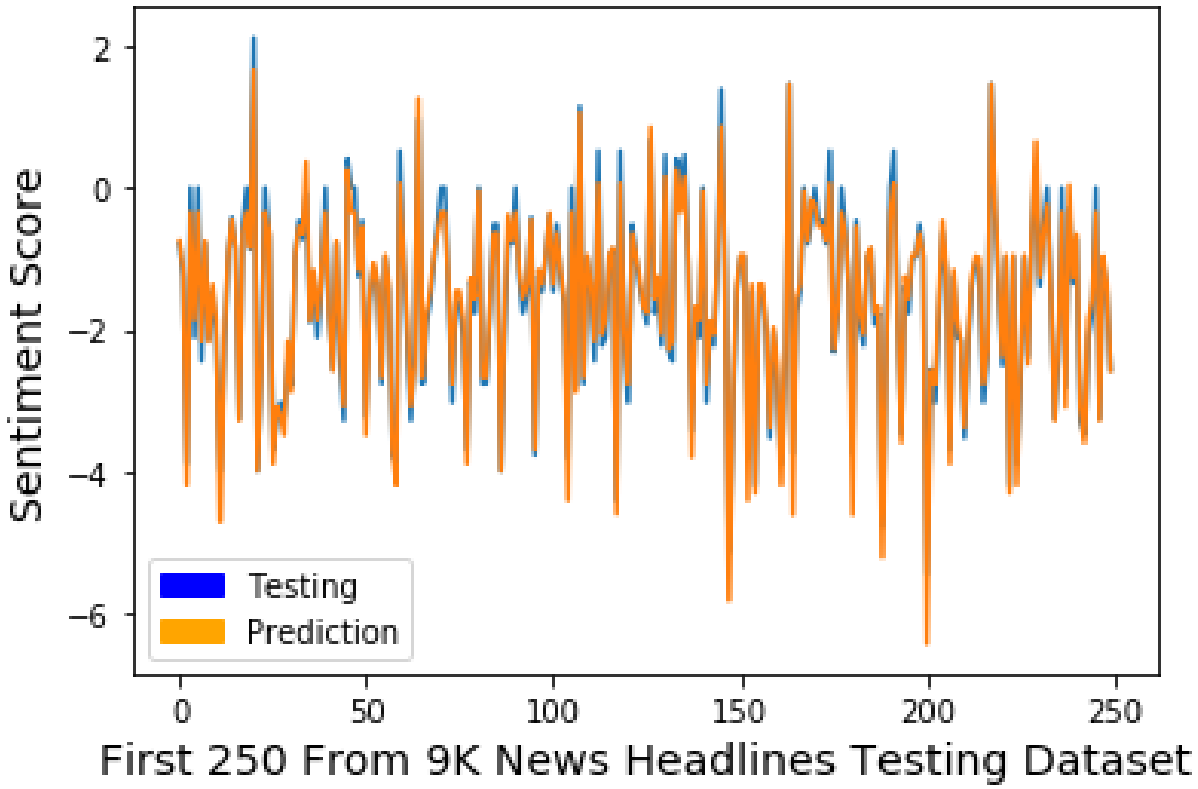

Figure 4.10: The difference between Y_Test and Y_Predict for the experiment B(30000 headlines)

Figure 4.11 shows relationship between the summation of positive Score of headline and sentiment Score when training and testing data is 30000 headlines. Where we note that the more total positive increased sentiment Score for the better, ie less tension. Also, Figure 4.11 shows relationship between the summation of negative Score of headline and sentiment Score. Where we note that the less total negative decreased sentiment Score for the worst, ie high tension. The blue color indicates the relationship between sum of positive score for each headline in 30000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each headline in 30000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the inverse proportions between the two axes.

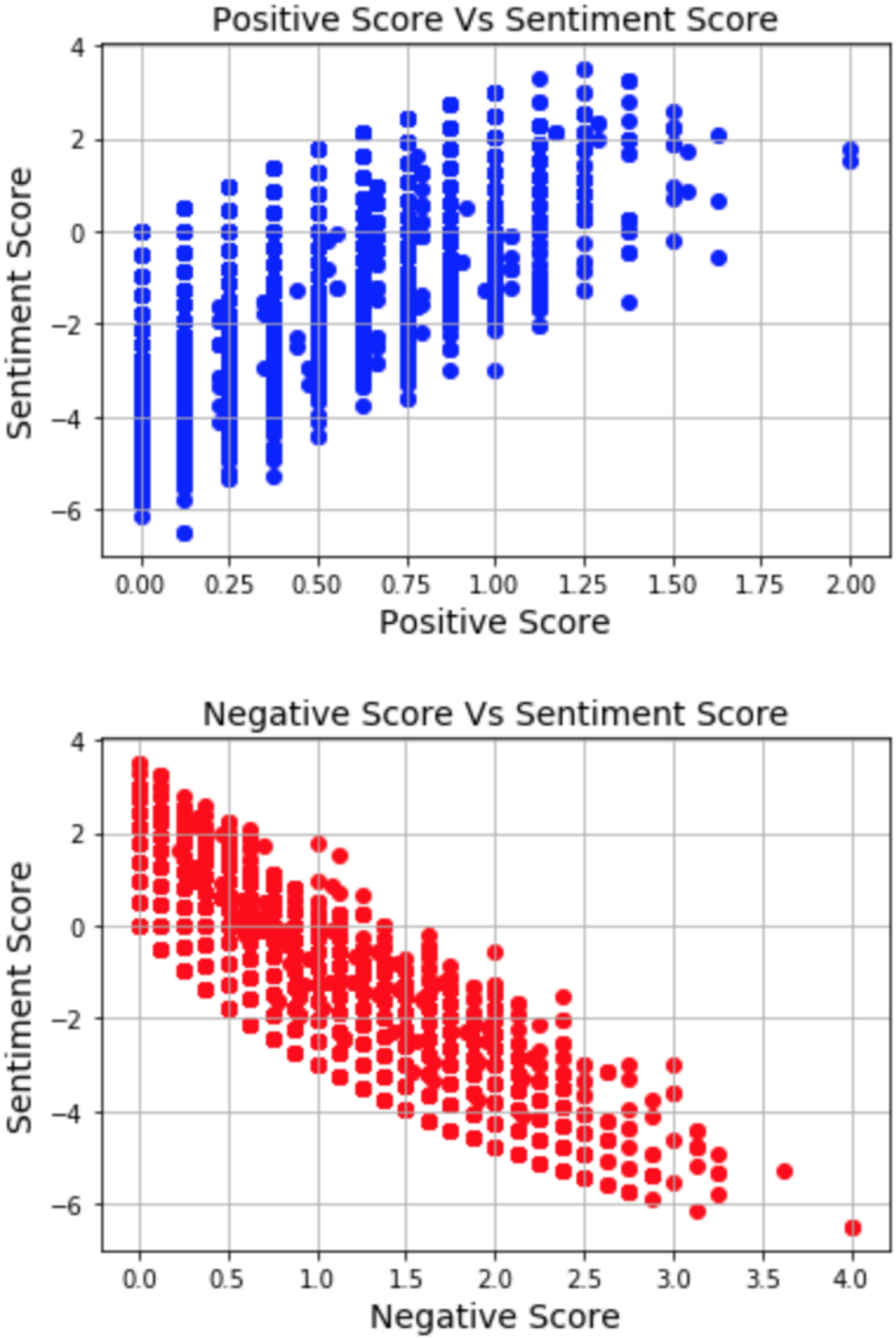

Figure 4.11: Relationship between SumPS, SumNS and Sentiment Score in the experiment B(30000 headlines)

Figure 4.12 shows sentiment score frequency in relation to the number of headlines when training and testing data is 30000 headlines. The figure shows the x-axis that represents the sentiment score for 30000 news headlines divided in periods with the y-axis that represents

the number of score frequency within these periods.

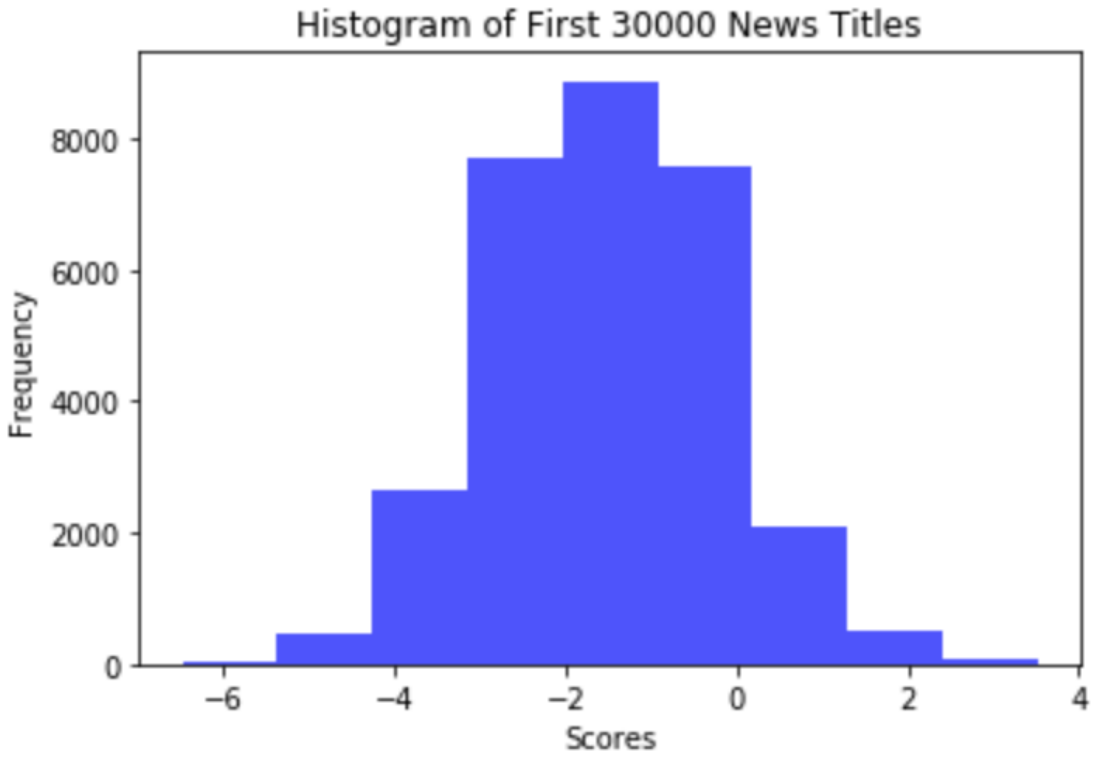

Figure 4.12: Sentiment Score Frequency in the experiment B(30000 headlines)

Figure 4.13 shows shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict) when training and testing data is 80000 headlines. The x-axis represents the first 250 of the news headlines out of 30% of the testing data. We notice here the y-axis represents the value of the Sentiment Score where the blue color represents the real output value of the Score, and the orange color represents the value that was measured by the model.

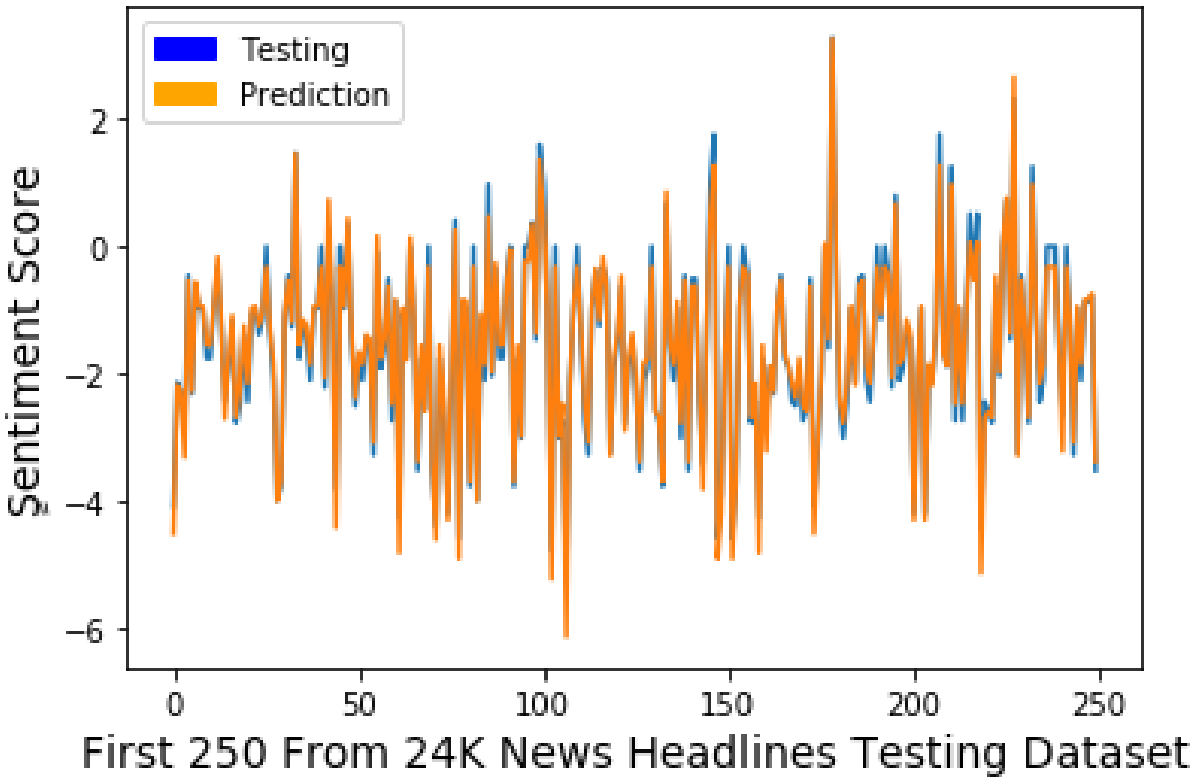

Figure 4.13: The difference between Y_Test and Y_Predict for the experiment B(80000 headlines)

Figure 4.14 shows relationship between the summation of positive Score of headline and sentiment Score when training and testing data is 80000 headlines. Where we note that the more total positive increased sentiment Score for the better, ie less tension. Also, Figure 4.14 shows relationship between the summation of negative Score of headline and sentiment Score. Where we note that the less total negative decreased sentiment Score for the worst, ie high tension. The blue color indicates the relationship between sum of positive score for each headline in 80000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each headline in 80000 news headlines in the x-axis and the sentiment score for them in the y-axis, where the figure shows the inverse proportions between the two axes.

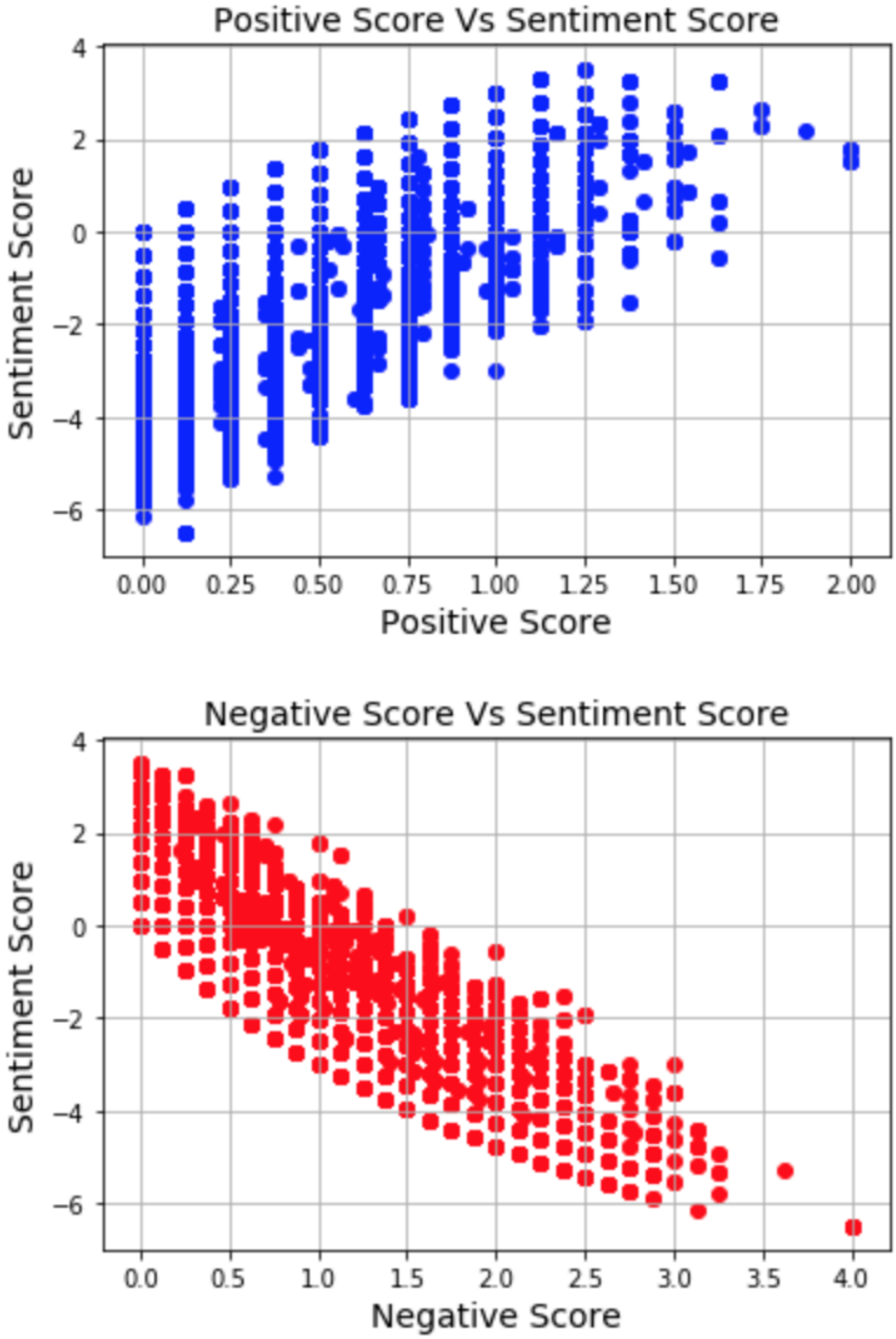

Figure 4.14: Relationship between SumPS, SumNS and Sentiment Score in the experiment B(80000 headlines)

Figure 4.15 shows sentiment score frequency in relation to the number of headlines when training and testing data is 80000 headlines. The figure shows the x-axis that represents the sentiment score for 80000 news headlines divided in periods with the y-axis that represents the number of score frequency within these periods.

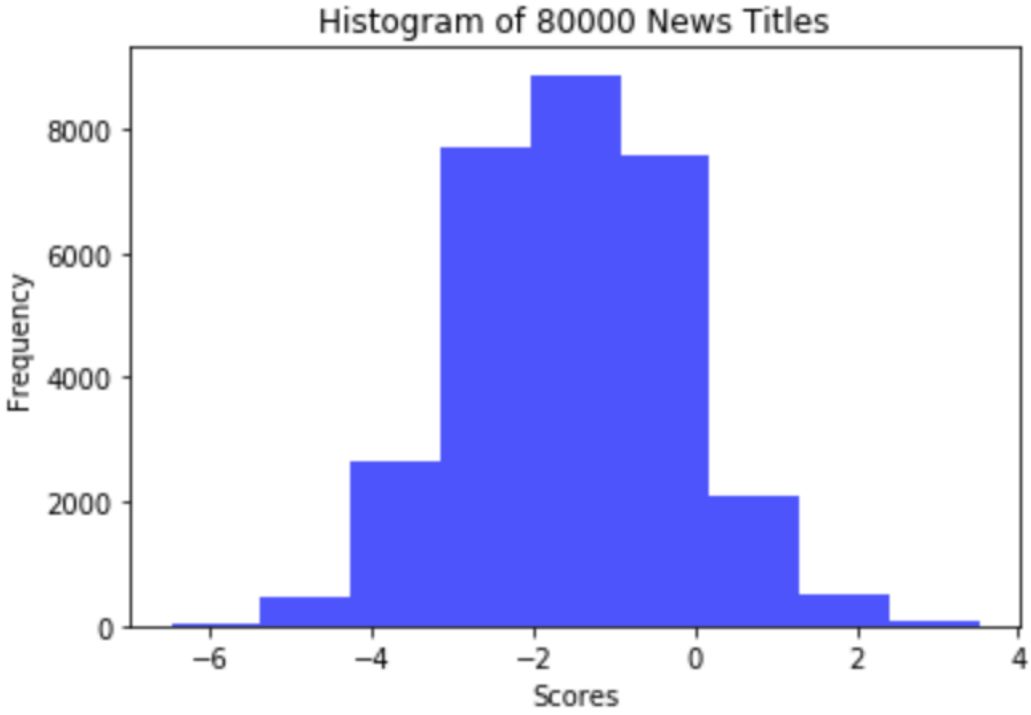

Figure 4.15: Sentiment Score Frequency in the experiment B(80000 headlines)

The following table 4.13 shows Four Dataset results of regression model metrics in different size information of experiment C.

Table 4.13: Experiment C metrics results

Metrics	3 month	6 month	9 month	12 month
Explained Variance Score	**0.816**	**0.922**	**0.931**	**0.937**
Mean Secure Error	**0.10**	**0.06**	**0.05**	**0.04**
R^2 Score	**0.81**	**0.92**	**0.93**	**0.94**

Figure 4.16 shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict) when training and testing data is first 3 month connected period (daily input) in 2014. The x-axis represents one month of daily data as testing data which is 30% of the 3 month dataset. We notice here the y-axis represents the value of the Sentiment Score for each day where the blue color represents the real output value of the Score, and the orange color represents the value that was measured by the model.

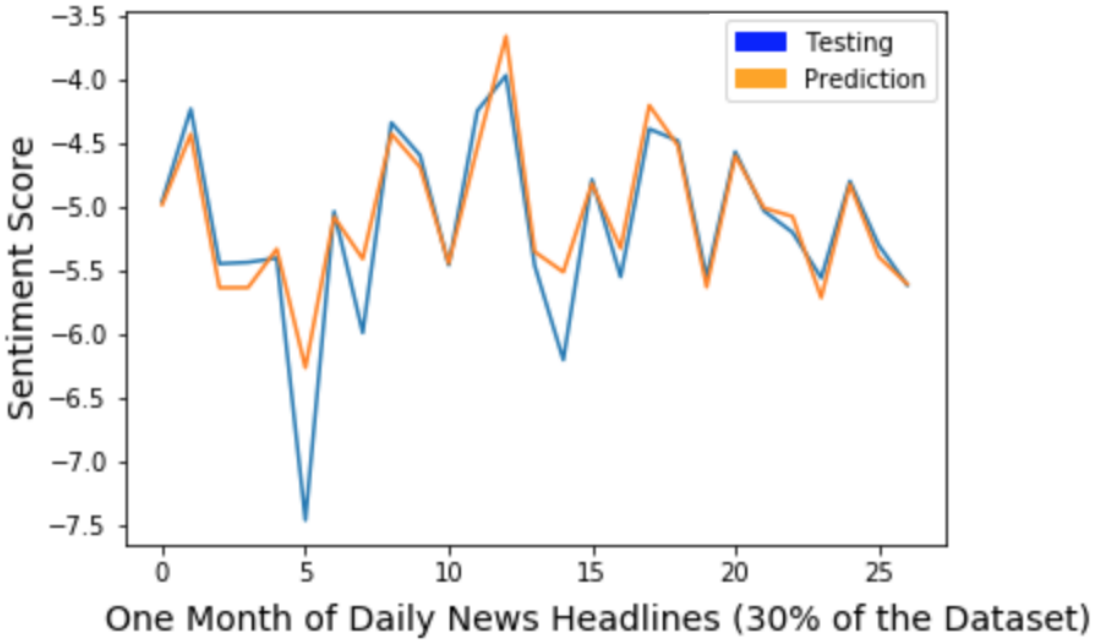

Figure 4.16: The difference between Y_Test and Y_Predict for the experiment C(3 month)

Figure 4.17 shows relationship between the summation of positive Score of headline and sentiment Score per day when training and testing data is first 3 month connected period (daily input) in 2014. Where we note that the more total positive increased sentiment Score for the better, ie less tension. Also, Figure 4.17 shows relationship between the summation of negative Score of headline and sentiment Score per day. Where we note that the less total negative decreased sentiment Score for the worst, ie high tension. The blue color indicates the relationship between sum of positive score for each day in 3 month dataset in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each day in 3 month dataset in the x-axis and the sentiment score for them in the y-axis, where the figure shows the inverse proportions between the two axes.

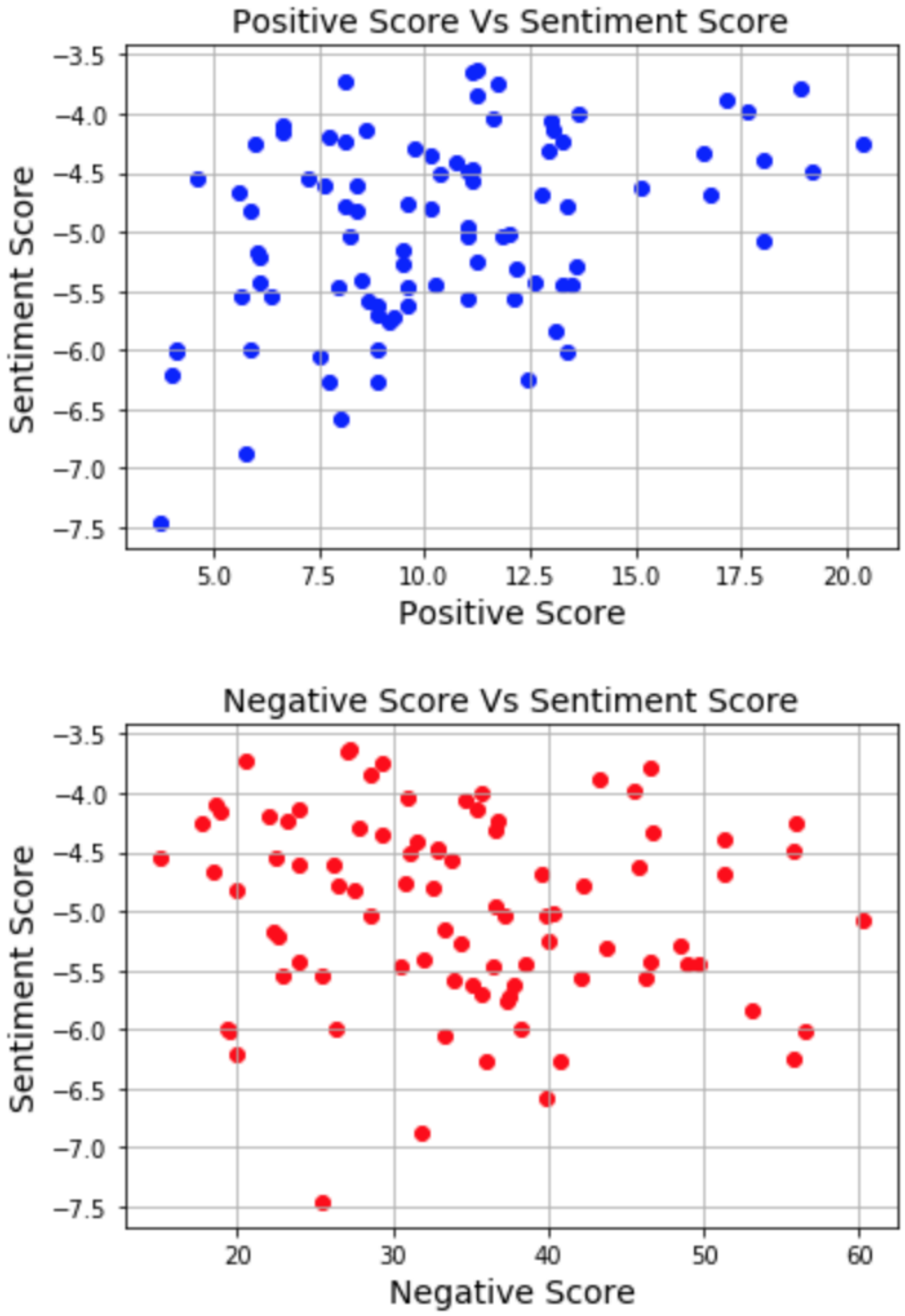

Figure 4.17: Relationship between SumPS, SumNS and Sentiment Score in the experiment C(3 month)

Figure 4.18 shows sentiment score frequency in relation to the number of headlines when training and testing data is first 3 month connected period (daily input) in 2014. The figure shows the x-axis that represents the sentiment score for days in 3 month news

headlines divided in periods with the y-axis that represents the number of score frequency within these periods.

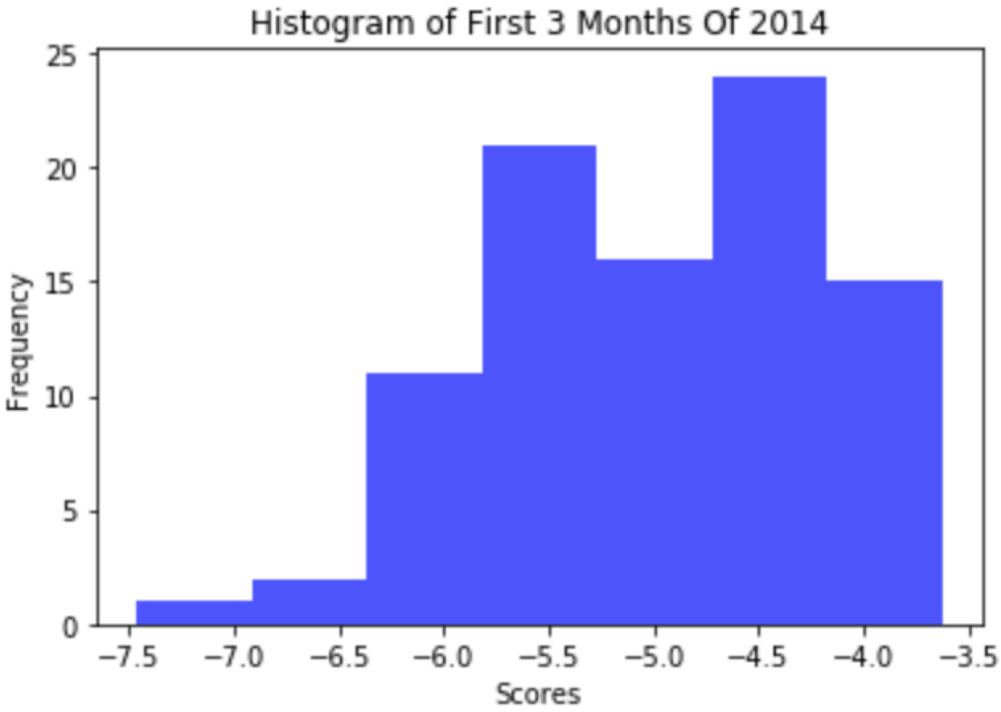

Figure 4.18: Sentiment Score Frequency in the experiment C(3 month)

Figure 4.19 shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict) when training and testing data is first 6 month connected period (daily input) in 2014. The x-axis represents two month of daily data as testing data which is 30% of the 6 month dataset. We notice here the y-axis represents the value of the Sentiment Score for each day where the blue color represents the real output value of the Score, and the orange color represents the value that was measured by the model.

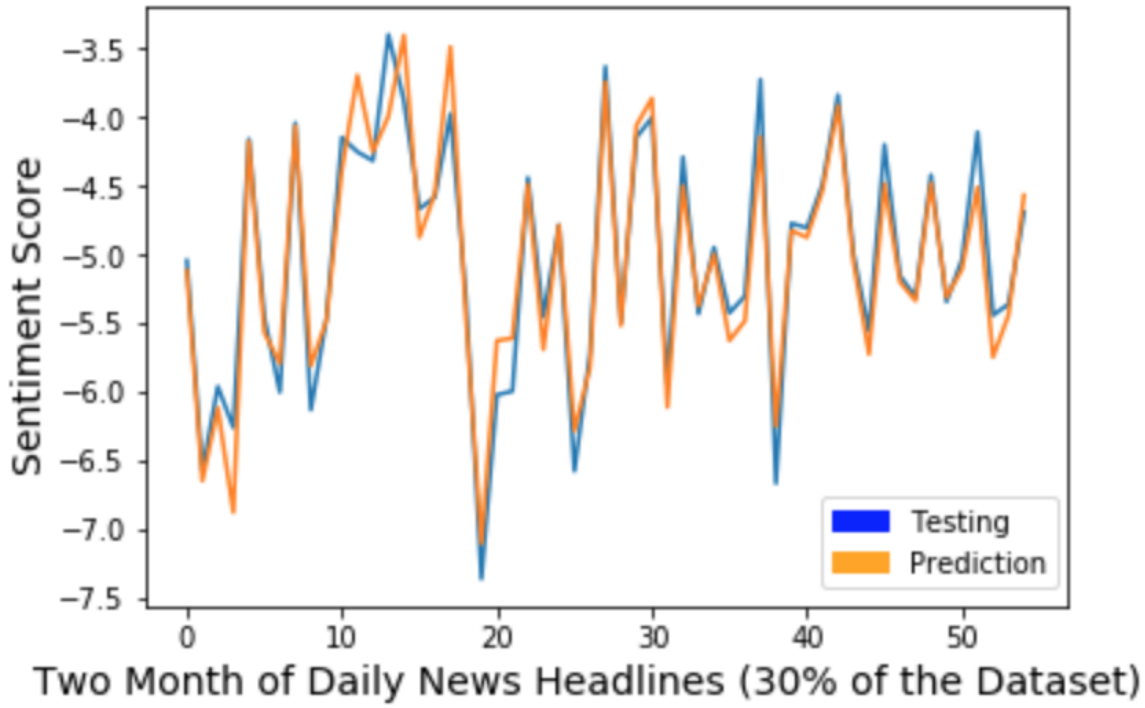

Figure 4.19: The difference between Y_Test and Y_Predict for the experiment C(6 month)

Figure 4.20 shows relationship between the summation of positive Score of headline and sentiment Score per day when training and testing data is first 6 month connected period (daily input) in 2014. Where we note that the more total positive increased sentiment Score for the better, ie less tension. Also, Figure 4.20 shows relationship between the summation of negative Score of headline and sentiment Score per day. Where we note that the less total negative decreased sentiment Score for the worst, ie high tension. The blue color indicates the relationship between sum of positive score for each day in 6 month dataset in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each day in 6 month dataset in the x-axis and the sentiment score for them in the y-axis, where the figure shows the inverse proportions between the two axes.

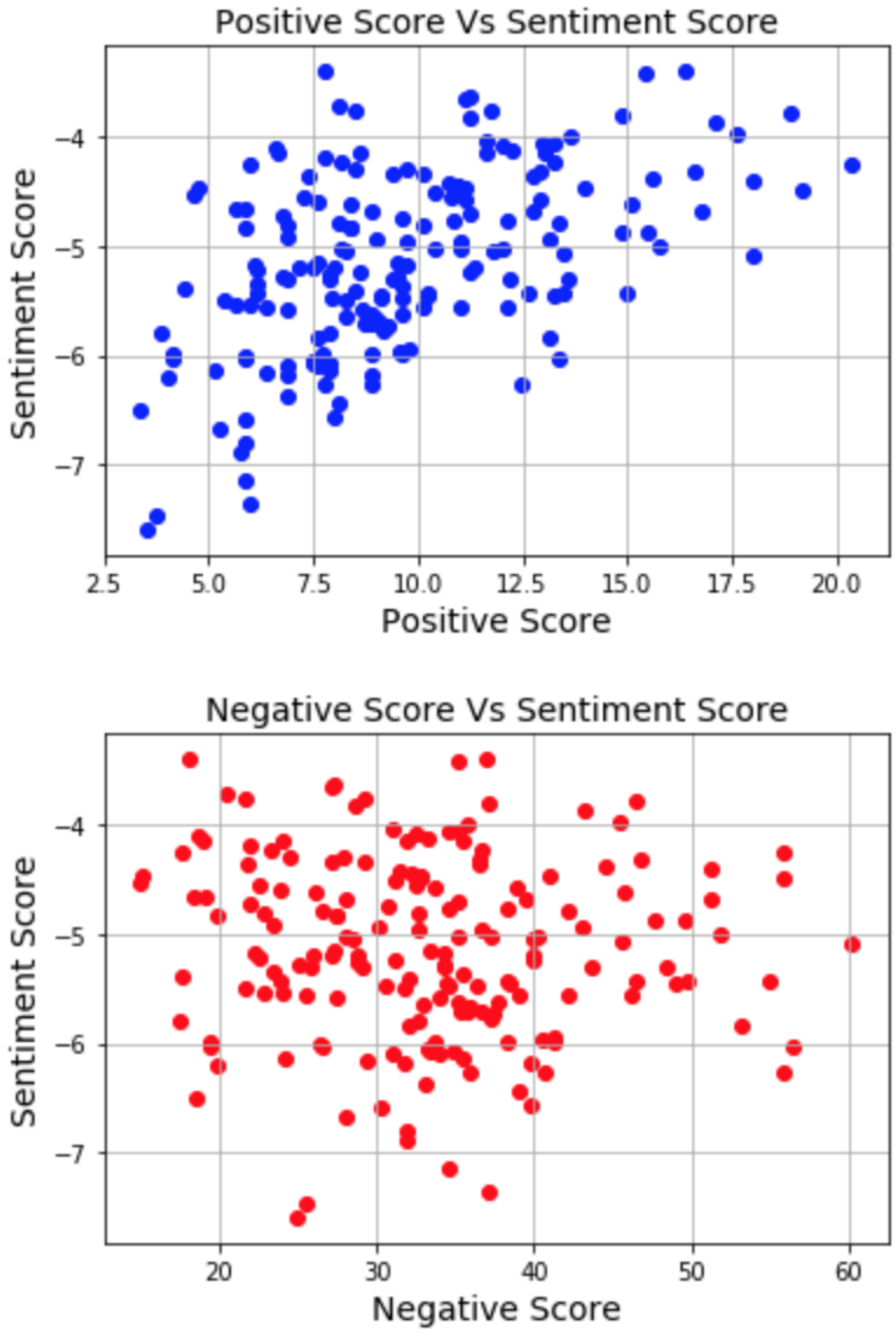

Figure 4.20: Relationship between SumPS, SumNS and Sentiment Score in the experiment C(6 month)

Figure 4.21 shows sentiment score frequency in relation to the number of headlines when training and testing data is first 6 month connected period (daily input) in 2014.

The figure shows the x-axis that represents the sentiment score for days in 6 month news headlines divided in periods with the y-axis that represents the number of score frequency within these periods.

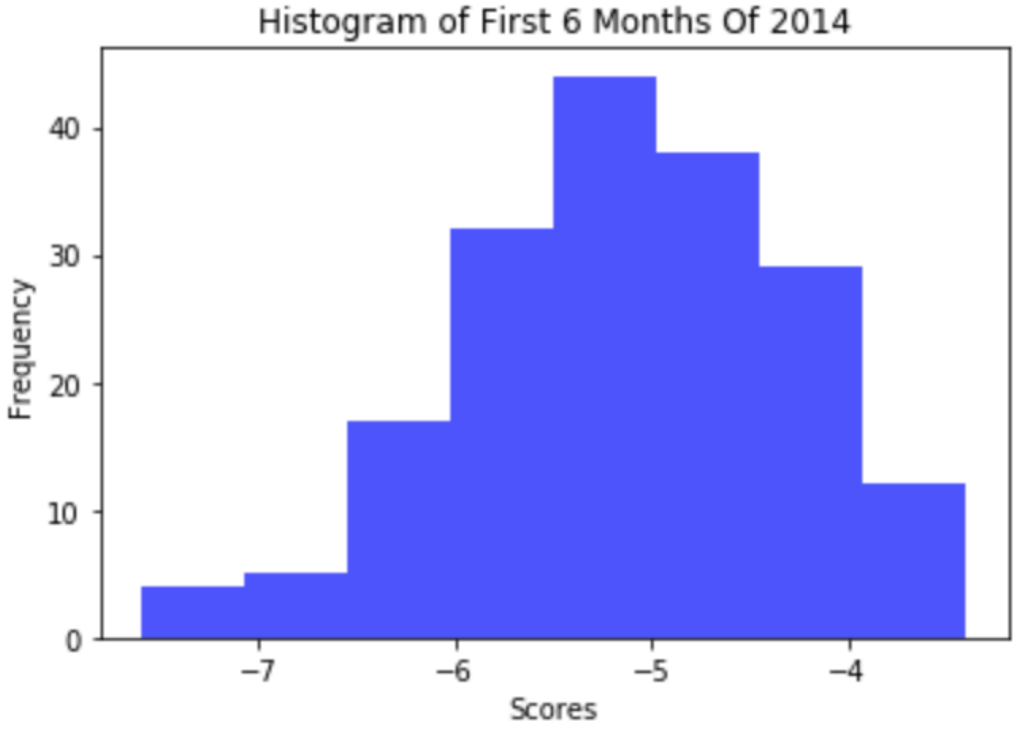

Figure 4.21: Sentiment Score Frequency in the experiment C(6 month)

Figure 4.22 shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict) when training and testing data is first 9 month connected period (daily input) in 2014. The x-axis represents three month of daily data as testing data which is 30% of the 9 month dataset. We notice here the y-axis represents the value of the Sentiment Score for each day where the blue color represents the real output value of the Score, and the orange color represents the value that was measured by the model.

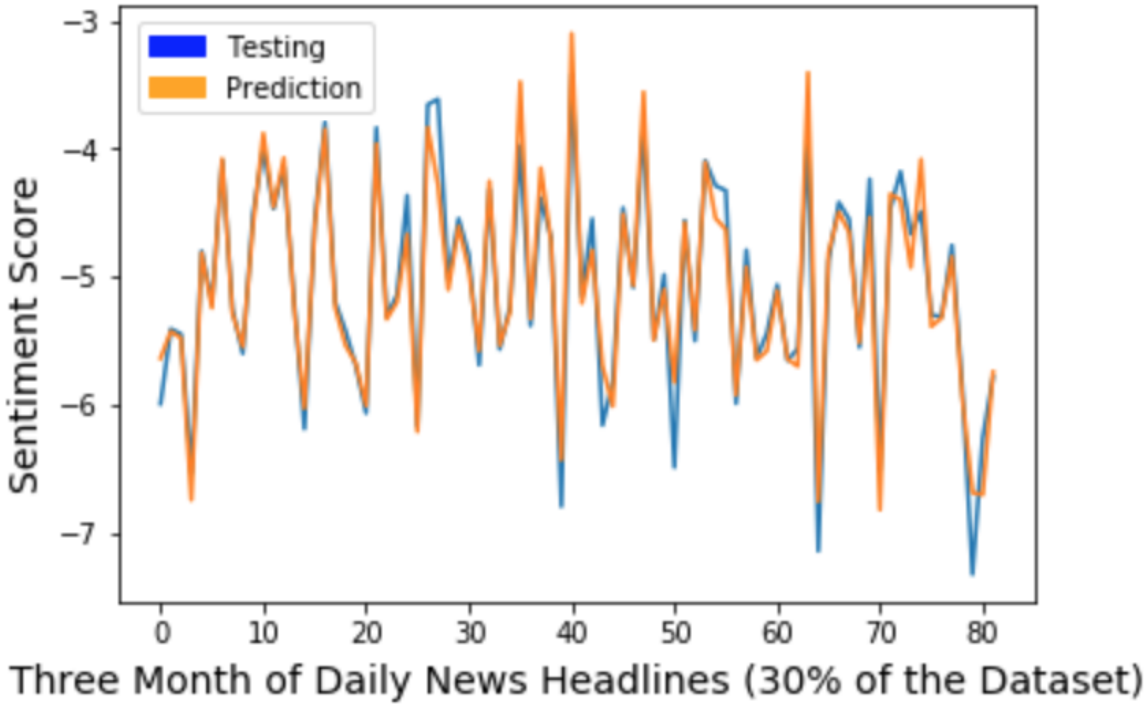

Figure 4.22: The difference between Y_Test and Y_Predict for the experiment C(9 month)

Figure 4.23 shows relationship between the summation of positive Score of headline and sentiment Score per day when training and testing data is first 9 month connected period (daily input) in 2014. Where we note that the more total positive increased sentiment Score for the better, ie less tension. Also, Figure 4.23 shows relationship between the summation of negative Score of headline and sentiment Score per day. Where we note that the less total negative decreased sentiment Score for the worst, ie high tension. The blue color indicates the relationship between sum of positive score for each day in 9 month dataset in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each day in 9 month dataset in the x-axis and the sentiment score for them in the y-axis, where the figure shows the inverse proportions between the two axes.

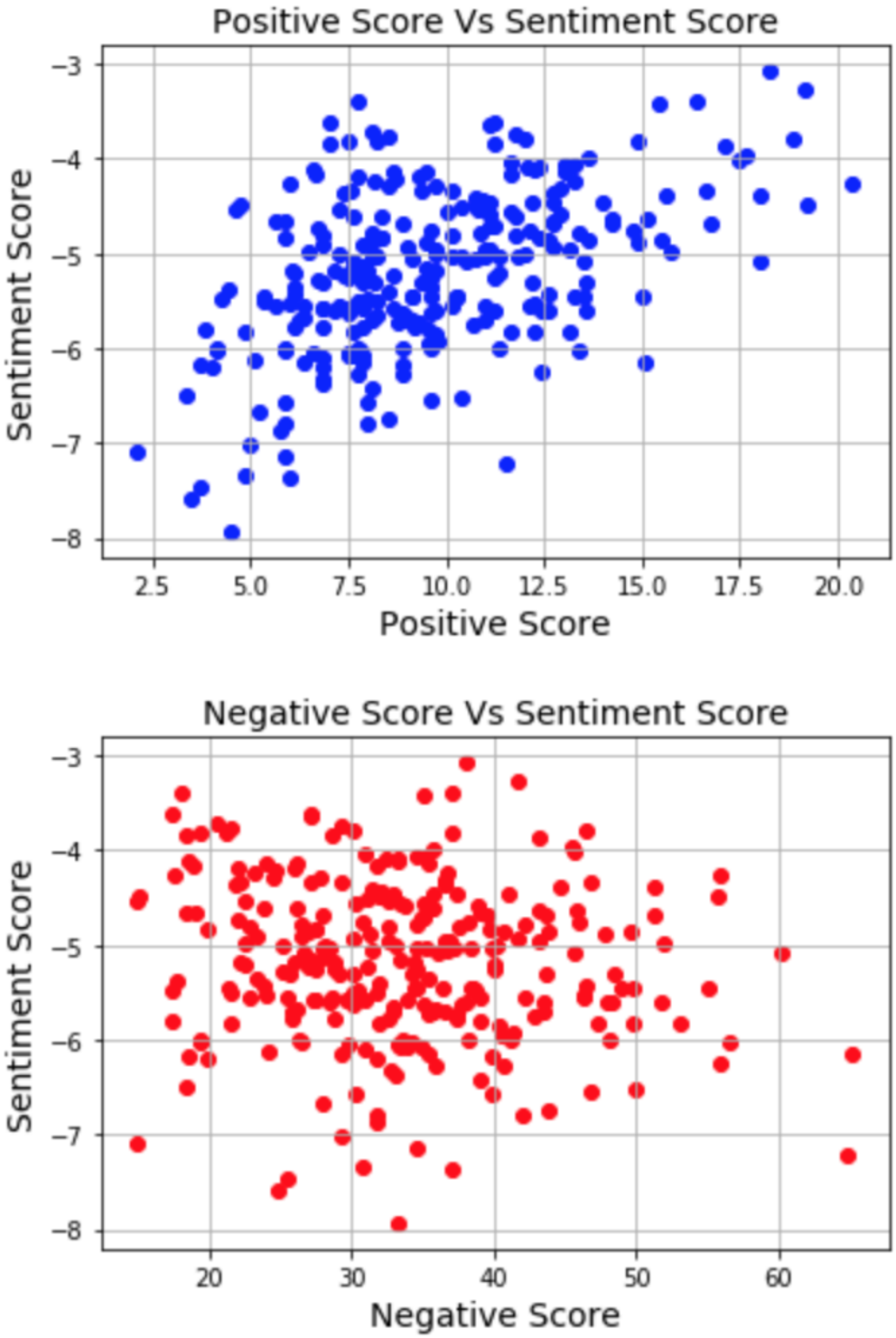

Figure 4.23: Relationship between SumPS, SumNS and Sentiment Score in the experiment C(9 month)

Figure 4.24 shows sentiment score frequency in relation to the number of headlines when training and testing data is first 9 month connected period (daily input) in 2014.

The figure shows the x-axis that represents the sentiment score for days in 9 month news headlines divided in periods with the y-axis that represents the number of score frequency within these periods.

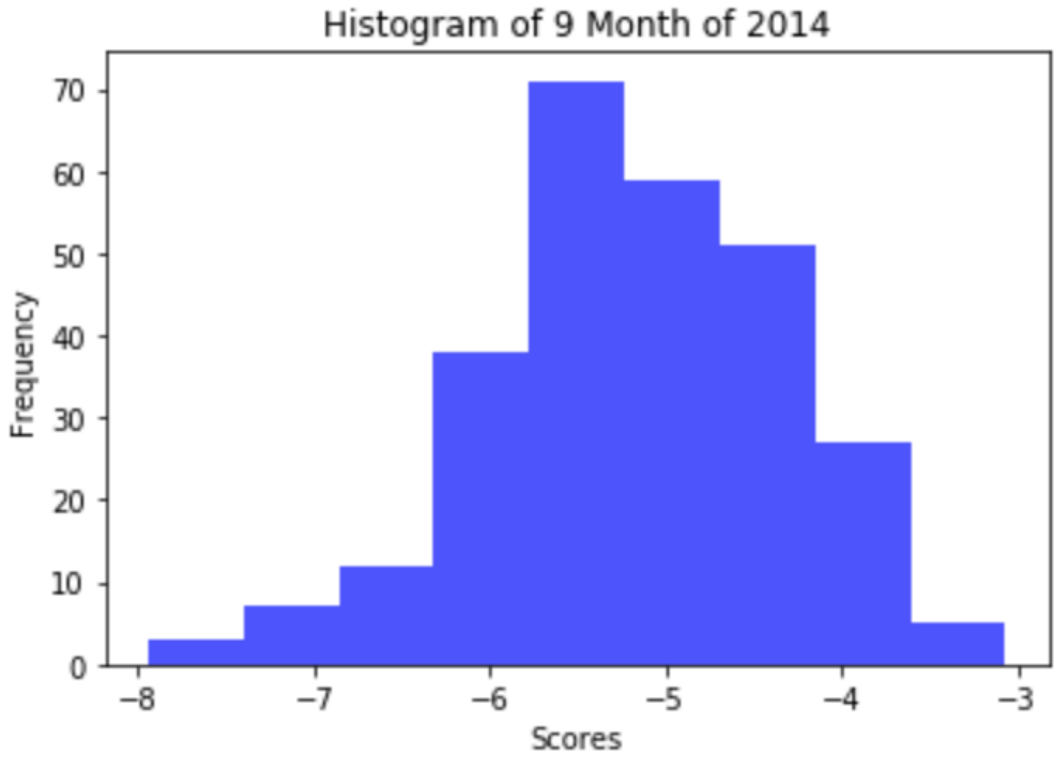

Figure 4.24: Sentiment Score Frequency in the experiment C(9 month)

Figure 4.25 shows the difference between the real output value of the test data (Y_ Test and the measurement value of the test data (Y_ Predict) when training and testing data is all 12 month connected period (daily input) in 2014. The x-axis represents four month of daily data as testing data which is 30% of the 12 month dataset. We notice here the y-axis represents the value of the Sentiment Score for each day where the blue color represents the real output value of the Score, and the orange color represents the value that was measured by the model.

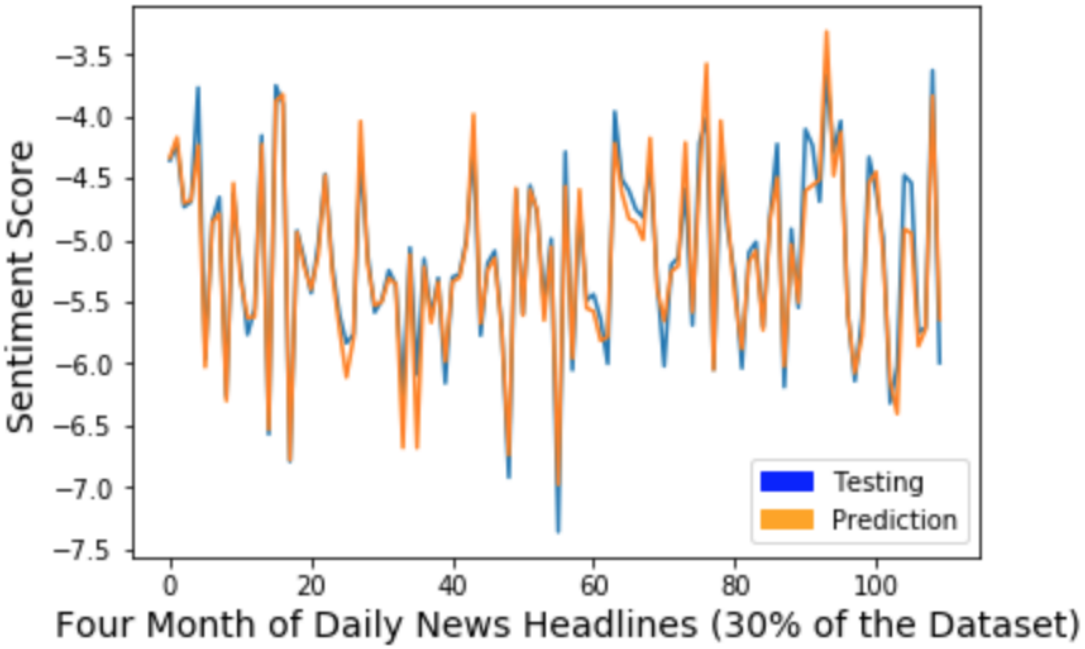

Figure 4.25: The difference between Y_Test and Y_Predict for the experiment C(12 month)

Figure 4.26 shows relationship between the summation of positive Score of headline and sentiment Score per day when training and testing data is first 12 month connected period (daily input) in 2014. Where we note that the more total positive increased sentiment Score for the better, ie less tension. Also, Figure 4.26 shows relationship between the summation of negative Score of headline and sentiment Score per day. Where we note that the less total negative decreased sentiment Score for the worst, ie high tension. The blue color indicates the relationship between sum of positive score for each day in 12 month dataset in the x-axis and the sentiment score for them in the y-axis, where the figure shows the positive proportions between the two axes. The red color indicates the relationship between sum of negative score for each day in 12 month dataset in the x-axis and the sentiment score for them in the y-axis, where the figure shows the inverse proportions between the two axes.

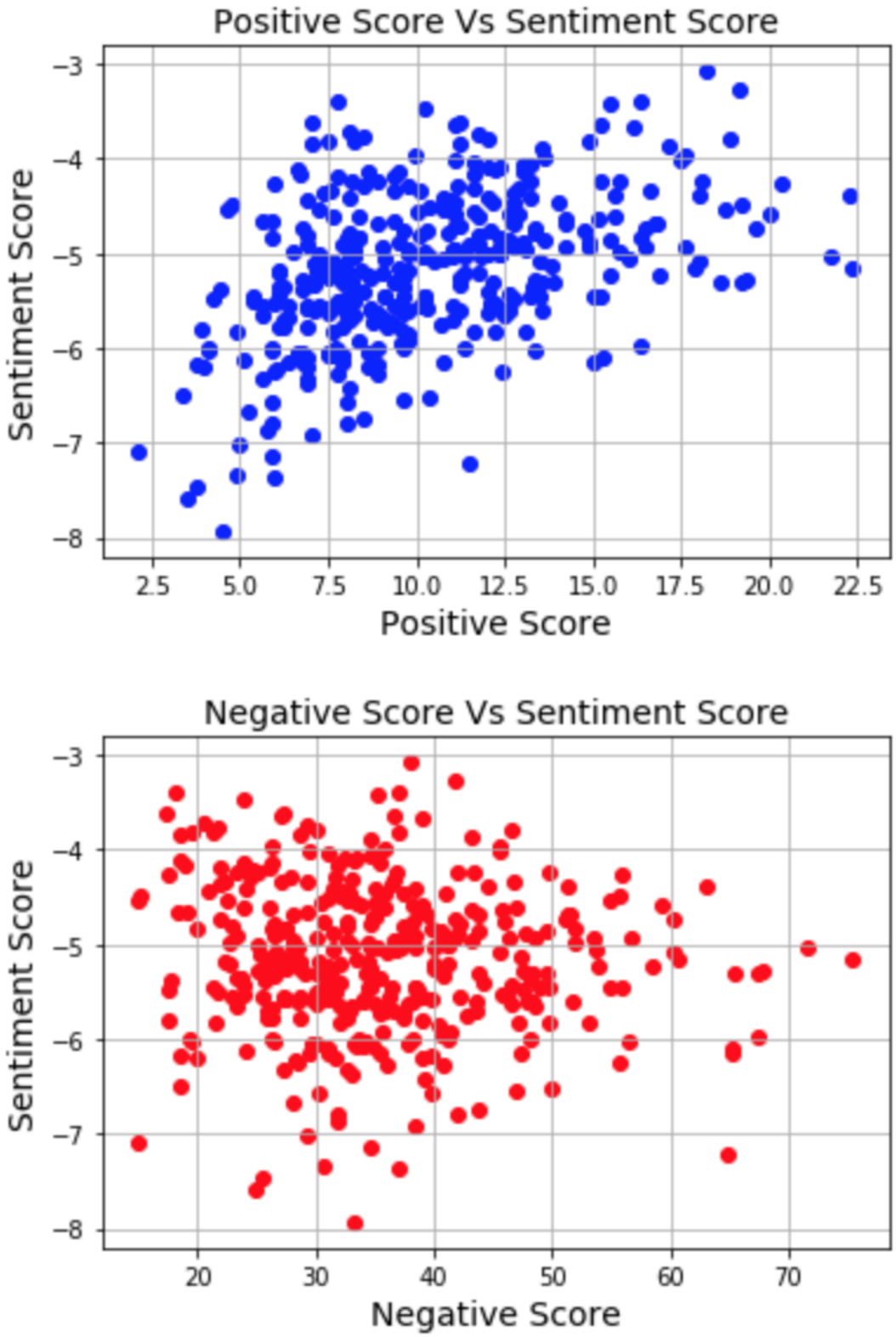

Figure 4.26: Relationship between SumPS, SumNS and Sentiment Score in the experiment C(12 month)

Figure 4.27 shows sentiment score frequency in relation to the number of headlines

when training and testing data is first 12 month connected period (daily input) in 2014. The figure shows the x-axis that represents the sentiment score for days in 12 month news headlines divided in periods with the y-axis that represents the number of score frequency within these periods.

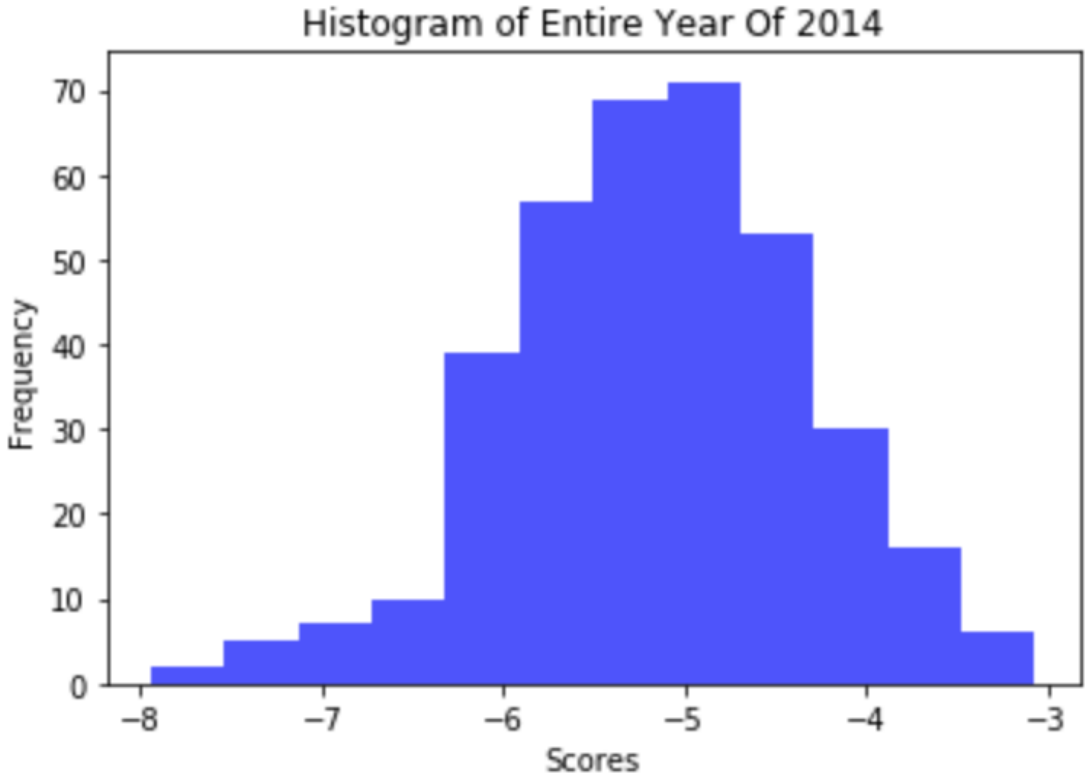

Figure 4.27: Sentiment Score Frequency in the experiment C(12 month)

4.2.2 Discussion

As we noted in the first experiment, the experiment was free of translation, and news headlines in the Arabic language have worked directly without making an English translation. The results were very excellent, but if we look at the results of the data, most of scores go towards zero because there are no values for words in dictionaries and lack of words compared to the Arabic language.

For the second experiment, it was good compared to the first, and the results are high as we noted in the previous figures and tables. We have noted that the number of news headlines taken in this experiment did not affect the results after the 5000 headlines, as we noted in the previous tables, most of the results are close and indicate good results all.

The third experiment had to be done to add meaning to the model, as training the model on random addresses does not give sufficient meaning to the tension.

The third experiment, which we conducted in 4 stages. It was the best as it added the element of time on the model, where we trained the model to know the tension on a particular day, and from it we can know the week, month and year. We got the highest results when the training data was a full year. The model accuracy is tested with important measures, we have obtained 0.937 Explained Variance Score, 0.94 R^2 Score and 0.04 MSE through a full year training data.

4.3 Real Time Analysis Experiment

We have connected the model specifically the procedure responsible for the prediction process shown in figures 8 and 9 with SQLite3 Database and Flask server to achieve real time prediction. The Website of prediction built using python code in flask, HTML and CSS. Figure 4.28 shows how the data is distributed in the database upon arrival from model. We have four real time analysis charts, as shown in appendix II, and every chart has its own web page, so each function navigate and open one specific chart based on the matching name that has been entered from the URL.

As we have seen in previous experiment c, we have chosen 2014 as a sample of training and testing the model. Since, in this year many events took place, as there were events in Syria and Yemen, in addition to a long war in GAZA. The war on Gaza 2014 military conflict between Israel and the Palestinian resistance movements in a sector that actually began on July 8, 2014 and was called "العصف المأكول" after a wave of violence that erupted with the kidnapping, torture and burning of the child Muhammad Abu Khudair from Shuafat by a group of settlers on July 2, 2014, and the re-arrest of dozens of editors of the "Shalit" deal, followed by widespread protests in Jerusalem and within the Arab 48 as

	id	date	year	month	week	dailyPositiveScores	dailyNegativeScores	dailySentimentScore
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1142	2014-01-01	2014	1	1	5.625	18.375	-4.66055828569502
2	1143	2014-01-02	2014	1	1	8.125	26.5	-4.7909982070175
3	1144	2014-01-03	2014	1	1	8.5	32	-5.4079033458904
4	1145	2014-01-04	2014	1	1	8.138	23.237	-4.23627709131149
5	1146	2014-01-05	2014	1	1	8.875	35.125	-5.63270751466106
6	1147	2014-01-06	2014	1	2	4.125	19.5	-6.02059991327962
7	1148	2014-01-07	2014	1	2	16.75	51.25	-4.68887937391979
8	1149	2014-01-08	2014	1	2	9.625	36.375	-5.46252262610137
9	1150	2014-01-09	2014	1	2	11.819	39.931	-5.041982076429
10	1151	2014-01-10	2014	1	2	13.25	36.75	-4.23102091620678
11	1152	2014-01-11	2014	1	2	6.125	23.875	-5.42978220737215
12	1153	2014-01-12	2014	1	2	20.375	55.875	-4.25015286264959
13	1154	2014-01-13	2014	1	3	13.584	48.416	-5.29990932682717
14	1155	2014-01-14	2014	1	3	11	37.25	-5.03450193442012
15	1156	2014-01-15	2014	1	3	7.75	36	-6.26193671044682
16	1157	2014-01-16	2014	1	3	12.444	55.806	-6.25865705668378
17	1158	2014-01-17	2014	1	3	10.25	38.5	-5.45444573179079
18	1159	2014-01-18	2014	1	3	5.75	31.875	-6.87561988666789
19	1160	2014-01-19	2014	1	3	3.75	25.5	-7.46552264311941
20	1161	2014-01-20	2014	1	4	18	51.25	-4.39332693830263
21	1162	2014-01-21	2014	1	4	16.625	46.75	-4.32844250256329
22	1163	2014-01-22	2014	1	4	8.667	33.958	-5.58254859436698
23	1164	2014-01-23	2014	1	4	12.625	46.5	-5.42357098676186

Figure 4.28: SQLite3 Database Screenshot of Distribution Data

well as areas of the West Bank, and intensified after an Israeli ran over two Arab workers near Haifa, and the escalation was interrupted by mutual shelling between Israel and the Palestinian resistance in the Gaza Strip. This war included several military operations. So, most of the headlines were of a negative nature.

In the real time experiment, we predicted the same data for the whole of 2014 and the results were as shown in the upcoming figures. Then, we predicted during a continuous period, which is the year 2015 over days, weeks, months and whole year.

All the following figures shows the shape of the chart, which updates itself automatically continuously, then when adding a new data from the Database, it will represent it, and We can pass the pointer on any point to know the score, whether the sum of the positive or negative score or the sentiment Score for that day or week or month or year. Figure 4.29 shows the daily news headline prediction for year 2014. We can pass any point to see the date and sum of the positive or negative score or the sentiment Score for that day.

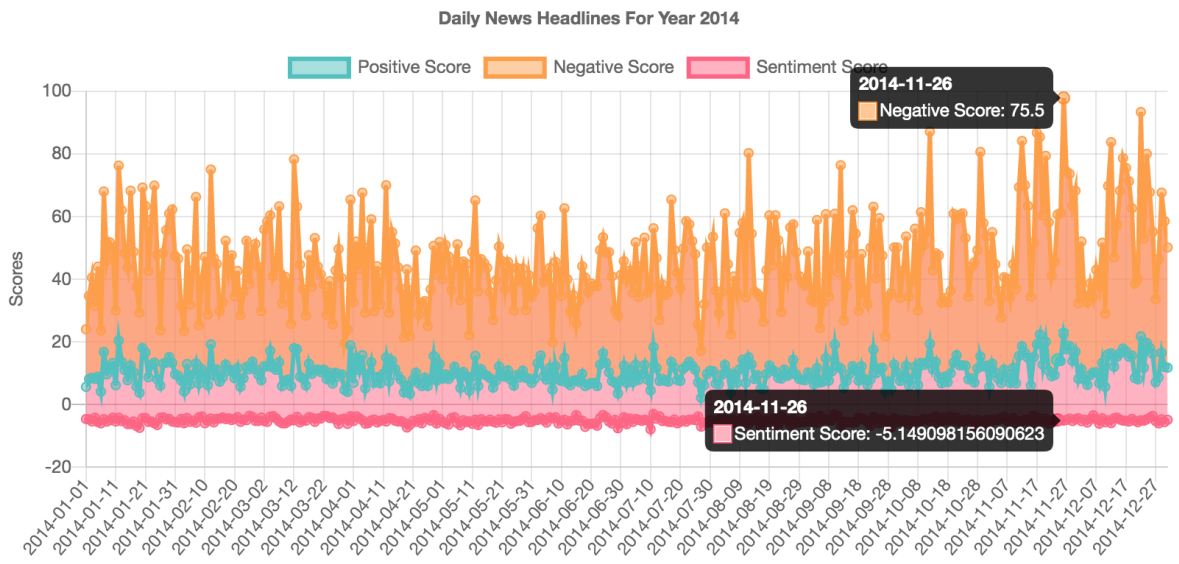

Figure 4.29: Predicted News Headlines per day in year 2014

Figure 4.30 shows the news headline prediction per week for year 2014. As you can see the horizontal axis shows the week number in the year 2014, compared to the three score shown above.

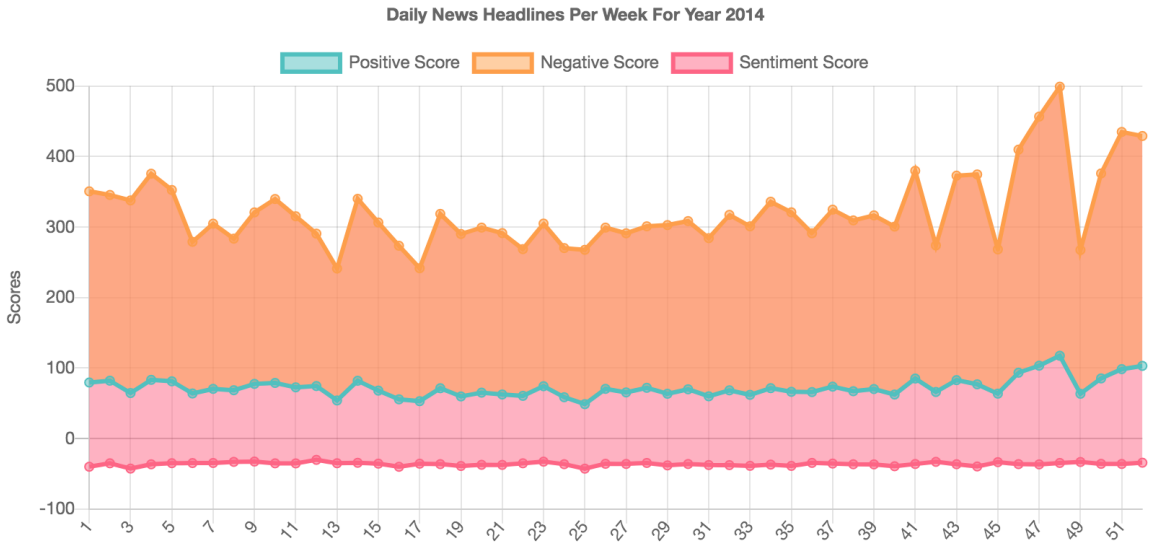

Figure 4.30: Predicted News Headlines per Week in year 2014

Figure 4.31 shows the news headline prediction per month for year 2014. As you can see the horizontal axis shows the name of month in the year 2014.

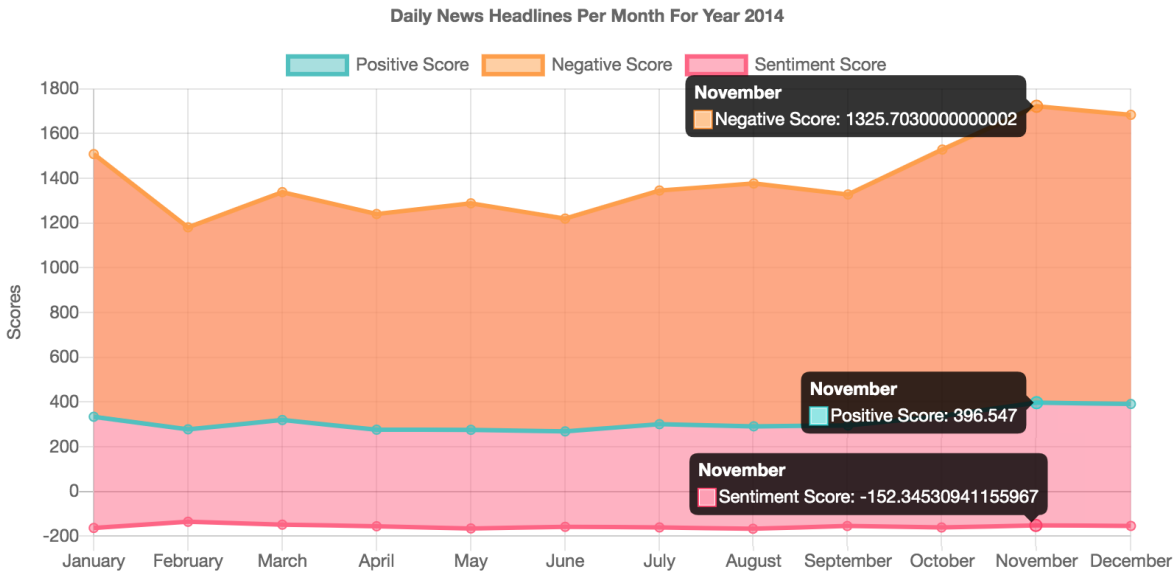

Figure 4.31: Predicted News Headlines per month in year 2014

Figure 4.32 shows the news headline prediction per day for year 2015. As you can see the eighteenth of May appears with a very high negative score, which led to the descent of the Sentiment Score. Figure4.33, 4.34 and show the news headline prediction of year 2015 per week,month.

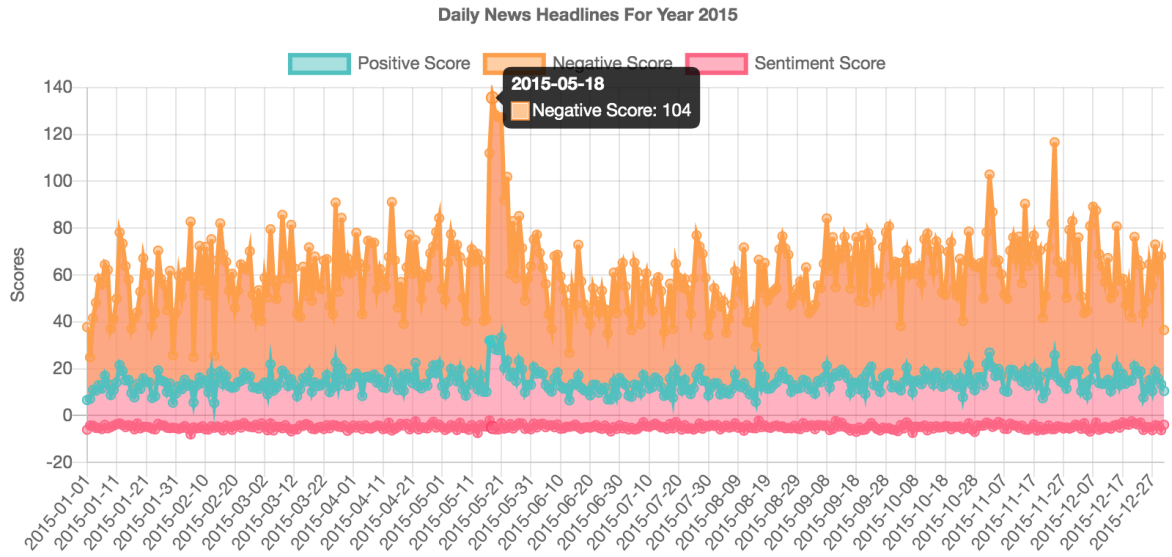

Figure 4.32: Predicted News Headlines per day in year 2015

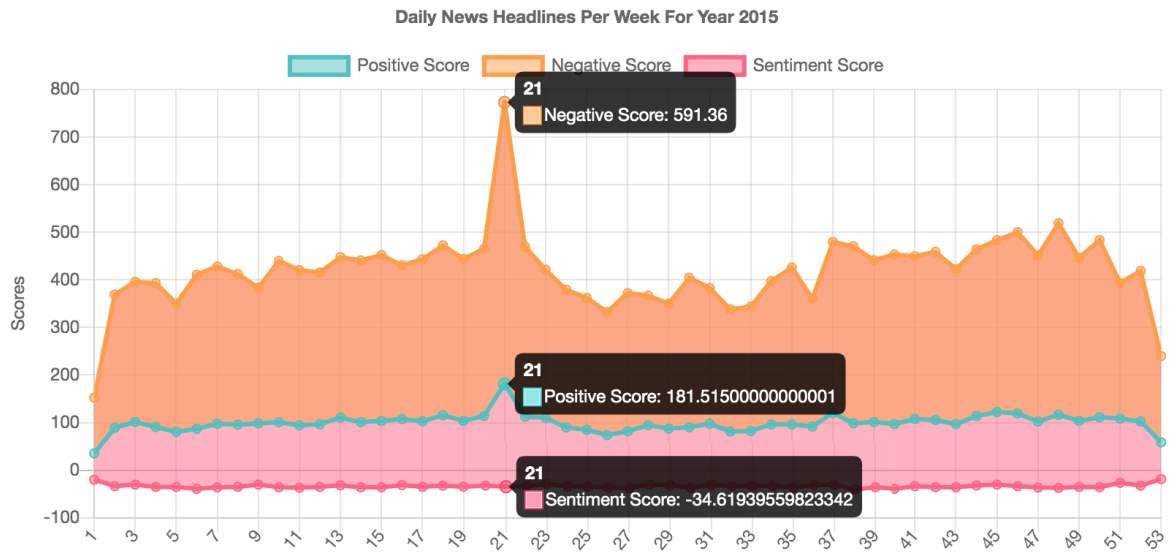

Figure 4.33: Predicted News Headlines per week in year 2015

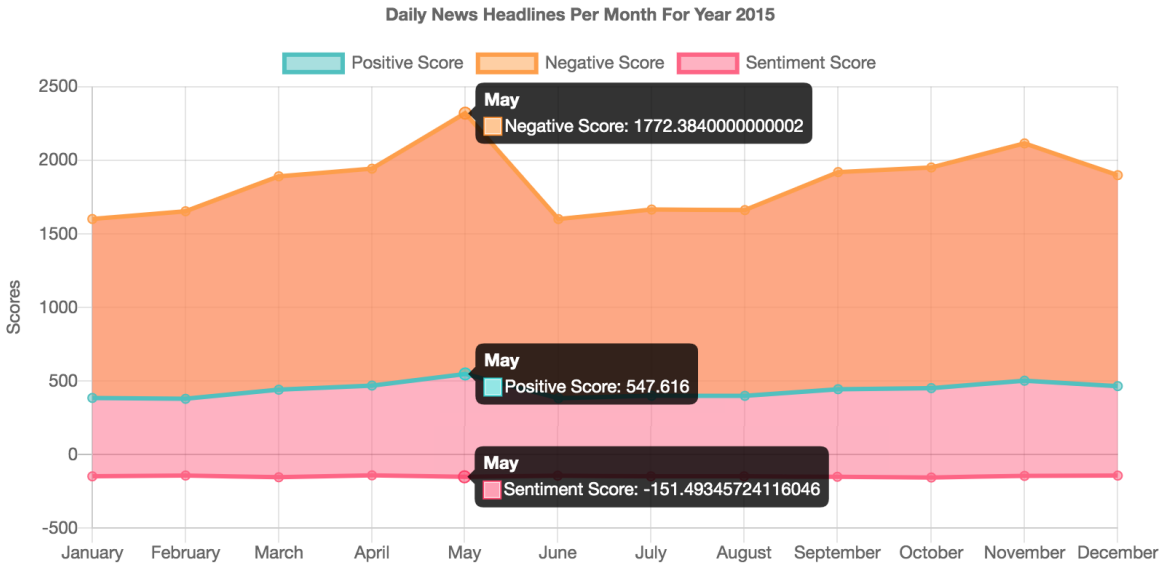

Figure 4.34: Predicted News Headlines per month in year 2015

Chapter 5

Conclusions and Future Works

5. Chapter Outline:

   5.1. Conclusions

   5.2. Future Works

5.1 Conclusions

In this research, we have proposed a customized model for sentiment evaluation to measure tensions level (using negative and positive scores) for every day on Middle East news headlines in the Arabic media. Here, we summarize the most important contribution of this master's thesis.

- The data are collected from Arabic media websites like Aljazeera, then the required pre-processing steps are applied. Steps such as stop words and punctuation marks removal. We have designed an automated news gathering tool from any RSS feed for any news website.
- The data were processed and revised by several important tools in Python.
- We have used Google Cloud Translation API in an innovative way to translate headlines automatically.
- We devised a method for headline labeling to give a score for each one, the Decibel formula is used as a quality measure (sentiment score) for every headline, based on two main lixicons, namely WordNet and SentiWordNet.
- We have trained a multiple linear regression model based on two important entries for every day, the sum of positive scores and the sum of negative scores on that day, so that the model will predict the sentiment score for that particular day.
- We have connected the model with SQLite3 Database and Flask server to achieve real time prediction. The Website of prediction built using python code in flask, HTML and CSS.
- We did three main experiments and each experiment was implemented with different settings. The best experiment was to train a multiple linear regression model based on two important entries for every day. We have tested the model with important measures, we have obtained 0.937 Explained Variance Score, 0.94 R^2 Score and 0.04 MSE through a full year training data.

5.2 Future works

We build a tool using google script app to collect headline of news from any RSS of news websites. This tool not only collects headlines and their dates but also the category and description for each news headline. We add to the context that this tool automatically collects in the form of real time and archiving and filtering through timers operating during the day. We tried this tool on Al Jazeera site RSS feed[1]where the tool collected data from the beginning of September a year ago and is still working so far. We built this tool to make future forecasts in the Middle East and apply them to the current model. In addition to the development of research and open the door for new ideas to develop on this research.

What we have mentioned earlier opens the way for future work in the field of natural languages and analysis of feelings in the Arabic language in order to scarce research in this area. For future work, we plan to expand our research to build corpus and polarity lexicon in arabic language. There are many ideas that can be achieved also after this work, the most important of which we can think of a model that can predict the future through the information of the previous days where we can predict a day, week, or month to come.

[1] `https://www.aljazeera.net/aljazeerarss/a7c186be-1baa-4bd4-9d80-a84db769f779/73d0e1b4-532f-45ef-b135-bfdff8b8cab9`

Bibliography

[1] About python — python.org. https://www.python.org/about/. (Accessed on 10/18/2018).

[2] Anaconda — the world's most popular data science platform. https://www.anaconda.com/. (Accessed on 10/17/2018).

[3] Arabic-news/titles_date at master motazsaad/arabic-news. https://github.com/motazsaad/Arabic-News/tree/master/titles_date. (Accessed on 01/01/2019).

[4] Cloud translation — google cloud. https://cloud.google.com/translate/. (Accessed on 10/19/2018).

[5] Github - motazsaad/arabic-news: Arabic news. https://github.com/motazsaad/Arabic-News. (Accessed on 01/01/2019).

[6] Google translate. https://translate.google.com/. (Accessed on 10/20/2018).

[7] Model evaluation: quantifying the quality of predictions scikit-learn 0.21.3 documentation. https://scikit-learn.org/stable/modules/model_evaluation.html. (Accessed on 01/26/2019).

[8] motazsaad (motaz saad) github. https://github.com/motazsaad. (Accessed on 01/01/2019).

[9] Natural language toolkit nltk 3.4.5 documentation. https://www.nltk.org/. (Accessed on 10/14/2018).

[10] Project jupyter — installing the jupyter software. https://jupyter.org/install. (Accessed on 10/17/2018).

[11] Python 3.0 release — python.org. https://www.python.org/download/releases/3.0/. (Accessed on 02/14/2020).

[12] scikit-learn: machine learning in python scikit-learn 0.21.3 documentation. https://scikit-learn.org/stable/. (Accessed on 11/12/2018).

[13] Text learning group. http://sentiwordnet.isti.cnr.it/. (Accessed on 10/18/2018).

[14] Wordnet — a lexical database for english. https://wordnet.princeton.edu/. (Accessed on 10/18/2018).

[15] Ahmed Abbasi, Hsinchun Chen, and Arab Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–34, 2008.

[16] Muhammad Abdul-Mageed and Mona T Diab. Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In *LREC*, volume 515, pages 3907–3914, 2012.

[17] Yousif Almas and Khurshid Ahmad. A note on extracting sentiments in financial news in english, arabic & urdu. In *The Second Workshop on Computational Approaches to Arabic Script-based Languages*, pages 1–12, 2007.

[18] Reinald Kim Amplayo and Min Song. An adaptable fine-grained sentiment analysis for summarization of multiple short online reviews. *Data & Knowledge Engineering*, 110:54–67, 2017.

[19] Taiwo Oladipupo Ayodele. Introduction to machine learning. *New Advances in Machine Learning*, pages 1–9, 2010.

[20] Pimwadee Chaovalit and Lina Zhou. Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th annual Hawaii international conference on system sciences*, pages 112c–112c. IEEE, 2005.

[21] D Chopra et al. Sentiment analysis of news headlines using naave bayes classifier. *Council For Research And Development Enterprise*, 2015.

[22] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[23] Xiaowen Ding and Bing Liu. The utility of linguistic rules in opinion mining. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 811–812. ACM, 2007.

[24] Rehab M Duwairi, Raed Marji, Narmeen Sha'ban, and Sally Rushaidat. Sentiment analysis in arabic tweets. In *2014 5th International Conference on Information and Communication Systems (ICICS)*, pages 1–6. IEEE, 2014.

[25] Samhaa R El-Beltagy and Ahmed Ali. Open issues in the sentiment analysis of arabic social media: A case study. In *2013 9th International Conference on Innovations in Information Technology (IIT)*, pages 215–220. IEEE, 2013.

[26] Alaa M El-Halees. Arabic opinion mining using combined classification approach. *Arabic opinion mining using combined classification approach*, 2011.

[27] Mohamed Elarnaoty, Samir AbdelRahman, and Aly Fahmy. A machine learning approach for opinion holder extraction in arabic language. *arXiv preprint arXiv:1206.1011*, 2012.

[28] Mohamed Elhawary and Mohamed Elfeky. Mining arabic business reviews. In *2010 ieee international conference on data mining workshops*, pages 1108–1113. IEEE, 2010.

[29] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, volume 6, pages 417–422. Citeseer, 2006.

[30] Noura Farra, Elie Challita, Rawad Abou Assi, and Hazem Hajj. Sentence-level and document-level sentiment mining for arabic texts. In *2010 IEEE international conference on data mining workshops*, pages 1114–1119. IEEE, 2010.

[31] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. Pulse: Mining customer opinions from free text. In *international symposium on intelligent data analysis*, pages 121–132. Springer, 2005.

[32] Anindya Ghose, Panagiotis Ipeirotis, and Arun Sundararajan. Opinion mining using econometrics: A case study on reputation systems. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 416–423, 2007.

[33] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12):2009, 2009.

[34] Alaa El-Dine Ali Hamouda and Fatma El-zahraa El-taher. Sentiment analyzer for arabic comments system. *Int. J. Adv. Comput. Sci. Appl*, 4(3), 2013.

[35] Nitin Hardeniya, Jacob Perkins, Deepti Chopra, Nisheeth Joshi, and Iti Mathur. *Natural Language Processing: Python and NLTK*. Packt Publishing Ltd, 2016.

[36] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[37] Minqing Hu and Bing Liu. Opinion extraction and summarization on the web. In *AAAI*, volume 7, pages 1621–1624, 2006.

[38] Akhil Kadiyala and Ashok Kumar. Applications of python to evaluate environmental data science problems. *Environmental Progress & Sustainable Energy*, 36(6):1580–1586, 2017.

[39] Joshi Kalyani, Prof Bharathi, Prof Jyothi, et al. Stock trend prediction using news sentiment analysis. *arXiv preprint arXiv:1607.01958*, 2016.

[40] Khalid Khalifa and Nazlia Omar. A hybrid method using lexicon-based approach and naive bayes classifier for arabic opinion question answering. *JCS*, 10(10):1961–1968, 2014.

[41] Rawan T Khasawneh, Heider A Wahsheh, Mohammed N Al-Kabi, and Izzat M Alsmadi. Sentiment analysis of arabic social media content: a comparative study. In *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*, pages 101–106. IEEE, 2013.

[42] Nabil Khoufi and Manel Boudokhane. Statistical-based system for morphological annotation of arabic texts. In *Proceedings of the Student Research Workshop associated with RANLP 2013*, pages 100–106, 2013.

[43] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90, 2016.

[44] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. 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*, pages 607–614. ACM, 2007.

[45] Daniel Loureiro, Goreti Marreiros, and José Neves. Sentiment analysis of news titles. In *Portuguese Conference on Artificial Intelligence*, pages 1–14. Springer, 2011.

[46] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.

[47] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[48] Shereen Oraby, Yasser El-Sonbaty, and Mohamad Abou El-Nasr. Exploring the effects of word roots for arabic sentiment analysis. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 471–479, 2013.

[49] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[50] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[51] Ahmed Rafea and Nada A Mostafa. Topic extraction in social media. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pages 94–98. IEEE, 2013.

[52] Òscar Romero Llombart. Using machine learning techniques for sentiment analysis.

[53] Mohammed Rushdi-Saleh, M Teresa Martín-Valdivia, L Alfonso Ureña-López, and José M Perea-Ortega. Oca: Opinion corpus for arabic. *Journal of the American Society for Information Science and Technology*, 62(10):2045–2054, 2011.

[54] R Sathya and Annamma Abraham. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2):34–38, 2013.

[55] SĆ Sonnenburg, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, VojtÄ Franc, et al. The shogun machine learning toolbox. *Journal of Machine Learning Research*, 11(Jun):1799–1802, 2010.

[56] Riya Suchdev, Pallavi Kotkar, Rahul Ravindran, and Sridhar Swamy. Twitter sentiment analysis using machine learning and knowledge-based approach. *International Journal of Computer Applications*, 103(4), 2014.

[57] SM Vohra and JB Teraiya. A comparative study of sentiment analysis techniques. *Journal JIKRCE*, 2(2):313–317, 2013.

[58] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Third IEEE international conference on data mining*, pages 427–434. IEEE, 2003.

Appendix I

```
#Comment: This Appendix includes the data processing, machine learning
and #        real time analysis code implementation.
# Here we are importing the needed Python libraries.
from nltk.tokenize import sent_tokenize, word_tokenize
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet as wn
from nltk.corpus import sentiwordnet as swn
from nltk import sent_tokenize, word_tokenize, pos_tag
from nltk.corpus import stopwords
import pandas as pd
from string import punctuation
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn import datasets
from sklearn.cross_validation import train_test_split
from sklearn.metrics import f1_score
from sklearn.metrics import explained_variance_score
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import confusion_matrix
import matplotlib.pylab as plt
import math
from oauth2client.service_account import ServiceAccountCredentials
from google.cloud import translate
import csv
import sys
from sqlite3 import Error
import sqlite3
import matplotlib.patches as mpatches
%matplotlib inline
#Comment: Define stop word removal for english and arabic languages
with   #     the special characters and punctuation marks,Also reading
the model     #     training and testing dataset and create an Pandas
DataFrame.
englishStopWords = set(stopwords.words('english'))
arabicStopWords = set(stopwords.words('arabic'))
englishStopWords.update(set(punctuation))
arabicStopWords.update(set(punctuation))
dataset = 'titles-dates/aljazeera.net_20190419_date_titles.txt'
df = pd.read_csv(dataset, sep='\t', lineterminator='\n', header=None)
df.columns = ['title', 'Date']
#Comment: This functions responsible for the authentication to Google
#          Cloud API (GCP) service.
def gcpAuthenticate():
```

```
    translateClient = translate.Client.from_service_account_json('GCPKey/NewsSentimentAnalysis-39d7f67c0be9.json')
    return translateClient

#Comment: This function responsible for making Google Cloud Translation
#        API call to translate the news headlines from arabic to english.
def googleCloudTranslateAPICall(word, translateClient):
    translation = translateClient.translate(word,  source_language='ar', target_language='en')
    return translation['translatedText']

#Comment: This function responsible for going over the arabic news
# headlines and translate each headline to english using google cloud API.
def titleTranslations(arabicTitlesDF):
    translateClient = gcpAuthenticate()
    counter = 1
    dataset = []
    try:
        for index, row in arabicTitlesDF.iterrows():
            if (counter % 400) == 0:
                translateClient = gcpAuthenticate()
                print('create new authentication object :: ', counter)

            englishTitle = googleCloudTranslateAPICall(row.title, translateClient)
            counter += 1
            dataset.append([row.title.replace('\n',''), englishTitle, 0, 0, 0, row.Date])


        df = pd.DataFrame(dataset, columns =['ArabicTitle', 'EnglishTitle', 'PosativeScore', 'NegativeScore', 'Sentiment', 'Date'], dtype=float)
        return df
    except Exception as e:
        df = pd.DataFrame(dataset, columns =['ArabicTitle',  'EnglishTitle', 'PosativeScore', 'NegativeScore', 'Sentiment', 'Date'], dtype=float)
        print(str(e))
        return df




#Comment: This function responsible for preparing the dataset for model
#   training and testing by doing the following steps:
#1. Remove stop words from each headline and none alpha text like numbers #   and special characters
#2. Calculate the total positive and negative scores for each news
#   headline
```

```
#3. Compute the sentiment score from total positive and negative scores
#4. Create new data frame using Pandas from the latest dataset with
#   calculated score features.
def prepareTitlesScoresForTraining(arabicTitlesDF):
   try:
        englishTitleScoreDataset = []
        for index, row in arabicTitlesDF.iterrows():
            englishTitle = row.EnglishTitle
            words = word_tokenize(englishTitle)
            filteredWords = [w for w in words if w.isalpha() and not w in englishStopWords]
            titleTotalNegativeScore = 0
            titleTotalPositiveScore = 0
            scoreCountPerTitle = 0
            titleSentimentScore = 0

            for word in filteredWords:
                synsets = wn.synsets(word)
                if not synsets:
                    continue

                selectedSynset = None
                maxNetagiveScore = 0
                for synset_word in synsets:
                    senti_synset = swn.senti_synset(synset_word.name())
                    negativeScore = senti_synset.neg_score()
                    if negativeScore > maxNetagiveScore :
                        maxNetagiveScore = negativeScore
                        selectedSynset = synset_word

                if maxNetagiveScore == 0:
                    selectedSynset = synsets[0]

                swn_synset = swn.senti_synset(selectedSynset.name())
                titleTotalPositiveScore += swn_synset.pos_score()
                titleTotalNegativeScore += swn_synset.neg_score()
                titlePreScore = (1 + titleTotalPositiveScore) / (1 + titleTotalNegativeScore)
                titleSentimentScore = 10 * math.log10(titlePreScore)
                scoreCountPerTitle += 1

            englishTitleScoreDataset.append([row.ArabicTitle.replace('\n',''), englishTitle,
scoreCountPerTitle, titleTotalPositiveScore, titleTotalNegativeScore, titleSentimentScore,
row.Date])

        df = pd.DataFrame(englishTitleScoreDataset, columns =['ArabicTitle', 'EnglishTitle',
'NumberOfWords', 'SumPositiveScores', 'SumOfNegativeScores', 'TitleSentimentScore', 'Date' ],
dtype=float)
        return df
   except Exception as e:
        print(str(e))
```

```
datasetWithSentiment = 'DatasetWithSentimentScoreAndEnglishTitles.txt'
englishTitlesDF = pd.read_csv(datasetWithSentiment, sep='\t', lineterminator='\n')
englishTitlesWithScoreFile = 'englishTitleWithCalculatedScores.txt'
englishTitlesWithScore = pd.read_csv(englishTitlesWithScoreFile, sep='\t')
englishTitlesWithScore['Date']= pd.to_datetime(englishTitlesWithScore['Date'])

#Comment: This function responsible for filtering the news headlines
per  #          time period.
def filterDatasetByTimePeriod(dataframe, start_date, end_date):
    filteredDataset = dataframe[(dataframe['Date'] >= start_date ) & (dataframe['Date'] <=
end_date)]
    return filteredDataset

#Comment: This function responsible for grouping and accumulating the
#          calculated news headlines per each day.
def groupDatasetPerDay(dataframe):
    groupedDataByDay = dataframe.groupby('Date').sum().reset_index()
    groupedDataByDay['TitleSentimentScore'] = (1 + groupedDataByDay['SumPositiveScores']) /
(1 + groupedDataByDay['SumOfNegativeScores'])
    groupedDataByDay['TitleSentimentScore'] =
groupedDataByDay['TitleSentimentScore'].apply(lambda x : 10 * math.log10(x))
    groupedDataByDay['Week_Number'] = groupedDataByDay.Date.dt.week
    return groupedDataByDay

#Comment: The following part is responsible for:-
#         1- Splitting the dataset into training and testing data.
#         2- Fitting the regression model.
#         3- Evaluate the model.
datasetFile = 'sampleDFOf2014English.txt'
sampleDFOf2014English = pd.read_csv(datasetFile, sep='\t')
sampleDFOf2014English['Date']= pd.to_datetime(sampleDFOf2014English['Date'])
News2014EnglishScores = prepareTitlesScoresForTraining(sampleDFOf2014English)
News2014EnglishScores['Date']= pd.to_datetime(News2014EnglishScores['Date'])
filteredByTimePeriod = filterDatasetByTimePeriod(News2014EnglishScores, '2014-01-01','2014-
12-31')
groupedDataPerDay = groupDatasetPerDay(filteredByTimePeriod)
X = np.asarray(groupedDataPerDay[['SumPositiveScores', 'SumOfNegativeScores']])
Y = np.asarray(groupedDataPerDay['TitleSentimentScore'])

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.30, train_size=0.70)
linearRegressionModel = LinearRegression()
linearRegressionModel.fit(X_train, Y_train)
Y_predict = linearRegressionModel.predict(X_test)
print('R^2 The Coefficient of Determination of The Prediction: %.2f' % r2_score(Y_test,
Y_predict))
print('The Coefficient: ', linearRegressionModel.coef_)
print('Intercept_: ', linearRegressionModel.intercept_)
print('Explained Variance Regression Score: ', explained_variance_score(Y_test, Y_predict))
print("Mean Squared Error: %.2f" % mean_squared_error(Y_test, Y_predict))
```

```
#Comment: This function is responsible for reading any arabic news
#  headline dataset and creating new Dataframe object with the right
#  columns.
def readNewsData(dataFile):
    newsDF = pd.read_csv(dataFile, sep='\t', lineterminator='\n', header=None)
    newsDF.columns = ['title', 'Date']
    return newsDF

#Comment: This function is responsible for creating a new SQlite DB
#         connection.
def create_connection():
    db_file = 'flask/db/NewsSentimentAnalysis.db'
    conn = None
    try:
        conn = sqlite3.connect(db_file)
    except Error as e:
        print(e)
    return conn

#Comment: This function is responsible for insert news headlines
#         into the SQlite database.
def insertDailyRecord(row):
    try:
        conn = create_connection()
        cur = conn.cursor()
        month = int(str(row.Date).split('-')[1])
        year = int(str(row.Date).split('-')[0])
        NewsTitle = (row.Date.strftime('%Y-%m-%d'), year, month, row.Week_Number,
row.SumPositiveScores, row.SumOfNegativeScores, row.TitleSentimentScore)
        sql = ''' INSERT INTO dailyanalysis (date, year, month, week, dailyPositiveScores,
dailyNegativeScores, dailySentimentScore)
              VALUES(?,?,?,?,?,?,?) '''

        cur.execute(sql, NewsTitle)
        conn.commit()
        return cur.lastrowid
    except Error as e:
        print(e)

#Comment: This function is responsible for taking the real time
translated #    news headline and calculate their positive and negative
scores before     #    sending them to the prediction.
def prepareTitlesScoresForPrediction(trasnlatedSample):
    try:
          englishTitleScoreDataset = []
          for index, row in trasnlatedSample.iterrows():
              englishTitle = row.EnglishTitle
              words = word_tokenize(englishTitle)
              filteredWords = [w for w in words if w.isalpha() and not w in englishStopWords]
              SumPositiveScores = 0
```

```
            SumNegativeScores = 0
            scoreCountPerTitle = 0
            titleSentimentScore = 0

            for word in filteredWords:
                synsets = wn.synsets(word)
                if not synsets:
                    continue

                selectedSynset = None
                maxNetagiveScore = 0
                for synset_word in synsets:
                    senti_synset = swn.senti_synset(synset_word.name())
                    negativeScore = senti_synset.neg_score()
                    if negativeScore > maxNetagiveScore :
                        maxNetagiveScore = negativeScore
                        selectedSynset = synset_word

                if maxNetagiveScore == 0:
                    selectedSynset = synsets[0]

                swn_synset = swn.senti_synset(selectedSynset.name())
                SumPositiveScores += swn_synset.pos_score()
                SumNegativeScores += swn_synset.neg_score()
                scoreCountPerTitle += 1

            englishTitleScoreDataset.append([row.ArabicTitle.replace('\n',''), englishTitle,
SumPositiveScores, SumNegativeScores, 0, row.Date])

        beforePredictDF = pd.DataFrame(englishTitleScoreDataset, columns =['ArabicTitle',
'EnglishTitle', 'SumPositiveScores', 'SumOfNegativeScores', 'TitleSentimentScore', 'Date'],
dtype=float)
        return beforePredictDF
    except Exception as e:
        print(str(e))

#Comment: This function is responsible for sending the real time
predicted  #        days to the DB by calling the insert function.
def sendPredictedNewsToDB(predictedDataset):
    try:
        for index, row in predictedDataset.iterrows():
                insertDailyRecord(row)
    except Exception as e:
        print(str(e))

#Comment: This function is responsible for grouping the real time news
#       per each day.
def groupDatasetPerDayForPrediction(dataframe):
    dataframe['Date']= pd.to_datetime(dataframe['Date'])
    groupedDF = dataframe.groupby('Date').sum().reset_index()
```

```
    groupedDF['Week_Number'] = groupedDF.Date.dt.week
    return groupedDF

#Comment: This function is responsible for predicting the daily
sentiment  #        score.
def predictDatasetPerDay(dataframe):
    X_Scores = np.asarray(dataframe[['SumPositiveScores', 'SumOfNegativeScores']])
    predictedSentimentScore = linearRegressionModel.predict(X_Scores)
    dataframe['TitleSentimentScore'] = predictedSentimentScore
    return dataframe
```

```
#Comment: This function is responsible for running the real time
analytics #     procedure, by taking the translated news headlines and
calling the      #     required above functions to calculate the news
titles scores,         #     group them, predict them, and finally
insert them into database.
def realTimeAnalysis(translatedDF):
    try:
        preparedScoresDF = prepareTitlesScoresForPrediction(translatedDF)
        groupedDataPerDay = groupDatasetPerDayForPrediction(preparedScoresDF)
        predictedDataset = predictDatasetPerDay(groupedDataPerDay)
        sendPredictedNewsToDB(predictedDataset)
    except Exception as e:
            print(str(e))
    return predictedDataset

arabicEuroNews = 'titles-dates/arabic.euronews.com_20190409_date_titles.txt'
arabicCNN = 'titles-dates/arabic.cnn.com_20190419_date_titles.txt'
newsDF = readNewsData(arabicCNN)
newsDF['Date']= pd.to_datetime(newsDF['Date'])
start_date = pd.to_datetime('2015-01-01')
end_date = pd.to_datetime('2015-12-31')
filterDatasetDF = filterDatasetByTimePeriod(newsDF, start_date, end_date)
predictedDF = realTimeAnalysis(translatedDataset)
```

Appendix II

```
#Comment: This appendix file contains the code implementation of real time #        analysis dashboard with charts using Python Flask framework.
# Here we are importing the needed Python libraries.
from flask import Flask, jsonify, request
from flask import render_template
import ast
import sqlite3
from flask import g
from flask import json


#Comment: Creating Flask App and defining the global variables.
app = Flask(__name__)
DATABASE = 'db/NewsSentimentAnalysis.db'
monthDictionary = {1:'January', 2:'February', 3:'March',
4:'April', 5:'May', 6:'June', 7:'July', 8:'August',
9:'September', 10:'October', 11:'November', 12:'December'}

#Comment: This function responsible for creating a SQLite database
#         connection.
def get_db():
    db = getattr(g, '_database', None)
    if db is None:
        db = g._database = sqlite3.connect(DATABASE)
    return db



#Comment: This function responsible for closing the database connection
@app.teardown_appcontext
def close_connection(exception):
    db = getattr(g, '_database', None)
    if db is not None:
```

```
        db.close()


#Comment: This function responsible for displaying the daily news
#        headlines chart accumulated and grouped per each year.
@app.route('/DailyTitlesPerYear')
def DailyTitlesPerYear():
    return render_template('DailyTitlesPerYear.html')


#Comment: This function responsible for displaying the daily news
#        headlines chart accumulated per month for specific year.
@app.route('/DailyTitlesPerMonth')
def DailyTitlesPerMonth():
    return render_template('DailyTitlesPerMonth.html')


#Comment: This function responsible for displaying the daily news
#         headlines chart accumulated and grouped per week for specific
#         year.
@app.route('/DailyTitlesPerWeek')
def DailyTitlesPerWeek():
    return render_template('DailyTitlesPerWeek.html')


#Comment: This function responsible for displaying the daily news
#        headlines chart, by representing the 356 days for specific
year.
@app.route('/DailyTitles')
def DailyTitles():
    return render_template('DailyTitles.html')




#Comment: This function (API) responsible for returning the daily news
#       headlines accumulate and grouped per year in JSON
representation.
@app.route('/getDailyTitlesGroupedPerYear')
def getDailyTitlesGroupedPerYear():
    sql='''SELECT year, sum(dailyPositiveScores) as positive,
sum(dailyNegativeScores) as negative, sum(dailySentimentScore)
as sentiment_score
                    FROM  dailyanalysis
```

```
                    GROUP BY year
                    ORDER BY year'''
    datasetGroupedPerYear = query_db(sql)
    years=[]
    positiveScores=[]
    negativeScores=[]
    sentimentScores=[]
    if(datasetGroupedPerYear):
        for row in datasetGroupedPerYear:
            years.append(row[0])
            positiveScores.append(row[1])
            negativeScores.append(row[2])
            sentimentScores.append(row[3])

    return jsonify(years=years, positive=positiveScores,
negative=negativeScores, sentiment=sentimentScores)
```

```
#Comment: This function (API) responsible for returning the daily news
#         headlines accumulate and grouped per month for specific year
in    #         JSON representation.
@app.route('/getDailyTitlesGroupedPerMonth')
def getDailyTitlesGroupedPerMonth():
    selectedYear = request.args.get('year')
    sql='''SELECT year, month, sum(dailyPositiveScores) as
positive, sum(dailyNegativeScores) as negative,
sum(dailySentimentScore) as sentiment_score
                    FROM  dailyanalysis
                    WHERE year={selectedYear}
                    GROUP BY year, month
```

```
                    ORDER BY month'''.format(selectedYear =
selectedYear)
    datasetGroupedPerMonth = query_db(sql)
    months=[]
    positiveScores=[]
    negativeScores=[]
    sentimentScores=[]
    if(datasetGroupedPerMonth):
        for row in datasetGroupedPerMonth:
            months.append(monthDictionary.get(row[1]))
            positiveScores.append(row[2])
            negativeScores.append(row[3])
            sentimentScores.append(row[4])

    return jsonify(year=selectedYear, months=months,
positive=positiveScores, negative=negativeScores,
sentiment=sentimentScores)
#Comment: This function (API) responsible for returning the daily news
#         headlines accumulate and grouped per week for specific year
#         in JSON representation.
@app.route('/getDailyTitlesGroupedPerWeek')
def getDailyTitlesGroupedPerWeek():
    selectedYear = request.args.get('year')
    sql='''SELECT year, week, sum(dailyPositiveScores) as
positive, sum(dailyNegativeScores) as negative,
sum(dailySentimentScore) as sentiment_score
                    FROM  dailyanalysis
                    WHERE year={selectedYear}
                    GROUP BY year, week
                    ORDER BY week'''.format(selectedYear =
selectedYear)
    datasetGroupedPerWeek = query_db(sql)
    weeks=[]
    positiveScores=[]
    negativeScores=[]
```

```
    sentimentScores=[]
    if(datasetGroupedPerWeek):
        for row in datasetGroupedPerWeek:
            weeks.append(row[1])
            positiveScores.append(row[2])
            negativeScores.append(row[3])
            sentimentScores.append(row[4])

    return jsonify(year=selectedYear, weeks=weeks,
positive=positiveScores, negative=negativeScores,
sentiment=sentimentScores)

#Comment: This function (API) responsible for returning the daily news
#         headlines for specific year in JSON representation.
@app.route('/getDailyTitles')
def getDailyTitles():
    selectedYear = request.args.get('year')
    sql='''SELECT date, dailyPositiveScores as positive,
dailyNegativeScores as negative, dailySentimentScore as
sentiment_score
                    FROM  dailyanalysis
                    WHERE year={selectedYear}
                    ORDER BY date'''.format(selectedYear =
selectedYear)
    dailyTitlesPerYear = query_db(sql)
    days=[]
    positiveScores=[]
    negativeScores=[]
    sentimentScores=[]
    if(dailyTitlesPerYear):
        for row in dailyTitlesPerYear:
            days.append(row[0])
            positiveScores.append(row[1])
            negativeScores.append(row[2])
            sentimentScores.append(row[3])
```

```
    return jsonify(year=selectedYear, days=days,
positive=positiveScores, negative=negativeScores,
sentiment=sentimentScores)



#Comment: This function responsible for executing a database query and
#         returning the result.
def query_db(query, args=(), one=False):
    cur = get_db().execute(query, args)
    result = cur.fetchall()
    cur.close()
    return result
```