http://www.surrey.ac.uk/currentstudents/images/100834_current-students-surrey-logo_small.png

University of Surrey

Faculty of Engineering and Physical Sciences

Department of Computer Science

PhD Thesis

Aspect-based Sentiment Analysis for Arabic Reviews Using Deep Learning

Nada Mohammed Almani

Principal supervisor: Dr. H Lillian Tang

October 2019

# Abstract

Sentiment analysis is a branch of machine learning that concerns about finding and classifying the polarity for given text. Because of the availability of huge amount of opinionated data that need to be analyzed and interpreted, a lot of recent machine learning research is focused on sentiment analysis applications. it gained a lot of interest due to the.

Many sentiment analysis systems are modeled by using different machine learning techniques, but recently, deep learning, by using Artificial Neural Network (ANN) architecture, has showed significant improvements with high tendency to reveal the underlying semantic meaning in the input text. However, the output of these models could not be explained and the efficiency could not be analyzed because ANN models are considered as a black box and the success of these models comes at the cost of interpretability.

The main objective of the presented work is developing Arabic sentiment analysis system that understands semantics in input reviews without using any linguistic resources. The first proposed model is Deep Attention-based Review Level Sentiment Analysis model (DARLSA) that use binary classifier to detect reviews’ polarities. Different scenarios and architectures were examined to test the ability of the proposed model to extract salient words out of the input. The results proved the ability of the proposed model to understand a given review by highlighting the most informative words to the class label. The model detected Arabic natural language linguistic features, such as intensification and negation styles, efficiently. Also, the effect of applying transfer learning technique on the problem of Arabic sentiment analysis is experimented on review level model.

The second proposed architecture is Deep Attention-based Aspect Level Sentiment Analysis model (DAALSA) for classifying reviews polarity with respect to an aspect into three classes, positive, negative and neutral. Different models were proposed to test the effect of using different attention scoring functions on the classification performance. The results distinguished one model with superior performance compared to other proposed models.

To obtain intuitive explanation of the trained models, both models are enhanced with visualization option. The final review representation is a distributed dense vector generated after passing through multi-layers neural network. Heatmap representation is used to visualize the final review representation. In addition, the attention layer’s scoring vector is visualized as well.

# Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful

I thank Allah for granting me the determination, health and guidance to complete this thesis successfully and ask Him that this work will be beneficial to other researchers.

There are several people I want to thank for supporting me in the compilation of this thesis.

First and foremost, my parents… thank you for your unconditional support… you were my inspiration and the one who taught me the value of curiosity and hard working … I am so proud and lucky to have parents like you.

I would like to express my sincere appreciation and gratitude to my supervisor Dr. H. Lillian Tang for her guidance and support to narrow my search and complete the thesis. She directed me to several valuable research areas and was responsive. I thank her for careful reading, constructive and positive criticism my work.

I would like also to thank King Abdulaziz University in Jeddah, Saudi Arabia, especially Joint Supervision Program, for giving me this opportunity to complete my Ph.D. at University of Surrey.

This journey would not have been possible without the support of my little family …

My lovely husband Khalid, I am speechless how grateful I am for your support and encouragements and your valued comments, you literally just completing what my parents started. Thank you for being my best friend.

To my kids… I love you to the moon and back.

**Table of Content**

Abstract I

Acknowledgements II

List of Abbreviations X

Chapter 1: Introduction 1

1.1 Motivation 1

1.2 Research Objectives and Questions 4

1.3 Research Hypothesis 5

1.4 Contributions 5

1.5 Thesis Structure 6

1.6 Summary 6

Chapter 2: Background 8

2.1 Arabic Language 8

2.1.1 Arabic Natural Language Processing (ANLP) Challenges 8

2.1.1.1 Handling Diglossia 9

2.1.1.2 The Lack of Punctuation Rules 9

2.1.1.3 The Lack of Capitalization 9

2.1.1.4 The Problem of Tokenization 9

2.1.1.5 Handling Negation 9

2.2 Sentiment Analysis Systems 9

2.2.1 Sentiment Analysis Levels 10

2.2.3.2 Machine Learning 11

2. 3 Machine Learning Text Classification Techniques 12

2.4. Artificial Neural Network 13

2.4.1 An Artificial Neuron 13

2. 4.2 Network Architecture 14

2.4.3 Activation Functions 15

2.4.4 Network Topologies 16

2.4.4.1 Feed-Forward Artificial Neural Networks 16

2.4.4.2 Recurrent Neural Networks (RNN) 16

2.4.4.3 The Long Short-Term Memory (LSTM) 17

2.4.5 Neural Networks Training & Gradient Descent 19

2.4.5.1 Back-Propagation Algorithm 19

2.4.5.2 Practical Issues When Training ANN 20

2.5 Neural Language Model 22

2.5.1 Learning Word Representation 22

2.5.2 Distributed Representation Advantages 23

2.5.3 Limitations 23

2.6. Deep Learning 24

2.7 Attention Mechanism 25

2.7.1 The Power of Attention Mechanism 26

2.7.2 Attention Model 26

2.8 Summary 29

Chapter 3: Related Works 30

3.1 English Sentiment Analysis 30

3.1.1 Trends in Sentiment Analysis 30

3.1.2 Review-level sentiment analysis 33

3.1.3 Aspect-based sentiment analysis 34

3.2 Arabic Sentiment Analysis 36

3.2.1 Trends in Arabic Sentiment Analysis 36

3.2.2 Review-level Arabic Sentiment Analysis 38

3.2.3 Aspect-based Arabic Sentiment Analysis 38

3.3 Summary 39

Chapter 4: Deep Attention-Based Review Level Sentiment Analysis Model (DARLSA): The Methodology 43

4.1 Review Level Sentiment Analysis System 43

4.1.1. Text Pre-Processing Tool 44

4.1.2. Deep Attention-Based Review-Level Sentiment Analysis Model (DARLSA) 45

4.1.2.1 Input Layer: 45

4.1.2.2 Embedding Layer. 46

4.1.2.3 GRU Layer 46

4.1.2.4 Attention Layer: Soft Attention Layer (SAL) 47

4.1.2.5 Output Layer 48

4.2. Attention-only Review-Level Sentiment Analysis Model (ARLSA) 48

4.3 Discussion 49

4.4 Summary 49

Chapter 5: Deep Attention-Based Aspect-Level Sentiment Analysis Model (DAALSA): The Methodology 51

5.1 Aspect-Based Sentiment Analysis System 51

5.1.1 Problem Formulation 51

5.1.2 Model Details 52

5.2 Deep Attention-Based Aspect-Level Sentiment Analysis Model (DAALSA) 52

5.2.1 Encoding Phase 52

5.2.1.1 Input Layer 53

5.2.1.2 Embedding Layer 53

5.2.1.3 GRU Layer 53

5.2.2 Reduction Phase 53

5.2.2.1. Deep Aspect-Based Sentiment Analysis Model Based on Soft Attention Layer (DSAL) 54

5.2.2.2 Deep Aspect-Based Sentiment Analysis Model Based on Learned Vector Attention Layer (DLVAL) 56

5.2.2.3 Deep Aspect-Based Sentiment Analysis Model Based on Aspect Vector Attention Layer (DAVAL) 57

5.3. Aspect-Based Sentiment Analysis Model Based on Multi-Layer Attention Model 59

5.3.1. Parallel Attention-Only Model (PAO) 60

5.3.1.1 Encoding Phase: 60

5.3.1.2 Reduction phase 60

5.3.1.3 Classification phase 62

5.3.2. Deep Hierarchical Attention-Only Model (DAO) 62

Model Details 63

5.4 Discussion 64

5.5 Summary 64

Chapter 6: Experimental Setup 66

6.1. Deep Attention-Based Review Level Sentiment Analysis Model (DARLSA) Experimental settings 66

6.1.1 Dataset 66

6.1.2 Pre-trained word embeddings 66

6.1.3 Training details 67

6.1.4 Comparison models 67

6.1.4.1 Baseline 67

6.1.4.2. DARLSA 69

6.1.4.3. ARLSA 70

6.1.5 Tools 70

6.2 Deep Attention-Based Aspect-Level Sentiment Analysis Model (DAALSA) Experimental Settings 71

6.2.1 Dataset 71

6.2.2 Training details 72

6.2.3 Tools 72

6.3. Performance Monitoring 72

6.3.1 Evaluation Step: Process Visualization 72

6.3.1.1 Model Visualization: Neural Network Structure- Block Diagrams 73

6.3.1.2 Training Process Information Visualization 73

6.3.2. Qualification Analysis: Feature Visualization 75

6.3.2.1. Review Representation 75

6.3.2.2. Attention score 76

6.4 Summary 78

Chapter 7: Results and Analysis 78

7.1 Deep Attention-Based Review Level Sentiment Analysis Model (DARLSA): The Evaluation and Qualification 79

7.1.1 Evaluation Process Results 79

7.1.2 Qualification Process Results 81

7.1.2.1 Word’s Context Test: Objective 84

7.1.2.2 Handling Negation and Intensification: objective 87

7.1.2.2.1 Using Explicit Negation 88

7.1.2.2.4 Using Implicit Negation 95

7.1.2.3 Error Analysis 95

7.2 Deep Attention-Based Aspect-level sentiment analysis model (DAALSA): The Evaluation and Qualification 99

7.2.1 Evaluation Process Results and Discussion 99

7.2.2 Qualification Results and Discussion 100

7.3 Discussions and Summary 108

8.1 Summary 111

8.2 Future Works 112

Bibliography 114

Appendix 122

Appendix A: Experiments Accuracy Plots 122

Appendix B: DARLSA Case Studies 124

Appendix C: DAALSA Case Studies 129

**List of Figures**

Figure 2. 1: A Comparison Between (a) Simple Biological Neuron (b) Artificial Neuron (Anderson and McNeill, 1992) 14

Figure 2. 2: Artificial Neuron Processing (Hagan et. al, 2002) 15

Figure 2. 3: Activation Functions (Anderson and McNeill, 1992) 16

Figure 2. 4: RNN Architecture (LeCun et. al, 2015) 17

Figure 2. 5: LSTM Gates (Young et. al, 2018) 18

Figure 2. 6: GRU Gates (Young et. al, 2018) 19

Figure 2. 7: Attention Model Architecture 28

Figure 4. 1: High-level overview of the proposed approach 46

Figure 4. 2: DARLSA Model Architecture 47

Figure 4. 1: High-level overview of the proposed approach 46

Figure 4. 2: DARLSA Model Architecture 47

Figure 5. 1: High-Level Overview of The Proposed Approach 54

Figure 5. 2: DSAL Model Architecture 58

Figure 5. 3: DLVAL Model Architecture 59

Figure 5. 4: DAVAL Model Architecture 61

Figure 5. 5: PAO model architecture 64

Figure 5. 6: DAO Model Architecture 66

Figure 6. 1: GRU-based Model. (a) Model Plot (b) Model Summary 71

Figure 6. 2: Different Initialization Cases 72

Figure 6. 3: DARLSA Model. (a) Model Plot, (b) Model Summary 73

Figure 6. 4: ARLSA model. (a) Model plot, (b) Model summary 74

Figure 6. 5: Review Level Sentiment analysis Baseline Model Block Diagram 76

Figure 6. 6: Confusion Matrix 77

Figure 6. 7: GRU layer hidden-representation heatmap.(a) Review heatmap. (b) Negated review heatmap. (c) Representation difference 79

Figure 6. 8: Review representation overtime (a) Baseline GRU-based model. (b) Attention-based DARLSA . 79

Figure 6. 10: Uniformed Scaled Score 81

Figure 6. 11: Html representation for attention scores 81

Figure 7. 1: Models Training Accuracies (a) DARLSA. (b) Baseline 84

Figure 7. 2: Baseline GRU layer hidden-representation heatmap. (a) Review heatmap. (b) Negated review heatmap “gẖyr”. (c) Negated review heatmap “lys”. (d) Representation difference a-b (e) Representation difference a-c 93

Figure 7. 4: Performance compared between all models for both test set and training set. (a) Accuracy. (b) F-measure. 103

**List of Tables**

Table1. 1: Example of Arabic Review 15

Table 3. 1: Summary of Sentiment Analysis Reviewed works 55

Table 7. 1: Evaluation Process Results for Proposed DARLSA Models 93

Table 7. 3: Results For The Positive Case 98

Table 7. 4: Results For The Negative Case 99

Table 7. 5: Results For “sʿyd” Verb Case 100

Table 7. 6: Results For “Sʿyd” Proper Noun Case 100

Table 7. 7: Baseline Results For Negation Test 102

Table 7. 8: Attention-Only Model Results For Negation and Intensification Tests 103

Table 7. 9: Deep Attention-based DARLSA model results for negation test 104

Table 7. 10 : Baseline Results For Intensification Test 106

Table 7. 11: Attention-Based DARLSA Model Results For Intensification Test 106

Table 7. 12: Baseline Model Results For Intensification Test And Irony Style 107

Table 7. 13: Attention-Based DARLSA model results for implicit negation and irony style 108

Table 7. 14: Example 1 Classification Results 109

Table 7. 15: Example 2 Classification Results 110

Table 7. 16: Example 3 Classification Results 111

Table 7. 18: List of DAALSA Test Cases 114

Table 7. 19: Inference Gating Rules 115

Table 7. 20: DLVAL Layer result, Example 3 116

Table 7. 21: DSAL1 Layer, Example 3 117

Table 7. 22: AVAL Layer, Example 3 117

Table 7. 23: LSAL Layer, Example 3 118

Table 7. 25: DAO Model Final Layer Results, Example 1 119

Table 7. 26: DAVAL Model, Example 1 119

Table 7. 27: DSAL Model Results, Example 1 120

Table 7. 28: DLVAL Model Results, Example 1 120

Table 7. 29: DLVAL Model, Example 2 120

Table 7. 30: DLVAL Model, Example 4 121

Table 7. 31: Baseline Model Results 123

# List of Abbreviations

ANLP Arabic Natural Language Processing

ANN Artificial Neural Networks

ABSA Aspect Based Sentiment Analysis

AVAL Aspect Vector Attention Layer

ARLSA Attention-only Review-Level Sentiment Analysis Model

BOW Bag of Word

BNB Bernoulli Naive Bayes

CA Classical Arabic

CRF Conditional Random Fields

CBOW Continuous Bag-of-Words

CNN Convolutional Neural Network

DAVAL Deep Aspect-Based Sentiment Analysis Model Based on Aspect Vector Attention Layer

DLVAL Deep Aspect-Based Sentiment Analysis Model Based on Learned Vector Attention Layer

DSAL Deep Aspect-Based Sentiment Analysis Model Based on Soft Attention Layer

DAALSA Deep Attention-based Aspect Level Sentiment Analysis model

DARLSA Deep Attention-Based Review-Level Sentiment Analysis Model

DBN Deep Belief Net

DA Dialect Arabic

EWGA Entropy Weighted Genetic Algorithm

GRU Gated Recurrent Unit

GAP Global Averaging Pool

GPU Graphical Processing Unit

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

LVAL Learned Vector Attention Layer

LSTM Long Short-Term Memory

ML Machine Learning

MSA Modern Standard Arabic

MLP Multilayer Perceptron

MNB Multinomial Naive Bayes

NB Naive Bayes

NLP Natural Language Processing

NNLM Neural network language model

POS Part-of-Speech

RNN Recurrent Neural Networks

RNTN Recursive Neural Tensor Network

RBM Restricted Boltzmann Machine

SA Sentiment Analysis

SAL Soft Attention Layer

SVM Support Vector Machine

TF Term Frequency

TF-IDF Term Frequency Inverse Document Frequency

# 

# Chapter 1: Introduction

With the existence of the World Wide Web, taking decision regarding any product or services is mainly based on exchanging others opinions. With customers high demands for reviewing others experience, the growth of the reviews collecting resources, such as blogs, forum discussion and social networks, are increasing as well. Back in the early of 2001, the research community enriched with huge amount of published papers regarding sentiment analysis (Farghaly and Shaalan, 2009).  The increased interest in natural language processing (NLP) and machine learning (ML) along with the existence of large amount of dataset to train the model on, led researchers to give high attention to improve sentiment analysis (SA) systems to meet customers’ needs (Pang and Lee, 2008). Due to the importance of sentiment analysis to business, the interest has shifted form computer science to management, economics and to the whole society. Nowadays, almost all of the big companies and organizations are having a sort of voice of the customer channel, such as emails or call centers, and a mean to analyze it. This will help organizations to reshape their services and reengineer their business processes into the best practice ones.

Natural Language Processing (NLP), machine learning (ML) and Computational linguistics techniques are used in the field of sentiment analysis systems. This thesis is focusing on proposing models to classify sentiment contained in Arabic reviews with respect to two different levels, review level, and aspect-based level. In review level, the polarity is evaluated depending on the review text. For aspect-based sentiment analysis level, the text is evaluated with respect to an input aspect. All experiments are done on Arabic datasets by using deep learning mechanism.

# Motivation

Arabic Language is the widely spoken Semitic language with more than 422 million native and non-native speakers, which placed it as one of the six official languages in the united nation[[1]](#footnote-1). Because it is the language of the Qur’an, Muslims’ holy book; this makes it a second language of many Muslims. (Mäntylä et. al, 2018) had published a study in 2018 to evaluate sentiment analysis systems’ history and investigate its current trends. They found out that Chinese and Arabic languages are gaining growing interest in the area of sentiment analysis since 2013. However, by the end of 2016 the interest in sentiment analysis for Arabic language had dominated other languages, as shown in figure 1.1.

Figure 1. 1: Word cloud of sentiment analysis papers published between 2013 and 2016 - (2014-2016) papers on top and (-2013) on bottom (Mäntylä et. al, 2018)



Unlike English and other European Languages, a little work is done for Arabic Natural Language Processing (ANLP) area. Some potential reason is the limited number of reliable linguistics tools and resources. In addition, the morphological complexity nature found in Arabic language adds extra challenges on ANLP (Farghaly and Shaalan, 2009; Hamdi et. al, 2016). Arabic has a rich, complex morphological structure, with no capitalization. Arabic language script uses 28-alphabet letters, connected from right to left. Table 1.1 is an example of Arabic review, where there are three representations, Arabic review, Latin transliteration[[2]](#footnote-2), English translation. This format will be followed throughout the thesis whenever Arabic text is encountered.

|  |  |
| --- | --- |
| Arabic Review | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-14 at 10.01.47 AM.png |
| Latin transliteration | fndq rạỷʿ ạlfndq ạktẖr mn rạỷʿ wkẖṣwṣạ mstwy̱ ạlkẖdmẗ wsrʿẗ tlbyẗ rgẖbạty̱ mn qbl mkạtb ạlạstqbạl wạlrqy̱ fy̱ ạltʿạml mn bdạyẗ ạlạstqbạl fy̱ ạlmṭạr wḥty̱ ạlmgẖạdrẗ. |
| English translation | The hotel is great, the hotel is more than wonderful, especially the level of service and the speed of satisfying my wishes by the reception and handling offices from the arrival at the airport until departure. |
| Table1. 1: Example of Arabic Review | |

The choice of data representation, or features, will have a strong effect on the performance of the classification algorithm (LeCun et. al, 2015; Bengio et. al, 2013a; Deng and Yu, 2014). Sentiment Classification process will benefit more from extracting semantic features that could be learned from unprocessed input without using human intervention. Feature Learning, also known as Representation learning or Deep Learning, is the field of machine learning that discovers tools and techniques to construct an expressive representation and extract useful information of the input data that is going to be held during classifier training. Deep learning having an advantage of learning and discovering important features found in the input data by using multi-layer Artificial Neural Networks (ANN), or level of understanding (LeCun et. al, 2015; Deng and Yu, 2014).

Historically, deep learning approaches were published in the 80’s but did not gain any attention from the researchers as the performance of these approaches were low compared to their competitors. In addition, deep models were difficult to train and needed high performance hardware to get results comparable with SVM in those days (Glorot and Bengio, 2010).

Despite its efficiency, neural deep learning models are hard to understand and interpret. Unlike tradition machine learning classifiers such as SVM, there is no individual assigned weights to the input features as they are working with distributed representation word embeddings. Each layer is output a hidden representation vector and it is not clear how the final output is composed and generated. With the existence of many layers with millions of hyper-parameters, the neural network is locked inside a black box that makes the behavior of the model hard to predict and understood. The researcher will not be able to trust the final decision or to optimize models easily and faster if they do not know the source of the problems (Zupan, 1994; Li et. al, 2016; Karpathy et. al, 2015).

# 1.2 Research Objectives and Questions

The main objective of the presented work was to model deep neural network to classify the sentiment in a given Arabic review. Reviews usually are natural language representations that convey feelings and emotions about a specific topic in any format, such as voice reviews or text reviews. To reach this goal, the following three main research questions are modeled and proposed:

*Research Question 1:* does using Deep Attention-based Review Level Sentiment Analysis model (DARLSA) for classifying Arabic reviews polarity is beneficial in terms of classification performance and selecting salient features? .

* Does adding attention layer after recurrent-based layer will generate a better final review representation?
* Does applying transfer learning technique on the process of initiating review’s words embedding have a positive effect on the classification process?
* Does using attention only layers in the model will be enough to generate a representative final review distributed representation?

*Research Question 2*: Does using Deep Attention-based Aspect Level Sentiment Analysis model (DAALSA) will produce the final Arabic review polarity in correlation to a specific aspect?

1. Does adding attention layer after recurrent-based layer will generate a better final review representation regarding specific aspect?
2. What is the effect of using different logic to calculate the attention weights on the final review distributed representation and thus on the classification accuracy?
3. Does it beneficial in terms of classification accuracy to consider the aspect-distributed representation by concatenating review-hidden representation with aspect representation before attention layer?
4. Does feeding attention layer with the aspect vector to guide the attention scoring function will help to highlight the important words in the review regarding this aspect?
5. Does using hierarchical or parallel attention-based only deep model will capture the salient features in a given review and can give high classification rate?

*Research Question 3:* Does visualizing attention layer weights helps in opening the black-box of the neural network models?

* 1. Does having different visualization technique helps in evaluating and understanding the behaviors of the proposed model?
  2. Does visualization technique helps to have a detailed analysis of an Arabic review and can reveal the most informative words considered by the model?

# 1.3 Research Hypothesis

There are several hypotheses that are claimed by this thesis while proposing models follow these hypotheses. The main hypothesis that are claimed by this research are presented as follows:

*Hypothesis 1:* Using deep artificial neural network model is suitable to capture sentiment orientation for low resources languages, such as Arabic, without relying on any external resources such as using handcrafted features or using sentiment lexicon.

*Hypothesis 2:* All models are sharing general architectures, where all are deep models accepting distributed word representation as input, for both aspects and reviews.

*Hypothesis 3:* Using Attention-based model will generate distributed review representation that focused more on the salient words in a given Arabic review.

*Hypothesis 4:* Using pre-trained Arabic word embedding representation as an input will accelerate the training and give more accurate results.

*Hypothesis 5:* Having a hierarchical model of only attention-based layers will have a positive effect on capturing the most salient features in given Arabic review as it reflects the hierarchical nature found in Arabic sentences.

*Hypothesis 6:*  Using visualization techniques will help in evaluating and analyzing the proposed models’ results, by monitoring the performance of running process and by qualifying the generated review representation.

# 1.4 Contributions

This section will demonstrate the approach followed by this work in to meet research questions and investigate hypotheses results. The ultimate goal is to develop deep models that are computationally efficient and does not rely on external resources. Multiple models are proposed and investigated to find a polarity for an Arabic review either in review level or toward specific given aspect. All proposed models are based on deep models that accept distributed word representation as input, for both aspects and reviews. Studying the effect of transfer learning on the problem at hand will be investigated as well by examining the weights of different pre-trained models. After GRU layer, attention-based layer is added into the model to examine its effect on the final review representation. For aspect-based sentiment analysis, different score calculation functions in attention layer are examined with different aspect distributed representation consideration. In addition, using attention-only layers in both level review and aspect based sentiment analysis are examined to test its ability to capture the polarity into given review. The experiments’ results show the potentials of using this architecture with Arabic sentiment analysis systems. Because all models’ components are differentiable, the models will be trained by using gradient descent cross-entropy error function.

Some of these models proved its efficiency in a particular level of classification, such as review level or aspect-based level in terms of accuracy. However, evaluating the proposed models’ results in two levels, overall precision accuracy and having a detailed qualification analysis for the examined reviews distinguish this thesis. In review level, a detailed analysis of handcrafted use cases were conducted, such as testing the ability of the proposed models to handle word context and handling negation and intensification styles. In aspect level, use cases were extracted out of test dataset to check the ability of models to classify sentiment and identify the most salient words in a given review toward an aspect.

In Arabic natural language processing, there is a limitation on the linguistic resources. This thesis shows that by using neural language modeling with the mean of word embedding, the need to have linguistic resources is disregarded and the model could understand the sentiment of given review. The testing is achieved by using different visualization techniques to examine the model training and analyze their outcomes. In addition, using these different visualization representations helps is having comprehensive and analytical study on the performance of generated models on Arabic reviews from linguistics point of view and helps Arabic sentiment analysis new researchers to go further steps.

# 1.5 Thesis Structure

In this chapter, we have presented the motivation to conduct Arabic sentiment analysis research. The main objectives of this thesis are also explained. The remainder of the thesis is organized as follows. Chapter 2 summarizes the background foundation needed to understand the whole thesis in the area of sentiment analysis, artificial neural network, neural language modeling, deep learning and attention mechanism. Then, in chapter 3, related works are discussed. Chapter 4 is presenting the methodology used in designing the proposed Arabic Deep Attention-Based Review Level Sentiment Analysis model (DARLSA). Chapter 5 is explaining the methodology used to design the second proposed model Arabic deep Attention-based Aspect Level Sentiment Analysis model (DAALSA). In chapter 6, experimental setup is presented. In chapter 7, studying the performance of different proposed models is conducted by having an evaluation process and qualitative analysis. Evaluation process is estimating and analyzing the sentiment distribution for each class, whereas qualitative analysis is adopting the visualization techniques to justify model efficiency in selecting salient words for task at hand. Finally, chapter 8 is concluding the final remarks of the thesis and ending with suggested directions for future improvements and researches.

# 1.6 Summary

This chapter gives introductory information about the topic of this thesis. It starts with the motivation section that explains the importance of the problem at hand. It had been clarified that there exist a lot of opinionated Arabic data that worth to be analyzed and studied. Sentiment analysis process is achieved by using different techniques such as using NLP tools, linguistics resources and ML algorithms. Arabic sentiment analysis is a growing research area with limited supportive tools. For this reason, the need to have a model that does not rely on any external resources is crucial. Deep learning models is a type of representation learning models that learn and understand the underlying meaning in a given text, which makes it suitable for low resource languages such as Arabic. Then, research’s objectives and questions that this work tries to address are presented in section 1.2. Then, research’s hypothesis is presented followed by this work contribution. In this thesis, the effect of using attention-based models for both levels, review and aspect levels, for Arabic language is investigated. This chapter ends with listing chapters’ structure of the thesis in section 1.5.

# Chapter 2: Background

This chapter explains the basic information regarding the current work. An overview of Arabic language is presented in the first section. Then, in section 2.2 general backgrounds related to sentiment analysis is presented. Section 2.3 gives detailed steps to classify text based on machine learning techniques. Next, in section 2.4 artificial neural networks with all its related aspects will be discussed. Neural Language model will be the subject of section 2.5. An overview of deep learning will be given in section 2.6. Then, attention mechanism is going to be presented in section 2.7. Finally, section 2.8 is concluding the chapter.

# 2.1 Arabic Language

Arabic is a natural language that shares some common characteristics with other language but has its own identity too. As Arabic language is Qur’an language, the Muslims’ holy book, it is more commonly to be spoken by any Muslim around the world. Three different representations exist for Arabic language including Classic Arabic, Modern Standard Arabic and Dialect Arabic. Historically, Arabs had used Classical Arabic (CA) for more than fourteen centuries. Nowadays, the formal known representation used in academia, books and media is Modern Standard Arabic (MSA) where new reinvented terminologies needed to explain the modern life are included. Dialect Arabic (DA) is a local language as each Arabic region has it own variation of the language (Habash, 2010; Darwish and Magdy, 2014). This thesis investigating sentiment Analysis by using MSA datasets and uses the word of Arabic to represent MSA.

Arabic language is cursive language where the word is composed by connecting letters from right to left. Twenty-eight letters could be used in words; three are long vowels, with no capitalization. Arabic uses diacritics to help in the pronunciation of the words and act as a substitution of short vowels in English. The letter could be written by using different ways, depending on its location in the word. Arabic words are derived from three-letters words called root. The first word in any sentence determines the type of the sentence, which are either nominal or verbal sentences. The nominal sentence consists of a subject and predicate and does not contain any verb, whereas the verbal sentence follows the structure of Verb-Subject-Object. Nouns have different formats as well such as singular, dual and plural. Verbs in Arabic could be in the past, present and future tenses in imperative, perfect and imperfect actions. (Habash, 2010; Darwish and Magdy, 2014).

# 2.1.1 Arabic Natural Language Processing (ANLP) Challenges

Morphological complexity nature found in Arabic language is considered as a major problem faced by Arabic Natural Language Processing researchers. If not considered, these complexities may have negative influence on system performance. The following are some of these challenges that should be studied carefully, especially for the task of sentiment analysis.

# 2.1.1.1 Handling Diglossia

As mentioned before Arabic has three different styles CA, MSA and DA. These language variations share some common structure but differ in some features such as terminologies used, styles, grammars and lexicons. It is almost impossible for a single ANLP to deal with all variegations at once. When developing ANLP, a specific style of Arabic should be selected with a suitable dataset to work on (Habash, 2010; Farghaly and Shaalan, 2009).

# 2.1.1.2 The Lack of Punctuation Rules

In Arabic language, writers usually do not follow rules in adding punctuation when writing sentences. A full sentence may have no single punctuation and only ends with full stop, which makes it hard for an ANLP system to draw boundary for phrases (Farghaly and Shaalan, 2009).

# 2.1.1.3 The Lack of Capitalization

In Arabic language letters’ shapes depend on it position within the word. There is no upper case letters as all letters in the paragraph have the same case. This adds extra challenge on ANLP to identify the beginning of a sentence, especially if writers have omitted the punctuation rules (Farghaly and Shaalan, 2009).

# 2.1.1.4 The Problem of Tokenization

Tokenization in European language includes cutting the text into tokens, or units, any text before a space is a token. This situation does not work for Arabic language due to it morphological complexity. Arabic is an agglutinative language, a single token may contain many words connected into one token in prefixes, suffixes, and infixes mean. Therefore, for any ANLP a customized tokenization should be used that is empowered with rules about how Arabic words are concatenated (Farghaly and Shaalan, 2009; Monroe et.al, 2014).

# 2.1.1.5 Handling Negation

Negation is an importance writing style in any natural language. It converts the whole meaning of the sentence into the opposite direction, which makes the detection of it essential for sentiment analysis systems. In Arabic, negation could be expressed by using negation words. However, negation words could be used in Arabic as an intensification styles to stress more on the meaning not to negate it (Hamdi et. al, 2016).

# 2.2 Sentiment Analysis Systems

Recently, business has realized the important role that voice of the customer plays in their organizations. The need to have classification systems that are able to handle their feedback efficiently data is emerged. Sentiment analysis is considered as branch social information where retrieved text are intended to be classified into many classes depending on the detected emotions. Since its early trails, sentiment analysis system is implemented by using the classical binary classification system techniques as machine learning classification algorithms had been proven its efficiency in classifying reviews (Pang et. al, 2002).

Sentiment analysis, or opinion mining, is the science of extracting, studying and investigating people’s sentiments, experience and point of view that documented into a peace of text called, *review*. This extracted information could express general feelings of the authors, such as “I am happy”, or extracting sentiments regarding entities such as service, or a product (Abbasi et. al, 2008). In computer science, it is about modeling a system that classifies the polarity of a given review. Sentiment analysis model trying to classify reviews into labeled polarities, negative or positive, and some adding neutral class (Pang and Lee, 2008).  Terms “sentiment” and “polarity” will be used interchangeably throughout the thesis.

The significant definition for an opinion is a quadruple of four components (G, S, H, T). G is sentiment target, also known as entity or aspect, such as camera, product or service under review. S is the sentiment with respect to a target. Opinion’s holder is representing by H, whereas T is the time that the opinion was given in. However, not all of these components may exist in all opinions, specifically online ones. In this thesis, we are concerned about S and G, whereas investigating T and H is out of our scope.

# 2.2.1 Sentiment Analysis Levels

Analyzing the sentiment of given input could be done on different levels. In the literature, document level sentiment analysis is investigated when calculating the overall sentiment for the whole document, having that the document is about single entity. Another level of a scope is the sentiment level when each sentence is analyzed and given a polarity (Pang and Lee, 2008).  In this level, it is assumed that the review consists of several opinionated targets, aspects or entities and their polarities, which is the scope of this thesis.  The classification also goes into type of opinions, where two types of reviews are differentiated regular reviews and comparative reviews. Regular reviews are a sort of opinion that contains a sentiment regarding one entity or aspects of that particular entity. For example, “The hotel is huge.” and “Although the room is small, the hotel is considered to be luxury.” are two examples of regular reviews. Comparative reviews are opinions that comparing multiple entities based on shared aspects. For example “Although room is small, it is better than Hilton.” comparing two rooms for two different hotels, or entities.

**2.2.2 Sentiment Analysis Challenges**

While sentiment analysis system is one of NLP classification systems, it is faced by many challenges not found in the classical text classification systems. These challenges could be exist because of the nature of the problem, such as expressing positive opinion without using explicit positive terms.

Traditionally, the classical topic classification systems use keywords to guide the classification process into topics that could be unrelated. However, in sentiment classification systems no keywords are used, as it is hard in some cases to associate keywords with sentiments in reviews, such as implicit one. In addition, the difficulties will be raised if the review contains contracted sentiments regarding two different entities (Pang and Lee, 2008).

The context used has a strong correlation with the domain of the sentiment and the classification results. For example, a review “go walk now” could be classified as a positive or negative, according to its domain. If the system tries to classify sentiments for domain that is about reviewing a car, it should definitely be classified as a negative one. However, if the examined domain is about a sport shoes, this review should be classified as a positive one (Hamdi et. al, 2016).

In classical topic modeling systems, the classified topic of a given documents will depend mainly on the majority of the topics contained in a document regardless of the order of the appearance. However, in SA words’ order should be considered carefully as it affects the overall sentiment detected. For example, a given hotel’s review could be “the lobby is great, the lighting is charming, the furniture is extremely comfortable. However, you cannot tolerate it more than half an hour. ”. If this review was processed by classical topic modeling systems, a positive polarity will be assigned as the majority of the sentences contain positive thoughts. However, the last sentence is essential as it flipped the sentiment of the review into the opposite orientation (Pang and Lee, 2008).

Finally, writing style could add more challenges to the field of sentiment classification. Styles such as irony, sarcasm, explicit writing and adding negation all considered as a challenges for these systems.

**2.2.3 Sentiment Analysis Approaches**

Researches have used different approaches to classify sentiment in a given review. Lexicon-based approach and machine approach are two methodologies used in implementing such systems.

**2.2.3.1 Lexicon- based**

Lexicon-based approach is focusing on extracting words or phrases that could guide the classification process toward specific semantic orientation. Each indicator word has its own semantic value that is extracted using some means of sentiment word dictionary. The overall review’s polarity is calculated as an average sum of its words semantic values’.

# 2.2.3.2 Machine Learning

Machine learning is the science concerned about training computer programs to learn from previous examples and apply the model on new samples and monitor the outcomes. It has been successfully applied in many classification problems, including sentiment analysis and image processing, and a variety of Natural Language Processing (NLP) applications (Abbasi et. al, 2008; Deng and Yu, 2014).

Machine learning could be trained and applied on different modes. The first learning mode is supervised learning where cases with labels are offered to the computer model, or the learner, to adjust the learning point on them, and then unseen point will be presented to test the model’s ability to generalize on unseen data. Unsupervised learning is another learning mode where the learner model trying to group similar data points together without having labels that makes the learner evaluation hard (Abbasi et. al, 2008; El-Beltagy et. al, 2017; Deng and Yu, 2014).

The concept of learning is quite different in the third learning mode, reinforcement learning. The training data is replaced by the concept of agent that interacts with its environment by following its own-programmed rules. The agent learns in trial-and-error process by updating its knowledge base with feedback from previous experience (Pang and Lee, 2008).

Supervised learning will be used in this thesis to train models, where the goal is to classify sentiment in Modern Standard Arabic (MSA) reviews.

# 2. 3 Machine Learning Text Classification Techniques

Several Natural Language Processing (NLP) techniques should be applied on an opinion to detect its polarity. Before the final classification step, several steps should be followed to get an accurate result. Next section is discussing these steps in details.

**Step1: Data Collection and Pre-Processing**

Before working on any machine learning based system, an enough amount of suitable data should be collected and processed. These datasets are used in system training to learn classifying and predicting sentiment into given text. First, the unstructured free form of the text should be transformed into structured format understood by learning algorithm at hand. One of these formats is Bag of Word (BOW) representation where the text is split into terms or tokens composed by one word (n-gram), or many words (2,3,…..n) grams. By using the detected tokens, word vector, or feature vector is created (AL-Smadi et. al, 2016). To structured free form text, several techniques are done such as tokenization, stemming, pruning (Abbasi et. al, 2008).

**Tokenize**. Creating a "bag of words" representation for the text is the first step to structure the reviews by going through tokenization process. Any subsequent filtering is done on this new format of the input text, or feature vector (Martins et. al, 2003)

**Filtering Stop Words.** Then removing the common, frequent words from the input. Depending on the purpose of the classification system, the stop words list could include any word that appear frequently in given text such as articles, conjunctions, prepositions, adverbs and so on (Martins et. al, 2003).

**N-Gramming.** This step aiming in detecting the sequence of words that usually goes together in the examined natural language. For sentiment classification systems, detecting such words helps to preserve words’ context and to understand the semantics of given reviews, so identify the correct polarities. N-gramming could be done on any level such as 2-word pairs level, 3-word terms or 4-word terms (Abbasi et. al, 2008).

**Creating Word Vector.** The generated tokens are placed into a matrix to represent the dataset on hand. Rows in this matric are identified by document ID, and columns contain the generated tokens from precedent steps. Depending on the system perspective, different weighting schema could be used to create this vector, so getting different information representations. Binary term presence, Term Frequency (TF) and Term Frequency Inverse Document Frequency (TF-IDF) are different methods used to represent word vector (Pang and Lee, 2008).

**Step 2: Classification and Evaluation**

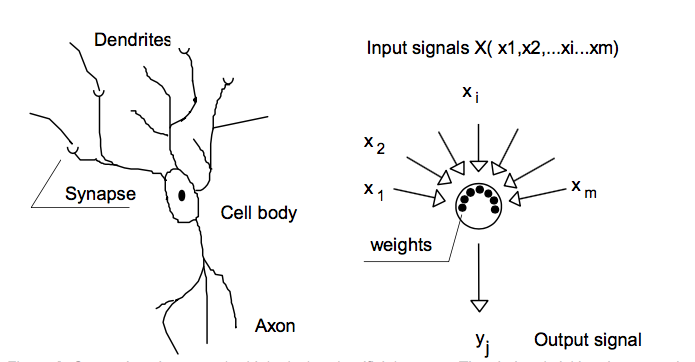
Technically, training phase is the first phase processed by supervised machine learning classification algorithms and followed by testing phase. In training phase, the system is trained to classifying polarities found in given reviews by using different known algorithms such as Support Vector Machine (SVM) that has proven its efficiency among other classifiers for text dataset (Pang et. al, 2002). By using statistical concepts, SVM algorithm divides the training data point by a separation line, into two classes and generate support vectors and the rest of training points are ignored (Pang et. al, 2002; Martins et. al, 2003; Elkan, 2013). In the testing phase, the classifier is predicting a class for a given review and then comparing the prediction with the actual one. Then, the performance of a given system could be computed by using K-fold cross validation methodology. Here, K folds of the dataset are generated and testing phase is using 10% of folds and training will consume the remaining folds. The process is repeated K times with using different testing fold at each time. Mean of all repeated testing errors is calculated to generate testing evaluation report that usually includes four different measures: accuracy, recall, precision and F-measure (Elkan, 2013).

# 2.4. Artificial Neural Network

One of the paradigms that have been used widely in the classification task is the Artificial Neural Network (ANN). It is a mimicking of the processing of information in the biological nervous systems. Collections of neurons, the processing units, are distributed among many layers to compose ANN. In the biological nervous systems, as seen in figure 2.1 (a), natural neurons are connected by synapses to receive signals. The neuron is activated when signal strength exceeds a certain threshold, and the signal will be released through the axon to activate other neurons (Anderson and McNeill, 1992; Hagan et. al, 2002).

# 2.4.1 An Artificial Neuron

When applying natural neural system metaphor on the problem of ANN, a massive number of neurons are interconnected to construct the network. Like synapses in natural neural system, each neuron receives an input and generates another signal with certain strength that is going to be calculated by using an activation function. The input is going to be multiplied by the connection’s weights to determine whether to fire the neuron or not. The whole ANN consists of number of layers, visible and hidden, that consists of a huge number of neurons, with activation functionality (Hagan et. al, 2002). Figure 2.1 (b) is demonstrating a simple comparison between the inner anatomy of natural biological and artificial neuron.



**(a)**

**(b)**

Figure 2. 1: A Comparison Between (a) Simple Biological Neuron (b) Artificial Neuron (Anderson and McNeill, 1992)

# 2. 4.2 Network Architecture

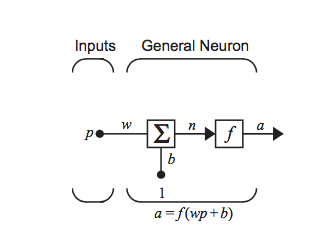
The input layer is the first visible layer and consists of number of neurons equal to the number of input tokens. In ANN, the training phase objective is to find a model of structure in the input data to allow output layer take decisions on this certain structure. The output layer is the last layer that is visible and has a number of neurons equal to the intended task. For example, if the task is binary classification problem, then the output layer should include two neurons (Hagan et. al, 2002). Beside visible layers, the input and output layers, intermediate layers between them could be existed, called hidden layers. The number of hidden layers could be varied but in most ANN one to two hidden layers are used. The optimal number of layers and number of neurons in each layer depends mainly on the problem and architecture used. Many researches have been done to detect this number and specify the main factors that could have an effect on the classification performance (Karsoliya, 2012).

Technically, dataset, of any type, should be converted into numbers as ANN could only process numerical inputs. Figure 2.2 summarizes the steps required to calculate the activation for each neuron. The input values, along its connection weights, are going to be passed to the neuron to generate a squashed output value by using its activation function (Anderson and McNeill, 1992). Adjusting connection weights until obtaining the desirable output draws a structure in the dataset during training phase. In general, the output of the neuron is computed by following two steps:

1. Computing neuron pre-activation, or input activation.

Where *w* in the connection weight, *x* in an input and *b* is bias, which is a constant equal to one, and could be omitted in specific neurons (Hagan et. al, 2002).

2. Computing neuron output activation.



***X***

Figure 2. 2: Artificial Neuron Processing (Hagan et. al, 2002)

# 2.4.3 Activation Functions

The actual amount of output released from the neuron will depend mainly on the activation function used. The non-linear activation function gives the representational power to ANN by squashing any range of input into specific output to model complex data without the need for manually identifying features set. Different activation functions types are used; varying from linear form, which simply pass the pre-activation value, to more sophisticated. The linear activation function uses the following formula to pass the strength of the neuron to the next one without any means of squashing (Anderson and McNeill, 1992):

Sigmoid function, known as logistic regression, is another activation option. It is non-linear, differentiable function that and squashes the pre-activation value into a range of values between 0 and 1 (Anderson and McNeill, 1992; Hagan et. al, 2002). Logistic regression is commonly used in the classification problems. The mapping function used is:

Another possible activation function is hyperbolic tangent, where the output is squats to be in the range between -1 and 1 by using this activation:

Figure 2.3 concludes the calculated output range for sigmoid and hyperbolic tangent activation functions.

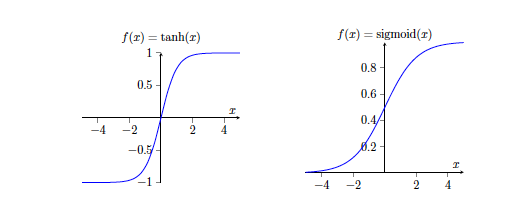


Figure 2. 3: Activation Functions (Anderson and McNeill, 1992)

# 2.4.4 Network Topologies

Since its origin, artificial intelligence’s researchers are intended to develop systems that interact with users in time. Dialogue systems, modern interactive systems such as robotic surgery and self-driving cars are all examples of such a system with interaction took place during a sequence of time. To reach this functionality, combination of classifiers and activation functions should be connected, with respect to time (Lipton et. al 2015).

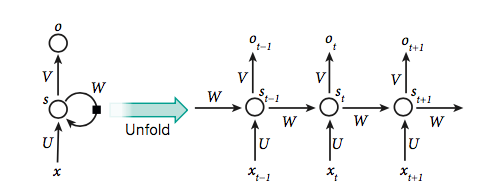
# 2.4.4.1 Feed-Forward Artificial Neural Networks

Each layer is connected to the adjacent layer in feed-forward neural network, without using any backwards connections (Lipton et. al 2015). The interaction between input and output layers are taken place by means of some statistical calculations done of hidden layers. Feed-forward neural network gained popularity as it proved its effectiveness in many problems such as classification tasks (Abbasi et. al, 2008).

# 2.4.4.2 Recurrent Neural Networks (RNN)

Recurrent neural networks (RNN) (Elman, 1990) are topologies that use backward connections. The activation functions are executed in a mean of loop when one or more of layers’ inputs are connected with a backward connection to a preceding layer. A form of network memory is constructed by this recurrent behavior, which make this topology suitable for temporal data, such as NLP applications, when the input and output form sequences (Socher, 2014; Lipton et. al 2015).

RNN architecture is representing into figure 2.4. An ordered list, presented in a sequential manner, is accepted as input sequence represented by input vectors, and initial state vector . After observing inputs, an ordered list of state vectors with its corresponding output vectors are returned. V is output layer weight vector, and U is hidden layer weight vector. W is a shared weight vector across all time steps.

Figure 2. 4: RNN Architecture (LeCun et. al, 2015)

The current hidden state is calculated by linearly combining the previous state, with the current input,

Input sequence, is mapped by RNN into , the output sequence, where each element is dependent on all previous .

Despite its simplicity, RNN has reached an acceptable performance in many applications, including language modeling and machine translation. However, it suffers from different problems such as vanishing gradient and inefficiency to manipulate long sequences. As a result, different variations to RNN are existed to improve the generated final representation.

# 2.4.4.3 The Long Short-Term Memory (LSTM)

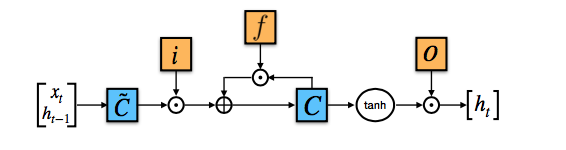
Long Short-Term Memory (LSTM), proposed by (Hochreiter and Schmidhuber, 1997), to overcome vanishing gradient phoneme faced by classical RNN, where the gradients grow or decrease exponentially when dealing with long input sequences. It is a simple RNN designed with Long Short-Term Memory in form of weights. Figure 2.5 is presenting architecture for LSTM gates.

Figure 2. 5: LSTM Gates (Young et. al, 2018)

Beside standard RNN neural gates, LSTM cell has three additional ones: input gate *i*, forget gate *f* and output gate *o*. A gate is a metaphor for matrix multiplication and got its name due to the fact that if the result is zero then flow of values will be stopped. At each time step, previous cell state and previous hidden state are used to calculate the current cell state , and hidden state , as follows:

Where is element-wise multiplication, is sigmoid function, W, b are the parameters to be trained. Memory components and are two vectors representing the current state at time-step *t*, and are going to be used to calculate the gates’ values. Gate output values are generating by linearly combining current input vector and the previous state vector. By using *tanh* activation function, the candidate value that is generated by update gate *g* is computed to update the memory . Then, the forget gate *g* and input gate *i* are going to be used. *f* is used in () to choose the amount of information to be held, and *i* in (*g* *i*) to choose the amount of information to be updated. Finally, by using *tanh* activation function in the output gate *o*, the output value , or the current hidden state , is calculated by using .

By using the gating mechanism proposed, the gradient stays high for long time-steps, and thus overcome the problem of vanishing gradient. Despite its effectiveness, the complexity found in the gating mechanism make the architecture expensive to execute and hard to analyze. As a result, many variations have been proposed but gated recurrent unit (GRU), introduced by (Chung et. al, 2014), is one of the strongest competitors to LSTM. Compared to LSTM, it is much simpler and faster and needs fewer gates to calculate as presented in figure 2.6. An explanation of GRU formulas is found in section 4.1.2.3.

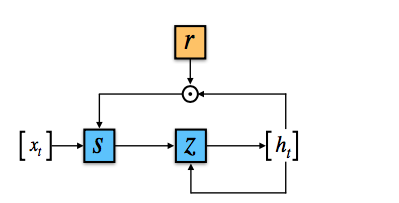


Figure 2. 6: GRU Gates (Young et. al, 2018)

Several researches had used the final output as an input into another layers to achieve certain task (Bahdanau et. al, 2014; Sukhbaatar et. al, 2015; Cho et. al, 2015; Yang et. al, 2016; Lin et. al, 2017). For example, for language-based application such as sentiment classification task, RNN based layer could be used to generate sentence hidden representation and then, this representation could be used as input to the classification function to find a polarity into given review (Wang et. al, 2016; Lin et. al, 2017).

This architecture is known as encoder-decoder framework, where RNN based layer is included in the architecture to encode the input sequence into an output representation that used, later in the model, by another layer to supervisory learn the objective function (Wang and Raj, 2017). This approach shows its efficiency in many tasks such as classification problem (Ruder et. al, 2016a; Yang et. al, 2016; Ruder el. al, 2016b; Wang et. al, 2016; Tang et. al, 2016; Alayba et. al, 2018) and machine translation application (Bahdanau et. al, 2014).

# 2.4.5 Neural Networks Training & Gradient Descent

ANN is one of the computation paradigms that reason and derive new information out of the input data during training phase, without having to do any manual feature engineering (LeCun et. al, 2015). This structure is generated during the training phase by keeping updating the network parameters, depending on the learning paradigm used, until minimizing the output errors compared to training labels (Anderson and McNeill, 1992). Next section is discussing back-propagation algorithm as a kind of training ANN in supervised learning manner.

# 2.4.5.1 Back-Propagation Algorithm

Historically, during the 1970’s back-propagation algorithm was proposed but did not get any attention until (Rumelhart and McClelland, 1986) had published their work. The overall objective is using gradient descent method in training the ANN to identify weights set with minimized error rate. To learn the correct network weights, errors will be propagated backwards from the output layer to the input layer to get new weights set.

To train back-propagation algorithm, supervised learning technique is used, where training examples with their labels are presented to learn and adjust model’s weights before starting another cycle of learning, or epoch. This process is done into two steps, from input to output in forward pass and from output adjusting error back to input in the backward pass. Initially, random weights are set in the first pass to calculate an output and then compare the calculated value with the desired one. Fixing the errors will be done, for each training example, by updating parameters and connections’ weights. After reaching the end of the learning process, under some predefined criteria, testing examples will be presented to predict their labels and compared the predicted with the expected one to calculate learning performance (Anderson and McNeill, 1992).

In each neuron, formulas 2.1 and 2.2 will be calculated for each training set, until getting the final output from the output layer. Then, taking the root mean squared error to calculate and justify error then compare it with the desired one (Zupan, 1994), as follows:

Backwardly, gradient descent methods will be used used as individual weights are going to be adjusted to minimize the error in the network, as following:

The term (Δwji), the adjustment of each weight, could be translated as the product of the derivative of the error in respect to *wji* with a negative constant eta (). This formula is going to be calculated until error is minimized or the training should be stopped. To this amount the derivative of the error with respect to Oj , need to be calculated first.

Then, the dependency of the output on the activation will be calculated by finding the derivative of the output with respect to the weights as follows:

And we can see that (from (2.16) and (2.17)):

And so, the computing of the adjustment to each weight will be

= 2.19

Where the quantity is defined as , the local error gradient.

Finally, adjusting weight is computing by:

2.20

# 2.4.5.2 Practical Issues When Training ANN

During the neural network design process, several factors should be given extra attention and care to get successful neural network model. These factors include, but not limited to, the use of specific activation function, topology, hidden layers’ numbers and neurons counts in each hidden layer. Some of these factors are going to be discussed in the next sections.

**Choosing when to Stop the Training**

A critical decision should be taken to stop the training either when by hitting predefined threshold or considering results of validation test to avoid over-fitting phenomena. Validation test set is extracted from a training set or testing set and used to test model generalization performance. Using the validation set testing results gives the actual increasing of the accuracy if the model is actually learning not memorizing so ensuring that the over-fitting sate will not be entered. Stopping the training is a good decision at the point where the training set accuracy increases while the validation set test decreases or even stops increasing (Anderson and McNeill, 1992; Bengio et. al, 2013a).

**Selecting the Number of Neurons in the Hidden Layers**

Selecting a proper number of neurons in each layer is crucial as it plays a main role in the success of classification process. It should be experimentally selected as it has a huge influence on the learning algorithm outcomes although theses nodes do not have a direct contact with model’s external environment. Selecting a small number of neurons will not let the model to learn from the dataset and under-fitting situation will be encountered. On the other hand, putting huge number of hidden neurons will overwhelmed the network with unneeded parameters which leads to unnecessary increase in the training time and a lot of neurons will remain unfired which leads to over-fitting situation (Karsoliya, 2012).

**Choosing The Hypermeters**

Several parameters need to be identified before the training of back-propagation algorithm. These parameters are:

**A. Learning Rate**

η is the learning rate that is important in identifying step size that should be taken in the learning space. Error derivative, with respect to , in weight update equation going to be multiplied by η to generate the new weights. Choosing the suitable η is a critical step, as it will effect on the algorithm convergence. Algorithms with too small η value will need a lot of time to converge, whereas choosing a large number will let the algorithm to diverge (Anderson and McNeill, 1992).

**B. Momentum**

There are some techniques that are used with gradient descent algorithms to speed up algorithm’s convergence such as using momentum technique. Adding a small amount of value to weights’ update equation will prevent the model from stuck into shallow local minima and response to recent changes in the error space to ignore small values of errors (Zupan, 1994). In the weight update equation, a combination of gradient decreasing term with fraction of the previous weight change is used to stabilize weight change as follows:

# 2.5 Neural Language Model

Traditionally, bag-of-words features are used by most of NLP, where converting input words into a symbolic ID creates a one-hot feature representation. The length of the feature vector is equal to vocabulary size but only one dimension will be on, equal to one, for each input word (Elkan, 2013). Based on trail and error, the extracted features are manually designed, to suit the task at hand, and then used as an input to any ML classification algorithm (Mnih, 2010). The one-hot word representation is suffering from high sparsity and dimensionality as the size of the vector increasing with the size of dataset. In addition, a drawback of losing words order and context is encountered as well which leads to loose sentences’ semantics (Bengio et. al, 2003). Moreover, the model will not be able to handle unseen words during test time. With the existence of such limitations, the need to investigate another feature representation is raised.

# 2.5.1 Learning Word Representation

Recently, ANN models tend to learn feature representation vector, along with the learning phase, rather than having handcrafted one. A distributed word representation, called word embedding, has been used to replace the using of one-hot encoding representation and overcome many problems faced by such encoding schema. It is an n-dimensional, mathematical object assigned with each word. It has a fixed size, regardless of vocabulary size of the dataset. A relation between words in the dataset is drawn by the learning process and translated into numbers updated in the distributed representation vector ending up in distribution similar words adjacent to each other is the continuous space. It is believed that each dimension in the vector is capturing semantic and syntactic features of words (LeCun et. al, 2015; Bengio et. al, 2013a; Deng and Yu, 2014).

Neural Network is used successfully to learn and generate distributed word representations. Work presented by (Bengio et. al, 2003) is considered one of the first successful attempts to generate neural network language model (NNLM) by training multilayer feed-forward neural network to learn the representation along with the associated objective function. After training process completed, words share the same context will be mapped adjacent to each other in the distributed space. For example, “table” and “chair” are neighboring to each other, whereas “table” and “good” are more apart. Inspired by research done by (Bengio et. al, 2003), many researches have released modified, efficient neural language model architectures that generate distributed word representations (Mikolov et. al, 2013; Pennington et. al, 2014). In addition to English NNLM, Arabic researches find its way to generate high quality distributed word representation as proposed by (Zahran et. al, 2015). Their model proves its efficiency in capturing the semantics of Arabic language and tested in extrinsic evaluation phase for many ANLP systems.

# 2.5.2 Distributed Representation Advantages

One of the biggest advantages of distributed word representation is its ability to generalize during testing time to unseen words in the training time. The training methodology followed opens many advantages, as the training objective is mapping words with similar meaning to the same area. The model estimates the conditional probabilities the next words in the sequence, so words that share the same context will be sharing the same meaning (Mnih, 2010). Therefore, this will reduce the high dimensionality nature found in a dataset, as features space will be compactly represented into nearby representations. In addition, the data sparsity problem is improves as the model taking longer context into account. Finally, the process of domain adaptation will be easier with in hand dataset, as the model keep updating words representations during training process (Socher, 2014).

# 2.5.3 Limitations

The success of distributed representation comes with its cost. The first problem is that, as its name implies, the representation is only powerful for individual words and cannot be efficient to represent whole sentences. Many researches focus to solve this problem by proposing models to learn sentence representation such as one designed by (Socher et. al, 2010). They suggested learning a matrix that combines two tokens together and scores them by using recursive neural network. Finally, sentence distributed representation will be generated by greedy parser that combines elements with the highest scores.

Recently, attention mechanism is incorporated into a model to represent a meaningful representation for the input sentence (Bahdanau et. al, 2014; Raffel and Ellis, 2015; Yang et. al, 2016; Lin et. al, 2017), which is facilitated in this thesis by trying to find an overall review representation using attention mechanism.

Nowadays, a massive amount of data is available online which attracts researches to use these data to train much larger neural models. However, computation burden does not cope well with the availability of data because the computation time will increase as the size of the training data increase. Many training methodologies have been proposed to break the link and dependency between the size of corpora and the computation time and new models were proposed (Bengio et. al, 2013a).

However, the overall accuracy of any language model will be affected by the quality of the input words’ representations used. Beside the complexity faced when using large dataset, in terms of time and space, it will have a high influence on the overall accuracy of the downstream task. The ideal scenario is to spend a good time to collect carefully the required dataset and train a language model to generate word embeddings, known as feature learning (Glorot and Bengio, 2010; 4, Bengio et. al, 2013a; Deng and Yu, 2014). These leaned embeddings, or features, will be used later as an input features for any language application. This technique is known as transfer learning, and gaining an incremental importance in the field of NLP with the existence of distributed representations (Bengio et. al, 2013a; Tang et. al, 2015).

Pre-trained word embedding will be used in the input layer for the current proposed work to apply transfer-learning methodology. In addition, experiments will be held to show the effect of applying such concept on a problem at hand.

# 2.6. Deep Learning

Training neural network to learn distributed word representation involves using an architecture of many layers to generate multiple representation, or level of abstraction, to the same word from each layer (Bengio et. al, 2013a). These models are so called deep models as they deepen into many layers. Higher-level layers are producing more abstract representation than low, inner layers, which helps in understanding the semantics of input and extracting relations between them (Bengio et. al, 2013a; Socher, 2014; Deng and Yu, 2014; Tang et. al, 2016).

(Hinton et. al, 2006) had proposed a model based on Deep Belief Net (DBN) with stacked Restricted Boltzmann Machine (RBM) layers that were trained in greedy, un-supervised manner. After training, this model was used in a multi-layer neural network as initialization of model’s weights. This model was the transformation point for deep learning as researchers started to investigate the topic more by using different techniques and applied it on different tasks and applications, for example machine translation, speech recognition and sentiment analysis. In addition, three more reasons were contributed to the increase of deep learning interest. The first reason is the availability of high performance hardware with reasonable cost. Nowadays, Graphical Processing Unit (GPU) is equipped with every machine and has a great ability to execute complicated model with short time. In addition, neural network models need a lot of data to learn and behave efficiently. Therefore, the massive dataset available recently has pulled attention towards deep learning models (Qiu et al., 2014; LeCun et. al, 2015). Furthermore, the availability of efficient algorithms, such as Recurrent Neural Network (RNN) (Elman, 1990), Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), and optimization techniques (Goodfellow et. al, 2013a; Bengio et. al, 2013b;) with the existence of open source neural network libraries such as Keras (Chollet, 2015), makes the development process easy and attracts more researchers to work in this area.

Recently, a lot of deep learning methods were proposed and proved its efficiency in generating meaningful representations that increase the popularity of deep learning in the field of NLP. As a result, feature engineering role has subsided in machine learning projects with the ability of the model to extract features and learn their by using raw data on multilayer neural network (Qiu et al., 2014; LeCun et. al, 2015).

In machine learning, deep learning model is differentiated that ANN architectures by having more layers that could be trained in supervised or unsupervised manner. Learned information is stored as network weights and could be used later in many models, which has proven that it improves results in many applications (Bengio et. al, 2013a).

One of the most powerful features of deep learning is its ability to fight the curse of dimensionality and reduce data sparsity, by using optimization and regularization techniques, which consider as a crucial issue with the availability of massive data. In addition, the existence of high specification hardware with high processing capabilities acts an influencing role in deep learning popularity. Deep learning models need a lot of matrix multiplication to train and learn internal representations. This could be achieved with the existence of high speed, low cost multi core processors and Graphics Processing Unit (GPU) computing architecture (LeCun et. al, 2015). Learning representation, distributed representation and learning multiple level of representations are three additional factors has boost deep learning models’ performance (Socher, 2014). In addition, the popularity of deep learning has been increased with recent improvements in NLP research, which increases its performance, and pushes the state of art into further steps. Related works chapter is presenting many of these words and link them to the area of deep learning.

Many companies use deep learning methodology to develop their own solutions after it proved its efficiency in academic research. Google has developed a deep model with an accuracy of 98% after spending six days to train deep model with eleven layers, which has the capability to detect numbers in 200,000 images taken by Google’s Street View cameras (Goodfellow et. al, 2013b).

Microsoft[[3]](#footnote-3) has used deep learning to produce its speech recognition algorithm. Apple has incorporated deep models as well in the voice assistance feature of Apple’s iPhone, Siri[[4]](#footnote-4).

# 2.7 Attention Mechanism

Attention mechanism is pure reduction process that transforms the input hidden representation into final dense representation. It is relying on the concept of memory network, which is a general machine-learning framework of persisting memory, or vector. It is an imitation of visual attention methodology used to focus on different image regions. It had been applied in machine translation as a first NLP application to help memorizing long source sentences (Bahdanau et. al, 2014). Attention mechanism is relying on the idea of having a long-term memory used for prediction and has the ability to be retrieved and modified during training time. The neural network is able to search through their memory component to learn where to focus on places that matches what they are looking for. The most important point is including differentiable components in the attention architecture to allow end-to-end ANN training.

# 2.7.1 The Power of Attention Mechanism

As we have noticed that recurrent-based neural network, such as RNN and LSTM, may suffer from vanishing gradient problem and have difficulties remembering long input sequence. Practically, LSTMs fails to keep up with dependency for more than few time steps (Karpathy et. al, 2015). Therefore, having a sort of memory to preserve most relevant information from the sequence is beneficial. Recurrent-based neural networks consider sequence of hidden states to generate a single hidden representation for input sequence. Compressing the whole sentence information into single vector is not reasonable as we are trying to represent too much information into limited space representation. Thus, the need to represent just the important parts of input that have a strong influence on the classification process is arisen. For example, considering the encoder-decoder architecture, the decoder uses the final output of the encoder and fits information generated into a new representation. Decoder phase is an accumulation of hidden representations, generated by encoder phase, and then weighted scores are saved for each time-step in a given sequence with respect to input experimented so far, and used later on when calculating the next upcoming scores. Because the generated output of the decoding process is a fixed length vector representation, there is a potential high risk to loss information during encoding-decoding process. Attention mechanism exists to serve as a short-term memory for a model and allows zooming in or out to focus on local or global features by optimizing weights and parameters using mathematical formulas (Bahdanau et. al, 2014).

Attention mechanism tries to exceed limitation of global sentence representation by attending and focusing on smaller parts of data, in our case words or phrases in a given review. Finally attention mechanism can add some sort of interpretability to the deep neural model.

# 2.7.2 Attention Model

In encoder-decoder metaphor, attention mechanism scores each hidden representation generated by the encoder phase for the input sequence by using weight matrices learned during training process. Then, some of the representations, depending on their importance, are passed to the subsequent layers (Wang and Raj, 2017). The number of learned matrices within attention layer varies depending on the attention model used. There are different kinds of attention mechanism such as soft, hard and self-attention. Next sections are going to discuss soft attention and self-attention as they are within the scope of this thesis.

A soft attention model, illustrated in figure 2.7, is a method takes an argument, the encoding phase output vectors, i.e. LSTM or RNN based layer. One of the attention mechanism advantages is that the calculated weights, could be retrieved and analyzed which reveal and explain more about how the neural network works.

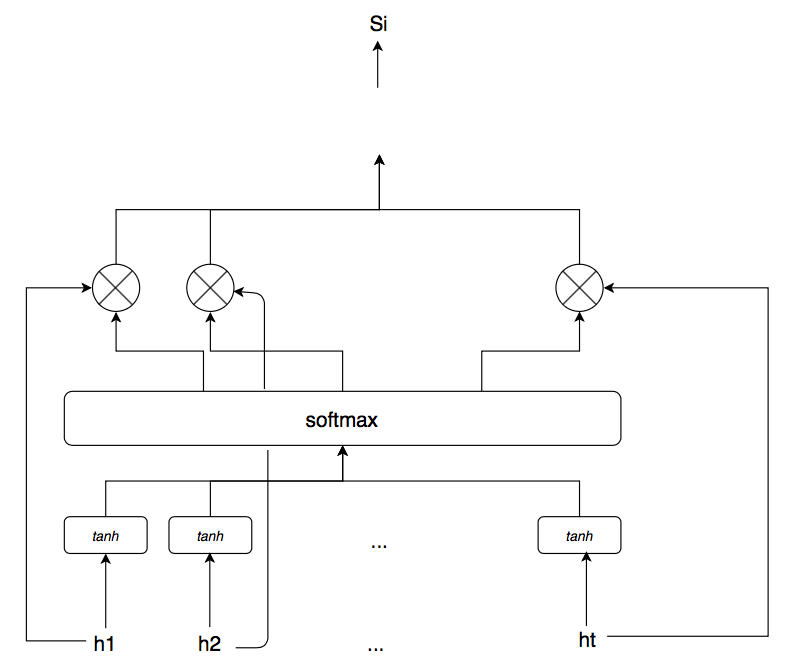


Figure 2. 7: Attention Model Architecture

**2.7.2.1 Opening Attention Model Black-Box**

Generally, the attention layer outputs a distributed vector representation for the whole review according to most salient words. A score is produced to calculate the similarity of each word with the overall text by dot-producted each word in the sequence with the final output of the encoding phase. The generated score gives an intuition about how well the word relevance to the final polarity. Then, softmax function is used by passing this score to it and gets the attention distribution to calculate the output of final layer. The following section explains this process in a step-by-step manner for more clarification.

**Step 1:** the first step is to calculate an intermediate state which is calculated by passing the value of encoding phase output vector, , with , weights matrix through one-layer MLP to learn the value of considering as the hidden representation of by using tanh layer. The un-scaled weights are calculated by using the following equation:

**Step 2:** In this step, the alignment score, or attention weights are going to be calculated. This score is the weight of each word to the overall sequence. Softmax function usually used within machine learning applications to output a categorical probability distribution. Here, a simple softmax function is used as it is differentiable and the total weight sums to 1. A weight is going to be calculated for each hidden representation as follow:

can be interpreted as the probability that is the salient words in the sentence that should be attended, where *n* is the maximum tokens number in input review.

The attention mechanism could vary depending on the calculated . Self-attention model, proposed by (Lin et. al, 2017; Yang et. al, 2016), is one variation of soft attention where a context vector is learned and dot-producted with the attention weights. At the beginning of the training, the context vector is initialized randomly, with other ANN parameters, and then learned and justified during training process. This vector is trained to focus on related words, such as aspect or products, in a given input sequence. An alignment score is going to measure the similarity of with the context vector, and calculated by

The output of this activation function is a weighted arithmetic mean of . The weights are updated according to the relevance of each word in the sequence to the with a context vector .

**Step 3:** A weighted arithmetic mean,, is calculated. It is the softmax value of projected on learned direction that could be interpreted as the value of the relevance of the given variable according to the weights.

2.25

The steps explained is known as soft attention model, as it is a soft deterministic trained via back propagation. There is another type of attention known as hard attention. It is like soft version except that softmax function is not used. Instead of spreading out the attention weights over the entire input, the focus is on specific selected input. It is ‘hard’ stochastic process, instead of calculating a weighted average; hard attention uses  as a sample rate to pick one to be the salient feature (Wang and Raj, 2017).

It is important to distinguish attention mechanism from memory network. Memory network uses an external data memory that helps the model achieving its final task (Sukhbaatar et. al, 2015; Tang et. al, 2016; Weston et. al, 2014). On the other hand, short-term internal memory used in attention models contain calculated data by using input data (Bahdanau et. al, 2014; Raffel and Ellis, 2015; Yang et. al, 2016; Lin et. al, 2017). Memory network is a useful technique to be used in kind of application that needs external information in order to make a decision, such as using an external memory for storing answers to response to an input question.

# 2.8 Summary

This chapter gives the theoretical background needed to understand upcoming chapters. First, an overview of Arabic language is presented. Then, sentiment analysis system is introduced. Text classification process based on machine learning was discussed as well. A brief introduction to ANN along with its main concept, components and topologies are presented. In addition, deep learning techniques and neural language models were reviewed. Finally, the chapter gives a brief introduction about attention models.

# Chapter 3: Related Works

Different methods and techniques were used to classify text into classes, which took a lot of researchers’ concerns and efforts. In this chapter, sentiment analysis previous works, for both English and Arabic languages will be presented. For the investigated languages, the trends in sentiment analysis work will be classified into three subsections according to their scopes. The first subsection will discuss miscellaneous topics in the field, including attention-based technology and visualization techniques used. The second subsection is discussing papers conducted for review-level sentiment analysis, where the third subsection is presenting aspect-based level sentiment analysis works.

# 3.1 English Sentiment Analysis

In this section, researches conducted to examine and analyze sentiments found in English text are reviewed on different levels. First, researches that are conducted to analyze the English sentiment found in the text are discussed without specific level. Then, papers that work on review-level sentiment analysis for English language is presented. Finally, works done for English aspect-based sentiment analysis level are reviewed.

# 3.1.1 Trends in Sentiment Analysis

Compared to other algorithms, neural network produces high quality representation and proves its efficiency to tackle many potential problems such as fighting curse of dimensionality found in the text data. Inspired by (Bengio et. al, 2003; Mikolov et. al, 2013), a researcher in (Le and Mikolov, 2014) learn vector representation for unfixed length sentences by proposing an unsupervised learning algorithm called Paragraph Vector. A state-of-the-art classification performance was recorded with sentiment analysis datasets. For sentence-level sentiment analysis, they used movie reviews dataset proposed by (Pang et. al, 2002) to learn a sentence vector representation during training, and then this representation is going to be fed into logistic regression to predict a class. Their method outperformed all studied baseline in the paper by achieving an improvement of 16% relative improvement. For document-level experiments, they used IMDB proposed by (Maas et. al, 2011) and achieved an improvement in error rates over state-of-art by or 15% relative improvement.

Work presented in (Maas et. al, 2011), is intended to generate document level vector representation that by using mixed training methodology, using supervised and unsupervised learning. Inspired by (Bengio et. al, 2003) work, they capture the semantic meaning among document’s words by using a probabilistic model. Words with similar sentiment share the similar representation by relying on the star rating, tagged with online reviews. Bag-of-words weighting models were used in the baseline experiments beside other word vector representation such as approaches Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA) and TFIDF term weighting scheme. Then the performance for all models were monitored and compared with the proposed model. Two tasks were conducted in the experiments phase; sentiment binary classification done at document-level trained using (Pang et. al, 2002) dataset, subjectively prediction done at sentence-level trained on 50,000 reviews from IMDB. Their proposed model’s performance is slightly higher, compared to other models in the paper. The best performance achieved out of their methods for document-level classification task was 88.90% compared to 85.80% achieved by classical bag of words. The performance of their models in sentence-level slightly outperformed the baseline bag of word model with 88.89% compared to 87.80%.

Coooolll, presented by (Tang et. al, 2014), is a sentiment classification system trained on 10M tweets, using unsupervised deep learning methodology. The generated sentiment-specific continuous word distribution from the learning model is then merged with handcrafted features to be fed into sentiment classifier model. Using their model pushed the performance with more than 10% as it reached an accuracy of 87.61% compared to 71.52% in the case of baseline when only word embedding representation is used as classifier’s features.

The efficiency of deep learning model in sentiment analysis domain adaptation applications is examined by (Glorot et. al, 2011). Stacked denoising auto-encoders model was used to design the model trained in an unsupervised manner to learn features representation. Their experiments were done on 340,000 Amazon reviews, in two classes, from 22 different domains to do domain adaptation task. For all experiments, they trained a linear SVM on raw data to be considered as a baseline whereas their method is a trained linear SVM as well but trained and tested on data for features extracted by their system. The averaged transfer generalization error achieved by their method is 16.7% compared to 24.3% achieved by the baseline method.

Recently, attention mechanism, which is a sort of memory network, finds its way and proves its efficiency in different deep learning NLP applications. (Weston et. al, 2014) proposed the mechanism of memory network by using an external memory to read from and write into, and jointly learned during training process. Basically, the model contains *m,* memory representation, and additional four vectors I, G, O and R. I represents the function of converting input sequence into internal feature representation. Updating the memory with new input is done by G component. O is going to calculate a new output by using new input and current memory state. Finally, based on the output representation O, R outputting a response. Question answering problem was used in the experimental phase to examine the suitability of memory networks for the problem. A dynamic knowledgebase was used by the mean of using external long-term memory, to guide the decision of the output answer, and given an input the current sentences list, and a question. During the process of generating an answer, I component processes one sentence at a time, and then a sentence representation is created. Next, G component is going to update memory *m* with the value generated by I. By processing the final sentence, a memory matrix *m* maintains all sentences representations to be used by other components. Then O component will read the memory *m* and responsible about calculating and returning the relative parts of the memory *m* to the input question. Finally, the decision is done by R component given O as an input.

Applying the concept of memory network is different from model to model. For example R component could be an RNN, getting O as an input. In addition, one or more layers could be existed to express O component. Subsequent work proposed by (Sukhbaatar et. al, 2015), used memory network mechanism but with multiple layers to generate O representations. They proposed multi-hops attention architecture that recursively read from external memory. The flexibility of the model relies on end-to-end training mechanism, with back-propagation techniques. By applying the proposed model on language modeling and question answering tasks, where each word was considered as memory entry, the results showed that multiple levels of attention could lead to more expressive representation and returns comparative performance to deep LSTM.

The idea behind attention model is selecting the relevant parts of the input that most help the classification process instead of considering the whole input. Machine translation area had the first application of attention mechanism in NLP. The model, was proposed by (Bahdanau et. al, 2014), calculate an attention score for the hidden states of the decoder, and then alignment between source and target languages is conducted by visualizing the attention score of the input sequence.

Research conducted by (Raffel and Ellis, 2015), proved that, by using feed-forward neural network, attention mechanism can solve the problems faced by traditional recurrent-based neural network when dealing with long-term sequence length. They have proven that using soft attention mechanism not suitable for temporal data, when the ordering is important, as words’ order will be vanished in computing average overtime equation. To test their proposed method, they used a toy problem of “addition” and “multiplication”.

One of the interested published models is Transformer presented by (Vaswani et. al, 2017). The proposed model relies on multi-headed self-attention and proposed as part of Google's machine translation team effort. The authors distinguished their work by training their model using only self-attention mechanism, instead of the encoder-decoder traditional architecture, and generating final representations. The proposed work had been used on two machine translation tasks efficiently.

Recently, visualizing deep models gains attention to understand the decisions made by the network and opens the possibilities for better optimization. (Karpathy et. al, 2015) tried to find an interpretation by analyzing RNN and LSTM character-level language models behaviors. Models’ outcomes such as representations, predictions and error types are visualized and studied. Their experiments aimed to track neurons that are response to certain language characteristics and maintain dependencies in the language such as detecting end of line and open and close parentheses.

Inspired by computer vision visualization, an effort to understand neural models for NLP is a one done in (Li et. al, 2016). The deep models visualization techniques have been used to understand the compositionality of negation, intensification and concessive clauses in sentences. Heatmap visualization is conducted by means of first-order derivative to visualize unit’s salience, the importance of each input unit to the final decision for a given input. Intensification and negation styles handled by LSTM neural network are visualized by plotting the final dense vector representing the required clause by composing word representations from the pre-trained model. Representations over time from LSTMs are also plotted for clauses under examination, by plotting each word embedding in the clause.

# 3.1.2 Review-level sentiment analysis

Historically, sentiment analysis researches used classical topic-classification implementation techniques to develop sentiment classification systems. (Pang et. al, 2002) had proposed one of the first works that tried to classify sentiment in given text. In their work, traditional machine learning algorithms, such as Naive Bayes; maximum entropy classification, and Support Vector Machines (SVM), on movie reviews dataset. The outperforming algorithm was SVM with an accuracy of 82.9%. However, they have noticed that sentiment analysis performance of machine learning algorithm classification task remains lower than topic-based text classification.

For a period of time, researches claimed that the suitable topology to represent natural language is Recurrent Neural Networks (RNNs) as it has the ability to work on sequential data with different length. A model proposed by (Irsoy and Cardie, 2014) trained deep RNN on news articles dataset, to implement a model for opinion expression extraction, for sentence-level. They deep RNN performance were compared with RNN with one hidden layer. Their proposed method received an f1-score of 63.83% compared to 60.35% for shallow RNN.

Sentiment Treebank was produced by (Socher et. al, 2013) to train Recursive Neural Tensor Network (RNTN) that accepted an input vector to represent variable length phrases. Then, the model generates vectors from higher nodes by parsing tree based on compositionality function. The model was trained on a dataset produced by (Pang et. al, 2002) and proved that the abstracted representation has the ability to deal with semantic space problems, such as data sparsity, more than word level representation. The experiments showed that by eliminating non-linear activation function the performance of the recursive model had decreased. Moreover, their model proved its efficiency in capturing negation styles along with their scopes at different tree’s levels. Their model registered a state-of-art performance by an accuracy of 85.4%, compared to 79.4% for SVM and 82.4% for RNN.

In this work (Guan et. al, 2016), the authors fed customer reviews into binary classifier by using deep-based neural network model. Their proposed model tends to learn distributed review representation by using convolutional neural network. They use 11754 Amazon customers’ reviews and achieved f1-score of 87.6% for their proposed method compared to 81.8% for SVM classifier.

(Lin et. al, 2017) used the concept of self-attention for generating meaningful sentences embedding that preserve linguistics semantics. Sentence representation is generated as a result of passing LSTM hidden states into attention-based layer, which modeled to generate the final representation into the shape of 2D matrix. Sentiment classification is one of the applications that were tested to examine the efficiency of their model. They used a dataset consists of 2.7M yelp reviews with 64.21% accuracy.

(Yang et. al, 2016) proposed HAN model to generate a document representation and classify documents. The proposed model reflects the hierarchical found in documents by designing neural attention model to generate informative document representation. The model tries to improve the performance of traditional soft attention mechanism by learning an additional context vector in the attention layer. This vector preserves the importance of words along with its order in the document. Two attention layers were used one for words level the other one for the sentences level to stress more on important words when the final document representation generated out of sentence representations from the first level. They have tested their model on six document classification datasets and proved its efficiency by having an outperformed accuracy increasing between 4% and 6% compared to state-of-are results. HAN model gained a high classification performance on the selected dataset and visualizing attention layers weights proved the model’s ability in selecting select salient features that represent the document and lead to better classification.

# 3.1.3 Aspect-based sentiment analysis

The work done by (Jebbara and Cimiano, 2017) is proposing a sentiment analysis model from perspective of relation extraction. They addressed the problem by identifying aspects in the review first and then the aspect is going to be labeled with sentiment. Finally, a relation between aspect and review’s terms is going to be drawn. To achieve these tasks, they used customer review dataset to train deep hybrid architecture where a convolutional neural network layer (CNN) was inserted after recurrent neural network layer (RNN). The intuition behind their model is that CNN have been used in sequence tagging problems and RNN have been applied successfully on NLP tasks, so combining them could help achieving good performance for a problem at hand. RNN on top of CNN could help preserving far information and incorporating knowledge and relations between an aspect and its sentiment words by using two features, its word distance and Part-of-Speech (POS) tags. They examined their model by using SemEval2015 Task 12 restaurant domain dataset and the results have outperformed the state-of-art by 15% for aspect-opinion relation extraction.

An aspect-based sentiment analysis model was proposed by (Ruder et. al, 2016a). The model is based on hierarchical structure that considering the interdependencies of sentences. The words’ weights initialized using 300 dimensions pre-trained embedding, and then 200 dimensions bidirectional LSTM layer is placed. They used multilingual dataset for five languages, including Arabic, in multi-domains. A high performance was achieved for all examined languages without using any external resources such as lexical or handcrafted features. The best achieving model was English one, using restaurant dataset with an accuracy of 88%.

Work done by (Tang et. al, 2015), concluded that the performance of aspect-based classification system could be significantly increased if the aspect information incorporates into review representation, without including external sentiment lexicons or syntactic parser. The proposed target dependent LSTM (TD-LSTM) model captures the dependencies between aspect words, called target words in their model, and their context within the review. To generate the final representation, the target word is placed in the middle then state information is propagated, word-by-word, by LSTM layer in both ways. The processing starts from the beginning of the phrase and forwarded until the target word, and then from the end of the phrase and backward to the target word. They trained their model on a dataset consists of 6,248 sentences and test set consists 692 sentences. Their results proved that incorporating target information into the review boost the performance by getting f1-score of 69.5% compared to 64.7% when using LSTM classifier without considering target information.

Work of this paper (Wang et. al, 2016) is a proposed framework for classifying reviews with respect to an aspect by using LSTM network enhanced by attention mechanism. In this work, the authors consider the aspect embedding with the input review embedding before fed them into attention layer. Two different ways of considering aspect information were examined. The first way is appending aspect vector with word vector before LSTM layer and the second way is making the concatenation after LSTM. The model had been trained by using customers’ reviews SemEval 2014 Task 4 (Pontiki et. al, 2014). Results proved the usefulness of including aspect information on the classification process as their proposed model gained better performance over baseline model by having an accuracy of 89.9% compared to 88.3% when using classical LSTM. The proposed work also conducted qualitative analysis for the generated results by visualizing attention score for words in each examined review to check the ability of a model to detect important words related to an aspect.

(Chen et. al, 2017) proposed non-linear framework based on multiple-attention mechanism to encode relevant information about an aspect located apart in a review. The model used bidirectional LSTM to generate un-weighted hidden representation that is going to be scored according to their relative position towards a target, or aspect. Then the model facilitates multiple attentions to filter out unrelated information to current aspect from weighted position vector weighted. Finally, by using GRU layer, the vectors resulted from different attentions are going to be transformed into a hidden representation then classified by using softmax sentiment classifier. In their experiments, they used four datasets and their models registered an increase of 4% in f1-score, for all datasets, compared to the baseline models.

Like (Chen et. al, 2017) model, work proposed by (Tang et. al, 2016) is applying deep memory network concept to generate contextual review representation with respect to an aspect by learning the importance vector along with location vector. Unlike (Chen et. al, 2017) work; they are classifying the final attention layer’s output by using multiple attentions with softmax layer. It is an optimization for TD-LSTM (Tang et. al, 2015) by including attention mechanism to the model. Their approach is faster and simpler with comparison to traditional models such as LSTM-based architecture. Despite its simplicity, the depth that included in the model had proven its efficiency and improved the performance. The results showed that the best achieving accuracy was with depth of seven layers for laptop reviews dataset with an accuracy of 72.37%. In addition, the system had registered the best performance by using a depth of nine layers for restaurants reviews dataset with an accuracy of 80.9%.

# 3.2 Arabic Sentiment Analysis

Researches conducted to examine and analyze polarity in Arabic text are discussed on different levels. First, Arabic sentiment analysis works are discussed without specific level. Then, works done on the level of review sentiment analysis for Arabic language are presented. Finally, Arabic aspect-based sentiment analysis level researches are reviewed.

# 3.2.1 Trends in Arabic Sentiment Analysis

Research conducted by (Abbasi et. al, 2008) is considered one of the earliest trials in Arabic sentiment analysis area, by examining sentiment for English and Arabic text. In the feature selection phase, Entropy Weighted Genetic Algorithm (EWGA) was used, and then SVM classifier was applied. The works on relatively small dataset, extracted from Web forum postings, with 2000 samples where 200 of them was for testing purposes, for each language. Their experiments suggested that using stylistic and syntactic features could help to increase the overall accuracy as they received an accuracy of 91% for Arabic dataset and 90% for English one.

Work done by (Shoukry and Rafea, 2012), used SVM and Naïve Bayesian classifiers to compare their results in finding polarities for 1000 Arabic tweets, specifically Egyptian dialect. They found out that SVM outperformed NB by achieving f1-score of 72% compared to 62%. One of their findings was that the performance will not benefit from stop words removal step in sentiment classification system and has a negative effect on the overall accuracy.

(Dahou et. al, 2016) had built an Arabic continuous word representation by collecting 10 billion words corpus to train 3.4 billion words. Using convolutional neural network, the generated word embedding is tested by initializing words’ weights in an Arabic sentiment classification model examined on different datasets. They tested their architecture on different dataset, such as LABR (Aly and Atiya, 2013) and twitter reviews. The performance accuracy had been increased with high quality data in the simulation results. There was a significant improvement in the accuracy for LABR dataset by achieving 86.7% compared to 78.3% in the case of using linear SVM. However, twitter dataset did not record any improvement in the accuracy, where it achieved 86.3% compared to 87.2% in the case of using SVM.

Another work to generate Arabic word embedding is done by (Altowayan and Tao, 2016), where Arabic words had been embedded into distributed vector space to develop a word embedding. To achieve this task, large Arabic corpus is collected, and then they trained a language model to generate word representation from the collected corpus. Finally, the generated embedding had been used as a feature representation into subjectivity and sentiment classifier for Arabic text. The dataset used for the classifier had been gathered from different resources, such as twitter and book reviews LABR (Aly and Atiya, 2013), to build a larger labeled Arabic dataset. By using SVM, they achieved f1-score of 77.97% for twitter dataset and 80.87% for LABR dataset.

With 1.5-billon words corpus, using ten newspapers from different Arab countries with different Arabic dialects, (Alayba et. al, 2018), had generated an Arabic distributed representation based on word2vec implementation. In addition, the generated word2vec Arabic model was used to create an Automatic Arabic Lexicon that was tested on many machine-learning methods. The best-generated word embedding, the CBOW with 200 dimensions, is used into sentiment classification system based on convolutional neural networks for processing 2026 health services Arabic tweets. The proposed approaches had proven they’re effectively by achieving an accuracy of 92% compared to 88% when using SVM approaches with classical feature selection.

Due to huge amount of data available on the Web, Arabic researches have turned to work on social media domain to investigate the problem of Arabic sentiment analysis. (Abdul-Mageed and Diab, 2011) has initiated the research on sentiment analysis in social media. They tried to analyze sentiments found in Arabic newswire along with using manually created lexicon. On sentiment detection, their work had achieved f1-score of 95.52%.

(Abdul-Mageed et. al, 2014) conducted another research to analyze sentiment of MSA and DA of social media data. Beside using handcrafted features morphological and lexica information, the social features, such as user information gender and weather is a person or organization, are automatically detected and added into the classification process. They used different forms of social media data including chat data, tweets and web forums data. In sentiment classification task, they found out that using sentiment lexicon is more informative than any other used features. They found out that adding author’s gender information but not the language variation feature will increase the classification results. The best performing model for sentiment classification experiments was SVM, which outperformed other methods with an accuracy of 70.59%.

Another work done by (Abdul-Mageed, 2018) compared the performance of deep models and SVM based models with selected handcrafted features for detecting subjective language in Twitter domain. Both models achieved high performance on the experimental problems, which encourage them to suggest investigating the area of combining both approaches on the performance of sentiment analysis systems. Their GRU-based deep classifier gain 14.50% more accuracy than the baseline method, SVM classifier, with an accuracy of 77.66% compared to 62.6% in the case of using SVM.

A hybrid method is used to develop a sentiment analysis model by (Al-Twairesh et. al, 2018) for Arabic Twitter data. A mix of lexicon-based, for extracting words’ intensities scores, and corpus-based methods were used by the mean of using machine learning SVM for classifying sentiment in tweets by using lexicon-based extracted information as an input. Their model is based on extracting features by using lexicon. They investigated the final feature selected by the method on the overall classification accuracy for four different levels of classes starting from binary classification, positive and negative classes, and ending in four classification labels for positive, negative, neutral and mix labels. They recorded the best performing model to be the two-classes model with an F1-score of 69.9%, 61.63% for three-way classes classification and 55.07% for four-way classes classification.

# 3.2.2 Review-level Arabic Sentiment Analysis

Historically, different algorithms were applied on the area of sentiment classification by using machine-learning techniques, as discussed in earlier sections. Works conducted on various reviews dataset to find polarities in is discussed in this section. Work done by (Omar et. al, 2013) used 3450 review extracted from online Arabic product and services reviews site, and classified into three classes. Rocchio classifier, Naïve Bayes (NB) and SVM were used in this stage. SVM outperformed other classifier for sentiment analysis with an accuracy of 89.81%.

Work by (Aly and Atiya, 2013) has developed a large scale Arabic book reviews (LABR) dataset. To construct the dataset, Arabic reviews for 2,131 different books were used, in a total of 63, 257 reviews, provided with two splits, training and testing splits. As the original dataset is highly biased towards positive class, the authors have provided a balanced version of the dataset also provided by down sampling the positive class. Sentiment polarity classification and rating classification models were two proposed architectures that were examined in the paper by using SVM, Bernoulli Naive Bayes (BNB) and Multinomial Naive Bayes (MNB) algorithms. For balanced version of the dataset, they achieved good results, accuracy of 91% for SVM and 83% for both BNB and MNB. For the unbalanced version, MNB has outperformed other classifier with 82.7% while SVM recorded an accuracy of 82.1% and 51.1% for BNB.

# 3.2.3 Aspect-based Arabic Sentiment Analysis

The work done in (Al-Smadi et. al, 2015) is an effort to support the Arabic research community with an annotated Aspect Based Sentiment Analysis (ABSA) dataset for books reviews in Arabic. The proposed dataset, human annotated Arabic dataset (HAAD), had been extracted out of the LABR dataset (Aly and Atiya, 2013) and then aspect-based annotations were added to the dataset. 1,513 Arabic book reviews were annotated with 2838 aspect term occurrences.

The researchers in this work (AL-Smadi et. al, 2019) proposed deep architecture aspect-based sentiment classifier using LSTM to classify Arabic Hotels’ reviews. On character and word levels, a Bi-directional LSTM with conditional random field classifier was used to extract aspects out of reviews, and then the sentiment classification was achieved by using an aspect-based LSTM layer. They used a dataset of 24,028 manually annotated reviews using the SemEvalABSA16 annotation guidelines. They use word2vector for word representation. The results show that their implementations outperform the baseline, SVM in the paper, with an accuracy of 82.6%.

Researches in (El-Beltagy et. al, 2017) presents three models to solve problems found in Arabic sentiment analysis. The first model explained is the one that classifies three classes polarity for a given text by using a scored lexicon. The second part of this work is to develop topic-based sentiment classification, where the classifier should decide the polarity for a given text toward a topic. The third part that the paper discussed is developing a model to generate distribution probability of classes for a given text toward an aspect. Multi-Layer Perceptron, a Logistic regression classifier and Convolutional neural network were three techniques used in developing models. For the first model, they used 6100 unlabeled tweets. The system has placed in the first place in SemEval 2017 task for Arabic SA with a registered f1-score of 61%. To decide on the final tweet’s label, voting among classifiers’ results is considered. The model trained on 1322 tweets with topics and their sentiment labels. The system is placed in the first place in SemEval 2017 for this task as well with an f1-score of 76.7% and Logistic regression classifier dominated other classifier by f1-score of 75.9% compared to 75.0% for Multi-Layer Perceptron and 73.7% for CNN.

The work done in (Ruder el. al, 2016b) classified Arabic Hotels’ reviews for multilingual aspect-based sentiment analysis model using SemEval 2016 Task 5 dataset (Kumar et. al, 2016), were the authors incorporated classifying sentiments found in Arabic reviews with another seven languages along with aspect category detection.. They used deep convolutional neural network model for both aspect extraction and sentiment analysis. They received the best performing system on Arabic with an accuracy of 82.7%. For English language, they placed in the seventh position with an accuracy of 80.2% compared to 88% for the best performing system. SVM was used to classify the baseline model, and Conditional Random Fields (CRF) was used for opinion target expression identification. For every language, additional external corpus and seed sentiment lexicon was used. They have received an accuracy of 81.7 % for Arabic dataset compared to 82.7% to the best performing model.

# 3.3 Summary

This chapter had reviewed related work done in the research community. The reviewed works had fallen into four categories. Neural language models, sentiment classification, both in review level and aspect-based, attention based models and works that conducted to visualize neural models. The work reviewed had been categorized according to their language, English or Arabic. Table 3.1 is concluding sentiment analysis systems researches discussed in this chapter from different dimensions. As shown in the table, the reviewed papers’ level of the sentiment is checked, whether it is classifying the sentiment in the sentence level or aspect-based level. Then, the inclusion of attention mechanism is examined. After that the language of the research is listed. Finally, the visualization option is checked, whether the system is visualizing the model representation or not.

There are different reviewed papers that work on Arabic datasets, either at review level or aspect-based level. Unlike our work, models proposed by (Aly and Atiya, 2013; Dahou et. al, 2016; Altowayan and Tao, 2016), sentiments are classified in reviews by using different machine-learning classifiers. The current proposed work in this thesis is similar to work done in (Alayba et. al, 2018), where deep learning model is trained to classify sentiment by using CNN layer, whereas the model proposed in this thesis uses GRU-based layer with attention mechanism. In addition, our work is dissimilar by proposing a model to classify reviews by using attention-only layers in the neural network without including any recurrent-based layer. To compare our work with published works on the same datasets, (Dahou et. al, 2016) and (ElSahar and El-Beltagy, 2015) both classified sentiments for LABR dataset. However, work done by (Dahou et. al, 2016) used CNN based model without having attention layer, therefor no visualization option is equipped with the model. The generated results could not be analyzed and the classification results are not understood, whereas our proposed model tries to cover this point. Work done by (ElSahar and El-Beltagy, 2015) used LABR dataset as well to train SVM classifier by using a set of handcrafted features which is unpractical solution and does not produce model that capture semantic meaning of the input review, which is our model tries to alleviate.

Works done in (Al-Smadi et. al, 2015; Ruder et. al, 2016a; Ruder el. al, 2016b; Zeng, 2016; Jebbara and Cimiano, 2017; El-Beltagy et. al, 2017; AL-Smadi et. al, 2019) are all proposing aspect-based sentiment analysis models by incorporating different techniques. Works done in (Zeng, 2016; Jebbara and Cimiano, 2017; El-Beltagy et. al, 2017) are all working on classifying Aspect-based Arabic reviews by using machine learning techniques. In (Ruder et. al, 2016a; Ruder el. al, 2016b; El-Beltagy et. al, 2017; AL-Smadi et. al, 2019), different models based on ANN are presented. CNN-base is used in (Ruder el. al, 2016b; El-Beltagy et. al, 2017), whereas hierarchical LSTM model is used by (Ruder et. al, 2016a). Our work is similar to (AL-Smadi et. al, 2019) as they have attention-based deep model, but they concatenated the aspect embedding with the review embedding, and then they fed the embedding into LSTM layer. However, the presented aspect-level models in this thesis use two GRU-based layers to generate hidden representations for review and aspect. After that, a concatenation for aspect and review representation will be done before proceeding to attention-based layer. Comparing our proposed models with published researches work on the same dataset for ABSA will identify some gaps in their works that ours tries to fill. (Pontiki et. al, 2016) proposed SVM, with handcrafted features, trained on the same dataset for ABSA phase. In addition, (Kumar et. al, 2016) work is based on using external sentiment lexicon to classify sentiments, which is not suitable for a low resources language such as Arabic. (Ruder el. al, 2016b) proposed a CNN based model, without using attention layer, which close the neural network into a black box, and the behavior of the classifier will remain un-understood. Therefor, the proposed models in this these try to overcome some of limitations found in previous model to produce better representations. Our work compares the performance of three different implementations for the attention mechanism. Like review-level, a model based on attention-only layers are proposed, one with parallel layers and one with deep, stacked attention layers. Our work distinguished by enriching the models with visualization option that add extra transparency layer. This open the accessibility to inner hidden representations, which let us conduct a detailed analysis about the performance achieved by the models in different levels, review and aspect-base levels.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Research Work** | **Level of Sentiment** | **Including Attention** | **Language** | **Visualization** | **Approach** |
| (Karpathy et. al, 2015) | Review | No | English | Yes | LSTM-based neural model |
| (Guan et. al, 2016) | Review | No | English | No | CNN-based deep model |
| (Li et. al, 2016) | Review | No | English | Yes | RNN, LSTM and Bidirectional LSTM |
| (Le and Mikolov, 2014) | Review | No | English | No | Logistic regression classifier |
| (Glorot et. al, 2011) | Review | No | English | No | Deep Learning system based on Stacked Denoising Auto-Encoders |
| (Irsoy and Cardie, 2014) | Review | No | English | No | Deep RNN |
| (Maas et. al, 2011) | Review | No | English | No | SVM |
| (Tang et. al, 2014) | Review | No | English | No | Deep learning for generating features |
| (Lin et. al, 2017) | Review | Yes | English | Yes | Deep Bidirectional LSTM with attention |
| (Yang, 2016) | Review | Yes | English | Yes | Deep GRU-based model with attention |
| (Jebbara and Cimiano, 2017) | Aspect-based | No | English | No | Stacked neural model from CNN and RNN layers |
| (Tang et. al, 2016) | Aspect-based | Yes | English | Yes | Memory network for only-attention model |
| (Wang et. al, 2016) | Aspect-based | Yes | English | Yes | Deep attention-based LSTM model |
| (Chen et. al, 2017) | Aspect-based | Yes | English | Yes | Deep Bidirectional LSTM |
| (Tang et. al, 2015) | Aspect-based | Yes | English | No | LSTM-based model |
| (Dahou et. al, 2016) | Review | No | Arabic | No | CNN model |
| (Alayba et. al, 2018) | Review | No | Arabic | No | CNN model |
| (Aly and Atiya, 2013) | Review | No | Arabic | No | SVM |
| (Altowayan and Tao, 2016) | Review | No | Arabic | No | Naive Bayes and SVM classifier |
| (El-Beltagy et. al, 2017) | Both | No | Arabic | No | MLP, logistic regression and CNN |
| (Pontiki et. al, 2016) | Both | No | Multilingual including Arabic | No | SVM classifier |
| (Al-Smadi et. al, 2015) | Aspect-based | No | Arabic | No | SVM classifier |
| (AL-Smadi et. al, 2019) | Aspect-based | Yes | Arabic | No | LSTM-based deep model |
| (Ruder et. al, 2016a) | Aspect-based | No | Multilingual including Arabic | No | Hierarchical LSTM |
| (Kumar et. al, 2016) | Aspect-based | No | Multilingual including Arabic | No | SVM |
| (Ruder el. al, 2016b) | Aspect-based | No | Multilingual including Arabic | No | CNN model |

Table 3. 1: Summary of Sentiment Analysis Reviewed works

# Chapter 4: Deep Attention-Based Review Level Sentiment Analysis Model (DARLSA): The Methodology

Explaining the methodology used in designing the first proposed model, deep attention-based review-level sentiment analysis (DARLSA) model, is presented in this chapter. First, an overview of the proposed model is presented in section 4.1; in section 4.2 the proposed DARLSA model is discussed in detail, starting with the problem definition and ending by detailed description of the method. Section 4.3 discusses the proposed model novelty and what it adds to the research community. The chapter concludes with a summary in section 4.4.

# 4.1 Review Level Sentiment Analysis System

Review level sentiment analysis system is about analyzing and classifying sentiment in given input text. Next sections are defining the research problem to make a base understanding of the proposed method. In the subsequent sections the components of the model will be discussed.

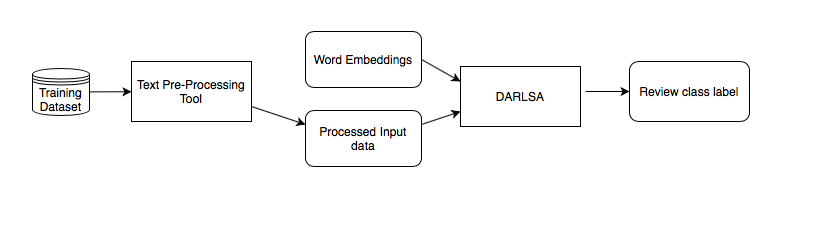
**Problem formulation**

The problem of binary classification of a review can be defined as follows:

Let a review , consisting of number of words w, padded to fixed length *n* by including padding tokens. The proposed method applies a binary sentiment classification model without using any handcrafted features. The problem is formalized into deep attention-based neural network model that has the ability to distinguish salient words in a given review. The input into the model is distributed vector representations for words within review, and the output will be the whole review distributed vector representation that is going to be classified by using linear classifier into positive or negative class.

**Model Details**

As shown in figure 4.1, the proposed approach consisting of the tools required to predict the final review class. Text-preprocessing tool is a required step to get the reviews in a format suitable for DARLSA. The dataset will be used after making the required pre-processing step. Next sections will discuss each component of the proposed architecture.

Figure 4. 1: High-level overview of the proposed approach

# 4.1.1. Text Pre-Processing Tool

Text preprocessing and normalization is the first step that should be done in any NLP system. The process will remove the noise from the used dataset such as repetition and non-Arabic text like URL. The pre-processing tool is developed to produce a cleaned version of the dataset. The normalization process is taken place first by removing emails, URLs, user handles (@user), emoticons and emojis from the text. The normalization process also transforms all similar words that appear in the dataset into the same form. In Arabic, a word could be written in many ways by having different letters to represent the same pronunciation. For example, the phrase “breakfast meal”, could be written as “وجبة الإفطار”, “وجبه الأفطار” and “وجبه الافطار”, “wjbẗ ạlạ̹fṭạr”. There are different variation forms of writing “ء”, “hamza”, letter because they are presented in Arabic dictionaries as one root form. It could be written as “أ”, hamza above, “إ”, hamza below or “آ”, “mad above”. All these forms are changed into “ا”. In addition, “ة” and “ه” forms are changed into “ه”, also “ي” and “ى” all became “ي”. The reason behind this normalization step is because there is no conventional rule to follow when writes these letters at the end of words so a lot of writers misspelled them. Therefore, letters normalization strategy considers spelling variations to increase efficiency of the model (Altowayan and Tao, 2016; Habash, 2010; Darwish and Magdy, 2014). Traditionally, stop words removal is done at this step by removing common words such as prepositions, conjunctions, and articles. For this current research, preliminary experiments are done to test the effect of eliminating stop words, using the list created by (El-Khair, 2017). However, it had been found that eliminating stop words would not have a positive effect on Arabic sentiment analysis. These experiments were done also by (Shoukry and Rafea, 2012) and also had proved that in the case of sentiment classification will not benefit from eliminating stop words. In addition, the ability of the system to detect negation phrases is going to be experimented, in later phases of this research, and negation words, which considered as one of the stop words. Therefore, having stop words elimination step should be handled carefully by removing negation words from the stop words list to get an accurate analysis.

Finally, the text will be tokenized, by using custom tokenizer. Generally, English-based natural language processing library parses the given text to produce tokens. Spaces used in the sentence will help the tokenizer to identify tokens. However, spaces are not good tokens’ indicator for Arabic language, because a single token may consist of many words before a space encountered. NLP tokenizer, such as NLTK library for Python (Rumelhart, 1986), failed to tokenize Arabic text probably, therefor a customized tokenizer need to be implemented for out system. Stanford Word Segmenter library is used to solve this issue and produce a proper tokenized pre-processed text (Monroe et.al, 2014). The outcome of this process is an integer sequence representing the input words, where the entire corpus is encoded into integer values and each word is substituted by corresponding integer value from the corpus. Also, all reviews are trimmed to be in the same length, and shorter ones will be padded by an integer zero.

# 4.1.2. Deep Attention-Based Review-Level Sentiment Analysis Model (DARLSA)

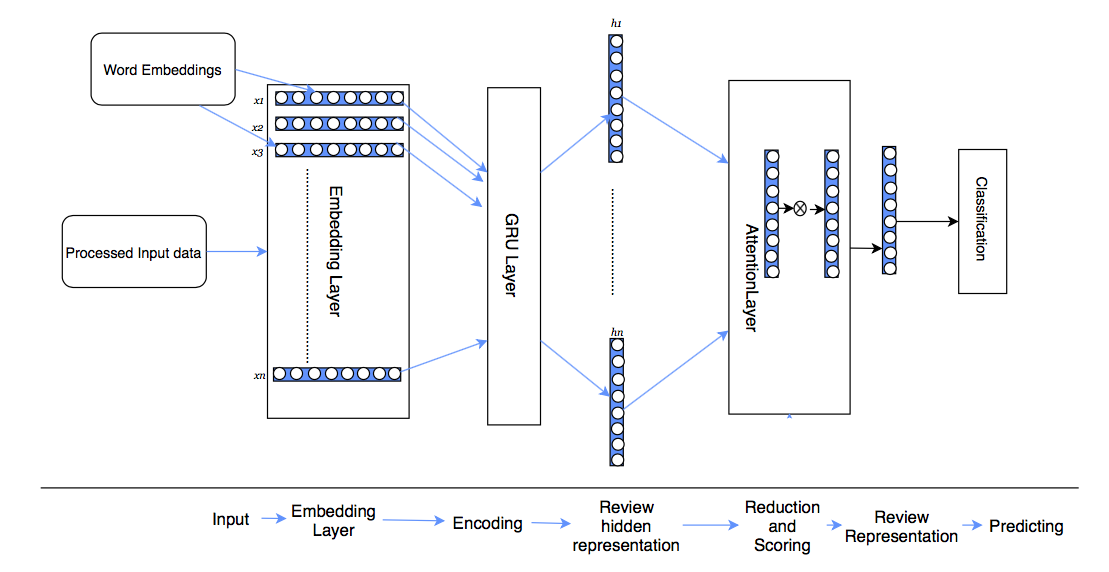
 Examining the use of attention mechanism in detecting the most informative words that contribute to the final reviews’ polarity for Arabic reviews is considered as the main objective of this model. Figure 4.2 illustrates the DARLSA model architecture. Because deep models proved its ability to reveal structures from the data (Deng and Yu, 2014; Young et. al, 2018), the model is deepening into many layers. First, the model facilitates the distributed representation by using embedding layer that passes the distributed word representations for the input review to GRU-based layer to produce hidden review representation. Then, soft attention layer is added after GRU layer to produce a weighted score representation for a given review. To get the final review vector representation, the attention layer sum up GRU hidden representation according to the calculated weights for each word. The following sections will discuss each layer in detail.

Figure 4. 2: DARLSA Model Architecture

# 4.1.2.1 Input Layer:

The network input is *n* sequence of words, ( ), which represent a review. These words are going to be processed and transformed into vector of integers as discussed in section 4.1.1.

# 4.1.2.2 Embedding Layer.

The input review consists of words that are represented as continuous real-valued vector, i.e. word embedding. When working with language model, the quality of the input words representations is very important, as it is a strong influence on model accuracy. To learn word embeddings’ weights that capture semantics of the language for the task at hand, the model should have large labeled data to work on. However, one of the major obstacles when working with Arabic models is lack of labeled resources. One solution is to apply transfer-learning technique is to initialize the embedding layer’s weights with pre-trained model’s weights. In this way, the model already inherits some common patterns of the words from the pre-trained model, which was trained on a large dataset, and further relations need to be detected only to classify reviews. Transfer learning is considered as a shortcut for the model to reach the optimal accuracy with the minimum time (Bengio et. al, 2013a; Tang et. al, 2015; Chen et. al, 2017).

In this layer, by using a dictionary of word embeddings object, the initialization of the embedding layer weights is achieved by mapping the input words’ vectors of integers into their equivalent distributed word representations. First, the words in a review are projected into a low-dimensional vector space by Embedding layer, where E Embedding layer size and *n* the total number of words in a review. The number selected for E is dependent on the size of pre-trained model used. In our case, both pre-trained embeddings used are in 300-dimensions shape.

Experiments were developed and evaluated to measure the power of using such pre-trained models on our problem. In the first experiment, layers’ weights, the weights of words’ distributed representations are initialized randomly without using any pre-trained weights. A second case is to apply transfer learning on the current model by using the weights of pertained model. Two different pre-trained models weights were used (Altowayan and Tao, 2016; Zahran et. al, 2015) and the results were monitored.

Initializing the weights for word embedding layers’ implies loading the pre-trained word embeddings weights first and placing the values into a dictionary of keys, representing a word index, and values, representing the embedding vectors for the keys. Then, for each word the model searches for input review’s words in the pre-trained model to get the corresponding weights. If a word is not found in the pre-trained dictionary, a random initialization will be done for the word embedding value a value between -0.25 and 0.25. So out-of-vocabulary rate for the pertained model is considered as a factor to measure the suitability of pre-trained model for the dataset used.

A third case is to examine the effect of fine-tuning the pre-trained weights to fit into the problem at hand.

# 4.1.2.3 GRU Layer

Choosing to use Recurrent Neural Networks (RNN) to train NLP tasks is a natural choice because the structure of the data is suitable for the logic presented by RNN. These networks try to mimic human way of observing NLP by processing the input sequentially. For every element in the sequence of input data, the activation function is performed where is a hidden state *h* at a time-step *t*, and w is network’s weights. Here, the context of the sentence is preserved as the word order is maintained by the dependence of the computation of the current state on the previous one.

For the proposed models, GRU layer (Chung et. al, 2014) is used rather than LSTM as its GRU works better in our experiments as the validation loss on the dataset was lesser and it was faster also. Compared with standard RNN, GRU cell uses two gates to handle the flow of the data, to represent update gate and to represent reset gate. determines the amount of previous information to keep while determines the methodology of combining the old memory with the new input. The entire internal memory is output without an additional activation. At each step, each word-distributed representation is given as an input word *xi*, the current cell state ct and hidden state *ht* can be updated with the previous cell state *ct-1* and previous hidden state *ht-1* as follows:

4.1

4.2

4.3

4.4

The output of this layer is the hidden representation of the given review that is going to be fed into the next layer, the attention layer, to highlight the most informative words and to generate the final review representation.

# 4.1.2.4 Attention Layer: Soft Attention Layer (SAL)

For the DARLSA model, soft attention mechanism, inspired by (Raffel and Ellis, 2015; Bahdanau et. al, 2014; Sukhbaatar et. al, 2015), is used to amplify the most informative words in a given review with higher weights. Therefore, an attention layer is placed after GRU layer to point out the most influence words to the review meaning and then the specific words’ representations are combined to form a review vector representation. To produce the final review representation, the attention layer uses the following mathematical formulas,

4.5

4.6

4.7

Using each hidden stateand dot producted with its weights is producing the hidden representation . T is number of time-steps in the input; and are the attention layer’s weights that are optimized during training to assign bigger weights to the most important words of a sentence. Softmax function is then used by passing the un-normalized hidden representation for the current word, to get the normalized importance weight . Finally, the review high level vector representation is produced by having weighted sum of the word annotations based on the weights.

# 4.1.2.5 Output Layer

To classify reviews into the labeled classes, the final review vector representation *S* is fed into fully connected sigmoid logistic regression layer. It produces a probability distribution over all given class by squashing the output in the range between 0 and 1. An output value greater than 0.5 signifies a positive polarity, and if it is less than 0.5 a negative polarity will be assigned.

# 4.2. Attention-only Review-Level Sentiment Analysis Model (ARLSA)

In this model, the effect of using only attention-based layers to generate the final review distributed representation is examined. The model is based on having three attention layers to calculate review-hidden representation without using any recurrent layers. As shown from figure 4.3, the input review representation will be fed into three SAL layers that are going to calculate three separate review distributed representations. These representations, each of which will focus on the most informative words in a given review are going to be averaged and merged, then projected into fully connected linear layer that pass it to the classification phase.

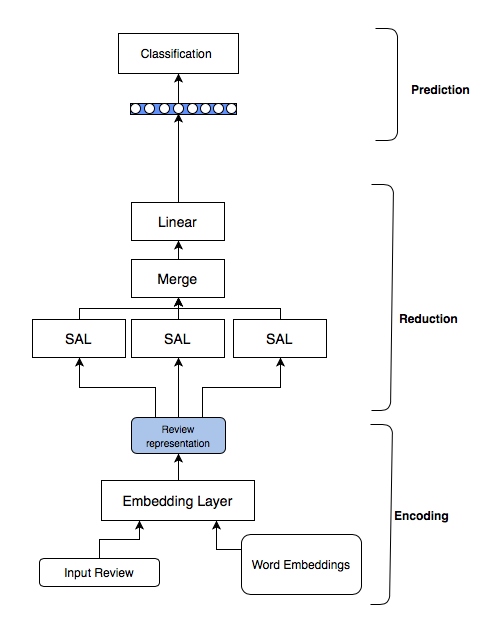


Figure 4. 3: ARLSA Model Architecture

# 4.3 Discussion

In this chapter DARLSA models have been discussed, where some of them were an implementation of existing methods and others were novel to be used in Arabic sentiment analysis. The soft attention mechanism had been examined previously for English language by literature with some variations. Research done by (Yang et. al, 2016; Lin et. al, 2017) is an implementation of the concept of soft attention in the area of review-level sentiment analysis by using deep recurrent neural network. For Arabic language, all of the reviewed researches used deep models without attention mechanism for review-level sentiment analysis. (Dahou et. al, 2016; Alayba et. al, 2018) model was using CNN-based models with word embeddings to find the overall sentiment review representation. In addition, our models add a visualization layer to the neural network by facilitating attention mechanism. This opens neural network to understand its working mechanism and analyze its output, which has not been done before for Arabic language. Finally, this work, by proposing ARLSA model, is the first work that examines using neural network with attention only layers to generate the final review representation.

# 4.4 Summary

In summary, the methodology to model the proposed Arabic sentiment analysis model in review-level is discussed this chapter. The architectures discussed in this chapter are divided into three categories, GRU-based model (baseline model), deep attention-based DARLSA model and attention-only based ARLSA model. The effect of transfer learning is studied on the baseline model and DARLSA models with three different cases. The experimental setup used for these model are going to be discussed in chapter 6, and the evaluation of these models will be discussed and compared in detail in chapter 7.

# Chapter 5: Deep Attention-Based Aspect-Level Sentiment Analysis Model (DAALSA): The Methodology

Explaining the methodology used in designing the second proposed model, deep attention-based aspect-level sentiment analysis model (DAALSA), is presented in this chapter. Section 5.1 is starting with presenting the problem statement of the proposed method, and then models architectures along with their main components will be presented and discussed in section 5.2. Attention-only proposed models will be discussed in section 5.3. The chapter will be concluded in the summary section, section 5.4.

# 5.1 Aspect-Based Sentiment Analysis System

Most neural sequence modeling techniques are relying on encoder-decoder architecture (Bahdanau et. al, 2014). Here, input sequence distributed hidden representation is generated by the encoder, which is going to be processed by the decoder to generate one element at a time to represent an output sequence of the input. In recent deep architecture, it trending to use attention layers to connect the two parts of the model, encoder and decoder, in order to have more focused distributed representation of input data (Bahdanau et. al, 2014; Sukhbaatar et. al, 2015; Raffel and Ellis, 2015; Yang et. al, 2016; Lin et. al, 2017). This chapter will use a metaphor of encoding phase and reduction phase, to generate a final review representation in response to an aspect by using attention mechanism.

# 5.1.1 Problem Formulation

The problem of review classification with respect to an aspect is the focus of this chapter and can be defined as follows:

Given an opinion with *n* words and an aspect word occurring in . Each aspect could contain many words that represent single aspect entity. For example, aspect “breakfast meal” consists of two words that represent single aspect. Aspect-level sentiment analysis considers the classification of reviews’ polarities towards certain aspect in a given review. In this model, attention layer uses aspect information in order to calculate attention weights, i.e. the importance score. Different models, relying on different attention-based layers, are proposed to generate the final review representation for a given aspect, and then the accuracy of classification is examined. The polarity of an aspect term occurrence can be positive, negative, or neutral. A review “cozy atmosphere but the food was awful!”, the sentiment polarity towards aspect “atmosphere” is positive, while the sentiment towards aspect “food” is negative.

A very important issue when modeling an aspect-based sentiment analysis is considering the relationship between aspects and their contexts, and that relation should be taken into account when generating the final review representation toward this specific aspect. Each aspect has its own context that, in turn, has its own semantic and sentiment as well. Therefore, not all context phrases in the review effect on all aspects, the saliency word list is completely different with each aspect. For example, in the review “cozy atmosphere but the food was awful!”, for aspect “food”, context word “cozy” is not as important as “awful” for this specific aspect.

# 5.1.2 Model Details

As shown in figure 5.1, the proposed model is part of a proposed approach that tries to find a polarity for a given review with respect to a given aspect. First, pre-processing is done on the dataset by normalizing text and removing tokens such as URLS, text repetition and stop words. Next, by using Stanford Word Segmenter library the text will be tokenized (Monroe et.al, 2014). The described pre-processing tool in section 4.1 will be used. Finally, the corresponding word embeddings representation will be fed into the system for both review and aspect under consideration.

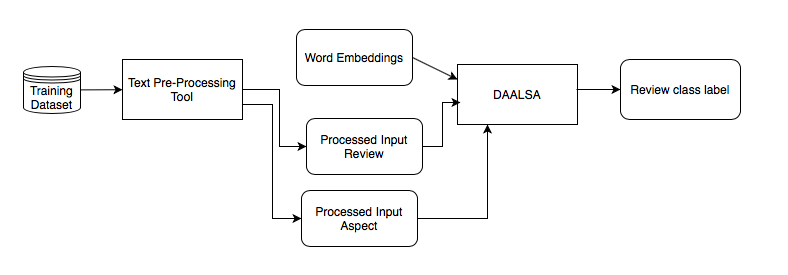


Figure 5. 1: High-Level Overview of The Proposed Approach

# 5.2 Deep Attention-Based Aspect-Level Sentiment Analysis Model (DAALSA)

Many layers are facilitating to generate the polarity of the review in responding to specific aspect, starting with an embedding layer as an input and ending with an output layer. These layers are grouped to be into encoding and reduction phases. Encoding is the process of accepting review and aspect as an input, and then review-hidden representation is generated for this specific aspect. For reduction phase, a vector that represents the review regarding specific aspect is calculated. Here, context words are considered by attention mechanism to find their contribution to the semantic meaning of the input review with respect to an aspect. Different proposed attention layers are examined in different experiments. The following section is describing the model’s layers in details.

# 5.2.1 Encoding Phase

The input to this phase is the review words along with the aspect words. Aspect words denote to an aspect as it many contain. For example, an aspect of “breakfast meal” in a restaurant dataset could be written as “وجبة الإفطار”ه “wjbẗ ạlạ̹fṭạr”, consists of two words that represent a single aspect. The output from this phase is the review-hidden representation regarding this aspect. There are different layers involved in the review hidden representation generation process. These include input layer, embedding layer and GRU layer.

# 5.2.1.1 Input Layer

The model has two input, the review in sequence of words form in *n* length, , where *rv* is the review notation, and an aspect represented as a sequence of words in *m* length where *as* denotes an aspect.

# 5.2.1.2 Embedding Layer

All input words, for both review and aspect, are projected into , where E is the number of dimensions for the embedding layer.

# 5.2.1.3 GRU Layer

In this model, GRU layers are trained to produce reviews’ and aspects’ hidden representations. Both, review and aspect representations, and , are concatenated into two dimensions matrix and fed to the next layer, i.e. attention layer, to compute the final review representation. As different models were proposed, different mechanisms to concatenate these representations were proposed as well. These mechanisms are discussed in detail in the reduction phase section.

Aspect hidden representation will be passed through Global Averaging Pool layer (GAP), to calculate , which is the mean overtime for all hidden representations overtime of the aspect to generate a single hidden representation that represent the aspect. is defined as,

5.1

Where *m* is the number of aspects’ words.

Then, “RepeatVector” layer will be placed after GAP layer to repeat the output of the encoder phase as many times as the allowed review length.

All these layers will be demonstrated graphically in each model’s figure in the reduction phase, section 5.2.2.

# 5.2.2 Reduction Phase

In this phase the review hidden representation is transformed into scored-review vector by using one of attention mechanism layers. The aspect information is playing a role when calculating the attention scoring function.

The input to this phase is hidden representation considering an aspect. The consideration of an aspect is different depending on the attention layer examined. For example, DSAL and DLVAL layers are accepting an input, the review-hidden representation, concatenated with the aspect representation. On the other hand, DAVAL layer will take an additional separate vector representing an aspect as an input. The output will be one dimension vector representation of the review in responding to that aspect. In addition to weights and bias, attention layer could learn extra parameters depending on the examined case. Three cases were examined in different models, and then accuracy and loss were examined for the given dataset. In the following sections each of these cases is proposed in a model and explained in detail.

# 5.2.2.1. Deep Aspect-Based Sentiment Analysis Model Based on Soft Attention Layer (DSAL)

As discussed earlier, attention mechanism has gained a lot of studies recently as it helps to focus more on aspect related context words. Soft attention layer, described in section 2.7.2, accepts hidden representation matrix , where d is the representation’s dimension and *n* is the maximum number of words in the input review, as an input and try to reduce the matrix into a vector *o* that represent the review with its semantic features. The output of MLP is a hidden representation that is passed through softmax function to produce attention weight vector that estimates the importance of each word to the final review polarity by where Attention mechanism permits accessing the previous GRU hidden states, , and a weighted sum of , by attention weight vector is produced.

(5.2)

tanh

Softmax

*u*

*h*

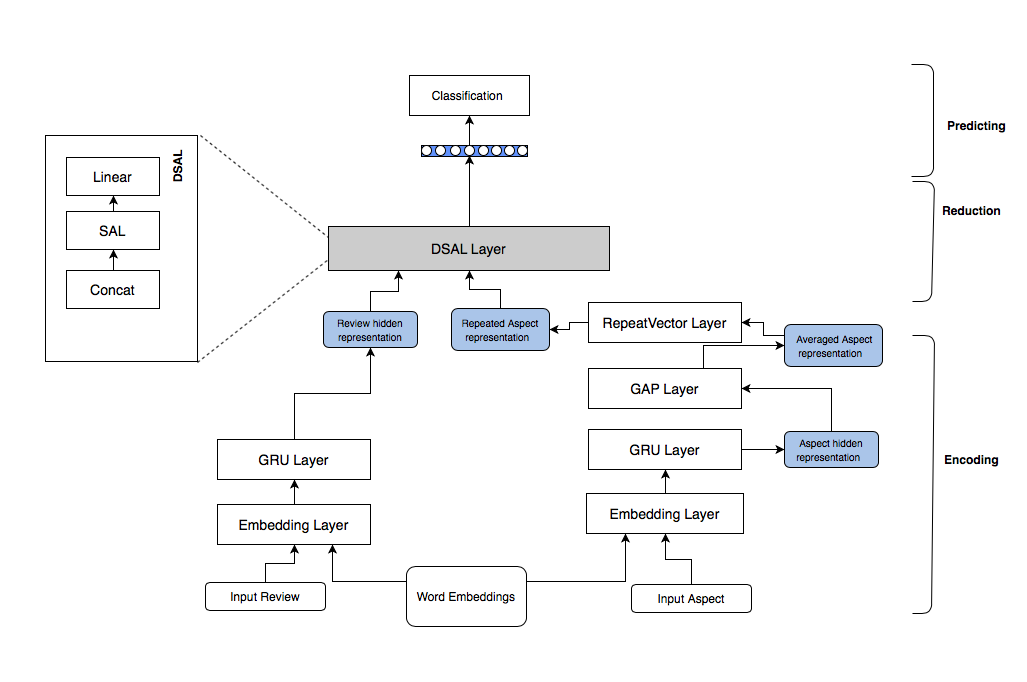
*o*

(5.3)

(5.4)

Figure 5.2 illustrates the proposed model’s components. As shown, the reduction phase accepts an input of the hidden representation of the input review and aspect generated by the encoding phase. GRU layer will generate a matrix of shape (batches number, words number in each aspect, dimension number). For the aspect vector this shape need to be modified into shape (batches number, words number in review, dimension number) to make the concatenation process possible.

This transformation will be done by having GAP layer which averages the representation for all aspect’s words, and then repeats aspect vector by the maximum number allowed for one review, then, aspect representation will be concatenated by the end of each word representation to get a matrix of shape (number of batches, maximum number of words in each aspect, 2\* number of dimension). Finally, this representation is going to be fed into SAL layer to get the overall hidden representation in shape of (batches number, 2\* dimension number). Experimentally, it has been found that the efficiency is increased if the output is projected into fully-connected linear transformation layer before classification layer to reduce the dimension into shape of (batches number, d), where d is the number of reduced dimensions. Three different reduced dimension d were experimented 50, 100 and 150, but 100 had achieved the best accuracy.

Figure 5. 2: DSAL Model Architecture

# 5.2.2.2 Deep Aspect-Based Sentiment Analysis Model Based on Learned Vector Attention Layer (DLVAL)

This model is concerned about studying the effect of using another logic to calculate attention score on the overall accuracy regarding specific aspect. Figure 5.3 presents the proposed model. It is the same implementation of DSAL model but another attention function is calculated to get attention weights. The attention layer in this model is an implementation of self-attention mechanism, inspired by (Wang et. al, 2016; Yang et. al, 2016; Lin et. al, 2017), that trained to select contextual words related to the polarity of specific aspect only. This is achieved by adding a context vector *c*, learned with layer’s parameters, *W* and *b,* during training process that answers the question of the most salient words in the review toward an aspect. The following equations, from 5.5 to 5.7, are the required calculation in LVAL layer.

(5.5)

tanh

Softmax

*u*

*h*

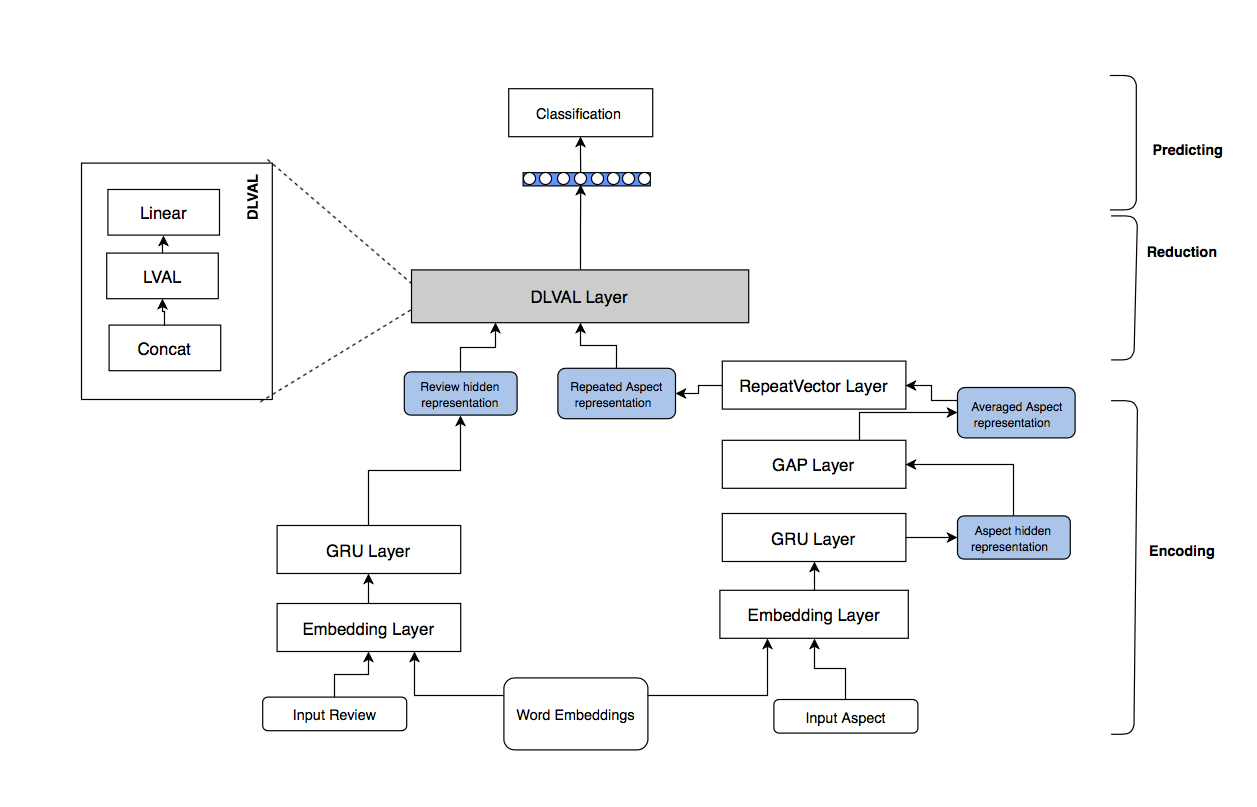
*o*

*c*

(5.6)

(5.7)

First, by using a one-layer MLP, the layer accepts a matrix *h*, which is hidden representations for review’s words, appended by an aspect embedding , where *d* is the representation dimension and *n* is the maximum words number in the input review. Then, by using softmax function, the similarity between hidden representation *u* and context vector *c* is going to be measured to generate the importance score, or the attention weights vector.Finally, the final output *o* will be calculated, which represent review-hidden representation, as a weighted sum of input hidden representation *h* based on the calculated weights

Figure 5. 3: DLVAL Model Architecture

# 5.2.2.3 Deep Aspect-Based Sentiment Analysis Model Based on Aspect Vector Attention Layer (DAVAL)

In this model, the effect of incorporating aspect representation in calculating the attention weights is investigated. The structure of the proposed DAVAL model is presented in Figure 5.4. The model has two parts each is responsible for producing a hidden representation for aspect and review. Each part has its own GRU layer to generate a hidden representation and a GAP layer to produce another averaged form for each aspect and review respectively.

DAVAL layer was first proposed by (Cho et. al, 2015) for machine translation, speech recognition, image caption generation and video description generation applications. The adeptness of DAVAL attention layer in the field of NLP will be examined in this work. Our proposed model is using two DAVAL attention layers to compute a scored representation of a given representation with respect to specific vector, one in each part.

The first attention layer, DAVAL, accepts two inputs, a matrix and a vector. The attention score weights is calculated for the given review representation, first input that is generated by GRU layer, with the aid of the aspect averaged representation vector. In the other DAVAL layer the attention score weights will be calculated for aspect hidden representation with the aid of averaged review representation. In this way, the aspect and review representations have a balanced effect on the scoring process. Finally, both representations are going to be averaged and projected into linear transformation to reduce dimensionality before the prediction phase.

Calculating the relevance of each word to an aspect will be done and the scored review representation will be generated. AVAL layer has the same logic of SAL and LVAL discussed previously but has an extra aid vector that is accepted as an input to the layer and multiplied by its own weight matrix.

tanh

Softmax

*u*

*h*

*o*

*c*

c is the aid vector, and V is its weight matrix, where d is the representation dimension and t is the number of maximum time-steps in the input matrix. In the first case, will be an input matrix representing the review-hidden representation so that it’s notated as *rv*, and will be the averaged aspect representation. In the second case, where d is the number of dimension and z is the maximum number of words in an input aspect, and will be the averaged review representation.

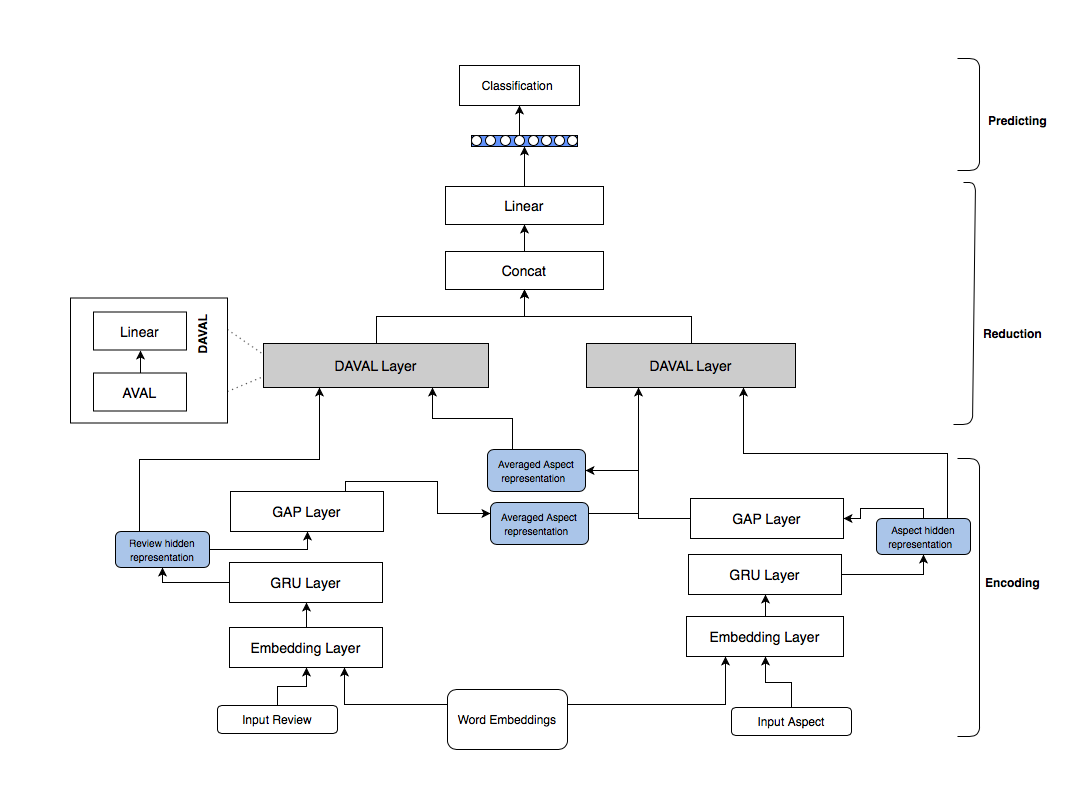


Figure 5. 4: DAVAL Model Architecture

# 5.3. Aspect-Based Sentiment Analysis Model Based on Multi-Layer Attention Model

Attention mechanism used in the previous experiments was basically a weighted average compositional function over the review-hidden representation generated by GRU layer. Many deep learning researchers have proven that using multiple layers leads to better final representation of the examined data, with multiple levels of abstractions (LeCun et. al, 2015; Zeng, 2016). Learning complex form of review representation towards an aspect is achieved by composing enough abstracted representations.

In this work, the effect of using hierarchical attention-only based models is experimented. Inspired by the use of question answering memory network (Sukhbaatar et. al, 2015), instead of performing a single attention layer after encoding phase, the goal is to project the input words and input aspect linearly into many attention-based layers without using any recurrent-based layers. We want to test the ability of attention mechanism to handle the hierarchical structure of Arabic natural language with different levels of compositionality calculated with different degree of depth. Integration the training of the memory representation into end-to-end neural network model by passing distributed representations to be processed via multiple layers and the final review representation will be generated with respect to specific aspect. During training, error could be back-propagated to the input through multiple memory accesses.

Two attention-only architectures were examined by arranging different attention layers in different architectures. The first proposed model is Parallel Attention-Only model (PAO), where the overall review representation is a result of averaging three hidden representations generated by independent three attention layers. The second proposed architecture is deep hierarchical attention-only, where the overall review representation is calculated by using deep-stacked attention layers with respect to specific aspect. The following sections will discuss each model in details.

# 5.3.1. Parallel Attention-Only Model (PAO)

The proposed approach is illustrated in Figure 5.5. Three hidden review representations will be generated independently by using three different attention-based layers. The following section is discussing the components of the model.

# 5.3.1.1 Encoding Phase:

The input to this phase is the review words along with the aspect words. The outputs from this phase are the hidden representations for both aspect and review that are going to be used later by the reduction-phase layers. There are different layers involved in the review hidden representation generation process. These include input layer, embedding layer, GAP layer and RepeatVector layer.

**Input Layer**

The model inputs are sequence of *n* words representing review, , and sequence of *m* words representing an aspect.

**Embedding Layer**

The projected distributed representation for input aspect and for input review are generated by using pre-trained word embedding. These representations are going to be passed to the encoding phase.

**GAP and RepeatVector layers**

The aspect representation is going to be represented in different formats so that later layers can use it. The generated aspect representation from embedding layer is a matrix of dimension , where m is the maximum words number in aspect sequence and E is the dimension number used. In order to be used by AVAL layer, the aspect representation matrix is averaged to generate a vector with E dimension. On the other hand, this representation is needed to be expanded into a matrix of shape by RepeatVector layer, where *n* is the maximum words number in review, used by SAL and LVAL layers.

# 5.3.1.2 Reduction phase

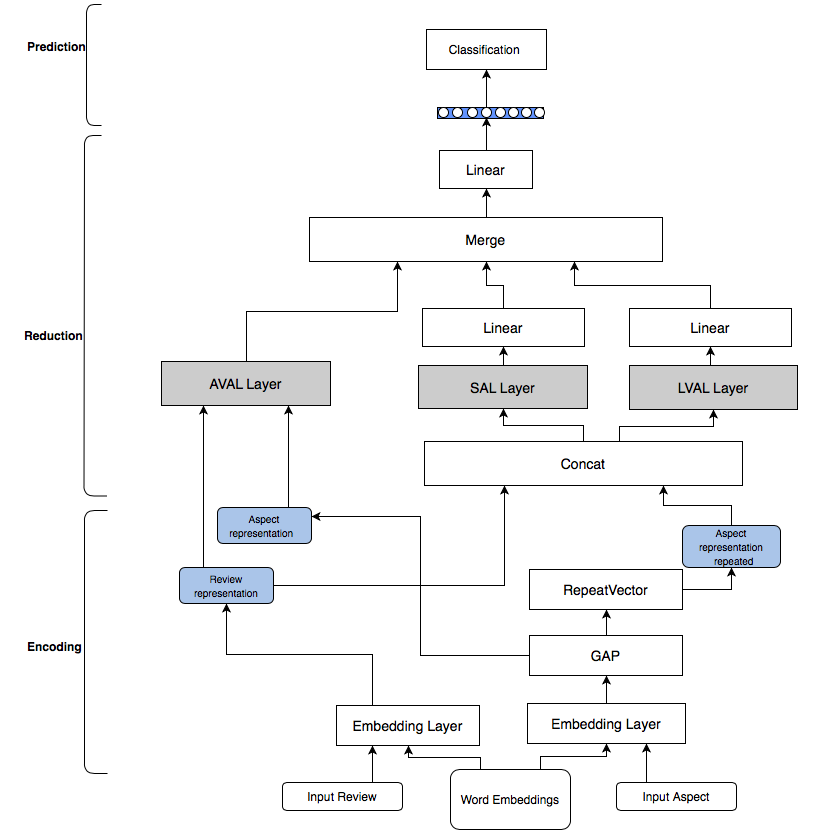
In this phase, three hidden review representations will be generated independently by using three different attention-based layers. Review distributed representation and aspect repeated representation going to be concatenated and fed into SAL and LVAL layers. These layers going to calculate the most informative words in a given review with considering an aspect, as its representation is explicitly included in the input. The output dimension of these two layers will be vector with shape that is going to be reduced into by using linear layers. AVAL attention layer is using review representation along with GAP layer output of aspect representation to calculate a weighted score representation for a given review. Then, averaging representation of all of these three hidden representations going to be generated. Finally, passing the generated vector into Maxout Layer (Goodfellow et. al, 2013a), to get the final review representation. It is simply a linear layer where the activation function used is the maximum of the input.

Figure 5. 5: PAO model architecture

# 5.3.1.3 Classification phase

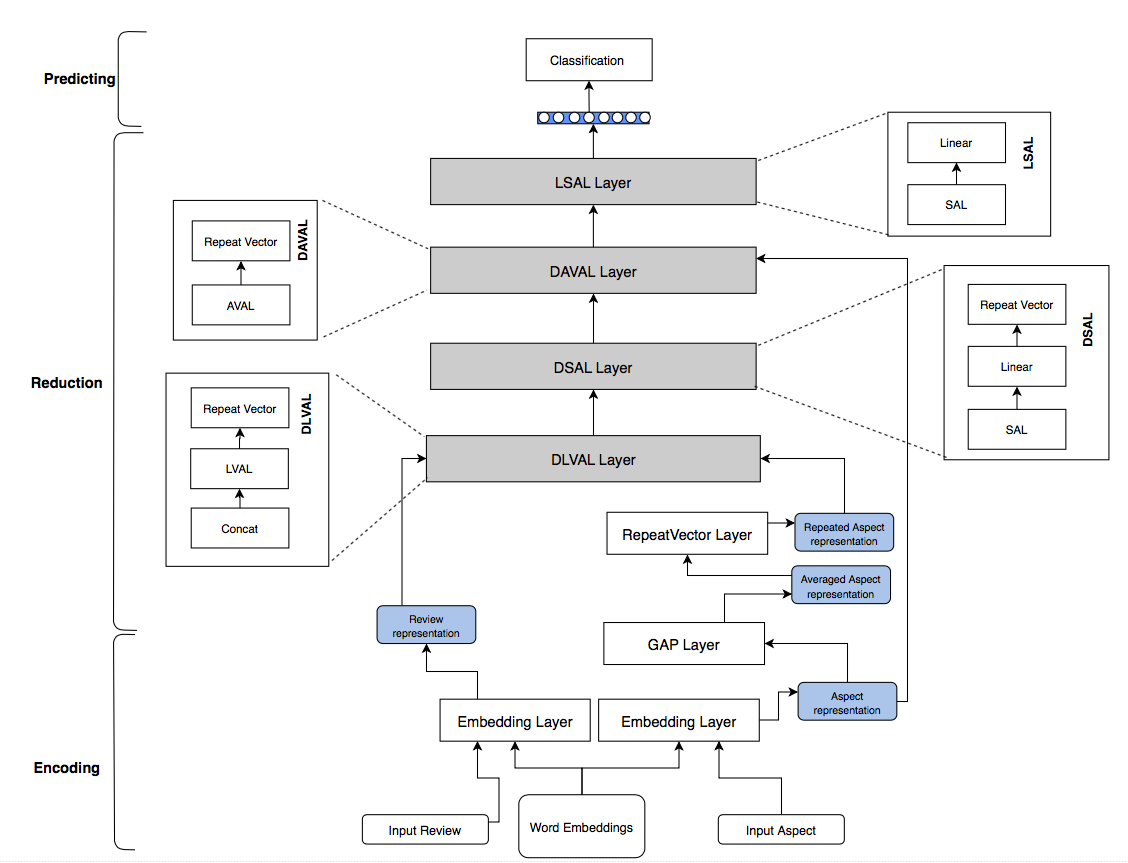
To generate a probability distribution over all three classes, Maxout linear layer’s output is passed into the final fully-connected softmax layer.

Experimentally, the effect of using multiple hops over long-term memory on the performance will be investigated. Memory representations are jointly learned with network training process.

# 5.3.2. Deep Hierarchical Attention-Only Model (DAO)

Most of aspect based sentiment analysis (ABSA) researches conducted try to identify the most informative words in response to a specific aspect (Al-Smadi et. al, 2015; AL-Smadi et. al, 2016; Kumar et. al, 2016; Ruder et. al, 2016a; Jebbara and Cimiano, 2017; AL-Smadi et. al, 2019). However, most of these proposed systems suffer from different problems that affect the overall accuracy. Considering the context of the review partially is one of the current drawbacks of the proposed models. We should have a model that captures the global context information by considering the examined aspect while generating review representation. In addition, most of attention-based model are considering word-level scoring scheme without taking into account the whole sentence semantic meaning. Moreover, considering aspect embedding is crucial to calculate a good review representation correlated to a specific aspect. Previous researches conducted to solve this issue and considering the aspect information go in two paths, using memory-based models (Tang et. al, 2016) or considering aspect-hidden states based model (AL-Smadi et. al, 2016; Ruder et. al, 2016a; Jebbara and Cimiano, 2017; AL-Smadi et. al, 2019). In this proposed model, we will try to elevate these problems by constructing a hierarchical model and simulating the linguistic characteristics of the sentence by learning the review representation in different level of abstractions. Each sentence consists of several phrases that in turn consist of words. By applying the hierarchical metaphor of sentence compositionality, we hope to get better review representation in response to specific aspect.

The architecture of the proposed model is illustrated figure 5.6. Inspired by (Wang et. al, 2016), the model is considering the aspect information by including the aspect embedding with the review embedding. In addition, we will incorporate the aspect information by having context-based vectors that learned through out the model with different level of abstraction.

Figure 5. 6: DAO Model Architecture

# Model Details

Let denotes an input review with *n* words and an aspect word *wi*occurring in *r*. Predicting the polarity of the review *r* towards certain aspect *wi* is the goal of the model.

As the encoding-reduction is a metaphor for all proposed models in this thesis, in DAO model the encoding phase will be considered when the review intermediate hidden representation is generated. The reduction phase is when this intermediate review representation enhanced and then fed into the classification phase. Finding the most informative words in the review toward an aspect is the overall objective of the model. The word importance in any review is highly dependent on its context. To include this fact into our model, the model first scores the input review by considering the whole meaning of the sentence.

After conducting preliminary experiments to decide on the depth of the architecture, it has been found that the best accuracy of deep architecture is achieved if the model deepens into four attention-based layers. By using two embedding layers, we will have the distributed vectors representation for both review , where d is dimension number and k is the maximum words number in a review, and aspect z is the maximum words number in input aspect. The first step is to learn a memory vector that measures how important a word is at each time step to a specific aspect. The embeddings’ representations are going to be concatenating and fed into the first attention-based layer used, which is Learned Vector Attention Layer (LVAL). This layer will learn a context-based scoring vector for the entire sequence and serve as context representation that gives high weights to the most informative words in this context-level. The output of this layer answers the question about the most informative words in local context but it does not reflecting the meaning of the entire review. There can be multiple contexts in a sentence that forms the overall semantics together. Thus, focusing on different parts of the review is needed, by incorporating different scoring functions. To extend the context, SAL layer is applied onto the generated output to scan the first attention layer output again and smooth the score salient features from global perspective. The output from this layer could be considered as an intermediate review hidden representation.

Then, in Aspect Vector Attention Layer (AVAL), aspect distributed representation vector is going to be incorporated into the scoring process based on previous SAL output. To increase the depth of our deep learning model and add more abstraction to the final representation, the output of AVAL will be passed again to non-linear SAL layer with the current time-step word embedding and aspect representation. By reaching the final layer, final review representation is calculated by using smoothed importance score with considering an aspect from a global perspective.

# 5.4 Discussion

Models proposed in this thesis share some of the implementation found in other works and some were the first that applied for Arabic aspect-based sentiment analysis. Work done by (Ruder et. al, 2016a) has proposed ABSA recurrent-based neural network, without using any attention-mechanism, for multi-language including Arabic. Similar to our work, (AL-Smadi et. al, 2019) used soft attention to classify Arabic aspect-based sentiment analysis with LSTM-based network. Unlike our models, they concatenated aspect words’ representation with the reviews’ word representation after LSTM layer. The logic of DLVAL had been implemented previously by (Tang et. al, 2016) for English language but this work is the first work that implements the concept for Arabic language. In addition, the proposed work incorporating the aspect representation to calculate the attention weights, proposed in DAVAL model, which is the first time used in sentiment analysis field. This work is discriminated also by proposing models that are based mainly on the previous used attention layers, either in parallel or hierarchical manner, which was not proposed before in this way by any other work. Finally, in this work attention mechanism is used and examined by using different logic to open the black box of deep models and analyze its outcomes to conduct a comprehensive analysis of the network behavior, which had not been examined by any other Arabic sentiment analysis research.

# 5.5 Summary

To conclude this chapter, different proposed aspect-based sentiment analysis models were proposed by using attention mechanism to classify Arabic reviews with respect to an aspect. The chapter discussed the methodologies used in designing these models and the objectives behind each model. The chapter proposed three deep attention-based aspect-based sentiment analysis model including DSAL, DLVAL and DAVAL. In addition, the chapter highlighted the importance of considering aspect when calculating the final review representation and presented the design of proposed attention-only models including PAO and DAO models.

# Chapter 6: Experimental Setup

This chapter will present and explain the experimental setup followed, tools, dataset and training details to implement the proposed models. Section 6.1 is discussing the experimental setup for DARLSA model. Then, in section 6.2 experiments’ settings for DAALSA is presented. Finally, in section 6.3 the evaluation process and the qualification process are discussed. The evaluation process is the process of estimating the sentiment distribution for each class, while the qualification process is the process of visualizing the salient words considered by a model for each classified review. Section 6.3.1 is presenting the evaluation process methodology followed for all models. Then in section 6.3.2, the methods used in the qualification analysis process are discussed. Finally, section 6.4 concludes this chapter.

# 6.1. Deep Attention-Based Review Level Sentiment Analysis Model (DARLSA) Experimental settings

The baseline GRU-based model and DARLSA model were trained and examined with five different cases for word embedding initialization. ARLSA model was trained and examined with pre-trained word embedding initialization. For all DARLSA models, the same hyper-parameter settings were used for all models, with the optimal configuration that suits our existing hardware capability.

# 6.1.1 Dataset

The dataset used is extracted from LABR (Aly and Atiya, 2013), a two classes’ polarities, negative and positive Arabic book review dataset. The dataset were collected reviews from Arabic books’ reviews website[[5]](#footnote-5), from 16,486 users for 2,131 different books Reviews with one and two rating were considered as negative one, reviews with four and five ratings were considered as positive one and reviews with three rating were neutral but not included in the dataset. For training phase, 12,336 reviews were used with 6,119 positive reviews and 6,217 negative reviews. 4,111 reviews were used for testing and validation purposes. The length of each review is set to 460 words, as it is the calculated mean of the reviews’ length in the dataset. Short reviews were padded, and reviews longer than the mean were clipped to this length.

# 6.1.2 Pre-trained word embeddings

The effect of having pre-trained embedding models to initialize embeddings’ layer weights is examined. Two different Arabic pre-trained word embeddings were used. The first examined Arabic pre-trained embeddings model is a CBOW 300 dimensions word representations publicly proposed by (Zahran et. al, 2015) generated by training a large Modern Standard Arabic dataset including more than 5.8 billion words.

Additionally, the effect of using different pre-trained word embeddings is examined by using another pre-trained Arabic distributed word representation. (Altowayan and Tao, 2016) 300 dimensions trained CBOW word embedding is used. Their word embeddings were generated by training a model on a large corpus with around 190 millions words. The dataset collected is enriched with dialectal vocabulary from different spoken Arabic. The initialization of embedding layer weights is achieved by using these pre-trained word vectors’ weights. Additionally, the results acquired by the two embeddings initialization are compared with the case were no pre-trained embedding used. The weights of the embedding layer were initialized by uniform initialization (Glorot and Bengio, 2010) and also used to initialize weights for unknown words in the case of using pre-trained models.

# 6.1.3 Training details

The model had been trained in an end-to-end way by using back-propagation with mini-batches of size 32. Minimizing loss function, cross-entropy loss function is used, is the goal for the training all models, and the target distribution for review is the predicted sentiment distribution. The optimization function used is Adam (Kingma et. al, 2014), with learning rate 0.001 and clipped norm of the gradients at 5 (Bengio et. al, 2013b). A sigmoid layer with two outputs, for positive and negative sentiment, is used in the output layer.

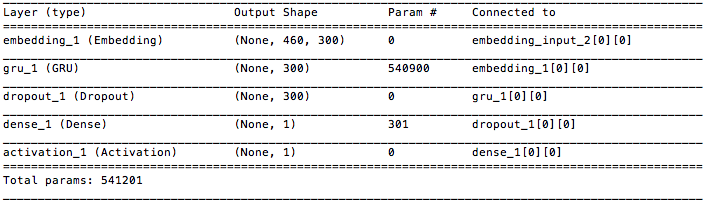
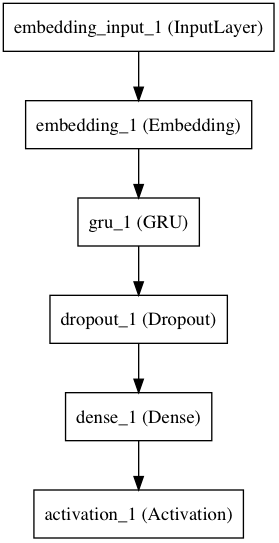
DARLSA and baseline models were trained with 30 epochs, as in the preliminary experiments the models was not improving after epoch 30 for the dataset size used. Early stopping technique is used as well, when training will be stopped if there is no progress on the validation set after 10 consecutive epochs.

# 6.1.4 Comparison models

Different models were developed and the accuracy of the classification for these models was compared.

# 6.1.4.1 Baseline

As shown in figure 6.1, the baseline model, using pre-trained embedding, is a GRU-based neural model with 541,201 parameters. Examining the power of using attention layer is the main objective of this experiment. The model uses 300 dimension for embedding layer and GRU memory cells. 200 dimensions is experimented for GRU layer size but the performance was better with 300 dimensions for our current dataset’ vocabulary size. This is agreed with result found in a research done by (Bansal and Sangeet, 2018) that concludes that accuracy of the classification results is increased with the size of the vector dimensions. This is because more dimensions means more accurate features’ similarity relations could be drawn. Dropout technique is used by switching off randomly number of GRU units (with probability 0.2) and between layers (with probability 0.4) during training to prevent over-fitting (Srivastava et. al, 2014).



(a)

(b)

Figure 6. 1: GRU-based Model. (a) Model Plot (b) Model Summary

Figure 6.2 illustrates all experimented cases. The same experiment was repeated with three different techniques to initialize weights for embedding layer to get five different cases for the baseline model. The first experiment used (Zahran et. al, 2015) embeddings and second experiment used (Altowayan and Tao, 2016) embeddings. Two modes were experiments for both pre-trained models, either fine-tuning the weights of the pre-trained models during training or keep it unchanged. Finally, the third experiment does not use any embeddings value for embedding layer weights. It randomly initializes layer’s weights with normal weight initialization without using any pre-trained model.

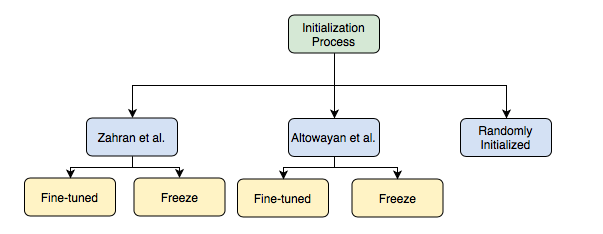
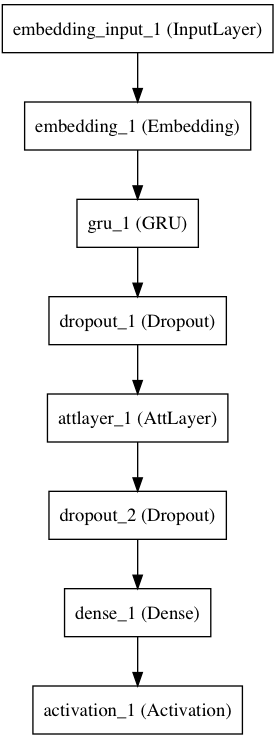


Figure 6. 2: Different Initialization Cases

# 6.1.4.2. DARLSA

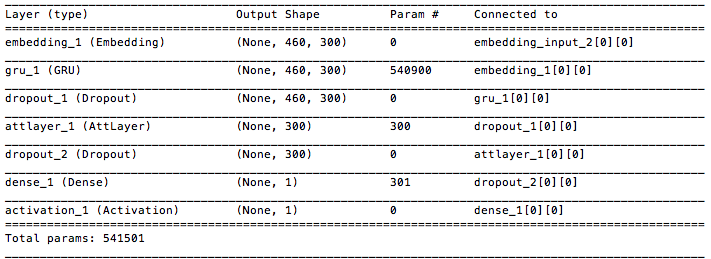
The model plotted in figure 4.2 is experimented and the results were observed. The four cases used in baseline experiment to initialize the embedding layer are examined for this model as well. It used 300 dimensions for GRU layer and also for the final review representation. By changing GRU layer configuration and letting the layer, for each input time step, return one output, the layer output is changed into 3D array and used as input for subsequent attention layer. The dropout rate used is 0.2 on the GRU units and the rate was 0.4 between layers. Figure 6.3 is the Python detailed model summary and plot.

Figure 6. 3: DARLSA Model. (a) Model Plot, (b) Model Summary



(a)

(b)

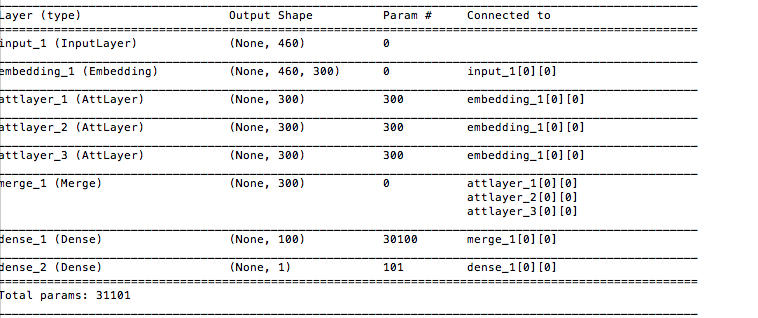
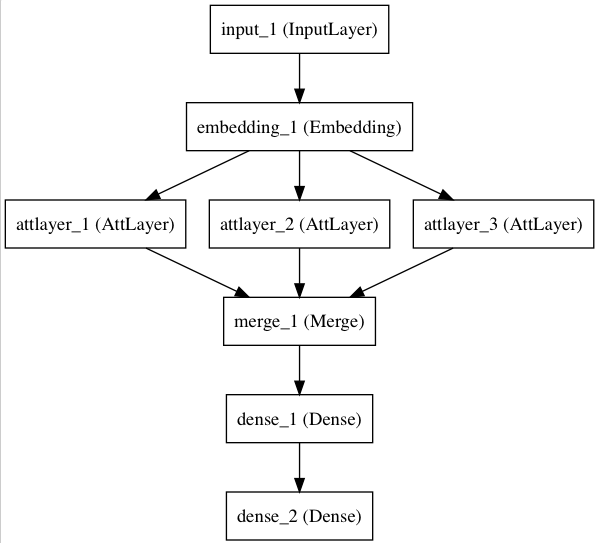


# 6.1.4.3. ARLSA

In attention-only based model experiments, the embedding layer weights initialized using 300 dimensions (Zahran et. al, 2015) pre-trained embedding. The experiment used the model plotted in figure 4.3 with total parameters of 31,101. Figure 6.4 is Python model plotting and detailed model summary.

# 6.1.5 Tools

The experiments were run on an iMac computer with 3.4GHz GPU and 32GB of RAM. Python 2.7.8 is used to implement the proposed model, with Keras 1.1 (Chollet, 2015) using Theano 0.8.2 as backend (Al-Rfou et. al, 2016).



(b)

(a)

Figure 6. 4: ARLSA model. (a) Model plot, (b) Model summary

# 6.2 Deep Attention-Based Aspect-Level Sentiment Analysis Model (DAALSA) Experimental Settings

In DAALSA, five different models were trained and examined, beside the baseline model, using the same hyper-parameter settings with the configuration that meets our existing hardware capability. The following sections discuss the experimental setup in details.

# 6.2.1 Dataset

Proposed models were trained and tested using the Arabic Hotels reviews dataset that was prepared as part of the SemEval-2016 Aspect-based Sentiment Analysis task (Pontiki et. al, 2016), using the provided train and test splits. The Advanced Arabic Text Mining group at Jordan University of Science and Technology technical team did the annotation task for the Arabic dataset. The model uses a total of 10,509 ABSA annotated tuples extracted out of this dataset, where 8,407 samples were used in the training phase, 1,051 for validation and 1,051 used for testing purposes. The reviews were repeated and unrolled in the dataset with every aspect with appropriated sentiment label. Each review may fall in one of three available sentiment polarities, positive, negative or neutral. The dataset contains 3629 negative, 683 neutral and 6197 positive reviews.

# 6.2.2 Training details

All models were trained with objective function of minimizing the cross-entropy, with mini-batches of size 32. Back-propagation was used to train the models with stochastic gradient descent. The optimization function used is Adam, with 0.001 as learning rate (Kingma et. al, 2014), and to fight against gradient explosion, the norm of the gradients was clipped at 5, which is the threshold that gives the best results in the preliminary experiments (Bengio et. al, 2013b).

(Altowayan and Tao, 2016) 300-dimension pre-trained word embedding was used to initialize the embedding layer’s input words for all models. For the deep attention-based models with 200 dimensions for each GRU layer, where 300 and 200 options were examined but with 200 the performance were better. One potential reason is that the word embedding for words in the review are going to be concatenated with word aspect distributed representation to get 400 dimension in total, which may put the model in an under fitting situation if the neurons exceeded a certain point. In other words, 400 dimensions are enough to express our dataset than 600 dimensions. The models were trained with 30 epochs, as no improvement is achieved after it, with early stopping technique. For attention-only models, 50 epochs were used as the models stopped to improve after that.

# 6.2.3 Tools

The experiments were run on iMac computer with 3.4 GHz GPU and 32GB of RAM. The code is implemented in Python 2.7.8 with Keras 1.1 (Chollet, 2015), using Theano 0.8.2 as backend (Al-Rfou et. al, 2016).

# 6.3. Performance Monitoring

Despite the fact that deep neural network is efficient, it is non-linear model as well, which leads to a lack of transparency that is considered one of its drawbacks. Therefore, one of the objectives of this work is trying to add an interpretability means to the current models. In the conducted experiments, several visualization tools are utilized to open the black box of the proposed deep models and to understand the underlying dependencies between the input text and the output classification score.

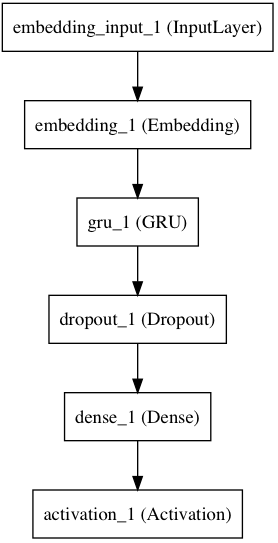
By going through the visualization process, evaluation and qualification results were generated. Better understanding could be achieved about the proposed models and their functionality by analyzing these outcomes. In the evaluation process, the training process progress is monitored and the overall results are compared with other models’ results to get the efficiency of each model. In the qualification step, different learned reviews representations are visualized and compared to get intuitive explanation of the trained models.

# 6.3.1 Evaluation Step: Process Visualization

Model visualization and training process visualization are two tools used to get more information about the trained models. In model visualization, the structure of each model is plotted. By visualizing and monitoring the training process information, better understanding of the model could be achieved and leads to well training deep learning models (Zeng, 2016).

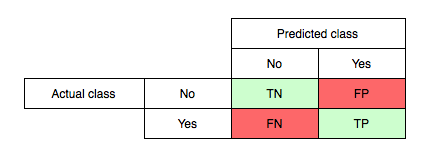
# 6.3.1.1 Model Visualization: Neural Network Structure- Block Diagrams

Figure 6.5 is an example of the block diagram. In block diagrams, each layer is replaced with a solid block connecting by a single arrow line.

Figure 6. 5: Review Level Sentiment analysis Baseline Model Block Diagram

# 6.3.1.2 Training Process Information Visualization

Generally, two phases are included to train a supervised machine learning algorithm. The first phase is the training phase where the model is trained by using examples from the dataset. During testing phase, the model is predicting the polarity for unseen examples and then comparing the predicted value with the desired one to calculate models performance numbers. The outcomes of the prediction process could be one of four cases, either false positive (FP), false negative (FN) true positive (TP) or true negative (TN) predictions. These numbers could be analyzed by using confusion matrix, shown in figure 6.6, which is a table used to measure model performance and gives details of number of success and failed classifications for each class. A good classification model should minimize the value of false positive and false negative (Elkan, 2013).

Figure 6. 6: Confusion Matrix

Four numbers are used to evaluate the performance: accuracy, recall, precision and F-measure. Accuracy is measure how accurate is a classifier. It is defined by the following formula:

Using precision and recall metrics is highly recommended especially if the dataset is unbalance. Precision measures the probability that a positive prediction is really positive (Jebbara and Cimiano, 2017), and defined by:

While recall measures the efficiency of a model to find all positive reviews in a dataset,

In addition, a weighted harmonic mean of precision and recall is calculated to get F-measure. It is defined by:

Training information is visualized to get some insight about the training process and to help designing and debugging a better deep learning models. Basically, the training information is used during and after training. During the training process, monitoring the output data, such as loss function and classification accuracy for each epoch helps us detect any exceptions that lead to refine the model. After training each model, the training information is stored and then retrieved and visualized to compare its efficiency with other models (Zeng, 2016).

Calculating error rate is the simplest way to measure the performance of the model. The methodology is to compare the final results with the desired one over time and visualize it. This error shows the ability of the model to learn and converge. Also balancing the training accuracy versus testing accuracy will prevent the model to over fit.

# 6.3.2. Qualification Analysis: Feature Visualization

In deep neural network, a hidden layer output, vector of activations, is the distributed representation of input review. In RNN-based neural network, it is called hidden states. Many researches have proven that these hidden states can capture the important information, or salience features, in the input text (Zeng, 2016). By visualizing these vectors we can reveal what neurons have learned.

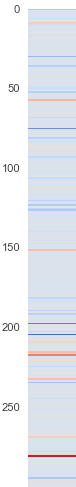
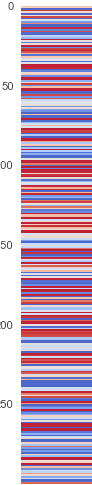
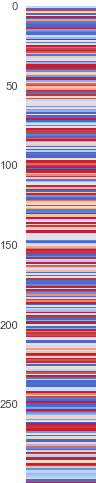
In the qualification analysis phase, different techniques could be used to understand what information these networks have captured. Next sections will discuss visualization methodologies used to evaluate models’ classification results.

# 6.3.2.1. Review Representation

Mainly, all the proposed models are trying to generate a review representation that best depict the semantic meaning of the input. Attention layer is generating the overall sentence representation and pass its hidden representation to the output layer to calculate the overall score that identify the polarity of the review. Therefore, the review representation drawn by the model is plotted in a heatmap representation (Li et. al, 2016; Karpathy et. al, 2015). Heatmaps are generated to visualize the saliency words that the model tried to capture during the classification process. But because the representation heatmap is hard to interpret, representations for different reviews with different structures are plotted to get some insights about the behavior of the model.

Therefore, the methodology followed is trying to find a pattern in the heatmap by linking each representation with its corresponding input review and comparing it with other reviews heatmaps to understand the input text characteristics captured by the model. Figure 6.7 is an example of two heatmaps for a review and its negation. The difference between two heatmaps is plotted as well to detect the dimensions that are responsible for the negation.

Figure 6. 7: GRU layer hidden-representation heatmap.(a) Review heatmap. (b) Negated review heatmap. (c) Representation difference



(c)

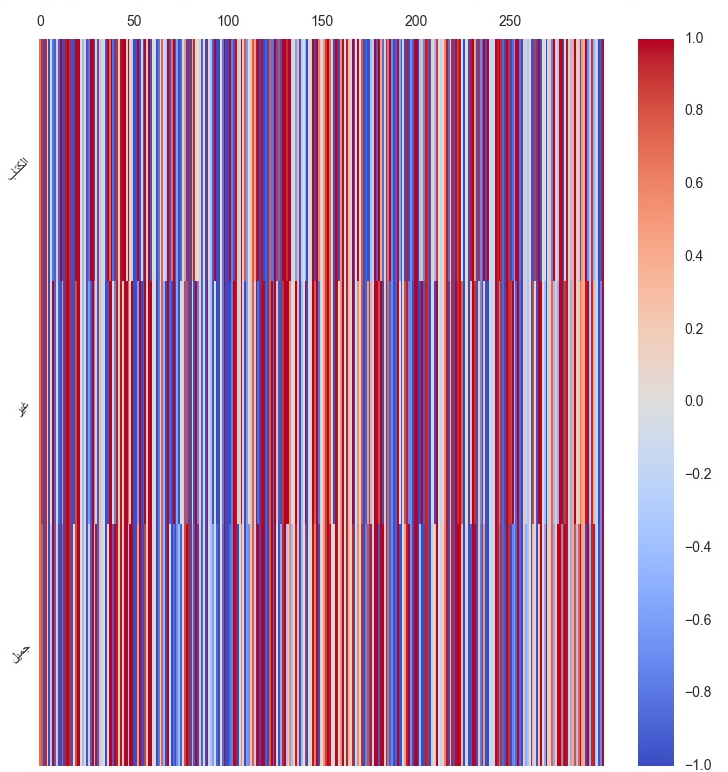
(b)

(a)

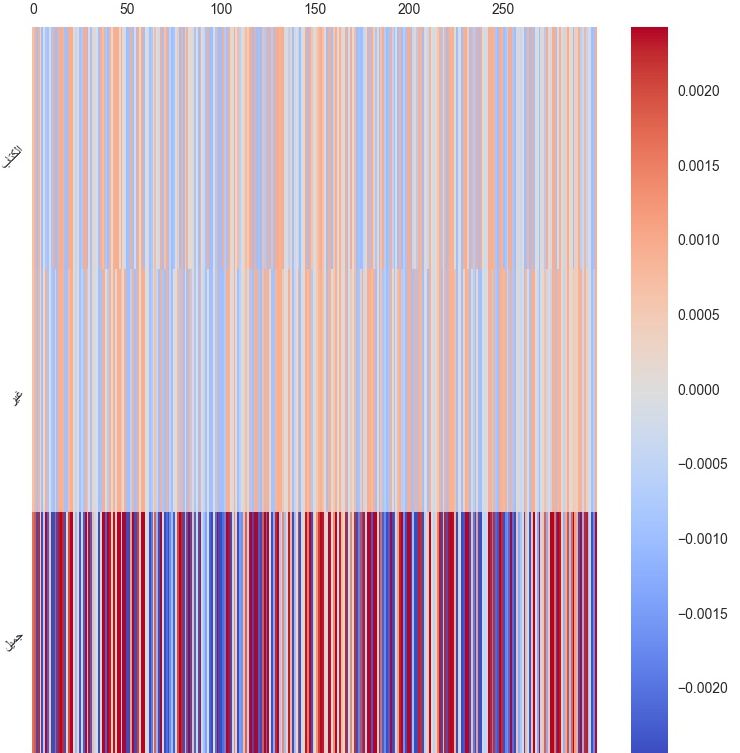
Also, plotting heatmap for representation overtime may help to reveal some informative facts about model behavior. Figure 6.8 is an example of review representation over-time generated as an output from baseline GRU-based model, and (b) review representation over-time generated from DAALSA for the same review.

Figure 6. 8: Review representation overtime (a) Baseline GRU-based model. (b) Attention-based DARLSA.

(a)



(b)



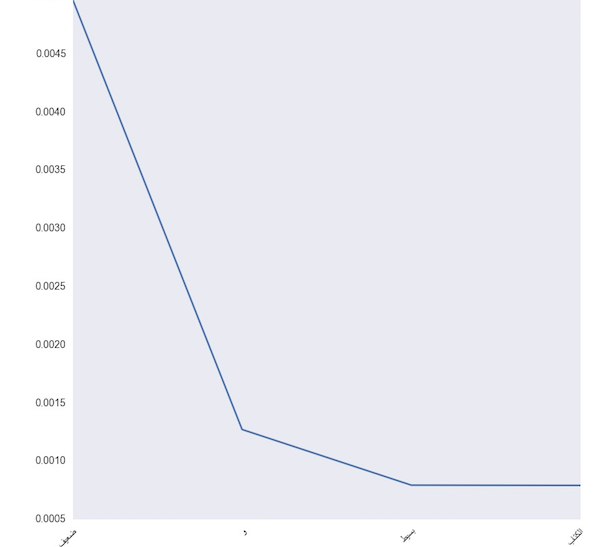
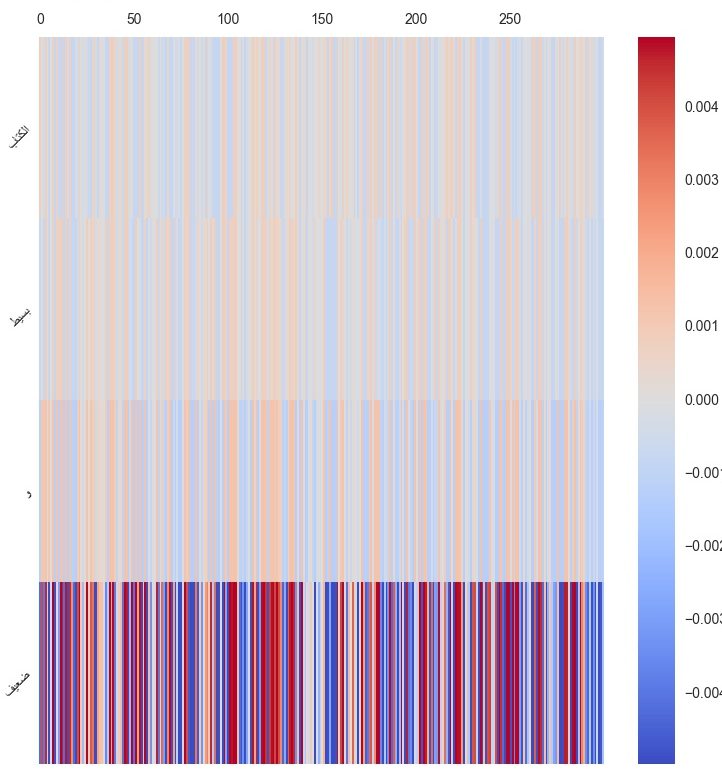
# 6.3.2.2. Attention score

As seen in the previous section, review representation heatmaps are hard to understand without comparing them with other heatmaps. In addition, we need to find out what are the most influence words to the classification process as it is considered one of our main objectives. The output value from the attention model is examined to check whether the model can detect the salient words in a given review or not. In attention-based models, the model calculates a score to measure the importance of the current word to the overall sentiment of the review, at each time step. In order to achieve the visualization process, the score calculated by the attention layer will be returned at each time step to plot the local composition of the input review. By visualizing the attention weight matrix, the model could visualize the importance of each word to the classification process and hence adding extra output interpretability layer of a deep neural network.

Figure 6.9 are four different plotting for the same review. Plot (a) is the attention weights score word importance, where red is the highest score and dark blue is the least. Plot (b) is another representation for the same vector; 2D plotting where the words of the review are the labels of x-axis and y-axis is the attention score value. Plot (c) is a plotting of heatmap representation of the review representation overtime generated as an output of the attention layer. Finally, plot (d) is another simplified representation of plot(c), where each representation for each word are averaged to get one scalar number score for each word in review level to show how each word contributes to the final review representation.

Because our main objective is to reveal the most informative words for each model from the overall review representation, the only way to know if the review representation is presenting the real score of importance is to compare it with attention score importance scale. As we can see from figure 6.9, (a) and (d), each of them has its own scale that makes it illogical to compare them. Therefore, we should have a uniform scale to make the comparison acceptable. Figure 6.10 shows the two representations with uniform scale starts from the minimum value in both vectors and peeks into the maximum value for each.

Figure 6. 9: DARLSA Model Visualization Plotting for the Same Review



(d)

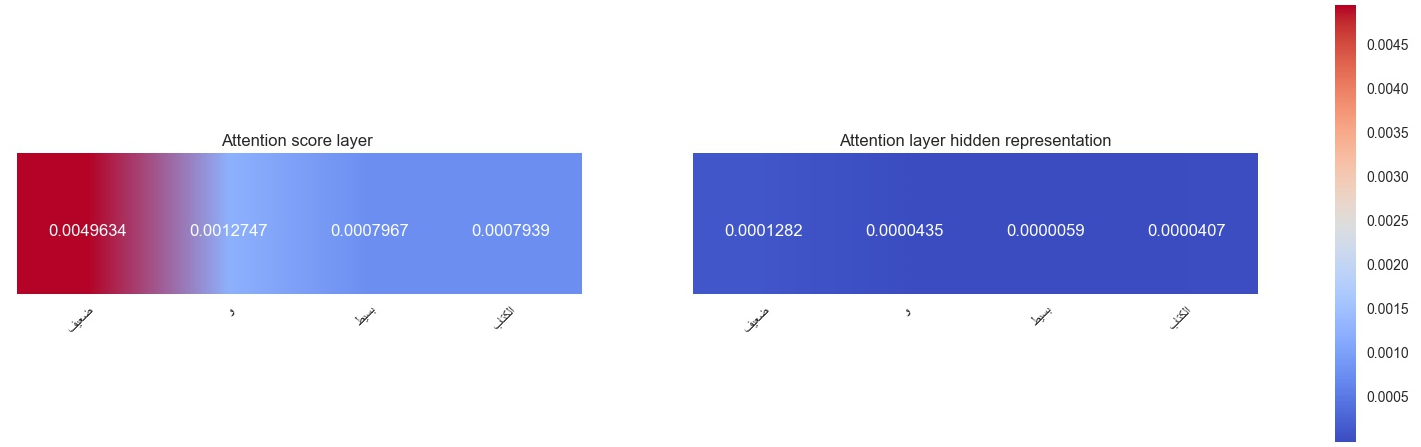


(c)

(b)

(a)



Figure 6. 10: Uniformed Scaled Score

To achieve the visualization objective programmatically, a new model is defined and truncated just at the attention layer to get its outputs, then the weights of the pre-trained model is loaded. Next, the sentiment for a given review can be predicted and got its activation values.

In attempt to generate more readable plotting, the words importance local score is dumped into an html file and highlighted according to its importance as shown in figure 6.11, reds get more attention than greens.

Machintosh:Users:Nada:Desktop:Screen Shot 2018-11-29 at 3.52.00 PM.png

Figure 6. 11: Html representation for attention scores

# 6.4 Summary

This chapter has listed tools, dataset and training details done to implement the proposed models. In the first two subsections, sections 6.1 and 6.2, the experimental setup followed to implement DARLSA and DAALSA are discussed. Finally, this chapter discussed the methodologies used in the evaluation process by the means of approximating the sentiment distribution for each class, and the tools used to support the qualification process by visualizing the salient words ranked by their importance measured by the attention layer in each model. The evaluation of these models will be discussed and compared in detail in chapter 7.

# Chapter 7: Results and Analysis

The results of experiments are discussed in this chapter and the outcomes are analyzed. Section 7.1 is documenting and analyzing the evaluation and qualification results for all DARLSA models. Section 7.2 is discussing and investigating DAALSA results. Finally, in section 7.3 a conclusion and discussion for both models is drawn.

# 7.1 Deep Attention-Based Review Level Sentiment Analysis Model (DARLSA): The Evaluation and Qualification

This section will demonstrate the results of DARLSA model experiments. Section 7.1.1 demonstrates the evaluation results for models’ outcomes and section 7.1.2 is discussing the qualification for the experimented models.

# 7.1.1 Evaluation Process Results

Table 7.1 is listing all testing performances for all DARLSA models. There are mainly three models to compare with different settings. The first model is the baseline model where a GRU-based model is defined and used to classify sentiments. The second model is attention-based DARLSA model, and the third model is Attention-only Review-Level Sentiment Analysis Model (ARLSA). Different word embedding initialization techniques were used. Two pre-trained embedding weights, (Zahran et al., 2014) and (Altowayan and Tao, 2016), were examined by having two cases either freezing or fine-tuning the weights during training. Also the case of random initializing the weights without using any pre-trained weights was examined as well. These numbers are analyzed from different perspectives. Next sections are discussing and analyzed all cases.

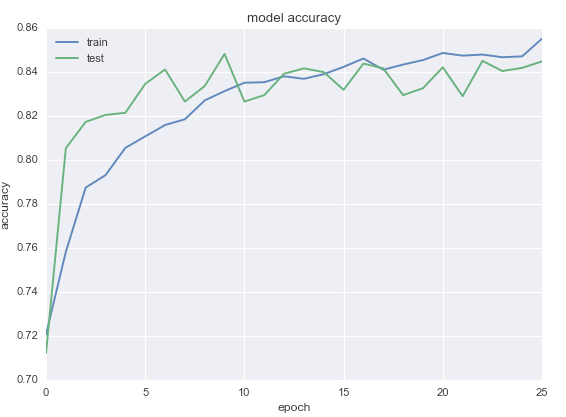
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Pre-trained Word Embedding** | | **Acc.** | **Precision** | **Recall** | **F-measure** | **Trainable Parameters no.** | **Average Time/ epoch** |
| Baseline | Zahran et al. | | .87 | .88 | .86 | **.87** | 541,201 | 90s |
|  | Altowayan et al. | | .84 | .84 | .82 | .83 |
| Fine-tune | Zahran et al. | .83 | .84 | .81 | .82 | 9,242,401 | 90s |
| Altowayan et al. | .84 | .82 | .86 | .84 |
| No pre-trained | | .82 | .81 | .83 | .82 | 9,541,201 | 100s |
| DARLSA | Zahran et al. | | .85 | .85 | .85 | **.85** | 541,501 | 95s |
|  | Altowayan et al. | | .84 | .82 | .88 | .85 |
| Fine-tune | Zahran et al. | .84 | .82 | .87 | .84 | 9,542,401 | 90s |
| Altowayan et al. | .84 | .81 | .88 | .84 |
| No pre-trained | | .81 | .86 | .84 | .84 | 9,541,501 | 100s |
| ARLSA | One level | | .78 | .77 | .79 | .78 | 601 | 4s |
| 3 Parallels | | .82 | .82 | .81 | **.82** | 31,101 | 7s |

Table 7. 1: Evaluation Process Results for Proposed DARLSA Models

**Effect of Model Architecture**

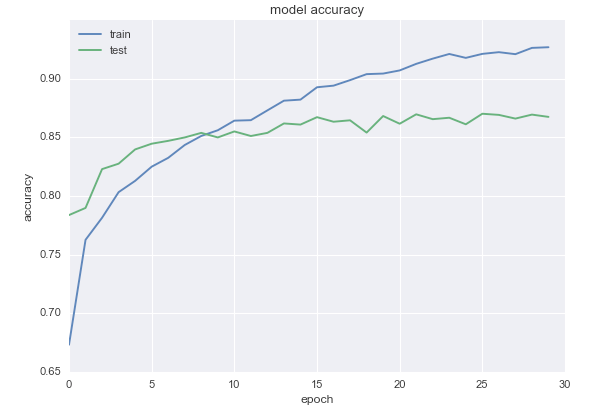
As seen from table 7.1, the best-achieved result, number wise, was the baseline model with (Zahran et al., 2014) pre-trained word embedding weights with f-measure of 87%, compared to 85% achieved by DARLSA model, whereas it got 83% when using (Altowayan and Tao, 2016) embedding layer weights. A possible reason of the outperformance of the baseline model because of the attention model has larger parameters compared to the baseline model, which is harder to optimize on small size dataset. Looking at the learning curves, as seen in figure 7.1, we found out that the baseline model has a tendency to over-fit. By the end of epoch 30, the training accuracy exceeded 92% where the testing accuracy still in the range of 86%. Comparing this result to attention-based proposed model where the training accuracy and testing accuracy at epoch 30 are both in the range of 85%. Attention-based proposed approach has achieved F-measure of 85% with both (Zahran et al., 2014) and (Altowayan and Tao, 2016) pre-trained embedding. For the dataset used, (Zahran et al., 2014) pre-trained word embedding has lesser rate of out-of-vocabulary rate with 1,607 compared to 5,973 for (Altowayan and Tao, 2016) pre-trained embedding. Comparing to existing methods that using the same dataset, it can be shown that GRU-based baseline model outperform their work and DARLSA got a comparable performance with (Dahou et. al, 2016) that achieved an accuracy of 86% on a balanced version of LABR dataset using CNN. Comparing DARLSA performance with the model presented by (ElSahar and El-Beltagy, 2015), a significant improvement is achieved as they achieved an accuracy of 78% using linear-SVM classifier on the same dataset.

Figure 7. 1: Models Training Accuracies (a) DARLSA. (b) Baseline



(a)

(b)



**Effect of pre-training**

For the transfer learning experiments, it has been proven that using pre-trained embedding helps the model to achieve higher accuracy compared to a model that does not use pre-trained weights. Looking at table 7.1, we found out that in the case of baseline model, using (Zahran et al., 2014) pre-trained embedding is achieving an F-measure of 87% and 85% with the attention-based model. Looking into the number of parameters required for each case is another aspect that should be highlighted. In the case of using pre-trained model the number of hyper-parameters required are 541,201 with baseline model and 541,501 for attention-based model, because of the 300 parameters used by attention layer. In the absence of applying transfer learning case, the number will be as high as 9,541,201 for baseline model and 9,541,501 for attention-based model, which means that more time and space are needed in the training process. However, the models achieved 82% for baseline model and 84% for attention-based model with very high tendency to over fit when no pre-training weights are used. In addition, with the case of fine-tuning the pre-trained model into models at hand, it has been found again that the models tends to over fit for all cases.

In addition, we found out that using only attention layers to generate review representation achieved acceptable results with F-measure of 82%, run for 30 epochs. Each epoch took 4 seconds to finish for one attention level layer and with only 601 trainable parameters. On the other hand, in the case of three parallel attention layers the model run for only 30 epochs and stopped because of early stopping condition. Each epoch needs 7 seconds to finish with 31,101 model trainable parameters. However, we cannot decide on the best-proposed model until we moved to the next level of analysis, the salient words analysis for each model, i.e. qualification analysis.

# 7.1.2 Qualification Process Results

As discussed in section 1.2, one of the objectives of this work is to open the black box of deep models decisions and to highlight the main saliency features that led to the classification phase results. Different objectives were studied by examining different reviews’ cases to help reaching the main objective of the presented work, which is producing a model that understands the semantic of a given review. Several semantic related testing are conducted to draw a conclusion about the overall performance of a given classifier.

Detecting context meaning of an adjective was the objective of the first sets of experiments. Then, the ability of the system to handle negation and intensification is detected. The methodology used to evaluate each case study goes into two directions. First, examining the ability of the model to detect polarity for the whole review, i.e. examining the prediction phase results. In addition, detecting the most informative words in the review is investigated as well, i.e. examining the reduction phase.

During testing phase, a review was fed into the model to predict its class and then different visualization techniques were used, as mentioned in section 6.3, to give different interpretation for a given result. For simplicity purpose, in the following sections only html representation along with its scores will be presented. For hidden representation overtime, the representation is summed and averaged to get a scalar number, and then these numbers were used to plot a colored representation of the review’s words depending on the intensity. The red color is a high intensity degraded into green to be the lowest. For attention-based DARLSA both local word importance and unified scales attention layer review representations are plotted. The local importance is the ranked intensity of the words in one vector, while the unified importance is ranking the importance against two vectors, the weights scores and representation overtime scalar vectors. The local importance is plotted in the mean of html representation. While the unified scaled is plotted as a numbered scale just under the local importance representation to make the comparisons.

The test cases are handcrafted carefully to test the objectives listed above. Each of these cases is examining and helping to verify a linguistic feature. In every section, the purpose of the test will be explained first, and then the test case will be presented along with results. Table 7.2 lists 17 modeled, numbered test cases, with their English translation and Arabic-Latin transliteration.

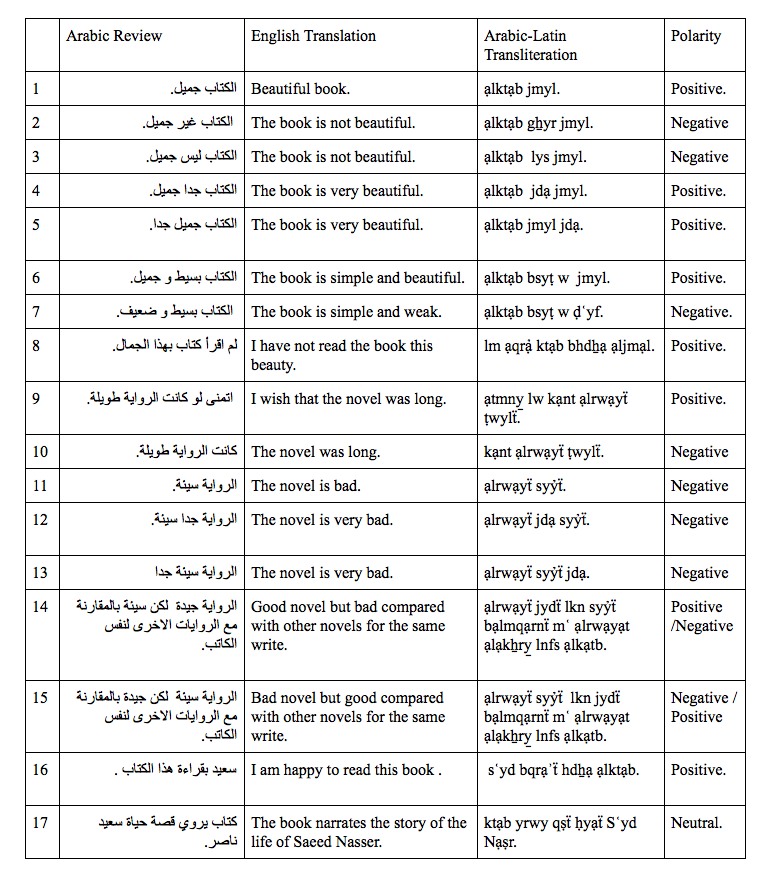


Table 7. 2: List of DARLSA Test Cases

In the following sections, an analysis for each case is discussed by describing first the test objective, then the results for all examined model will be listed.

# 7.1.2.1 Word’s Context Test: Objective

The objective of this test is to examine the ability of the model to detect the correct polarity for the same word placed in different contexts. The first test case is examining reviews number 6 and 7 in table 7.2. The word “بسيط”, “bsyṭ”, “simple” found in both reviews and could hold positive or negative sentiment depending on its context. The second test case is the one conducted on reviews number 16 and 17. Both of them contain the word “سعيد”, “Sʿyd”, which is a male name or and adjective means “happy”. The behavior of proposed models will be observed with both cases.

**Examining “bsyṭ”: Results**

Tables 7.3 and 7.4 are listing the results for the first example “bsyṭ”. For the positive case, we found out that all models were able to detect the polarity of the given review as they all classified it with positive polarity. For baseline model, GRU layer representation overtime has highlighted the word “جميل”, “jmyl”, “beautiful”, and “بسيط”, “bsyṭ”, “simple”, to be the most informative ones. Comparing the results to the attention-based model, we found out that the salient words, based on attention-layer output representation overtime, is “الكتاب”, “ạlktạb”, “book”. But we cannot guarantee that the final representation is ranked by the most informative words that led to the classification result without looking into the scores vector generated by attention layer scoring function. By unifying the scale with the attention layer score, all words received the same importance and we can only rely on the attention score to get the answer about the real most informative words that guided the calculation of the class probability. From SAL scoring vector, one of the least informative word was “بسيط”, “bsyṭ”, “simple”. SAL out vector in the attention-based DARLSA also got the same importance rank, but if we observe SAL score in table 7.3, we can noticed that the words that coms after and before “و”, “w”, “and”, which are “جميل”, “jmyl” and “بسيط”, “bsyṭ”, are sharing the same score range. This leads to an inference that the model can correlate that the two words have the same meaning because of the existence of “و”, “w”, “and”. This means that the model has the ability to draw the linguistic structure of the language.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Original polarity / detected** | | Positive | | |
| Baseline:  Positive 0.75 | Attention-based:  Positive 0.52 | Only-Attention:  Positive 0.52 |
| **Baseline model** | GRU layer output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.25.31 PM.png Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.25.54 PM.png | | |
| **Attention-based**  **DRLSA** | SAL output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.25.11 PM.png Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.26.19 PM.png | | |
| SAL score | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.25.19 PM.png Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.26.13 PM.png | | |
| **Only-attention** | | Machintosh:Users:Nada:Desktop:Screen Shot 2018-11-30 at 1.20.21 PM.png | | |
| Table 7. 3: Results For The Positive Case | | | | |

Only-attention model has highlighted the word “بسيط”, “bsyṭ”, “simple”, to be the salient word that led to the resulting polarity for both cases. This final representation is the resulted averaged three SAL representations. But for the positive case, the positive sentiment word “جميل”, “jmyl” got a higher score than the case with the negative sentiment word “ضعيف”, “ḍʿyf”.

For the negative case, both attention-based and attention-only models have succeeded to predict the review as negative one. Attention-based DARLSA model chosen sentiment word “ضعيف”, “ḍʿyf” with high score in the SAL scoring function, whereas the baseline failed to give the correct polarity and output a positive probability with 51%, but consider “بسيط”, “bsyṭ”, “simple”, as the salient word in its review representation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Original polarity / detected** | | Negative | | |
| Baseline:  Positive 0.51 | Attention-based: Negative 0.39 | Only-Attention:  Negative 0.41 |
| **Baseline model** | GRU layer output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-04-17 at 3.51.16 PM.png | | |
| **Attention-based**  **DRLSA** | SAL output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.49.10 PM.png | | |
| SAL score | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.49.48 PM.png | | |
| **Only-attention** | | Machintosh:Users:Nada:Desktop:Screen Shot 2018-11-30 at 1.19.09 PM.png | | |
| Table 7. 4: Results For The Negative Case | | | | |

**Examining “sʿyd”: Results**

Tables 7.5 and 7.6 are listing the results for the first example “سعيد”, “sʿyd”. The baseline model succeeded to classify the positive one, may be because of the strong sentiment contained in the word “سعيد”, “sʿyd”, “happy”. By tracking the representation overtime generated by GRU layer shown in table 7.5, the salient word detected by the model is “سعيد”, “sʿyd”, “happy”. However, for example 17 the model failed to classify it and give it high positive probability because of the existence of “سعيد”, “sʿyd” as it is considered again the most informative word.

Attention-based DARLSA succeeded to classify both examples correctly. For example 16, “سعيد”, “sʿyd”, “happy” was the word with highest score in SAL score vector. For example 17, the prediction decreased to the probability of 0.39 to be positive, and the model gives “سعيد”, “sʿyd” the highest score in both scoring vector and output review representation, which proved to be beneficial techniques to lower the value of unwanted words in analysis conducted in subsequent experiments in section 7.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Original polarity / detected** | | Positive | | |
| Baseline:  Positive 0.97 | Attention-based: Positive 0.66 | Only-Attention:  Positive 0.55 |
| **Baseline model** | GRU layer output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-04 at 10.04.15 PM.png | | |
| **Attention-based**  **RLSA** | SAL output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-04 at 10.03.56 PM.png | | |
| SAL score | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-04 at 10.04.07 PM.png | | |
| **Only-attention** | | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-04 at 10.08.09 PM.png | | |

Table 7. 5: Results For “sʿyd” Verb Case

Attention–only models, succeeded to classify the positive example with positive prediction and also the probability had decreased for the second example to 0.40, and “سعيد”, “sʿyd” was selected as salient feature correctly and ignored when needed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Original polarity / detected** | | Neutral – labeled as Negative | | |
| Baseline:  Positive 0.85 | Attention-based: Negative 0.39 | Only-Attention:  Negative 0.40 |
| **Baseline model** | GRU layer output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-04 at 10.10.37 PM.png | | |
| **Attention-based**  **RLSA** | SAL output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-04 at 10.10.19 PM.png | | |
| SAL score | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-04 at 10.10.27 PM.png | | |
| **Only-attention** | | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-28 at 8.52.11 AM.png | | |

Table 7. 6: Results For “Sʿyd” Proper Noun Case

# 7.1.2.2 Handling Negation and Intensification: objective

Testing the ability of the system to detect negation and intensification style is crucial for understanding the semantic of the review. Several case studies were conducted to study how the model will classify reviews with negations and how the model will behave with negation and intensification state. Handling negation is very important concept when modeling sentiment analysis system as it converts the polarity and the meaning of a sentence. In Arabic language, negation plays an important role by changing the whole meaning in different ways and styles, depending on the intended purposes. Explicit negation in Arabic could be used by including one of the negation words to reverse the meaning either positively or negatively. There are different words that could be used to negate the sentences. Beside using these words in the negation, they could be used also to express questions or as intensification style where the negation words are used as a metaphor and a style to stress more on the given polarity of a phrase. Review 8 from table 7.2 is an example of negation words used to stress on the positive polarity of a given review. The review contains “لم”, “lam”, which is considered one of negation words in Arabic, whereas the review contains a strong positive opinion.

Implicit negation, known as irony in English language, could be used in Arabic language as well, where the meaning of the sentence is negated without using explicit terms. One of the known methods to achieve implicit negation is by using wishing style to negate the phrase after the wishing term. Review 9 found in table 7.2 is an example that used to test the ability of the classifier to detect the meaning of the review and give it a proper polarity. Wishing style is to express that the reading of the novel was interesting that let the reader wish that the novel is longer.

In negation experiments there are two important factors we have to test them. The first factor is assuring that the negation is understood and used probably by the model by checking the polarity of a given sentence and comparing it with non-negated one. The other factor that should be tracked is the scope of the negation term, is it affect only the word after it or the effect of the negation is globally on the whole review. This is studied by looking after the attention weights for each word in the review.

# 7.1.2.2.1 Using Explicit Negation

In this test the simple negation form will be experimented by feeding the model with simple positive review, example 1 from table 7.2. Then, different modifications will be done on this review and then models’ behaviors will be monitored. The simple review form will be negated by using two of Arabic negation words, “غير”, “gẖyr” and “ليس”, “lys”. From Arabic language grammatically point of view, “غير”, “gẖyr” is considered as a noun and “ليس”, “lys” is a verb.

**Results**

As shown in table 7.7, baseline model could detect the positive polarity held in review 1 with correct informative word “جميل” , “jmyl”, “beautiful” but failed to classify negated reviews for both examples 2 and 3. Regardless the polarity test, we can see that the polarity probability had been decreased with “غير”, “gẖyr” but not with “ليس”, “lys”, which let one assumes that the model can detect the first keyword “غير”, “gẖyr” as a negation word and decrease the positivity of the review in this case.

|  |  |  |
| --- | --- | --- |
| **ID** | **GRU Layer Hidden Representation** | **Original Polarity / Classified** |
| 1 | Machintosh:Users:Nada:Desktop:Screen Shot 2018-11-24 at 4.20.11 PM.png | Positive /Positive with .83 |
| 2 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.24.22 AM.png | Negative/ Positive 0.74 |
| 3 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.25.14 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.25.25 AM.png | Negative/ Positive 0.85 |
| Table 7. 7: Baseline Results For Negation Test | | |

To prove this hypothesis, the review representation for each review, listed in table 7.7, is output from the model, and the difference between the opposing reviews is calculated. This may help us to extract the dimensions in the final representation that are responsible for the negation, if exist. Figure 7.2 shows the review representation for the experiments. As shown, there is no fixed representation between the differences representations in the case of using “ليس”, “lys” that we can rely on. But, in the case of using “غير”, “gẖyr”, we can find a neuron that fired clearly, and could be the negation dimension, surrounded by dotted rectangle in plot (d).

Figure 7. 2: Baseline GRU layer hidden-representation heatmap. (a) Review heatmap. (b) Negated review heatmap “gẖyr”. (c) Negated review heatmap “lys”. (d) Representation difference a-b (e) Representation difference a-c



(c)

(b)

(a)

(e)

(d)

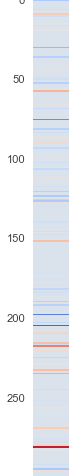
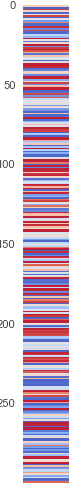


Table 7.8 shows the output results from attention-only model. For all examples all words got the same scores regardless using intensification or negations which means that the model cannot understand the semantic of a given review and even in the case of successful classification cases. The point that noticed is the ability of the model to detect salient words correctly in all examples.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Final Averaged Review Representation** | **Original polarity / Classified** | |
| 1 | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-28 at 5.27.18 PM.png | Positive/Positive 55% | |
| 2 | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-28 at 5.26.54 PM.png | Negative/Positive 57% | |
| 3 | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-28 at 5.19.49 PM.png | Negative /Positive 54% | |
| 4 | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-28 at 5.20.34 PM.png | Positive/Negative 48% | |
| 5 | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-28 at 5.20.15 PM.png | Positive/Negative 48% | |
| Table 7. 8: Attention-Only Model Results For Negation and Intensification Tests | | |

Table 7.9 shows the results for DARLSA model. As shown, the model successfully classified all reviews, which suggest that the model understood the semantic of the given text and negated it probably. To check the scope of the negation words, the attention weights are examined in all reviews. We can see that for the positive example the most salient word is “الكتاب”, “ạlktạb”. In the case of negation state, the score of the word that comes before negation term, in the final SAL output, does not affected by the negation process. Observing SAL output, the scope starting just after the negation term where the score of the negated adjective had been changed.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **SAL Output** | **Attention Score** | **Original Polarity / Classified** |
| 1 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.50.08 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 8.52.29 AM.png | Machintosh:Users:Nada:Desktop:Screen Shot 2018-11-24 at 4.20.01 PM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.04.20 AM.png | Positive /Positive with .51 |
| 2 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.17.21 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.17.37 AM.png | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.17.28 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.17.42 AM.png | Negative/ Negative 0.47 |
| 3 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.26.33 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.26.54 AM.png | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.27.08 AM.png | Negative/ Negative 0.45 |
| Table 7. 9: Deep Attention-based DARLSA model results for negation test | | | |

To study the effect of using negation on the final review representation when using DARLSA model, the review representation for each review and the difference between opposite reviews are extracted. Figure 7.3 shows the review representation for these experiments. The difference representation in both negation cases, shown in the figure plots (d) and (e), share a pattern and some neurons are fired clearly, surrounded by dotted rectangles in the figure.

**7.1.2.2.2 Using intensification keyword**

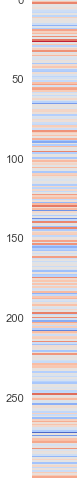
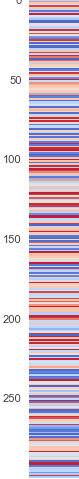
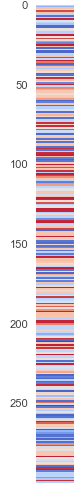
Figure 7. 3: Attention layer hidden-representation heatmap. (a) Review heatmap. (b) Negated review heatmap “gẖyr”. (c) Negated review heatmap “lys”. (d) Representation difference a-b (e) Representation difference a-c



(c)

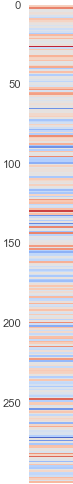
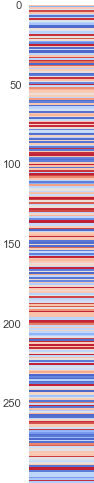
(b)

(a)



(e)

(d)



To test the ability of models to detect intensification style, intensity words, such as “جدا”, “jdạ”, “very and too”, is added just before the sentiment word “جميل”, “jmyl”, “beautiful” in example 4 and after it in example 5.

**Results**

Table 7.10 shows the results achieved by the baseline model. If we add intensity words, such as “جدا”, “jdạ”, “too” just before the sentiment word “جميل”, “jmyl”, “beautiful”, the baseline model responded well to the addition before the adjective by the increasing in the classification probability, but failed to recognize it after the adjective as the positivity probability decreased. To prove this findings, another example was used, examples 11,12 and 13 from table 7.2. By looking on the results, found in appendix B, even that the model failed to give example 11 a correct polarity, but the model handled the addition of intensification word before adjective by getting a lowest classification probability. This proves that the review semantic is well understood.

|  |  |  |
| --- | --- | --- |
| **id** | **GRU Layer hidden representation** | **Original polarity / Classified** |
| 4 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.07.15 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.07.31 AM.png | Positive /Positive with .90 |
| 5 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.31.55 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.32.08 AM.png | Positive /Positive with .80 |
| Table 7. 10 : Baseline Results For Intensification Test | | |

|  |  |  |  |
| --- | --- | --- | --- |
| id | SAL output | Attention score | **Original polarity / Classified** |
| 4 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.08.46 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.09.16 AM.png | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.08.54 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.09.23 AM.png | Positive /Positive with .53 |
| 5 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.29.14 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.30.02 AM.png | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.29.25 AM.png  Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 9.29.36 AM.png | Positive /Positive with .56 |
| Table 7. 11: Attention-Based DARLSA Model Results For Intensification Test | | | |

Table 7.11 shows the results gained by the attention-based DARLSA model. It had been noticed that the model response better, in terms of classification polarity, if the keyword placed after the adjective as the polarity have been increased from 0.51 into 0.56 compared to .0.53 in the case of placing it before. In addition, the intensification keyword had been selected as salient word when placed after the intended adjective. Observing the SAL output vector proves these findings as the score of the intensification word much higher, 0.0000538 with the case after, compared to .0000191 in the case of before. As a result, the sigmoid function in classification phase will output higher probability if the input is larger. These findings have been proved when the negative review tested, in examples 11, 12 and 13, found in appendix B. The original negative review had been classified as negative with 0.49 probabilities. Then the probability to be positive decreased to be 0.46 in the case of using intensification keyword before negative adjective and decreased more to 0.44 if it is used after the adjective.

For attention-only model, the test had been failed for classification phase but the intensification keyword had been selected correctly in all examples, as concluded from table 7.8.

**7.1.2.2.3 Using Negation as Intensification Style**

In this test, the ability of the system to understand the semantic of a given review is examined. Examples 8 is used form table 7.2 in this test, the review is using negation words as a style to boost the positivity implicitly.

**Results**

The baseline model failed to get the sentiment of the given review and classify it as negative because of the existence of the word “lam”, “not”, which had been given the lowest score. Table 7.12 shows the output results for the model.

|  |  |  |
| --- | --- | --- |
| **ID** | **GRU Layer Hidden Representation** | **Original Polarity / Classified** |
| 8 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.09.32 AM.png | Positive / Negative .40 |
| 10 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.11.52 AM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.19.39 AM.png | Negative / Positive 0.96 |
| 9 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.13.34 AM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.22.21 AM.png | Positive/Positive 0.86 |
| Table 7. 12: Baseline Model Results For Intensification Test And Irony Style | | |

Tables 7.13 Lists the results for attention-based DARLSA model. For all cases, the model gave correct polarities. For example 8, the negation word “lam” is highlighted with the highest score in the final review representation generated by the attention layer. In addition, the scores for all words increased by the existence of the negation. For attention only model, the model failed to classify the polarity correctly but succeed in the scoring function.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **SAL Output** | **Attention Score** | **Original Polarity / Classified** |
| 8 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.10.04 AM.png | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.10.11 AM.png | Positive/  Positive 0.50 |
| 10 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.11.35 AM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.19.30 AM.png | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.11.44 AM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.19.25 AM.png | Negative/ Negative 0.47 |
| 9 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.13.15 AM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.22.38 AM.png | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.13.25 AM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-01-03 at 11.22.31 AM.png | Positive /  Positive with .51 |
| Table 7. 13: Attention-Based DARLSA model results for implicit negation and irony style | | | |

# 7.1.2.2.4 Using Implicit Negation

Here, the test is using wishing style to switch the meaning of the sentence. Examples 9 and 10 are used for this test from table 7.2.

**Results**

In the baseline model, we can see, in table 7.12, that the model failed in the polarity test but can detect the irony style by decreasing the probability of the original review.

For the attention-only DARLSA model, we can see, in table 7.13, that the model can detect the negation as it classify the review without using wishing style with negative label, whereas the polarity had been switched into positive one with the existence of the wishing keyword which had been selected as the salient word. Depending on the context in review 10, the sentiment word “طويلة”، “ṭwylẗ”, “long”, is considered as negative word and has been doubled in the score when using wishing style in example 9 from 0.0000149 to 0.0000328, which explain the result of the classification phase.

# 7.1.2.3 Error Analysis

Because the proposed models are deep models, it is hard to identify the specific cause of an error. Therefore, the methodology of error analysis followed is to get the output of the attention model for the misclassified examples and tries to find the expected reason of error incident. Here, two misclassified samples by DARLSA model were extracted out of the test cases and analyzed.

**Example 1**

Table 7.14 shows the classification results outputs from baseline and DARLSA models. The original review contains contrasting opinions but the dominated one is the negative. The baseline model succeeded in the classification process whereas the attention-based DARLSA failed to detect the negative polarity. By observing the outcomes from the models, it can be noticed that the baseline distinguished the negative phrase with higher scores, which led to correct classification label. In the other hand, the failing of Attention-based DARLSA model is because of the failing of detecting the negation phrase as a negative style and got scores equal to the positive phrase.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **English Translation** | | Some of the selected characters were not important to me, but I enjoyed knowing the details of their success The paragraph of the writer Turki al-Dakheel was interesting in my opinion. | | |
| **Arabic-Latin Transliteration** | | bʿḍ ạlsẖkẖṣyạt ạlmkẖtạrẗ lm tkn mhmẗ bạlnsbẗ ly lkny ạstmtʿt bmʿrfẗ tfạṣyl njạḥhm fqrẗ ạlkạtb trky ạldkẖyl kạnt mmtʿẗ fy nẓry. | | |
| **Original polarity / detected** | | Negative | | |
| Baseline:  Negative 0.32 | Attention-based:  Positive 0.98 | Only-Attention:  Positive 0.52 |
| **Baseline model** | GRU layer output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-05-21 at 10.59.16 PM.png | | |
| **Attention-based**  **DRLSA** | SAL output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-05-21 at 10.59.01 PM.png | | |
| SAL score | Machintosh:Users:Nada:Desktop:Screen Shot 2019-05-21 at 10.59.26 PM.png | | |
| GRU layer output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-05-21 at 10.59.36 PM.png | | |

Table 7. 14: Example 1 Classification Results

**Example 2**

Table 7.15 shows the classification results outputs from baseline and DARLSA models. The original polarity for the review is positive with out using a clear positive phrase. It conveys a style of writing in Arabic where the writer explains the beauty of the novel in expressive criticism style. The attention-based DARLSA failed to detect the positivity contained in the review as no specific phrase where discriminated with higher value. The baseline model detected the correct polarity for this review and has a variation in the scores assigned for the phrases in the review.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **English Translation** | | I traveled with him in every detail and sailed with him in a world full of theories and philosophers that nature arose by chance and this great creature. | | |
| **Arabic-Latin Transliteration** | | sạfrt mʿh fy kl ạltfạṣyl ạ̉bḥrt mʿh fy ʿạlm mlỷ wnẓryạt ạlflạsfẗ fy ạ̉n ạlṭbyʿẗ nsẖạ̉t bạlṣdfẗ whdẖạ ạlmkẖlwq ạlʿẓym . | | |
| **Original polarity / detected** | | Positive | | |
| Baseline:  Positive 0.99 | Attention-based:  Negative 0.27 | Only-Attention:  Positive 0.52 |
| **Baseline model** | GRU layer output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-05-21 at 11.54.42 PM.png | | |
| **Attention-based**  **DRLSA** | SAL output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-05-21 at 11.55.03 PM.png | | |
| SAL score | Machintosh:Users:Nada:Desktop:Screen Shot 2019-05-21 at 11.55.11 PM.png | | |
| GRU layer output | Machintosh:Users:Nada:Desktop:Screen Shot 2019-05-21 at 11.54.53 PM.png | | |

Table 7. 15: Example 2 Classification Results

**Example 3**

Table 7.16 shows the classification results outputs from baseline and DARLSA models. The original polarity for this review is positive but written in a style that describes the pain; empathy and emotions touched the reader. Both models, baseline and DARLSA, had failed to detect the correct polarity for this review’s style. The output for both models weighted the pain related phrases with high score that led to negative classification.

|  |  |  |  |
| --- | --- | --- | --- |
| **English Translation** | | Is there anything more than Palestinian pain? Is there room for dreams far from home? Is there a rest felt by the Palestinians away from Palestine?  My heart squeezed with pain, a novel in which Kanafani's fingers were painful and suffering | |
| **Arabic-Latin Transliteration** | | hl hnạk mạ hw ạ̉qṣy̱ mn ạlwjʿ ạlflsṭyny?hl hnạk mtsʿ llḥlm bʿydạ ʿn ạlwṭn? hl hnạk rạḥẗ ysẖʿrhạ ạlflsṭyny bʿydạ ʿn flsṭyn ʿṣrt qlby bạlạ̉lm rwạyẗ kẖṭt fyhạ ạ̉nạml knfạny ạlwjʿ w ạlmʿạnạẗ ạlflsṭynyẗ | |
| **Original polarity / detected** | | Positive | |
| Baseline:  Negative 0.39 | Attention-based:  Negative 0.074 |
| **Baseline model** | **GRU layer output** | Machintosh:Users:Nada:Desktop:Screen Shot 2019-07-09 at 3.27.18 PM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-07-09 at 3.27.26 PM.png | |
| **Attention-based**  **DRLSA** | **SAL output** | Machintosh:Users:Nada:Desktop:Screen Shot 2019-07-09 at 3.25.11 PM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-07-09 at 3.25.23 PM.png | |
| **SAL score** | Machintosh:Users:Nada:Desktop:Screen Shot 2019-07-09 at 3.26.45 PM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-07-09 at 3.27.02 PM.png | |
| **GRU layer output** | Machintosh:Users:Nada:Desktop:Screen Shot 2019-07-09 at 3.26.19 PM.pngMachintosh:Users:Nada:Desktop:Screen Shot 2019-07-09 at 3.26.28 PM.png | |

Table 7. 16: Example 3 Classification Results

# 7.2 Deep Attention-Based Aspect-level sentiment analysis model (DAALSA): The Evaluation and Qualification

This section will discuss the results of DAALSAmodel experiments. Section 7.2.1 demonstrates the evaluation process results for model outcomes and section 7.2.2 is discussing the qualification analysis for the experimented models.

# 7.2.1 Evaluation Process Results and Discussion

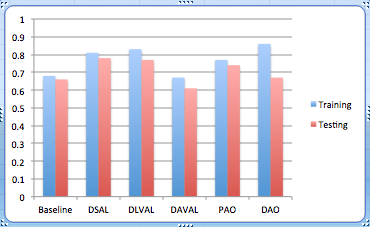
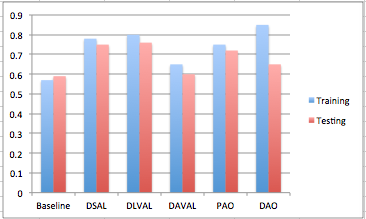
Table 7.17 is showing the classification report for all proposed models. Experimentation results show that our proposed approaches, highlighted in the table 7.17, outperform the GRU-based baseline experimentation. The accuracy of this attention model is significantly higher than the standard GRU-based model, our baseline in table 7.17, by 17% that proves the benefit of including attention mechanism to ABSA deep model. However, compared to other works done on the same dataset, it has been noticed that our models did not outperform their results with 76% by using handcrafted features and SVM model reported by (Pontiki et. al, 2016), 82.7% for CNN model proposed by (Ruder el. al, 2016b) and 81.7% for (Kumar et. al, 2016) model, with the aid of external sentiment lexicon. These findings suggested exploring and investigating the effect of hyper-parameters for the attention-based model on the deep model performance in a detailed way. However, what discriminated our models is that the models had been taken into another level of analysis to understand the behavior of the models and explain their outcomes. In addition, it does not relay on any handcrafted features and can adapt to test time out of domain features.

The best performing model during training phase was deep hierarchical attention-only model (DAO) with 85% F-measure but it got bad accuracy in the testing phase with 65% which means that the model tends to over fit. Based on metrics, using aspect vector to guide the classification was not helpful, where it gained F-measure of 65% in the training phase and 60% in the testing set, just the second worst model after the baseline. On the other hand, using learned context vector in the attention layer, DLVAL model, boost the performance to 80% training accuracy and 76% in testing dataset, compared to the result of 78% training phase and 75% in the testing dataset for soft attention layer and 59% for GRU-based baseline model. Figure 7.4 is presenting the performances of models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1-measure** |
| Baseline | .66 | .77 | .66 | .59 | |
| DSAL | .78 | .80 | .78 | .75 | |
| DLVAL | .77 | .73 | .77 | **.76** | |
| DAVAL | .61 | .76 | .61 | .60 | |
| PAO | .74 | .71 | .74 | .72 | |
| DAO | .67 | . 64 | .67 | .65 | |

Table 7. 17: Evaluation Process Results for Proposed DAALSA Models

(b)



(a

Figure 7. 4: Performance compared between all models for both test set and training set. (a) Accuracy. (b) F-measure.

# 7.2.2 Qualification Results and Discussion

The main objective of the qualification analysis is to examine the ability of the model to classify each review with respect to an aspect correctly. As a second level of analysis, each attention score is viewed to check the salient words selected by a model. Several reviews were selected out of the test dataset, with different details. As seen from table 7.18, the selected experimental cases include uni-aspet and multi-aspects reviews with same polarity and contracted polarities. The analysis will check first the ability of the proposed model to classify the given review with the correct class with respect to an aspect, and then, the most informative words in each review for each examined aspect is returned. Next sections will discuss the methodology used in the analysis process and how the inference is drawn from the results.



Table 7. 18: List of DAALSA Test Cases

**Analysis Methodology**

To obtain an intuition on the performance of the examined models, the behavior of scoring function in all models are observed first. It has been noticed that the scoring process tends to act as a gating mechanism to pass certain strength into the final output representation. The overall review representation is a resulted dense vector that does not show any words importance. By using attention mechanism, this dense vector is broken down into its compositional components and the words’ saliency could be observed. However, in most cases the word saliency is an intermediate step but not the final result. The final review representation will be achieved after applying the scoring function on the resulting representation overtime. This information was not obvious until DAO model was tested and visualized. It opens the black box of the dense vectors and allows returning reviews vectors generated by intermediate layers.

Based on observations, hypothetical gating rules were built, found in table 7.19, and these rules were used to evaluate models scoring. We will show that, after applying these rules, on the resulting vector representation, the actual words importance that reflects the classification results are returned.

|  |  |  |
| --- | --- | --- |
| **Input value** | **Score value** | **Output value** |
| Low | High | High |
| Low | Low | High |
| High | High | Low |
| High | Low | Low |
| Table 7. 19: Inference Gating Rules | | |

**DAO model**

Because deep attention-only model (DAO) opens the access to the representation overtime before and after applying the scoring function, we will start the discussion by analyzing the successful classification cases from this model. As seen before, this model deepens into four different attention layers, or level of abstractions. Each layer returns three different representations for a review with respect to an aspect. The score for each word, generated by the attention layer scoring function, is the first vector representation returned. This vector is incorporating in calculating the overall sentence representation generated as a layer output.

In addition to score vector, representation overtime for the whole input sequence is returned from each layer. Because we are allowing return sequence option for three deepen layers, the review hidden representation is returned in the mean of word importance , not a dense output, where d is the number of dimension and k is the number of words in a review. This option opens the access to the final importance in layer output representation after applying the scoring function and available in the first three layers but not in the final layer that returns a dense vector representing the final review representation generated by the model. This shows the ability of attention models to open the black box of current deep neural network representations.

The classification results generated by DAO model for example 3, from table 7.18, are analyzed in this section. DAO model succeeded to classify the review for each aspect correctly. It returned a negative polarity for “food” aspect, and a positive polarity for the “staff” aspect. First, the output vector representations along with scoring process are discussed for the first layer, DLVAL layer. After applying the inferred truth table on layer’s representation outputs, we can notice that the aspect under consideration got the highest score along with its sentiment words for both cases in the final output vector, as shown in table 7.20.

|  |  |  |
| --- | --- | --- |
| Aspect 1 | Output vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.36.52 PM.png |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.37.03 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.37.20 PM.png |
| Aspect 2 | Output vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.38.05 PM.png |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.38.13 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.38.20 PM.png |
| Table 7. 20: DLVAL Layer result, Example 3 | | |

For example, the first word “food” which is aspect 1, in the representation over time vector got a low score, when passed into the scoring vector with high score, the final output for this word in the output vector is high as well, which match the hypothesized gating rules. As a conclusion, we can set a rule, to get accurate classification for DLVAL layer, the representation overtime prefer to have the aspect under consideration related parts with low score.

By focusing on the scoring process for the second layer, the DSAL layer, we can infer the gating mechanism for the scoring function for this layer. It has been noticed that this layer tends to maintain the intensity orientating held in the input. As seen in the table 7.21, the scoring function trying to keep the value of the aspect along with its sentiment low. The layer tends to behave as a normalization layer by having a high uniform score generated by the scoring function which can be considered as a gating mechanism that flip the intensity into the other direction. For example, if the pass intensity of the word from the older representation is red it is going to be green. If it is light green it will be yellow, and so on.

Recall the goal of including DSAL layer in this proposed model, listed in section 5.3.2, this finding supports the modeling arguments and proves its usefulness.

|  |  |  |
| --- | --- | --- |
| Aspect 1 | Output vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.47.44 PM.png |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.47.30 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.47.22 PM.png |
| Aspect 2 | Output vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.48.02 PM.png |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.47.53 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.47.22 PM.png |
| Table 7. 21: DSAL1 Layer, Example 3 | | |

Stepping into layer three, AVAL layer, where the review is fed along with an aspect vector into the layer. The effect of the existence aspect vector in the final calculation could be seen obviously. The scoring function in this layer does not follow the rules inferred in table 7.19 and tends to pass the intensity of representation overtime into the final layer output. As seen in table 7.22, the final representation is produced by lowering the value of words related to aspect under consideration as a result of the dot product done in the scoring process representation.

|  |  |  |
| --- | --- | --- |
| Aspect 1 | Output vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 7.00.35 PM.png |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 7.00.44 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 7.00.52 PM.png |
| Aspect 2 | Output vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 7.01.00 PM.png |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 7.01.07 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 7.01.16 PM.png |
| Table 7. 22: AVAL Layer, Example 3 | | |

Finally, in the last layer, LSAL, this representation is going to be normalized again to get the final set of salient words. As seen in the table 7.23, the layer does not return the final importance score vector as a dense vector is returned to represent the review. But applying the truth table inferred will fix the puzzle. According to the representation overtime, the salient parts for each aspect is set probably in the final representation which explains the successful classifier prediction for both aspects.

|  |  |  |
| --- | --- | --- |
| Aspect 1 | Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 7.07.41 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.47.22 PM.png |
| Aspect 2 | Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 7.07.49 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 6.47.22 PM.png |
| Table 7. 23: LSAL Layer, Example 3 | | |

However, successful classifier depends also on other factors. Embedding vector values, weight initialization and all are hidden factors that could affect the accuracy of any model.

**DAVAL Model**

Studying the behavior of model when processing the same example in DAVAL model is the next analysis point. The model has succeeded to classify the review for both aspects. However, it could be noticed after observing the behavior of the layer in both models, DAO and DAVAL, that the layer has different gating schema than LVAL and SAL. This is expected because the layer used here has different architecture where the aspect vector is multiplied by learned weight matrix before passing the value into *tanh* activation function. Looking at table 7.24, it could be noticed that the attention score function tends to highlight the aspect under consideration. This is viable because the model is passing the aspect vector into the layer separately. After observing many output representations of different examples from DAVAL layer in DAO model, it has been noticed that the final output representation preserves the intensity orientation of the words in the representation overtime. As a result, for the example at hand, the aspect alone with its sentiment are going to be highlighted in the final review representation for each aspect, which explains the successful classification results.

**Further analysis**

This section is discussing more examples classified by different models by applying the knowledge inferred from the previous analysis. Both successful and misclassified reviews are studied to analyze the source of errors. The methodology followed is to apply the hypothetical truth table on the review representation overtime to check if the final review representation expresses the polarity of the intended aspect.

|  |  |  |  |
| --- | --- | --- | --- |
| Aspect1 | Output representation | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-10 at 7.25.07 PM.png | Negative .49 |
| Attention score | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-10 at 7.25.25 PM.png |
| Aspect2 | Output representation | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-10 at 7.25.15 PM.png | Positive .91 |
| Attention score | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-10 at 7.25.31 PM.png |

First, example 1 from table 7.18, a review contains one positive aspect, will be analyzed as all models classified it accurately. The first examined model will be DAO model. As shown in table 7.25, the model classified the review with a 0.77 probability to be positive. As we get a dense vector representation as an output from the layer, we have to apply the inferred rules to get the final importance. The result will highlight the main aspect with the highest score and left the sentiment clause with low score.

Table 7. 24: Table 7.24: DAVAL Model Results, Example 3

|  |  |
| --- | --- |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 8.41.37 AM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 8.42.58 AM.png |
| Table 7. 25: DAO Model Final Layer Results, Example 1 | |

Next, DAVAL model results will be presented. The model succeeded to classify the review with positive prediction. Looking at table 7.26, we can conclude that the model consider the sentiment clause as the most informative part of the review, after applying the knowledge that the scoring function tends to preserve the highest score in the representation overtime.

|  |  |
| --- | --- |
| Output representation | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 9.16.07 AM.png |
| Attention score | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 9.15.56 AM.png |
| Table 7. 26: DAVAL Model, Example 1 | |

Then, DSAL model output is examined. We can see in table 7.27, the scoring vector generated and the representation overtime for this review. However, word importance in the generated review representation is forbidden as we get a dense vector as the layer final output.

|  |  |
| --- | --- |
| Output representation | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-10 at 1.40.35 PM.png |
| Attention score | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-10 at 1.40.46 PM.png |

Table 7. 27: DSAL Model Results, Example 1

Here, the model succeeds to classify the review as positive. The review-hidden representation gives the highest score to the aspect itself. After applying the truth table, we can infer that in the final review representation the sentiment clause gets the highest score among surrounding words, which could explain the high positive probability, 0.94, for this aspect. Therefore, for DSAL to succeed in the classification process the sentiment clause should get a low score before proceeding into the scoring process.

Table 7.28 is the generated representation out from DLVAL model. Like DSAL based model, the model distinguishes the aspect sentimental clause in the final review representation with high scores, which explain the correct classification result.

|  |  |
| --- | --- |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 5.26.45 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 5.26.34 PM.png |
| Table 7. 28: DLVAL Model Results, Example 1 | |

For both models, DSAL and DLVAL, The review representation accepted by the attention layer almost has the same intensity in both cases because both layers accept the review-hidden representation concatenated by the aspect-hidden representation generated by the preceding GRU layer. On the other hand, the scoring function is different because of the existence of the learned context vector in the case of DLVAL attention layer. As a conclusion, for both models it has been noticed that the aspect with related clause gets the lowest score in the review representation.

To support DLVAL model analysis, example 2 from table 7.18 is going to be discussed in the next section. Table 7.29 is illustrating the representations generated by the model for the review with respect to aspect “room”. The review has a negative label and the probability generated by the model is 0.50 to be positive and 0.49 to be negative.

|  |  |
| --- | --- |
| Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 5.36.15 PM.png |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 5.36.06 PM.png |

Table 7. 29: DLVAL Model, Example 2

By applying our truth table, we can conclude that the salient part in the overall review representation, beside the sentiment word of the aspect, the other clause, “تسع لشخصين”, “tsʿ lsẖkẖṣyn”, “settle for two”, which can be considered as positive note, will be included in the saliency list. This could explain the failing of the system to give strong negative classification result.

Taking a third example for two-aspects review could give us more intuition about DLVAL model behavior. A misclassified review for one aspect and succeed for the other one is the example given in table 7.30.

|  |  |  |  |
| --- | --- | --- | --- |
| Aspect 1 | Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 5.50.13 PM.png | Positive/  Negative .57 |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 5.50.06 PM.png |
| Aspect 2 | Representation overtime | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 5.50.26 PM.png | Negative/ Negative .54 |
| Scored vector | Machintosh:Users:Nada:Desktop:Screen Shot 2018-12-13 at 5.50.19 PM.png |

Table 7. 30: DLVAL Model, Example 4

By applying the truth table, aspect 1 will have high score as well in the final review representation, which is unwanted in the ideal scenario. Another observation is that the sentiment word for aspect 1 “جيد”, “jyd”, “good”, remains with high intensity in both scoring functions, which could mislead the classifier. As a result, the model misclassified aspect 1 with negative label, and gave aspect 2 accurately negative label as well.

Observing baseline results could be achieved by applying the concept of transfer learning on the problem in hand. Defining a model based on the attention model, then truncate the model just after GRU-layer and load the baseline weights will open the possibility to get representation overtime of the current input. Table 7.31 shows the results generated by the model when predicting some of the examples listed in table 7.18. As shown, there is no definite rule that we can infer to explain the behavior of the model with the absence of any scoring function. We can see from the first row of the table that the model succeeded to classify the review and select the aspect itself to be the most informative words in the review. On the other hand, with the second row also the model succeed in the classification with respect to aspect 1 but the aspect is selected with least score. More experimented examples are found in appendix C.

# 7.3 Discussions and Summary

In this chapter, assessment took place with respect to the classification accuracy and salient words detection ability for both proposed models, DARLSA and DAALSA. It has been shown that having a model with different level of attention will generate more focused final review representation than having shallower network. This could be explained because the contextual words lost their sentiment and compacted into the final review representation. But with more refining, the scores normalized and semantically important words of the review became stronger.

The implemented model DARLSA has achieved an acceptable performance with an average f-measure of 87% for the baseline mode, 85% for attention-based and 82% for only-attention model. In the qualification analysis, DARLSA attention-based model proved its efficiency and its ability to handle linguistic variations conducted on the test cases. Baseline model failed the word context tests, which is a testing done to check model abilities in detecting the meaning of words placed into different contexts. In the other hand, DARLSA attention-based model passed the test and showed its ability to detect linguistic characteristic such as detecting synonyms. In addition, DARLSA passed all test done to handle negations but baseline model failed to detect “ليس”, “lys” and recognized “غير”, “gẖyr”. This gives a clue about the behavior of the model to handle linguistic grammars as “ليس”, “lys” is considering as verb and “غير”, “gẖyr” is a noun in Arabic. DARLSA attention-based model also shows its ability to handle intensification efficiently and some dimensions, believed to be intensification dimensions, are fired in the overall review representation clearly.

Furthermore, the performance of DAALSA has also been evaluated. DLVAL model gave F-measure of 76% on the test dataset, and be able to classify reviews with conflict aspects polarities.

The best achieving model was DAO with F-measure of 85% on training dataset, however the model tends to over fit and gave an accuracy of 65% during testing phase. But after the qualification tests, it could be noticed that having multiple levels of attention in DAO model, beside its efficiency with respect to training time and parameters number needed, the model opens possibilities to explain the underlying results generated by inner layers. Add to this, the model succeeded to classify multi aspect reviews with opposite polarities, where most of the examined models failed to detect the accurate polarities correctly. Although these cases faced rarely, it is considered as a challenging situation for aspect-based sentiment analysis system. Therefore, it will be beneficial if the experiment is repeated with more training data and for loner epochs to take the model into another level of analysis.

|  |  |  |
| --- | --- | --- |
|  | Representation overtime | Prediction  Label/predicted |
| Aspect 1 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 8.15.19 AM.png | Positive/  Positive .88 |
| Aspect 1 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 8.17.58 AM.png | Positive/  Positive .96 |
| Aspect 2 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 8.18.05 AM.png | Negative/  Positive .95 |
| Aspect 1 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 8.20.49 AM.png | Neutral /  Positive .85 |
| Aspect 2 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 8.20.56 AM.png | Neutral /  Positive .87 |
| Aspect 1 | Machintosh:Users:Nada:Desktop:Screen Shot 2019-01-01 at 8.25.46 AM.png | Negative /  Positive .96 |
| Table 7. 31: Baseline Model Results | | |

The analysis of the results proved that SAL attention layer and LVAL tends to have similar gating mechanism to pass specific intensity out of the scoring function, inferred in table 7.20. However, because of the incorporating of aspect vector in calculating the scoring function in AVAL layer, the gating technique tends to be different by preserving the intensity found in the representation overtime. On the other hand, attention-only models failed to detect the class of given reviews efficiently in all conducted tests, but have a distinguished ability to detect salient words for all examined reviews correctly which could be invested, if optimized, in a type of aspect extraction systems.

To draw a link between DAALSA and DARLSA models, the inferred gating mechanism is going to be applied on reviews to test its effectiveness. We can seen from table 7.5 that the semantic word “سعيد”, “sʿyd”, “happy” got the highest score in SAL vector but low score in SAL output vector. By applying the hypothesized gating mechanism, words “سعيد”, “sʿyd”, “happy” and “بقراءة”, “bqrạʾẗ”,“to read” will got high score which reflects the real semantic of the review. Recall from table 7.6, SAL output salient words were “سعيد ناصر”, “Sʿyd Nạṣr” which are not related to the polarity detected. It can be noticed that both of words got high scores in both representations, SAL score vector and SAL output representation. After passing the words into inferred gating test, we found out that both words will lose their high score and “كتاب”, “ktạb”, “book” and “قصة”, “qṣẗ , “story” will be the most important words, which is semantically correct.

Finally, observing GRU-layer output does not draw any conclusion about DAALSA baseline model behavior in selecting the salient words. For simplicity purposes, only subsets of the examined cases were presented in the discussion but more of test cases, for both DARLSA and DAALSA, are found in appendix B and C.

Chapter 8: Conclusion and Future Works

The thesis is about developing sentiment analysis framework for Arabic language by adopting the continuous space neural language model by optimizing state-of-art models and alleviating their limitations. In this work, the effect of applying deep attention-based models on Arabic sentiment analysis is investigated and the ability of the models to capture semantic representations without using any linguistic resources are studied. To achieve this goal, Deep Attention-based Review Level Sentiment Analysis (DARLSA), and Deep Attention-based Aspect-level Sentiment Analysis (DAALSA) models were proposed and examined. For DARLSA, different pre-trained embedding weights were used and the results were analyzed. For both models, the effect of using attention-only architecture is investigated. In addition, visualizing attention weights is conducted to open the models’ black box and to verify their intuition, which helps to interpret the learned representations. Recalling the main research questions that are addressed in this thesis:

* Does using Deep Attention-based Review Level Sentiment Analysis model (DARLSA) for classifying Arabic reviews polarity is beneficial in terms of classification accuracy and selecting salient features? .
* Does using Deep Attention-based Aspect Level Sentiment Analysis model (DAALSA) will produce the final review polarity in correlation to a specific aspect?
* Does visualizing attention layer weights helps in opening neural network models’ black-box?

The following sections will discuss the main findings of this thesis. Possible future work directions will be presented in the last section of this chapter.

# 8.1 Summary

In this thesis, deep models based on attention mechanism are proposed to classify sentiment in different levels. To meet the first research question, pre-processing tool is developed to normalize, tokenize and remove stop words from datasets. Then, two levels of analysis, review level and aspect-level sentiment classification are proposed to analyze reviews’ polarities. The challenge was to select the most informative words that reflect a sentiment in a review, and to correlate aspect with its sentiment words in aspect-level sentiment analysis models. For DARLSA, binary classifier is trained and then the effect of using attention mechanism is examined. Three models were compared, GRU-based baseline model, attention-based and only-attention models. Based on F-measure, the GRU-based baseline model outperformed other models, but in the second level of analysis, qualitative analysis, when the most salient word considered by the model was visualized, the attention-based model proves its efficiency in handling different language situation such as negation, intensification and getting the right word context. In the other hand, beside the poor F-measure for attention-only model compared with other models, qualitative analysis showed its efficiency in detecting salient words. In addition, transfer learning is studied by investigating the affect of using different pre-training to initialize embeddings’ layer weights. Three models were tested, the first two models use two different pre-trained word embeddings, and in the third model, random weights initialization is facilitated. The results proved that using pre-trained model accelerates learning compared to the randomly weight initialization.

To answer the second research question, DAALSA model is proposed. By using deep attention-based architecture, DAALSA is classifying the polarity of given review into three classes, positive, negative or neutral, with respect to an aspect. The aspect embedding is playing a role when calculating the final review representation to draw inter-dependencies between aspects and their contextual words. To facilitate the aspect information into proposed models, aspect-embedding vector is learned for each aspect. Different models were proposed to experiment different cases of considering aspect information. In DSAL and DLVAL models the aspect representation is concatenated with the review representation before proceeding to the attention layer to calculate the saliency words regarding this aspect. In DAVAL model, the aspect representation is passed separately in a form of vector to guide the scoring function. In addition, multi-layer attention models were proposed where the models depend only on attention layers without using any recurrent-based layers. The results showed that DLVAL model outperform other models in both F-measure and also picking aspect related words to be salient words in the final representation. In addition, attention-only DAO model, output good performance and attends the objective of this thesis by opening access to all layers’ internal representations, which gives explanations for the output and the behavior of neural network and meets the third research question.

Using multiple computational layers, by means of deep models, the salient words in a given review is given high scores in the final generated review representation vector, which is considered as a feature for the sentiment classification. As every component of the proposed models is differentiable, all models could be trained entirely in an end-to-end manner with gradient descent, where the loss function is cross-entropy error of sentiment classification.

# 8.2 Future Works

The work that has been done in this study gives promising results in the area of Arabic sentiment analysis systems, which is still in its early times. There are many different areas of improvement that could be done to move these systems into further steps. In this thesis, methods to predict aspect-level review polarity are developed, regarding a known one aspect. One possible and interesting direction would be modeling many aspects concurrently by using attention mechanism that distinguishes different aspects in a given review, and generate multiple vector-representations for each aspect.

Another possible direction is to rely on local, focused context vector when calculating the scoring function. The proposed DLVAL model uses the concept of self-attention proposed by (Wang et. al, 2016; Yang et. al, 2016; Lin et. al, 2017), where the global attention concept is used by calculating a context vector considering the whole review’s words. However, considering all the words in the review when calculating review representation with respect to an aspect is too expensive and may contain unrelated words to aspect under consideration. Sentiment words closed to an aspect under consideration should be more important. Inspired by (Luong et. al, 2015), it will be beneficial to study the effect of facilitating the local context onto the overall review representation by altering self-attention mechanism to include relative position representations to specific aspect.

The results of our experiments showed that using attention-only layers for both DARLSA and DAALSA has potential success if optimized. The models showed its ability to select the saliency words for tested reviews in the qualification tests. It will be worth to repeat the experiments with bigger data and more epochs to get better accuracy. It would be beneficial to apply generalization concept on the developed model by having a mean of cross-dataset training and testing. For example, after training the model on LABR book reviews dataset, it would be interesting to use Arabic movie reviews dataset.in the testing phase.

Multiple layers of attention proved its proficiency to extract the underlying semantics of input text and highlight features in complicated sentences. However, having fixed numbers of layers to be applied on different length of reviews is not practical and may lead to un-needed redundant tasks. Therefore, a mechanism is needed to stop deepen into attention layers if no more salient words are going to be extracted.

Another promising direction is to generate a summary presentation for all opinions exist in the dataset regarding specific aspect. For example, when classifying hotel reviews, each review can be analyzed to extract, for example room aspect and their sentiment. Then, all reviews for room aspect is aggregated and summarized to get a shared opinion. If common sentiments are shared for specific aspect, then that sentiment is reflecting the real situation for this aspect. Having such summarization interface can take sentiment analysis system into higher level of analyzing and inference ability.

# Bibliography

Abbasi, Ahmed, 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 (2008): 12.

Abdul-Mageed, Muhammad, and Mona T. Diab. "Subjectivity and sentiment annotation of modern standard Arabic newswire." In Proceedings of the 5th linguistic annotation workshop, pp. 110-118. Association for Computational Linguistics, 2011.

Abdul-Mageed, Muhammad, Mona Diab, and Sandra Kübler. "SAMAR: Subjectivity and sentiment analysis for Arabic social media." Computer Speech & Language 28.1 (2014): 20-37.

Abdul-Mageed, Muhammad. "Learning Subjective Language: Feature Engineered vs. Deep Models." OSACT 3: The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools. 2018.

Al-Rfou, Rami, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien et al. "Theano: A Python framework for fast computation of mathematical expressions." arXiv preprint arXiv:1605.02688 (2016).

Al-Smadi, Mohammad, Omar Qawasmeh, Bashar Talafha, and Muhannad Quwaider. "Human annotated Arabic dataset of book reviews for aspect based sentiment analysis." In 2015 3rd International Conference on Future Internet of Things and Cloud, pp. 726-730. IEEE, 2015.

AL-Smadi, Mohammad, Omar Qwasmeh, Bashar Talafha, Mahmoud Al-Ayyoub, Yaser Jararweh, and Elhadj Benkhelifa. "An enhanced framework for aspect-based sentiment analysis of Hotels' reviews: Arabic reviews case study." Internet Technology and Secured Transactions (ICITST), 2016 11th International Conference for. IEEE, 2016.

Al-Smadi, Mohammad, Bashar Talafha, Mahmoud Al-Ayyoub, and Yaser Jararweh. "Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews." International Journal of Machine Learning and Cybernetics 10, no. 8 (2019): 2163-2175.

Al-Twairesh, Nora, Hend Al-Khalifa, AbdulMalik Alsalman, and Yousef Al-Ohali. "Sentiment analysis of Arabic tweets: Feature engineering and a hybrid approach." arXiv preprint arXiv:1805.08533 (2018).

Alayba, Abdulaziz M., Vasile Palade, Matthew England, and Rahat Iqbal. "Improving Sentiment Analysis in Arabic Using Word Representation." arXiv preprint arXiv:1803.00124 (2018).

Altowayan, A. Aziz, and Lixin Tao. "Word embeddings for Arabic sentiment analysis." Big Data (Big Data), 2016 IEEE International Conference on. IEEE, 2016.

Aly, Mohamed, and Amir Atiya. "LABR: A large scale Arabic book reviews dataset." Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2013.

Anderson, Dave, and George McNeill. "Artificial neural networks technology." Kaman Sciences Corporation 258 (1992): 13502-462.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014)

Bansal, Barkha, and Sangeet Srivastava. "Sentiment classification of online consumer reviews using word vector representations." Procedia computer science 132 (2018): 1147-1153.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. "A neural probabilistic language model. journal of machine learning research, Vol. 3, No." (2003): 1137-1155.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." Pattern Analysis and Machine Intelligence, IEEE Transactions on 35.8 (2013a): 1798-1828.

Bengio, Yoshua, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. "Advances in optimizing recurrent networks." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013b.

Chen, Peng, Zhongqian Sun, Lidong Bing, and Wei Yang. "Recurrent attention network on memory for aspect sentiment analysis." In Proceedings of the 2017 conference on empirical methods in natural language processing, pp. 452-461. 2017.

Cho, Kyunghyun, Aaron Courville, and Yoshua Bengio. "Describing multimedia content using attention-based encoder-decoder networks." IEEE Transactions on Multimedia17.11 (2015): 1875-1886.

Chollet, Fran ̧cois. 2015. Keras. https://github. com/fchollet/keras.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

Dahou, Abdelghani, Shengwu Xiong, Junwei Zhou, Mohamed Houcine Haddoud, and Pengfei Duan. "Word embeddings and convolutional neural network for Arabic sentiment classification." In Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers, pp. 2418-2427. 2016.

Darwish, Kareem, and Walid Magdy. "Arabic information retrieval." Foundations and Trends® in Information Retrieval 7, no. 4 (2014): 239-342.

Deng, Li, and Dong Yu. "Deep learning: methods and applications." Foundations and Trends® in Signal Processing 7.3–4 (2014): 197-387.

El-Beltagy, Samhaa R., Mona El Kalamawy, and Abu Bakr Soliman. "NileTMRG at SemEval-2017 Task 4: Arabic Sentiment Analysis." arXiv preprint arXiv:1710.08458 (2017).

El-Khair, Ibrahim Abu. "Effects of Stop Words Elimination for Arabic Information Retrieval: A Comparative Study." arXiv preprint arXiv:1702.01925 (2017).

Elkan, Charles. Predictive analytics and data mining. University of California, 2013.

Elman, Jeffrey L. "Finding structure in time." Cognitive science14.2 (1990): 179-211.

ElSahar, Hady, and Samhaa R. El-Beltagy. "Building large Arabic multi-domain resources for sentiment analysis." In International Conference on Intelligent Text Processing and Computational Linguistics, pp. 23-34. Springer, Cham, 2015.

Farghaly, Ali, and Khaled Shaalan. "Arabic natural language processing: Challenges and solutions." ACM Transactions on Asian Language Information Processing (TALIP) 8.4 (2009): 14.

Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach." Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y "Maxout networks." arXiv preprint arXiv:1302.4389 (2013a).

Goodfellow, Ian J., Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. "Multi-digit number recognition from street view imagery using deep convolutional neural networks." arXiv preprint arXiv:1312.6082 (2013b).

Guan, Ziyu, Long Chen, Wei Zhao, Yi Zheng, Shulong Tan, and Deng Cai. "Weakly-Supervised Deep Learning for Customer Review Sentiment Classification." In IJCAI, pp. 3719-3725. 2016.

Habash, Nizar Y. "Introduction to Arabic natural language processing." Synthesis Lectures on Human Language Technologies 3.1 (2010): 1-187.

Hagan, Martin T., Howard B. Demuth, and Mark Beale. Neural network design. University of Colorado Bookstore, 2002.

Hamdi, Ali, Khaled Bashir Shaban, and Anazida Zainal. "A Review on Challenging Issues in Arabic Sentiment Analysis." JCS12.9 (2016): 471-481.

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18.7 (2006): 1527-1554.

Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

Irsoy, Ozan, and Claire Cardie. "Opinion mining with deep recurrent neural networks." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014.

Jebbara, Soufian, and Philipp Cimiano. "Aspect-Based Relational Sentiment Analysis Using a Stacked Neural Network Architecture." arXiv preprint arXiv:1709.06309 (2017).

Karpathy, Andrej, Justin Johnson, and Li Fei-Fei. "Visualizing and understanding recurrent networks." arXiv preprint arXiv:1506.02078 (2015).

Karsoliya, Saurabh. "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture." International Journal of Engineering Trends and Technology 3.6 (2012): 713-717.

Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980(2014).

Kumar, Ayush, Sarah Kohail, Amit Kumar, Asif Ekbal, and Chris Biemann. "Iit-tuda at semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis." In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp. 1129-1135. 2016.

Le, Quoc V., and Tomas Mikolov. "Distributed representations of sentences and documents." arXiv preprint arXiv:1405.4053 (2014).

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436.

Li, Jiwei, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. "Visualizing and understanding neural models in NLP." In Proceedings of NAACL (2016).

Lin, Zhouhan, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. "A structured self-attentive sentence embedding." arXiv preprint arXiv:1703.03130 (2017).

Lipton, Zachary C., John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning." arXiv preprint arXiv:1506.00019 (2015).

Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.

Mäntylä, Mika V., Daniel Graziotin, and Miikka Kuutila. "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers." Computer Science Review 27 (2018): 16-32.

Martins, Claudia Aparecida, Maria Carolina Monard, and Edson Takashi Matsubara. "Reducing the dimensionality of bag-of-words text representation used by learning algorithms." Proc of 3rd IASTED International Conference on Artificial Intelligence and Applications. 2003.

Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, in Proceedings of Workshop at ICLR, 2013.

Mnih, Andriy. "Learning Distributed Representations for Statistical Language Modelling and Collaborative Filtering." PhD diss., 2010.

Monroe, Will, Spence Green, and Christopher D. Manning. "Word segmentation of informal Arabic with domain adaptation." Proceedings of the 52d Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2014.

Omar, Nazlia, Mohammed Albared, Adel Qasem Al-Shabi, and Tareq Al-Moslmi. "Ensemble of classification algorithms for subjectivity and sentiment analysis of Arabic customers' reviews." International Journal of Advancements in Computing Technology 5, no. 14 (2013): 77.

Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." Foundations and Trends® in Information Retrieval2.1–2 (2008): 1-135.

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

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12 (2014).

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect based sentiment analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation. Dublin, Ireland, SemEval ’14, pages 27–35.

Pontiki, Maria, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub et al. "Semeval-2016 task 5: Aspect based sentiment analysis." In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), pp. 19-30. 2016.

Qiu, L., Cao, Y., Nie, Z., Yu, Y., and Rui, Y. "Learning Word Representation Considering Proximity and Ambiguity." Twenty-Eighth AAAI Conference on Artificial Intelligence. 2014.

Raffel, Colin, and Daniel PW Ellis. "Feed-forward networks with attention can solve some long-term memory problems." arXiv preprint arXiv:1512.08756 (2015).

Ruder, Sebastian, Parsa Ghaffari, and John G. Breslin. "A hierarchical model of reviews for aspect-based sentiment analysis." arXiv preprint arXiv:1609.02745 (2016a).

Ruder, Sebastian, Parsa Ghaffari, and John G. Breslin. "Insight-1 at semeval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis." arXiv preprint arXiv:1609.02748 (2016b).

Rumelhart, David E. "Parallel distributed processing: Explorations in the microstructure of cognition." Learning internal representations by error propagation 1 (1986): 318-362.

Shoukry, Amira, and Ahmed Rafea. "Sentence-level Arabic sentiment analysis." Collaboration Technologies and Systems (CTS), 2012 International Conference on. IEEE, 2012.

Socher, Richard, Christopher D. Manning, and Andrew Y. Ng. "Learning continuous phrase representations and syntactic parsing with recursive neural networks." Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop. 2010.

Socher, Richard, et al. "Semi-supervised recursive autoencoders for predicting sentiment distributions." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. "Recursive deep models for semantic compositionality over a sentiment treebank." In Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 1631-1642. 2013.

Socher, Richard. "Recursive deep learning for natural language processing and computer vision." PhD diss., Stanford University, 2014.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15, no. 1 (2014): 1929-1958.

Sukhbaatar, Sainbayar, Arthur Szlam, Jason Weston, and Rob Fergus. “End-to-end memory networks”. arXiv preprint arXiv:1503.08895. (2015)

Tang, Duyu, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. "Coooolll: A deep learning system for twitter sentiment classification." In Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pp. 208-212. 2014.

Tang, Duyu, Bing Qin, Xiaocheng Feng, and Ting Liu. "Effective LSTMs for target-dependent sentiment classification." arXiv preprint arXiv:1512.01100 (2015).

Tang, Duyu, Bing Qin, and Ting Liu. "Aspect level sentiment classification with deep memory network." arXiv preprint arXiv:1605.08900 (2016).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In Advances in neural information processing systems, pp. 5998-6008. 2017.

Wang, Haohan, and Bhiksha Raj. "On the origin of deep learning." arXiv preprint arXiv:1702.07800 (2017).

Wang, Yequan, Minlie Huang, and Li Zhao. "Attention-based lstm for aspect-level sentiment classification." Proceedings of the 2016 conference on empirical methods in natural language processing. 2016.

Weston, Jason, Sumit Chopra, and Antoine Bordes. "Memory networks." arXiv preprint arXiv:1410.3916 (2014).

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. "Hierarchical attention networks for document classification." Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.

Young, Tom, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. "Recent trends in deep learning based natural language processing." ieee Computational intelligenCe magazine 13.3 (2018): 55-75.

Zahran, Mohamed A., Ahmed Magooda, Ashraf Y. Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. "Word representations in vector space and their applications for Arabic." In International Conference on Intelligent Text Processing and Computational Linguistics, pp. 430-443. Springer, Cham, 2015.

Zeng, H. A. I. P. E. N. G. "Towards better understanding of deep learning with visualization." Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (2016).

Zupan, Jure. "Introduction to artificial neural network (ANN) methods: what they are and how to use them." Acta Chimica Slovenica 41 (1994): 327-327

# Appendix

# Appendix A: Experiments Accuracy Plots

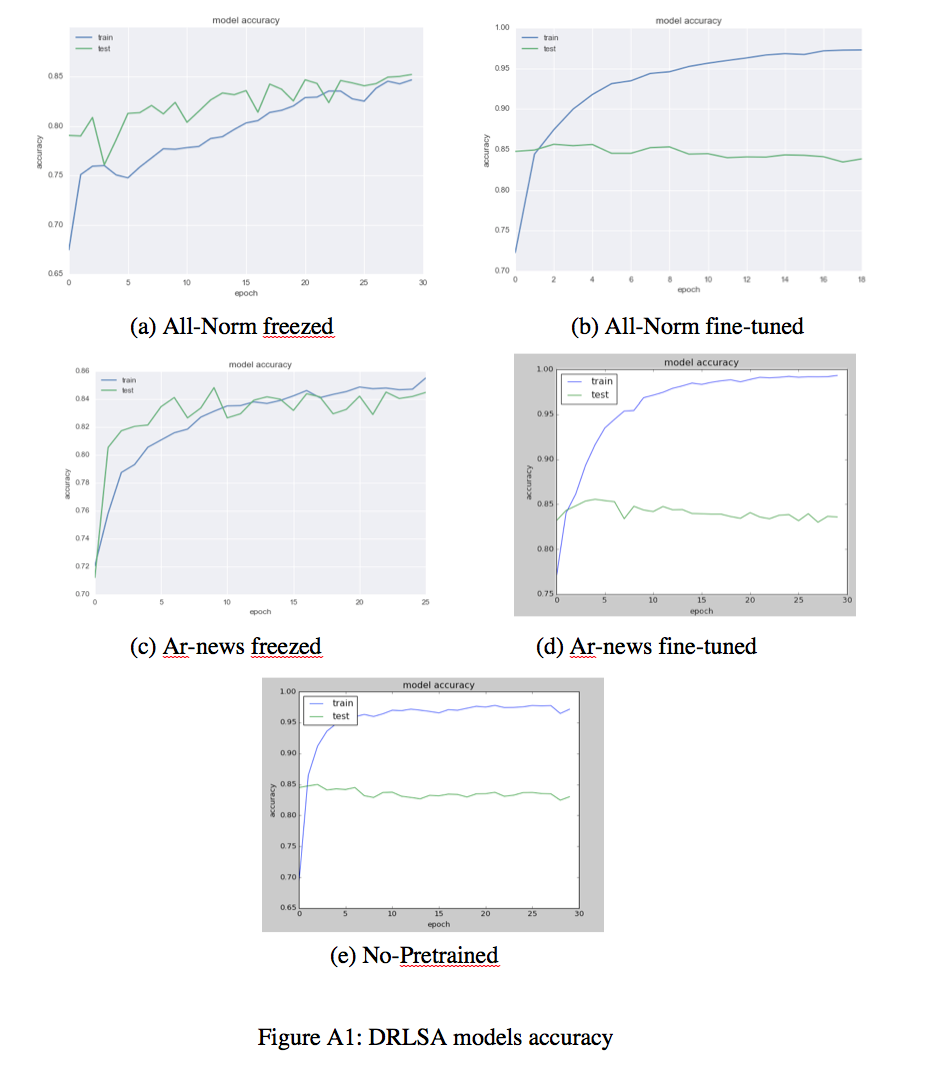
****

Figure A1: Attention-based DARLSA Models’ Accuracy

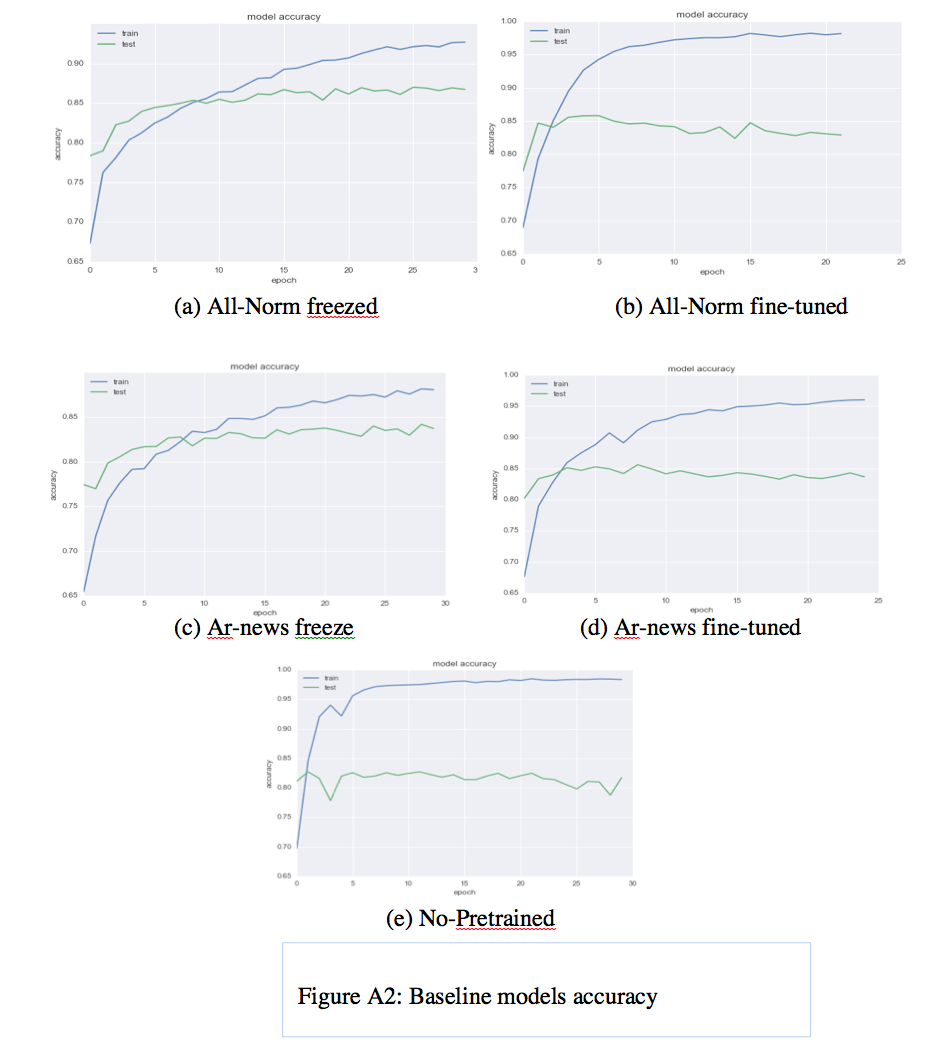
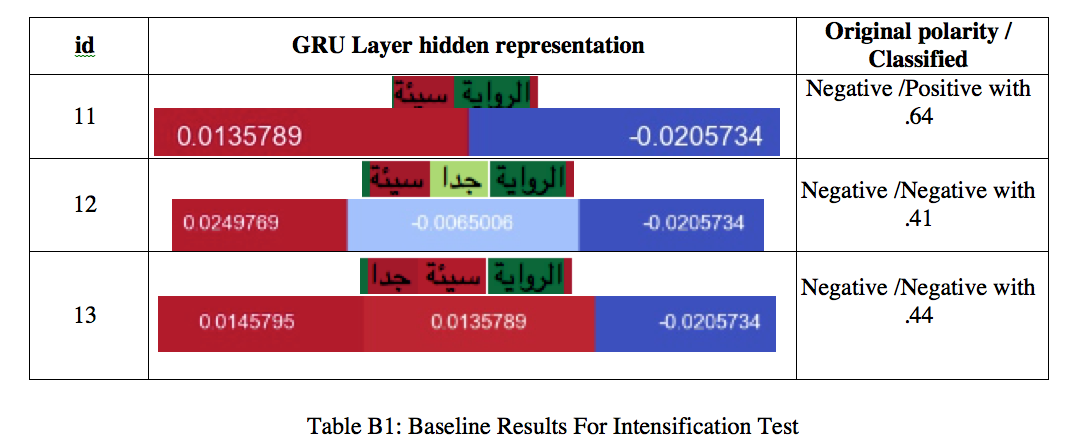
****

Figure A2: DARLSA Baseline Models’ Accuracy

# Appendix B: DARLSA Case Studies

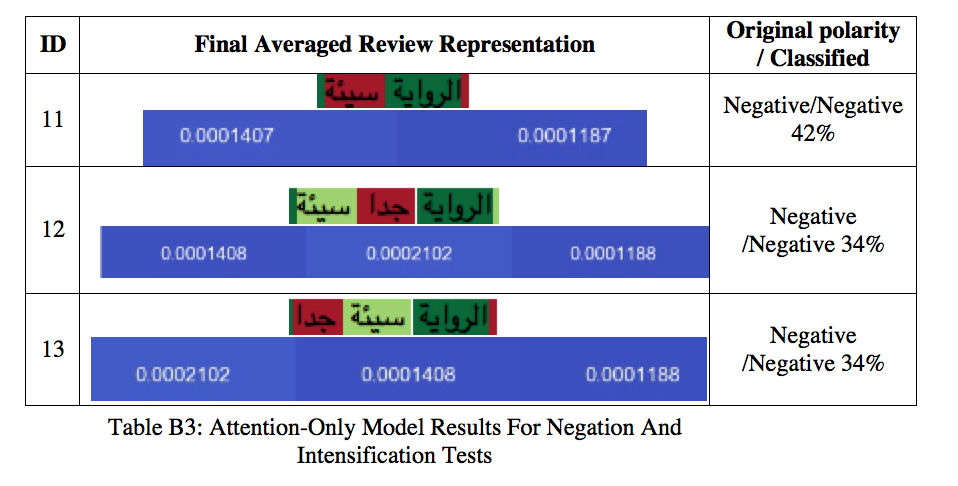
Examples 11, 12 and 13 are negative reviews used to prove findings inferred from examples 4 and 5 in section 7.2.2.2. For GRU-Base model, the original negative review, example 11, had been classified as negative with 0.49 probabilities. Then the probability to be positive decreased to be 0.46 in the case of using intensification keyword before negative adjective, example 12, and decreased more to 0.44 if it is used after the adjective. Table B1 shows the results for the baseline model.

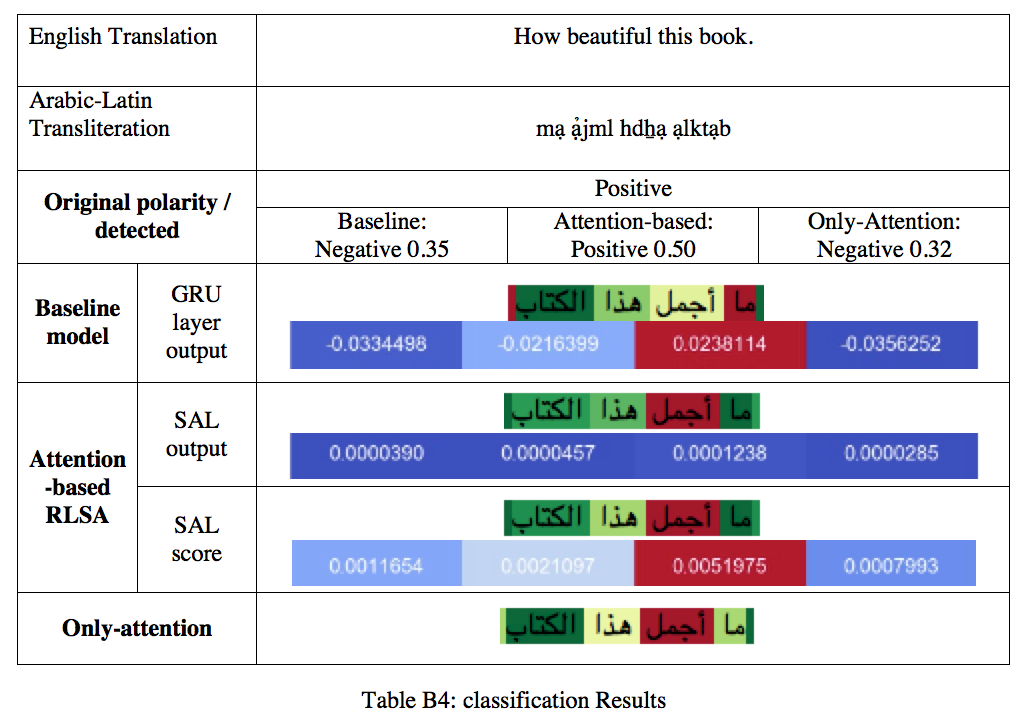


The same examples had been tested with attention-based DARLSA. Findings for examples 4 and 5 have been proved when the negative review tested, in examples 11, 12 and 13. As shown in table B2, The original negative review had been classified as negative with 0.49 probabilities. Then the probability to be positive decreased to be 0.46 in the case of using intensification keyword before negative adjective and decreased more to 0.44 if it is used after the adjective.

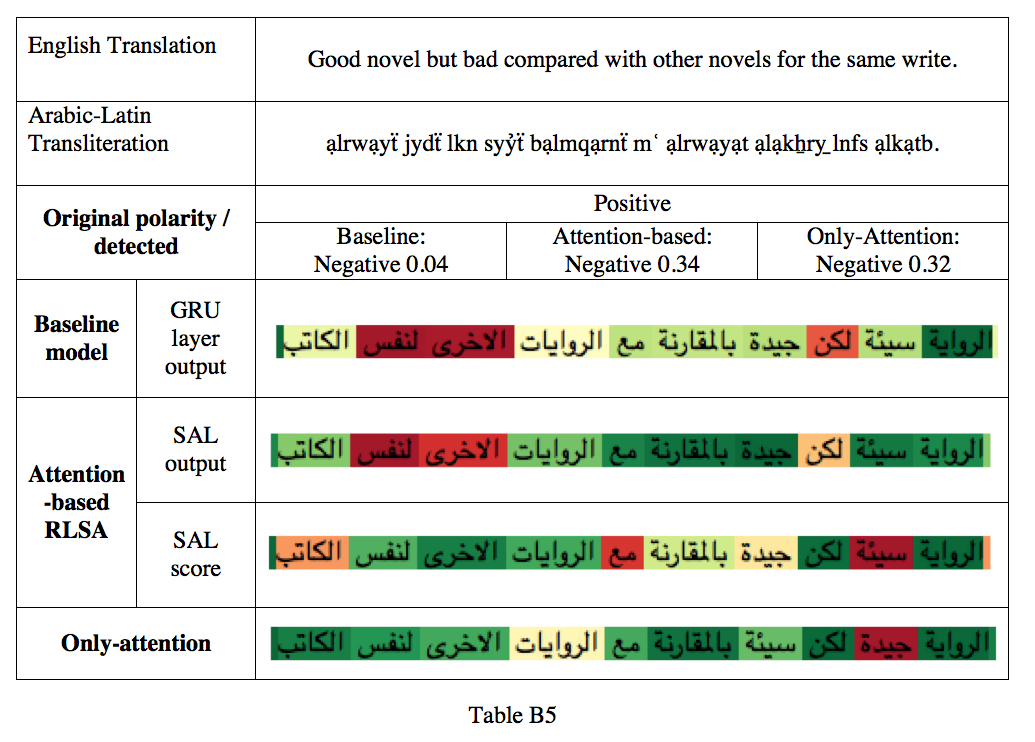


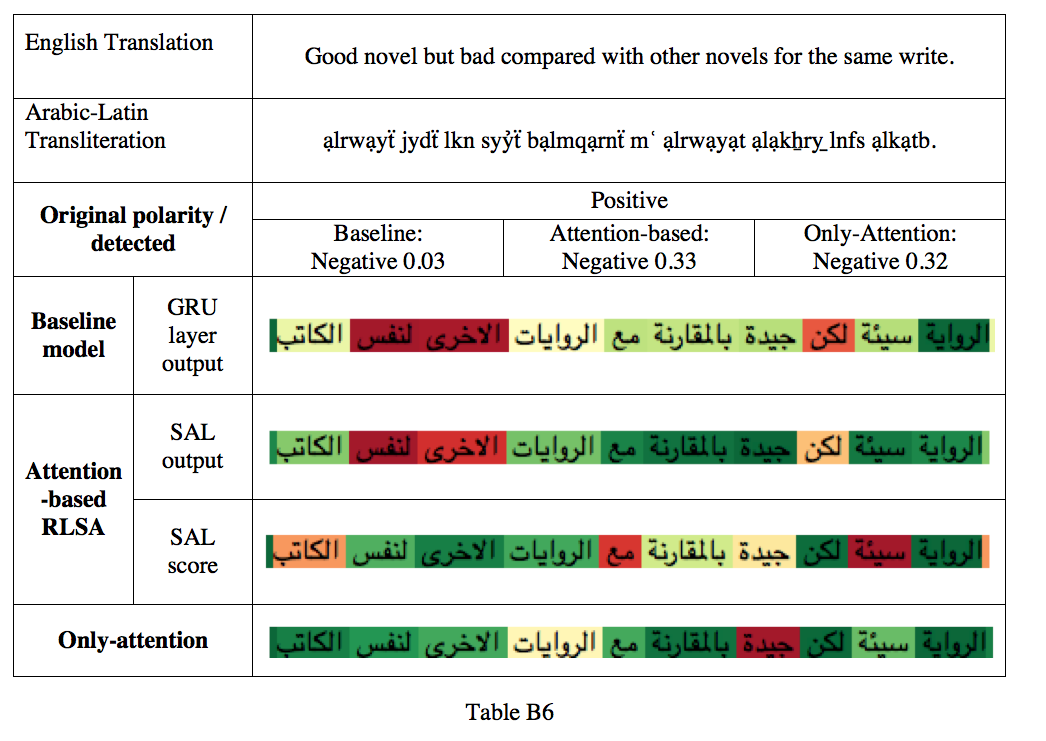
For attention-only model, the reviews had been classified correctly and the intensification keyword had been selected correctly in all examples, as concluded from table B3.



Here is another example conducted to test the ability of the models to detect the negation words used as intensification style. As shown from table B4, we can see that baseline model had failed to detect the polarity of the given review, and give the negation word the highest score in the review. In the other hand, attention-based model succeeded to classify the review and select the sentiment word correctly. Finally, attention only model, failed to detect the polarity but select the salient word correctly.

Another example is conducted to test the ability of the model to detect the polarity of concessive phrases. Two reviews were given one is started with positive sentiment and ends with negative one. And the other started with negative sentiment and ends with positive. As shown from tables B5 and B6, the tendency of the attention-based DARLSA model is to bias toward negative polarity.





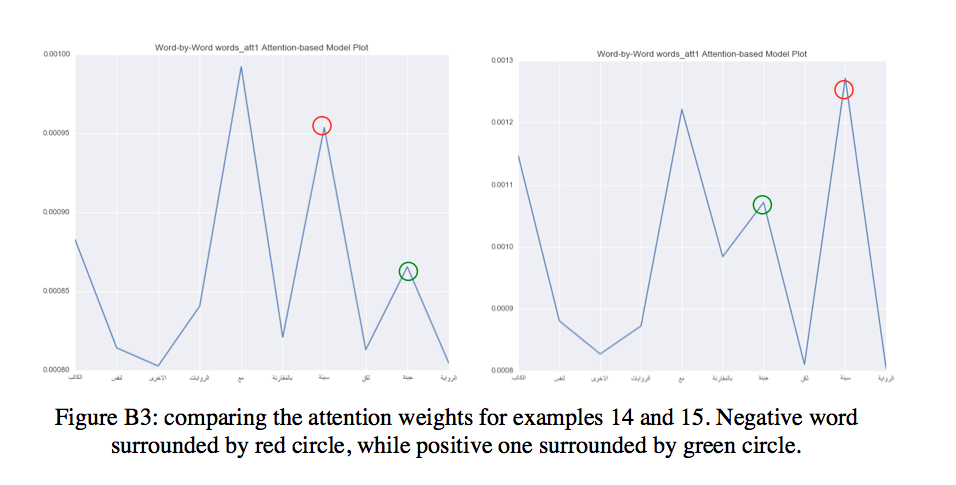
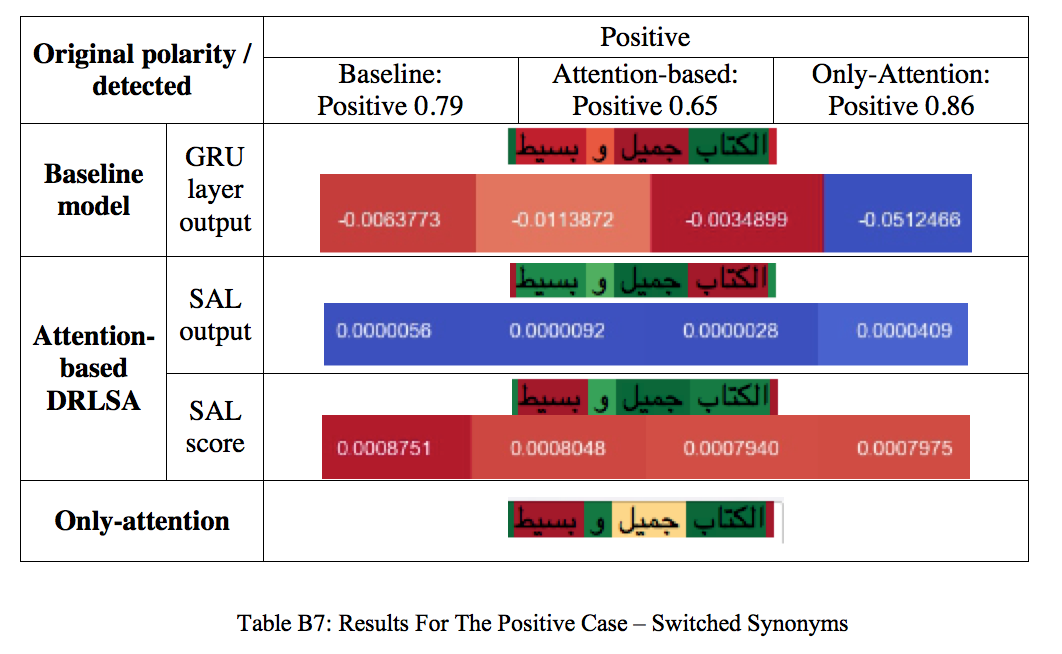
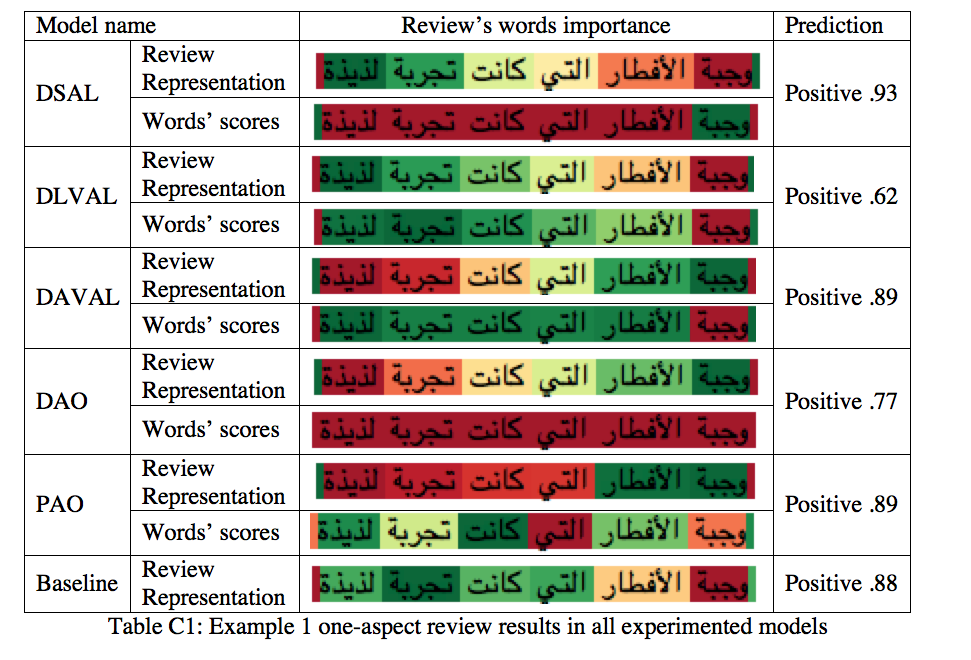
Plotting the attention weights could explain the behavior of the model towards this situation. We can see from figure B1 that attention-based DARLSA model tends to weight the negative sentiment more than the positive. In the figure, negative word is surrounded by red circle, while positive one is surrounded by green circle. Therefore, if the review contains contracted sentiment the model gives it negative polarity as it is weighted more.

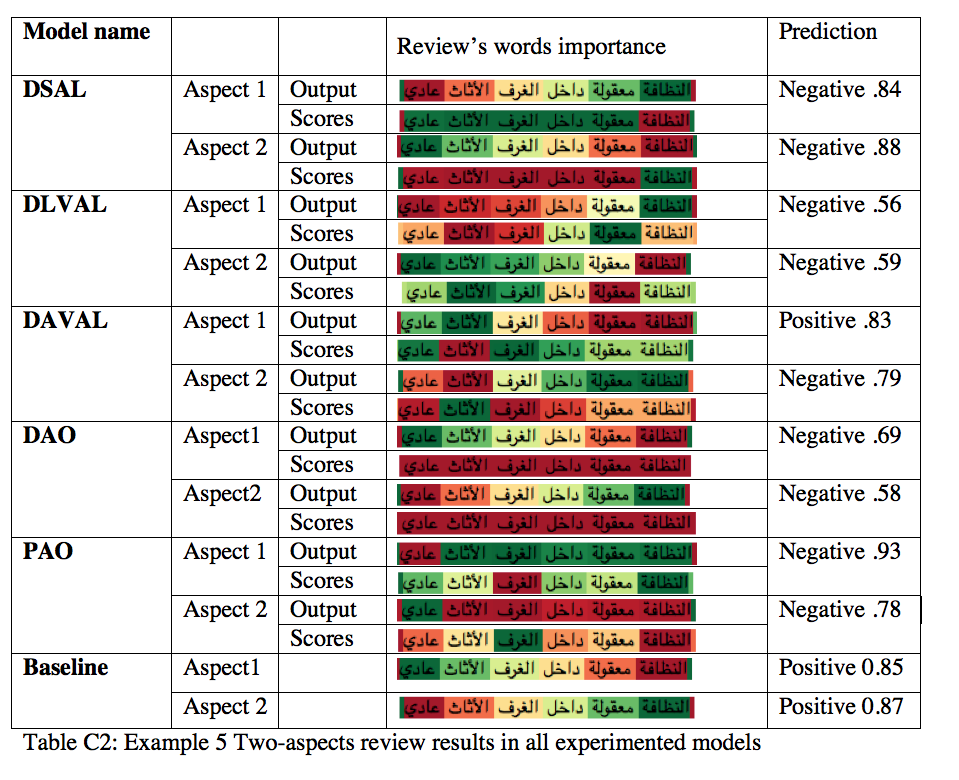
Figure B1: Comparing Attention Weights for Example 14 and 15.

Another test is conducted o make a justified analysis with example mentioned in table 6.3. The two words, “جميل”, “jmyl”, “beautiful” and “بسيط”, “bsyṭ”, “simple” were switched and the classification results were monitored as shown in table B7. Here, all models succeeded to classify the review correctly and we can see that GRU layer output preserve on its importance list with both cases. SAL out vector in the attention-based DARLSA also got the same importance rank.

Only-attention model has highlighted the word “بسيط”, “bsyṭ”, “simple”, to be the salient word that leads to the resulted polarity for both cases. This final representation is the resulted averaged three SAL representations.

# Appendix C: DAALSA Case Studies





1. <https://en.wikipedia.org/wiki/Arabic#cite_note-World_Arabic_Language_Day-12>. *.*  Retrieved 12 January 2019 [↑](#footnote-ref-1)
2. Using https://glosbe.com/transliteration/Arabic-Latin [↑](#footnote-ref-2)
3. Richard Eckel :”Microsoft researchers achieve speech recognition milestone”, September 13, 2016. <https://blogs.microsoft.com/ai/microsoft-researchers-achieve-speech-recognition-milestone/>

   Retrieved 18 January 2019 [↑](#footnote-ref-3)
4. Siri Team: “Deep Learning for Siri’s Voice: On-device Deep Mixture Density Networks for Hybrid Unit Selection Synthesis”. Vol. 1, Issue 4 ∙ August 2017

   <https://machinelearning.apple.com/2017/08/06/siri-voices.html>

   Retrieved 18 January 2019 [↑](#footnote-ref-4)
5. http://www.goodreads.com. [↑](#footnote-ref-5)