T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SENTIMENT CLASSIFICATION OF ARABIC TWEETS USING A NOVEL LEARNING SENTIMENT-SPECIFIC WORD EMBEDDING TECHNIQUE

Hala MULKI

Ph.D. THESIS

COMPUTER ENGINEERING DEPARTMENT

JULY - 2019

KONYA

ACCEPTANCE AND APPROVAL OF THE THESIS

Thesis titled as " *Sentiment Classification of Arabic Tweets using a Novel Learning Sentiment-Specific Word Embedding Technique*" prepared by Hala MULKI has been accepted as the PHD THESIS on 17/07/2019 by Selçuk University, Institute of Graduate Studies by the majority of the jury members from Computer Engineering Department.

TEZ KABUL VE ONAYI

Hala MULKI tarafından hazırlanan "*Yeni Bir Duygu-OdaklI Kelime Gömme Tekniği Kullanarak Arapça Tvitlerin Duygu Sınıflandırması*" tez adlı çalışması 17/07/2019 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Selçuk Üniversitesi, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri **İmza**

Başkan

Assoc. Prof. Oğuz FINDIK ____________

Danışman

Assoc. Prof. İsmail BABAOĞLU ____________

Üye

Assoc. Prof. Mustafa Servet KIRAN ____________

Üye

Assoc. Prof. Mesut GÜNDÜZ ____________

Üye

Assist. Prof. Mehmet HACIBEYOĞLU ____________

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mustafa YILMAZ

FBE Müdürü

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığınıbildiririm.

Hala MULKI
Date: 17/07/2019

ÖZET

DOKTORA TEZİ

YENİ BİR DUYGU-ODAKLI KELİME GÖMME TEKNİĞİ KULLANARAK ARAPÇA TVİTLERİN DUYGU SINIFLANDIRMASI

Hala MULKI

SELÇUK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALİ

Danışman: Doç. Dr. İsmail BABAOĞLU

2019, 150 Sayfa

Jüri
Doç. Dr. Oğuz FINDIK
Doç. Dr. İsmail BABAOĞLU
Doç. Dr. Mustafa Servet KIRAN
Doç. Dr. Mesut GÜNDÜZ
Dr. Öğr. Üyesi Mehmet HACIBEYOĞLU

“Arap Baharı” olayları sırasında sosyal medyanın yoğun kullanımı, Arapça görüşlü içeriğin artmasına sebep olmuştur. Duygu Analizi, gerçek zamanlı ve uzun vadeli görüşler sunarak paylaşılan metinlere gömülü görüşleri tanıyabilir. Sosyal medyadaki Arapça içeriğin diyalektik Arapça baskın olması nedeniyle, Arapça duygu analizi modellerinin, Arapça dilin karmaşık olmayan morfolojik doğası bir yana, Arapçanın standart olmayan gramer özelliklerini ve Arapça lehçeler arasındaki varyasyonları da ele alması gerekir.

Mevcut Arapça duygu analiz modelleri, diyalektik Arapça içeriğin duygusallığını el yapımı özelliklerle veya gömülü metinlerle temsil eder. El yapımı özellikler genellikle lehçeye özgü Doğal Dil İşleme (DDİ) araçları ve kaynaklarına göre oluşturulur. Bir diğer yandan, metin gömme özellikleri, derin sinirsel mimarilerde öğrenilen cümle/paragraf gömme işlemlerini üretmek için düzenli, söz dizimine duyarlı kompozisyon işlevlerini kullanma eğilimindedir.

Geçerli el yapımı ve gömme özellikleri ele alındığında bir lehçe için geliştirilen bir Arapça duygu analiz sistemi, özellikle lehçenin özgür kelime sırası, değişken söz dizimsel doğası ve Arapça lehçeler arasındaki esaslı söz dizimsel/anlamsal farklılıklarla diğer lehçeler için etkili olmayabilir.

Bu tezde, el yapımı ve metin gömme özellikleri ile donatılmış lehçe bağımsız iki Arapça duygu analizi modeli sunuyoruz. Her modelin kendine özgü duygu özellikleri ve sınıflandırma yöntemleri olsa da, her iki model de Arapça DDİ araçlarına en az bağımlı olarak ve dış bilgi kaynaklarına ihtiyaç duymadan birden fazla Arapça lehçenin duygu analizini sunulmaktadır. El yapımı temelinde olan Tw-StAR (HCB Tw-StAR) modelinde, evrensel metin bileşenleri Adlandırılmış Varlıklar (AV) ve ön işleme görevlerinin çeşitli kombinasyonlarını temel alan yeni el yapımı özellikler önerilmiştir. Sağlanan bu özellikler ile HCB Tw-StAR modeli, Arapça olan/Arapça olmayan içerikler için farklı analiz düzeylerinde geliştirilmiş bir duygusallık sınıflandırma performansı elde edebilir. Gömme özellikleri tabanlı sinirsel Tw-StAR (Neu Tw-StAR) isimli ikinci modelde ise, etiketli verilerden öğrenilen ve sırasız kelime gömme toplamı “Sum Of Word Embeddings (SOWE)” toplamsal kompozisyon işlevi kullanılarak oluşturulan yeni duygu-özgü, söz dizimi dikkate alınmayan n-gram gömme özellikleri sunulmuştur. Önerilen n-gram gömme özellikleri ile eğitilmiş olan Neu Tw-StAR modeli, literatürde temel model olarak kabul edilen “word2vec” ve “doc2vec” isimli iki söz dizimi temelindeki gömme metodundan daha iyi bir performans göstererek çok sayıda doğu ve batı Arapça lehçesini işleyebilme etkinliğini göstermiştir.

Ayrıca, sığ bir ileri beslemeli sinir modeli olarak uygulanan Neu Tw-StAR modeli, Konvolüsyonel Sinir Ağları ve Uzun Kısa Süreli Bellek gibi derin sinir modelleri ile karşılaştırıldığında yetenekli bir model olmuş, bazen daha iyi bir performans ve derin sinir modellerine kıyasla kayda değer ölçüde daha az eğitim süresi sergilemiştir.

Anahtar Kelimeler: Makine öğrenmesi, duygu analizi, adlandırılmış varlıklar, Arapça lehçeleri, el-yapımı özellikleri, metin gömme özellikleri.

ABSTRACT

Ph.D. THESIS

SENTIMENT CLASSIFICATION OF ARABIC TWEETS USING A NOVEL LEARNING SENTIMENT-SPECIFIC WORD EMBEDDING TECHNIQUE

Hala MULKI

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING

Advisor: Assoc. Prof. Dr. İsmail BABAOĞLU

2019, 150 Pages

Jury
Advisor: Assoc. Prof. Dr. İsmail BABAOĞLU
Assoc. Prof. Dr. Oğuz FINDIK
Assoc. Prof. Dr. Mustafa Servet KIRAN
Assoc. Prof. Dr. Mesut GÜNDÜZ
Assist. Prof. Dr. Mehmet HACIBEYOĞLU

The intensive use of social media during the “Arab Spring” incidents, has led to a sudden growth of the online Arabic opinionated content. Sentiment Analysis can recognize the opinions embedded in shared texts, providing real-time and long-term insights. With the Arabic social media data being dominated by dialectal Arabic, Arabic sentiment analysis models need to handle the complex morphological nature of the Arabic language, let alone, the non-standard grammatical properties and the variances among the Arabic dialects.

Existing Arabic sentiment analysis models represent the sentiment embedded in dialectal Arabic either by hand-crafted features or text embedding ones. Hand-crafted features are usually generated based on dialect-specific Natural Language Processing (NLP) tools and resources. On the other hand, text embedding features tend to use ordered, syntax-aware composition functions to produce sentence/paragraph embeddings learned within deep neural architectures. Given the current hand-crafted/embedding features, an Arabic sentiment analysis system developed for one dialect might not be efficient for the others, especially with the free word order, the varying syntactic nature and the drastic syntactic/semantic differences among the Arabic dialects.

In this thesis, two dialect-independent Arabic sentiment analysis models equipped with hand-crafted and text embedding features are presented. While each model has its own type of sentiment features and classification methods, they both perform sentiment analysis of multiple Arabic dialects with the least dependence on Arabic NLP tools and without the need for external knowledge resources. In the Hand-Crafted based Tw-StAR model (HCB Tw-StAR), novel hand-crafted features based on the universal text components Named Entities (NEs) and various combinations of preprocessing tasks are proposed. Provided with these features, HCB Tw-StAR could achieve an improved sentiment classification performance for Arabic/non-Arabic contents at different analysis levels. In the second model Embedding Features-based Neural Tw-StAR (Neu Tw-StAR), novel sentiment-specific, syntax-ignorant n-gram embedding features learned from labeled data and composed using the additive unordered composition function SOWE, are presented. Neu Tw-StAR trained with the proposed n-gram embeddings proved its efficiency to handle multiple Eastern and Western Arabic dialects, as it outperformed two state-of-the-art syntax-aware embedding methods: word2vec and doc2vec. Moreover, being implemented as a shallow feed-forward neural model, Neu Tw-StAR exhibited a competent and some times better performance, in addition it could decrease the consumed training time compared to deep neural models: Convolutional Neural Networks (CNN) and Long short Term Memory netwotks (LSTM) models.

Keywords: Machine learning, sentiment analysis, Arabic dialects, named entities, hand-crafted features, embedding features.

PREFACE

“*If I were to start a company today, the goal would be to teach computers how to read so that they can understand all the written knowledge of the world*”

-Bill Gates, *CNBC, 2019*

With the exponential growth of online textual contents spread across the different platforms of social media, solid text analysis technologies are needed to extract meaningful information out of the vast amounts of raw textual data. Natural Language Processing (NLP) powered by the revolutional Artificial Intelligence (AI) techniques could teach machines to read, understand and interpret written texts as humans do. This introduced a new level of human-machine interaction and set the scene for future smart applications. Today, one of the important tasks of NLP is sentiment analysis through which attitudes, preferences, and even mood can be recognized from a short piece of text. Sentiment analysis along with machine learning tools played an influential role in providing text-based evidences to guide the decision making process in many vital sectors such as the health, business and politics. As sentiment analysis continues to evolve going beyond the coarse-grained analysis into fine-grained analysis levels, it will be more engaged in many Natural Language Understanding (NLU) applications; among these applications, we can mention smart business assistants or chatbots, data-driven healthcare decision making systems and automated policies for prohibiting hate speech and racist content on social media.

Hala MULKI
KONYA-2019

ACKNOWLEDGMENT

A PhD is a long journey full of inspirations and creativity with new areas to discover and challenging issues to resolve. I am thankful for the many people I got to meet and accompanied me on the way.

First, I would like to thank my PhD supervisor Assoc. Prof. Dr. İsmail Babaoğlu for his sensible advice and patience, not to mention, his continuous scientific and practical support by which this study could be evolved within a productive research environment.

I also want to express my gratitude to my advisor and mentor Assist. Prof. Hatem Haddad who introduced me to the NLP world and gave me the freedom to pursue my interests and follow my curiosity towards new ideas and concepts.

I am also very fortunate to have such a caring and loving family who was always there for me and supported me in every single step during this journey.

Lastly, I would like to thank Turkey Scholarships (Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı) who funded my PhD and granted me the opportunity to study in a wonderful city like Konya.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

NLP	**N**atural **L**anguage **P**rocessing
Tw-StAR	**Tw**itter **S**entimen**t** Analysis for **AR**abic
HCB	**H**and-**C**rafted Features-**B**ased
NE	**N**amed **E**ntity
Neu	**Neu**ral
SOWE	**S**um **O**f **W**ord **E**mbeddings
CNN	**C**onvolutional **N**eural **N**etworks
LSTM	**L**ong **S**hort **T**erm **M**emory
SA	**S**entiment **A**nalysis
ASA	**A**rabic **S**entiment **A**nalysis
MSA	**M**odern **S**tandard **A**rabic
DA	**D**ialectal **A**rabic
Avg	**Av**era**g**e
RTs	**R**e**T**weets
URL	**U**niform **R**esource **L**ocator
POS	**P**art-**O**f-**S**peech
TF-IDF	**T**erm **F**requency **I**nverse **D**ocument **F**requeny
SO	**S**emantic **O**rientation
ML	**M**achine **L**earning
DL	**D**eep **L**earning
SVM	**S**upport **V**ector **M**achines
NB	**N**aïve **B**ayes
DT	**D**ecision **T**ree

ME	**M**aximum **E**ntropy
SFS	**S**traight **F**orward **S**um
DP	**D**ouble **P**olarity
NER	**N**amed **E**ntity **R**ecognition
TF	**T**erm **F**requency
KNN	**K** **N**earest **N**eighbor
DNN	**D**eep **N**eural **N**etwork
DBN	**D**eep **B**elief **N**etwork
DBN	**D**eep **A**uto **E**ncoder
RAE	**R**ecursive **A**uto **E**ncoder
CBOW	**C**ontinuous **B**ag **O**f **W**ords
SG	**S**kip **G**ram
Nu-SVM	**S**upport **V**ector **M**achines with the regularization parameter **Nu**
BNB	**B**ernoulli **N**aïve **B**ayes
MLP	**M**ulti **L**ayer **P**erceptron
BiLSTM	**Bi**directional **L**ong **S**hort **T**erm **M**emory
LLR	**L**ocal **L**inear **R**egression
OOV	**O**ut **O**f **V**ocabulary
CRFs	**C**onditional **R**andom **F**ields
LIBSVM	a **LIB**rary for **S**upport **V**ector **M**achines
NLTK	**N**atural **L**anguage **P**rocessing **T**ool **K**it
MLC	**M**ulti **L**abel **C**lassification
VSO	**V**erb-**S**ubject-**O**bject
SSWE	**S**entiment-**S**pecific **W**ord **E**mbeddings
DAN	**D**eep **A**veraging **N**eural network
CNN-MC	**C**onvolutional **N**eural **N**etworks-**M**ulti **C**hannel

PV-DBoW	**P**aragraph **V**ector- **D**istributed **B**ag **of** **W**ords
PV-DM	**P**aragraph **V**ector- **D**istributed **M**emory
***t*-SNE**	**t**-**D**istributed **S**tochastic **N**eighbor **E**mbedding
C	Size of the sliding window in Neu Tw-StAR model
M	Weight embedding matrix
w	Input word
i	Integer index of a word
V	Vocabulary size
d	Embedding dimension
$\mathbf{vec_i}$	One-hot vector of w_i
$\mathbf{v}_i$	Embedding vector of w_i
hl	The hidden layer
$\mathbf{O}_{\text{lambda}}$	Output of lambda layer
$\mathbf{W_{hl}}$	Weights of the hidden layer
$\mathbf{b_{hl}}$	Biases of the hidden layer
$\mathbf{O_{hl}}$	Output of the hidden layer
$\mathbf{h}\_\sigma$	Hard sigmoid activation function
$\hat{\mathbf{y}}$	Predicted sentiment label
y	Gold sentiment label
k	Number of the classes
θ	Weights and biases of Neu Tw-StAR
$\mathbf{J}(\theta)$	Loss function of Neu Tw-StAR

1. INTRODUCTION

"What's happening?" and *"What's on your mind?"* are the daily greetings of Twitter and Facebook to their users all over the world. Everyday, impressions, reactions and feelings of hundreds of millions of people are being shared across social media platforms. Twitter, Facebook and other micro-blogging systems are, therefore, becoming a rich source of feedback information in several vital sectors such as politics, economics, sports and other issues of general interest. Consequently, many analytical studies seek to explore and recognize online opinions aiming to exploit them for planning and prediction purposes such as measuring the customer's satisfaction, establishing sales and marketing strategies, tracking the popularity of election candidates or predicting results of an election or a referendum. Sentiment Analysis (SA) is a Natural Language Processing (NLP) task that facilitates performing such studies by providing the techniques and tools to mine the subjective content in a piece of text and categorize it into three main polarities: positive, negative or neutral. A further analysis of the sentiment can be performed at a finer level of granularity beyond the three primary sentiment classes, where specific human emotions, such as joy, sadness, anger,...etc. are recognized (Liu (2012)).

The SA problem has been addressed using either machine learning or handcrafted approaches. Both methods require considering the specifications of the given language while developing NLP tools and semantic/linguistic resources. Since English is the most common language on social media, it was tackled in the majority of the proposed SA research and was supported by a wide variety of NLP tools. With the recent rapid growth of the online Arabic opinionated content, Arabic Sentiment Analysis (ASA) has attracted the attention of the NLP research community especially with the numerous challenges it involves. These challenges are often related to the language special properties and the limited Arabic semantic resources and tools (Section 3).

Arabic is a Semitic language, spoken by more than 422 million people worldwide and classified by the British Council as the second most important language of the future (Council (2013)). According to Badaro et al. (2018), Arabic language has two main variants: (a) Formal Arabic known as Modern Standard Arabic (MSA) which is used in books and news and has standard grammatical rules and syntactic nature and (b) Informal or Dialectal Arabic (DA) which represents the colloquial language used in the daily communication in different Arab countries. While MSA is the official form of the Arabic language, it cannot be considered the mother tongue of any of the Arab countries. In contrast, DA denotes the linguistic identity of each country or region in the Arab world. Dialectal Arabic is drastically different from MSA; it combines a wide variety of dialects differ from one country to another and within the same country in syntax, semantics, words order and vocabulary (Chiang et al. (2006); Al-Kabi et al. (2013); Duwairi and El-Orfali (2014); Badaro et al. (2018)). Due to these complexities, most of the previous work has focused on the formal type of Arabic. therefore, providing a proper SA model to target the informal Arabic variant, which is widely used on social media, remains an interesting issue to investigate; particularly for under-represented Arabic dialects.

1.1. Motivation

Since the ”Arab Spring” that started in Tunisia at the end of 2010, there has been a sudden boost in the online Arabic content across micro-blogging systems. The number of Arabic tweets, for instance, has increased from 30,000 per day in 2010 to 2 million daily tweets in 2011 (Semiocast (2011)). For such rich opinionated data resources and within such major events, SA provides the means to analyze the political atmosphere in the internet landscape and to give an insight into the outcome of the domestic situation and its impact on several vital sectors. Therefore, many research studies have recently focused on developing ASA models within which several Arabic NLP tools and sentiment/semantic resources were introduced.

The success of any SA system is highly related to how the input text is represented. Therefore, manipulating the textual content through the proper NLP tools and preprocessing tasks can contribute in generating expressive sentiment representations or features and, thus, improves the quality of the sentiment recognition.

Considering the complex nature and rich morphology of the Arabic language, compared to Latin languages for example, advanced text manipulation and preprocessing are required to prepare the input textual data for the SA task. For this purpose, several Arabic NLP tools and preprocessing tasks have been developed; most of which targeted the formal Arabic variant (MSA) through modeling its standard syntactic and grammatical rules (Khoja and Garside (1999); Taghva et al. (2005); Abdelali et al. (2016)). In contrast, as each Arabic dialect has an unstructured, noisy data with neither standardization nor unified grammatical rules, it was remarkably less tackled in ASA research. This evoked further labor-intensive efforts to either develop dialect-specific NLP tools, or to adapt and exploit the existing MSA NLP tools on the basis of the common vocabulary between MSA and DA (Abdulla et al. (2013); Duwairi and El-Orfali (2014); Brahimi et al. (2016); El-Beltagy et al. (2017)).

Nevertheless, the drastic differences between most Arabic dialects and MSA make the dialectal sentiment features, generated by MSA-based NLP tools, of limited value (Habash et al. (2012)). In such case, recognizing the sentiment of DA would be better conducted, if subjectivity and sentiment indicative components, embedded in the text itself, could be captured and tagged during the preprocessing phase. This, on one hand, would enrich the SA models with more expressive sentiment features without the need to develop dialect-dependent NLP tools; and, on the other hand, could be generalized across the different Arabic dialects.

Arabic social media posts are rich of specific proper nouns denoting the names of masculine/feminine persons, geographic locations or official associations and business brands, known as named entities (NEs) (Yasavur et al. (2014); Jansen et al. (2009)). NE types are often correlated with major events took place in a certain period of time. Therefore, the polarity of a sentence containing an NE, posted during a specific period of time, is affected by this very NE and the attitudes towards it at that time. Thus, NEs can be thought of as universal sentiment features for DA; especially for being dialect-independent sentiment indicators. The role of NEs in ASA has not been tackled in previous studies; as NEs were ignored or eliminated in most of the proposed ASA models (El-Makky et al. (2014); El-Beltagy and Ali (2013)). We believe that, instead of ignoring or reducing NEs, they could be exploited in the preprocessing phase to generate more expressive sentiment features for DA and, hence, enhances the sentiment classification performance.

As seeking the best formula of sentiment features, generated with the least efforts, has always been the goal of all ASA systems. The novel type of features known as text embeddings are therefore considered an efficient replacement of the so-called hand-crafted features (LeCun et al. (2015)). This is because embedding features do not need preprocessing or NLP tools; instead, they are learned automatically from raw, non-preprocessed text and can surprisingly capture and incorporate the regularities and semantic/syntactic relations of words within a fixed-length, real-valued and low-dimensional vector (Bengio et al. (2003); Mikolov et al. (2013); Pennington et al. (2014)).

Text embedding features such as word, sentence, phrase or n-gram embeddings have been recently leveraged by ASA systems. For longer pieces of text, embedding vectors are composed out of their constituent word embeddings either with the words' order considered, or with it ignored. This introduces two compositionality types: ordered and unordered (Gormley et al. (2015)). While ordered compositionality can provide expressive sentiment features for the standard variant of Arabic MSA (Al Sallab et al. (2015)), it is not always guaranteed to have similar expressive features for DA; where the varying words' usage, syntactic and linguistic patterns might not be efficiently captured and represented within the composed embeddings vector. Therefore, we hypothesize that; unordered compositionality can produce efficient sentiment embedding features, that could address the challenges imposed by DA. This assumption is based on the ability of unordered embeddings to represent longer pieces of text regardless of the words' order which means that the syntactic information would be ignored, whereas the semantic and synonymous regularities would be better incorporated (White et al. (2015); Iyyer et al. (2015)).

On the other hand, while ordered compositionality is usually adopted by SA models of sophisticated deep neural architectures, unordered compositionality exhibit a low computation complexity (Mitchell and Lapata (2010)). Thus, unordered embedding features form an efficient option when it is aimed to design a less complicated SA neural model, while saving the time overhead (Ba and Caruana (2014); Iyyer et al. (2015)).

Pairing between simple embedding features and less complicated neural architectures has been studied in several research works dedicated for English SA (Iyyer et al. (2015); Shen et al. (2018)). However, no similar efforts have been recorded for ASA; as most of the proposed studies employed ordered/unordered embeddings within complicated deep neural architectures (Al Sallab et al. (2015); Dahou et al. (2016); Baniata and Park (2016); Al-Sallab et al. (2017)), where computation complexity and time overhead issues were never investigated.

In this study, a novel, less time-consuming and non-deep i.e. shallow neural ASA model is presented to be used across different Arabic dialects. In the proposed model, DA is efficiently supported by expressive unordered embedding features, that focus on the semantic and sentiment regularities and ignore the syntactic contextual information. Moreover, being of a shallow architecture, the presented model enabled conducting SA with less time overhead while retaining a high performance comparable to more complicated deeper models.

1.2. Research Goal

This dissertation aims to develop a SA model able to mine, recognize and analyze sentiments embedded in the DA content on social media. The proposed model was designed with the objective of being dialect-independent such that it could be easily applied across different Arabic dialects. Through our model, different Arabic dialects were tackled, novel hand-crafted and embedding features were proposed and several model variants and architectures were evaluated. The main contributions of this thesis are summarized in Figure 1.1.

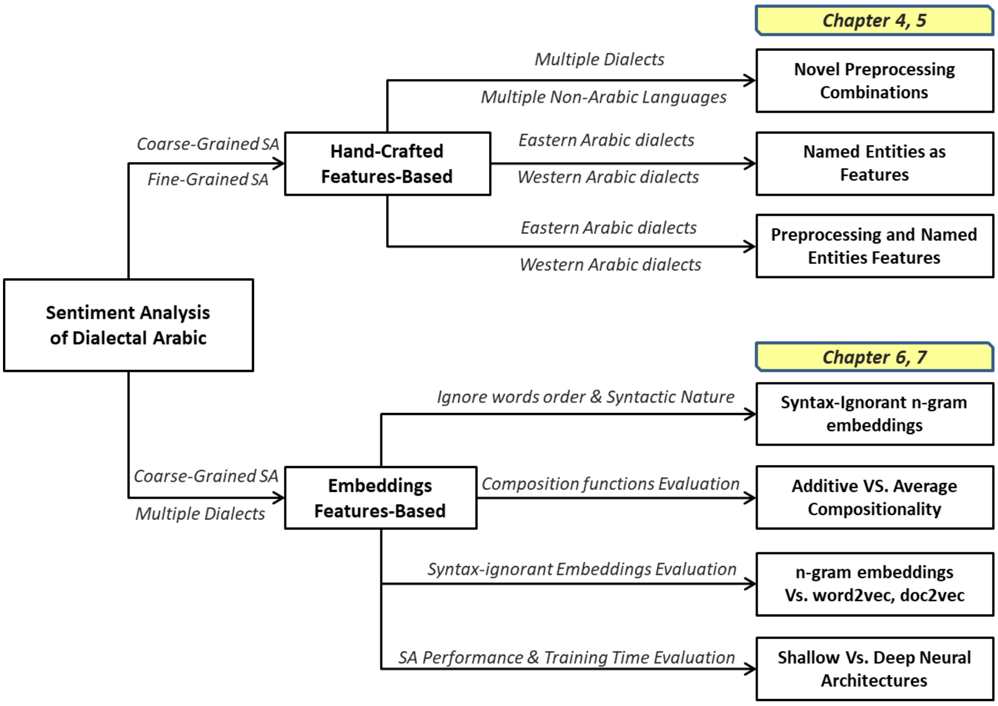

Chapter 4, 5
Multiple Dialects
Multiple Non-Arabic Languages
Novel Preprocessing Combinations
Coarse-Grained SA
Fine-Grained SA
Hand-Crafted Features-Based
Eastern Arabic dialects
Western Arabic dialects
Named Entities as Features
Eastern Arabic dialects
Western Arabic dialects
Preprocessing and Named Entities Features
Sentiment Analysis of Dialectal Arabic
Chapter 6, 7
Ignore words order & Syntactic Nature
Syntax-Ignorant n-gram embeddings
Coarse-Grained SA
Multiple Dialects
Embeddings Features-Based
Composition functions Evaluation
Additive VS. Average Compositionality
Syntax-ignorant Embeddings Evaluation
n-gram embeddings Vs. word2vec, doc2vec
SA Performance & Training Time Evaluation
Shallow Vs. Deep Neural Architectures

Figure 1.1. Thesis contributions

A detailed review of the thesis contributions can be listed as follows:

1. To the best of our knowledge, the role of NEs in SA within supervised and lexicon-based models, has not been investigated in the State-Of-The-Art. Here, we present a pioneering attempt to include NEs among the hand-crafted sentiment features considering them as sentiment indicatives (Section 4.2). For this purpose, we present an algorithm that correlates an NE with a specific sentiment polarity based on the local contextual content (Section4.2.2). Furthermore, compared to the presented ASA research, in which NEs were ignored or eliminated, we could successfully exploit NEs to infer the sentiment in Eastern (Levantine) and Western (Tunisian) Arabic dialects;

2. Through the proposed hand-crafted features-based SA model, we examine novel combinations of preprocessing tasks (Section 4.3). This enables the generation of hand-crafted n-gram sentiment features from the preprocessed and tagged text. The impact of the proposed combinations was not only proved to be positive for DA datasets but also for Turkish texts (Section 5.4). Moreover, adopting similar combinations of preprocessing tasks could yield an improved performance for the task of multi-label emotion classification applied for DA, English and Spanish textual contents (Section5.4);

3. Within our embedding features-based model, novel sentiment-specific, syntax-ignorant n-gram embedding features are learned from a raw input text and used for training. Previous studies adopted context-aware, syntactic-aware embedding methods which learned the embeddings from the so-called corrupted n-grams (missing one word) along with the original ones (Mikolov et al. (2013); Le and Mikolov (2014); Tang et al. (2014)). In contrast, given the challenging nature of DA, we assume that the syntactic information cannot be relied on to provide expressive features for DA. Therefore, our proposed embeddings are learned from non-corrupted, whole and original input n-grams such that the order and the syntax of the context words are both ignored, while the semantic/sentiment regularities were better captured and integrated within the resulting composed n-gram embeddings (Section 6.2);

4. In contrast to most of the Stat-Of-The-Art, where unordered embeddings were composed and learned within deep neural models (Iyyer et al. (2015); Dahou et al. (2016); Al-Azani and El-Alfy (2017); Baniata and Park (2016); Gri-

dach et al. (2017)), the embeddings introduced here are generated and learned within a shallow feed-forward neural model as we are seeking to accomplish the SA task of DA using a less complicated neural architecture and within a less training time (Section 6.2);

5. While previous studies have mostly employed the unordered average composition function (Avg) to produce sentiment embedding features (Le and Mikolov (2014); Iyyer et al. (2015)), we use the additive function; the so-called Sum of Word embeddings (SOWE) (White et al. (2015)) as an efficient replacement of the average function adopted by (Iyyer et al. (2015))(Section 6.2). To prove that, we investigate SOWE efficiency by conducting a comparison between the sentiment classification performances yielded from n-gram embeddings composed by SOWE and Avg functions, respectively (Section 7.2.3);

6. Due to the limited work on ASA, especially for under-represented Arabic dialects, it was not always possible to compare our shallow model with deep neural SA baselines. Therefore, we developed our own deep neural models using Convolutional Neural Networks (CNN) and Long Short Term Memory netwotks (LSTM) as building units; then applied these models to mine the sentiment in the tackled datasets. Hence, the presented model could be evaluated, in terms of the consumed training time and the achieved classification metrics, against more complicated and deeper neural models (Section 7.2.4, Section 7.2.6);

7. Through this study, and within the SA task of DA, we conduct a statistical/visual evaluation of our n-gram embedddings against syntax-aware, context-aware embedding methods such as word2vec (Mikolov et al. (2013)) and doc2vec (Le and Mikolov (2014)). The comparison involves exploring the sentiment classification performances produced by the presented syntax-ignorant n-gram embeddings towards those yielded from word2vec and doc2vec embeddings (Section 7.2.1). Moreover, using a proper visualization tool, we provide a visual representation for the proposed embeddings alongside word2vec and doc2vec embeddings, in a two dimensional space. Thus, based on the spatial relations among the mapped sentimental words, it becomes possible

to distinguish the most discriminating sentiment embedding features, among the three investigated embedding models (Section 7.2.2);

8. This study provides a SA model able to be used with both Eastern (Levantine) and Western (Tunisian, Moroccan) Arabic dialects. This is considered crucial for such under-represented dialects whose native speakers are among the most active users on social media since 2011 (Mayard (2013)). On the other hand, including these dialects emphasizes the ability of the presented model to bridge the drastic differences between Eastern and Western Arabic dialects (Section 7.1.1).

1.3. Research Contributions

Through the proposed SA model variants and the developed hand-crafted and embedding features, we seek to answer the following research questions:

1. Are NEs reliable enough to infer the DA sentiment within hand-crafted feature-based SA models? And is it more likely to have a better SA performance for datasets rich of NEs? (Section 5.3);

2. Which combination of preprocessing tasks can lead to an improved performance in hand-crafted features-based SA models? (Section 5.4);

3. Would the sentiment classification performance improved if NEs were included together with specific combinations of preprocessing tasks? (Section 5.5);

4. Compared to context-aware embedding algorithms: word2vec and doc2vec, can the proposed syntax-ignorant embeddings provide a better mapping of sentimental words and, hence, a better SA performance? (Section 7.2.1);

5. With Avg and SOWE composition functions being employed to compose our n-gram embeddings, which composition function can produce more expressive embedding sentiment features for DA? (Section 7.2.3);

6. How likely is it for a shallow neural model, trained with embeddings specifically formulated for DA, to rival complicated neural architectures? (Section 7.2.4);

7. At the implementation level, is it worthy to give up the newly-emerged deep architectures and adopt a feed forward shallow one, in return for reducing the consumed training time? (Section 7.2.6).

1.4. Thesis Outline

In Chapter 2, we provide the needed background to understand the research problem tackled in this thesis. We introduce the concept of the sentiment analysis problem and outline its importance and applications in multiple domains focusing on social media as the most important domain of SA. We further include a detailed description of the general pipeline used to solve SA problems along with the common sentiment classification methods adopted in the literature.

In Chapter 3, we explore the Arabic sentiment analysis domain focusing on the specificity of the Arabic language and the challenges it poses towards sentiment analysis. In addition we review the Arabic SA models, NLP tools, sentiment and semantic corpora and lexicons developed in the state-of-the-art. At the end of this chapter, we provide a summary of the reviewed studies highlighting their limitations. In light of the listed limitations, we propose a summary of both our SA models, where we outline the gaps it bridge and the merits it provide to handle the challenging nature of DA.

In Chapter 4, we describe our hand-crafted features-based SA model known as HCB Tw-StAR. Within the proposed model, we introduce named entities as sentiment indicatives and present a novel algorithm to exploit them in the sentiment analysis task. In addition, we employ novel combinations of preprocessings tasks to obtain more expressive sentiment features. At the end of the chapter, NEs and preprocessing tasks are both combined to train a supervised model or to assist in the lookup process of a lexicon-based SA.

In Chapter 5, we explore the experiments conducted to evaluate HCB-Tw-StAR. We focus on the ability of the proposed model to handle Eastern/Western Arabic dialects in addition to non-Arabic languages such as English, Spanish and Turkish through novel combinations of NLP preprocessings tasks. The efficiency of the introduced preprocessing combinations is then assessed for both coarse-grained (binary polarity classification) and fine-grained (multi-label emotion classification) sentiment analysis using DA/multi-lingual datasets. Moreover, we introduce Named Entities as sentiment indicatives and investigate their role in the SA task for Jordanian, Egyptian, Tunisian and Gulf dialects. Later, we evaluate the best-performing preprocessing together with Named Entities as sentiment features within supervised and lexicon-based SA models.

In Chapter 6, we propose our embedding features-based Neural model known as Neu Tw-StAR. First, we describe the layers that composes the shallow neural architecture of our model outlining the function of each of them. Then, we review the sentiment embeddings generation and learning mechanism in addition to the parameters adopted by each layer. Finally, we provide the training details used to tune the model in terms of parameters calibration and optimization.

In Chapter 7, we review the experimental study carried out to evaluate Neu Tw-StAR as an efficient SA model of Eastern/Western DA. We, first, investigate the ability of the learned syntax-ignorant n-gram embeddings to efficiently represent the DA sentiment compared to state-of-the-art, context-aware, syntax-aware embedding algorithms. We further examine how expressive are our n-gram features, based on their embedding visualization maps. Then, we justify our selection for the additive composition function by exploring its performance against those of the average composition function. At the implementation level, we investigate the ability of our model's shallow architecture to rival more complicated, deep neural architectures and the baseline models in terms of the achieved sentiment classification performances and the consumed training time. By the end of this chapter, we provide a comprehensive assessment of the proposed model highlighting the merits it introduces to support the specificity of DA.

Chapter 8, finally, combines the research conclusions through a summary of the findings and provide an insight into the future work.

2. SENTIMENT ANALYSIS

This chapter includes the key concepts related to the thesis research topic. Here, we introduce the definition of the sentiment analysis problem and its applications in the social media context, describe the general pipeline adopted in SA systems and review the common sentiment classification approaches adopted in the state-of-the art.

2.1. Sentiment Analysis Problem

Sentiment refers to the human attitudes, judgments, views or emotions towards entities, events, ideas or concepts (Turney (2002); Liu (2012); Pozzi et al. (2016)). In NLP research domain, some researchers differentiate *"sentiment"* from *"opinion"*, considering that *"sentiment"* reflects the feeling while the latter refers to the concrete judgment of the writer. Nevertheless, both terms are being used interchangeably in the majority of SA research based on the fact that in most cases, sentiments and opinions are strictly related to each other through a reason-result relationship; where an opinion can indicate a specific feeling or sentiment and vice versa (Pozzi et al. (2016)). To clarify that, the opinion in a sentence like *"I think that the performance of Win 10 is fantastic"* implicitly shares the same appraisal feeling expressed in the sentence *"I liked the Win 10 release!"*. In this thesis, we adopt the point of view of most SA studies considering that sentiments are an equivalent of opinions.

The sentiment embedded within written online contents usually implies a positive, negative or neutral polarity (Turney (2002)). Moreover, at a fine-grained analysis level, sentiment is represented by positive/negative emotions such as love, happiness, joy, surprise, anger, hate, pessimistic and so on.

According to Liu (2012) and Pozzi et al. (2016), sentiment analysis (SA) or opinion mining aims to develop automated techniques to analyze the opinions encountered in a piece of text where an opinion is formally identified as a quintuple

$(\boldsymbol{e_i}; \boldsymbol{a_{ij}}; \boldsymbol{s_{ijkl}}; \boldsymbol{h_k}; \boldsymbol{t_l})$ where:

- $\boldsymbol{e_i}$: the name of an entity that could be a restaurant, organization, person, etc.

- $\boldsymbol{a_{ij}}$: an aspect of the entity $\boldsymbol{e_i}$ such as the food quality at a restaurant or the WiFi service at a hotel. In the case the opinion is required for the whole entity, the special value GENERAL is used.

- $\boldsymbol{s_{ijkl}}$: the sentiment on an aspect $\boldsymbol{a_{ij}}$ of the entity $\boldsymbol{e_i}$ which might be positive, negative, neutral or have different levels of intensity on a specific scale.

- $\boldsymbol{h_k}$: refers to the opinion holder either a person or an organization.

- $\boldsymbol{t_l}$: the time at which the opinion was expressed by the opinion holder $\boldsymbol{h_k}$.

Based on the previous definition, unstructured raw texts are transformed into a structured data type such that it could be handled by computational language models (Pozzi et al. (2016)). Sentiment analysis can be conducted at several linguistic levels: word or phrase, aspect, sentence and document (Liu (2012); Piryani et al. (2017)). They are defined as follows:

- Document-level: where a piece of text is analyzed as a whole then an overall sentiment is given (Kolkur et al. (2015)).

- Sentence-level: which provides the sentiment for each sentence in a dataset (Collomb et al. (2014); Bongirwar (2015)).

- Entity-level: recognizes the sentiment related to specific aspects in a piece of text (Kolkur et al. (2015)).

- Word-level: it identifies the polarity or semantic orientation of subjective terms (words/phrases) in a dataset (Hercig (2015)).

In the last decade, many computational social science studies have focused on sentence-level SA to cope with the widespread of micro-blogging platforms where opinions are mostly shared in the form of sentences. In addition, sentence-level SA can essentially support several opinion mining applications such as opinion question/answering, summarization and opinion retrieval (Yang and Cardie (2014)).

2.2. Sentiment Analysis Applications

In a world where internet penetration ratios have become extremely high, it is not surprising that more than 2.5 quintillion bytes of data are generated and shared every day (Marr (2018)). This has led many companies, research centers and even ordinary customers to adopt data-driven decision making strategies. In this context, SA plays a key role as it can make sense of the online textual data and, thus, obtains real-time and long term insights in multiple vital domains such as:

- **Politics**: politicians, today, can easily reach their voters, proponents and opponents through micro-blogging and broadcasting systems. With politicians-public interaction data being analyzed using SA techniques, politics is now managed in a different way where the real-time outcomes of SA are exploited to reformulate a candidate's image, reshape a presidential crucial decision or draw road-maps and future policies (Ringsquandl and Petkovic (2013); Magdy and Darwish (2016)).

- **Economy**: many investors, traders and financial analysts are carefully tracking specific social media posts as reliable inspiration for their subsequent steps. The reactions of individuals towards major events such as political crises and social incidents can be considered as an economic data point in itself. Sometimes to move ahead the market and other times to shed light on new markets. Several studies have introduced SA as an economical analysis/prediction tool to serve in multiple economic applications including business conditions and stockmarket analysis (Bollen et al. (2011); Ruiz-Martínez et al. (2012); Bharathi and Geetha (2017); Chang and Wang (2018)).

- **Health**: The ability to reveal opinions embedded in clinical narratives, e-health forums and patients blogs enables health professional to understand and improve the patients experience. Given that medical facts are usually expressed via sentimental words and phrases (e.g. The surgery was completed successfully), therefore, SA of health-related texts can indicate critical information such as the health status of a patient, the effectiveness of a treatment or the certainty of a diagnosis (Denecke and Deng (2015)). Recently, developing medical context SA models and supporting them with domain-specific resources have been the focus of many studies; especially with the increasing demand for drug assessment and automated diagnosis systems (Carrillo-de Albornoz et al. (2018); Satapathy et al. (2018); Yadav et al. (2018)).

- **Marketing & Advertisement**: according to (Liu (2012)), online opinions are mostly composed of reviews of products. Being publicly shared and easily accessed by millions of users, online reviews are becoming of a significant impact on the reputation of a firm as they can control the purchasing decisions of new customers (Shayaa et al. (2018)). Hence, most organizations have developed SA-based marketing strategies to timely fix issues and to avoid customers churn (Rambocas et al. (2013)). On the other hand, tracking products-related opinions has contributed in the emergence of the intelligent online advertisement concept where customers are targeted based on their own preferences (Adamov and Adali (2016); Al-Otaibi et al. (2018)).

Considering the aforementioned applications of SA, it is obvious that in the era of Web 4.0 technology, social media have become the largest pool from which multi-domain valuable informative data can be retrieved. This explains why SA of social media has recently sparked increasing attention in the NLP research community leading to a revolutional development of SA tools, resources and learning methods (Piryani et al. (2017)).

2.3. Challenges of Social Media Sentiment Analysis

Despite being a fascinating problem, SA of social media is not a trivial task as it involves dealing with user-generated contents (Saif et al. (2016)). Such textual contents are different from any other types of raw data and difficult to be analyzed which poses multiple challenges towards social media SA systems. To name the main challenges of social media SA, we can list the following:

1. Length of posts: social media messages are usually very short either for readability purposes or due to length limitations imposed by some micro-blogging systems such as Twitter. A tweet or a Facebook comment may have few words, yet, can be semantically rich and adequate to imply the feeling or the opinion of the writer (Zhang et al. (2018)). To compensate for the lack

of content, SA methods need to employ additional information derived from external semantic resources or obtained based on specific markers within the textual message itself (Kiritchenko et al. (2014); Pozzi et al. (2017)).

2. Noisy content: due to text length limitations, social media users tend to condense their posts using abbreviations (e.g. OMG, LOL, ILY, etc.), badly-formed words (e.g. 2morrow) or specific punctuation patterns (e.g. ”:)) ;)”). On the other hand, some users emphasize their meant sentiment via word lengthening (e.g. Superrrr) or by combining expressive graphical symbols known as emoji (e.g. ☺,☹). Moreover, in some micro-blogging platforms, additional symbols or characters are automatically injected within the posted messages as in Tweets (e.g. RT, @, #). With all that random, unstructured, badly-written content, handling social media texts forms a difficult task to SA model developers where they have to clean and normalize these raw texts while retaining and tagging some noisy content for its potential ability to indicate the sentiment (Saif et al. (2016); Mohammad (2017)).

3. Ambiguity: being a user-generated content, social media posts combine various expression styles to deliver the sentiment. While some users express their opinions explicitly, others adopt indirect expression such as sarcasm in which the written content implies an opposite sentiment of the user’s actual opinion (Pozzi et al. (2017)). In addition, it is common to encounter posts containing words of contradict sentiments (e.g. The film was extensively horrible, I enjoyed it!) or words preceded by negation tools (e.g. I don’t like pasta). Such texts are considered tricky and ambiguous for SA models as it is difficult to recognize the correct sentiment unless a proper handling of the misleading content is provided (Sumanth and Inkpen (2015)). This was performed either by using deep learning (DL) systems equipped with semantic compsitionality learning techniques (Poria et al. (2016); Pasha et al. (2016)) or through exploiting specific text-derived markers such as emoji, negation, certain phrases which contribute in the detection of the the sarcastic-, negated- and conflicted sentiment issues and, thus, enhance the quality of sentiment recognition (Tungthamthiti et al. (2014); Hung and Chen (2016); Mukherjee and Bala (2017)).

4. Informality: with, almost, no presence of constraints over the shared textual content on social media, users prefer to use informal or colloquial language in order to reach the majority of the public. Consequently, capturing the sentiment, based on the traditional text features, becomes more difficult as informal languages have neither unified grammatical rules nor syntactic structure (Iyyer et al. (2015)). Moreover, since users do not commit to the spelling rules, typos are frequently found within the posted messages leading to several writing shapes of the same word (Kiritchenko et al. (2014); Pozzi et al. (2017)). These issues were investigated in recent studies where informal language-dedicated tools and resources have been employed to produce sentiment features for social media texts (Taboada et al. (2011); Thelwall et al. (2012); Socher et al. (2013); Thelwall (2017); Rout et al. (2018)).

2.4. Sentiment Analysis Pipeline

When exploring the state-of-the-art in the SA domain, it could be observed that in most of the proposed SA models, a unified series of processes were followed in order to end up with the predicted polarity labels of an input text. In the following subsections, we will review, in details, the phases adopted to develop SA models.

2.4.1. Data Preprocessing

Preprocessing is a crucial step in the development pipeline of any SA model. It aims to reduce the complexity and noisy nature of the input text especially the one derived from informal resources such as social media. Preprocessing phase involves subjecting the input raw data to a series of NLP-based techniques which on hand normalize, clean and eliminate the non-sentimental content and on the other hand, can detect and mark the potential sentiment indicators within the processed text. Among the most common preprocessing tasks employed in SA models, we can list the following:

- Text normalization: platform-inherited noisy components such as the the symbols of retweets (RT), mentions (@), hashtags (#), URLs,...etc. are frequently encountered within social media texts (Satapathy et al. (2017)). As these components have no impact on the text polarity, retaining them would just increase the dimensionality of the sentiment classification problem (Pozzi et al. (2016)). Therefore, the very first step in data preparation for SA is to remove such noisy content or replace it with proper tags.

- Tokenization: is the process of breaking down a piece of text into smaller meaningful chunks or tokens such as words, phrases, clauses or sentences. Tokenization enables obtaining the text statistical properties along with the syntax/semantics information born by tokens; which is considered essential to generate the features in the subsequent phase (Sarkar (2016)). Text tokenization is conducted based on the recognition of orthographic conventions such as white spaces, hyphenation and punctuation. Special tokenizers are needed to handle the social media texts where orthographic conventions are remarkably less and difficult to be detected as they might be confused with alphanumeric symbols (punctuation used as emoji) (Owoputi et al. (2013)).

- Stopwords removal: stopwords (e.g. prepositions, determiners, pronouns, conjunctions or year/day names) are function words with high frequency of presence in texts (Ghag and Shah (2015)). They, mostly, do not carry significant semantic meaning by themselves as their role is limited to modify other words or define grammatical relationships. Within the context of sentiment analysis, stopwords are usually eliminated using pre-compiled stoplists; where best performances could be obtained for stoplists constructed considering the specific characteristics of the studied language (Saif et al. (2014)).

- Stemming: concerns about reducing the variants of inflected words to their shared basic form known as stem or root (Duwairi and El-Orfali (2014)). This is done by stripping the word's suffix and prefix representing variations of the words as a single token. Consequently, stemming can be considered a feature reduction step as it significantly reduces the vocabulary size and, hence, the dimensionality of the generated feature vectors leading to less processing time and increased recall (Darwish and Magdy (2014)). Most of the stemmers were designed to target formal language variants (Khoja and Garside (1999); Porter and Boulton (2002e); Taghva et al. (2005); Sirsat et al. (2013)); however, some recent morphological analyzers have combined stemmers that support colloquial languages (Pasha et al. (2014)). It should be noted that, in specific scenarios, common affixes are removed from words without reducing them to their stems or roots. This is done by a stemming variant called light stemming (Abdulla et al. (2013)).

- Lemmatization: lemmatization shares the same principle of stemming, however, unlike stemming, which may produce invalid or language irrelevant stems, lemmatization ensures that a group of inflected words will be mapped into a root word that belongs to the tackled language (Di Nunzio and Vezzani (2018)). To clarify that, considering the words "*accusing*" and "*accused*", having these word stemmed, would give the root "*accus*" which is not a valid English word; whereas when subjecting these two inflected words to lemmatization they will be reduced to the valid base word "*accuse*". This is due to the fact that lemmatization chops off only the inflectional endings of a word yielding its canonical form or dictionary form known as Lemma (Liu (2012)). Lemmatizaters are developed based on POS-tagged dataset or lookup table derived from a dictionary and have been used in information retrieval and SA applications as an effective feature reduction step (Plisson et al. (2004); Ingason et al. (2008); Abdelali et al. (2016)).

- Emoji tagging: emoji are special iconic symbols used frequently in social networks to reflect specific emotions, ideas or opinions. Recognizing emoji and tagging them with textual expressive tags can produce a clean input text, enable automatic sentiment annotation for large corpora and assist in indicating the embedded sentiment considering tags as informative features (Guibon et al. (2016); El-Beltagy et al. (2017)).

- Negation handling: from a linguistic perspective, negation is the process that can turn an affirmative statement into its opposite denial and, thus, flips the polarity implied by that statement (Wiegand et al. (2010)). The majority of SA models exploit sentiment-bearing words or expressions to predict the polarity. Therefore, given the ability of the negation terms to alter the sentiment of a word next to them, negation contexts detection and negation terms tagging would assist in inferring the sentiment more accurately (Dadvar et al.

(2011); Sharif et al. (2016); Nakov (2017)). This is usually performed based on semantic lexicons or pre-compiled lists of negation terms related to the tackled language (Farooq et al. (2017)).

The preprocessing impact on sentiment analysis has been investigated in many studies. Most of them emphasized that subjecting the input textual data to specific preprocessing strategies can favorably affect the sentiment classification performance. While normalization, stemming and lemmatization reduce the dimensionality of the generated feature vectors and enhance the classification performance, tagging sentiment indicative components such as emoji and negation assist in indicating the implicit sentiment especially within informal, ironic or sarcastic contexts (Shoukry and Rafea (2012a); Uysal and Gunal (2014); Duwairi and El-Orfali (2014); Brahimi et al. (2016); Angiani et al. (2016); El-Beltagy et al. (2017)).

2.4.2. Feature Extraction

Feature extraction is an important step in the SA pipeline as it is a essential for training the SA model. Sentiment features are defined as a set of distinctive useful attributes of the textual input which might be words, tags, specific counts,..,etc. Extracted features are usually incorporated within n-dimensional numerical vectors. The sentiment features used in the state-of-the-art can be categorized into:

1. **Hand-crafted features**: refer to those features which are extracted based on the Vector Space Model concept (Sarkar (2016)). The vector space model exploits dataset terms, either words or ordered sequence of words i.e. n-grams, to transform and represent raw textual data (documents/sentences) into numeric vectors of n dimensions (Sarkar (2016)). Where n is the vocabulary size of the dataset while the values of a document/sentence vector are computed for all the terms contained in that document/sentence and reflect the terms frequency, terms presence/absence or terms importance represented by term frequency-inverse document frequency (TF*IDF) (Saif et al. (2012)). Among multiple hand-crafted features extraction techniques, bag-of-words and bag-of n-grams are the most naive, though, effective ways to generate text features and formulate them in a proper shape needed for the subsequent classification

phase (Abbasi et al. (2008); Bespalov et al. (2011)). Bag-of-words and n-grams features can be enriched with further text-based features such as: (a) Syntactic: are the outcome of certain preprocessing tasks (stemming, POS tagging and lemmatization), (b) Stylistic: they are more about the structure of the text than the content as they combine lexical attributes and special symbol frequencies like the count of exclamation/question marks or the presence of emoji and (c) Semantic: work on tagging specific tokens or contexts with the proper semantic orientation (SO) using external semantic resources such as sentiment lexicons. Consequently, hand-crafted feature vectors may contain additional numerical values that indicate the quantitative scores related to stylistic/semantic features (Refaee (2017)).

2. **Text embedding features**: also known as distributed text representations; they are discriminative features learned automatically from the text using multi-layer nonlinear neural networks. The learning process involves transforming the representation at one level into a representation at a higher and more abstract level (LeCun et al. (2015). Thus, the dataset vocabulary are mapped into unique points in the embeddings space where each point is a real-valued, low-dimensional embedding vector. Text embeddings features can be divided into two main types: (a) Word embeddings: where every word in the dataset is projected to an embedding vector using one of the word mapping algorithms such as word2vec (Mikolov et al. (2013)) and GloVe (Pennington et al. (2014)) and (b) Document or paragraph embeddings: in which continuous representations are generated for larger blocks of text such as phrases, sentences, paragraphs or whole documents using a document mapping algorithm such as doc2vec (Le and Mikolov (2014)).

Using hand-crafted features in SA models has led to good performances. However, hand-crafted features generation is a labor-intensive task that requires language-specific or dialect-specific morphological tools (Piryani et al. (2017)). Moreover, the high dimensionality and sparsity of hand-crafted feature vectors may drown the classifier with noisy features or lead to memory issues (Duwairi et al. (2014)). On the other hand, lexicon-derived features generated using a certain dialectal lexicon might not be efficient for other datasets even within the same dialect. This is due to the fact that, most lexicons are dataset-based which makes them domain-specific and dataset-specific while highly-coverage and large-scale dialectal lexicons are, relatively, difficult to build and compile (Abdulla et al. (2013)). Hence, many SA systems replaced the hand-crafted features with word/document embeddings to either train sentiment classifiers or enrich sentiment lexicons.

2.4.3. Sentiment Classification

The subjectivity concept of a piece of text usually include the opinion, feeling and sentiment aspects of the writer (Wiebe (1994)). Hence, the sentiment mining process involves conducting a subjectivity classification task first such that a text unit (term, phrase, sentence or document) is classified as either sentiment-free i.e. objective (e.g. *iPhone new series have been released*) or subjective (e.g. *Samsung Galaxy S9 is outstanding in every way!*). The subjective text is, then, classified into the polarity it implies which might be positive (e.g. *It was an amazing experience!*), negative (e.g. *What a bad performance of Barsa today* ☹), neutral (e.g. *I think Russia should withdraw forces from Syria*) or even mixed (e.g. *Asus new notebook has a brilliant display but it weighs too much*). Beyond these three polarities, and at a finer granularity level, the subjectivity text can be mined for multiple emotions such as anger, sadness, happiness, optimism, pessimism,.., etc.

According to the granularity level at which the sentiment is captured, SA can be addressed as a classification problem that belongs to one of these categories:

- Binary classification: also known as binomial, is about classifying the input text instances into one of two distinct classes or polarities. Binary sentiment classification models are useful for applications which cares about the satisfied (positive opinions) and dissatisfied (negative opinions) users (Liu (2012)).

- Multi-class classification: or multinomial correlates the input text with a single predicted polarity selected from three or more distinct polarity classes (Chen et al. (2015)). This type of classification has been used in several SA studies (Agarwal et al. (2011); El-Makky et al. (2014); Duwairi et al. (2014)) in addition to ratings-related sentiment applications such as movies reviewing systems; where numerical/star ratings is usually transformed into three or more polarity classes (Cherif et al. (2015)).

- Multi-label classification: unlike single-label classification (binary/multi-class), multi-label classification associates each instance with a set of labels at the same time (Zhang and Zhou (2014)). It is used for fine-grained SA which seeks to mine a set of human emotions included in a piece of text (Yang et al. (2014); Liu and Chen (2015); Li et al. (2016); Yu et al. (2018)).

On the other hand, regardless of the linguistic/granularity aspect, sentiment prediction is conducted using one of the following frameworks:

- Machine learning (ML): following the standard scheme adopted by these methods, SA using ML requires providing two fundamental sets: training set and test set. The first set combined data sentimentally-annotated by humans; it is used to extract useful hand-crafted or embedding features (see section Section2.4.2). The obtained features will be employed later to train a statistical classifier how to predict the text polarity of the unseen data in the second set (Biltawi et al., 2016). The learning process is carried out by inferring that a combination of a sentence's specific features yields a specific polarity class: positive, negative or neutral (Shoukry and Rafea (2012b)). With the training data being fed with sentiment labels included, ML-based SA classifiers learn the features via a supervised learning strategy (Turney (2002); Chesley et al. (2006); Boiy and Moens (2009); Tumasjan et al. (2010)). Several supervised learning classification algorithms such as Support Vector Machines (SVM), Naive Bayes (NB), Maximum Entropy, Decision Tree (DT),.., etc. were used in the state-of-the art either individually or as an ensemble (Refaee (2017)).

  Recently, using neural networks classifiers, SA has achieved good performances especially with the advent of text embedding features alongside deep neural architectures (LeCun et al. (2015); Altowayan and Tao (2016); Al-Sallab et al. (2017)). In this context, it is worth mentioning that for ML classifiers trained with text embedding features, unannotated (unlabeled) training data could be used (Mikolov et al. (2013); Le and Mikolov (2014)). This eliminates the human interference as unsupervised learning strategy is adopted to learn the embedding features. Nevertheless, while this might be useful for text generation, summarization or semantic similarity applications, it has been claimed that learning the embedding features from labeled data have led to better performances in the SA task (Tang et al. (2014); Iyyer et al. (2015)).

- Lexicon-based methods: adopt the unsupervised learning strategy as neither labeled data nor training step are required to design the sentiment classifier (Taboada et al. (2011)). The polarity of a sentence is determined using lexicon-derived sentiment scores assigned to its constituent words (Liu (2012)). A sentiment lexicon combines a list of subjective words and phrases along with their positive or negative sentiment scores (Piryani et al. (2017)). Sentiment lexicons can be general-purpose or domain-specific according to the compilation method. Sentiment lexicons are compiled via three strategies: manually with the assistance of a linguist and native speakers, automatically based on another dictionary (dictionary-based) or using the dataset itself (dataset-based) or semi-automatically where manual interference is needed to normalize the automatically-built lexicon (Liu (2012)).

  For each entry in the lexicon, a sentiment score is assigned using these weighting algorithms: (a) Straight forward sum (SFS): adopts the constant or uniform weight scheme to assign weights to the lexicon's entries such that the score of negative words is -1 while positive ones scored as 1. The polarity of a given text is thus calculated by accumulating the weights of negative and positive terms and the total polarity is determined by the sign of the resulted value (Abdulla et al. (2013)) or (b) Double polarity (DP): assigns both a positive and a negative weight for each term in the lexicon are which complementary to each other. Polarity is calculated by summing all the positive weights and all the negative weights in the input text. Then, the final polarity is determined according to the greater absolute value of the resulted sum (El-Makky et al. (2014)).

- Hybrid: these methods were proposed with the objective of exploiting the merits of the two previous methods. In these methods, lexicon-based models are employed to provide automatic annotation of the instances in the training set which wil be fed late to a ML sentiment classifier (El-Makky et al. (2014); Salameh et al. (2015a)). In addition, lexicon-derived features such as a term's polarity score are considered along with the syntactic/linguistic features providing an enriched features set Badaro et al. (2018).

With the variety of preprocessing tasks, feature types and semantic/sentiment resources, numerous SA models were proposed to mine the sentiment in English and Indo-European languages. In contrast, for a rich powerful language such as Arabic, NLP repository still suffers from the lack of resources and tools that support the formal Arabic let alone the under-represented Arabic dialects used widely on social media.

2.5. Evaluation Metrics

The performance of SA models is usually evaluated based on standard metrics adopted in the state-of-the-art. They are as follows:

1. **Precision:** indicates the ability of a classification model to identify only the relevant instances of a specific class category. It represents the ratio of the correctly predicted instances out of the whole number of predictions under a specific class category.

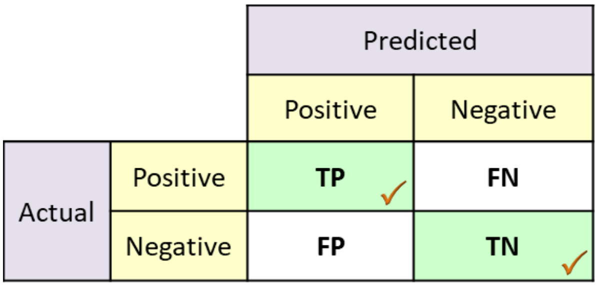
2. **Recall:** denotes the ability of a model to find all the relevant cases within a dataset. It represents the ratio of the correctly identified instances out of the total number of instances that are actually belong to a specific class category.

3. **F-measure:** provides a measure of the overall quality of a classification model as it combines the precision and recall through a weighted harmonic mean. For problems where classes are imbalanced, F-measure can be an efficient indicator of the performance of a classification model.

4. **Accuracy:** is the traditional way to measure the performance of a system. It represents the percentage of instances predicted correctly by the model for all class categories.

The previous metrics are obtained based on the confusion matrix shown in Figure (2.1). This matrix contains the statistics of the classifier predictions organized as:

- True Positive (TP): the instances correctly assigned to the given class.
- False Positive (FP): the instances incorrectly assigned to a certain class.
- False Negative (FN):the instances incorrectly not assigned to some class.



		Predicted	
		Positive	Negative
Actual	Positive	TP ✓	FN
	Negative	FP	TN ✓

Figure 2.1. The confusion matrix of a binary classification problem

- True Negative (TN): the instances correctly not assigned to a specific class.

Assuming that *m* denotes the number of classes under which the input instances should be classified, Precision, Recall, F-measure, and the average or macro for each of them besides accuracy (Acc.) are calculated as described below:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$MacroPrecision = \frac{1}{m} \sum_{i=1}^{m} Precision(i) \tag{5}$$

$$MacroRecall = \frac{1}{m} \sum_{i=1}^{m} Recall(i) \tag{6}$$

$$MacroF - measure = \frac{1}{m} \sum_{i=1}^{m} F - measure(i) \tag{7}$$

2.6. Conclusion

In this chapter, we have laid out background knowledge in sentiment analysis. We highlighted the importance of sentiment analysis and reviewed its applications in multiple vital domains. In addition, we presented the general pipeline adopted to solve the SA problem. This involved exploring the different preprocessing tasks, feature schemes and classification and assessment methods that are commonly used in the state-of-the-art. Moreover, we shed light on the challenges faced while handling the SA problem, where it has been observed that most of these challenges are highly related to the morphological and linguistic nature of the language to be mined for sentiment. Therefore, SA for languages having complex nature and morphology, such as the Arabic language, is considered much more challenging, especially with the lack of the supportive Arabic resources and NLP tools. More details about Arabic sentiment analysis and the relevant preprocessing tasks, features and sentiment classification models are provided in the next chapter.

3. ARABIC SENTIMENT ANALYSIS

In this chapter, we explore the Arabic sentiment analysis domain focusing on the specificity of the Arabic language and the challenges it poses towards sentiment analysis. This involves reviewing the major Arabic sentiment analysis models along with the recently-developed corpora, features, NLP tools and semantic resources.

3.1. Arabic Language

Arabic is a Semitic language, it is one of the six official languages of the United Nations and forms the first language of more than 422 million individuals worldwide (Council (2013)). The Arabic alphabet combines 28 letters including consonants and long vowels. Like other languages, Arabic has its own pronunciation, spelling, grammatical rules and idioms. Besides its written form from right to left, what makes Arabic unique compared to Indo-European languages, is its language variants which are: (a) Modern Standard Arabic (MSA) or formal Arabic which is used in books, news and formal speeches and (b) Dialectal Arabic (DA) also known as colloquial Arabic; it is the daily spoken language in the Arab countries and widely used on social networks. DA combines multiple dialects that vary according to the geographical region and include: Egyptian, Gulf (Saudi Arabian, Qatari, Omani, Yemeni, Bahraini, etc.), Levantine (Syrian, Jordanian, Palestinian, Lebanese, etc.) which form the Eastern Arabic dialects and Maghrebi (Libyan, Tunisian, Algerian, Moroccan) that can be referred to as Western Arabic dialects. These dialects exhibit drastic differences in terms of semantics, syntax and pronunciation (Huang (2015)).

Thanks to Web 4.0 technology, the Arabic language could find its way to be shared online with high growth ratios. Indeed, Arabic has achieved the largest proportional growth compared to other major linguistic groups with an increment of 150% during the period (2010-2015) (Northwestern (2016)). Moreover, the Arabic language has recently ranked 4th among the top ten languages used online (Sitsanis (2018)). As most of the Arabic textual content do exist on social media, it has been observed that the sudden boost in the size of the online Arabic content was essentially associated with the “Arab Spring” incidents started in Tunisia at the end of 2010 (Akaichi (2014)). Activists, demonstrators or even ordinary people have intensively used social networks such as Twitter and Facebook to express their feelings, opinions and impressions towards the ongoing events on the ground. On Twitter, for instance, the number of the Arabic tweets has increased into 2 million tweets per day compared to 30,000 in the middle of 2010 (Semiocast (2011)). Accordingly, within such conditions, the ability to harvest, analyze and mine opinions from the informal dialectal Arabic texts on social media becomes an interesting problem to investigate; especially with the multiple challenges it involves. To this end, many ASA research have been proposed; within which Arabic NLP tools and MSA/DA semantic and sentiment resources were developed.

3.2. Arabic Sentiment Analysis Challenges

Despite the recent growth of the public Arabic content across social networks and with the continuous development of Arabic NLP tools, ASA research still faces challenges, most of which are related to the Arabic language itself. With each form of DA having special linguistic and morphological features besides the varying syntactic and semantic properties (Badaro et al. (2018)), SA has to handle further issues beyond those already existing for textual data. Here, we highlight the major challenging issues encountered while conducting ASA:

- **Complex morphology**: being a Semitic language, Arabic adopts the root-and-pattern representation where a single set of consonants (usually three) called the “root” is used to derive a variety of related words. This is done by adding vowels (a,o,i) (ا، و، ي) or short vowels (diacritics) in addition to other consonants according to specific patterns Habash (2010). For example, from the root “ktb” “ك ت ب” that means “writing”, multiple words of different meanings and even different part of speech can be created such as: “كَتَبَ” (write, verb), “كَاتِب” (writer, noun), “مكتب” (desk, noun) and “كِتَاب” (book,noun). The inflectional morphology, however, is observed through the ability of the Arabic language to express a word in several grammatical categories while keeping the same meaning. The word's inflected forms can be obtained for several categories such as person, tense, voice (active/passive), number, gender, etc. Consequently, with such high derivational and inflectional morphology, handling Arabic texts through customizing current English SA systems and tools might be limited Habash (2010). Thus, special preprocessing tasks supported by Arabic-oriented morphological analyzers should be combined in ASA systems.

- **Lack of resources**: despite the abundant online Arabic content; there is a lack of Arabic sentiment corpora and sentiment lexicons. During the last decade, some datasets have been constructed either for MSA or DA, nevertheless, the number of sentiment datasets which are publicly available remains little (Badaro et al. (2018)). In addition, as the sentiment analysis accuracy depends on the size of the manipulated data, the limited size of the Arabic sentiment datasets makes it difficult to evaluate ASA systems against English SA models (Refaee and Rieser (2014)). On the other hand, the issues associated with the construction and annotation processes of sentiment lexicons have hindered the provision of large-scale and highly-coverage Arabic lexicons especially with the existence of different Arabic dialects.

- **Negation and sarcasm**: negation in Arabic is expressed by specific negation words that indicate the meaning "not"; some of them are: "ما", "لم" and "لا". Negation should be accurately detected and handled as it can convert the meaning of a sentence yielding a quite opposite polarity (Duwairi and El-Orfali (2014)). This task becomes more difficult and challenging when dealing with DA where negation words are so different from those in MSA and have several meanings such as "مو" which means "not" in the Levantine dialect and used for negation (e.g. السلطة مو تازة) (The salad is not fresh) or interrogative (e.g. تجي بوكرا، مو) (you're coming tomorrow, aren't you?). Such instances might mislead the sentiment classifier and degraded the classification performance. Another ambiguity faced by ASA models is the sarcasm issue in which the explicit polarity totally opposites the meant sentiment as in (e.g. نفدت كل التذاكر، كم انا محظوظ ,بعد الانتظار لساعات) which corresponds to (After waiting for hours, all tickets were sold; Lucky me), where "محظوظ", which means "lucky", indicates a positive sentiment while in the example it actually refers to the opposite.

- **Named entity recognition**: Named entity recognition (NER) plays a key role in aspect-level sentiment analysis as it refers to the target (person, location or organization) towards which the attitude is expressed Liu (2012). Compared to Indo-European languages, Arabic text has no notion of capital letters, therefore, any Arabic Named Entity Recognition (NER) system cannot use capitalization as a feature to recognize Named Entities (NEs) Mulki et al. (2018). Moreover, some NEs in Arabic can be used as adjectives having a specific polarity such as the person name “سعيد” which also means “happy”.

- **Arabizi usage**: Arabizi is considered a newly-emerged Arabic variant written using the Arabic numeral system and Roman script characters (darwish 2013). It is commonly used while expressing DA across social media, SMS and chat applications Duwairi et al. (2014). Arabizi poses a significant challenge to sentiment analysis especially when it is mentioned along with Arabic (e.g. 3an jad كتير الفلم 7elou) which means (The film is really amazing). Hence, proper tools are required to interpret Arabizi into either MSA or DA before conducting the sentiment classification task.

- **Dialects Variances**: DA forms the majority of the online opinionated Arabic content as it is commonly used across social media platforms. DA combines various dialects which differ according to the geographical location. Each dialect has its own vocabulary, syntactic and grammatical rules in addition to special idioms. On the other hand, despite that all dialects are derived from MSA and share some vocabulary, yet, common words or expressions among two dialects might have drastically different sentiments (Mulki et al. (2017)). For example, “يعطيك العافية” is a compliment of a positive sentiment that means “May God grant you health” in the Levantine dialect, while this very same phrase has an aggressive meaning of “Burn in hell” in the Tunisian dialect. Considering these variances, an ASA system that targets one dialect might not be efficient for another as it is developed with a dialect-dependent tools such as the morphological analyzer, stopwords/negation words and sentiment lexicons.

3.3. Arabic Sentiment Analysis background

During the last decade, social media has been the most rich resource of Arabic opinionated content. As opinions on social networks are usually shared in the form of sentences (tweets/comments) or documents (reviews), therefore, document and sentence-level SA studies have formed the majority of the recent ASA research. In line with the thesis scope, we will consider the research studies proposed for document and sentence-level SA. ASA models at the sentence/document level are implemented using machine learning approaches: supervised and deep learning, lexicon-based approaches or a combination of both known as hybrid. A detailed review of the state-of-the-art under the aforementioned method categories is provided in the following subsections.

3.3.1. Supervised Approaches

Supervised approaches require a labeled dataset from which the classifier can learn how to recognize the sentiment according to certain features (Liu (2012)). Hand-crafted and embedding features are both employed in supervised ASA models along with various classification algorithms. Research works that adopted supervised approaches were concerned about which preprocessing tasks, features or classification algorithms can lead to a better classification performance either for MSA or DA. A summary of the supervised ASA models proposed in the reviewed research is listed in Table 3.1 where Best, acc and F1 indicate the best-performing method, the scored accuracy and F-measure, respectively.

Considering the wide spread of the Egyptian dialect across social networks, enriching the Arabic sentiment resources with a pure Egyptian sentiment dataset along with Egyptian-specific preprocessing tools was the aim of Shoukry and Rafea (2012b). They collected a dataset of 1,000 positive/negative Egyptian tweets to test their SA model. The preprocessing included removing usernames, hashtags, URLs and non-Arabic letters. In addition, a list of Egyptian stopwords was constructed to enable an efficient stopwords removal. Unigrams and bigrams features were extracted using Term Frequency weighting (TF). The sentiment was, then, recognized using SVM and NB algorithms first with stopwords kept, then, with stopwords omitted. Results revealed that SVM performed better in both experiments achieving a best accuracy of 72% compared to 65% scored by NB.

The impact of combining emoji among SA features was investigated in (Al-Osaimi and Badruddin (2014)). The study introduced a SA model to be applied on a DA dataset composed of 3,000 positive, negative and neutral tweets. Hand-crafted features reduced by TF-IDF were fed into NB and KNN algorithms. The results showed that preserving emoji enhanced the performance of the model as the best accuracy achieved by NB classifier increased from 58.28% to 63.79%.

The recently-emerged form of Arabic (Arabizi) was investigated in (Duwairi et al. (2014)). The study seeked to convert the dialectal and Arabizi content into MSA. A dataset of 1,000 positive/negative/neutral tweets written in Jordanian and Arabizi was collected. For preprocessing, stemming, tokenization, stopwords filtering tasks were applied in addition to the conversion of Jordanian and Arabizi to MSA. Morphological features, negations and emoji were also included in the features set. The authors observed that, if stemming and stopwords removal are disabled, better performance can be achieved, while negation detection and conversion from Arabizi to MSA did not achieve a remarkable improvement in the evaluation measures. KNN, SVM and NB classifiers were used, where NB was the best with an accuracy of 76.78%.

Given the complex nature of the Arabic language, syntactic information was considered useful within a subjectivity classification context. This has been studied in (Abdul-Mageed (2015)) where it was claimed that using specific tokens would favorably impact the subjectivity classification performance. The proposed model was trained with words having certain POS tags such as adjective (ADJ), adverb (ADV) and proper noun (NOUN_PROP). The experiments were conducted with SVM and NB classifiers trained via Instance-based learning strategy on the Penn Arabic Treebank dataset (Popescu and Etzioni (2007)). Two features setting types were adopted: frequency and presence vectors. In all experiments, the preprocessing step was essential as the study highlighted that the rich morphology of MSA imposes using the compressed form of words in order to obtain a better model generalization. The obtained results emphasized the positive impact of using certain tokens rather than all the words for training; moreover, similar to the SA task, SVM was found of the best performance for subjectivity classification, compared to NB where it scored a high accuracy equals to 85%.

In (Salamah and Elkhlifi (2016)), an under-represented dialect was tackled; where a dataset of 340,000 positive/negative Kuwaiti tweets was mined for sentiment. Tweet-related features and opinions-oriented ones were extracted. The opinion-oriented features were obtained from 22 manually-built classes that combine emotions-bearing words. SVM, J48, Random Tree (RT) and DT classifiers were used. SVM scored the best results with an F1-score of 71.5% against 42%, 48% and 51% achieved by J48, DT and RT, respectively.

Table 3.1. Summary of Supervised ASA research works

Paper	Algorithm/features	Dataset	Evaluation
Shoukry and Rafea (2012b)	SVM, NB unigrams+bigrams	tweets Egyptian pos/neg	Best: SVM acc=72%

Al-Osaimi and Badruddin (2014)	NB, KNN
	TF-IDF
	unigrams
		tweets
		multi-dialects
		pos/neg/neut
			Best: NB
			acc=58.28% (-emoji)
			acc=63.79% (+emoji)

Duwairi et al. (2014)	KNN, SVM, NB syntactic, negation emoji	tweets Jordanian/Arabizi pos/neg/neut	Best: NB acc=76.78%
Abdul-Mageed (2015)	SVM, NB, IB1 POSs tokens	sentences MSA subj/obj	Best: SVM acc=85%
Salamah and Elkhlifi (2016)	SVM, J48, RT, DT tweet-related emotion-bearing words	tweets Kuwaiti pos/neg	Best: SVM F1=71.5%
Oussous et al. (2018)	Ensemble of SVM, NB, ME BoW	tweets Moroccan pos/neg	Best: Ensemble F1=83.4%

As an attempt to enhance the sentiment classification performance, the authors in (Oussous et al. (2018)) investigated how an ensemble model would impact the SA of Moroccan dialect. Their model was developed based on SVM, NB and Maximum Entropy (ME) classifiers such that the output is combined using voting and stacking strategies. The model was evaluated using 2,000 Moroccan tweets annotated manually as positive or negative in addition to semeval-2017 dataset (Rosenthal et al. (2017)). Having the tweets preprocessed and BoW features generated, they were first fed into each of the studied classifiers, then to all of them combined in the ensemble model. The results indicated that, for the Moroccan dataset, the ensemble model could slightly improve the classification performance achieving an F-measure of 83.4% against 82.6% scored by the individual classifier SVM.

3.3.2. Deep Learning Approaches

Deep Learning approaches combine multi-layer neural networks implemented using various architectures and trained with text embedding features such as word/document embeddings. In these approaches, the embedding features are learned automatically from the data through either an unsupervised or supervised manner (Gomez et al. (2017)). Many ASA research studies have recently developed deep learning models through which Arabic embedding features were provided to target MSA and some Arabic dialects (Al-Rfou et al. (2013); Soliman et al. (2017)). A summary of the deep learning ASA models proposed in the reviewed research is listed in Table 3.2 where Best, acc and F1 indicate the best-performing method, the scored accuracy and F-measure, respectively.

The variety of deep learning architectures has evoked the question about which architecture can perform better for ASA analysis. Therefore, Al Sallab et al. (2015) explored four deep learning models of different architectures and compared their performances in an ASA task. The first three models are: Deep Neural Network (DNN), Deep Belief Network (DBN) and Deep Auto Encoders (DAE). While DNN model employs the back propagation in a conventional neural network with several layers, DBN avoids overfitting through a pretraining phase before feeding a discriminative fine tuning step; whereas DAE provides a compact representation of the input sentence with a reduced dimensionality. These models were trained using the ordinary BoW features along with lexicon features derived from ArSenL lexicon Badaro et al. (2014). As for the fourth model known as Recursive Auto Encoder (RAE); it was suggested to address the lack of context handling procedures issue found in the previous three models. RAE can parse raw sentence words in the best order for which the error of recreating the same sentence words in the same order is as minimum as possible. This is done via a recursive parse tree where the sentence words are parsed recursively till finding the best order of the words. The evaluation was performed using Linguistic Data Consortium Arabic Tree Bank (Li et al. (2013)). When comparing the performances of the four models in positive/negative sentiment classification against an SVM model with hand-crafted features, it was noted that the performance of DNN, DBN and DAE was close to SVM's, while DAE provided a better representation for the input sparse sentence vector. The

RAE model outperformed all the other models achieving an accuracy of 74.3% and F-measure of 73.5%, compared to an accuracy of 45.2% and F1-score of 44.1% scored by linear SVM. This indicates the privilege of recursive models compared to one-shot models in terms of learning accurate semantic representations.

Aiming to enrich the Arabic resources with pretrained embeddings, Altowayan and Tao (2016) introduced Arabic word embeddings generated from an Arabic dataset of 190 million words using Continuous Bag of Words (CBOW) algorithm (Mikolov et al. (2013)). The authors indicated that their embeddings could handle dialects efficiently as different writing shapes of similar DA words were mapped close to each other in the embeddings space. To perform subjectivity and SA, the produced embeddings were used to train several binary classifiers. A combination of twitter datasets: ASTD (Nabil et al. (2015)), ArTwitter (Abdulla et al. (2013)) and QCRI (Mourad and Darwish (2013)) in addition to other two datasets representing book reviews: LABR (Aly and Atiya (2013)) and MSA news articles obtained from (Banea et al. (2010)) were used. It has been noted that the performance of the proposed model was slightly better than that of (Mourad and Darwish (2013)) in subjectivity classification, while for the polarity classification of twitter datasets, the best-performing classifier was Nu-SVM with an accuracy of 80.2% and an F-measure of 79.6%.

Another pretrained word embeddings with an improved compositionality were proposed in (Dahou et al. (2016)) within a CNN-based deep learning SA model. The model was trained with word embeddings learned from a dataset of 3.4 billion Arabic words using CBOW and Skip-Gram (SG) (Mikolov et al. (2013)). Inspired by Kim (2014), a CNN-based neural model with one non-static channel and one convolutional layer was developed. Multiple filter window sizes were adopted to perform the convolutional operation while a max-overtime pooling layer was utilized to capture the most relevant global features (Collobert et al. (2011)). The model was applied on several datasets such as ASTD (Nabil et al. (2015)) and ArTwitter (Abdulla et al. (2013)). Results revealed that the performance of the presented model mostly outperformed all the state-of-the-art systems where the best accuracy achieved for ArTwitter was 85.0%.

The idea of exploiting Arabic pretrained word embeddings in a deep neural SA model was investigated by Gridach et al. (2017). The authors used word embeddings previously trained with MSA/Egyptian corpora using Glove, SG and CBOW algorithms (Zahran et al. (2015)). These embeddings were used to initialize the input word embedding features that will be employed to train the proposed CNN-ASAWR model. CNN-ASAWR was developed as a variant of Collobert et al. (2011) system and customized to conduct SA on two MSA/dialectal datasets: ASTD (Nabil et al. (2015)) and SemEval-2017 (Rosenthal et al. (2017)). The results showed that using pretrained word embeddings led to better evaluation measures compared to the baseline systems. In ASTD dataset, for instance, the best F-measure scored by CNN-ASAWR was 72.14% compared to 62.60% achieved by Nabil et al. (2015) while for SemEval-2017 collection, an F-measure of 63% is achieved against 61% scored by the system in (El-Beltagy et al. (2017)).

Similarly, Alwehaibi and Roy (2018) compared the impact of involving different pretrained word embeddings in SA of MSA/DA tweets. Using an LSTM model, the authors initialized the embeddings generated at the embeddings layer with three publicly available Arabic pretrained word embeddings including Aravec (Soliman et al. (2017)), Arabic FastText (Joulin et al. (2017)) and word embeddings from (Altowayan and Tao (2016)). The model was applied to recognize positive, negative and neutral tweets in AraSenTi dataset (Al-Twairesh et al. (2017)). The experimental study concluded that among the used word embeddings, those from (Altowayan and Tao (2016)) achieved the best performance with an F-measure of 43% compared to 40% and 41% scored by Aravec and Arabic FastText, respectively.

With the lack of lexical and semantic resources for under-represented Arabic dialects, text embeddings represent an alternative expressive features. To prove that, the authors in (Mdhaffar et al. (2017)) investigated representing Tunisian comments by document embedding features within a Tunisian SA model. Their model was evaluated using a combination of publicly available MSA/multi-dialectal datasets: OCA (Rushdi-Saleh et al. (2011)), LABR (Aly and Atiya (2013)) and a manually annotated Tunisian Sentiment Analysis dataset (TSAC) obtained from Facebook comments about popular TV shows. Doc2vec algorithm by Le and Mikolov (2014) was applied to generate document vectors of each comment. The produced document embeddings were then used to train SVM, Bernoulli NB (BNB) and Multilayer Perceptron (MLP) classifiers using several combinations of MSA, dialects and Tunisian training sets. The best results were scored by MLP classifier when TSAC dataset was solely used for training where it achieved an accuracy of 78% and an

F-measure of 78%.

As each DL model has specific merits, usually related to its building unit, the authors in (Baniata and Park (2016)) investigated the impact of using a combination of CNN and Bidirectional-Long Short Term Memory (BiLSTM) on SA of MSA/dialectal tweets. They relied on the fact that the phrase representation of every sentence captured by CNN can be further enhanced by using BiLSTM network which can capture the contextual information and thus yields an improved performance. Two configurations were examined: CNN-BiLSTM, which involves composing the sentence representations to be improved later by the context information derived from both direction, and BiLSTM-CNN, where contextual information is first captured then fed to CNN to assist in generating the sentence representation. The used CNN model contained layers of filter sizes 3, 4 and 5 with the activation function ReLu used in both configurations. The model ensembles were evaluated using LABR dataset (Aly and Atiya (2013)). The data was subjected to normalization and the vocabulary size was reduced by keeping words of frequency greater than 10. Word embeddings were, then, obtained based on pre-trained word vectors by Al-Rfou et al. (2013). It was noted that CNN-BiLSTM architecture achieved an accuracy of 86.43%, whereas BiLSTM–CNN architecture has suffered from of overfitting after the fifth epoch yielding an accuracy of 66.26%.

In the same context, (Al-Azani and El-Alfy (2017)) examined various DL configurations for SA of MSA/DA tweets. The models were built using either a separate/stacked units of CNN, LSTM or by cascading CNN and LSTM within a single model. For training, the authors used word embeddings provided by word2vec algorithms: CBOW and skip-gram with static/non-static word initialization enabled. To evaluate the model variants, ASTD dataset from (Nabil et al. (2015)) and ArTwitter Abdulla et al. (2013) were used. The study indicated that updating the word embeddings during learning achieved the best results in most model variants. In addition, while LSTM outperformed CNN in general, combined LSTMs architecture was the best-performing model with an accuracy of 87.2% for ArTwitter.

Arabic lexical sparsity and ambiguity usually limits the ability of deep learning models to generalize and causes over-fitting. Al-Sallab et al. (2017) have addressed this issue in an RAE-based model through developing a Recursive Deep Learning Model for Opinion Mining in Arabic (AROMA). To enable modeling the semantic interactions at the morpheme level and to reduce the lexical sparsity

Table 3.2. Summary of Deep Learning-based ASA research works

Paper	Embedding	Dataset	Classifier	Evaluation
Al Sallab et al. (2015)	Recursive parsing tree	LDC-ATB MSA pos/neg	DNN, DBN DAE, RAE Linear-SVM	Best: RAE acc=74.3%

Altowayan and Tao (2016)	word2vec
	(CBOW)
		LDC-ATB
		ASTD, ArTwitter
		QCRI, LABR
		MPQA
		MSA/dialects
		pos/neg
			LR, SGD
			GNB, RF
			Linear-SVM
			Nu-SVM
				Best (MSA):
				Linear-SVM
				acc=77.87%
				Best (DA):
				LR
				acc=81.88%

Dahou et al. (2016)	word2vec
	SG, CBoW
		ASTD, ArTwitter
		MSA/dialects
		pos/neg
			CNN	Best: CNN
				acc=85.0%
				for ArTwitter

Baniata and Park (2016)	word2vec pretrained word embeddings	LABR MSA/Dialects pos/neg	CNN-BiLSTM BiLSTM-CNN	Best: CNN-BiLSTM acc=86.43%

Gridach et al. (2017)	word2vec
	(SG,CBOW)
	Glove
		ASTD
		SemEval-2017
		MSA/dialects
		pos/neg/neut
			CNN	F-score=72.14%
				(ASTD)
				F-score=61%
				(SemEval-2017)

Mdhaffar et al. (2017)	Doc2vec	OCA
		LABR, TSAC
		Tunisian/dialects
		pos/neg
			SVM
			MLP
			BNB
				Best: MLP
				F1=78%
				recall=78%

Al-Sallab et al. (2017)	Recursive syntactic parsing tree	Tweets,QALB ATB MSA/dialects pos/neg	AROMA RAE DNN, DBN DAE-DBN NB, Linear-SVM	Best: AROMA acc=86.5%

Al-Azani and El-Alfy (2017)	word2vec
	(SG,CBOW)
	static/non-static
		ASTD, ArTwitter
		MSA/DA
		pos/neg
			CNN, LSTM
			separate/stacked
				Best:
				combined LSTM
				F1=87.2%
				for ArTwitter

Alwehaibi and Roy (2018)	pretrained word embeddings Aravec, ArNews Arabic FastText	AraSenTi MSA/DA pos/neg/neut	LSTM	Best: LSTM+ArNews acc=43%

and ambiguity, the training data was subjected to morphological tokenization using MADAMIRA (Pasha et al. (2014)) before being fed to AROMA. In addition, semantic and sentiment embeddings were used to provide improved word representations. Moreover, instead of using the greedy algorithm to define the order of the model's recursion, AROMA employed phrase structures to automatically generate syntactic parse trees by which a better modeling of composition was achieved. The presented model was evaluated using three datasets of binary positive/negative polarities: an MSA dataset from (Abdul-Mageed and Diab (2011)) called ATB, DA Twitter dataset (Refaee and Rieser (2014)), an MSA/DA comments referred to as QALB (Farra et al. (2015)). The experiments involved using different combinations of the contributions augmented to the standard RAE. The results indicated that compared to the standard RAE, AROMA with all the contributions combined could improve the classification accuracy significantly by 12.2%, 8.4% and 7.2% for the ATB, QALB and Twitter datasets, respectively. Moreover, AROMA was evaluated against several ML and DL models where it overcome all of them scoring an accuracy increment of 7.3%, 1.7% and 7.6% for the same previous datasets, respectively.

3.3.3. Lexicon-based Approaches

Lexicon-based approaches adopt the unsupervised learning strategy saving the efforts of providing labeled training Arabic data. To assist deducing the Arabic sentiment, several Arabic sentiment lexicons were compiled to provide lexicon-derived features. A summary of the lexicon-based ASA models proposed in the reviewed research is listed in Table 3.3 where Best, acc and F1 indicate the best-performing method, the scored accuracy and F-measure, respectively.

Lexicon features represent the semantic scores associated with the lexicon entries; therefore, using a proper weighting method to assign polarity scores can impact the obtained sentiment classification performance. An attempt to to develop a novel weighting algorithm was presented in (El-Beltagy and Ali (2013)); where the authors noticed that sentiment terms often appear with other terms having the same polarity. Based on this theory, they constructed a dataset-based lexicon. Using the complied lexicon, SFS and DP methods were adopted to determine the positive, negative and neutral sentiment of the input text. In addition, uniform weighting scheme with negation switch policy, intensification words weighting and person names removal were applied prior to sentiment classification. Two manually-collected and annotated Egyptian datasets were used. The first one, called Dostour, combines 100 comments, while the second represents a Twitter dataset of 500 tweets. The best performance was achieved by the complementary weights strategy alongside DP method with an accuracy of 83.3% scored for Twitter dataset.

Aiming to evaluate manually-built against the automatically-built lexicons for the SA task, Abdulla et al. (2014) classified the sentiment of MSA/DA using three lexicon variants built via manual, semi-automatic and automatic construction methods. A forth integrated lexicon resulted from merging the three previous lexicons was utilized for the final system evaluation. Two datasets were used in the experiments, the first contains 2,400 positive/negative comments from Maktoob collected by Al-Kabi et al. (2013), while the second combines 2,000 positive, negative and neutral tweets (Abdulla et al. (2013)). The input data were subjected to normalization and light stemming. Sentiment classification was, then, performed using the four lexicons one by one, with SFS method and switch negation policy applied. Experiments showed that the stemming degraded the performance with manually-built and dictionary-based lexicons. In contrast, the accuracy was improved when dataset-based lexicon was used. The integrated lexicon with stemming applied could achieve the best performance for Maktoob dataset where it scored an accuracy of 74.6% compared to 70.2% with non-stemming option.

In (Duwairi et al. (2015)), the authors claimed that when dealing with MSA data, the likelihood of finding a stem in the sentiment lexicon is higher than that of finding the original word. This has been studied using an MSA sentiment lexicon constructed manually using seed words from SentiStrength (Thelwall et al. (2010)). Sakhr dictionary (Reyes and Rosso (2014)) was then employed to generate the synonyms. To evaluate the model, a dataset composed of 4,400 positive/negative tweets was used. The preprocessing phase included stopwords removal while retaining negation words in addition to stemming using MSA Khoja stemmer (Khoja and Garside (1999)). To examine the stemming impact, experiments were conducted with/without stemming while SFS method equipped with switch negation policy was employed to determine the sentiment. The results revealed that stemming has improved the classification performance where the accuracy improved from 23% to

46%, while F-measure increased from 31.3% to 55.51%.

Table 3.3. Summary of Lexicon-based ASA research works

Paper	Scoring method	Lexicon/Features	Dataset	Evaluation
El-Beltagy and Ali (2013)	SFS, DP	Egyptian size:4,392 unigrams	1: comments 2: tweets Egyptian pos/neg/neut	Best: DP 1: acc=83.3% 2: acc=63%
Abdulla et al. (2014)	SFS	MSA/dialectal size: 19,800 unigrams	1: comments 2: tweets MSA/dialectal pos/neg/neut	+stemming 1: acc=74.6% 2: acc=70.2%
Duwairi et al. (2015)	SFS	MSA size: 2,376 unigrams	tweets MSA pos/neg	+stemming F1 =55.51%
Assiri et al. (2017)	WLBA, SFS DP	Saudi/dialects size:14,000 lexicon term length, negation and supplication	1: tweets Saudi pos/neg 2: tweets Egyptian pos/neg/neut	Best: WLBA 1: acc=81% 2: acc=76%

Unlike the aforementioned methods, which employed pre-weighted lexicons to determine the sentiment score, Assiri et al. (2017) developed a polarity weighting method called WLBA. This method considers the context of the polarity-bearing words by exploring and counting how frequently a pair of (polarity, non-polarity) words co-occurs. Later, it assigns a weight to the polarity word due to the count of its associations with the non-polarity word in the whole dataset. A Saudi lexicon was built using dataset-based and dictionary-based approaches. Upon applying the model on Egyptian tweets from (El-Beltagy and Ali (2013)) and a Saudi dataset of 4,700 tweets, results showed that WLBA achieved poor results compared to SFS and DP for both datasets. This was due to ignoring complex structural and lexical specifications of the Saudi dataset. However, when features like negation and supplication were accurately handled via rule-based methods, WLBA outperformed other methods with an accuracy of 81%, compared to 72% and 43% scored by SFS and DP methods respectively. While, for the Egyptian dataset, the achieved accuracy was 76%, compared to 71% and 68% scored by SFS and DP method, respectively.

3.3.4. Hybrid Approaches

Hybrid approaches facilitate the annotation process of Arabic contents and enable a better expression of the Arabic sentiment through combining lexical/linguistic features with lexicon-derived features (Biltawi et al. (2016)). A summary of the hybrid ASA models proposed in the reviewed research is listed in Table 3.4 where Best, acc and F1 indicate the best-performing method, the scored accuracy and F-measure, respectively.

Within the proposed ASA hybrid models, novel combinations of features could be investigated as in (Abdul-Mageed et al. (2014)). In that study, it was aimed to seek seeking for the best scheme to represent lexical information within the context of SA. This was done by building an adjective sentiment lexicon of 3,982 entries to enrich the lexical features. The proposed model SAMAR composed of two classification stages: subjectivity and polarity classification. Four MSA/DA datasets of positive and negative reviews and tweets were collected manually. The used features included syntactic features, extracted via AMIRA morphological analyzer Diab (2009), in addition to an extra feature resulted from the matches between the input tokens and the adjectives of the built lexicon. Moreover, a novel feature that distinguishes MSA from DA was added. The experimental study showed that using SVM trained with the previous features could beat the baselines for most datasets either for subjectivity classification with an accuracy of 73%, or for sentiment classification with an accuracy of 70.30%.

Beyond using lexicons to obtain the sentiment scores, Alhumoud et al. (2015) introduced the idea of including the lexicon words in the training set of a hybrid sentiment classifier. The authors hypothesized that using the whole tweet for training, degrades the model's performance because non-sentimental words contained in each tweet may confuse the classifier. Therefore, aiming to improve the classification accuracy, 2,690 sentimental word tokens from 1,000 Saudi tweets merged with 1000 MSA single words, derived from the Arabic MPQA lexicon (M.ElArnaoty et al. (2012)), were used as a training set of SVM and KNN classifiers. Additional experiments were conducted using SVM and KNN as supervised models trained only with the Saudi tweets. SVM was the best-performing classifier in both supervised and hybrid model variants. Moreover, the hybrid SVM increased the accuracy

Table 3.4. Summary of Hybrid ASA research works

Paper	Features	Algorithm	Dataset	Evaluation
Abdul-Mageed et al. (2014)	Linguistic syntactic adjective polarity Adj-Lex	SVM several kernels	DAR, TGRD THR, MONT MSA/DA pos/neg/neut	Best: SVM linear kernel acc=70.3% (DAR)
Alhumoud et al. (2015)	unigrams scores from MSA/Saudi-Lex	SVM, KNN	1000 tweets Saudi pos/neg	Best: KNN acc=90.5%
Salameh et al. (2015a)	linguistic word N-grams Char N-grams score from translated-Lex	SVM NRC	tweets, comments MSA/DA pos/neg/neut	Best: SVM acc=85.23%
Baly et al. (2017)	linguistic syntactic tweet-related MSA/DA Lex	SVM	tweets MSA/DA pos/neg/neut	acc=43%
Al-Moslmi et al. (2017)	N-grams sentence-level syntactic score from ArabicSenti-Lex	SVM, NB, LLR, KNN NN	reviews DA pos/neg	Best: LLR, NN F1=97%

achieved by the supervised SVM by6.9%.

To compensate for the lack of publicly available Arabic resources, Salameh et al. (2015a) suggested using English NLP tools and lexical resources. Thus, the authors presented an ASA model that employs an English SA system with an English lexicon on a translated Arabic content. The evaluation datasets combine positive, negative and neutral tweets in addition to social media posts written in MSA/DA. Normalization, tokenization and POS tagging were appliead in the preprocessing phase. Before using the English SA model NRC-Canada (Mohammad et al. (2013)), it was modified to handle the Arabic text through employing a translated version of NRC Hashtag Sentiment Lexicon. On the other hand, the Arabic content was translated to English then targeted by the system of Kiritchenko et al. (2014). The best obtained accuracy value was 78.65% scored for the Syrian dataset.

In the same context, Baly et al. (2017) introduced a hybrid model OMAM whose features were inspired from the English SA model (Balikas and Amini (2016)). An equivalent set of surface, syntactic and semantic features were obtained with the assistance of MADAMIRA (Pasha et al. (2014)) and SAMA (maamouri2010) morphological analyzers. Additional features were provided by ArSenL Badaro et al.

(2014), AraSenti Al-Twairesh et al. (2016) and ADHL Mohammad et al. (2016) lexicons. The preprocessing phase included replacing emotions, URLs and hashtags with special tokens. The model was applied on DA tweets from (Rosenthal et al. (2017)). It has been noted that SVM classifier, trained with the previous features, achieved an F1 score of 42.2%, a recall of 43.8% and an accuracy of 43% ranking fifth at the official results of SemEval-2017.

With the key role of lexicon-derived features in improving the performance of hybrid SA systems, there was a crucial need for a large-scale, domain-independent, high-coverage and publicly-available Arabic lexicon. To meet that need, Al-Moslmi et al. (2017) introduced the Arabic senti-lexicon to assist in sentiment classification of multi-domain, DA reviews. The quality of the constructed lexicon towards SA task was assessed through training the model with five feature types, most of which were lexicon-derived. Features included sentiment words' polarity-based, sentiment words' presence-based, frequency POS-based, sentence level-based and other features related to words and sentences statistics. SVM, NB, LLR, KNN and neural network (NN) were employed. To evaluate the presented model, the authors created a dataset called Multi-domain Arabic Sentiment dataset (MASC) and combined 8,861 positive/negative DA customer reviews. Data was first preprocessed in terms of tokenization, normalization, stemming and stopwords removal. The model was, then, trained on each feature type solely, then on all of them combined in one set. Results indicated that, SVM achieved the best results when only POS-based features are included. However, when all features are used for training, LLR, NN and NB were of better performance where LLR and NN achieved an F1-score of roughly 97%, while NB achieved 96% compared to 82.07% and 77.97% F1-scores achieved by SVM and KNN respectively.

3.4. Background Limitations

When exploring the state-of-the-art of ASA, it could be observed that the proposed SA models have employed several feature types, preprocessing tasks, classification algorithms and implementation architectures. Each model has its own limitations. With respective to the used approach, the major limitations of the existing ASA models are as follows:

1. **Supervised models:** being relied on labeled training data, supervised ASA models requires providing sentimentally annotated Arabic corpora which is a difficult task especially for DA. In addition, with the complex nature and morphology of MSA and DA, generating features becomes a laborious-intensive task which involves using or developing MSA/DA specific morphological analyzers, semantic resources and NLP preprocessing techniques. On the other hand, the wide variety of hand-crafted features, adopted by these models, have yielded feature vectors of a high dimensionality and sparsity. This has increased the training time overhead and sometimes led to memory issues (Duwairi et al. (2014)). While many models applied weighting schemes to reduce the features size, it was not always guaranteed to retain sentiment indicative tokens among the features; since most weighting schemes decide the tokens to be kept, based on their frequency within the dataset. Hence, sentiment-bearing tokens of less frequencies might be excluded leading to less expressive feature vectors.

2. **Deep learning models:** the combination of text embedding features and deep neural networks was found effective to address the issues encountered in the ordinary machine learning models (Dahou et al. (2016); Al-Sallab et al. (2017)). Most of the proposed models adopted context-aware word/document embeddings (Altowayan and Tao (2016); Mdhaffar et al. (2017); Al-Azani and El-Alfy (2017)), ordred compositional embeddings (Al Sallab et al. (2015)) or pretrained word embeddings (Baniata and Park (2016); Gridach et al. (2017); Alwehaibi and Roy (2018)). Considering the free word order and the varying syntactic nature of DA, such syntax-aware, context-aware and ordered embedding types may not always succeed in capturing the sentiment, especially when analyzing social media data where informal Arabic dialects are dominant. Moreover, the available Arabic pretrained word embeddings are learned from MSA/Egyptian (Al-Rfou et al. (2013); Zahran et al. (2015); Soliman et al. (2017)) which can lead to Out-Of-Vocabulary (OOV) issues when used for dialectal-mixed contents or under-represented dialects such as Syrian, Tunisian and Moroccan (Alwehaibi and Roy (2018)). On the other hand, the efficient training of deep neural models requires large-sized Arabic training datasets, this is reflected on the consumed training time as it increases

by the size of the input data. In addition, the complex architectures of some deep neural networks introduces high computation overhead besides a large number of hyper parameters need to be tuned; which makes the developed models difficult to be trained and maintained (Iyyer et al. (2015); Shen et al. (2018)).

3. **Lexicon-based models:** Although these models avoid the training overhead and require no labeled input, their performance in SA was mostly outperformed by other model variants (Abdulla et al. (2013)). This could be due to the fact that, the scoring algorithms, adopted by lexicon-based models, identify the sentiment regardless of the contextual-related information and language subtleties such as sarcasm, negations, etc. (El-Beltagy and Ali (2013); Abdulla et al. (2014)). In addition, as some Arabic person names and adjectives that bears sentiment are identical, sentiment scores can be assigned to certain person names while computing the polarity of a sentence yielding miss-classified instances. Despite that many lexicon-based models eliminated person names from the dataset, other NEs such as locations or organizations which can be also confused with adjectives, were retained. Moreover, lexicon-based SA models cannot be generalized across multiple corpora of different domains, let alone different dialects, unless a very large-sized, multi-domain and multi-dialectal lexicon is built, which is considered a non trivial task (Al-Twairesh et al. (2016); Al-Moslmi et al. (2017)).

4. **Hybrid models:** These models have exploited the merits of supervised and lexicon-based approaches; However, they also inherited the drawbacks of both previous methods, represented in the laborious tasks of designing the features and building the lexicons.

Besides the aforementioned gaps, and regardless of the used SA approach, it is noted that within the presented ASA systems, MSA (Abdul-Mageed (2015); Al Sallab et al. (2015)), single dialects (Shoukry and Rafea (2012b); Duwairi et al. (2014); Assiri et al. (2017)) or a combination of MSA and major dialects (i.e. Egyptian, Gulf) (Baniata and Park (2016); Gridach et al. (2017)) were targeted. On the other hand, under-represented dialects such as Levantine and Maghrebi were remarkably less tackled (Salameh et al. (2015a); Mdhaffar et al. (2017); Oussous et al. (2018)). Moreover, although some studies constructed the features based on multi-dialectal corpora (Altowayan and Tao (2016); Dahou et al. (2016)), the models trained with these features have not been evaluated with both Western and Eastern Arabic dialects. This can be attributed to the difficulty of handling the wide syntactic and semantic variances among the Arabic dialects which, in turn, hindered the provision of universal ASA models.

3.5. Towards New Models For DA Sentiment Analysis

Within the scenario of social media SA, where DA is dominating the textual content, it is becoming crucial to provide a universal SA model that can be generalized across the different Arabic dialects without modifications. This requires an efficient handling of the challenging nature of DA along with addressing the issues resulting from the differences among the Arabic dialects. To this end, we opt to develop two SA models: Hand-Crafted features-based Tw-StAR (HCB Tw-StAR) and Embedding Features-based Neural Tw-StAR (Neu Tw-StAR). While both models supports the specificity of DA, each of which employs a different type of training features. In the following subsections, we briefly review the contributions introduced by each of the proposed SA models.

3.5.1. HCB Tw-StAR Model

Through this model, we aim to handle the variances among the Arabic dialects based on hand-crafted features. These features were formulated without the need for dialect-dependent NLP tools and with the least dependence on dialectal resources. HCB Tw-StAR was implemented using two sentiment classifiers with two learning strategies: Lexicon-based and supervised. The novelty introduced by HCB Tw-StAR can be briefed as follows:

- **Novel combinations of preprocessing techniques:** to generate efficient hand-crafted features for the supervised model. The obtained features had a reduced dimensionality while retaining the sentiment expressive tokens.
- **Named Entities (NEs) as universal sentiment indicatives:** being unified

across the Arabic dialects, and with the impact of the sentiment borne by an NE on the polarity of the text containing it, instead of ignoring or eliminating NEs, they were used to enrich the hand-crafted features of the supervised model. In addition, NEs were combined in the sentiment lexicon used by the lexicon-based model.

3.5.2. Neu Tw-StAR Model

For this model, we used embedding features to train a feed-forward neural network SA classifier. While this model can support various Arabic dialects, it does not rely on external knowledge resources or dialectal NLP tools/resources. Moreover, the training data are fed to the model without any type of manipulation or preprocessing. The main novel merits combined in Neu Tw-StAR are as follows:

- **Unorderd n-gram embedding features:** as efficient representations generated for a variety of Arabic dialects and used to train the neural model. Composing the n-gram embeddings by an unordered composition function enabled handling the free word order and the varying syntactic nature of the Arabic dialects. This involved focusing on the semantic regularities and ignoring the syntactic/contextual ones yielding improved embedding features to be generalized across the different Arabic dialects.

- **Shallow neural architecture:** is used to implement the neural model as it can reduce the time/computation overhead introduced by deep neural architectures. The shallow neural architecture of the proposed neural model, made it possible to avoid the high time/computation cost which is usually related to multi layer-level complicated calculations and numerous hyper parameters tuning. In addition, with the lack of publicly-available Arabic corpora and the difficulties of data collection and annotation, adopting a shallow architecture for the neural model enabled obtaining good results for small and medium sized Arabic datasets.

3.6. Conclusion

In this chapter, we explored the Arabic sentiment analysis research domain. First, we highlighted the specifications of the Arabic language and the challenges they introduce towards sentiment analysis. Then, while exploring the proposed ASA research works, we focused on the used preprocessing techniques, the constructed Arabic sentiment corpora and lexicons, the generated hand-crafted/embedding features and the adopted machine learning/lexicon-based classifcation methods. Consequently, at the end of this chapter (Section 3.4), it was possible to outline the limitations encountered in the presented ASA models in terms of the features and classification methods. Lastly, in section (Section 3.5), we introduced our novel ASA models along with the merits they provide to perform an efficient SA of DA contents. This sets the scene for the next chapter where we will review our first model HCB Tw-StAR and provide a practical evaluation of its effectiveness as a SA model of DA.

4. HCB TW-STAR MODEL

This chapter describes our hand-crafted features-based SA model called HCB Tw-StAR. Within the proposed model, we introduce named entities as sentiment indicatives and present a novel algorithm to exploit them in the sentiment analysis task. In addition, for a better handling of the DA contents, we employ various preprocessings tasks to obtain more expressive sentiment features. Finally, with the named entities included and the preprocessing tasks applied, the sentiment is recognized using supervised and lexicon-based classification algorithms.

4.1. HCB Tw-StAR Model Description

Our hand-crafted features-based Tw-StAR model (HCB Tw-StAR) was implemented according to the standard pipeline adopted in most SA models (Section 2.4). As it can be seen from Figure (4.1), the proposed HCB Tw-StAR model is composed of a preliminary step: NEs processing followed by three main phases: data preprocessing, features extraction and sentiment classification.

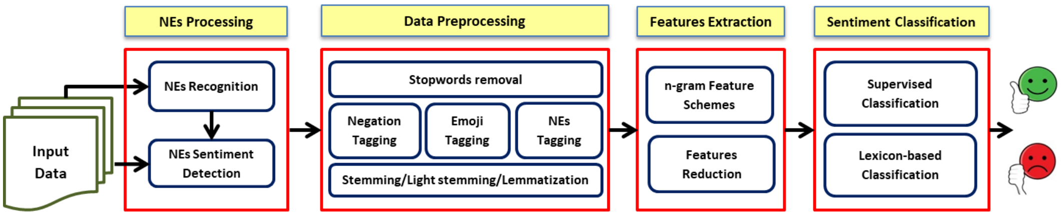


Figure 4.1. The general schema of HCB Tw-StAR sentiment analysis model.

In line with the motivation and goals of this thesis, HCB Tw-StAR was developed with the objective of addressing the dialect dependency and features high dimensionlity issues that have been mentioned in Section 3.4. To this end, we opted to enhance the preprocessing phase through adopting several combinations of preprocessing tasks conducted with a least dependence on dialect-specific NLP resources and without employing dialectal morphological analyzers. Moreover, we leveraged NEs as sentiment indicatives by integrating them among the used preprocessing techniques. This requires a preparation step in which NEs are, first, recognized then associated with a proper sentiment label before feeding them to the next preprocessing phase.

Having the input text manipulated in the preprocessing phase, it is directed to the features extraction phase where several schemes of n-gram features are generated. The produced features are, then, fed to the sentiment classification stage to be used either to train the supervised classifiers or to assist in looking for single and compound terms in the lexicons employed by the lexicon-based classifier. In the following sections, a detailed review of each phase of HCB Tw-StAR is provided.

4.2. Named Entities Processing for Sentiment Analysis

This phase is an initial or preparatory step that precedes the preprocessing phase in the proposed model. Unlike previous studies that ignored or eliminated NEs while conducting SA, we believe that NEs can be exploited as sentiment indicatives which improves the sentiment recognition quality. Our assumption is based on the fact that opinionated contents on social media are rich of NE types: locations, persons and organizations. These NEs are often correlated with major events, took place in a certain period of time, such as the names of candidates (person), political parties (organization) and cities (location) during elections or the names of players (person), sport clubs (organization) and stadiums (location) during some sport leagues. Therefore, each NE in a dataset, collected during a certain period of time, can bear an implicit sentiment defined by the attitudes towards this very NE during that period of time.

According to Yasavur et al. (2014), while investigating the SemEval 2007 trial dataset, it was indicated that 82% of the headlines contain at least one NE. In the same context, Jansen et al. (2009) reported that out of the entire population of the tweets they tackled, 19% included an organization or product brand in some way. Consequently, NEs in a sentence can be considered one of the essential components without which the subjectivity of the sentence might be lost as in Example 4.1 which includes two NEs denoting two political parties: "نداء تونس" (*Nidaa Tounes party*) and "حركة النهضة" (*Ennahdha Movement Party*):

Example 4.1 من لم ينتخب نداء تونس كأنه انتخب حركة النهضة

(Those who didn't vote for Nidaa Tounes Party as if they voted for Ennahdha Movement Party)

If we omit these NEs, the subjectivity of the sentence cannot be recognized; while with them retained, the sentence's polarity would not be inferred unless the sentiment borne by each NE is identified.

In addition, considering the biased nature of social media texts which varies over the time, it could be noted that the polarity of a sentence (tweet, comment, review), containing an NE and posted during a specific time period, is affected by this very NE and the attitudes towards it at that period of time. This, somehow, makes NEs sentiment indicative text components. To clarify that, when exploring the dataset collected by Sayadi et al. (2016) during the post-revolution Tunisian elections, we find that 80% of the tweets containing the person name "بن علي" (*Ben Ali*) who is the former Tunisian president, has a negative sentiment. Similarly, in the dataset used in (Altowayan and Tao (2016)), the location name "سوريا" (*Syria*) which is a country facing recent war incidents, was encountered in 30 tweets; 75% of them was negative.

Accordingly, given a Twitter/Facebook dataset collected in a certain period of time, we hypothesize that identifying the sentiment of an NE can contribute in inferring the polarity of the sentence in which it is mentioned. To do that, two successive procedures are needed: NEs recognition and NEs sentiment detection.

4.2.1. Named Entities Recognition

In order to involve NEs within our SA task, all NE types should be extracted from the input data. For this purpose, we use an Arabic Named Entity Recognition (NER) system developed by Gridach (2016). This model has proved its efficiency over the existing NER models, it can handle NEs encountered in DA contents with the ability to avoid Out-of-Vocabulary (OOV) issues. The building architecture of the employed NER model is based on deep neural networks, it combines a Bi-directional Long Short Term Memory (Bi-LSTM) with Conditional Random Fields (CRFs) building units. Through the model layers, NEs are learned and recognized as follows:

- For each word in the input sentence, a word vector is constructed as a concatenation of two vectors, the first is obtained from a lookup table containing the pretrained word embeddings while the second was created using character-level embeddings.

- In the subsequent LSTM layers i.e. Bi-LSTMs, the sentence is read in two directions yielding left (backward) and right (forward) word representations. These representations are concatenated and linearly projected onto the next layer.

- The last CRF layer is used on the top of the bidirectional LSTM in order to capture contextual features in the form of neighboring NER tags.

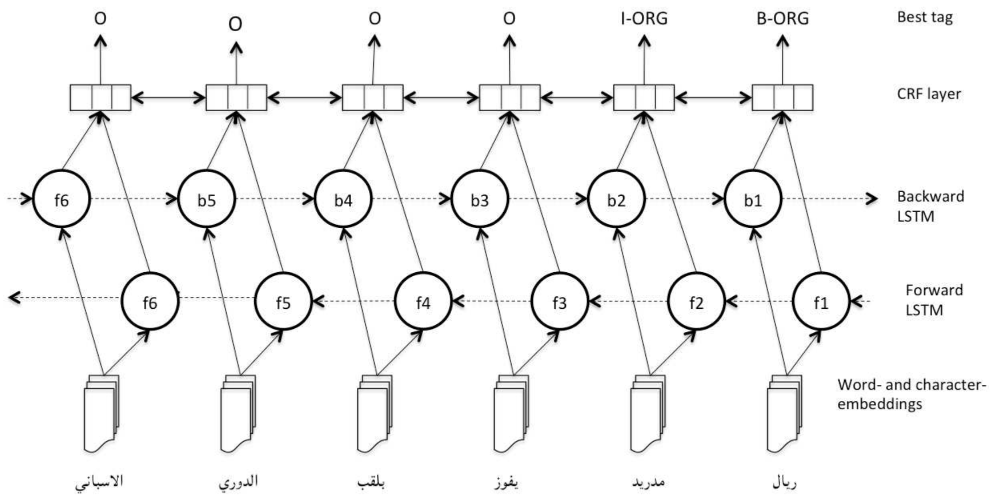


Figure 4.2. The architecture of the used Arabic NER system.

Figure (4.2) shows the architecture of NER system while recognizing the NEs: "ريال" (*Real*) and "مدريد" (*Madrid*) embedded in the sentence "ريال مدريد يفوز بالدوري الاسباني" (*Real Madrid wins the Spanish league title*).

4.2.2. Named Entities Sentiment Detection

To accomplish the NEs processing phase, each NE extracted in the previous step needs to be associated with the sentiment it bears. NEs sentiment recognition has not been tackled in previous studies as most of them focused on NEs recognition rather than exploiting them for further NLP tasks. Inspired by SFS method (El-Beltagy and Ali (2013)), we have developed an algorithm (see Algorithm 1) to detect the sentiment of the NEs obtained by Gridach (2016).

Algorithm 1: NEs sentiment detection Algorithm

```
Data: sentence tokens: $T$, Named Entity: $N$, sentence polarity: $pol\_t$,
      NE computed score: $N\_score$
Result: NE assigned polarity: $N\_pol$
$N\_score \leftarrow 0$
foreach N in NEs do
    foreach tweet in dataset do
        if $N \subset T$ then
            if pol_t=positive then
                increase $N\_score$ by 1
            else if pol_t=negative then
                decrease $N\_score$ by 1
    end
end
if $N\_score > 0$ then
    $N\_pol$ = $positive$
else if $N\_score < 0$ then
    $N\_pol$ = $negative$
```

The proposed algorithm aims to identify an NE as having a positive or a negative sentiment based on its local contextual information as follows:

- NEs extracted from a dataset are compared against the tokens of each sentence included in the training division of that dataset.

- When a match between a specific NE and a sentence is found, an aggregated score is assigned to this NE due to the polarity of that sentence such that 1 is added if the sentence's polarity is positive while 1 is subtracted if the sentence has a negative polarity whereas 0 is added when no match is found.

- The polarity of a certain NE is, thus, determined by the sign of its accumulated resulting score where positive and negative signed scores define positive and negative NEs, respectively.

- As for NEs of zero-valued scores, they are eliminated since they were mentioned equally in positive and negative input sentences.

According to our algorithm, the polarity of an NE in a dataset is defined by the majority of attitudes towards it. Thus, the sentiment of each NE in the dataset is identified as positive or negative according to how frequently this NE is mentioned within positive or negative sentences. This can address the confusion of detecting the sentiment of two NEs having contradict polarities and mentioned in the same tweet as in Example (4.2) where "هتلر" (*Hitler*) known as a dictator and the named entity "خالد بن الوليد" (*Khalid ibn Al-Waleed*), who was a noble commander, were mentioned together in a single positive sentence.

Example 4.2 إنه خالد بن الوليد القائد النبيل يا من يفتخر بهتلر"
(To those who boast of Hitler, it is Khalid ibn al-Walid, the noble leader)

In this case, the algorithm gives both NEs a positive score at the beginning, however, after browsing the rest of tweets the score of "هتلر" (*Hitler*) will decrease since it is mostly mentioned in negative contexts while the score related to "خالد بن الوليد" (*Khalid ibn Al-Waleed*) will increase if the majority of the tweets containing it was positive.

4.3. Data Preprocessing Tasks

Through this step, we aim to handle the noisy and complex nature of the input DA data and, thus, enable the next phase to generate efficient features with a reduced dimensionality. Beyond the single application of preprocessing or NLP tasks used in previuos studies (Duwairi and El-Orfali (2014); Brahimi et al. (2016); Ghadeer et al. (2017)), we evaluated the sentimentally-annotated NEs resulting from the previous phase along with novel combinations of preprocessing tasks for SA of DA. The proposed preprocessing combinations included the following NLP techniques:

- **Normalization:** was applied as an initial cleaning, where the input text is cleaned from the non-sentimental content such as punctuation, URLs, dates, digits and platform-inherited symbols like hashtags, retweet, mentions, etc. For instance, applying normalization on the sentence in Example 4.3 would produce the clean sentence in Example 4.4.

Example 4.3 ما يدور في رأس ترامب #احزر *http://t.co/w3kpo*

Example 4.4 ما يدور في رأس ترامب احزر

- **Stopwords removal (stop):** as stopwords have a high frequency of presence in the input Arabic texts and do not carry a significant semantic meaning by themselves, we opt to reduce them based on predefined lists containing MSA stopwords (KACST (2017)). To use the proposed model for non-Arabic SA, we have replaced the Arabic stopwords list with another lists that support English (Ganesan (2014); Porter and Boulton (2002b)), Spanish (Porter and Boulton (2002d)) and Turkish (FatihUniv (2010)).

- **Stemming (stem):** can address the inflectional nature of the Arabic language as it replaces a set of inflectional words or word variants with a single word representing the basic form (stem) or root. This contributes in reducing the features dimensionality and increasing the recall (Darwish and Magdy (2014)). With the absence of DA stemmers, and based on the lexical overlap between MSA and DA, we employed two MSA stemmers:

  1. Farasa stemmer: it is an MSA stemmer developed by Abdelali et al. (2016). It employs an SVM-based segmenter to rank the potential ways to segment words into prefixes, stems, and suffixes. This is performed based on a variety of features and lexicons from which probabilistic models of stems, prefixes, suffixes and their combinations are obtained (Khalifa et al. (2016)). For example, considering the word "وقلبي" (*and my heart*) which can be written as "wqalby", it is segmented into three clitics "w+qalb+y" (*and+heart+my*), namely the conjunction article "w" as a prefix, the stem "qalb", and the possessive pronoun "y" as a suffix. Stemming will, then, be conducted by eliminating the segmented affixes and keeping the stem.

  2. Information Science Research Institute's Arabic stemmer (ISRI): this stemmer was proposed in (Taghva et al. (2005)) to provide MSA stemming without the need for a root dictionary. Hence, ISRI can produce a normalized form for words whose root are not found and, thus, avoids obtaining invalid roots. Moreover, aiming to facilitate deducing the correct stem, ISRI was designed as a context-sensitive stemmer such that it conducts several normalization processes including hamza normalization, diacritics removal, connectors handling and removal before deducing the stems (Dahab et al. (2015)).

To support non-Arabic languages, the previous stemmers were replaced with porter2 (Porter and Boulton (2002a)) for English, snowball for Spanish (Porter and Boulton (2002c)) and Zemberek (Haddad and Ali (2014)) for Turkish.

- **Light stemming (LightS):** in order to retain the variety of words having the same root and different meanings, we employed a light stemmer called **light10** introduced in (Larkey et al. (2007)). Compared to other versions (Larkey et al. (2002)), light10 was the most robust and powerful. It was developed to chop off affixes that were commonly defined as prefixes or suffixes, but infrequently encountered at the beginning or ending of stems. This requires conducting multiple normalization techniques along with different numbers and depths of the suffixes to be stripped (Larkey et al. (2007)). For example, given the words: "الكاتبون" (*the writers*) and "الكتاب" (the book), while both words would have the same stem "كتب" (*to write*), light stemming maintains the differences between the two words as their light stems would be: "كاتب" (*writer*) and "كتاب" (*book*), respectively.

- **Lemmatization (Lem):** coping with the high derivational and inflectional morphology of the Arabic language, we further subjected the input words to lemmatization using Farasa MSA lemmatizer (Abdelali et al. (2016)). To produce the lemma of a word, Farasa removes the inflectional endings only, if exist, and returns the base or dictionary form of a word. For example, the lemma of the word "مكتبات" (*libraries*) is "مكتبة" (*a library*) which bears the same meaning and represents a correct Arabic word; while its stem would be "كتب" (*to write*) and its light stem would be "مكتب" (*a desk*).

  As the proposed model was extended to support non-Arabic languages, the previous lemmatizer was replaced with TreeTagger (Schmid (1999)) that can handle both English and Spanish.

- **Negation tagging (Neg):** negation words in Arabic can directly impact the polarity implied by the contexts containing them. Through our model, we opt to identify the negation words and replaced them with a unified distinctive tag "*NegWord*". To infer the negation in the input data we built our own list of MSA/DA negation words/terms that usually precede verbs, noun phrases or adjectives. For non-Arabic data, namely, Turkish, we replaced the content of the previous list with Turkish negation words. The used negation words along with their meaning for both Arabic and Turkish languages are listed in Table 4.1.

Table 4.1. Negation words for Arabic/Turkish datasets

Language/Dialect	Negation word	Meaning	Negation word	Meaning
MSA	لا	not	لستم	not
	ما	not	ليس	not
	لن	not	لست	not
	دون	without	لسن	not
	أبداً	never	ليسوا	not
	غير	without	بغير	without
	بلا	without	بدون	without
DA	ماكش	not	مانيش	not
	ماكمش	not	ماهوش	not
	ماهمش	not	مفماش	without
Turkish	hiç	never	sız	without
	asla	never	siz	without
	olmaz	no	suz	without
	değil	not	süz	without

- **Emoji tagging (Emo):** with the important role played by emoji to express the sentiment embedded in social media texts, we fixed a list of the most common emoji detected in the input text in order to be detected and replaced with specific textual tags. Emoji were recognized based on UTF-8 encoding, while the textual tags: "ايموشنموجب" and "ايموشنسالب" were used to imply the positive and negative emoji icons, respectively. For example, the positive emoji in "أداء الممثل كان رهيب☺" (*the performance of the actor was terrific* ☺) is replaced as follows: "أداء الممثل كان رهيب ايموشنموجب". Since our model was further employed for fine-grained SA, we modified the previous tags to represent different human emotions such as hate, angry, happy, sad, etc. with textual emoji tags such as "HappyEmoj", "SadEmoj" and so on. In addition, for non-Arabic input data, English textual tags were used.

- **NEs tagging (NE):** after NEs were recognized and their sentiment was detected in the first phase of our model (Section 4.2), they were looked up in the input datasets such that positive NEs were replaced with a positive tag "*PosNE*" while negative ones were replaced with a negative tag "*NegNE*".

Doing so, the vocabulary size was reduced as all the NEs are unified into one of two tags.

4.4. Features Extraction

Having the data manipulated by multiple single/combinations of preprocessing tasks, it was subjected to tokenization to generate n-grams features. Three n-grams schemes including unigrams, bigrams and trigrams and combinations of them were produced as they can capture information about the local word order (Joulin et al. (2017)). For a certain n-gram scheme, the feature vector of a sentence is constructed via examining the presence/absence of this scheme among the sentence's tokens. Thus, the resulting feature vectors are formulated as one-hot encoding vectors with the binary values "1" (presence) or "0" (absence). Term frequency (TF) weighting property, which measures how frequently a term is repeated in a document, was employed to reduce the features size according to predefined frequency thresholds.

4.5. Sentiment Classification

At this stage, the model predicts the proper polarity correlated with an input sentence based on the features produced in the previous phase. To do that, two classification approaches were used:

1. **Supervised classification:** with labeled data used as an input, various schemes of n-grams including unigrams, bigrams and trigrams and combinations of them were used to train:

   - SVM classifier: implemented using LIBSVM with the linear kernel (Chang and Lin (2011)). LIBSVM provides fast training and accurate classification along with ease of implementation (Chang and Lin (2011); Sun et al. (2012)).
   - NB classifier: we adopted the multinomial variant of NB algorithm provided in the Natural Language Tool Kit (NLTK) (Bird et al. (2009)). NB

from NLTK works as a rule-based classifier together with binary-valued features and was previously proved to be efficient for SA of DA (Itani et al. (2012)).

Figure (4.3) shows the schema of the supervised approach included in the proposed model HCB Tw-StAR.

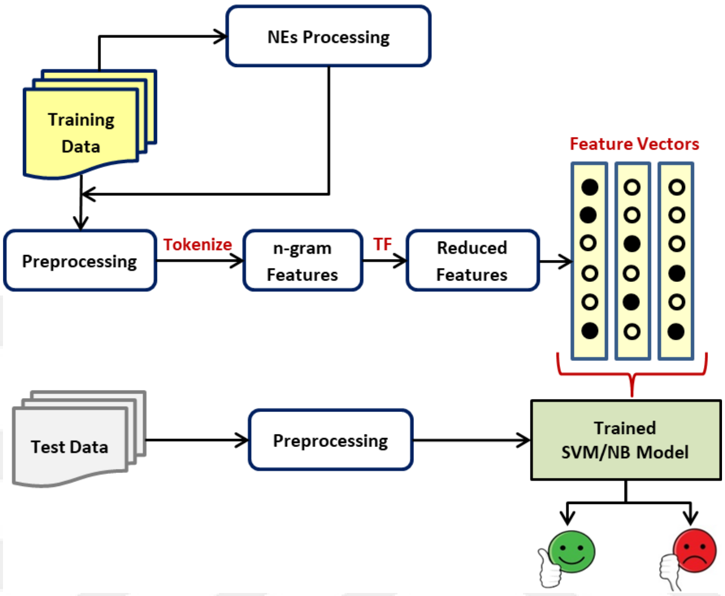


Figure 4.3. HCB Tw-StAR: Supervised sentiment analysis pipeline

2. **Lexicon-based classification:** uses an integrated lexicon constructed out of pre-built lexicons: MSA/Egyptian NileULex (El-Beltagy et al. (2016)), MSA/DA seeds from Arabic Emotion Lexicon (AEL) and Arabic Hashtag Lexicon seeds (AHL) (Salameh et al. (2015b); Mohammad et al. (2016)). To support the levantine, Gulf and Tunisian dialects we have manually constructed three lexicons for Levantine (LevLex), Gulf (GulfLex) and Tunisian (TunLex) dialects. In addition, as the model was extended to be applied on non-Arabic languages, we used a Turkish sentiment lexicon called SentiTurkNet obtained from (Dehkharghani et al. (2016)). Table 4.2 lists these lexicons and their sizes.

   The sentiment recognition task is conducted by subjecting the input data to preprocessing; then, the tokens of a sentence either unigrams or combinations of unigrams and bigrams are looked up in the proper lexicon. When a match is found, the sentence's polarity score is calculated using SFS algorithm with

Table 4.2. The used sentiment lexicons

Sentiment Lexicon	Positive	Negative	Size
NileULex	1,697	4,256	5,953
AEL	12	11	23
AHL	107	118	225
LevLex	258	559	817
GulfLex	33	67	100
TunLex	1,953	3,329	5,282
SentiTurkNet	1,437	1,970	3,407

the constant weight strategy applied. Thus, negative and positive words have the weight of -1 and 1, respectively.

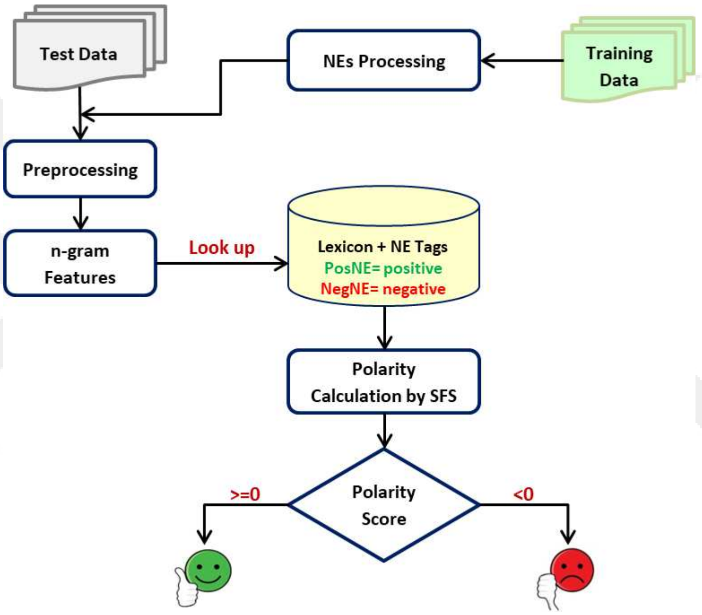


Figure 4.4. HCB Tw-StAR: Lexicon-based sentiment analysis pipeline

To enable considering NEs as sentimental words while calculating the polarity score, both NE tags (PosNE, NegNE) were added to the lexicon as positive and negative entries having the scores of 1 and -1, respectively. Fig (4.4) shows the pipeline of the lexicon-based approach.

4.6. Conclusion

In this chapter, we introduced our HCB Tw-StAR model and reviewed the pipeline it adopts to perform SA of DA. We described how HCB tw-StAR can bridge the differences among the Arabic dialects based on the novel NE features which are unified across the Arabic dialects. In this context, we presented our NEs sentiment detection algorithm with which NEs in a dataset are correlated with its relevant polarity and included among the sentiment features. Later, we explored the different NLP tasks employed at the preprocessing phase of HCB Tw-StAR highlighting their ability to produce more expressive DA sentiment features without the need for dialect-specific morphological analyzers or DA external semantic resources. Finally, the employed preprocessing tasks together with the presented NEs features were combined within supervised and lexicon-based sentiment classifiers. In the next chapter, the impact of the our NE features along with novel combinations of preprocessing tasks will be, practically, investigated through various experiments and with multi-lingual datasets.

5. HCB TW-STAR EXPERIMENTS AND EVALUATION

In this chapter, we review the different experiments conducted to evaluate HCB Tw-StAR as an Arabic/multilingual SA model at two granularity sentiment analysis levels. Through the conducted experiments, we investigate how the performance of sentiment classification is affected by: (a) novel combinations of preprocessing tasks, (b) NEs as sentiment indicatives and (c) the joint impact of preprocessing tasks together with NEs.

5.1. Experiments Setup

Following the SA pipeline adopted by HCB Tw-STAR (see Figure 4.1), we conducted various experiments using DA and non-Arabic evaluation datasets. As it can be seen from Figure 4.3, the experiments were designed such that the single/joint impact of each of the preprocessing task combinations and NEs features on SA, could be investigated. Therefore, HCB Tw-StAR was used in the experiments in three different ways:

1. With the preprocessing phase excluded: to investigate the impact of NEs on SA.
2. With the NEs processing phase excluded: to investigate the impact of preprocessing combinations on SA.
3. With both NEs and preprocessing phases included: to investigate the joint impact of NEs and preprocessing combinations on SA.

Within the conducted experiments, HCB Tw-StAR was examined with Arabic datasets of different Eastern/Western dialects in addition to non-Arabic datasets including: English, Spanish and Turkish. Our model was further used to recognize multiple emotions embedded, simultaneously, in a sentence at a fine-grained analysis level.

5.2. Evaluation Datasets

The datasets used to evaluate HCB Tw-StAR combine Arabic and non-Arabic tweets, Facebook comments and reviews, harvested from several social media platforms and review websites. The statistics of all datasets and their training (Train) , developing (Dev) and testing (Test) sets are listed in Table 5.1; while the details about the contents and resources of these datasets are given as follows:

1. **Arabic Datasets:** Four publicly available datasets with an MSA/DA of tweets and Facebook comments were used. These datasets were manually collected and annotated for positive and negative polarity.

   - Tunisian Election dataset (TEC): a set of 5,521 tweets collected by Sayadi et al. (2016) during the Tunisian elections period. It combines MSA and Tunisian dialect where Tunisian tweets form the majority of the data. with neutral tweets reduced, 3,043 tweets were used.
   - Tunisian Arabic dataset (TAC): a dataset composed of 800 tweets which cover multiple topics such as media, telecom and politics. This dataset was collected by Karmani (2017) and annotated for positive, negative and neutral polarity. We only handled the positive and negative instances such that 746 tweets were adopted.
   - Tunisian Sentiment Analysis dataset (TSAC): a dataset of 9,976 Facebook comments provided by Mdhaffar et al. (2017). These comments represent the reactions of the audience towards popular Tunisian TV shows, they were annotated manually for positive and negative polarity. In this study, we filtered the Arabizi instances out of this dataset such that 7,366 comments were used.
   - Arabic Jordanian General Tweets (AJGT): a dataset composed of 1,800 positive/negative social Jordanian tweets obtained by Alomari et al. (2017).
   - SemEval-Arabic (SemAr): a dataset of 4,381 DA tweets labeled with multiple human emotions such as anger, happiness, sadness and so on. It was presented within the context of SemEval-2018 shared task for affect detection in tweets (Mohammad et al. (2018)).

2. **Turkish Datasets:** five Turkish datasets obtained from (Demirtas and Pechenizkiy (2013)) were used to evaluate the model with the Turkish language. These datasets contain positive and negative, multiple-domain reviews distributed evenly in each of:

   - Products: includes four data collections: Kit, DVD, Elec and Books. Each of which contains 1,400 reviews harvested from the online retailer Turkish website (hepsiburada.com).
   - Movies: a dataset of 10,662 reviews collected from the common movie reviews Turkish website (Beyazperde.com).

3. **English Dataset (SemEng):** along with the Arabic dataset provided in (Mohammad et al. (2018)), an English datasets of 10,983 tweet annotated with multiple emotions was proposed for evaluation.

4. **Spanish Dataset (SemEs):** it is also presented in (Mohammad et al. (2018)) and combines 7,092 Spanish tweets associated with multiple emotions.

Table 5.1. Statistics and polarity distribution across the used datasets

Lang./Dial.	Dataset	Train+Dev		Test		Total
		positive	negative	positive	negative	
Tunisian	TEC	968	1,466	276	333	5,521
	TAC	306	290	76	74	746
	TSAC	2,782	3,451	672	890	7,795
Jordanian	AJGT	758	682	142	218	1,800
Turkish	Movies	4,270	4,258	1,061	10,73	10,662
	Kit	561	559	139	141	1,400
	DVD	544	576	156	124	1,400
	Elec	552	568	148	132	1,400
	Books	553	567	147	133	1,400
Multi-dialects	SemAr	2,863		1,518		4,381
English	SemEng	7,269		3,259		10,983
Spanish	SemEs	4,238		2,854		7,092

5.3. Named Entities Impact on Sentiment Analysis

Here, we evaluate the effectiveness of NEs in inferring the sentiment of MSA/DA tweets and Facebook comments. Therefore, we employed HCB Tw-StAR described in Figure (4.1) with the prerprocessing phase excluded. The model was applied to mine the sentiment of tweets/comments written in MSA in addition to Tunisian and Jordanian dialects and combined in TEC, TAC, TSAC and AJGT datasets (see Table 5.1).

In order to exploit NEs for the SA task, they should be manipulated within the NEs processing phase. First, NEs were extracted from the training division of each of the tackled datasets using the Arabic NER model from (Gridach (2016)). Then, using the NEs sentiment detection explained in Section 4.2.2 and illustrated in Figure (4.2), NEs were identified as positive or negative, finally, they were replaced with a textual tag that defines the polarity they bear: PosNE or NegNE. The statistics of the extracted NEs are listed in Table 5.2 where E-NEs, Pos-NEs, Neg-NEs and A-NEs denote the number of the extracted NEs, positive NEs, negative NEs and the sentiment-annotated NEs, respectively.

Table 5.2. NEs statistics extracted from each dataset

Dataset	E-NEs	Pos-NEs	Neg-NEs	A-NEs
AJGT	175	52	118	170
TAC	240	99	129	228
TEC	658	192	410	602
TSAC	615	198	350	548

We notice that, from large-sized datasets such as TSAC and TEC, more NEs could be extracted. In addition, in all datasets, the number of negative NEs is greater than that of positive NEs. On the other hand, although TSAC has a larger size than TEC; yet, less NEs were extracted from it compared to the NEs extracted from TEC. This is due to the fact that, the used NER system exploited pre-trained word embeddings from (Zahran et al. (2015)). These embeddings were produced with corpora composed of MSA, Egyptian and Levantine content which could support the MSA content of TEC, while it is, relatively, far from the pure-Tunisian dialect in TSAC. Consequently, most of the Tunisian terms will not be found in the lookup table of the NER system. Hence, their embeddings were initialized randomly instead of being initialized with pre-trained word embeddings (Gridach (2016)).

In line with HCB Tw-StAR pipeline, having the NEs extracted, annotated for sentiment and replaced with specific textual tags, they are included among the n-gram training features to be fed later to the sentiment classification phase.

5.3.1. Supervised Classification

The supervised classifier of HCB Tw-StAR was trained once without tagging NEs (Tw-StAR), then, with NEs tagged and included among the features (Tw-StAR+NEs). Three experiment variants were conducted, the first involved using all n-gram features, while the second and third used a reduced number of features obtained by the TF scheme for the thresholds: 2 and 3, respectively. We chose to review the results of the experiment of the best achieved average F-measure, with/without NEs. Table 5.3 lists this model's results where uni, bi and tri refer to unigrams, bigrams and trigrams, respectively. While Prec, Rec, F1 and Acc. indicate the averaged precision, recall, F-measure and accuracy, respectively. A comparison with baseline systems is shown in Table 5.4.

Table 5.3. Supervised Tw-StAR with/without NEs for all datasets

Dataset	NEs	Algorithm	Prec.(%)	Rec(%)	F1(%)	Acc.(%)
AJGT	No	NB	**87.2**	**85.3**	**86.0**	**86.9**
		SVM	82.8	80.7	81.5	82.8
	yes	NB	**88.4**	**86.5**	**87.2**	**88.1**
		SVM	83.4	81.3	82.1	83.3
TAC	No	NB	83.4	81.9	81.8	82.0
		SVM	**85.2**	**84.6**	**84.6**	**84.7**
	yes	NB	84.4	83.2	83.2	83.3
		SVM	83.4	83.3	83.3	83.3
TEC	No	NB	71.8	68.8	68.7	70.4
		SVM	**75.0**	**71.4**	**71.4**	**73.1**
	yes	NB	72.3	69.6	69.6	71.1
		SVM	**74.4**	**71.2**	**71.2**	**72.7**
TSAC	No	NB	91.2	92.0	91.4	91.4
		SVM	**92.8**	**92.5**	**92.7**	**92.8**
	yes	NB	91.6	92.4	91.7	91.7
		SVM	**92.4**	**92.2**	**92.3**	**92.4**

When exploring the performances of the supervised classifiers in Table 5.3, we notice that with NEs included, a degraded performance was observed as F-measure values were decreased in all datasets, except AJGT which exhibited a slight improvement. On the other hand, the F-measure values obtained with/without NEs for the Tunisian datasets: TEC, TAC and TSAC were comparable. This degrade could be due to the fact that, inferring the sentiment using n-grams depends on capturing the co-occurrence information contained within these n-gram schemes. Since NEs are identified as positive or negative based on how frequent they are mentioned within a sentence of a positive or a negative polarity, and regardless of the co-occurrence words; therefore, it is possible for a positive NE to be included along with negative sequence of words (n-gram scheme) and vice versa which makes some of the n-grams, that contain NE tags, misleading features.

Table 5.4. Supervised Tw-StAR with/without NEs against baselines

Dataset	Model	Prec.(%)	Rec(%)	F1(%)	Acc.(%)
AJGT	**Alomari et al. (2017)**	**92.1**	84.9	**88.2**	**88.7**
	Tw-StAR	87.2	85.3	86.0	86.9
	Tw-StAR + NEs	88.4	**86.5**	87.2	88.1
TAC	**Karmani (2017)**	63.0	72.9	67.3	72.1
	Tw-StAR	**85.2**	**84.6**	**84.6**	**84.7**
	Tw-StAR + NEs	83.4	83.3	83.3	83.3
TEC	**Sayadi et al. (2016)**	67.0	71.0	63.0	71.1
	Tw-StAR	**75.0**	**71.4**	**71.4**	**73.1**
	Tw-StAR + NEs	74.4	71.2	71.2	72.7
TSAC	**Mdhaffar et al. (2017)**	78.0	78.0	78.0	78.0
	Tw-StAR	**92.8**	**92.5**	**92.7**	**92.8**
	Tw-StAR + NEs	92.4	92.2	92.3	92.4

5.3.2. Lexicon-based Classification

Considering the lexicon-based classifier of HCB Tw-StAR, the experiments where NEs are not included (Tw-StAR) were conducted considering two lexicons: (a) an integrated lexicon constructed out of NileULex, AEL, AHL, LevLex and GulfLex (see Table 4.2) to handle AJGT dataset whose content is mostly Levantine combined with MSA and (b) TunLex to mine the sentiment of TAC, TEC and TSAC datasets. While for the experiments that include NEs (Tw-StAR+NEs), the same previous lexicons were used but with positive and negative NEs tags: PosNE, NegNE added as entries having positive and negative scores, respectively. The sentiment detection procedure was carried out by looking for a sentence's unigrams (uni) then unigrams and bigrams (uni+bi) in the relevant lexicon, once with NEs tagged then with them treated as ordinary tokens. The best results of the lexicon-based classifier are shown in Table 5.5. The obtianed performances were, further, compared against baseline systems as it is shown in Table 5.6.

Unlike the supervised classifier, the performance of the lexicon-based classifier was favorably impacted by NEs inclusion in the SA task. As it can be seen in

Table 5.5. Lexicon-based Tw-StAR with/without NEs for all datasets.

Dataset	NEs	Feats.	Prec.(%)	Rec(%)	F1(%)	Acc.(%)
AJGT	No	uni+bi	82.4	83.8	81.6	81.7
	yes	uni+bi	**84.7**	**86.3**	**84.5**	**84.7**
TAC	No	uni+bi	66.9	66.7	66.6	66.7
	yes	uni+bi	**70.8**	**70.6**	**70.6**	**70.7**
TEC	No	uni+bi	66.6	61.5	59.8	64.0
	yes	uni+bi	**69.1**	**65.6**	**65.0**	**67.5**
TSAC	No	uni+bi	84.5	83.8	81.8	81.8
	yes	uni+bi	**84.6**	**84.7**	**82.8**	**82.8**

Table 5.5, for uni+bi features, the sentiment classification performance, with NEs considered and NE tags added to the lexicons, could outperform the one obtained by the ordinary lexicons. Indeed, the evaluation measures increased in all datasets as the F-measure values of Tw-StAR+NEs were 84.5%, 70.6%, 65% and 82.8% compared to 81.6%, 66.6%, 59.8% and 81.8% achieved by Tw-StAR for AJGT, TAC, TEC and TSAC datasets, respectively. The reason behind such improvement is that uniform weight scheme lexicons ignore the contextual-related information where a sentence's polarity is defined based on the polarity scores of its constituent words (El-Makky et al. (2014); El-Beltagy and Ali (2013)). This, in turn, enables the sentiment-annotated NEs deduced regardless of the context, to effectively contribute in recognizing the polarity of the sentence containing it. Moreover, with NEs tagged in the test dataset, it became possible to employ person names in the SA task. Hence, the issue caused by confusing a person name with an adjective could be avoided without the need to eliminate person names as in (El-Makky et al. (2014); El-Beltagy and Ali (2013)).

Table 5.6. Lexicon-based Tw-StAR with/without NEs against baselines

Dataset	Model	Prec.(%)	Rec(%)	F1(%)	Acc.(%)
AJGT	**Alomari et al. (2017)**	**92.1**	84.9	**88.2**	**88.7**
	Tw-StAR	82.4	83.8	81.6	81.7
	Tw-StAR + NEs	84.7	**86.3**	84.5	84.7
TAC	**Karmani (2017)**	63.0	72.9	67.3	**72.1**
	Tw-StAR	66.9	66.7	66.6	66.7
	Tw-StAR + NEs	**70.8**	**70.6**	**70.6**	70.7
TEC	**Sayadi et al. (2016)**	67.0	**71.0**	63.0	**71.1**
	Tw-StAR	66.6	61.5	59.8	64.0
	Tw-StAR + NEs	**69.1**	65.6	**65.0**	67.5
TSAC	**Mdhaffar et al. (2017)**	78.0	78.0	78.0	78.0
	Tw-StAR	84.5	83.8	81.8	81.8
	Tw-StAR + NEs	**84.6**	**84.7**	**82.8**	**82.8**

Considering Table 5.6 which compares the lexicon-based model against the baseline systems, it should be noted that this comparison is considered meaningful only for for TAC dataset where the baseline system (Karmani (2017)) is also a lexicon-based model; yet, we observed that Tw-StAR+NEs outperformed the baselines in Tunisian datasets: TAC, TEC and TSAC. This could be explained by the positive impact of NEs on the polarity detection in addition to the large coverage provided by the used Tunisian lexicon. In contrast, it is reasonable that the performance degraded in AJGT dataset as the F-measure decreased by 3.7% compared to Alomari et al. (2017) in which the data was subjected to several preprocessing (stemming/light stemming) before feeding them to the supervised classifires (SVM/NB). It should be noted that, the best performances of the lexicon-based classifier of HCB Tw-StAR, with/without NEs, were obtained with unigram and bigram features. This is attributed to the fact that, our lexicons are rich of compound terms; therefore, looking up for uni+bi tokens in the lexicon, increases the matching ratios of compound terms and, thus, raises the sentiment recognition accuracy.

Finally, for the datasets rich of NEs (see Table 5.2) such as TSAC and TEC; we could not determine the impact of the number of the sentiment-annotated NEs on SA within Tw-StAR+NE lexicon-based model. To clarify that, although TEC has the greatest number of sentiment-annotated NEs, the improvement recorded in the F-measure value was 2%, while for TSAC that has less annotated NEs, the achieved improvement was remarkably better as the F-measure increased by 4.8%. We believe that, when NEs are included among the features, the performance of the lexicon-based model for a specific dataset, is not related to the number of the sentiment-annotated NEs in the dataset as much as it is to the data consistency of that dataset. More specifically, in a dataset having a good degree of consistency, the training and test data tend to contain more similar NEs. Thus, it is more likely to have a consensus on the sentiment of a specific NE which leads to an accurate sentiment assignment of that NE and, hence, to a better sentiment classification.

5.4. Preprocessing Impact on Sentiment Analysis

Within the proposed SA model HCB Tw-StAR, the preprocessing tasks listed in Section 4.3 were examined, first, one by one, then, combined in various schemes. This enabled defining the preprocessing task/combination for which the SA performance is better improved. The preprocessing impact on SA was evaluated for multiple languages in addition to MSA/DA and considering two analysis levels: coarse-grained and fine-grained. For coarse-grained sentiment analysis experiments, HCB Tw-StAR with its supervised and lexicon-based SA approaches, was used to classify the sentiment embedded in each input tweet into one of two polarities: positive or negative. The performances obtained for each single/combination preprocessing tasks were compared against each other and against baseline systems. It should be noted that, given that, we selected the positive/negative instances of TEC and TAC datasets, a fair comparison against the systems that used these datasets should be enabled. This was possible for TEC dataset as Sayadi et al. (2016) included the results of the binary classification experiments in their study. However, for TAC, we could deduce the baseline evaluation measures considering only the positive/negative tweets since Karmani (2017) provided the confusion matrix data.

In the following sections, a detailed review of the experiments and the results obtained in both analysis levels is provided.

5.4.1. Supervised Coarse-grained Sentiment Analysis Experiments

In the supervised classification approach, three variants of experiments were conducted. The first one involved using all N-grams features: unigrams (uni), bigrams (bi), trigrams (tri) and combinations of them (uni+bi, uni+bi+tri), while the second and third experiments used a reduced number of the same features resulted from applying TF weighting with two threshold values defined empirically as 2 and 3, respectively. Table 5.7, Table 5.8, Table 5.9 and Table 5.10 list the best performances achieved by either NB or SVM in the supervised model, compared to the baseline approaches. The used algorithm, precision, recall, F-measure and accuracy are referred to as (Alg.), (P.), (R.), (F1.) and (Acc.), respectively.

The results in Table 5.9 clearly suggest that SVM always performed better than NB for large-sized datasets such as TSAC. This could be explained by the ability of LIBSVM to handle the sparsity and high-dimensionality of the training feature vectors (Chang and Lin (2011)).

Table 5.7. Preprocessing with supervised HCB Tw-StAR for TEC

Preprocessing	Features	Alg.	P.(%)	R.(%)	F1.(%)	Acc.(%)
Sayadi et al. (2016)	uni+bi	SVM	67	71	63	71.1
Stop	uni	SVM	72	70.5	70.6	71.6
Stem	uni	NB	75.3	**73.4**	**73.6**	**74.5**
LightS	uni	NB	74.9	72.4	72.5	73.7
Neg	uni+bi	SVM	**75.7**	71.7	71.7	73.4
Stem + Stop	uni	NB	**75.7**	73.3	73.4	**74.5**
LightS + Stop	uni	NB	74.9	71.7	71.7	73.2
LightS + Neg	uni	NB	74.5	72	72.1	73.4

Table 5.8. Preprocessing with supervised HCB Tw-StAR for TAC

Preprocessing	Features	Alg.	P.(%)	R.(%)	F1.(%)	Acc.(%)
Karmani (2017)	morph.	Lex	63	72.9	67.3	72.1
Stop	uni	NB	82.9	79.8	79.5	80
Stem	uni	SVM	86.3	**85.9**	**85.9**	86
LightS	uni+bi	NB	85.8	84.5	84.5	84.7
Neg	uni+bi	SVM	**86.6**	**85.9**	**85.9**	**86**
Stem + Stop	uni+bi	NB	83.9	82.5	82.5	82.7
LightS + Stop	uni+bi	NB	85.3	83.9	83.3	84
LightS + Neg	uni+bi	NB	85.8	84.5	84.5	84.7

It has been noted that stemming using Farasa improved the supervised sentiment classification performance in TEC, TAC datasets (Table 5.7, Table 5.8) where it achieved the second best F-measure (85.9%) in TAC outperforming the baseline by 18.6%. Although Farasa was trained with MSA corpora, it succeeded in identifying the affixes to be cut in Tunisian words because of the lexical overlap between MSA and DA in general (Samih et al. (2017)). In order to retain the variety of words having same root and different meanings, we have also used light stemming. Compared to other single preprocessing tasks, light stemming had the best impact on the F-measure in AJGT dataset and achieved the best performance among all the preprocessing schemes when it was combined with the negation detection task. However, it could not overcome the stemming impact in TEC,TAC and TSAC datasets even when it was combined with other preprocessing techniques.

When tracking the sentiment classification results, yielded from conducting stopwords removal, across the datasets: TEC, TAC, TSAC, it could be observed that stopwords reduction could remarkably improve performance, compared to base-

Table 5.9. Preprocessing with supervised HCB Tw-StAR for TSAC

Preprocessing	Features	Alg.	P.(%)	R.(%)	F1.(%)	Acc.(%)
Mdhaffar et al. (2017)	doc emb.	MLP	78	78	78	78
Stop	uni	SVM	92.5	92.3	92.4	92.6
Stem	uni	SVM	93.4	93.4	93.4	93.5
LightS	uni+bi	SVM	93.1	92.8	92.9	93.1
Neg	uni	SVM	92.6	92.5	92.5	92.7
Emo	uni	SVM	92.4	92.39	92.4	92.5
Stem + Stop	uni	SVM	93.8	93.8	93.8	93.9
LightS + Stop	uni+bi	SVM	93.4	93.2	93.33	93.5
LightS + Neg	uni+bi	SVM	93.1	92.8	92.9	93.1
Emo + Stop	uni	SVM	92.1	92.1	92.2	92.3
Emo + Stem	uni	SVM	**93.9**	**93.8**	**93.9**	**94**
Emo + LightS	uni+bi	SVM	93.1	93.1	93.1	93.2
Emo + Neg	uni	SVM	92.5	92.4	92.5	92.6
Emo + Stem + Stop	uni	SVM	93.8	93.8	93.8	93.9
Emo + LightS + Neg	uni+bi	SVM	93.1	93.1	93.1	93.2

line systems, in all datasets. As it can be seen in Table 5.7, Table 5.8 and Table 5.9, with stopwords eliminated, the achieved F-measure values were 70.6%, 79.5% and 92.4% against 63%, 67.3% and 78% scored by the baseline models for TEC, TAC and TSAC datasets, respectively. On the other hand, it was observed that combining stopwords with stemming was not always useful. For instance, in TSAC, the sentiment classification performance resulting from the single application of stemming was slightly improved by 0.4% when stemming and stopwords removal were applied. However, in TEC, TAC and AJGT datasets (Table 5.7, Table 5.8 and Table 5.10, combining stopwords reduction with stemming has degraded the performance providing less F-measure values. This could be due to the fact that, the adopted MSA stopwords can improve the stemming performance if the corpora to be stemmed are also of an MSA content. While for our dialectal datasets, it is not guaranteed that all the stopwords could be detected and removed; this leads the stemmer to provide invalid stems for the undetected Tunisian and Jordanian stopwords yielding less efficient sentiment classification performance.

Emoji were detected only in TSAC dataset as TEC, TAC and AJGT datasets do not contain any emoji. In TSAC, emoji tagging had no significant impact on the performance when it was separately applied whereas combining emoji tagging along with stemming scored the best F-measure among all the experiments with a value equals to 93.9%. Moreover, applying emoji tagging together with negation

Table 5.10. Preprocessing with supervised HCB Tw-StAR for AJGT

Preprocessing	Features	Alg.	P.(%)	R.(%)	F1.(%)	Acc.(%)
Alomari et al. (2017)	bi	SVM	**92.1**	84.9	88.2	88.7
Stop	uni	SVM	84.6	83.1	83.7	84.7
Stem	uni+bi	NB	87.3	86.4	86.8	87.5
LightS	uni+bi	NB	87.3	87.2	87.2	86.4
Neg	uni+bi	NB	88.0	86.2	86.9	87.8
Stem + Stop	uni+bi+tri	NB	85.9	86.7	86.2	86.7
LightS + Stop	uni+bi+tri	NB	86.3	85.6	85.9	86.7
Stem+ Neg	uni+bi+tri	NB	87.8	86.5	87.0	87.8
LightS + Neg	uni+bi+tri	NB	88.6	**88.0**	**88.3**	**88.9**

achieved almost the same results scored by the negation preprocessing task. This could be due to the sarcastic content in which emoji do not express the actual sentiment but its opposite.

Considering the negation tagging task, it could improve the sentiment classification performance in the Tunisian datasets as it is shown in Tables 5.7, 5.8, 5.9 and 5.10. Nevertheless, the least improvement was reported in TEC as the accuracy was increased by 2.31% compared to 13.9% and 14.7% increment ratios scored in TAC and TSAC datasets respectively. This could be attributed to the context in which the negation words are used, where in datasets of a political domain such as TEC, negation can be used in a narrative way and does not necessary indicates a negative sentiment. On the other hand, the positive impact of negation on the SA performance in AJGT datasets was clear when it is combined with light stemming (Table 5.10) achieving the best F-measure with an increment equals to 1% over the baseline F-measure value. This might be attributed to the natural use of negations in AJGT dataset whose content is mostly social.

5.4.2. Lexicon-based Coarse-grained Experiments

In these experiments approach, each tweet/comment was tokenized into unigrams (uni) then into combinations of unigrams and bigrams (uni+bi) to be looked up later in the manually-built Tunisian lexicon where SFS algorithm was used to calculate the polarity score; while emoji and negation textual tags were added to the lexicon as positive/negative sentiment words. Table 5.11, Table 5.12, Table 5.13

and Table 5.10 list the best performances achieved by the lexicon-based model, for several preprocessing tasks, against the baseline models. It should be noted that, this comparison is only meaningful for TAC dataset where the baseline system (Karmani (2017)) is a lexicon-based one.

Table 5.11. Preprocessing with lexicon-based HCB Tw-StAR for TEC

Preprocessing	Features	Alg.	P.(%)	R.(%)	F1.(%)	Acc.(%)
Sayadi et al. (2016)	uni+bi	SVM	67	**71**	63	**71.1**
Stop	uni+bi	Lex	66.6	61.5	59.8	64
Stem	uni+bi	Lex	67.2	64.9	64.5	66.5
LightS	uni+bi	Lex	64.4	63.4	63.3	64.4
Neg	uni+bi	Lex	**68.1**	62.3	60.5	64.9
Stem + Stop	uni+bi	Lex	67.1	65.7	**65.7**	67
LightS + Stop	uni+bi	Lex	65.7	64.4	64.2	65.7
LightS + Neg	uni+bi	Lex	64.6	63.4	63.3	64.7

Table 5.12. Preprocessing with lexicon-based HCB Tw-StAR for TAC

Preprocessing	Features	Alg.	P.(%)	R.(%)	F1.(%)	Acc.(%)
Karmani (2017)	morph.	Lex	63	**72.9**	67.3	**72.1**
Stop	uni+bi	Lex	65	64.8	64.5	64.7
Stem	uni+bi	Lex	65.3	65.3	65.3	65.3
LightS	uni+bi	Lex	67.3	67.3	67.3	67.3
Neg	uni+bi	Lex	**69.1**	68.8	**68.6**	68.7
Stem + Stop	uni+bi	Lex	62.4	62.1	61.8	62
LightS + Stop	uni+bi	Lex	66.7	66.7	66.7	66.7
LightS + Neg	uni+bi	Lex	68	68	68	68

Considering Table 5.11 which contains the results produced for TEC dataset, we noticed that stemming combined with stopwords removal has led to the best SA performance with an F-measure of 65.7% and outperformed the supervised baseline model. While this contradicts with the behavior observed in the supervised approach, it could be justified based on the SA principle of lexicon-based methods. Where recognizing the sentiment through lexicon-based approaches merely relies on the hits found between the studied dataset and the employed lexicon. Consequently, the invalid stems, resulting from not removing DA stopwords, that usually mislead the supervised classifier, will not be considered here; since there are no correspondent entries for them in the used lexicon.

The sentiment classification results obtained for TAC dataset (see Table

Table 5.13. Preprocessing with lexicon-based HCB Tw-StAR for TSAC

Preprocessing	**Features**	**Alg.**	**P.(%)**	**R.(%)**	**F1.(%)**	**Acc.(%)**
Mdhaffar et al. (2017)	doc emb.	MLP	78	**78**	**78**	**78**
Stop	uni+bi	Lex	82	69	68.3	73.2
Stem	uni+bi	Lex	82.6	72	72	75.6
LightS	uni+bi	Lex	82	72.4	72.5	75.9
Neg	uni+bi	Lex	82.8	69.5	68.8	73.6
Emo	uni+bi	Lex	82.5	70.9	70.6	74.4
Stem + Stop	uni+bi	Lex	81.7	71.78	71.7	75.3
LightS + Stop	uni+bi	Lex	81.9	72	72	75.5
LightS + Neg	uni+bi	Lex	81.9	72.2	72.3	75.7
Emo + Stop	uni+bi	Lex	82.3	70.4	70	74.3
Emo + Stem	uni+bi	Lex	83	73	73.2	76.5
Emo + LightS	uni+bi	Lex	82.3	73.3	73.6	76.6
Emo + Neg	uni+bi	Lex	**83.1**	70.8	70.5	74.7
Emo + Stem + Stop	uni+bi	Lex	82.2	72.9	73.1	76.3
Emo + LightS + Neg	uni+bi	Lex	82.2	73.2	73.4	76.5

Table 5.14. Preprocessing with lexicon-based HCB Tw-StAR for AJGT

Preprocessing	**Features**	**Alg.**	**P.(%)**	**R.(%)**	**F1.(%)**	**Acc.(%)**
Alomari et al. (2017)	bi	SVM	**92.1**	84.9	**88.2**	**88.7**
Stop	uni+bi	Lex	82.2	83.5	81.3	81.4
Stem	uni+bi	Lex	77.5	78.4	77.8	78.3
LightS	uni+bi	Lex	82.4	79.3	80.2	80.6
Neg	uni+bi	Lex	86.0	**87.4**	86.4	86.7
Stem + Stop	uni+bi	Lex	76.6	77.4	76.8	77.5
LightS + Stop	uni+bi	Lex	82.8	79.7	80.6	82.2
Stem+ Neg	uni+bi	Lex	87.8	78.5	78.2	78.9
LightS + Neg	uni+bi	Lex	82.1	79.1	80.0	81.7

5.12), indicated that, negation detection and tagging was the best-performing preprocessing task with an F-measure of 85.9% outperforming the baseline performance by 18.6%. We believe that this could be explained as follows: besides the efficient list of Tunisian negation words, adding the negation textual tags to the lexicon as sentiment words having a negative polarity score, enabled the lexicon-based classifier to consider the impact of negation word encountered within an input sentence.

For TSAC dataset, it could be observed in Table 5.13 that, regardless of the performance of the supervised baseline model, the best performance in terms of F-measure, was achieved by the combination of emoji tagging and light stemming.

This indicates the merit given by light stemming, as the word's affixes are chopped-off such that valid stems are provided; which, in turn, enabled the produced stems to retain the sentiment born by the original inflected words. On the other hand, emoji tagging enhanced the ability of the lexicon-based approach to recognize the sentiment especially that, all the emoji icons were unified into two textual tags; both of them were added to the used lexicon as positive/negative entries.

Regarding AJGT dataset, according to the results listed in Table 5.14, we can notice that regardless of the performance of the supervised baseline model, negation was of the best performance among the other preprocessing schemes with an F-measure of 86.4%. This complies with the impact of negation on SA in the supervised classifier where using negations to mostly imply negative sentiments makes negation detection of an effective impact on the sentiment recognition in both supervised and lexicon-based models.

5.4.3. Turkish Datasets Experiments

As an attempt to target non-Arabic languages, HCB Tw-StAR was examined with Turkish datasets (see Table 5.1). The same pipeline shown in Figure (4.1), with NEs phase excluded, was adopted. However, the Arabic NLP resources and tools such as stopwords, negation words, sentiment lexicons and stemmers were replaced with Turkish-specific ones. More details about the NLP Turkish tools and resources can be found in Section 4.3.

The supervised classifier of HCB Tw-StAR was used to mine the Turkish sentiment with the same preprocessing tasks, n-gram schemes, TF thresholds and supervised classification algorithms that were employed in the Arabic model variant. Table 5.15 and 5.16 list the accuracy values achieved by NB and SVM classifiers, respectively; where the performances were obtained for different single/combinations of preprocessing tasks. The baseline represents the accuracy scored by the system of Demirtas and Pechenizkiy (2013).

The results in Table 5.15 and Table 5.16 suggest that NB outperforms SVM for almost all datasets. On the other hand, both NB and SVM algorithms could outperform the baseline for all datasets with a considerable margin especially in the Movies dataset; where the proposed model achieved an accuracy of 92.8% when

Table 5.15. NB accuracy (%) for all preprocessing tasks

Preprocessing	Datasets				
	Movies	Kit	DVD	Elec	Books
Demirtas and Pechenizkiy (2013)	69.5	75.9	76.0	73.0	72.4
Stop	91.0	78.2	**81.4**	**85.7**	83.9
Stem	91.3	78.6	80.0	85.4	**88.6**
Stop+Stem	90.9	81.4	72.5	82.1	85.7
Emo	**92.8**	80.4	80.4	84.6	87.9
Emo+Stem	90.8	81.8	76.8	81.4	84.3
Neg	92.3	**85.0**	80.4	83.9	86.1
Neg+Stem	90.7	82.1	74.3	83.2	85.4

Table 5.16. SVM accuracy (%) for all preprocessing tasks

Preprocessing	Datasets				
	Movies	Kit	DVD	Elec	Books
Demirtas and Pechenizkiy (2013)	66.0	70.0	70.3	72.4	66.6
Stop	88.8	72.9	80.0	81.8	85.0
Stem	88.2	77.5	**81.4**	82.9	**87.1**
Stop+Stem	87.6	78.6	71.4	75.4	81.8
Emoji	89.5	73.2	74.6	80.0	83.6
Emoji+Stem	88.3	77.5	76.8	75.4	79.6
Neg	**91.8**	**84.3**	78.2	**83.2**	82.5
Neg+Stem	87.7	78.9	76.8	81.8	80.4

emoji icons were tagged compared to 69.5% scored by the baseline system (Demirtas and Pechenizkiy (2013)) increasing the accuracy by 23.3%.

While stopwords are considered a noisy data and are, therefore, usually eliminated in English SA system, our experiments revealed that removing stopwords does not have a significant impact on the classification performance. Having the stopwords removed, it can be seen in Table 5.16 that the sentiment classification accuracy values for DVD and Elec datasets were slightly better than those obtained when stemming was applied.

Table 5.16 reveals that with NB classifier used, stemming has increased the accuracy in most datasets especially in Books dataset with an accuracy of 88.6%. This can be attributed to the formal nature of the content of this dataset which enables the extraction of identical stems for inflectional words. Thus, the sparsity of features is reduced and sentiment indicative words are retained yielding good classification results.

In contrary to what was expected, as it is shown in Table 5.15 and Table 5.16, applying other preprocessing techniques such as stopwords reduction, emoji recognition or negation tagging combined with stemming, could not score the best accuracy in both SVM, NB classifiers for all datasets. For instance, the sentiment classification accuracy in Books dataset, when NB classifier is used, has decreased from 88.6% scored by stemming to 84.3% achieved when emoji recognition is applied together with stemming. The reason behind that is the low number of emoji contained in the Books dataset; where the effective impact of emoji recognition on SA is related to the frequent appearance of emoji in the tackled dataset. This made emoji tagging useless and sometimes confusing to the classifier. Nevertheless, Table 5.16 indicates that using emoji recognition separately could enhance the performance in the datasets containing a considerable number of emoji icons such as Movies where the achieved accuracy was the best among all the experiments with a value equals to 92.8%.

Although negation tagging yielded the best classification accuracy in the Kit dataset for both NB and SVM algorithms (Table 5.15, Table 5.16), it failed to increase the accuracy in the remaining datasets. The inconclusive impact of negation can be attributed to the ignorance of the negated verbs; as they require a special manipulation; especially that negation affixes in Turkish language are embedded within verbs and cannot be captured easily (Yıldırım et al. (2015)). Nevertheless, the proposed negation detection strategy scored better results compared to Yıldırım et al. (2015) where the best accuracy increment achieved by the lexicon-based classifier of HCB Tw-StAR via negation was 25.8% for Kit dataset, compared to a degradation from 79.06% to 78.27% in (Yıldırım et al. (2015)). This could be due to the fact that, we inferred the negation in an adjective by replacing a specific affix with the negation tag whereas Yıldırım et al. (2015) added a negative tag to the adjectives preceding the negated verbs.

As for the lexicon-based classifier, we followed the same pipeline used in the Arabic lexicon-based method; However, the lexicon replaced by SentiTurkNet (see Table 4.2). Table 5.17 shows the best F-measure values scored for all preprocessing techniques; where the baseline refers to a multi-lingual lexicon based model developed in (Araujo et al. (2016)) to recognize the sentiment in the same datasets used in this thesis.

According to the results in Table 5.17, the lexicon-based classifier has a poor performance compared to the supervised model. This is due to the low coverage of the used lexicon. While comparable performances were scored by our model compared to the baseline for Kit, DVD, Elec and Books, our lexicon-based classifier of

Table 5.17. Lexicon-based F-measure (%) for all preprocessing tasks

Preprocessing	Datasets				
	Movies	Kit	DVD	Elec	Books
Araujo et al. (2016)	62.0	**62.0**	**62.0**	**62.0**	**62.0**
Stop	63.5	51.1	50.7	51.1	50.4
Stem	**66.1**	51.0	54.3	57.9	60.0
Stop+Stem	65.2	61.1	60.7	57.1	57.9
Emoji	63.4	50.7	54.3	52.5	56.1
Emoji+Stem	65.2	50.7	56.4	54.3	59.3
Neg	63.2	51.4	53.9	53.6	56.1
Neg+Stem	65.5	58.6	57.1	55.4	56.4

HCB Tw-StAR could outperform the baseline model for movies datasets; as the best F-measure achieved was 66.1% compared to 62% scored in Araujo et al. (2016). In addition, similar to the supervised model, stemming improved the performance in Books and movies datasets scoring an F-measure of 66.1% and 60% respectively. This could be due to the formal language used in both datasets which raises the number of hits between the stemmed tokens and the sentiment lexicon.

5.4.4. Fine-grained Sentiment Analysis Experiments

Fine grained sentiment analysis falls into the category of Multi-Label Classification (MLC) problems. In MLC problems, it is required to associated each input instance with a set of labels at the same time (Zhang and Zhou (2014)). Through our fine-grained SA experiments, we aim to investigate whether the previous preprocessing tasks (see Section 4.3) can achieve a better detection of 12 emotions including: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust in addition to “noEmotion” emotion-free texts. As each input instance can contain one or more emotions, therefore, we altered the sentiment classification phase in HCB Tw-STAR such that, the supervised and lexicon-based binary classifiers were replaced with One-Vs-All SVM which is a multi-lable classifier from Scikitlearn (Pedregosa et al. (2011)).

The Multi-Label Classifier MLC Tw-STAR was evaluated with Arabic, English and Spanish tweet datasets (see Table 5.1). For each dataset, different NLP tools and resources, in terms of stopwords, stemmers and lemmatizers, were employed to conduct the preprocessing tasks.

Table 5.18, Table 5.19 and Table 5.20 list the results obtained for each language when several single/combinations of preprocessing tasks were applied such that accuracy, macro average F-measure and micro average F-measure are denoted to as (Acc.), (Mic-F) and (Mac-F), respectively.

Table 5.18. Preprocessing impact on Arabic MLC

Preprocessing	**Acc.(%)**	**Mic-F(%)**	**Mac-F(%)**
Stop	38.0	50.9	36.7
Stem	43.1	55.9	42.4
Emo	41.4	54.3	39.0
Stem+Stop	43.4	56.4	43.5
Emo+Lem+Stop	43.4	56.1	41.5
Emo+ Stem+Stop	**44.9**	**58.0**	**44.4**

Considering Table 5.18, it could be noted that for the Arabic tweets, stemming using ISRI stemmer improved the accuracy by 5.1% percentage points compared to that scored by stopwords removal. Moreover, combining stemming with stopwords removal could further improve the micro F-measure as it increased from 55.9% to 56.4%. This is due to the fact that ISRI can handle wider range of Arabic vocabulary and, yet, returns a normalized form of the words having no stem, rather than leaving them unchanged (Taghva et al. (2005)).

Table 5.19. Preprocessing impact on English MLC

Preprocessing	**Acc.(%)**	**Mic-F(%)**	**Mac-F(%)**
Stop	44.6	57.7	42.9
Stem	44.9	58.0	44.3
Emo	45.9	58.8	43.4
Stem+Stop	46.2	59.3	45.8
Emo+Lem+Stop	**48.0**	**60.6**	46.1
Emo+ Stem+Stop	47.5	60.2	**46.6**

Unlike the Arabic dataset, Table 5.19 and Table 5.20 show that stemming had a different behavior when it was applied on both English and Spanish tweets. Compared to the accuracy achieved by stopwords removal, stemming has slightly increased the accuracy by 0.3% and 0.8% in English and Spanish datasets, respectively. This could be related to the insufficiency of the stemming algorithms, used in both porter2 and snowball stemmers, to handle informal English and Spanish tweets. Lemmatization by Treetagger, however, was a better choice to handle English and Spanish texts, as it forms a language-independent lemmatizer with implicitly POS tagger. Thus, combining emoji tagging with lemmatization and stopwords removal could achieve the best performances with a micro average F-measure of 60.6% and 52.3% for English and Spanish respectively.

Table 5.20. Preprocessing impact on Spanish MLC

Preprocessing	**Acc.(%)**	**Mic-F .(%)**	**Mac-F.(%)**
Stop	39.0	48.2	38.1
Stem	39.8	48.4	36.8
Emo	40.2	50.1	38.4
Stem+Stop	40.9	49.2	37.9
Emo+Lem+Stop	**43.1**	**52.3**	**41.3**
Emo+ Stem+Stop	42.8	51.8	40.1

Since the provided tweets were rich of emoji, emoji tagging could effectively contribute in improving the performance in all datasets especially when it was combined with the other best-performed tasks such as stem+stop in Arabic and lem+stop in both English and Spanish. This led to the best performances as the achieved micro F-measure was 58%, 60.2% and 52% in Arabic, English and Spanish datasets respectively.

The proposed multi-label HCB Tw-StAR was developed during our participation in SemEval-2018 shared Task 1. Considering that the official ranking will be calculated according to the achieved accuracy, therefore, our official submission included using the combination (Emoji tagging, stemming, stopwords removal) for Arabic, (Emoji tagging, lemmatization, stopwords removal) for english and (Emoji tagging+lemmatization+stopwords removal) for Spanish; where these combinations scored the best accuracy values for Arabic, English and Spanish, respectively.

Table 5.21 lists the official ranking of MLC Tw-StAR against the systems ranked first for each language where (Acc.), (Mic-F) and (Mac-F) refer to accuracy, micro F-measure and macro F-measure, respectively.

Table 5.21. The official ranking of MLC Tw-StAR

Language	Team	Rank	Acc.(%)	Mic-F .(%)	Mac-F.(%)
Arabic	EMA	1	**48.9**	**61.8**	**46.1**
	MLC Tw-StAR	3	46.5	59.7	44.6
English	NTUA-SLP	1	**58.8**	**70.1**	**52.8**
	MLC Tw-StAR	14	48.1	60.7	45.2
Spanish	MILAB-SNU	1	**46.9**	**55.8**	**40.7**
	MLC Tw-StAR	3	43.8	52.0	39.2

5.5. NEs and Preprocessing Impact on Sentiment Analysis

With considering the impact of NEs on supervised/lexcion-based investigated in Section 5.3 and single/combinations of preprocessing tasks evaluated in Section 5.4, it became possible to specify the best-performing preprocessing tasks for SA. Consequently, seeking for a further improvement in the SA performance, we decided which single/combinations of prerocessing to be used together with NEs as a novel preprocessing scheme. This was practically examined when HCB Tw-StAR shown in Figure (4.1) was used to perform SA of TEC, TAC, TSAC and AJGT datasets with novel schemes of NEs and preprocessing combinations included in the preprocessing phase.

For the supervised classifier of HCB Tw-StAR, and based on the results in Table 5.7, Table 5.8, Table 5.9 and Table 5.10, it was revealed that the preprocessing tasks: stemming, negation tagging, the combination (emoji tagging, stemming) and the combination of light stemming and negation were the best-performing preprocessing tasks for TEC, TAC, TSAC and AJGT datasets, respectively. Therefore, they were combined with NEs tagging to formulate novel preprocessing combinations to be applied on the studied datasets.

Table 5.22 reviews the impact of these novel preprocessing tasks on the sentiment classification using the supervised classifier and compares the obtained performances against baseline systems, where the used features, algorithm, precision, recall, F-measure and accuracy are referred to as (Feats.), (Alg.), (P.), (R.), (F1) and (A.), respectively.

Similarly, the preprocessing tasks that improved the lexicon-based sentiment classification performance for TEC, TAC, TSAC and AJGT datasets (Table

Table 5.22. Preprocessing+NEs with supervised HCB Tw-StAR

Data	Preprocessing	Feats.	Alg.	P.(%)	R.(%)	F1(%)	A.(%)
AJGT	**Alomari et al. (2017)**	bi	SVM	**92.1**	84.9	88.2	88.7
	LightS+Neg+NEs	uni+bi	NB	90.4	**89.1**	**89.7**	**90.3**
TEC	**Sayadi et al. (2016)**	uni+bi	SVM	67.0	71.0	63.0	71.1
	Stem+NEs	uni	NB	**75.7**	**74**	**74.2**	**75**
TAC	**Karmani (2017)**	morph.	Lex	63.0	72.9	67.3	72.1
	Neg+NEs	uni+bi	SVM	**87.4**	**86.6**	**86.6**	**86.7**
TSAC	**Mdhaffar et al. (2017)**	doc emb.	MLP	78.0	78.0	78.0	78.0
	Emo+Stem+NEs	uni	SVM	**92.8**	**92.9**	**92.8**	**93.0**

5.11, Table 5.12, Table 5.13 and Table 5.14), were integrated with NEs forming novel preprocessing combinations. Thus, in the lexicon-based classifier of HCB Tw-StAR, an input sentence is subjected to these new preprocessing tasks before looking its tokens up in the used lexicon.

Table 5.23 reviews the impact of the novel preprocessing tasks on the sentiment classification using the lexicon-based classifier and compares the obtained performances against baseline systems, where the used features, algorithm, precision, recall, F-measure and accuracy are referred to as (Feats.), (Alg.), (P.), (R.), (F1) and (A.), respectively.

Table 5.23. Preprocessing+NEs with Lexicon-based HCB Tw-StAR

Data	Preprocessing	Feats.	Alg.	P.(%)	R.(%)	F1(%)	A.(%)
AJGT	**Alomari et al. (2017)**	bi	SVM	**92.1**	84.9	88.2	88.7
	Neg+NEs	uni+bi	Lex	87.1	**88.4**	87.5	87.8
TEC	**Sayadi et al. (2016)**	uni+bi	SVM	67.0	**71.0**	63.0	**71.1**
	Stem+NEs	uni+bi	Lex	**68.1**	68.2	**67.8**	67.8
TAC	**Karmani (2017)**	morph.	Lex	63.0	72.9	67.3	72.1
	Stem+NEs	uni+bi	Lex	**74.0**	**74.0**	**74.0**	**74.0**
TSAC	**Mdhaffar et al. (2017)**	doc emb.	MLP	78.0	78.0	78.0	78.0
	Emo+Stem+NEs	uni+bi	Lex	**83.2**	**83.4**	**81.9**	**81.9**

It could be observed from Table 5.22 and Table 5.23 that, NEs form reliable indicators of Arabic sentiment especially when combined with the proper preprocessing tasks. This was practically examined with HCB Tw-StAR as its supervised classifier could remarkably outperform the baseline systems achieving the best F-measure values across all the datasets, While the lexicon-based classifier along with NEs and preprocessing schemes could outperform the baseline in the Tunisian datasets whereas in AJGT dataset, it achieved a close performance to the baseline with a difference in F-measure equals to 0.7%.

5.6. Evaluation Summary

The various experiments conducted using the proposed HCB Tw-StAR model have specified the role of preprocessing and NEs in the SA task of MSA/DA content. This enabled answering the first three research questions listed in Section 1.3 as follows:

RQ1: Are NEs reliable enough to infer the DA sentiment within hand-crafted feature-based SA models? And is it more likely to have a better SA performance for datasets rich of NEs?

- While the impact of NEs on the SA performance obtained by the proposed supervised SA classifier was inconclusive, it was revealed that including NEs within the lexicon-based classifier has remarkably improved the classification performances. Since, the context-ignorant manner adopted to associate NEs with their proper sentiments copes with the strategy followed by the lexicon-based method where context information are not considered while recognizing the sentiment.

- Based on the strategy followed by the developed algorithm for NEs sentiment detection, the consistency of the dataset can lead to an accurate NEs sentiment detection. Since, in corpora with a good degree of consistency, training and test sets tend to have overlapped NEs used within the same domain and thus an unanimous could be found for a specific NE. Hence, the quality of the SA performance depends on the accurate sentiment annotation of NEs more than the number of NEs in a dataset.

RQ2: Which combination of preprocessing tasks can lead to an improved performance in hand-crafted features-based SA models?

- For the supervised classifier of HCB Tw-StAR, it was revealed that the preprocessing tasks: stemming, negation detection and tagging, the combination (emoji tagging, stemming) and the combination (light stemming, negation) were the best-performing preprocessing tasks. On the other hand, Stemming, negation detection and te combination (emoji, stemming) were found of the best impact on the sentiment classification performance conducted by the lexiocn-based classifier.

RQ3: Would the sentiment classification performance improved if NEs were included together with specific combinations of preprocessing tasks?

- Further improvement in the sentiment classification performance could be obtained when integrating NEs with specific single/combinations of preprocessing tasks. These tasks were selected carefully according to their impact on the SA performance obtained by supervised and lexicon-based classifiers contained in HCB Tw-STAR.

5.7. Conclusion

In this chapter, we explored the various experiments carried out using HCB Tw-StAR to perform coarse-grained and fine-grained SA of DA/multi-lingual datasets. HCB Tw-StAR was found of an efficient performance with non-Arabic datasets (English, Spanish and Turkish); as the presented novel combinations of preprocessing tasks have improved the classification performances. On the other hand, for Arabic datasets, it has been revealed that, both NEs and preprocessing could favorably affect the classification performances, without the need for dialect-specific morphological analyzers tools and with the least dependency on dialectal resources such as the lists of stopwords and negation words. Moreover, combining the best-performing preprocessing task/tasks along with NEs features could further improve the performances. Throughout this chapter, we proposed potential, practical solutions to overcome the variances among the Arabic dialects based on universal text components such as NEs and novel combinations of preprocessing tasks. Nevertheless, the hand-crafted features produced by the different preprocessing tasks are often associated with the high dimensionality issue of feature vectors (Section 3.4). Therefore, it is better to dispense the preprocessing tasks and adopt the new type of features, known as embeddings, which have a low dimensionalty, yet, can express the sentiment of different Arabic dialects efficiently. This will be discussed in the next chapter where we will present novel embedding features within our second SA model Neu Tw-StAR.

6. NEU TW-STAR MODEL

This chapter introduces the embeddings-based SA model developed to target MSA/DA social media contents. First, a detailed review of the main research problem is provided. Then, a comprehensive description of the Neu Tw-StAR is given along with highlighting the developed embedding features, composition function and the implementation architecture adopted by the proposed model.

6.1. Dialectal Arabic: To Respect or Disrespect The Syntax

With the recent rapid growth of Arabic language across social media platforms along with the challenging morphological and high inflectional nature of Modern Standard Arabic (MSA) and Dialectal Arabic (DA), Arabic Sentiment Analysis (ASA) is gaining an increased interest from the Natural Language Processing (NLP) research community. According to the used features, existing ASA systems can be classified into either (a) hand-crafted-based systems where SA models employ linguistic and lexical features usually generated by morphological analyzers and semantic resources (Sayadi et al. (2016); Alomari et al. (2017); El-Beltagy and Ali (2013); Abdulla et al. (2013)) or (b) text embeddings-based systems which adopt text distributed representations the so-called word/sentence embeddings, where the sentence embeddings are composed out of its constituent word embedding vectors using one of the composition models(Altowayan and Tao (2016); Dahou et al. (2016); Gridach et al. (2017); Mdhaffar et al. (2017)).

Composition models aim to construct phrase/sentence embeddings based on its constituent word embeddings and structural information (Gormley et al. (2015)). Two main types of these models can be recognized: (a) Ordered or syntactic models where the order and linguistic/grammatical structure of the input words do count while constructing the phrase/sentence vector and (b) Unordered models in which the word embeddings are combined irrespective of their order using algebraic operations. Sum of Word Embeddings (SOWE), Average (Avg), minimum (Min), maximum (Max) and multiplication functions are examples of such models (Mitchell and Lapata (2010)).

Based on the theory “you shall know a word by the company it keeps” (Firth (1957)), context words along side their syntactic properties were considered essential to build effective word embeddings able to infer the semantic/syntactic similarities among word, phrases or sentences. Consequently, most of the recently-developed SA systems adopted Recursive Neural Networks (RecNNs) or Long Short Term Memory neural models (LSTMs) in which ordered composition models are employed to grasp the syntactic and linguistic relations between the words (Socher et al. (2013); Al Sallab et al. (2015)). These systems usually require more training time to learn words’ order-aware embeddings due to the high computational complexity consumed at each layer of the model (Iyyer et al. (2015)). However, the embeddings resulting from ordered constitutionality might not be sufficient to handle the Arabic dialects that have a free word order and varying syntactic/grammatical nature (Brustad (2000); Chiang et al. (2006)). While MSA can be described as a Verb-Subject-Object (VSO) or Subject-Verb-Object (SVO) language, the Arabic dialects go beyond that, for instance, the dialectal (Levantine) sentence investigated in Table 6.1 refers to the meaning *I liked this idea* and can be represented by several word orders: VSO, SVO, OSV and OVS, while retaining the meaning and the sentiment .

Table 6.1. Free word order of dialectal Arabic

Order 1	هالفكرة O	انا S	حبيتا V
Order 2	هالفكرة O	حبيتا V	انا S
Order 3	حبيتا V	انا S	هالفكرة O
Order 4	انا S	حبيتا V	هالفكرة O

On the other hand, the Arabic dialects show phonological, morphological, lexical, and syntactic differences such that the same word might infer different syntactic information across different Arabic dialects (Brustad (2000)). To clarify that, Table 6.2 reviews how the word “ماشي” has several Part Of Speech (POS) tags, multiple meanings and different sentiments across three Arabic dialects.

Table 6.2. Syntactic differences across the Arabic dialects

Dialect	**Sentence**	**Word**	**POS**	**sentiment**
Levantine	الوضع **ماشي** الحال *The situation is **okay***	ماشي *okay*	adjective	positive
Moroccan	نحن **ماشي** سعداء *We are **not** happy*	ماشي *not*	negation	negative
Egyptian	كنت **ماشي** فاتجاه البيت *I was **walking** towards home*	ماشي *walking*	verb	neutral

To handle such informality of DA, unordered composition models can replace the syntactic composition functions to construct sentence/phrase embeddings regardless of the order and the syntax of the context's words. Nevertheless, when coming to the sentiment analysis task, sentence/phrase embeddings that are merely composed and learned based on the context words do not always infer the sentiment accurately. This is due to the fact that some words that have contradict sentiments might be mentioned within identical contexts; This leads to map opposite words close to each other in the embedding space. To clarify that, both sentences in Example 6.1 and Example 6.2 contain the same context words organized in the same order; yet the first sentence implies a positive polarity while the second has a negative sentiment since the words "ممتع" (*interesting*) and "ممل" (*boring*) are antonyms.

Example 6.1 هالفيلم كان ممتع بشكل ما بينوصف
This movie was incredibly interesting

Example 6.2 هالفيلم كان ممل بشكل ما بينوصف
This movie was incredibly boring

One way to address this issue is to learn the embeddings from sentiment-annotated corpora such that the sentiment information is incorporated along with the contextual data within the composed embeddings during the training phase. This was examined with the English language as Tang et al. (2014) presented a supervised neural model in which both sentiment and the syntactic relations were integrated in the loss function yielding sentiment-specific word embeddings (SSWE). In that study, Min, Max and Avg composition functions were applied together to compose the embeddings. The learned embedding vectors were then fed to classical supervised classifiers where a better sentiment classification could be achieved compared to word2vec (Mikolov et al. (2013)) and hand-crafted features-based systems.

In another study by Iyyer et al. (2015), an English SA model called Deep Averaging Neural network (DAN) was presented. DAN was implemented as a neural network with two hidden layers. Based on the assumption that unordered composition functions can efficiently encode sentiment within the composed embeddings, DAN learned and produced n-gram embedding features for the SA task through pairing between the Avg unordered composition function and supervised learning. The experimental study indicated the ability of the embeddings learned by DAN to rival those resulting from ordered syntactic models. Where DAN achieved a comparable performance against RecNNs and CNNs-Multi Channel (CNN-MC) models in which both semantic and syntactic information were encoded using ordered composition functions.

While several recent ASA systems considered the syntactic information in the composed embeddings generated for MSA (Al Sallab et al. (2015)), other models used pretrained or unsupervised unordered word/document embeddings as features to mine the sentiment of MSA/DA content (Altowayan and Tao (2016); Gridach et al. (2017)). However, mining the sentiment of DA using syntax-aware ordered embeddings might be ineffective especially with the drastic differences between Eastern and Western Arabic dialects (Zaidan and Callison-Burch (2014)). In addition, for the SA task, the embeddings learned from unlabeled data are not as discriminating as those learned with sentiment information integrated in the embedding vectors (Tang et al. (2014)). This evokes the need to provide a sentiment-specific, dialect-independent embeddings with which the gap resulting from the differences among Arabic dialects can be bridged. One way to do that is by training an Arabic SA framework with sentence embedding features that ignore the words' order and contextual information i.e. the syntactic structure and focus on the semantic and sentiment information.

Inspired by Iyyer et al. (2015), we hypothesize that a DA sentence, with free word order and various syntax, can be better represented if their constituent word embedding vectors are composed using an unordered composition function. On the other hand, motivated by Tang et al. (2014), we assume that these composed embedding features can be more expressive for the SA task if they are learned with the sentiment information considered. Therefore, this study aims to present a SA framework (Neu Tw-StAR) whose features are n-gram embeddings learned from labeled data (sentiment-specific) and composed via the additive unordered composition function (syntax-ignorant) SOWE whose efficiency to capture the semantic information was proved in (White et al. (2015)). The embeddings composition and the sentiment learning processes were conducted within Neu Tw-StAR which was constructed as a shallow feed-forward neural network of single hidden layer.

6.2. Neu Tw-StAR Model Description

As seeking to answer the question: Can a shallow neural model, trained with embeddings specifically formulated to target the dialectal content, rival more complicated neural architectures?, therefore, Neu Tw-StAR model was implemented as a feed-forward neural network in which sentiment-specific, syntax-ignorant n-grams embeddings are composed via SOWE function, and learned in a supervised manner. The generated n-gram embeddings were then employed as discriminating features to predict the positive/negative sentiment of DA contents.

As it is shown in Figure 6.1, given the input negative training tweet in Example 6.3, a sequence of six n-grams is generated by going through the tweet using a fixed-size sliding window.

Example 6.3

ويندوز الجديد لما بيعمل ابديت بيخرب ما بيحسن نصيحة لا حدا ينزل الابديت

upon update, the new version of Windows makes things worse rather than better;
as an advice do not update your system

Having the n-grams generated, each of which is associated with the polarity [0,1] which represents the negative polarity of the previous tweet. Later, n-grams are, fed to Neu Tw-StAR model where the correspondent embeddings for their constituent words are constructed at the embeddings layer, composed and formulated as n-gram embeddings at Lambda layer and learned as sentiment features while being forwarded through Neu Tw-StAR layers. Finally, n-gram embeddings are exploited, at the output layer, to recognize the sentiment of the input tweet. Below is a detailed

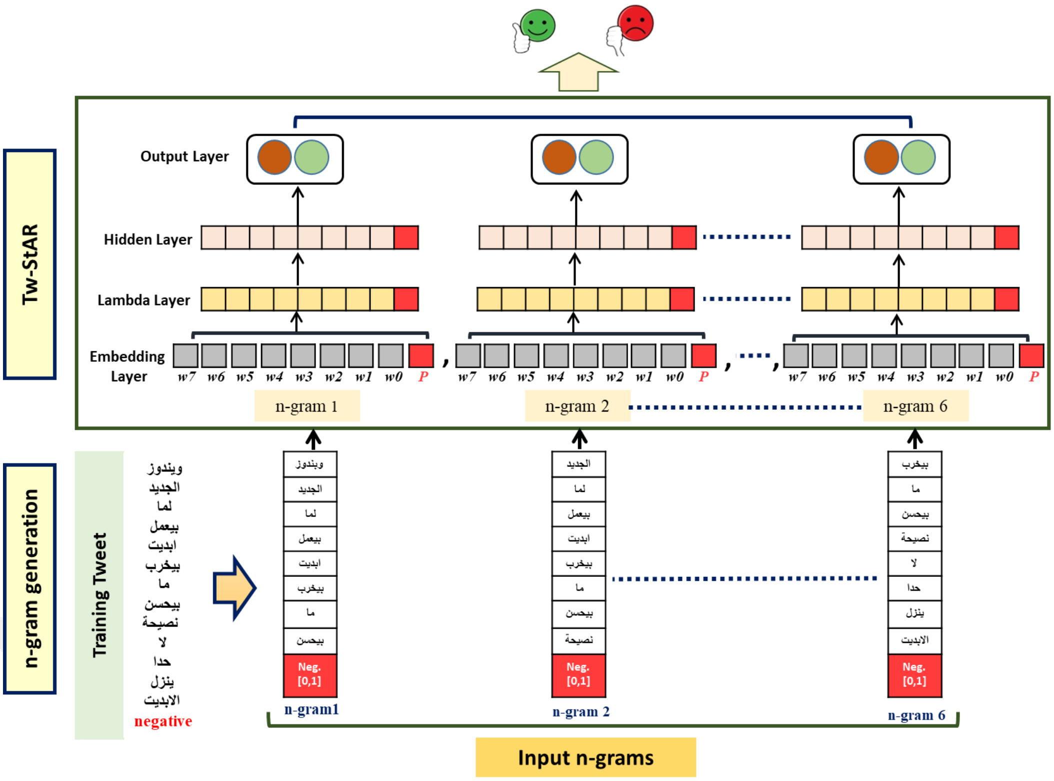


Figure 6.1. Neu Tw-StAR sentiment analysis model

description of each layer, where the model's notations are listed in Table 6.3.

The Embedding layer is responsible of projecting words in the input into their corresponding dense vector representations. Given the input sentences, in order to handle their varying lengths, each sentence S of l words was formulated as a sequence of fixed-length n-grams generated using a sliding window of a specific size C. Unlike (Tang et al. (2014)) who used corrupted input n-grams, where an input n-gram missing a word is being fed to the model aiming to learn the syntactic information of context words, whole n-grams were fed to the embedding layer of our model such that each n-gram is accompanied with the sentiment label of the sentence from which it was derived. Having the n-grams prepared and accompanied with the vector representing the sentiment label ([1,0] for positive and [0,1] for negative) of the sentence from which they were derived, their constituent words were mapped into the corresponding embeddings using the embeddings weight matrix M$\in$ R$^{|V| \, xd}$ where $|V|$ is the vocabulary size and d denotes the embedding dimension.

The Embedding layer is responsible of projecting words in the input into their corresponding dense vector representations. Given the input sentences, in or-

der to handle their varying lengths, each sentence *S* of *l* words was formulated as a sequence of fixed-length n-grams generated using a sliding window of a specific size *C*. whole n-grams were fed to the embedding layer of our model such that each n-gram is accompanied with the sentiment label of the sentence from which it was derived. Having the n-grams prepared and accompanied with the vector representing their relevant sentiment label ([1,0] for positive and [0,1] for negative), their constituent words were mapped into the corresponding embeddings based on the embeddings weight matrix $M \in R^{|V| xd}$ where $|V|$ is the vocabulary size and *d* denotes the embeddings dimension.

The weights of the embedding matrix M were initialized using Glorot uniform initialization (Glorot and Bengio (2010)) then optimized while training the model. It should be noted that, we could not use pretrained word embeddings for initialization, as the available Arabic pretrained word embeddings from Zahran et al. (2015) and (Al-Rfou et al. (2013)) were generated using MSA/Egyptian corpora which can lead to out-of-vocabulary (OOV) issues especially when dealing with the Tunisian and Moroccan content where less common words with MSA/Egyptian do exist. Thus, for a single fixed-length n-gram containing a sequence of words $\{w_i, w_{i+1}, w_{i+2}, ..., w_{i+C-1}\}$, each word w_i is represented by a unique integer index $i \in$ [0,V] and stored as a one-hot vector vec_i whose values are zero in all positions except at the i-*th* index. To obtain the embedding vector v_i of a word w_i, its one-hot vector vec_i is multiplied by the embedding matrix M as in equation (1)

$$v_i = vec_i * M \in R^{1 \text{ x } d} \tag{1}$$

As each row of the embedding matrix M denotes the dense embeddings of a specific word in the vocabulary, multiplying the one-hot vector of each word in the input by the embedding matrix M, will essentially select one of M rows that corresponds to the embeddings of this word.

The resulting word embeddings generated for each word of the input n-gram were then combined using the compositional model SOWE. This was performed by the next linear layer Lambda, where an element-wise sum is applied over the word embedding vectors obtained from the previous layer. Hence, the output of lambda layer is a single embeddings vector $O_{lambda} \in R^{1x|d|}$ resulting from the element-wise sum of the embeddings vectors produced by the embedding layer and correspond to the input words that denote a single n-gram and contained in a window of a

Table 6.3. Notations used of Neu Tw-StAR model

Symbol	Description
C	sliding window's size
w	input word
i	integer index of a word
M	weight embedding matrix
$\|V\|$	vocabulary size
d	embedding dimension
$\mathrm{vec_i}$	one-hot vector of w_i
v_i	embedding vector of w_i
O_{lambda}	output of lambda layer
W_{hl}	hidden layer's weights
b_{hl}	hidden layer's biases
O_{hl}	output of the hidden layer
h_σ	hard sigmoid activation function
$\hat{y}$	predicted sentiment label
y	the gold sentiment label
k	number of the classes
θ	model's weights and biases

fixed-size C :

$$O_{lambda} = \sum_{i=1}^{C} v_i \in R^{1\mathrm{x}d} \tag{2}$$

In the subsequent hidden layer (*hl*), the output from the previous layer O_{lambda} is subjected to a linear transformation using the weights matrix $W_{hl} \in \mathrm{R}^{d\mathrm{x}2}$ and biases $b_{hl} \in \mathrm{R}^{1\mathrm{x}2}$:

$$O_{hl} = f(O_{lambda} * W_{hl} + b_{hl}) \in R^{1\mathrm{x}2} \tag{3}$$

Where W_{hl} and b_{hl} form the model's parameters that are learned and optimized during the training process and f refers to the activation function that introduces non-linear discriminative features to our model. Here, we have used Hard sigmoid activation function h_σ (Courbariaux et al. (2016)) identified in equation (4). Hard sigmoid is an approximation of the standard sigmoid activation function; it is defined as a piece-wise function whose output are very similar to the traditional sigmoid, however, it is computationally cheaper which leads to a smarter model with the learning process accelerated in each iteration (Gulcehre et al. (2016)).

$$h\_\sigma(x) = clip((x+1)/2, 0, 1) = max(0, min(1, (x+1)/2)) \tag{4}$$

where clipping bounds are -2.5 and 2.5 and its derivative is given as below:

$$h\hat{\_}\sigma(x) = \begin{cases} \mathit{0} & \mathit{if} \quad x < \mathit{-2.5} \quad \mathit{or} \quad x > \mathit{2.5} \\ \mathit{0.2} & \mathit{otherwise} \end{cases} \tag{5}$$

Finally, the output O_{hl} resulting from the hidden layer is forwarded into the output layer (*Ol*) where a softmax function is applied to induce the estimated probabilities for each output label (positive/negative) of a specific n-gram. Where each n-gram is accompanied with the predicted two dimensional label [1,0] denoting positive or [0,1] indicating negative.

$$\hat{y} = softmax(O_{hl}) \in R^{1x2} \tag{6}$$

Softmax selects the maximum score among the two predicted conditional probabilities to denote positive or negative polarity of an input n-gram where the distribution of the form [1,0] was assigned for positive while [0,1] distribution form was adopted for negative. Thus, if the gold sentiment polarity of an n-gram is positive, the predicted positive score should be higher than the negative score while if the gold sentiment polarity of a word sequence is negative, its positive score should be smaller than the negative score. Then, to decide the polarity of the whole sentence, the predicted positive scores and negative scores of n-grams are summed then each of which is divided by the number of the n-grams contained in this sentence resulting two values representing the potential positive and negative scores of the input sentence. The final sentence polarity is, thus, decided according to the greater among these two values. Cross-entropy loss between gold sentiment distribution and predicted distribution was adopted such that the loss function of the model:

$$J(\theta) = -\sum_{k=\{0,1\}} y_k \log \hat{y}_\mathrm{k} \tag{7}$$

Where y $\in$ R^2 is the gold sentiment value represented by a one-hot vector, $\hat{y}$ is the sentiment predicted by the model while θ refers to the parameters (weights and biases) of the model to be learned and optimized during the training process.

6.3. Training details and Model's Parameters

The key hyper parameters of the proposed model are the sliding window size *C* which defines the n-gram scheme to be adopted and the embeddings dimension *d*. We have selected both parameters' values empirically during the model tuning period. According to (Socher et al. (2013)), narrower windows lead to better performance in syntactic tests while wider ones score a better performance in semantic tests. Therefore, we have tested different quite large values of the sliding window size *C* to select the size that achieves the best evaluation measures. Similarly, the emdeddings dimension size was selected empirically among several examined values.

For efficient training, Glorot uniform initialization by (Glorot and Bengio (2010)) was used to set the weights of the embeddings layer. Glorot initialization makes sure the weights are "just right" across the model's layers, keeping the signal in a reasonable range of values, through avoiding too massive and too tiny weight values with which learning could not be useful. This is achieved by drawing samples from a uniform distribution within *-limit, limit* where *limit* is defined in equation 8.

$$limit = \sqrt{\frac{6}{fan\_in + fan\_out}} \tag{8}$$

Where *fan_in* is the number of input units in the weight tensor, while *fan_out* is the number of output units in the weight tensor.

To train the proposed neural network, the back-propagation algorithm with Adaptive Moment estimation (Adam) stochastic optimization method (Kingma and Ba (2014)) has been used. Adam optimizer combines the early optimization speed of Adagrad (Duchi et al. (2011)) with the better later convergence of various other methods like Adadelta (Zeiler (2012)) and RMSprop (Tieleman and Hinton (2012)). This is done through calculating learning rates and storing momentum changes for each model's parameter separately. The parameters update rule in Adam (see Equation 11) uses the first moment m_t (the mean) that represents the decaying average of the past gradients computed by Equation 9 in addition to the second moment v_t (the uncentered variance) calculated by Equation 10 which refers to the decaying average of past square gradients.

$$m_{\text{t}} = \beta_1 m_{\text{t-1}} + (1 - \beta_1) g_{\text{t}} \tag{9}$$

$$v_{\text{t}} = \beta_2 v_{\text{t-1}} + (1 - \beta_2) {g_{\text{t}}}^2 \tag{10}$$

Where β_1 and $\beta_2 \in$ [0,1] control the exponential decay rates of the moving averages of the first current moment m_t and the second current moment v_t while g_{t} denotes the gradients at timestep t.

$$\theta_{\mathrm{t+1}} = \theta_{\mathrm{t}} - \frac{\eta}{\sqrt{\hat{v}_{\mathrm{t}}} + \epsilon}\hat{m}_{\mathrm{t}} \tag{11}$$

Where θ_{t} denotes the current parameters, $\theta_{\mathrm{t+1}}$ are the updated parameters, η refers to the learning rate, ϵ is a smoothing term that avoids division by zero while $\hat{m}_{\mathrm{t}}$ and $\hat{v}_{\mathrm{t}}$ represents the bias-corrected values of m_t and v_t respectively which are calculated as follows:

$$\hat{m}_{\mathrm{t}} = \frac{m_{\mathrm{t}}}{1 - {\beta_1}^t} \tag{12}$$

$$\hat{v}_{\mathrm{t}} = \frac{v_{\mathrm{t}}}{1 - {\beta_2}^t} \tag{13}$$

To handle the over-fitting issue, Dropout was used as a regularization mechanism. The value of the dropout parameter was selected empirically during the model's tuning period.

6.4. Conclusion

In this chapter, we presented our embedding features-based Neural model Neu Tw-StAR. Based on the varying syntactic nature and the free word order of DA, we presented our unordered, syntax-ignorant n-gram embeddings as sentiment features of DA. Throughout this chapter, we described how the proposed embeddings were generated based on the unordered additive composition function SOWE and learned within our shallow neural model Neu Tw-StAR. Adopting such a shallow architecture along with unordered compositionality, we aim to produce expressive sentiment embedding features for the DA contents and to provide a robust implementation with less training time compared to deep learning models. These concepts will be practically investigated through a variety of experiments introduced in the next chapter.

7. NEU TW-STAR EXPERIMENTS AND EVALUATION

This chapter reviews the experiments conducted using Neu Tw-StAR to mine the sentiment in Eastern/Western DA datasets. The following sections include an investigation of the efficiency of the proposed model in terms of the developed n-gram embedding features, the employed SOWE composition function, the consumed training time and the implementation shallow neural architecture. This is done by exploring the performances of Neu Tw-StAR against those obtained whether by the stat-of-the-art embedding methods, using other unordered composition functions, or with several deep neural architectures adopted for implementation. By the end of this chapter, we provide a comprehensive evaluation of the proposed model highlighting the merits it introduces to support the specificity of DA.

7.1. Experimental Setup

Aiming to examine the efficiency the proposed model across various Arabic dialects, and for different sizes of data, we employed Neu Tw-StAR to mine the sentiment embedded in seven benchmark datasets having Eastern (Syrian, Jordanian, Egyptian, Gulf) or Western (Tunisian, Moroccan) Arabic dialectal content. It should be noted that, due to the lack of embeddings-based ASA systems, it was not always possible to compare our model with deep neural ASA baselines; Since, for most of the studied datasets, the available baseline systems are hand-crafted features-based. Therefore, in order to enable a comprehensive evaluation of the proposed model, we developed our own deep neural systems to perform SA of the tackled datasets (see Section 5.2) and considered them as embeddings-based baselines. All the experiments were conducted using Tensorflow within Google Colab cloud service using the processing power of NVIDIA Tesla K80 GPU.

7.1.1. Datasets

Besides the Arabic datasets, of positive/negative polarity, described in Section 5.2 and listed in Table 5.1; Neu Tw-StAR was applied on the following datasets:

- Jordanian Egyptian Gulf (JEG): a medium-sized dataset investigated in (Altowayan and Tao (2016)). It combines 4,294 positive/negative tweets from three datasets of MSA/DA content including: (a) Jordanian: Artwitter (Abdulla et al. (2013)), (b) Egyptian: ASTD (Nabil et al. (2015)) and (c) Gulf: QCRI (Mourad and Darwish (2013)).
- Moroccan Election dataset (MEC): refers to a large-sized social/political dataset of 10,253 positive/negative tweets, collected by Elouardighi et al. (2017) during the Moroccan elections in 2016.
- Tweets Emoji Arabic Dataset (TEAD): A large scale dataset combines tweets from multiple domains (Abdellaoui (2018)). It is composed of 555,924 positive/negative tweets written in several Eastern and Western Arabic dialects.

Adopting the same divisions of training, development and test sets for each dataset, we review the detailed statistics of these sets in Table 7.1, where Jor, Egy, Gul, Train, Dev and Avg.S.L. refer to Jordanian, Egyptian, Gulf, training set, developing set and average lengths of sentences in a dataset, respectively.

Table 7.1. Statistics of Neu Tw-StAR evaluation datasets

Dialect	Dataset	size	Train	Dev	Test	#words	Avg-SL
Jordanian	AJGT	1,800	1,152	288	360	5,933	9
Jor/Egy/Gul	ArTwitter	1,979	1,266	317	396	6,083	9
	JEG	4,294	2,747	687	860	16,455	12
Tunisian	TEC	3,043	1,947	487	609	9,457	11
	TSAC	7,366	4,680	1,170	1,516	15,005	10
Moroccan	MEC	10,253	6,561	1,641	2,051	31,546	15
DA	TEAD	555,924	355,792	1,641	88,948	111,184	13

7.1.2. Hyper Parameters Adjustment of Neu Tw-StAR Model

The hyper parameters (C, d) of Neu Tw-StAR were assigned empirically. Among several window sizes ranging from 4 to 10, a window size value of 8 was adopted as it produced the best F-measure in all datasets using the validation Dev dataset (see Table 7.2). Consequently, each input sentence is represented by a set of 8-grams to be fed to the model. Similarly, upon examining three embedding dimension sizes equal to 50, 100 and 150, and several dropout rates ranging from 0.2 to 0.5, d=100 and dropout=0.2 were adopted, since these values scored the best F-measure during the model's tuning period.

Table 7.2. F-measure values (%) with dev sets for different window sizes

Dataset	**C=4**	**C=5**	**C=6**	**C=7**	**C=8**	**C=9**	**C=10**
AJGT	80.0	79.1	79.1	79.9	**82.0**	80.8	76.7
ArTwitter	81.4	82.7	82.7	83.0	**83.3**	82.3	81.5
JEG	71.3	72.4	73.4	73.4	**73.8**	73.3	72.5
TEC	84.7	85.1	87.6	**87.9**	**87.9**	83.6	81.2
TSAC	71.1	81.9	86.1	85.9	**86.6**	86.5	86.3
MEC	67.5	66.3	63.9	**68.6**	**68.6**	67.1	66.5

7.2. Neu Tw-StAR Evaluation Experiments

With the objective of answering the research questions listed in Section 1.3, the efficiency of Neu Tw-StAR, as a SA model of several Arabic dialects, was evaluated through the conduction of various experiments considering several aspects: (a) Training embedding features, (b) Embeddings composition function, (c) Implementation neural architecture and (d) Consumed training time. In the following sections, we review and analyze the obtained results for each experiment where we adopted the evaluation measures explained in Section 2.5.

7.2.1. Syntax-Ignorant n-gram Embeddings Evaluation

The efficiency of our n-gram embeddings, composed by SOWE, was evaluated against word embeddings (word2vec) and document embeddings (doc2vec). To conduct a fair comparison, word2vec (Mikolov et al. (2013)) and doc2vec (PV-DBoW/PV-DM) (Le and Mikolov (2014)) algorithms were trained on each of the studied datasets with sentiment labels included in the training process. In addition, the training parameters such as the window size and embedding dimensions were unified across the evaluated embedding methods: word2vec, doc2vec (PV-DBoW/PV-DM) and Neu Tw-StAR. It should be noted that, within doc2vec we can recognize two mapping methods: (a) Distributed Bag of Words (DBoW) which learns and composes the sentence embeddings regardless of the order of words; and (b) Distributed memory (DM) that follows the CBOW mechanism, as it considers the words' order while learning the sentence embeddings vector (Le and Mikolov (2014)).

Table 7.3. Neu Tw-StAR with n-gram, word2vec and doc2vec embeddings

Dataset	Embeddings	P. (%)	R. (%)	F1 (%)	Acc. (%)
AJGT	word2vec	72.1	73.1	71.2	71.4
	doc2vec (DM)	54.4	54.4	51.9	51.9
	doc2vec (DBoW)	58.0	57.7	54.4	54.4
	n-gram (Neu Tw-StAR)	**82.5**	**83.2**	**82.8**	**83.3**
ArTwitter	word2vec	72.0	71.9	71.9	72.0
	doc2vec (DM)	61.2	60.7	60.1	60.4
	doc2vec (DBoW)	63.1	60.6	58.2	59.9
	n-gram (Neu Tw-StAR)	**85.4**	**84.9**	**84.8**	**84.9**
JEG	word2vec	59.3	59.2	59.2	59.4
	doc2vec (DM)	58.5	57.9	57.4	58.4
	doc2vec (DBoW)	61.2	59.4	58.2	60.2
	n-gram (Neu Tw-StAR)	**75.8**	**74.3**	**74.3**	**74.8**
TEC	word2vec	62.6	59.7	58.4	61.9
	doc2vec (DM)	65.6	59.3	56.4	62.2
	doc2vec (DBoW)	62.9	58.9	56.7	61.4
	n-gram (Neu Tw-StAR)	**87.4**	**88.4**	**87.8**	**88.2**
TSAC	word2vec	78.0	77.2	77.4	78.2
	doc2vec (DM)	61.0	58.3	57.2	61.7
	doc2vec (DBoW)	55.9	54.1	52.1	58.0
	n-gram (Neu Tw-StAR)	**86.2**	**86.3**	**86.2**	**86.5**
MEC	word2vec	63.6	64.0	63.8	69.1
	doc2vec (DM)	74.7	65.0	66.4	76.6
	doc2vec (DBoW)	60.4	56.6	56.4	69.3
	n-gram (Neu Tw-StAR)	**76.2**	**71.2**	**72.8**	**79.2**
TEAD	word2vec	70.3	60.8	61.4	74.3
	doc2vec (DM)	69.5	60.3	60.8	74.0
	doc2vec (DBoW)	**73.5**	61.1	61.7	**75.3**
	n-gram (Neu Tw-StAR)	67.2	**61.9**	**62.7**	73.3

Having the word embeddings, document embeddings and n-gram embeddings generated for each of the studied corpora, they were used one by one as features to train Neu Tw-StAR on recognizing the sentiment of the datasets in Table 7.1. To train our model with word2vec and doc2v embeddings, the embedding layer in Neu Tw-StAR was replaced with the embeddings produced by word2vec and both variants of doc2vec, respectively. The learned word2vec/doc2vec embeddings were then passed through the shallow architecture of Neu Tw-StAR. Table 7.3, lists the sentiment classification performances achieved using syntax-ignorant n-grams composed by SOWE, word vectors mapped by word2vec and sentence vectors produced by doc2vec (PV-DBoW/PV-DM) for all datasets.

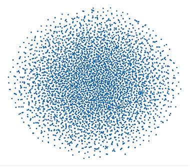
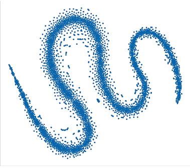
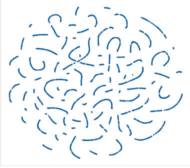
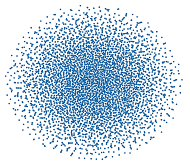
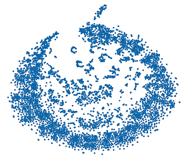
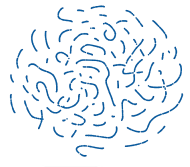
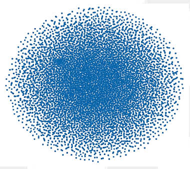
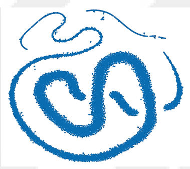
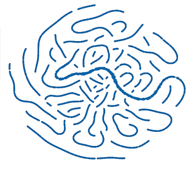
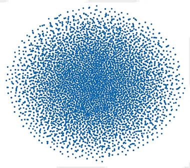
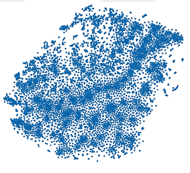
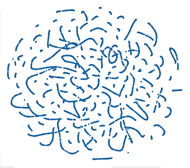
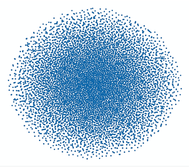
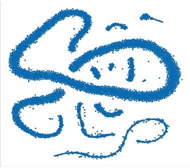
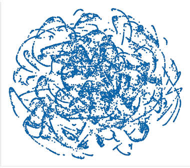
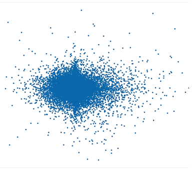
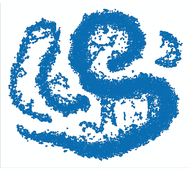
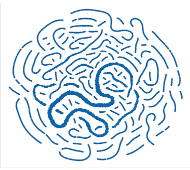
The results in Table 7.3 suggest the outperformance of the proposed embeddings over those generated by word2vec and doc2vec models with a significant margin in F-measure values for most datasets. The best F-measure value was achieved for TEC dataset with a value of 87.8% compared to 58.4%, 56.4% and 56.7% scored by word2vec, doc2vec (PV-DM) and doc2vec (PV-DBoW), respectively. This could be explained by the ability of SOWE to accurately capture the semantic information along with the synonymous relations among words coping with the claim stated in (White et al. (2015)). This was further emphasized through examples from the visualization maps provided in Section 7.2.2. In addition, for JEG dataset that combines three different dialects, the F-measure obtained using n-gram embeddings increased by 15.1%, 16.9% and 16.1% compared to word2vec, doc2vec (PV-DM) and doc2vec (PV-DBoW), respectively. This indicates how the proposed embeddings can address the differences among various dialects through ignoring the syntactic structure and word order, and focusing on the semantic relations based on the ability of SOWE to efficiently enrich the composed n-gram embeddings with semantic/synonymous regularities.

On the other hand, it can be seen from Table 7.3 that for datasets, having an MSA-dominated content such as MEC, doc2vec (PV-DM), which takes the words' order into account, performed significantly better than word2vec and doc2vec (PV-DBoW). Indeed, the achieved accuracy for MEC dataset with the embeddings learned by doc2vec (PV-DM) was 76.6% compared to 69.1% and 69.3% scored by word2vec and doc2vec (PV-DBoW), respectively.

7.2.2. Syntax-Ignorant n-gram Embeddings Visualization

Aiming to inspect the performance of the developed n-gram embeddings thoroughly, we visualized the learned n-gram embedding vectors learned by Neu Tw-StAR and compared them towards word embedding vectors of word2vec and document embeddings of doc2vec (PV-DBoW). This was done by projecting the embedding vectors into a two dimensional space using the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique (Maaten and Hinton (2008)).

Table 7.4. Embeddings maps of word2vec, doc2vec and Neu Tw-StAR

Dataset	Word2vec	Doc2vec	Neu Tw-Star
AJGT			
ArTwitter			
JEG			
TEC			
TSAC			
MEC			

Considering the figures in Table 7.4, a clustering behavior of the words that compose n-grams or document embeddings could be observed in both Neu Tw-StAR and doc2vec (PV-DBoW) models. In word2vec model, however, word vectors tend to spread sparsely in the embeddings space. This was reflected on the ability of the word embeddings as expressive SA features. To clarify that, when exploring TSAC word map, we noticed that pure Tunisian dialectal words like “إنحبوك” (*we love you*), “يعجبنا” (*we like it*), “باهي” (*good*), “نحييك” (*we praise you*), which are all bear positive sentiments, were mapped by Neu Tw-StAR model close to each other in the embeddings space. However, when looking to the representations created for the same dataset by doc2vec (PV-DBoW), we come through the words “إنحبوك” (*we love you*), “مذهلة” (*magnificent*) and “هايلة” (*excellent*), which all refer to a positive sentiment, yet they are mapped close to the negative words “ممسطها” (*how boring*), “ماسطين” (*dull*) and “ خامج” (*a dirty man*) in the embeddings space.

7.2.3. SOWE Composition Function Evaluation

According to White et al. (2015), in the context of sentence semantic similarity NLP task, SOWE was proved to be the best-performing composition function in expressing and encoding the semantic information within the phrase/sentence embedding vectors. Based on that, we investigated how SOWE would perform for the sentiment analysis task, compared to another unordered composition function: Avg, which was used in (Iyyer et al. (2015)). For this purpose, we composed our n-gram embeddings by SOWE then using Avg function. Afterwards, both embedding variants were used to train Neu Tw-StAR model.

Table 7.5. AVG, SOWE impact on SA of the dialectal datasets.

Dialect	**Dataset**	**SOWE**		**AVG**	
		F(%)	**Acc. (%)**	**F (%)**	**Acc. (%)**
Jordanian (Eastern)	AJGT	**82.8**	**83.3**	82.2	83.1
	ArTwitter	**84.8**	**84.9**	83.8	83.9
Jor/Egy/Gul (Eastern)	JEG	**74.3**	**74.8**	73.5	74.2
Tunisian (Western)	TEC	87.9	88.3	**89.0**	**89.5**
	TSAC	86.2	86.5	**87.1**	**87.7**
Moroccan (Western)	MEC	72.8	79.2	**74.0**	**80.6**
Eastern+Western	TEAD	62.7	73.3	**68.1**	**75.7**

Table 7.5 reviews the performances resulting from training Neu Tw-StAR with embeddings composed by SOWE and Avg functions for the studied datasets.

Considering the results shown in Table 7.5, we can observe that for the Eastern dialect datasets: AJGT, ArTwitter and JEG, SOWE produced more expressive features than the Avg function. However, for the Western dialect datasets, Avg could slightly outperform SOWE as the achieved F-measure values were 89.0%, 87.1% and 74.0% compared to 87.9%, 86.2% and 72.8% for TEC, TSAC and MEC datasets, respectively. This could be attributed to the nature of the Western dialects where many transliterated words derived from French, Spanish and Tamazight languages do exist (Zaidan and Callison-Burch (2014)). Being transliterated, these words are usually written in different writing styles which makes SOWE missing the synonymous relations among such words bearing same or close sentiments, and hence, leads to less expressive embedding features compared to Avg.

7.2.4. Neu Tw-StAR Shallow Architecture Evaluation

Through the proposed model, we are seeking to obtain an efficient sentiment classification performance using a less complicated architecture and with the least time overhead. Therefore, Neu Tw-StAR was designed as a shallow feed-forward neural network with one hidden layer. The ability of Neu Tw-StAR to rival deep neural models was examined by feeding our n-gram embeddings to train two deep neural models having the building units: CNN and LSTM, in addition to the DAN model developed in (Iyyer et al. (2015)). The CNN-based model was cloned from (Kim (2014)), while LSTM-based model was developed with several depths (2,3,4,5) formulated by stacking LSTM layers. Within these architectures, two type of experiments were conducted:

- The first experiment involved using training n-gram embeddings composed by SOWE composition function.
- The second experiment, however, employed n-gram embeddings composed via Avg composition function.

The efficiency of the shallow Neu Tw-StAR model was, then, assessed through conducting a comparison between the sentiment classification performances yielding from the previous shallow/deep SA models for SOWE and AVG compositionalities. Tables 7.6 and 7.7 summarize the obtained results where (Arch.) and (Time) refer to the adopted model architecture and the consumed training time, respectively. It should be noted that, for LSTM-based model, we selected the best performances achieved by various depths. In addition, DAN system in Table 7.6 was trained with embeddings composed by SOWE and not by AVG as in (Iyyer et al. (2015)), while its 2-hidden layers deep architecture was retained.

As it can be seen from Table 7.6, in AJGT, ArTwitter and JEG datasets, Neu Tw-StAR outperformed LSTM-based, CNN-based and DAN deep models, whereas it achieved a slightly better F-measure for MEC and TEAD datasets. A compara-

Table 7.6. Neu Tw-StAR, CNN, LSTM and DAN performances by SOWE

Dataset	**Arch.**	**Depth #**	**F1 (%)**	**Acc. (%)**	**Time (sec)**
AJGT	LSTM	4	81.6	82.8	77 sec
	CNN	3	80.1	81.4	34 sec
	DAN	2	79.0	79.7	**20 sec**
	Neu Tw-StAR	1	**82.8**	**83.3**	23 sec
ArTwitter	LSTM	2	82.2	82.0	102 sec
	CNN	3	80.0	80.2	41 sec
	DAN	2	80.7	81.0	**24 sec**
	Neu Tw-StAR	1	**84.1**	**84.1**	26 sec
JEG	LSTM	5	72.3	72.7	290 sec
	CNN	3	73.9	74.1	130 sec
	DAN	2	72.0	72.7	**80 sec**
	Neu Tw-StAR	1	**74.3**	**74.8**	91 sec
TEC	LSTM	5	88.0	88.3	188 sec
	CNN	3	87.1	88.0	75 sec
	DAN	2	**88.6**	**89.2**	**47 sec**
	Neu Tw-StAR	1	87.8	88.2	52 sec
TSAC	**LSTM**	2	**89.1**	**89.4**	412 sec
	CNN	3	84.9	85.8	168 sec
	DAN	2	87.3	87.7	**107 sec**
	Neu Tw-StAR	1	86.2	86.5	121 sec
MEC	LSTM	5	72.1	77.2	1292 sec
	CNN	3	70.1	76.3	637 sec
	DAN	2	71.1	76.9	**388 sec**
	Neu Tw-StAR	1	**72.8**	**79.2**	466 sec
TEAD	LSTM	4	62.7	73.4	31 hrs 30 mins
	CNN	3	62.0	74.6	24 hrs
	DAN	2	62.2	**74.1**	**19 hrs 50 mins**
	Neu Tw-StAR	1	**62.7**	73.3	20 hrs

ble performance, however, was observed for TEC and TSAC, where LSTM-based model was the best performing system for these datasets. This could be attributed to the efficiency of SOWE in producing more discriminating features for Eastern dialect datasets such as AJGT, ArTwitter and JEG compared to Western dialect collections. However, in general, we cannot ignore the ability of Neu Tw-StAR to be an efficient replacement of more complicated deep models. Especially that, for most datasets, Neu Tw-StAR managed to provide a quite good sentiment classification performance with fewer parameters and much faster training time compared to LSTM-based, CNN-based and DAN deep models.

Similarly, the results listed in Table 7.7 shows the competent performance achieved by Neu Tw-StAR compared to the other models where it scored the best F-measure values for TEC, MEC and JEG datasets; While comparable performances were obtained for AJGT, ArTwitter, TSAC and TEAD datasets.

Table 7.7. Neu Tw-StAR, CNN, LSTM and DAN performances by Avg

Dataset	Arch.	Depth #	F1 (%)	Acc. (%)	Time
AJGT	**LSTM**	5	**82.4**	**83.3**	78 sec
	CNN	3	80.4	81.9	34 sec
	DAN	2	79.6	81.1	**21 sec**
	Neu Tw-StAR	1	82.2	83.1	23 sec
ArTwitter	**LSTM**	1	**84.4**	**84.4**	99 sec
	CNN	3	82.5	82.6	41 sec
	DAN	2	76.5	76.8	**25 sec**
	Neu TW-StAR	1	83.3	83.3	26 sec
JEG	LSTM	4	71.4	72.0	286 sec
	CNN	3	72.8	73.3	129 sec
	DAN	2	72.4	73.0	**81 sec**
	Neu Tw-StAR	1	**73.1**	**73.5**	91 sec
TEC	LSTM	4	87.5	87.8	194 sec
	CNN	3	86.3	86.7	76 sec
	DAN	2	87.9	88.3	**47 sec**
	Neu Tw-StAR	1	**89.0**	**89.5**	52 sec
TSAC	**LSTM**	3	**88.9**	**89.2**	411sec
	CNN	3	87.3	87.9	169 sec
	DAN	2	81.7	83.2	**108 sec**
	Neu Tw-StAR	1	87.1	87.7	123 sec
MEC	LSTM	5	72.9	79.0	1316 sec
	CNN	3	71.0	79.3	641 sec
	DAN	2	73.5	79.3	**400 sec**
	Neu Tw-StAR	1	**74.0**	**80.6**	478 sec
TEAD	LSTM	4	66.2	75.0	31 hrs 10 mins
	CNN	3	62.3	75.1	25 hrs 30 mins
	DAN	2	**69.5**	**77.4**	18 hrs 40 mins
	Neu Tw-StAR	1	68.1	75.7	**18 hrs 20 mins**

7.2.5. Neu Tw-StAR Vs. Baseline systems

The performances obtained by Neu Tw-StAR using SOWE composition function were further compared against the baseline systems that tackled the same datasets (see Table 7.8). Due to the lack of embeddings-based Arabic SA systems, we had to compare to the available hand-crafted baseline models: Alomari et al. (2017); Sayadi et al. (2016); Elouardighi et al. (2017) for AJGT, TEC and MEC, while for the datasets: ArTwitter, JEG and TSAC embedding-based baseline models were provided by Al-Azani and El-Alfy (2017); Altowayan and Tao (2016); Mdhaffar et al. (2017). Compared to the state-of-the-art applied on the investigated datasets (See Table 7.8), our results showed that Neu Tw-StAR, trained with syntax-ignorant n-gram embeddings, could improve the classification performance over the baseline systems in most datasets.

Table 7.8. Neu Tw-StAR Vs. baseline models

Dataset	model	F1 (%)	Acc. (%)
AJGT	hand-crafted (Alomari et al. (2017))	**88.3**	**88.7**
	Neu Tw-StAR	82.8	83.3
ArTwitter	combined LSTM(Al-Azani and El-Alfy (2017))	**87.2**	**87.2**
	CNN (Dahou et al. (2016))	-	85.0
	Neu Tw-StAR	84.1	84.9
TEC	hand-crafted (Sayadi et al. (2016))	63.0	71.1
	NeuTw-StAR	**87.8**	**88.2**
TSAC	MLP/doc2vec (Mdhaffar et al. (2017))	78.0	78.0
	NeuTw-StAR	**86.2**	**86.5**
MEC	hand-crafted Elouardighi et al. (2017)	-	78.0
	Neu Tw-StAR	**72.8**	**79.2**
JEG	word embeddings Altowayan and Tao (2016)	**79.6**	**80.2**
	Neu Tw-StAR	74.3	74.8

As we can see in Table 7.8, with Neu Tw-StAR applied, the accuracy values increased by 17.1%, 8.3% and 1.2% for TEC, TSAC and MEC datasets, respectively. Here, we can notice that, the less accuracy increment was reported in MSA/Moroccan MEC dataset; This defines the proposed embeddings as expressive features of pure dialectal content more than they are for MSA ones; as the free word order and varying syntactic structure of dialects can be be better handled by SOWE. Moreover, for ArTwitter dataset, a competent performance was achieved by Neu Tw-StAR against complicated neural architectures such as CNNs adopted by Dahou et al. (2016) and combined LSTMs used in (Al-Azani and El-Alfy (2017)), where the accuracy decreased by 0.1% and 2.3% compared to (Dahou et al. (2016)) and (Al-Azani and El-Alfy (2017)), respectively. Consequently, a shallow neural model such as Neu Tw-StAR trained with embeddings, specifically composed to target the Arabic dialectal content, can rival much more complicated neural architectures. In addition, for JEG dataset that contains three different dialects, although Neu Tw-StAR could not outperform the baseline system, a satisfying performance was achieved without the need for a huge external knowledge resources such as the training dataset used in (Altowayan and Tao (2016)) to provide the word embeddings.

7.2.6. Neu Tw-StAR Training Time Evaluation

Besides the competent performance of the our shallow model against more complicated deep architectures, Neu Tw-StAR could accomplish the training phase consuming less time compared to LSTM-based and CNN-based models. This is reviewed in Table 7.6 and Table 7.7 where we can notice that, in AJGT dataset, it took 23 seconds to train the features composed by SOWE (Table 7.6) while LSTM and CNN models consumed 77 seconds and 34 seconds, respectively. Similar behavior could be observed in Table 7.7.

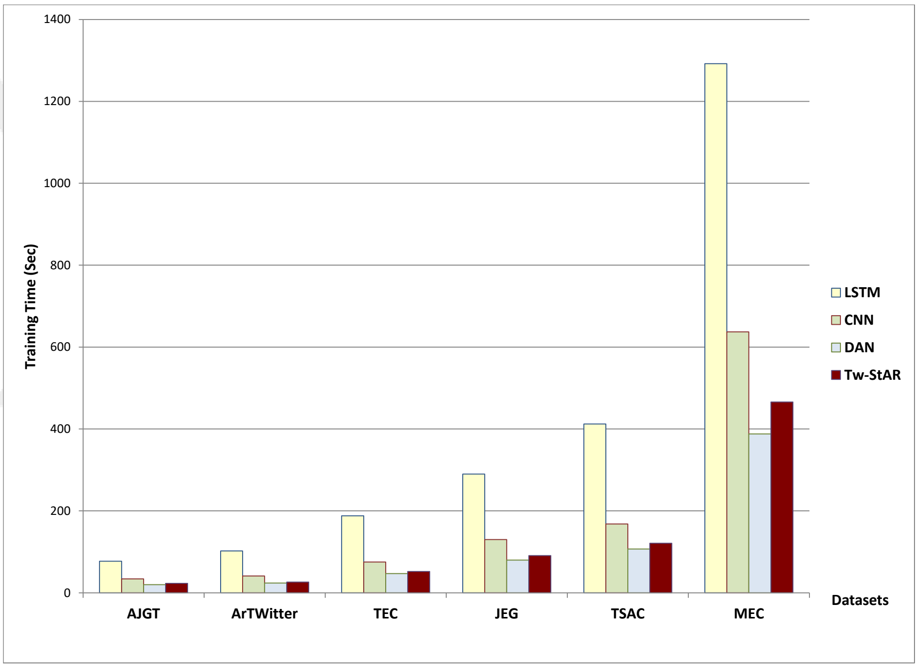


Figure 7.1. Training Time comparison for embeddings composed by SOWE.

This could be explained by the high computational complexity consumed at each layer of the LSTM model, where at every time step, in addition to the recurrent input, if the input is already yielded from an LSTM layer (in the case of stacked LSTMs), the current LSTM, then, can create a more complex feature representation of the current input (Al-Azani and El-Alfy (2017)); which, in turn, raises the time overhead. In addition, the learning mechanism adopted by the CNN-based model involves detecting multiple feature patterns through using various kernel sizes then concatenating their outputs at each convolution(Kim (2014)); Consequently, more

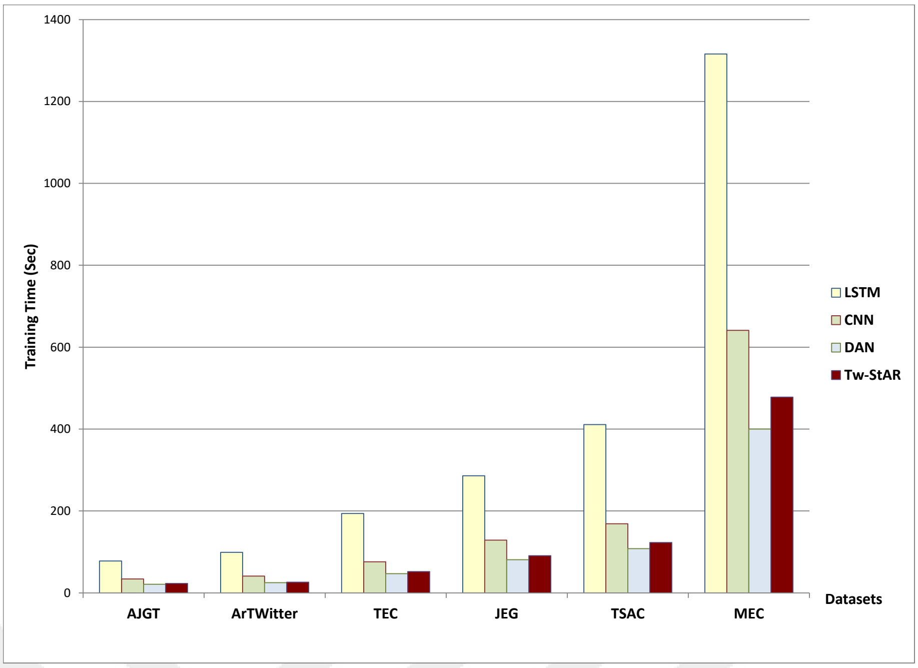


Figure 7.2. Training Time comparison for embeddings composed by Avg.

time is required for training compared to that needed by Neu Tw-StAR. The training time consumed across the studied models: CNN, LSTM, DAN and Neu Tw-StAR is illustrated in Figure 7.1 and Figure 7.2 for SOWE and Avg composition functions, respectively.

On the other hand, considering Figures 7.1 and 7.2, it could be noted that, the 2-hidden layer deep model (DAN) proposed by Iyyer et al. (2015) achieved the best training time among all the model architectures for both SOWE and Avg composition functions. However, when exploring the training time values recorded for our model and those of DAN's, we can see that the latter is consuming slightly less time compared to Neu Tw-STAR as DAN needed a training time less by 3, 2, 5 and 11 seconds for AJGT, ArTwitter, TEC and JEG datasets, respectively. In addition, for large-sized datasets such as MEC and TSAC, although this time difference increases, yet, it does not exceeds 78 seconds and it is compensated by the better sentiment classification performance achieved by Neu Tw-StAR for these datasets (see Table 7.7 and Table 7.6). Similarly, for the large-scaled dataset TEAD, DAN and Neu Tw-StAR consumed quite same training time and were of the least time overhead among the studied architectures, as it can be seen from Figure 7.3, where

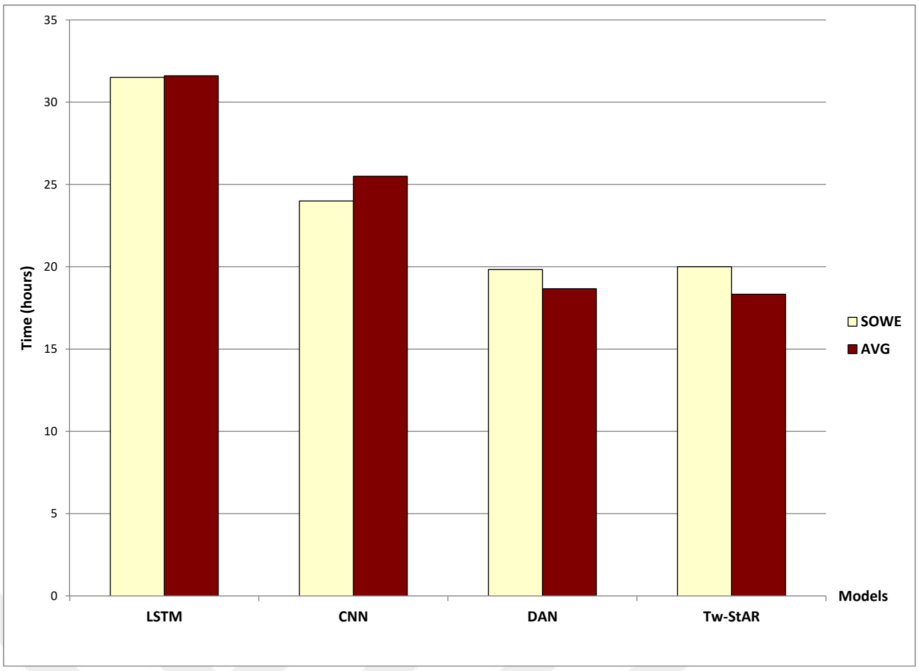


Figure 7.3. Training Time of all models for SOWE, Avg in TEAD.

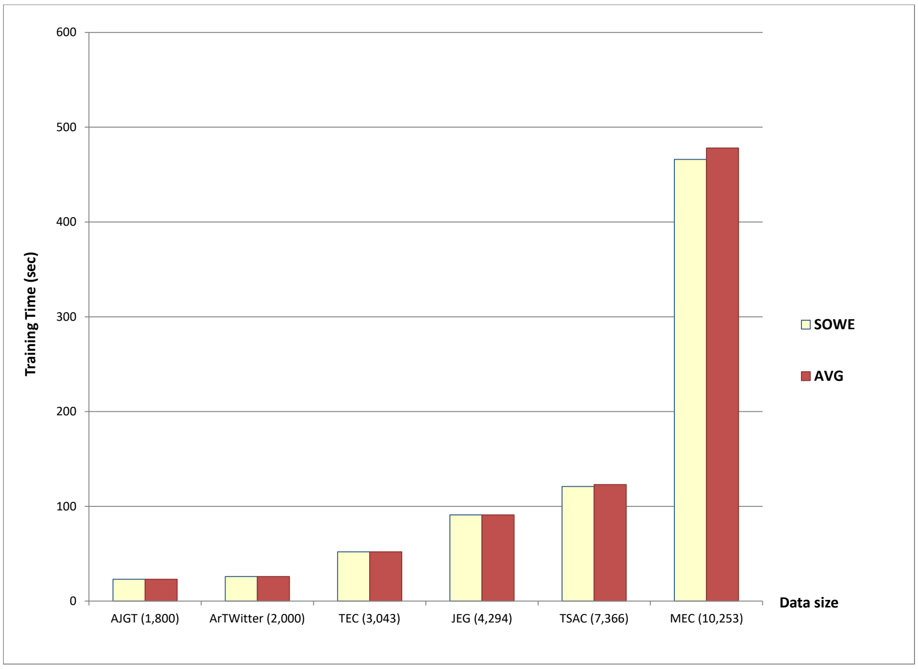


Figure 7.4. Training time of Neu Tw-StAR for SOWE, Avg due to data size.

the comparison was done for both SOWE and Avg composition functions. This reflects the ability of less complicated models such as Neu Tw-StAR and DAN to maintain a robust performance overhead while handling different sizes of datasets.

In the same context, for Neu Tw-StAR model, although learning features composed by Avg requires a little bit more time than those composed by SOWE for large-sized datasets, the time needed to train the model with both feature variants, increases according to the size of the input data as it can be seen from Figure 7.4.

7.3. Evaluation Summary

With such a variety of experiments, it was revealed that, the presented model Neu TwStAR trained with syntax-ignorant n-gram embeddings could classify the sentiment of several dialects better than most of the baseline systems and deep complicated architectures. The evaluation process considered several aspects including the learned n-gram embeddings and their efficiency as discriminating DA sentiment features, the adopted composition function and the robustness of the employed shallow architecture. This enabled answering the last four research questions listed in Section 1.3 as follows:

RQ4: Compared to context-aware embedding algorithms: word2vec and doc2vec, can the proposed syntax-ignorant embeddings provide a better mapping of sentimental words and, hence, a better SA performance?

- With Eastern/Western Arabic datasets used for evaluation, our learned syntax-ignorant n-gram embedding features proved their efficiency as expressive and discriminating features for multiple Arabic dialects where training Neu Tw-StAR with the propsoed n-gram embeddings achieved remarkably better evaluation measures compared to word2vec and doc2vec (PV-DBoW, PV-DM) embeddings.

- Based on exploring the visualization maps of the word embeddings learned by Neu Tw-StAR, word2vec and doc2vec (PV-DBoW) models, it was possible to deduce that several words of close sentiments were better mapped using Neu Tw-StAR model.

RQ5: With the existence of several unordered composition functions, is the quality of the proposed DA n-gram embeddings, related to a specific composition function?

- Being composed by SOWE function, our unordered, syntax-ignorant n-gram embeddings emphasized the efficiency of using unordered additive composition model in the SA task as the produced performances by n-gram embeddings were better than those learned via word2vec and doc2vec (PV-DM/PV-DBoW) models.
- The comparison between SOWE and Avg for the SA task of Eastern and Western Arabic dialectal content showed that, the sentiment of Eastern dialects were better expressed by SOWE-composed features, while the features formulated by Avg led to better sentiment classification performance for Western dialect datasets.

RQ6: How likely is it for a shallow neural model, trained with embeddings specifically formulated for DA, to rival complicated neural architectures?

- At the implementation level, it was revealed that, a shallow neural model such as Neu Tw-StAR, trained with unordered embeddings, can address the varying syntax structure and free word order issues of DA yielding a competent performance with much more complicated deep learning architectures. This was emphasized by evaluating Neu Tw-StAR performance against deep SA models having the building units: LSTM and CNN; where Tw-StAR rivaled or sometimes overcome these models.

RQ7: At the implementation level, is it worthy to give up the newly-emerged deep architectures and adopt a feed forward shallow one, in return for reducing the consumed training time?

- Considering the consumed training time, it was observed that, compared to LSTM/CNN-based models, Neu Tw-StAR consumed less training time in all datasets. In contrast, compared to DAN system (Iyyer et al. (2015)), our shallow model consumed quite similar training time for the large-scaled dataset TEAD indicating that, less complicated architectures with unordered composed embedding features can handle the increased size of training data exhibiting a reduced time overhead.

7.4. Conclusion

In this chapter, we reviewed the various experiments carried out to evaluate Neu Tw-StAR as an efficient SA model of Eastern/Western Arabic dialects. Through the proposed experiments, we investigated the ability of the our syntax-ignorant n-gram embeddings to represent the DA sentiment compared to the context-aware, syntax-aware embedding algorithms state-of-the-art embeddings: word2vec and doc2vec. We, further, examined how expressive are our n-gram features, based on exploring the embedding visualization maps of n-gram embeddings and study the spatial relations between words of similar/opposite sentiments. Then, we justified our selection for the additive composition function by investigating its performance against the unorderd average composition function. At the implementation level, we questioned the ability of the shallow architecture of Neu Tw-StAR to rival complicated, deep neural architectures besides the baseline models, in terms of the sentiment classification performances and the consumed training time. Finally, we provided a comprehensive assessment of the proposed model highlighting the merits it introduces to support the specificity of DA.

8. CONCLUSIONS AND FUTURE WORK

In this dissertation, we have investigated the problem of Sentiment Analysis (SA) of Dialectal Arabic (DA) on social media. We have introduced two SA models: HCB Tw-StAR (Section 4) and Neu Tw-StAR (Section 6). Through the proposed models, we employed novel preprocessing tasks, generated expressive feature variants and evaluated different classification methods and architectures. This final section, recaps the presented SA models, reviews our contributions and findings and provides an insight into the potential future directions.

8.1. Research Summary

The ”Arab Spring” incidents have been accompanied with a revolutional growth of the Arabic opinionated content on social media platforms. With most of the shared comments, tweets and reviews being written in DA, an efficient ASA model needs to consider the complex morphological and linguistic properties of the Arabic language (Section 3.2), let alone, the non-standard grammatical nature and the drastic semantic/syntactic variances among Eastern and Western Arabic dialects (Section 6.1). In this thesis, we tackled SA of the DA content on social media, we aimed to develop a dialect-independent ASA model that could be easily applied across a wide variety of Arabic dialects.

In line with our thesis goal, we presented two ASA models: HCB Tw-StAR (Section 4) and Neu Tw-StAR (Section 6). While each model has its own type of sentiment features and classification methods, they were both used efficiently to mine, analyze and recognize the sentiment of multiple Eastern and Western Arabic dialects. This was achieved with the least dependence on Arabic NLP tools and without the need for external knowledge resources. A summary of the specifications and contributions involved within each of the presented models is given below.

8.1.1. HCB Tw-StAR

In this model, we focused on bridging the variances among the Arabic dialects by adopting universal text components such as NEs to be included among the sentiment features. In addition, we provided expressive, dialectal, hand-crafted sentiment features generated based on novel combinations of preprocessing tasks where no dialect-specific morphological analyzers were employed and with the least dependence on dialectal resources. The efficiency of the proposed hand-crafted features was evaluated within supervised and lexicon-based classification methods. The main contributions introduced by this model are:

- **NEs As Sentiment Indicatives:** in contrast to previous studies which ignored or eliminated NEs while conducting SA, and given that NEs are universal text components across the different Arabic dialects, we introduced NEs as sentiment indicatives and included them among the hand-crafted features of supervised and lexicon-based classifiers (Section 4.2). This required associating each NE in the studied corpus with a specific polarity (positive/negative). Therefore, we developed an algorithm to detect the sentiment borne by an NE based on the local contextual content (Section 4.2.2). Having NEs sentiment identified, each NE was replaced by a specific textual tag indicating its polarity. This on one hand reduced the features size by unifying all NEs into two textual tags, and on the other hand resolved the issue of confusing some Arabic person names with sentimental adjectives (El-Beltagy and Ali (2013); El-Makky et al. (2014)). The role of NEs in inferring the sentiment was, then, investigated for Eastern (Levantine) and Western (Tunisian) Arabic dialects at a coarse-grained sentiment analysis level;

- **Novel Preprocessing Task combinations for Better SA:** aiming to develop a less dialect-dependent SA model, we formulated several combinations out of the following preprocessing tasks: stopwords removal, stemming, light stemming, lemmatization, emoji tagging, negation detection and NEs tagging (Section 4.3). Thus, novel sentiment hand-crafted features could be obtained based on the preprocessed text. With various combinations of preprocessing tasks applied, HCB Tw-StAR was evaluated as a coarse-grained sentiment analysis model for Arabic (Jordanian/Tunisian) and non-Arabic (Turkish) datasets (Section 5.4.3). In addition, the efficiency of HCB Tw-StAR was further assessed at a fine-grained sentiment analysis level where we conducted multi-label emotion classification of DA, English and Spanish textual contents (Section 5.4.4);

- **The Joint Impact of Preprocessing and NEs on SA:** with both preprocessing and sentimental NEs involved, we applied HCB Tw-StAR to mine the sentiment in Jordanian and Tunisian Arabic dialectal contents at a coarse-grained sentiment analysis level. This enabled examining the joint impact of NEs and preprocessing on SA of Eastern and Western Arabic dialects (Section 5.5).

8.1.2. Neu Tw-StAR

Through this model, we dispensed the preprocessing phase and provided a SA model that learns low-dimensional, real-valued expressive features from an input raw text. Neu Tw-StAR was developed as a dialectal-independent SA model such that it can handle the variances among dialects and could be applied across different Arabic dialects. This was done by adopting novel syntax-ignorant, sentiment-specific, unordered n-gram embedding features. The features generation and learning process was conducted within a shallow neural architecture which reduces the computation complexity and thus the consumed training time. The main contributions introduced by this model are:

- **Syntax-Ignorant and Sentiment-Specific n-gram Embeddings for DA:** when exploring the different Arabic dialects, we realized that with the free word order and the varying syntactic nature of DA (Section 6.1), the syntactic information cannot be relied on to provide expressive features for DA. Therefore, unlike the contextual-aware, syntactic-aware embedding methods used in (Mikolov et al. (2013); Le and Mikolov (2014); Tang et al. (2014)), we introduced syntax-ignorant, sentiment-specific n-gram embeddings in which the syntactic information were ignored by learning the embeddings from whole and non-corrupted (not missing a word) n-grams; while the sentiment information was better captured and integrated as the learned embeddings were sentiment-informed since the polarity labels were associated with the input training instances (Section 6.2). A comprehensive statistical and visual evaluation was provided for the proposed n-gram embeddings as they were compared against State-Of-The-Art context-aware, syntax-aware embedding methods: word2vec (Mikolov et al. (2013)) and doc2vec (Le and Mikolov (2014)) (Section 7.2.1,Section 7.2.2).

- **Unordered Compositionality for N-gram Embeddings:** we opted to compose our n-gram embeddings using the unordered additive composition function SOWE (Section 6.2); as it was proved to be the best in capturing the semantic information for the sentence similarity task exhibiting a low computation overhead (White et al. (2015)). This enabled the proposed n-gram embeddings to handle the free word order and incorporate the semantic and synonymous regularities of the input DA contents leading to more expressive sentiment embedding features (Section 7.2.1). Later, we evaluated SOWE as an efficient replacement of the unordered Avg function used in (Le and Mikolov (2014); Iyyer et al. (2015)) through conducting a comparison between the sentiment classification performances yielded from n-gram embeddings composed by SOWE and Avg functions, respectively (Section 7.2.3);

- **Shallow Neural Architecture:** as we are seeking to accomplish the SA task of DA using a less complicated neural architecture and within a less training time, we implemented Neu Tw-StAR as a feed-forward neural model of a single hidden layer (Section 6.2). Bearing in mind that most of the State-Of-The-Art studies adopted deep neural models (Iyyer et al. (2015); Dahou et al. (2016); Al-Azani and El-Alfy (2017); Baniata and Park (2016); Gridach et al. (2017)), we investigated the ability of our shallow Neu Tw-StAR to rival deep neural models having two hidden layers: DAN (Iyyer et al. (2015)) in addition to Convolutional Neural Networks (CNN) and Long short Term Memory netwotks (LSTM) models specifically-built for this study to be trained on the tackled datasets. This provided a comprehensive comparison between the performance of the proposed shallow neural model and that of deep neural models in terms of the achieved evaluation measures and the consumed training time (Section 7.2.4, Section 7.2.6).

8.2. Findings Summary

Over the course of this thesis, we have presented two novel models for SA of DA on social media. The efficiency of the proposed models was assessed at coarse-grained and fine-grained sentiment analysis levels using several Eastern and Western dialectal Arabic datasets in addition to English, Spanish and Turkish datasets (Section 5.2,Section 7.1.1). As seeking a comprehensive evaluation for both HCB Tw-StAR and Neu Tw-StAR models, we conducted various experiments (Section 5,Section 7) through which the employed preprocessing tasks, the generated sentiment feature types and the implementation details and architecture were, thoroughly, investigated. This enabled answering the research questions evoked by this thesis (Section 1.3). In the following list, we review and summarize our findings while associating them with the relevant research questions for both of the presented models.

RQ1: Are NEs reliable enough to infer the DA sentiment within hand-crafted feature-based SA models? And is it more likely to have a better SA performance for datasets rich of NEs?

- Considering the experiments conducted in Section 5.3, we found that the role of NEs in SA was more clear within the lexicon-based classifier of HCB Tw-StAR as the the classification performances were remarkably improved when NEs were considered among the features. However, the impact of NEs on the SA performance obtained by the proposed supervised SA classifier was inconclusive. This could be attributed to the fact that our NEs sentiment detection algorithm adopts a context-ignorant manner to associate NEs with their proper sentiments; which copes with the strategy followed by the lexicon-based method where context information do not count while recognizing the sentiment.

- When exploring the number of NEs extracted and sentimentally-annotated in a dataset (Section 5.3); then tracking the achieved improvement in the sentiment classification performance for this dataset, we found that the improvement in the SA performance depends on the accurate sentiment annotation of NEs more than the number of them in a dataset. In this context, we have noticed that in corpora having a good degree of consistency, training and test sets tend to contain overlapped NEs used within the same domain. Thus, the NEs sentiment detection algorithm (Section 4.2.2) can obtain an unanimous over the sentiment of a specific NE which, in turn, leads to an improved SA performance (Section 5.3.1, Section 5.3.2).

RQ2: Which combination of preprocessing tasks can lead to an improved performance in hand-crafted features-based SA models?

- The experiments conducted using the supervised classifier of HCB Tw-StAR (Section 5.4.1) specified stemming, negation detection and tagging, the combination (emoji tagging, stemming) and the combination (light stemming, negation) as the best-performing preprocessing tasks for SA of DA. On the other hand, stemming, negation detection and the combination (emoji, stemming) were found of the best impact on the sentiment classification performance conducted by the lexicon-based classifier of HCB Tw-StAR with DA datasets (Section 5.4.2).

RQ3: Would the sentiment classification performance improved if NEs were included together with specific combinations of preprocessing tasks?

- This has been investigated through the experiments carried out with both NEs processing and preprocessing phases included within HCBTw-StAR model (Section 5.5). The preprocessing tasks which will be combined with NEs were selected carefully based on their impact on the SA performance for each of supervised and lexicon-based classifiers of HCB Tw-StAR (Section 5.4.1,Section 5.4.2). The results indicated that further improvement in the sentiment classification performance could be obtained when integrating NEs with specific single/combinations of preprocessing tasks for Eastern (Jordanian) and Western (Tunisian) Arabic dialects.

RQ4: Compared to context-aware embedding algorithms: word2vec and doc2vec, can the proposed syntax-ignorant embeddings provide a better mapping of sentimental words and, hence, a better SA performance?

- The comparison conducted between the proposed syntax-ignorant n-gram embedding features and word2vec (Mikolov et al. (2013)), doc2vec (Le and

Mikolov (2014)) embedding algorithms indicated that our n-gram embeddings were more expressive and discriminating for multiple Arabic dialects. This was proved statistically as the proposed n-gram embeddings achieved remarkably better evaluation measures compared to word2vec and doc2vec (PV-DBoW, PV-DM) embeddings (Section 7.2.1).

- With t-SNE tool used to visualize our n-gram embeddings against word2vec and doc2vec embeddings in a two-dimensional space, we could explore the visualization maps of the word embeddings learned by Neu Tw-StAR with n-gram embeddings, word2vec and doc2vec (PV-DBoW) models. Hence, it could be noted that the proposed n-gram embeddings could map words of similar sentiments close to each other in the embeddings space (Section 7.2.2).

RQ5: With the existence of several unordered composition functions, is the quality of the proposed DA n-gram embeddings, related to a specific composition function?

- When we compared the SA performances obtained by our n-gram embeddings, word2ved and doc2vec against each other (Section 7.2.1), it could be deduced that the using SOWE to compose the proposed n-gram embeddings yielded better sentiment classification performances and, hence, more expressive sentiment features compared to word2vec and doc2vec whose embeddings were composed via the Avg composition function.

- On the other hand, when SOWE was replaced with Avg to compose the syntax-ignorant n-gram embeddings which were used to train Neu Tw-StAR, we found that while the sentiment of Eastern dialects were better expressed by SOWE-composed n-gram features, the n-gram embedding features formulated by Avg led to better sentiment classification performance for datasets of Western Arabic dialects (Section 7.2.3).

RQ6: How likely is it for a shallow neural model, trained with embeddings specifically formulated for DA, to rival complicated neural architectures?

- To answer this question, we evaluated Neu Tw-StAR performance against deep SA models: LSTM, CNN and DAN (Iyyer et al. (2015)) (Section 7.2.4). The results revealed that a shallow neural model such as Neu Tw-StAR, trained with unordered n-gram embeddings, could rival or sometimes overcome the investigated deep neural models. This indicates the ability of Neu Tw-StAR to address the varying syntactic nature and the free word order issues of DA yielding a competent performance with more complicated deep neural architectures.

RQ7: At the implementation level, is it worthy to give up the newly-emerged deep architectures and adopt a feed-forward shallow one, in return for reducing the consumed training time?

- Considering the results obtained by the experiments in Section 7.2.4), it was observed that, compared to LSTM/CNN-based models, Neu Tw-StAR consumed less training time in all datasets. In contrast, compared to DAN system (Iyyer et al. (2015)), our shallow model consumed quite similar training time for the large-scaled dataset TEAD indicating that, less complicated architectures with unordered composed embedding features can handle the increased size of training data exhibiting a reduced time overhead.

8.3. Future Directions

Through this thesis, we aimed to remedy some of the existing gaps in ASA domain by introducing two SA models for DA with novel hand-crafted and embedding features. Considering the challenging nature of DA, we believe that further improvement of the obtained SA performances could be achieved at different levels. The possible directions of future work may include:

- **Preprocessing tasks improvement and extension:** besides the preprocessing tasks employed by HCB Tw-StAR, the obtained SA performances would be further improved if negation detection strategy was extended to handle irony and sarcastic content. In addition, using a Tunisian stopwords list instead of the adopted MSA stopwords might enhance the stemming task. Regarding Turkish SA, we assume that adopting a negation detection pattern for verbs would assist in recognizing negated verbs more accurately and enhance the sentiment classification performance.

- **Supporting an improved exploitation of NEs in SA:** for underrepresented dialects and multilingual contents. Given the lack of the needed pretrained embeddings for underrepresented Arabic dialects, it would be better if Tunisian corpora were provided to produce the pretrained word vectors which will be used in the NER system (Gridach (2016)). Hence, special Tunisian NEs such as the singer name "كافون" could be recognized as a person name and tagged properly, rather than being, mistakenly, identified as the MSA word that means "enough" having the stem "كاف". This would also enable recognizing the different writing styles of NEs. In addition, it would be useful if the idea of involving NEs in the SA task could be extended for other languages such as English, French and Turkish.

- **Extending syntax-ignorant n-gram embeddings:** to be used for DA lexicon construction; where a multi-dialectal lexicon would be constructed based on the spatial distances among the word vectors of the n-gram embeddings learned within Neu Tw-StAR and visualized by t-SNE tool. It would be interesting to examine if the proposed syntax-ignorant n-gram embeddings can be employed in SA of informal social media posts written in other languages.

- **Towards devoting the ethical aspect of ASA:** with the freedom of expression privilege granted to social media users, it became easy to spread abusive/hate propaganda against individuals or groups. Beyond the psychological harm, toxic online contents can lead to actual hate crimes (Matsuda (2018)). This provokes the need for automatic detection of toxic contents on social media. Hate speech and abusive language detection can be considered as a subtask of SA for which hand-crafted/embedding features along with various machine learning techniques are used. However, while there is an increased number of hate speech/abusive language detection studies for Indo-European contents, similar research for DA remains very limited. This is due to the lack of the publicly-available hate speech/abusive language resources. Building such resources involves several difficulties in terms of data collection and annotation, especially for underrepresented Arabic dialects. Thus, there is still a lot to do to explore this new area of SA, especially with the volatile political/social atmosphere in the Arab world; where intensive debates on social media are, unfortunately, rich of abusive and hate speech.

REFERENCES

Abbasi, A., Chen, H., and Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12.

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, p. 11–16.

Abdellaoui, H. (2018). Tweets emoji arabic dataset (tead). `https://github.com/HSMAabdellaoui/TEAD`. (last visited on 10-06-2019).

Abdul-Mageed, M. (2015). *Subjectivity and sentiment analysis of Arabic as a morophologically-rich language*. PhD thesis, Indiana University.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Abdul-Mageed, M. and Diab, M. T. (2011). Subjectivity and sentiment annotation of modern standard arabic newswire. In *Proceedings of the 5th Linguistic Annotation Workshop*, p. 110–118. Association for Computational Linguistics.

Abdulla, N., Mohammed, S., Al-Ayyoub, M., Al-Kabi, M., and others (2014). Automatic lexicon construction for arabic sentiment analysis. In *2014 International Conference on Future Internet of Things and Cloud (FiCloud)*, p. 547–552. IEEE.

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

Adamov, A. Z. and Adali, E. (2016). Opinion mining and sentiment analysis for contextual online-advertisement. In *2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)*, p. 1–3. IEEE.

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data.

Akaichi, J. (2014). Sentiment classification at the time of the tunisian uprising: Machine learning techniques applied to a new corpus for arabic language. In *2014 European Network Intelligence Conference*, p. 38–45. IEEE.

Al-Azani, S. and El-Alfy, E.-S. M. (2017). Hybrid deep learning for sentiment polarity determination of arabic microblogs. In *International Conference on Neural Information Processing*, p. 491–500. Springer.

Al-Kabi, M. N., Abdulla, N. A., and Al-Ayyoub, M. (2013). An analytical study of arabic sentiments: Maktoob case study. In *2013 8th International Conference for Internet Technology and Secured Transactions (ICITST)*, p. 89–94. IEEE.

Al-Moslmi, T., Albared, M., Al-Shabi, A., Omar, N., and Abdullah, S. (2017). Arabic senti-lexicon: Constructing publicly available language resources for arabic sentiment analysis. *Journal of Information Science*.

Al-Osaimi, S. and Badruddin, K. M. (2014). Role of emotion icons in sentiment classification of arabic tweets. In *Proceedings of the 6th international conference on management of emergent digital ecosystems*, p. 167–171. ACM.

Al-Otaibi, S., Alnassar, A., Alshahrani, A., Al-Mubarak, A., Albugami, S., Almutiri, N., and Albugami, A. (2018). Customer satisfaction measurement using sentiment analysis. *International Journal of Avanced Computer Science and Applications*, 9(2):106–117.

Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*.

Al-Sallab, A., Baly, R., Hajj, H., Shaban, K. B., El-Hajj, W., and Badaro, G. (2017). Aroma: a recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):25.

Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El Hajj, W., and Shaban, K. B. (2015). Deep learning models for sentiment analysis in arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, p. 9–17.

Al-Twairesh, N., Al-Khalifa, H., Al-Salman, A., and Al-Ohali, Y. (2017). Arasenti-tweet: A corpus for arabic sentiment analysis of saudi tweets. *Procedia Computer Science*, 117:63–72.

Al-Twairesh, N., Al-Khalifa, H. S., and Alsalman, A. (2016). Arasenti: Large-scale twitter-specific arabic sentiment lexicons. In *ACL (1)*.

Alhumoud, S., Albuhairi, T., and Altuwaijri, M. (2015). Arabic sentiment analysis using weka a hybrid learning approach. In *7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K),*, volume 1, p. 402–408. IEEE.

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

Altowayan, A. A. and Tao, L. (2016). Word embeddings for arabic sentiment analysis. In *2016 IEEE International Conference on Big Data (Big Data),*, p. 3820–3825. IEEE.

Alwehaibi, A. and Roy, K. (2018). Comparison of pre-trained word vectors for arabic text classification using deep learning approach. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, p. 1471–1474. IEEE.

Aly, M. A. and Atiya, A. F. (2013). Labr: A large scale arabic book reviews dataset. In *ACL (2)*, p. 494–498.

Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F., and Manicardi, S. (2016). A comparison between preprocessing techniques for sentiment analysis in twitter. In *Proceedings of the 2nd International Workshop on Knowledge Discovery on the WEB (KDWeb)*.

Araujo, M., Reis, J., Pereira, A., and Benevenuto, F. (2016). An evaluation of machine translation for multilingual sentence-level sentiment analysis. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, p. 1140–1145. ACM.

Assiri, A., Emam, A., and Al-Dossari, H. (2017). Towards enhancement of a lexicon-based approach for saudi dialect sentiment analysis. *Journal of Information Science*.

Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? In *Advances in neural information processing systems*, p. 2654–2662.

Badaro, G., Baly, R., Hajj, H., El-Hajj, W., Shaban, K., Habash, N., Sallab, A., and Hamdi, A. (2018). A survey of opinion mining in arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications and visualizations. *ACM Transactions on Asian Language Information Processing*, p. 1–48.

Badaro, G., Baly, R., Hajj, H., Habash, N., and El-Hajj, W. (2014). A large scale arabic sentiment lexicon for arabic opinion mining. In *Proceedings of the EMNLP 2014 workshop on arabic natural language processing (ANLP)*, p. 165–173.

Balikas, G. and Amini, M.-R. (2016). Twise at semeval-2016 task 4: Twitter sentiment classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 85–91.

Baly, R., Badaro, G., Hamdi, A., Moukalled, R., Aoun, R., El-Khoury, G., Al Sallab, A., Hajj, H., Habash, N., Shaban, K., and El-Hajj, W. (2017). Omam at semeval-2017 task 4: Evaluation of english state-of-the-art sentiment analysis models for arabic and a new topic-based model. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, p. 603–610, Vancouver, Canada. Association for Computational Linguistics.

Banea, C., Mihalcea, R., and Wiebe, J. (2010). Multilingual subjectivity: Are more languages better? In *Proceedings of the 23rd international conference on computational linguistics*, p. 28–36. Association for Computational Linguistics.

Baniata, L. H. and Park, S.-B. (2016). Sentence representation network for arabic sentiment analysis. *Proceedings of the Korean Information Science Society*, p. 470–472.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bespalov, D., Bai, B., Qi, Y., and Shokoufandeh, A. (2011). Sentiment classification based on supervised latent n-gram analysis. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, p. 375–382. ACM.

Bharathi, S. and Geetha, A. (2017). Sentiment analysis for effective stock market prediction. *International Journal of Intelligent Engineering and Systems*, 10(3):146–153.

Biltawi, M., Etaiwi, W., Tedmori, S., Hudaib, A., and Awajan, A. (2016). Sentiment classification techniques for arabic language: A survey. In *7th International Conference on Information and Communication Systems (ICICS)*, p. 339–346. IEEE.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Boiy, E. and Moens, M.-F. (2009). A machine learning approach to sentiment analysis in multilingual web texts. *Information retrieval*, 12(5):526–558.

Bollen, J., Mao, H., and Pepe, A. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *Icwsm*, 11:450–453.

Bongirwar, V. K. (2015). A survey on sentence level sentiment analysis. *International Journal of Computer Science Trends and Technology (IJCST)*, 3:110–113.

Brahimi, B., Touahria, M., and Tari, A. (2016). Data and text mining techniques for classifying arabic tweet polarity. *Journal of Digital Information Management*, 14(1).

Brustad, K. (2000). *The syntax of spoken Arabic: A comparative study of Moroccan, Egyptian, Syrian, and Kuwaiti dialects*. Georgetown University Press.

Carrillo-de Albornoz, J., Vidal, J. R., and Plaza, L. (2018). Feature engineering for sentiment analysis in e-health forums. *PloS one*, 13(11):e0207996.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Chang, W.-L. and Wang, J.-Y. (2018). Mine is yours? using sentiment analysis to explore the degree of risk in the sharing economy. *Electronic Commerce Research and Applications*, 28:141–158.

Chen, Z., Ma, N., and Liu, B. (2015). Lifelong learning for sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, Volume 2*, p. 750–756.

Cherif, W., Madani, A., and Kissi, M. (2015). Towards an efficient opinion measurement in arabic comments. *Procedia Computer Science*, 73:122–129.

Chesley, P., Vincent, B., Xu, L., and Srihari, R. K. (2006). Using verbs and adjectives to automatically classify blog sentiment. *Training*, 580(263):233.

Chiang, D., Diab, M., Habash, N., Rambow, O., and Shareef, S. (2006). Parsing arabic dialects. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Collomb, A., Costea, C., Joyeux, D., Hasan, O., and Brunie, L. (2014). A study and comparison of sentiment analysis methods for reputation evaluation. *Rapport de recherche RR-LIRIS-2014-002*.

Council, B. (2013). Languages for the future. which languages the uk needs most and why. *Retrieved June*, 16:2015.

Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*.

Dadvar, M., Hauff, C., and de Jong, F. (2011). Scope of negation detection in sentiment analysis. In *Proceedings of the Dutch-Belgian Information Retrieval Workshop (DIR 2011)*, p. 16–20. Citeseer.

Dahab, M. Y., Ibrahim, A., and Al-Mutawa, R. (2015). A comparative study on arabic stemmers. *International Journal of Computer Applications*, 125(8).

Dahou, A., Xiong, S., Zhou, J., Haddoud, M. H., and Duan, P. (2016). Word embeddings and convolutional neural network for arabic sentiment classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, p. 2418–2427.

Darwish, K. and Magdy, W. (2014). Arabic information retrieval. *Foundations and Trends® in Information Retrieval*, 7(4):239–342.

Dehkharghani, R., Saygin, Y., Yanikoglu, B., and Oflazer, K. (2016). Sentiturknet: a turkish polarity lexicon for sentiment analysis. *Language Resources and Evaluation*, 50(3):667–685.

Demirtas, E. and Pechenizkiy, M. (2013). Cross-lingual polarity detection with machine translation. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, p. 9. ACM.

Denecke, K. and Deng, Y. (2015). Sentiment analysis in medical settings: New opportunities and challenges. *Artificial intelligence in medicine*, 64(1):17–27.

Di Nunzio, G. M. and Vezzani, F. (2018). A linguistic failure analysis of classification of medical publications: A study on stemming vs lemmatization. 2253.

Diab, M. (2009). Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking. In *2nd International Conference on Arabic Language Resources and Tools*, volume 110.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Duwairi, R., Ahmed, N. A., and Al-Rifai, S. Y. (2015). Detecting sentiment embedded in arabic social media–a lexicon-based approach. *Journal of Intelligent & Fuzzy Systems*, 29(1):107–117.

Duwairi, R. and El-Orfali, M. (2014). A study of the effects of preprocessing strategies on sentiment analysis for arabic text. *Journal of Information Science*, 40(4):501–513.

Duwairi, R. M., Marji, R., Sha'ban, N., and Rushaidat, S. (2014). Sentiment analysis in arabic tweets. In *in 5th ICICS)*, p. 1–6. IEEE.

El-Beltagy, S. R. and Ali, A. (2013). Open issues in the sentiment analysis of arabic social media: A case study. In *2013 9th international conference on Innovations in information technology (iit)*, p. 215–220. IEEE.

El-Beltagy, S. R., El kalamawy, M., and Soliman, A. B. (2017). Niletmrg at semeval-2017 task 4: Arabic sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, p. 790–795. Association for Computational Linguistics.

El-Beltagy, S. R., Khalil, T., Halaby, A., and Hammad, M. (2016). Combining lexical features and a supervised learning approach for arabic sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, p. 307–319. Springer.

El-Makky, N., Nagi, K., El-Ebshihy, A., Apady, E., Hafez, O., Mostafa, S., and Ibrahim, S. (2014). Sentiment analysis of colloquial arabic tweets. In *ASE BigData/SocialInformatics/PASSAT/BioMedCom 2014 Conference, Harvard University*, p. 1–9.

Elouardighi, A., Maghfour, M., Hammia, H., and Aazi, F.-z. (2017). A machine learning approach for sentiment analysis in the standard or dialectal arabic facebook comments. In *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, p. 1–8. IEEE.

Farooq, U., Mansoor, H., Nongaillard, A., Ouzrout, Y., and Qadir, M. A. (2017). Negation handling in sentiment analysis at sentence level. *JCP*, 12(5):470–478.

Farra, N., McKeown, K., and Habash, N. (2015). Annotating targets of opinions in arabic using crowdsourcing. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, p. 89–98.

FatihUniv (2010). Turkish stopwords. `http://nlp.ceng.fatih.edu.tr/blog/tr/?p=31`. (last visited on 22-05-2019).

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Ganesan, K. (2014). Terrier stopwords. `https://bitbucket.org/kganes2/text-mining-resources`. (last visited on 22-05-2019).

Ghadeer, A., Aljarah, I., and Alsawalqah, H. (2017). Enhancing the arabic sentiment analysis using different preprocessing operators. In *Proceedings of the New Trends in Information Technology (NTIT-2017)*, p. 113–117.

Ghag, K. V. and Shah, K. (2015). Comparative analysis of effect of stopwords removal on sentiment classification. In *2015 International Conference on Computer, Communication and Control (IC4),*, p. 1–6. Institute of Electrical and Electronics Engineers (IEEE).

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, p. 249–256.

Gomez, L., Patel, Y., Rusiñol, M., Karatzas, D., and Jawahar, C. (2017). Self-supervised learning of visual features through embedding images into text topic spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 4230–4239.

Gormley, M. R., Yu, M., and Dredze, M. (2015). Improved relation extraction with feature-rich compositional embedding models. *arXiv preprint arXiv:1505.02419*.

Gridach, M. (2016). Character-aware neural networks for arabic named entity recognition for social media. In *Proceedings of the 6th workshop on South and Southeast Asian natural language processing (WSSANLP2016)*, p. 23–32.

Gridach, M., Haddad, H., and Mulki, H. (2017). Empirical evaluation of word representations on arabic sentiment analysis. In *International Conference on Arabic Language Processing (ICALP)*, p. 147–158. Springer.

Guibon, G., Ochs, M., and Bellot, P. (2016). From emojis to sentiment analysis. In *Workshop Affect Compagnon Artificiel Interaction (WACAI 2016)*.

Gulcehre, C., Moczulski, M., Denil, M., and Bengio, Y. (2016). Noisy activation functions. In *International conference on machine learning*, p. 3059–3068.

Habash, N., Eskander, R., and Hawwari, A. (2012). A morphological analyzer for egyptian arabic. In *Proceedings of 12th meeting of the special interest group on computational morphology and phonology*, p. 1–9.

Habash, N. Y. (2010). Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Haddad, H. and Ali, C. B. (2014). Performance of turkish information retrieval: Evaluating the impact of linguistic parameters and compound nouns. In *International Conference on Intelligent Text Processing and Computational Linguistics*, p. 381–391. Springer.

Hercig, T. (2015). Aspects of sentiment analysis: technical report.

Huang, F. (2015). Improved arabic dialect classification with social media data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 2118–2126.

Hung, C. and Chen, S.-J. (2016). Word sense disambiguation based sentiment lexicons for sentiment classification. *Knowledge-Based Systems*, 110:224–232.

Ingason, A. K., Helgadóttir, S., Loftsson, H., and Rögnvaldsson, E. (2008). A mixed method lemmatization algorithm using a hierarchy of linguistic identities (holi). In *Advances in Natural Language Processing*, p. 205–216. Springer.

Itani, M. M., Zantout, R. N., Hamandi, L., and Elkabani, I. (2012). Classifying sentiment in arabic social networks: Naive search versus naive bayes. In *2012 2nd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, p. 192–197. IEEE.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, p. 1681–1691.

Jansen, B. J., Zhang, M., Sobel, K., and Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, p. 427–431. Association for Computational Linguistics.

KACST, N. (2017). Arabic stopwords. `https://github.com/abahanshal/arabic-stop-words-list`. (last visited on 22-05-2019).

Karmani, N. (2017). *Tunisian Arabic Customer's Reviews Processing and Analysis for an Internet Supervision System*. PhD thesis, Sfax University, Tunisia.

Khalifa, S., Zalmout, N., and Habash, N. (2016). Yamama: Yet another multi-dialect arabic morphological analyzer. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, p. 223–227.

Khoja, S. and Garside, R. (1999). Stemming arabic text. *Lancaster, UK, Computing Department, Lancaster University*.

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

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiritchenko, S., Zhu, X., and Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Kolkur, S., Dantal, G., and Mahe, R. (2015). Study of different levels for sentiment analysis. *International Journal of Current Engineering and Technology*, 5(2):768–770.

Larkey, L. S., Ballesteros, L., and Connell, M. E. (2002). Improving stemming for arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 275–282. ACM.

Larkey, L. S., Ballesteros, L., and Connell, M. E. (2007). Light stemming for arabic information retrieval. In *Arabic computational morphology*, p. 221–243. Springer.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, p. 1188–1196.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Li, J., Rao, Y., Jin, F., Chen, H., and Xiang, X. (2016). Multi-label maximum entropy model for social emotion classification over short text. *Neurocomputing*, 210:247–256.

Li, X., Grimes, S., Ismael, S., Strassel, S., Maamouri, M., and Bies, A. (2013). Linguistic data consortium arabic tree bank. `https://catalog.ldc.upenn.edu/LDC2013T14.` (last visited on 12-04-2019).

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Liu, S. M. and Chen, J.-H. (2015). A multi-label classification based approach for sentiment classification. *Expert Systems with Applications*, 42(3):1083–1093.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Magdy, W. and Darwish, K. (2016). Trump vs. hillary analyzing viral tweets during us presidential elections 2016. *Computing Research Repository (CoRR)*, abs/1610.01655.

Marr, B. (2018). How much data do we create every day? the mind-blowing stats everyone should read. *Forbes May*, 21:2018.

Matsuda, M. J. (2018). Public response to racist speech: Considering the victim's story. In *Words that wound*, p. 17–51. Routledge.

Mayard, A. (2013). 12 key statistics on how tunisians use social media. `https://www.wamda.com/2013/04/12-key-statistics-on-how-tunisians-use-social-media\\-infographic`. (last visited on 12-04-2019).

Mdhaffar, S., Bougares, F., Esteve, Y., and Hadrich-Belguith, L. (2017). Sentiment analysis of tunisian dialect: Linguistic resources and experiments. *WANLP 2017 (*co-located with *EACL 2017)*.

M.ElArnaoty, S.AbdelRahman, and A.Fahmy (2012). A machine learning approach for opinion holder extraction in arabic language. *IJAIA International Journal of Artificial Intelligence & Applications*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, p. 3111–3119.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Mohammad, S., Bravo-Marquez, F., Salameh, M., and Kiritchenko, S. (2018). Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, p. 1–17.

Mohammad, S. M. (2017). Challenges in sentiment analysis. In *A Practical Guide to Sentiment Analysis*, p. 61–83. Springer.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Mohammad, S. M., Salameh, M., and Kiritchenko, S. (2016). How translation alters sentiment. *J. Artif. Intell. Res.(JAIR)*, 55:95–130.

Mourad, A. and Darwish, K. (2013). Subjectivity and sentiment analysis of modern standard arabic and arabic microblogs. In *WASSA@ NAACL-HLT*, p. 55–64.

Mukherjee, S. and Bala, P. K. (2017). Detecting sarcasm in customer tweets: an nlp based approach. *Industrial Management & Data Systems*, 117(6):1109–1126.

Mulki, H., Haddad, H., Bechikh Ali, C., and Babaoğlu, I. (2018). Tunisian dialect sentiment analysis: A natural language processing-based approach. *Computación y Sistemas*, 22(4).

Mulki, H., Haddad, H., Gridach, M., and Babaoğlu, I. (2017). Tw-star at semeval-2017 task 4: Sentiment classification of arabic tweets. In *Proceedings of the 11th international workshop on semantic evaluation (SEMEVAL-2017)*, p. 664–669.

Nabil, M., Aly, M. A., and Atiya, A. F. (2015). Astd: Arabic sentiment tweets dataset. In *EMNLP*, p. 2515–2519.

Nakov, P. (2017). Semantic sentiment analysis of twitter data. *CoRR*, abs/1710.01492.

Northwestern, U. (2016). Media industries in the middle east. `http://www.mideastmedia.org/industry/2016/`. (last visited on 12-04-2019).

Oussous, A., Lahcen, A. A., and Belfkih, S. (2018). Improving sentiment analysis of moroccan tweets using ensemble learning. In *International Conference on Big Data, Cloud and Applications*, p. 91–104. Springer.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, p. 380–390.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, p. 1094–1101.

Pasha, M. U. R. and others (2016). *An analysis of word sense disambiguation in Bangla and English using supervised learning and a deep neural network classifier*. PhD thesis, BRAC University.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., and others (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543.

Piryani, R., Madhavi, D., and Singh, V. K. (2017). Analytical mapping of opinion mining and sentiment analysis research during 2000–2015. *Information Processing & Management*, 53(1):122–150.

Plisson, J., Lavrac, N., Mladenić, D., and others (2004). A rule based approach to word lemmatization.

Popescu, A.-M. and Etzioni, O. (2007). Extracting product features and opinions from reviews. In *Natural language processing and text mining*, p. 9–28.

Poria, S., Cambria, E., Hazarika, D., and Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*.

Porter, M. and Boulton, R. (2002a). Snowball english stemmer. `http://snowballstem.org/algorithms/english/stemmer.html`. (last visited on 22-05-2019).

Porter, M. and Boulton, R. (2002b). Snowball english stopwords. `http://snowball.tartarus.org/algorithms/english/stop.txt`. (last visited on 22-05-2019).

Porter, M. and Boulton, R. (2002c). Snowball spanish stemmer. `http://snowballstem.org/algorithms/spanish/stemmer.html`. (last visited on 22-05-2019).

Porter, M. and Boulton, R. (2002d). Snowball spanish stopwords. `lhttp://snowball.tartarus.org/algorithms/spanish/stop.txt`. (last visited on 22-05-2019).

Porter, M. and Boulton, R. (2002e). Snowball stemming algorithms. `http://snowballstem.org`. (last visited on 22-05-2019).

Pozzi, F. A., Fersini, E., Messina, E., and Liu, B. (2016). *Sentiment analysis in social networks*. Morgan Kaufmann.

Pozzi, F. A., Fersini, E., Messina, E., and Liu, B. (2017). Challenges of sentiment analysis in social networks: An overview. In *Sentiment analysis in social networks*, p. 1–11. Elsevier.

Rambocas, M., Gama, J., and others (2013). Marketing research: The role of sentiment analysis. Technical report, Universidade do Porto, Faculdade de Economia do Porto.

Refaee, E. (2017). Sentiment analysis for micro-blogging platforms in arabic. In *International Conference on Social Computing and Social Media*, p. 275–294. Springer.

Refaee, E. and Rieser, V. (2014). An arabic twitter corpus for subjectivity and sentiment analysis. In *LREC*, p. 2268–2273.

Reyes, A. and Rosso, P. (2014). On the difficulty of automatically detecting irony: beyond a simple case of negation. *Knowledge and Information Systems*, 40(3):595–614.

Ringsquandl, M. and Petkovic, D. (2013). Analyzing political sentiment on twitter. In *AAAI Spring Symposium: Analyzing Microtext*, p. 40–47.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, p. 502–518.

Rout, J. K., Choo, K.-K. R., Dash, A. K., Bakshi, S., Jena, S. K., and Williams, K. L. (2018). A model for sentiment and emotion analysis of unstructured social media text. *Electronic Commerce Research*, 18(1):181–199.

Ruiz-Martínez, J. M., Valencia-García, R., García-Sánchez, F., and others (2012). Semantic-based sentiment analysis in financial news. In *Proceedings of the 1st International Workshop on Finance and Economics on the Semantic Web*, p. 38–51.

Rushdi-Saleh, M., Martín-Valdivia, M. T., Ureña-López, L. A., and Perea-Ortega, J. M. (2011). Oca: Opinion corpus for arabic. *Journal of the Association for Information Science and Technology*, 62(10):2045–2054.

Saif, H., Fernández, M., He, Y., and Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of twitter.

Saif, H., He, Y., and Alani, H. (2012). Semantic sentiment analysis of twitter. In *International semantic web conference*, p. 508–524. Springer.

Saif, H., Ortega, F. J., Fernández, M., and Cantador, I. (2016). Sentiment analysis in social streams. In *Emotions and Personality in Personalized Services*, p. 119–140. Springer.

Salamah, J. B. and Elkhlifi, A. (2016). Microblogging opinion mining approach for kuwaiti dialect. *Computing Technology and Information Management*, 1(1):9.

Salameh, M., Mohammad, S., and Kiritchenko, S. (2015a). Sentiment after translation: A case-study on arabic social media posts. In *HLT-NAACL*, p. 767–777.

Salameh, M., Mohammad, S., and Kiritchenko, S. (2015b). Sentiment after translation: A case-study on arabic social media posts. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, p. 767–777.

Samih, Y., Attia, M., Eldesouki, M., Abdelali, A., Mubarak, H., Kallmeyer, L., and Darwish, K. (2017). A neural architecture for dialectal arabic segmentation. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, p. 46–54.

Sarkar, D. (2016). *Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data*. Apress.

Satapathy, R., Cambria, E., and Hussain, A. (2018). *Sentiment Analysis in the Bio-Medical Domain: Techniques, Tools, and Applications*, volume 7. Springer.

Satapathy, R., Guerreiro, C., Chaturvedi, I., and Cambria, E. (2017). Phonetic-based microtext normalization for twitter sentiment analysis. In *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*, p. 407–413. Institute of Electrical and Electronics Engineers (IEEE).

Sayadi, K., Liwicki, M., Ingold, R., and Bui, M. (2016). Tunisian dialect and modern standard arabic dataset for sentiment analysis : Tunisian election context. In *To appear in the ACLing 2016 IEEE proceedings*. CICLING.

Schmid, H. (1999). Improvements in part-of-speech tagging with an application to german. In *Natural language processing using very large corpora*, p. 13–25.

Semiocast (2011). Arabic highest growth on twitter. `https://semiocast.com/publications/2011_11_24_Arabic_highest_growth_on_Twitter`. (last visited on 12-04-2019).

Sharif, W., Samsudin, N. A., Deris, M. M., and Naseem, R. (2016). Effect of negation in sentiment analysis. In *Innovative Computing Technology (INTECH), 2016 Sixth International Conference on*, p. 718–723. IEEE.

Shayaa, S., Ainin, S., Jaafar, N. I., Zakaria, S. B., Phoong, S. W., Yeong, W. C., Al-Garadi, M. A., Muhammad, A., and Zahid Piprani, A. (2018). Linking consumer confidence index and social media sentiment analysis. *Cogent Business & Management*, 5(1):1–12.

Shen, D., Wang, G., Wang, W., Min, M. R., Su, Q., Zhang, Y., Li, C., Henao, R., and Carin, L. (2018). Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL).*

Shoukry, A. and Rafea, A. (2012a). Preprocessing egyptian dialect tweets for sentiment mining. In *The Fourth Workshop on Computational Approaches to Arabic Script-based Languages*, p. 47.

Shoukry, A. and Rafea, A. (2012b). Sentence-level arabic sentiment analysis. In *2012 International Conference on Collaboration Technologies and Systems (CTS)*, p. 546–550. Institute of Electrical and Electronics Engineers (IEEE).

Sirsat, S. R., Chavan, V., and Mahalle, H. S. (2013). Strength and accuracy analysis of affix removal stemming algorithms. *International Journal of Computer Science and Information Technologies*, 4(2):265–269.

Sitsanis, N. (2018). Top 10 languages used on the internet today. `https://speakt.com/top-10-languages-used-internet/`. (last visited on 12-04-2019).

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, p. 1631–1642.

Soliman, A. B., Eissa, K., and El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Sumanth, C. and Inkpen, D. (2015). How much does word sense disambiguation help in sentiment analysis of micropost data? In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, p. 115–121.

Sun, F., Belatreche, A., Coleman, S., McGinnity, Y. L. M., Li, Y., and others (2012). Evaluation of libsvm and mutual information matching classifiers for multi-domain sentiment analysis. In *The 23rd Irish Conference on Artificial Intelligence and Cognitive Science. Dublin*.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Taghva, K., Elkhoury, R., and Coombs, J. (2005). Arabic stemming without a root dictionary. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 1, p. 152–157. Institute of Electrical and Electronics Engineers (IEEE).

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, p. 1555–1565.

Thelwall, M. (2017). The heart and soul of the web? sentiment strength detection in the social web with sentistrength. In *Cyberemotions*, p. 119–134. Springer.

Thelwall, M., Buckley, K., and Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. *Icwsm*, 10(1):178–185.

Tungthamthiti, P., Kiyoaki, S., and Mohd, M. (2014). Recognition of sarcasms in tweets based on concept level sentiment analysis and supervised learning approaches. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, p. 417–424.

Uysal, A. K. and Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112.

White, L., Togneri, R., Liu, W., and Bennamoun, M. (2015). How well sentence embeddings capture meaning. In *Proceedings of the 20th Australasian Document Computing Symposium*, p. 9. ACM.

Wiebe, J. M. (1994). Tracking point of view in narrative. *Comput. Linguist.*, 20(2):233–287.

Wiegand, M., Balahur, A., Roth, B., Klakow, D., and Montoyo, A. (2010). A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, p. 60–68.

Yadav, S., Ekbal, A., Saha, S., and Bhattacharyya, P. (2018). Medical sentiment analysis using social media: Towards building a patient assisted system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Yang, B. and Cardie, C. (2014). Context-aware learning for sentence-level sentiment analysis with posterior regularization. In *ACL (1)*, p. 325–335.

Yang, J., Jiang, L., Wang, C., and Xie, J. (2014). Multi-label emotion classification for tweets in weibo: Method and application. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, p. 424–428.

Yasavur, U., Travieso, J., Lisetti, C. L., and Rishe, N. D. (2014). Sentiment analysis using dependency trees and named-entities. In *FLAIRS Conference*.

Yıldırım, E., Çetin, F. S., Eryiğit, G., and Temel, T. (2015). The impact of nlp on turkish sentiment analysis. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 7(1):43–51.

Yu, J., Marujo, L., Jiang, J., Karuturi, P., and Brendel, W. (2018). Improving multi-label emotion classification via sentiment classification with dual attention transfer network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, p. 1097–1102.

Zahran, M. A., Magooda, A., Mahgoub, A. Y., Raafat, H. M., Rashwan, M., and Atyia, A. (2015). Word representations in vector space and their applications for arabic. In *CICLing (1)*, p. 430–443.

Zaidan, O. F. and Callison-Burch, C. (2014). Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, M. and Zhou, Z. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837.

Zhang, W., Xu, M., and Jiang, Q. (2018). Opinion mining and sentiment analysis in social media: Challenges and applications. In *International Conference on HCI in Business, Government, and Organizations*, p. 536–548. Springer.

RESUME

Personal Information

Name and Surname: Hala MULKI

Nationality: Syrian/Turkish

Place and Date of Birth: Aleppo, Syria, 01/01/1980

Telephone: 0090-537 048 46 01

E-mail: hallamulki@gmail.com

Education

Degree	Institution/Province	Year
High School	Aleppo College-American School Aleppo, Syria	1998
BSc.	Computer Engineering Faculty Aleppo University, Syria	2003
MSc.	Computer Engineering Faculty Aleppo University, Syria	2009
Ph.D.	Computer Engineering Faculty Selçuk University, Turkey	2019

Work Experience

Year	corporation	Position
2013-2014	RedR UK Gaziantep, Turkey	IT Consultant English/Arabic Interpreter
2004-2012	Aleppo University, Syria	Research/Teaching Assistant

Technical Skills

- **Machine/Deep Learning:** Scikit-Learn, LibSvm, numpy, Keras, TensorFlow, MS Azure.
- **NLP:** NLTK, TextBlob, gensim, Multi-lingual text manipulation tools and t-SNE.
- **Tools & Programming:** Python, Matlab, R, Java, Perl, SQL/PL-SQL.

Foreign Languages

English, Turkish

PUBLICATIONS

International peer-reviewed journals

1. A.A. Altun and **H. Mulki.** Multistage Filtering Algorithm for Salt and Pepper Noise Removal from Highly Corrupted Microscopic Blood Images. In International Research Journal of Electronics and Computer Engineering, Vol.2, No.4 , pp.11–16.2016. ISSN: 2412-4370. 2016

2. **H. Mulki**, H. Haddad, I. Babaoglu. Modern Trends in Arabic Sentiment Analysis: A Survey. In RAITEMENT AUTOMATIQUE DES LANGUES, Vol.58, No.3 , pp.15–39. ISSN: 12489433. 2017.

3. **H. Mulki**, C. Bechikh Ali, H. Haddad and I. Babaoglu. Tunisian Dialect Sentiment Analysis: A Natural Language Processing-based Approach. In Computacion y Sistemas, Vol.22, No.4. ISSN:1405-5546. 2018.

4. **H. Mulki**, H. Haddad, M. Gridach and I. Babaoglu. Empirical Evaluation of Leveraging Named Entities for Arabic Sentiment Analysis. Accepted at International Arab Journal of Information Technology, Zarqa University, Jordan.

5. **H. Mulki**, H. Haddad, M. Gridach and I. Babaoglu. Syntax-Ignorant n-gram Embeddings for Dialectal Arabic Sentiment Analysis. Accepted at Natural Language Engineering Journal, Cambridge University Press, Cambridge, UK.

Papers in peer-reviewed international conference proceedings

1. **H. Mulki**, H. Haddad, M. Gridach and I. Babaoglu. Syntax-Ignorant N-gram Embeddings for Sentiment Analysis of Arabic Dialects. In WANLP 2019 co-located with ACL 2019, Florence, Italy, July 28-Aug 2, 2019.

2. **H. Mulki**, H. Haddad, C. Bechikh Ali and H. Alshabani. L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language. In ALW 2019 co-located with ACL 2019, Florence, Italy, July 28-Aug 2, 2019.

3. **H. Mulki**, C. Bechikh Ali, H. Haddad, I. Babaoglu. Tw-StAR at SemEval-2019 Task 5: N-gram embeddings for Hate Speech Detection in Multilingual Tweets. Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval'19) co-located with NAACL 2019, Minneapolis, USA, June 6-7, 2019.

4. **H. Mulki**, H. Haddad, C. Bechikh Ali, I. Babaoglu. Tw-StAR at SemEval-2018 Task 1: Affect in Tweets. In Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval'18) co-located with NAACL 2018, New Orleans, USA, June 5-6, 2018.

5. C. Bechikh Ali, **H. Mulki**, H. Haddad. Impact du prétraitement sur lÁnalyse de Sentiment du Dialecte Tunisien. In 25th French Conference on Natural Language Processing (Traitement Automatique des Langues Naturelles – TALN), Rennes, France, May 15-18, 2018.

6. **H. Mulki** , H. Haddad, C. Bechikh Ali, I. Babaoglu. Preprocessing Impact on Turkish Sentiment Analysis. In IEEE 26th Signal Processing and Communications Applications Conference (SIU2018), Izmir, Turkey, May 02-05, 2018.

7. **H. Mulki**, C. Bechikh Ali, H. Haddad, I. Babaoglu. Tunisian Dialect Sentiment Analysis: A Natural Language Processing-based Approach. 18th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2018), LNCS, Hanoi, Vietnam, Mars 18-24, 2018.

8. M. Gridach, H. Haddad, **H. Mulki.** Empirical Evaluation of Word Representations on Arabic Sentiment Analysis. In Proceedings of the 6th International Conference on Arabic Language Processing (ICALP 2017), CCIS, Fez, Morocco, pages 147-158, October 11-12 2017.

9. H. Haddad , **H. Mulki.** Empirical Evaluation of Preprocessing on Tunisian Dialect ("Darija") Sentiment Analysis. In Proceedings of the second International Widening NLP Workshop (WiNLP '18), co-located with NAACL 2018, New Orleans, USA, June 1, 2018.

10. M. Gridach, H. Haddad, **H. Mulki.** Churn Identification in Microblogs using Convolutional Neural Networks with Structured Logical Knowledge. In Proceedings of the 3rd Workshop on Noisy User generated Text (W-NUT) at EMNLP 2017, Copenhagen, Denmark, September 7, 2017.

11. **H. Mulki**, H. Haddad, M. Gridach, I. Babaoglu. Tw-StAR at SemEval-2017 Task 4: Sentiment Classification of Arabic Tweets. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval'17). Association for Computational Linguistics (ACL), pages 655-660, Vancouver, Canada, August 3-4, 2017.