# RESOURCES AND APPLICATIONS *for* DIALECTAL ARABIC

## The case of Levantine

UNIVERSITY OF
GOTHENBURG

CHATRINE QWAIDER

# Resources and Applications for Dialectal Arabic

## The case of Levantine

UNIVERSITY OF
GOTHENBURG

# Resources and Applications for Dialectal Arabic

## The case of Levantine

*Chatrine Qwaider*

Doctoral thesis in Computational Linguistics
The Centre for Linguistic Theory and Studies in Probability (CLASP)
Department of Philosophy, Linguistics and Theory of Science (FLoV)
University of Gothenburg, Sweden
May 2022

©**Chatrine Qwaider (Kathrein Abu Kwaik)** 2022

Resources and Applications for Dialectal Arabic
Doctoral thesis
Main Supervisor: Stergios Chatzikyriakidis
Co-supervisor: Simon Dobnik

Cover: designed by Fiverr International Ltd.
Printer: Stema Specialtryck AB
Publisher: University of Gothenburg (Dissertations)
ISBN 978-91-8009-803-8(print)
ISBN 978-91-8009-804-5(pdf)

SVANENMÄRKET
Trycksak
3041 0234

The author is supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg.

Distribution
Department of Philosophy, Linguistics and Theory of Science (FLoV)
Box 100, SE-405 30 Gothenburg

# Abstract

This is a thesis about the computational study of Dialectal Arabic (DA). In particular, the thesis studies DA, with a special emphasis on Levantine Arabic, and develops tools and resources for the computational study of Dialectal Arabic Natural Language Processing (DANLP). It investigates the creation of fine-grained resources that can be used for a variety of computational tasks, and a number of effective models that can deal with the complexity of fine-grained dialectal data. Dialect Identification (DI), as well as Sentiment Analysis (SA) are the Natural Language Processing (NLP) tasks investigated in this thesis.

In the first part (Study 1 and Study 2), I study the DI task on both coarse-grained and fine-grained levels. For this reason, I build the first annotated Levantine (SHAMI) Dialect Corpus (SDC). Furthermore, I explore the ability of n-gram language models, Machine Learning (ML) algorithms and ensemble learning techniques to classify and detect 26 Arabic varieties. In the second part, I conduct a linguistic study to measure the lexical distance between MSA and DA, and between the dialects themselves. This is done to check whether transferring knowledge from one variety to another is possible. In the third part, studies 4,5 and 6, I explore Arabic Sentiment Analysis (SA). I investigate the idea of knowledge transfer between MSA and the dialects using SA as a case study. Furthermore, I implement various models such as the pre-trained language model BERT, Deep Learning (DL), ML and feature engineering approaches to detect the sentimental polarity of DA data. I introduce two valuable resources for this task, one focusing on Levantine sentiment (Shami-Senti), and the other for DA in general (AT-SAD). I exploit different ways of annotation, e.g. human, lexicon-based and automatic distant supervision annotation. The last study is about choosing the best model for DI and SA. I exploit well-known models and approaches using various kinds of DA resources.

The thesis contributes to the field of DANLP in a number of ways. The introduced valuable resources can be seen as a stepping stone for a deeper investigation and understanding of issues in DANLP. They are also reliable and can be used by researchers to address different NLP tasks. The cross-dialectal linguistic studies will open up prospects for researchers to fine-tune models and transfer knowledge among Arabic varieties. A big part of the contribution lies in designing DI and SA models. I implement several ML models that use feature engineering approaches and N-gram language models to identify the dialect or detect the sentiment. For DI, I design and implement an ensemble learning model that is able to handle fine-grained dialects. Additionally, I exploit the usage of DL models on different SA dialectal datasets and achieve competitive results. For both tasks, I exploit

the recent pre-trained language models and perform a comparison to choose the best model. I also implement a semi-supervised approach for automatic labelling and annotating data with the help of self-training techniques to improve the performance of the dataset. These models will help researchers dive deeper into DANLP and create practical and industrial systems.

# Abstrakt

Denna uppsats är en serie av datorlingvistiska studier av Dialektal Arabiska (DA). Jag undersöker DA, med fokus på Levantinska Arabiska och utvecklar verktyg samt resurser för datorlingvistiska studier av dialektal Arabiska (DANLP). I uppsatsen undersöker vi resurser som kan användas i många olika syften, och datormodeller som kan hantera komplex dialektal Arabiska. Studierna som presenteras undersöker dialektidentifikation (DI) och sentiment analysis (SA).

I den första delen (Studie 1 och 2) studerar vi DI både på en generisk och specifik nivå. För detta bygger vi SHAMI-korpuset. Den första studien undersöker denna korpus med en språkmodell baserad på n-gram och sammansättning av modeller för att klassificera 26 olika Arabiska dialekter. I den andra delen gör vi en lingvistisk analys för att mäta lexikal distans mellan Modern Standardarabiska (MSA) och dialektal Arabiska samt mellan de olika arabilska dialekterna. Detta görs för att undersöka huruvida vi kan föra över kunskap från en dialekt till en annan dialekt. I den andra delen (studie 3, 4, och 5) undersöker vi sentimentanalys. Vi undersöker och det går att överföra kunskap mellan MSA och andra Arabiska dialekter som en fallstudie Jag implementerar även flera olika maskininlärnings modeller, så som BERT, och undersöker huruvida särdragstekniker kan användas för att predikera polaritet hos sentiment för dialektal Arabiska.Jag introducerar två resurser för detta, en som fokuserar på sentiment i Levantinska dialekter (Shami-Senti) och en annan för andra Arabiska dialekter (ATSAD). Jag använder mig av olika annoteringstekniker: mänskliga annoterare, ordböcker, och automatisk distans övervakning. Den sista studien handlar om hur vi kan välja den bästa modellen för DI och SA. Vi undersöker kända modeller och tekniker för detta och utnyttjar olika DA resurser.

Denna uppsats bidrar till fältet DANLP på många sätt. Vi introducerar ett antal värdefulla resurser för dialektal Arabiska som kan ses som ett första steg mot djupare undersökningar för forskning inom DANLP. Resurserna är också robusta och kan användas för många olika uppgifter inom datorlingvistik. De kors-dialektala lingvistiska studierna öppnar upp för forskning inom justering av förtränade modellers samt överförande av kunskap från en dialekt till en annan. En stor del av mitt bidrag ligger i designen av olika modeller för DI och SA. Jag implementerar flera olika modellers som använder särdragstekniker och n-grams språkmodeller som kan identifiera arabisk dialekt och sentiment. För DI så designar jag och implementerar en sammansättnings modell som kan hantera dialekter på detaljnivå. Yttligare så använder jag mig av djupinlärnings modeller för dialektal arabisk sentimentanalys och får bra resultat. För både DI och SA så använder jag mig av för-tränade språkmodeller och utvärderar dem för att välja den bästa modellen. Jag implementerar även en lätt-övervakad modell för automatisk annotering med hjälp av själv-övervakade tekniker som förbättrar resultatet för korpuset. Dessa modeller kan hjälpa forskare att dyka djupare in i DANLP för att skapa praktiska och industriella system.

To the soul of my father
My first leader and inspiration
The one who taught me my first letters
The one who enlightened me on the path of knowledge
Because of him I am here today

# Acknowledgments

First and foremost, I praise and thank Allah SWT for guiding me and taking care of me all the time. My life is so blessed because of his majesty.

I would like to take this opportunity to offer my sincerest gratitude to my supervisor, **Prof. Stergios Chatzikyriakidis**, for giving me the opportunity to work under his supervision for more than five years, guiding and helping me throughout this research. I am also thankful for my co-supervisor **Prof. Simon Dobnik**, for his valuable comments and support. Those meetings and the many discussions we went through, helped me and gave me the ability to continue and refine my academic personality.

Big thanks to **Prof. Shalom Lappin**. Thank you for reminding me of my strengths. Our discussions are encouraging me to continue my hard work.

I am grateful to **Dr Richard Johanson** from Chalmers University of Technology and **Dr Motaz Saad** from the Islamic University of Gaza, for their fruitful collaboration during the research.

I want to express my gratitude and thanks to all the members of CLASP as well as FloV from Göteborg University; I really had a great time and enjoyed every single moment.

I am very grateful to my dear husband **Dr Mithqal Qwaider**. Thank you for believing in me and encouraging me through the journey. Thank you is a small word to express my gratitude for your endless support. My beloved children (**Zein**, **Bana** and **Khaled**), you are the most beautiful part of my journey, so thank you very much.

I would like to thank everyone who believed in and supported me. My family, my family in law, my cousins, my friends in Gaza. Your prayers and support mean a lot to me. It was like a light shining out of the darkness.

I wish to express my considerable gratitude to many people who, in one way or another, have helped with the process of doing this research.

Thank you all for being always there when I needed you most. Thank you for believing in me and supporting me. I believe that without your support and your prayers, none of this work would have been accomplished.

Finally, I hope that this thesis will be a useful addition to research in Arabic natural language processing.

# List of appended papers

## Study 1: Towards a Levantine corpus (Shami) and Dialect Identification

*Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis and Simon Dobnik . "Shami: A corpus of levantine Arabic dialects." In proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). 2018*

Chatrine Qwaider (Kathrein Abu Kwaik) had the main responsibility for collecting the data manually, pre-processing and cleaning the whole corpus, implementing the *scikit-learn* models, conducting all the experiments and reporting on them. I had a collaboration with Motaz Saad from the Islamic University of Gaza. Motaz was responsible for collecting the data from Twitter and implementing the model using off-the-shelf python library for Dialect Identification (*Langid.py*). All authors had shared responsibility for the remaining aspects of this research. All authors read and approved the final manuscript.

## Study 2: Investigate Language Modelling and Ensemble Learning for Fine-Grained Arabic Dialect Identification

*Kathrein Abu Kwaik and Motaz K Saad. "ArbDialectID at MADAR Shared Task 1: Language Modelling and Ensemble Learning for Fine Grained Arabic Dialect Identification." In ArbDialectID at MADAR Shared Task 1: Language Modelling and Ensemble Learning for Fine-Grained Arabic Dialect Identification. In proceedings of the Fourth Arabic Natural Language Processing Workshop (2019)*

Chatrine Qwaider (Kathrein Abu Kwaik) had the main responsibility for implementing the Language models and Ensemble learning with *scikit-learn*, conducting all the experiments and reporting on them. All authors had shared responsibility for the remaining aspects of this research. All authors read and approved the final manuscript.

## Study 3: Computational Cross Dialectal Lexical Distance Study

*Kwaik, Kathrein Abu, Motaz Saad, Stergios Chatzikyriakidis and Simon Dobnik. "A Lexical Distance Study of Arabic Dialects." Procedia computer science 142, (2018): pp. 2-13.*

Chatrine Qwaider (Kathrein Abu Kwaik) had the main responsibility for establishing the experiments, implementing by Python, *scikit-learn* and Gensim libraries and then reporting the result. Motaz Saad contribute with his Wiki-corpus. All authors had shared responsibility for the remaining aspects of this research. All authors read and approved the final manuscript.

## Study 4: The usability of MSA NLP tools for DA

*Chatrine Qwaider, Stergios Chatzikyriakidis and Simon Dobnik. "Can Modern Standard Arabic Approaches be used for Arabic Dialects? Sentiment Analysis as a Case Study." In proceedings of the 3rd Workshop on Arabic Corpus Linguistics, pp. 40-50. 2019.*

Chatrine Qwaider (Kathrein Abu Kwaik) had the main responsibility for creating Shami-Senti, implementing the language models, conducting all the experiments and reporting on them. All authors had shared responsibility for the remaining aspects of this research. All authors read and approved the final manuscript.

## Study 5: Investigate the performance of Deep Learning methods for Dialectal Arabic Sentiment Analysis

*Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis and Simon Dobnik. "LSTM-CNN Deep Learning Model for Sentiment Analysis of Dialectal Arabic." In proceedings of the International Conference on Arabic Language Processing, pp. 108-121. Springer, Cham, 2019.*

Chatrine Qwaider (Kathrein Abu Kwaik) had the main responsibility for building the baselines, the off-the-shelf Kaggle network proposed deep learning network, conducting all the experiments and reporting on them.

Motaz Saad suggested the usage of the winner Kaggle network. All authors had shared responsibility for the remaining aspects of this research. All authors read and approved the final manuscript.

## Study 6: Investigate Distant supervision and Self training approaches on Dialectal Arabic Sentiment Analysis

*Kathrein Abu Kwaik, Stergios Chatzikyriakidis, Simon Dobnik, Motaz Saad and Richard Johansson. "An Arabic Tweets Sentiment Analysis Dataset (ATSAD) using Distant Supervision and Self Training." In proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 1-8. 2020.*

Chatrine Qwaider (Kathrein Abu Kwaik) had the main responsibility for pre-processing the data, monitoring the annotators, implementing the double checked self-training model, conducting all the experiments and reporting them. Motaz Saad was responsible for collecting the dataset and applying the distant supervision on it. Richard Johansson helped with explaining and suggesting self training approaches and supervising all the experiments. All authors had shared responsibility for the remaining aspects of this research. All authors read and approved the final manuscript.

## Study 7: Comparing the Pre-trained language models and the feature-based approaches on Dialect Identification and Sentiment Analysis[1]

*Kathrein Abu Kwaik, Stergios Chatzikyriakidis and Simon Dobnik "Pre-trained models or feature engineering? The case of Arabic Dialectal Identification and Sentiment Analysis"*
Chatrine Qwaider (Kathrein Abu Kwaik) had the main responsibility for implementing the BERT models, feature-based approaches and the DL models. Chatrine was also responsible for conducting all the experiments and reporting on them. All authors had shared responsibility for the remaining aspects of this research. All authors read and approved the final manuscript.

[1] under review

# Contents

# 1 Introduction

Arabic is the official/co-official spoken language in 23 countries. However, it is not the native language tongue of any speaker of Arabic. One can do a broad classification of Arabic into three main varieties: Classical Arabic (CA), Modern Standard Arabic (MSA) and Dialectal Arabic (DA, colloquialism). CA is the form used by the Holy Quran, pre-Islamic scripts, and was also in the Umayyad and Abbasid literary texts from 7th century AD to the 9th century AD (early Islamic Literature)[1]. DA, on the other hand, is used in everyday communication as well as informal writing, e.g. in social media[2]. Both CA and DA originated before Islam. The Quran is based on the variety of Arabic used by Quraysh, an influential tribe in Mecca at that time. Modern Standard Arabic (MSA) is the literary standard across the Middle East, North Africa and the Horn of Africa. It is the official language of all Arab League countries and it is the form used for education, news, politics, religion and, in general, for any type of formal interaction. However, people in the individual Arabic-speaking countries communicate orally and in writing in their own dialects. According to the Ethnologue 2020 statistics, the number of Arabic speakers who master at least one Arabic dialect as their first language is about 350 million,[1] while there are 270 million Arabs who know the official Arabic language. In Chapter 2, more information about these varieties will be given.

As a result of the rapid technological development, and with the increased usage of the Internet among Arabic speakers, Arabic content online has been drastically increasing. This led to a need for building Natural Language Processing (NLP) applications that are capable of handling the Arabic language in both its standard and dialectal form. Initially, NLP

[1] https://www.ethnologue.com/language/ara

researchers focused on MSA in terms of building computational resources and applications. Recently, with the emergence of social media platforms, the need to develop computational resources for DA has become necessary. Indeed, dialectal resources, models and applications for DA have started to be developed, but given that dialects are not standardized in terms of the writing form, and, as such, a lot of variation can be found across various countries and cities, this task becomes very difficult, and a lot of effort is required to build resources and develop various NLP models to cope with DANLP tasks.

This thesis is broadly about the computational linguistics and NLP study of Dialectal Arabic. I address two DANLP tasks: Dialect Identification (DI) and Sentiment Analysis (SA). DI systems are usually classified into coarse-grained classification systems, where the classification takes place in the level of groups of dialects or regions, and fine-grained classification systems[3], where classification moves deeper to the level of the country, province, city or even more. An example of this difference will be a DI system classifying Levantine dialects and a system identifying the varieties spoken in different cities in Syria[4]. Sentiment Analysis refers to the classification of sentence polarity[5]. They can be classified according to objectivity and subjectivity[6], according to polarity (positive, negative, mixed or neutral)[7], and emotion extraction (fear, angry, joy, sadness, happy, inspired and so on)[8, 9]. There is also aspect SA, which categorizes opinions by aspect and identifies the sentiment related to each aspect[10].

More specifically, this thesis is about developing resources and applications for Arabic dialects, in general, and Levantine, specifically. Levantine is a sub-group of Arabic dialects spoken in Levant. It is spoken in Palestine, Syria, Lebanon and Jordan. According to Ethnology (2022) there are more than 40 million Levantine native speakers all over the world, making Levantine varieties second in terms of speakers after Egyptian Arabic[2][3]. The first part of the research presented in this thesis investigates the task of DI on both levels (coarse-grained and fine-grained). For this reason, I built Machine Learning models that are able to classify the Levantine group of dialects (4-dialects), as well as other models that are able to classify between 26 Arabic varieties spoken in 25 Arab cities plus MSA. For the purpose of the Levantine DI task, I collected and built a Levantine corpus that included the 4 dialects (Palestinian, Syrian, Jordanian and Lebanese). Then, I handled the task of SA both for Arabic dialects and Levantine dialects as a case study, I built various learning models, exploiting the Levantine corpus, and extracted sentences annotated for sentiment from it, in order to build the first Levantine SA corpus that includes data from the four Levantine

[2] https://www.ethnologue.com/language/apc

[3] https://www.ethnologue.com/language/ajp

dialects. In addition to this resource, I also built an SA dataset for DA to study the task in a broader manner.

## 1.1 Motivations

There are two main motivations behind doing this thesis, that are, however, interconnected to some extent. The first concerns myself as an Arabic native speaker, while the other concerns myself as an NLP researcher interested in Arabic NLP.

I came to Sweden in 2016 from the Levant. In that year, according to the statistics in Sweden, there were 116,384 citizens of Syria (70,060 men, 46,324 women) residing in Sweden.[4]Most of these people arrived as asylum seekers following the Syrian civil war, which began in 2011. These numbers increased in 2020 and reached 194,000 Syrian citizens having Syria as their place of birth, and a total of 50,620 who have at least one Syrian-born parent.[5] In addition to the Syrian refugees, there were also Palestinian refugees, as well as Lebanese and Jordanian immigrants, working here in Sweden.

These people, especially in their first years in Swedish, and before having mastered Swedish properly, tried to use Google Translate or other translation applications to communicate and translate between Swedish and Arabic. All of them used their own Arabic dialects either in speech mediums or text processing applications. As a native Levantine speaker, I meet so many non-educated people who cannot express their needs and thoughts in MSA or CA, but can do so well in their own dialects. For this reason, however, they face many problems and frequently ask for the help of a translator or an interpreter anywhere they go in the country. These situations motivated me to further study the Arabic Dialect Identification task, especially for Levantine dialects, given that, as a Levantine native speaker, I am interested to detect the differences among the dialects of the Levant. Therefore, fine-grained Levantine DI is one of the issues I am investigating in this thesis.

Moreover, because of the political developments in my country, whether this is the recent conflict in Gaza 2021, the Syrian civil war, or in general, the turbulent events across the Levant, I used to browse the web to see how people feel towards these issues and others. People use the social media platforms to express and share their feelings, opinions and sentiments with regard to various situations including politics, economy, products and life in general. This too motivated me greatly to complete the research on the

[4] http://www.statistikdatabasen.scb.se/pxweb/en/ssd/

[5] statista.com/topics/7687/migration-and-integration-in-sweden/

subject of SA, whether at the level of Levantine dialects, or Arabic dialects in general, later.

The Arabic language ranks fourth by the number of Internet users after English, Chinese and Spanish.[6] Nowadays, as mentioned before, there are three Arabic varieties: CA, MSA and DA. While the DA is the spoken language all over the Arabic-speaking countries, one could always find dialectal data like audio or video recordings but not in text format. Recently, and with the increased use of social media, Arabs use their dialects while posting, tweeting and socializing. This means the availability of dialectal data on the web in terms of written resources. However, in order to be able to use the dialectal data, an NLP researcher should study the nature of the dialect, process it and build NLP applications for different usage like Machine Translation (MT), Part of Speech tagger (POS), Entity Recognition (ER) and others. Nonetheless, any application should be preceded by the process of identifying the dialect, which improves the performance of the desired NLP task. This matter motivates me as an NLP researcher to build models that help detect the dialects either on the country level, like Levantine DI, or in fine-grained level to detect the dialects of Arab cities.

My second interest as a researcher in Arabic NLP is to classify and detect feelings expressed in dialects. Most people use emotion, and a lot of discussions on social media on various issues can be found: these include political issues of the respective countries, discussions on the economy, or discussions about daily life matters, e.g. asking about a product or comparing schools or students' comments about exams and so on. That is, social media platforms have become a place for presenting opinions and discussions on a daily basis. Stakeholders such as companies, decision makers, and data analysts express their concerns about people and their opinions, so there is a constant need for SA software. SA applications can be used in many cases, such as determining the preferences of people towards a particular political party and can thus make predictions on election results or analyze the opinions of users of different products, so that, based on the results, companies can develop and improve production lines, which means more financial returns. However, SA applications are not only about customers' feelings towards small products, but can, furthermore, be used for more serious matters. Sentiment programs can analyze users' personalities, and thus can be used for issues of security, e.g. for maintaining state security and combating terrorism. Given that people use their dialects when posting on the Internet, my works in SA concern dialectal Arabic to a large extent. For this reason, I built a Levantine SA resource to use as a resource to train an SA model. A later step is to build much bigger SA datasets for all Arabic dialects.

[6] https://www.internetworldstats.com/stats7.htm

In addition, there are a number of reasons that can be the motivation for a researcher to study NLP – specifically to study DANLP with a special focus on the task of DI, SA and Lexical Distance.

The following are some of them:

- The scarcity of available dialectal Arabic resources (in the time I began my PHD in 2016). This is because most of the existing resources have either been built for the purpose of processing MSA, or to address the coarse-grained dialects. Two well-known workshops are conducted every year and both concern the ANLP, the Workshop on Arabic Natural Language Processing (WANLP)[7] and Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT),[8] where both were started in 2014.

- As my interest is the Levantine dialects, then in the time of my first work Shami corpus, there was not any other corpus that had all four Levantine dialects classified in the level of country. Thus, to build a Levantine DI model, I needed to build a four classes Levantine corpus that included the dialects from Palestine, Syria, Lebanon and Jordan (Study 1, page 99).

- Even though Levantine dialects are similar to some extent where they are mutually intelligible, many differences between them exist upon a closer look, for example in the level of morphology. As a consequence, the need for efficient DI systems that are capable of distinguishing between them efficiently has emerged.

- Some researchers and statistical NLP websites claim that Levantine is the closest vernacular variety to MSA in terms of lexicon[9][11]; this motivates me to study the lexical distance between the dialects and how far they are from MSA (Study 3, page 135).

- The desire to transfer knowledge and technology from the available MSA NLP tools and results to DANLP. In general, the desire to broaden MSA NLP to include under-resourced languages like Arabic dialects (Study 4, page 155). As the availability of ANLP tools, this motivates me to adapt these tools and fine-tune them to be able to handle and address DANLP. I also want also to transfer the knowledge between dialects, by building tools for some dialects and broadening these tools to be consistence with other dialects. This was a motivation for the second study (Study 2, page 123).

[7] https://aclanthology.org/venues/wanlp/

[8] https://scholar.google.com/citations?user=YmKaJR4AAAAJhl=enauthuser=2

[9] https://www.economist.com/middle-east-and-africa/2021/09/18/the-travails-of-teaching-arabs-their-own-language

- As mentioned before, SA is also under investigation in this thesis. Thus, I decided to build two SA resources. The first concerns the Levantine dialects, where there is a lack of SA corpus for Levantine and that already have our Levantine dialects corpus (Study 4, page 155), and the second is a SA corpus that includes all the Arabic dialects in general (Study 6, page 191). The task of SA considers hot and trendy topics, and there, NLP models should address the sentiment of the text regarding different polarities. Companies, social institutes and other parties are concerned with people's sentiments and opinions, and as such, good SA models for DA are needed. In addition, the availability of new technologies for researchers and developers such as the Twitter APIs,[10] makes it easier and less time consuming to build resources gathering a reasonable amount of available data. Using this technology, I can easily build a dialectal Arabic SA corpus (Study 6, page 191).

## 1.2 Research Questions

I address the following general research questions in the independent studies comprising the thesis:

- Is there any available Levantine corpus that includes data from the four dialects of the Levant? If not, then which tools and platforms could be used to collect and build such a Levantine corpus? (Study 1, page 99)

- What are the best feature combinations that could be extracted in order to build and train models for the Levantine DI task? (Study 1, page 99)

- Which features, algorithms and models could be used for very fine-grained DI systems to detect dialects among 25 Arab cities? (Study 2, page 123)

- What is the level of lexical similarity and divergence between MSA and dialects of Arabic? (Study 3, page 135)

- Depending on the answer to the previous question, is it possible to rely on the level of similarity and exploit the existing tools and NLP models of MSA, and fine-tune them to suit DANLP? (Study 4, page 155)

[10] https://developer.twitter.com/en/docs/twitter-api

- Is there any Levantine corpus built for the purpose of SA? How can I exploit our Levantine corpus to build an SA corpus? (Study 4, page 155)

- How could deep learning networks perform using different kinds of Arabic corpora (in terms of size, source of the data, included language/dialects and data-balancing)? (Study 5, page 175)

- Which solutions can be employed to collect and build a big corpus for dialectal Arabic SA, taking into consideration time and resource efficiency? (Study 6, page 191)

- Can I exploit semi-supervised approaches for the purpose of automatically annotating the corpus? Can I employ the self-training approach for labelling purposes? (Study 6, page 191)

- What kind of models can be used efficiently for under-resourced dialects? Is it machine learning approaches, deep learning networks, feature engineering, word embeddings or language models? (Study 7, page 211)

- Is there a way to choose the best models or are there different factors that influence the choice of the model?(Study 7, page 211)

## 1.3 Contribution

This thesis has made a number of contributions to the field of DANLP. The first main contribution is the development of resources for dialectal Arabic. Specifically, I built:

- The first Levantine corpus (SHAMI), which includes data from four dialects of the Levant: Palestinian, Syrian, Lebanese and Jordanian. The corpus is a valuable resource that can be used for different purposes like POS taggers, DI, morphological analysis and similarity studies, among other tasks. The way the four categories are labelled and classified in the corpus can help NLP researchers address many linguistic phenomena related to Levantine. (*Study 1*)

- The first SA corpus for Levantine based on SHAMI, which includes data from the four Levant countries. It is a three-way classification SA dataset (positive, negative and mix) and is considered a stepping stone for investigating Levantine SA. It addresses the differences between this group of dialects and other Arabic dialects. (*Study 4*)

- For the purposes of SA in Dialectal Arabic, as well as in order to investigate the task in more depth, I build An Arabic Tweets SA Dataset (ATSAD), including all the dialects from the Arabic speaking countries, using distant supervision and self-training. The corpus contains a gold standard data set that NLP researchers can use for evaluation purposes, as well as an automatic labelling SA dataset. (*Study 6*)

Using our dialectal resources, as well as other resources for DA, I study and contribute to the following linguistics tasks:

- Measuring the lexical distance and the relation between MSA and other dialects of Arabic, as well as the distance among dialects themselves. This can help NLP researchers to transfer knowledge among close dialects as well as assist linguists to understand the relations and differences among the dialects of concern (*Study 3*)
- The extent to which available MSA tools and models can be fine-tuned to be adapted for Dialectal Arabic. The degree of reliance on transfer knowledge between MSA and dialects as well as among dialects to each other. (*Study 4*)
- The factors which affect the choice of the best models for some DANLP tasks like DI and SA. I compare available and well-known approaches in the field of NLP, e.g. feature engineering approaches, DL and pre-trained language models. (*Study 7*)

In terms of the NLP tasks, our contributions can be summarized as follows:

- I build a number of ML models by applying language N-gram models for DI on both a coarse-grained and fine-grained level. Our models ranked the first in a shred task where I combined features engineering approach and ensemble learning to get the best results for DI of Arabic dialects. (*Study 1, Study 2*)
- I build various ML techniques for SA of dialectal Arabic, both for two-way and three-way classification. I contribute to different models by either employing machine learning on Levantine dialect or by building a DL network (LSTM-CNN) that can handle different sizes of datasets. In addition, I examined the usage of the pre-trained language model in Dialectal Arabic by exploiting different corpora. (*Studies 4 to 7*)
- I build and introduce a DL model that combines Bi-directional Long-Short Term Memory Networks (Bi-LSTM) with Convolutional Neural Networks (CNN) for the purpose of SA. The DL models outperform

many state-of-the-art models for some corpora. Moreover, it could handle small-size datasets. (*Study 5*)

- I build a semi-supervised approach for automatic labelling and annotating of huge data with the help of self-training techniques to improve the performance of the dataset. (*Study 6*)

- I investigate the recent pre-trained language models (BERT) on DANLP using a number of different datasets, as well as feature engineering-based approaches for both DI and SA, and compare them. (*Study 7*)

## 1.4 Research Studies summary

In **Study 1**, page 99, I focus on Levantine dialects which are spoken in Palestine, Jordan, Lebanon and Syria and build the first Levantine (Shami) Dialects corpus (SDC). I collect the data in two ways: automatic and manual collection. A number of pre-processing steps are applied without affecting the semantic meaning of the dialectal data. The total size of the whole collected corpus is 140K Levantine sentences. DI has been chosen as the task to evaluate the corpus. I analyze the impact of using a language model for DI and apply various n-grams model using different libraries. At the same time, I compare the SDC corpus with other well-known corpora in the field that contain dialects from the Levant. Several experiments are conducted in terms of corpus size, libraries used and number of dialects in classification. Given that the dialects that I am focusing on are very similar, our experiments also give new insights in respect of language identification.

In **Study 2**, page 123, I dive deeply into the fine-grained DI task. This was a shared task, where I have been requested to use two datasets, MADAR-5 (contains 5 dialects from 5 Arab regions +MSA) and MADAR-25 (contains 25 dialects for 25 Arab cities +MSA). In this study I address the ability of language models to distinguish between very similar dialects and, thus, perform fine-grained DI tasks. I address the problem of DI for all Arabic dialects on the level of Arab cities. Moreover, I examine how can ML classifiers work in conjunction with N-gram models and other features. The classification models consist of three parts:

- a coarse-grained identification model to classify each sentence into one out of six varieties ( 5 five dialects plus MSA) using MADAR-5 corpus.

- a fine-grained model that classifies the sentence among 26 varieties (25 dialects from different Arab cities + MSA). This model has the coarse-grained predicted label as well as N-gram features as its input.

- I apply ensemble voting leaning on both subsystems.

To be able to transfer knowledge from MSA to DA or between different dialects, so it is less time consuming in terms of building new tools for DA, I do **Study 3**, page 135. I study the similarities and differences cross-dialectally. The study focuses on lexical distance by exploiting different corpora, as I work on textual data. In addition to the qualitative study, I conduct a quantitative measurement and apply various distance metrics such as the vector space model (VSM) based on word distribution over documents (as common in information retrieval (IR) [12]), latent semantic indexing (LSI) [13], the divergence distance algorithm as Hellinger distance (HD) [14] and others. Considering the available resources, I measure the lexical distance between the MSA and other dialects. On the other hand, I measure the lexical distance among the dialects themselves. The study helps us to shed some light on the written form of the dialects which differs from the spoken form. This study can be seen as a basis for building NLP tools for dialectal processing by adapting what already exists for MSA and focusing on areas of similarity and degrees of difference. The study is the most extensive of its kind concerned with measuring similarities and differences in the field of Arabic and dialectal Arabic, and represents a basis for new, similar investigations, focusing on other criteria such as phonological distance, morphological distance and semantic distance.

In this part of my work I shift to the second task, sentiment analysis (SA). SA is considered a semantic task which, with DI (lexical task), could cover different aspects in NLP. They together form a union of characteristics of DANLP. Based on the results from the previous study and the SHAMI corpus I have built in Study 1, I conduct **Study 4**, page 155. I build a Levantine SA corpus (SHAMI-Senti), with data being extracted from SHAMI. The corpus contains 2k sentences labelled with positive, negative or mixed polarity. Shami-Senti is the first Levantine corpus that contains sentimental sentences covering all four countries (Palestine, Syria, Lebanon and Jordan). The corpus can be used to study the differences or the similarities between these dialects in how they express feelings and sentiments. Based on the similarity results from *Study 3*, the SA task is therefore used to examine whether one can adapt classification models that have been trained and built on MSA data for DA from the Levantine region, or whether one should build and train specific models for the individual dialects, therefore considering them as stand-alone languages. In addition, I exploit the available SA corpora and check if I can fine-tune the SA models across different dialects.

Given the results from Study 3 and Study 4, I conduct **Study 5**, page 175, to investigate the usage of deep neural network, which combines bi-directional long-short term memory networks (Bi-LSTM) with convolutional neural networks (CNN) for Dialectal Arabic SA. I test the productivity of DL models on achieving good results despite the size of the corpora, as well as the data balancing among the classes.

Continuing on SA tasks and as Shami-Senti is considered a small corpus, I now introduce **Study 6**, page 191. The contributions of this study can be summarized as follows: Firstly, I propose a 36K Arabic Tweets SA Dataset (ATSAD) which is automatically collected and labelled in terms of sentiment polarity by applying distant supervision. I exploit the emojis to collect as many sentimental tweets as possible, where the language is set to be Arabic. The emojis are used as weakly labels for automatic annotation of the corpus. Secondly, with a little help from manual annotation I could evaluate the corpus by both intrinsic evaluation and extrinsic evaluation approaches. I release a gold-standard (human annotated) corpus of 8k tweets as well as the full corpus of 36k tweets. I improve the quality of the full corpus by employing self-training approach with distant supervision technique as a double check approach.

Finally, in **Study 7**, page 211, I compare available well-known NLP methods and investigate the efficiency and the performance of these models on DA resources and with two tasks: DI and SA. My aim was to investigate the question of choosing the best methods for DANLP tasks, taking into consideration the differences among the resources. I exploit several Bidirectional Encoder Representations from Transformer (BERT) models such as (Multilingual-BERT, Ara-BERT and Twitter-Arabic-BERT). Meanwhile, I study the performance of feature engineering-based approaches and ML algorithms as well as the usage of pre-trained word embedding in DL models. I used all of these models and their results to figure out what is the best choice for DA, and whether there is a single model that can be used for different corpora and various tasks. By discussing the advantages and disadvantages of the applied models, we aim at helping future researchers to choose the model based on their priorities. For example, researchers can differentiate the models according to other factors like the language in question and balance in the data. Another finding in this paper is that in many cases, especially with under-resourced languages, simple methods like traditional ML algorithms can compete with more sophisticated models based on DL.

# 2 Arabic Language

Arabic is a Semitic language. It originated in the Arab peninsula. It is called Arab after the Arabs (people who were living in the Arabian Peninsula)[15]. At the beginning of Islam, there were two main Arabic linguistic sources: the Holy Qur'an and pre-Islamic poems. Arabs were famous for their linguistic eloquence, especially the Bedouins. Bedouin dialects spread at that time due to poetic markets and competitions, as well as the consideration of the Arabian Peninsula as the centre of trade among the Arabs. With the spread of Islam, scholars started to pay more attention to the Holy Qur'an and its texts in an attempt to transmit, explain and clarify its text and contents. Due to the expansion of Islamic conquests, particularly during the era of the Umayyad and Abbasid states, the need for a standardized language became an urgent need, and the most important prerequisite for the written codification of the desired new standard language was the invention of orthography. Among the reasons that led to language standardization is the amount of dialectal variation between Bedouin and urban varieties, as well as between these and the varieties used in the conquered countries. Additionally, the central government's policy was to control all subjects and matters in standardized linguistic norms, thus making standardization a prerequisite for effective governance. The result of this standardization of the Arabic language is what today call **Classical Arabic (CA)**[16, 17]. It is clear then, that the CA is not an ancestor of dialects, but it is a sister of the Arabic varieties.

Arabic expanded due to the Islamic conquests and ended up being spoken in a number of regions such as Egypt, the Levant and North African, in addition to Spain (Andalusia), Malta, parts of Cyprus and regions of Persia, among others. Of course, Arabic dialects existed before Islam, but, with the increased interactions between Arabs and residents of conquered countries, or the movement of people and tribes among them, further development of these dialects gave rise to what is called **New Arabic**. By New Arabic or Neo-Arabic, it is meant the **Arabic dialects** as we know them today. The situation emerging is one where CA was used for the official and religious subjects of the government, while the dialects were the spoken language of Arabic speakers[2, 18].

Arabic has been later on influenced by a number of European languages such as French, British, as well as Turkish. These were some of the languages used as official languages in regions where Arabic speakers exist. Each of these languages introduced new linguistic elements into Arabic. At the beginning of the twentieth century, academies in many Arab countries, especially Egypt, Syria, Iraq, Jordan and Morocco, worked on modernizing CA, as well as updating and expanding the lexicon. The goal of these academies was a rather prescriptive one: to "protect" Arabic from dialectal and foreign influence and, furthermore, to adapt it to the needs of modern times. This gave rise to what we know today as **Modern Standard Arabic (MSA)**.

MSA is the official/co-official spoken language for 23 countries, they are: Saudi Arabic, Qatar, Kuwait, Bahrain, United Arab Emirates, Oman, Yemen, Jordan, Syria, Lebanon, Palestine, Iraq, Egypt, Sudan; Libya, Tunisia, Algeria, Morocco, Somali, Mauritania, Chad, Comoros and Djibouti. In addition, it is recognized as a minority language or working language in Cyprus, Mali, Eritrea, Niger, Senegal, Turkey and Iran. Arabic language has a special status according to the constitution, mostly as a religious language in some countries such as Israel, Pakistan, the Philippines and South Africa.[1]

## 2.1 Diglossia in the Arabic world

Diglossia is the phenomenon where two or more distinct varieties of a language are spoken within the same speech community [19]. In a diglossic situation, the standard formal language assumes the role of the High variety (H), while the other languages or dialects act as the Low variety (L) [20].

The Arabic-speaking countries present us with clear examples of diglossic situations, where three varieties of Arabic usually co-exist: CA, i.e. the religious language, MSA, i.e. the official language, and DA, i.e. the spoken language.

Neither CA nor MSA are the native varieties for speakers of Arabic.

[1] https://en.wikipedia.org/wiki/List$_{o}f_{c}ountries_{w}here_{A}rabic_{i}s_{a}n_{o}fficial_{l}anguage$

The Arabic dialects are the native language for Arabic speakers, and every country in the Middle East and North Africa (MENA) has its own dialects. Arabic speakers use their own dialect for every day communication, and as such these are the varieties used to communicate between them on social media platforms. In some cases, more than two varieties are used within the same community, for example MSA, DA and French in Lebanon, and Arabic, Berber, French, English and Spanish in Morocco[21].

Arabic is considered the official/co-official spoken language for 23 countries: Saudi Arabic, Qatar, Kuwait, Bahrain, United Arab Emirates, Oman, Yemen, Jordan, Syria, Lebanon, Palestine, Iraq, Egypt, Sudan; Libya, Tunisia, Algeria, Morocco, Somali, Mauritania, Chad, Comoros and Djibouti. In addition, it is recognized as a minority language or working language in Cyprus, Mali, Eritrea, Niger, Senegal, Turkey and Iran. Also, Arabic has a special status mostly as a religious language in some other countries such as Israel, Pakistan, Philippine, Malaysia and South Africa.[2]

Many classifications using different parameters exist for the Arabic dialects. The most known categorization is the one by Nizar Habash[22]. There, the Arab dialects are categorized, according to geographic distribution, into five dominant groups, and they are:

- Gulf: at first sight, the Gulf varieties seem like one unified dialect. On a coarser look, however, one notices a number of differences[23]. This group of dialects includes all the dialects spoken in the Gulf Cooperation Council countries, as well as Yemen. Gulf Arabic shares many characteristics with MSA, and some researchers claim that Arabic actually originated in the Gulf region[24].

- Egyptian: this is the most understandable dialect between the Arab speakers from other Arab countries. To some extent, this is due to the Egyptian media industry and the role of Egypt in the middle east. Egyptian Arabic includes the different dialects of Egypt, Libya and the Sudanese dialects[25].

- Levantine: this is a set of four closely related dialects spoken in Palestine, Syria, Jordan and Lebanon. These dialects have their differences to each other, but they do, however, look very similar and close to each other in written form[26].

- Iraqi: even though some people count Iraqi as one of the Gulf dialects, it has a number of different dialect features that make it distinguishable[27].

[2] https://en.wikipedia.org/wiki/List_of_countries_where_Arabic_is_an_official_language

- Maghrebi: The north-Africa dialects spoken in the countries of Tunisia, Algeria, Morocco and some parts of Mauritania comprise this set of dialects. These are heavily influenced by French and Berber languages. As a result, they are hard to understand by Arabic speakers from other regions[28].

The International Organization for Standardization (ISO) goes beyond this categorization into more fine-grained based on the country itself.[3] It classified the Arabic Dialects as 30 different varieties, considering them 30 different languages; Figure 2.1 shows 26 out of 30 dialects.[4]

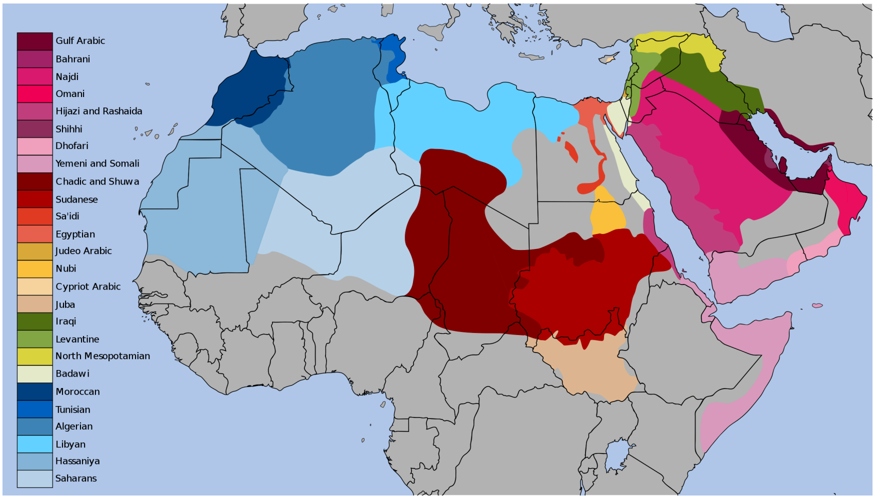


Figure 2.1: Maps of fine-grained Arab Dialects classification.

Others classify Arabic dialects according to the speech style and the level of modernization, classifying them into Urban, Rural and Bedouin dialects[29, 30]. Each one of these classifications can be further fine-grained to contain sub-dialects or varieties of the dialect: for example, Palestinian Dialects, which belong to the Levantine group, can be also split to varieties from Gaza, varieties of North Palestine, and different varieties in the west-bank and the south parts, or it can be split into Urban, Rural and Bedouin accents according to social and geographical location[31].

[3] https://iso639-3.sil.org/code/ara

[4] https://en.wikipedia.org/wiki/Varieties_of_Arabic#/media/File:Arabic_Dialects.svg

## 2.2 Characteristics and challenges of MSA

During the last decade Arabic started gaining a lot of interest in the field of NLP. This interest is due to the spread of Arabic and its huge literary heritage [32]. In this section I will explain the most challenging aspects for processing Arabic (MSA), as well as some interesting linguistic characteristics.

### 2.2.1 Arabic Orthography

1. The Arabic alphabet consists of 28 letters, written from right to left. The writing style is connected and every letter has many different forms depending on its' position in the word [33].

2. Arabic writing style lacks capitalization which is a feature in other languages and can help to identify proper names, acronyms and the beginning of sentences. Absence of capitalization makes it difficult for some tasks like Named Entity Recognition to recognize these features[34].

3. Recognizing sentence boundaries in a running text is a more difficult task in languages such as Arabic than it is in languages like English due to the absence of strict punctuation rules. In fact, it is common in Arabic discourse to write an entire paragraph without a single period except at the end of that paragraph. Sentences are often conjoined via the Arabic coordinators (و *w*) wa and (ف *f*) fa and Arabic discourse is characterized by excessive use of coordination, subordination and logical connectives[35].

4. Acronyms in Arabic are written as any other words due to the lack of capitalization; thus a special model or application is needed to process and identify the acronyms and the abbreviations. For example, in Arabic: the acronym (راما *rāmā*) used for (رابطة المرأة الأردنية *rābṭh ālmrah ālardnyh*) means (Jordanian Women Association), while the same word (راما *rāmā*) is a female name, so not very clear if this is an acronym or a person's name unless I understand the context.This ambiguity is considered as challenging for tokenization and processing the writing text[36].

5. Some letters are pronounced but not written such as الذي *ālḏy* (which), while others are written but not pronounced such as (hamza wasel) which is a silent Alef at the beginning of the words, for example امرأة *āmrah* (woman)[37].

6. Normalization style in writing Arabic text, where the text lack of consistency. For example, some people prefer to ignore all the hamzas in their writing style, while others use all their forms. When processing Arabic text, NLP experts apply normalization to the text to minimize the number of different forms or appearance of the same letters. For example, a name like *Ahmed* could be written in two style with or without the hamzas as أحمد *aḥmd* and احمد *āḥmd*. Thus, during the normalization process, they used to remove all the hamzas to get only one form such as احمد *āḥmd*[22]. In some cases it works, while in other cases removing the hamzas or applying any normalization rules could change the meaning of the word or convert two totally different words into the same words which then has an effect on the performance of some NLP tasks that depend on the word meaning (e.g. Sentiment Analysis. For example, the words علي *ly* (Ali), which is a name for a male, and على *lā* (on preposition)- after applying normalization, the two words become على *lā*! So even though normalization appears to solve the variability in input problem, it might also increase the probability of ambiguity[38]. Moreover, there is no agreement between the NLP experts on a normalization, so researchers apply their own view of how to normalize the text so that it improves the results for their desired task.

### 2.2.2 Arabic Phonology

1. Even-though the association between Arabic letters and their corresponding sounds is one-to-one mapping, Arabic is not an easy language to read without the system of diacritics. Two types of vowels are represented in Arabic, long and short vowels. Long vowels are represented by three letters (أ *a*,و *w*,ي *y*), while for the short vowels these are only identified by the diacritics[33].

2. Diacritics in Arabic are marks written above or below the letters to represent short vowels. Absence of diacritics increased the phonological ambiguity of the text. For example, the world علم *lm* without diacritics could hold several meanings, but with the presence of diacritics it would produce many words such as: عِلْمُ، عَلِمَ، علَّمَ *ilmu, alima, lam*(science, knowledge, learn, mark, ,..etc).

3. In Arabic text, short vowels are represented completely as diacritics, while there are no short vowels at all in the free-diacritics Arabic text

. That means the phonological information is incomplete, and the pronunciation of the words should be derived based on the semantics by fluent Arabic readers[39].

### 2.2.3 Arabic Morphology

1. Arabic is a highly derivational language. All verbs and most of the nouns in Arabic, as well as other Semitic languages in general, are derived from a base of three or four characters known as the root. Then, the Arabic word (Lemma) is composed of at least two main components: Root + Pattern. The root consists of three or four constants that carry out the semantic meaning of the word. The pattern is the template and mainly consists of vowels. It forms the syllabic structure of the word besides carrying syntactic and semantic information. For example, the word كتاب *ktāb* (book) composed of the root كتب *ktb* and the pattern فاعل *fāl*, while the word مكتوب *mktwb* (is written) consists from the same root and the pattern is مفعول *mfwl*. Therefore, building a morphological analyzer for Arabic is not a straightforward task, because such a system must be able to deal with all the patterns and the derivations found in the language. Furthermore, it is normal that some letters are ignored from the words in some cases and positions, such as the long vowels at the end of the verbs. This adds extra complications[40, 41].

2. Words in Arabic have more than the lemma; they can have zero or more affixes and clitics. The affix (prefix, infix, suffix) shows the person, the gender and the number, while the clitics are morphemes that are grammatically independent, but in terms of morphology they are dependent on other words or phrases, then, they could be attached or detached pronouns[42].

3. Clear interaction between the morphological derivation and phonological rule in Arabic makes it difficult to deconstruct the Arabic words. For example, the verb رأى *raā* (saw) ends with Alef Maqsora ى *ā*, which is converted to ي *y* if affixes are added to the verb, such as رأيت *rayt* (I saw) or رأيتهم *raythm* (I saw them). In the Jussive Case, the verb takes no vowels at all. Then, if the verb is preceded by one of the jussive particles such as لم *lm* (did not), and based on the declension system in the Arabic language, the vowel is omitted to be لم أر *lm ar* (I did not seen). Changing or deleting the origin of the

letter increases the complexity of Arabic NLP tasks like tokenization and morphological analyzer systems[43].

4. Arabic is a language exhibiting a lot of compounding, where two or more words are combined to form a new word. For example, الرأسمالية *ālrasmālyh* (Capitalism) consists of رأس *ras* (head) and مال *māl* (money) where the two parts are related to the new word. Another example is برمائي *brmāyy* (Amphibians) that composes بر *br* (land) and مائي *māyy* (water)[44].

5. Morphological Ambiguity: many words in Arabic share the same morphemes, although they have a different internal structure which leads to ambiguity. For example, the word وعد *wd* can be segmented into و *w* (the conjunction and) + عد *d* (count) or it could be a noun/verb that means (promise)[22, 40].

6. Arabic, as any other language, is rich in terms of diomatic multi-word expressions, which in turn increase the difficulty of processing the text. A phrase such as ضرب بكلامه عرض الحائط *ḍrb bklāmh rḍ ālḥāyṭ* literally means *he hit the wall by his words*, while the actual semantic meaning is *he rejected his argument*[45].

### 2.2.4 Arabic Syntax

1. Arabic is considered a relatively free-word order language. In CA as well as MSA the predominant order is verb-subject-object (VSO). However, the use of subject-verb-object (SVO) and object-verb-subject (OVS) and other orders are also permitted[46].

2. Arabic is a pro-drop language: it is normal for sentences in Arabic to be missing the subject[47].

3. Anaphora resolution: one of the most challenging tasks is to understand and resolve references to earlier or later words in the text. Arabic, as many other languages, needs a lot of effort to solve the anaphora resolution problem, especially given the absence of punctuation[48].

4. Arabic has a rich agreement system, where the verb must agree with the subject with regard to phi-features (i.e. gender, person and number). The level of agreement (full/partial) depends on the verb-subject orders. In SVO, the full agreement must be satisfied while in VSO

word order only the gender agreement as a kind of partially agreement can be obtained[49, 50].

5. Syntactic ambiguity also poses challenges for POS tagger systems and for annotators, as well as in the analysis of the internal structure of some sentences. For example, the sentence لقد قابلت مديرة المدرسة الجديدة *lqd qāblt mdyrh ālmdrsh ālǧdydh* can mean *I met the headmaster of the new school*, as well as *I met the new headmaster of the school*[46].

Many languages share these characteristics and challenges. For example, in Chinese there are no boundaries between the words[51], Italian is a pro-drop language as well[52]. However, what makes Arabic a complex and unique language is that all the aforementioned challenges occur in the same language.

## 2.3 Moving from MSA to DA

In addition to all the characteristics that Dialectal Arabic (DA) shares with MSA, there are also some characteristics that are unique to them as, follows:

- All Arabic dialects are under-resourced languages, i.e. they lack the availability of resources, even though researchers recently have been paying more attention to these dialects and have come up with valuable resources. However, compared to other languages such as English or French, or even compared to MSA, DA is still in its early development process.

- All Arabic speakers are multilingual or at least they are bi-lingual, i.e. they use their own dialect in social media websites as it is considered easier for them to communicate. Therefore, the emergence of social media platforms strengthened the Arabic Dialects and led to the emergence of dialectal written resources alongside audio resources, such as recordings and phone conversations, which were a primary source of vernacular in the past.

- MSA is the *lingua franca* in the Arabic-speaking countries. Dialects are not mutually intelligible - the greater the geographical distance between countries, the lower the level of understanding between speakers. In nearby or neighbouring countries, the ease of communication between speakers is more than if they are dealing or talking with people from distant countries. For example in the Levant, even with some slight differences between the dialects, it is very flexible for people to speak their own dialect in any of the four countries, as their dialects are overlapped and interrelated. However, they may have some difficulties in speaking to people from Morocco, as they try to use familiar dialects or other languages to communicate[53].

- All the dialect texts are unvowelled, and that makes the ambiguity more and more when processing the text. In addition, dialects in every country haves their own lexicon, and sometimes the dialect itself is mixed with other languages, such as the case of Algerian dialect that is mixed with French and Berber.

## 2.4 Qualitative differences between MSA and DA

MSA is the official language and the lingua-franca in the Arabic-speaking countries. Nevertheless, no one uses it in their daily conversation. Even though the two varieties share some common features, MSA and DA have a number of differences that make it difficult for one to apply state-of-the-art MSA natural language processing tools to DA. The degree of variation between MSA and dialectal Arabic depends on the specific dialect of Arabic. MSA and DA differ to a different degree phonologically, orthographically, morphologically, syntactically, lexically and semantically.

In this section, I describe some qualitative differences between MSA and the dialects based on our observation of examples.

### 2.4.1 Orthographical Differences

Dialectal Arabic (DA) does not have a unified established standard orthography like MSA. Habash and Diab proposed CODA which is conventional orthography for writing dialectal Arabic but for computational purposes[54]. Arabic script is used to write DA words reflecting the phonology or the history (etymology) of the word. However, in some cases, e.g. in Lebanese, the Latin alphabet is used for writing or posting on social media[55]. For example, كيفك *kyfk*(how are you) is represented as *Keifk*.

### 2.4.2 Phonological Differences

The most recognized phoneme between MSA and DA is the pronunciation of dialectal words whose original MSA cognate contains the letter ق *q*[56]. For instance, the Palestinian speakers from rural and urban regions pronounce it like /'/ glottal stop or /k/, while Bedouins pronounce it as a /g/ . The word قال *qāl* /say is pronounced and sometimes written as قال *qāl* , كال *kāl*

, ئال *yāl* or جال *ǧāl* [31]. In contrast, in the North-African dialects it is pronounced as /q/ (similar to MSA)[57].

### 2.4.3 Morphological Differences

Dialects such as MSA and other Semitic languages make extensive use of particular morphological patterns in addition to a large set of affixes (prefixes, suffixes, or infixes) and clitics. Therefore, there are some important differences between MSA and dialectal Arabic in terms of morphology because of the way these clitics, particles and affixes are used [58, 54]. For example the future marking particle س *s* or سوف *swf* is one of the most common differences between the MSA and between the dialects themselves also. In North-Africa, باش *bāš* in-front of the verb like باش نلعب *bāš nlb* is used (I will play), while in Levantine they put ح *ḥ* to be the first letter before the verb as حلعب *ḥlb* and sometimes they used راح *rāḥ* as راح ألعب *rāḥ alb*[57]. Some examples are illustrated in Table 2.1 and Table 2.2.

| **1. Using multiple words together** | | | | | |
|---|---|---|---|---|---|
| MSA | English | Leventine | | Gulf | |
| كيف حالك؟ ما أخبارك؟ *kyf ḥālk? mā aẖbārk?* | How are you? | كيفك *kyfk* | | شخبارك *šẖbārk* | |
| **2. Sharing the stem with different affixes** | | | | | |
| MSA | English | Syrian | Egyptian | Palestinian | Moroccon |
| لا يدرس *lā ydrs* | He does not study. | مابيدرس *mābydrs* | مبيدرسش *mbydrsš* | بدرسش *bdrsš* | مايدرسش *māydrsš* |
| **3. The future marker** | | | | | |
| MSA | English | Egyptian | Leventine | | Tunisia |
| سوف يلعب، سيلعب *swf ylb, sylb* | He will play. | هيلعب *hylb* | حيلعب *ḥylb* | راح يلعب *rāḥ ylb* | باش يلعب *bāš ylb* |
| **4. Clitics (for present tense)** | | | | | |
| MSA | English | Syria | | Egyptian | |
| هو يأكل *hw yakl* | He eats / He is eating | عم ياكل *m yākl* | | بياكل *byākl* | |

Table 2.1: Examples of morphological differences

### 2.4.4 Syntactic Differences

Syntactically, MSA and DA share a lot of similarities. However, some differences with respect to word order are attested. For example, OVS and OSV word orders are most commonly used in MSA, while in DA more word order pattern variation can be found. For example, in Levantine SVO is most commonly used, while in Maghrebi, VSO is used to a great extent [58]. Furthermore, in dialectal Arabic it is common to use masculine plural or singular forms instead of dual and feminine plural forms [59]. Another Difference between MSA and the dialects is the way each forms the question. MSA forms the constituent question by fronting the wh-element, e.g. أين ذهبت بالأمس *ayn ḏhbt bālams* (where did you go yesterday), while in

| **MSA** | لا أعرف *lā arf* | | |
|---|---|---|---|
| **English** | I do not Know | | |
| **Palestinian** | بعرفش *brfš* | **Syrian** | مابعرف *mābrf* |
| **Jordanian** | مش عارف *mš ārf* | **Lebanees** | مابعرف *mābrf* |
| **Egyptian** | معرفش *mrfš* | **Gulf** | مدري *mdry* |
| **Iraqi** | ما أدري *mā adry* | **Algerian** | مش نعرف *mš nrf* |
| **Tunisian** | منيش عارف *mnyš ārf* | **Morocon** | منعرفش *mnrfš* |

Table 2.2: Differences in negation between the dialects

DA, the wh-element stays in-situ روحت وين امبارح *rwḥt wyn āmbārḥ* (you went where yesterday?)[60].

### 2.4.5 Lexical and Semantic differences

Although most dialectal words have their equivalents in MSA, many words are borrowed from a variety of other languages such as Turkish, French, English, Hebrew, Persian and others, as a result of communication, trading and colonization of these regions. For example, in Lebanon and in North_African dialects a lot of French loan words are used, while in Palestinian dialects one sees more words of Hebrew origin, and also more borrowing from English. Table 2.3 shows some of the borrowed words. In some varieties such as Algerian and Moroccan, the original loan verb is used and modified by adding Arabic affixes and clitics. For example, the French verb (Charger) is modified to شرجاها *šrǧāhā* and يشرجيها *yšrǧyhā* means *he charged it* and *he charges it* respectively [58]. New lexical items appear mostly in dialects and not MSA as shown by the example in Table 2.4. Although MSA and DA share many words, the meanings are sometimes different. For example, the word دول *dwl* means (these) in Egyptian but it refers to (countries) in MSA.

| **Word** | **Origin** | **MSA** | **English** |
|---|---|---|---|
| طربيزة *ṭrbyzh* | Turkish | طاولة *ṭāwlh* | Table |
| أستاذ *astāḏ* | Persian | مدرس *mdrs* | Teacher |
| أفوكادو *afwkādw* | French | محامي *mḥāmy* | Lawyer |
| بندورة *bndwrh* | Italian | طماطم *ṭmāṭm* | Tomatoes |
| توف *twf* | Hebrew | جيد *ǧyd* | Good |
| تليفون *tlyfwn* | English | هاتف *hātf* | Telephone |

Table 2.3: Examples of borrowed words from other languages

| **Word** | **Language /Dialect** |
|---|---|
| الآن *ālān* | MSA |
| Now | English |
| هلأ، هسا، هلقيت *hla, hsā, hlqyt* | Levantine |
| هلحين *hlḥyn* | Bedouin |
| دحين *dḥyn* | Saudian |
| هالوقت *hālwqt* | Iraqi |
| توا *twā* | North African |
| دلوقتي، دلوقت *dlwqty, dlwqt* | Egyptian |

Table 2.4: Examples for new lexical items in dialects

# 3 Dialectal Arabic Natural Language Processing

NLP developers have been focusing on developing ANLP systems to enable both Arabic and non-Arabic speakers to process Arabic text. However, most of the proposed tools and systems are developed to process Arabic text written in MSA[61]. Even though this is good for MSA, nowadays DA is the most used variety in every Arab country, being the mother tongue of every single Arabic speaker. Applying the available MSA tools to Arabic dialects, the result I get will not be as accurate as desired, because the two varieties have significant differences in the way I have previously outlined. Thus, to develop Arabic tools, NLP developers and researchers must specify beforehand the variety they are aiming for in order to get accurate results. For example if the aim is to develop an application for news broadcasting, then the variety is MSA; if it is for Quran or Hadith processing text, it will be CA, but in the case of proposing a tool for analyzing user tweets for a marketing service in Egypt, this will have to be the Egyptian dialect, and so on.

## 3.1 Literature Review

Dialectal Arabic is still in it's developing stage, and the lack of significant and valuable resources is well-known. Most NLP researchers handle these problems of DA by introducing and building different kinds of resources such as lexicons, corpora, treebanks and others, depending on the task they are addressing [62]. The nature of the resource is affected by the type of task, where the Arabic computational linguistics address different tasks, ranging from fundamental language aspects like morphology up to very sophisticated tasks such as language generation and machine translation [63].

The available Dialectal Arabic resources can be found in two forms: either as a tool to perform Arabic computational linguistics tasks such as morphological analyzers and POS taggers, or as data, such as lexicons, corpora and sometimes special-purpose annotated corpora.

In this section, I will shortly mention some of the well known dialectal resources that have been proposed to the Arab research community. Most of these resources have been built for special NLP purposes like machine translation, or have been annotated for Dialect Identification, POS tagging and so on. The sources of the data collected vary and include newspapers, blogs, social media platforms and others. Moreover, the type of these corpora are different - either they are monolingual, including only Arabic and its dialects, or bi-lingual, which also include other languages such as English and French. The format of the corpora are mixed between parallel datasets, comparable or both.

Different Arabic dialect corpora have been published in the last decade. These are very different in terms of the dialects they include, and the designs of the corpus (mono-lingual, parallel or comparable). They also employ various collection tools and annotation processes. Table 3.1 summarizes the most well known resources. The Levantine dialects have been addressed in many works, where they are considered as one dialect [64, 3, 65, 66]. Other works focused on one dialect from the Levantine dialects, as it is the case with the Curras corpus [31], which is a pure Palestinian corpus. Curras is a valuable resource for the Palestinian dialects that sheds light on the characteristics of these dialects, and they are used for many different purposes, such as POS taggers. The work done by [67] covers the Levantine dialects but most of the Levantine data come from one Levantine dialect only, namely Jordanian, so can not be considered as a resource to study the four Levantine dialects. The first multi-dialect Arabic parallel corpus has been introduced by [57]. It includes three Levantine dialects: Syrian, Palestinian, Jordanian, as well as MSA, Egyptian, Tunisian and English. The Mutli-dialects corpus is based on an Egyptian-English corpus and they translated the Egyptian part to the rest of the dialects and MSA. It contains 1,000 instances for each variety. posteriorly; PADIC (Parallel Arabic Dialect Corpus) was presented in 2015 by Meftouh et al.[58]. It includes the Syrian and Palestinian dialects as two separated dialects. PADIC is a well-known resource as it contains dialects from North Africa and Levantine in addition to MSA in a parallel way, so one can study the characteristics and the differences of these dialects.

After the emergence of social media platforms, research has started leveraging social media data and used the available application programming interfaces (API) to collect the data automatically. That resulted in a reasonable amount of data in a very short time. Twitter is still the main platform that researchers depend on to collect the data according to their publishing guidelines and rules. Many corpus have been built using Twitter data, and the Levantine dialect is covered in [68] as one dialect, where Abdul-Mageed et al.[69] used Twitter API in addition to Python geocoding library geopy to collect tweets over 10 Arabic countries set where Jordan and Palestine are included.

Very fine-grained and valuable dialectal Arabic resources are constructed by Bouamor et al.[70]. They presented the MADAR-CORPUS-25, a parallel corpus of 25 Arabic dialects (25 Arab cities) with a total size of 50K sentences, alongside with MADAR-CORPUS-5, a parallel corpus of five Arabic dialects corresponding to five Arabic cities. The corpus is considered a valuable resource in terms of parallel fine-grained dialectal Arabic at the level of cities, where many Levantine cities were included, such as Jerusalem, Damascus, Beirut, Amman and others. MADAR corpus is a great resource for the Arabic linguistics community, in terms of the number of dialects it contains. Huge efforts have been utilized for the corpus construction and can be used for different purposes like dialect identification, machine translation, linguistics studies and so on.

Recently, Boujou et al.[71] proposed dialectal dataset, which was collected from Twitter and included 50K tweets. The dataset contains five Arabic dialects from five Arab countries: Algeria, Egypt, Lebanon, Tunisia and Morocco. They labeled the data for multi-purpose applications such as dialect identification, sentiment analysis and topic detection. In addition to the dataset they implemented machine learning baseline models so researchers can use them for comparison purposes.

In addition to the previous works I address in sections 7.2 and 8.1, however, an increased interest has been noticed recently for Arabic Dialect Identification, and therefore many shared tasks have been introduced to solve the problem. MADAR_2019[4], NADI_2020[72] and NADI_2021[73] are shared tasks, and were organized by the Arabic Natural Language Processing Workshop WANLP, which is held every year.

The first sub task at MADAR_2019[4] was MADAR travel domain dialect identification, where they asked for fine-grained city level dialect identification models. The used dataset was MADAR-CORPUS-25 [70], and thus every participating team had to build a model that was able to classify the sentences into one of 26 labels corresponding to the 25 Arab cities + MSA. The fine-grained task was explored firstly by Salemeh et al.[74] using the MADAR-CORPUS-25 mentioned above. Their system was able to identify the city of the targeted sentences and obtained a 67.89% averaged macro F-score by employing N-gram language models. Nineteen teams participated in the shared tasks and they were ranked based on the same evaluation metrics as the previous system (macro average F-Score). My work got the first place achieving 67.32% F-score by employing n-grams language models in addition to ensemble learning; for more details about the system, see details at 5.1.2, page 52. The second-ranked team was (SMART)[75], which proposed a Naive Bayes classifier based on word and character grams model as well as language model probabilities. The system achieved an F-score of 67.31% for the blind test set. The Mawdoo3 LTD system was ranked third in the shared task with 67.2% f-Score[76]. They used ensemble learning, n-gram models, including both words and characters, in addition to language model probabilities. Most of the participating teams used machine learning techniques in addition to feature engineering features as N-gram models and language modeling[77, 78, 79, 80].

NADI_2020 was the first Nuanced Arabic Dialect Identification Shared Task organized by WANLP, also in 2020[72]. The two sub tasks were about dialect identification either on the level of country or the level of provinces (sub-country). The introduced dataset was collected from Twitter and covered a total of 100 provinces form 21 Arab countries, and not equally distributed. While MADAR-CORPUS-25 was a hand-crafted travel domain dataset, NADI is based on naturally occurring data, as well as bigger in terms of size. NADI dataset contains 30,957 labelled tweets as well as 10M unlabelled. For the annotation process, they used the post location as a proxy for dialects labels, which is not accurate all the time. As the tasks concerned very fine-grained dialect identification systems, as well as the number of dialects being 21 for country level (task 1) and 100 for province level (task 2), the NADI shared task organizers proposed their baseline models by fine-tune on Google's pre-trained multi-lingual BERT (mBERT). They achieved an average F-score of 13.32% on country level and 2.13% for province-level identification. Then, winning teams obtained an F-score of 26.78% and 6.39% for task 1 and task 2 respectively. Most of the teams employed the transformer technology BERT as the latest trendy technique. Mawdoo3 AI [81], the top ranked team, introduced various models exploiting different strategies. They used feature engineering as the N-gram models and TFIDF features. They also implemented ensemble learning with different classifiers as Regression and N-Bayes. Moreover, they presented a deep learning model with the help of word-embeddings.The winning model was based on Multi dialect Arabic BERT [82], followed by a shallow feed forward neural network. They trained different models at various settings, and in the end they built a voting ensemble mechanism. All the top five teams used BERT to build identification models as well as the classical machine learning algorithms, and the same applied for the winners regarding task 2 which comprised more fine-grained identification processes[83, 84, 85, 86].

Concurrently with NADI_2020, Abdul-Magged et al.[87] proposed a novel task at the level of Micro-Dialect Identification (MDI) and they introduced a new language model called (MARBERT)[88] that has the ability to identify a fine-grained dialect given a short message. In addition to their model, they built a large scale dataset that contained fine-grained Arabic dialects. They collected ≈6 billion tweets from 2.7 million users, and after location verification, they were able to label ≈507M tweets corresponding to 233K users. The tweets covered 21 Arab country, 646 cities and 235 provinces. Different processing and verification techniques were applied such as location verification, code switching and Diglossia checking. For every process they introduced a dataset targeting the special procedure, for example Micor-Arab dataset, CodSw dataset and DigGloss dataset. The three datasets were used independently but shared the same methods as a multi-task classification process. Their methods exploit the Gated Recurrent Units GRU in addition to Googles' multi-BERT. Different models have been proposed as following (i) Single and Multi-tsak Bi-GRU models, (ii) Single task BERT and (iii) multi-task BERT. Their dataset as well as the models are publicly available to researchers.[1]

NADI_2021 [73] followed the previous mentioned works[87, 72], but they introduced an identification process on both DA and MSA separately. They used the same dataset as NADI_2020 but applied more processing steps to exclude non Arabic tweets in Persian and Farsi. As they addressed an identification for MSA so they teased a part MSA from DA by using the classification methods proposed in [88]. The NADI dataset is publicly available for research purposes.[2] The shared task consisted of four tasks, comprising two levels, country level and province level. On each level the classification was carried out for MSA and DA. They wanted to figure out to what extent a machine would be able to tease apart MSA data at the country and province level. It was the first time in which this kind of MSA classification was addressed. The model should detect the country or province where the MSA tweet was posted. NADI_2021 baseline models were nearly the same as NADI_2020 models based on multi-BERT and they obtained average F-scores of 14.15% for country level MSA (task 1.1), 18.02% for country level DA(task 1.2), 3.39% for province level MSA (task 2.1) and 4.08% for province level DA (task 2.2). In total, 68 teams submitted their identification models distributed between the four tasks. The majority of teams have used BERT transformers in addition to ensemble learning. Different BERT models have been utilized, such as MARBERT [88], AraBERT[89], AraELECTRA[90] and others. For all the tasks, CairoSquad team got the first places.

CairoSquad [91] built their model based on MARBERT [88]. In their

[1] https://github.com/UBC-NLP/microdialects

[2] https://github.com/UBC-NLP/nadi

system they applied adapter modules[92] and vertical attention to fine-tune MARBERT, so they did some changes on the architecture and training settings. At the end they used ensemble techniques for the final classification results. The same proposing system was used for all the four tasks and they achieved average F-scores of 22.38%, 32.26%, 6.43% and 8.60% for task 1.1, task 1.2, task 2.1 and task 2.2 respectively. Likewise, Team CS-UM6P [93] used the same MARBERT for their system, with different optimization settings. While the third-ranked team Phonemer, [94] used AraBERT in addition to AraELECTRA.

Most, if not all, of the aforementioned resources have been constructed and used to develop a variety of tools for different tasks: Dialect detection, morphological analyzers, POS taggers and machine translation. However, some other NLP tasks such as Entity Recognition, Semantic Labelling and sentiment analysis need more annotation efforts and processes given their special purpose usage. As part of the research reported in this thesis concerns sentiment analysis of DA, some of the available resources that have been built for sentiment analysis, in the form of lexicons and corpora are listed

Most of the sentiment analysis resources have been collected from web blogs, customer reviews, and recently, from social media platforms, especially Twitter. Some are manually annotated following specific annotation guidelines, while other researchers employ some kind of automatic annotation. The sentiment analysis resources differ in terms of their kind, whether a lexicon [95, 96, 9] or a dataset[97, 5, 98, 99, 100, 101, 102, 103]. Moreover, researchers employ different classification techniques: some datasets are classified according to the subjectivity and objectivity [6, 104, 105], while others focus only on the subjective data and classify them in a deep way[7, 106]. The proposed corpora did not use the same classification labels. Sentiment analysis data may belong to one of four categories (positive, negative, neutral or mix). Furthermore, some works use 4, 5, 6 or 7-way classifications scaling from extremely positive as 3 to extremely negative as -3[107]. More details about SA related works can be found under sections 10.2, 11.2, 12.2 and 13.2.

| Corpus | Nature | Source \Platform | Dialects | Size |
|---|---|---|---|---|
| COLABA [64] | monolingual search queries | online resources like weblogs and forums | Egyptian, Iraqi, Maghrebi and Levantine | 40 queries |
| (AOC) [3] | monolingual | users reviews and comments from local online newspapers | Gulf, Egyptian, Levantine | 44,6K sentences |
| [65] | parallel | dialectal sentences from an Arabic LDC | Levantine, Egyptian , English | 1.1M words (Lev)<br>380K words (Egy) |
| [66] | dialectal words | bootstrapping dialectal words | Gulf, Levantine, Egyptian and North Africa | 14.5M, 10.4M, 13M, 10.1M (words) |
| Curras [31] | monolingual | social media threads, Palestinian blogs, the Palestinian dialogue network,41 episode scripts from the Palestinian TV show | Palestinian | 5,836 sentences<br>43K words |
| [67] | monolingual | online five Arabic language newspapers and Twitter | Egyptian, Gulf, Levantine (Jordanian), Maghrebi and Iraqi | 6K sentences |
| [57] | parallel | 2,000 Egyptian sentence were extracted form the Egyptian part of the corpus presented in [65] | MSA, Egyptian, Syrian, Palestinian, Tunisian, Jordanian, English | 14K sentence |
| PADIC [58] | parallel | recording conversations in Annaba's dialect movies and TV shows from Algiers's dialect | two Algerian dialects ,Tunisia , Syrian, Palestinian, MSA | 6,400 sent /dialect |
| Gumar [108] | monolingual | 1,200 online available forum novels | Bahrain, Kuwait, Oman , Qatar, United Arab Emirates, Kingdom of Saudi Arabia | 110M words |
| [109] | comparable | Wikipedia-Arabic and Wikipedia-Egyptian | MSA, Egyptian | 10,197 documents |
| DART [68] | monolingual | Twitter | Egyptian, Levantine, Gulf, Maghrebi, Iraqi | 7K tweets |
| [110] | monolingual | various social media platforms | Saudi dialects (Najdi, Hijaz and Gulf) | 104K words |
| [69] | monolingual | Twitter | Oman, Egypt, Iraq, Jordan, Kuwait, Palestine, Qatar, KSA, UAE, and Yemen | 234,801,907 tweets |
| MADAR [70] | parallel | translated the Basic Traveling Expression Corpus (BTEC) [111] | 25 Arab cities, 5 Arabic cities. | 50K, 10K sentences |
| NADI [72, 73] | monolingual | Twitter | 21 Arab countries, 100 provinces | 30,957 labelled tweets, 10M unlabeled |
| Micro-Dialect dataset [87] | monolingual | Twitter | 21 Arab countries, 646 cities ,235 provinces | ≈507M tweets corresponds to 233K users |

Table 3.1: Some well-known resources for Dialectal Arabic NLP

LABR is a well-known corpus in the field of sentiment analysis[107]. It is a book reviews dataset considering stars rating done by Arab readers as sentiment polarity labels. It consists of over 63K book reviews written in MSA with some dialectal words. LABR is available with different subsets: the authors split it into 2, 3, 4 and 5 sentiment polarities with balanced and unbalanced divisions. The fact that LABR is limited to one domain, book reviews, makes it difficult to use it as a general SA model.

Focusing on one dialect like Egyptian, there are multiple works as ASTD [7], and the 40K tweets corpus [106] with 10K tweets and 40K tweets respectively. ASTD data has been classified mainly into subjective and objective polarities, and later the subjective tweets have been classified into four categories (positive, negative, neutral and mix), while the 40K tweets dataset has only two polarity classifications (positive and negative). Many works also have been developed for North African dialects (Tunisian, Algerian and Moroccan). An Algerian dialects lexicon followed by a two-way sentiment classification corpus has been presented in [112, 113]. TSAC is a Sentiment Analysis corpus targeting Tunisian dialect collected from Facebook comments with 17K terms[99]. AraSenTi-Tweet is a Saudi dialect corpus that contains 17,573 Saudi tweets semi-automatically annotated into four polarities: positive, negative, neutral and mixed [114]. In Levantine dialects, a Jordanian dialect corpus was presented by [115] and a similar effort was made by [116] to build ArSentD-LEV. They include 3,550 Jordanian dialect tweets, and 4k Levantine tweets respectively. Table 3.2 mentioned some of the well-known corpora that have been built for the purpose of Arabic sentiment analysis.

Guellil et al. presented a sentiment analysis for Algerian dialect without constructing any Algerian resources[117]. They showed that it is possible to build a sentiment analysis model for dialect X, if I have resources for dialect Y, conditioning that dialect X and Y must be in the same group of dialect. Thus, for their proposed model, they utilized the availability of Moroccan and Tunisian sentiment analysis resources to build their model. They achieved an F-score of 83% by Multilayer Perceptron (MLP) and Long short-term memory (LSTM).

A comprehensive study about sentiment analysis approaches has been done by Farah and Magdy [118]. They used three benchmark datasets (SemEval 2017 Task 4-A Datase [119], ArSAS Dataset[120], ASTD Dataset[7]) and applied different approaches to study and analyze a large variety of models. They showed that deep learning models combined with word embeddings achieved better results than machine learning classifiers such as SVM, while the usage of a transformer and pre-trained language model such as AraBERT[89] was the best.

Recently, a shared task on sarcasm and sentiment detection in Arabic has been organized by WANLP 2021[121]. It has two sub tasks, one for sarcasm detection (task 1) and the other for sentiment analysis (task 2). They used ArSarcasm-v2[3] dataset, which consists of 15,548 tweets labelled for sarcasm, sentiment and dialect detection. The data set is a combination of ArSarcasm dataset [122] and DAICT dataset[123]. The data set is not balanced in terms of sentiment polarity distributed. The tweets are classified into positive, negative and neutral. In addition it contains MSA plus four dialects from four Arab regions, they are: Egypt, Gulf, Levant, Maghreb.

The majority of the works used and fine-tuned pre-trained language models and transformers such as BERT and MARBERT[88]. CS-UM6P team achieved the first rank at the sentiment analysis task with an F-score 0.748[124]. Their model was based on MARBERT multi-task learning in addition to an attention layer for detecting the sentiment. DeepBlueAI whose came in second place by F-score 0.7392, proposed their model using an ensemble of AraBERT and XLM-R[125].

[3] https://github.com/iabufarha/ArSarcasm-v2

| **Corpus** | **Source \Platform** | **Dialects** | **Polarity** | **Size** |
|---|---|---|---|---|
| OCA[126] | Movie reviews | MSA | Neg, Pos | 500 review |
| LABR[107] | Readers' books reviews | MSA, dialects | Neg, Pos, Neutral | 36K |
| Multi-domain dataset[127] | Reviews for hotels, restaurants, movies and products | Multi-dialects | Neg, Mix, Pos | 33K |
| ASTD [7] | Twitter | Egyptian | Neg, Mix, Pos, Obj | 10K |
| BRAD [128] | Books reviews | MSA, dialects | Neg, Pos, Neutral | 510K |
| TSAC[99] | Facebook | Tunisian | Neg, Pos | 17K |
| AraSenti-Tweet [114] | Twitter | Saudi | Neg, Mix, Pos, Neutral | 17.5K |
| SemEval 2017[119] | Twitter | Multi-dialects | Neg, Pos, Neutral | 10K |
| HARD[129] | Hotel reviews | MSA, dialects | Neg, Pos, Neutral | 373K |
| ArSAS[120] | Twitter | Multi-dialects | Neg, Mix, Pos, Neutral | 21K |
| AraSentD-LEV[116] | Twitter | Levantine | V.Neg, Neg, V.Pos, Pos, Neutral | 4K |
| 40-K tweets corpus[106] | Twitter | Egyptian | Neg, Pos | 40K |
| ArSarcasm-v2[121] | Twitter | Multi-dialects | Neg, Pos, Neutral | 15.5K |

Table 3.2: Some well-known resources for Arabic Sentiment Analysis, Polarity: Neg:Negative, Pos:positive, V.: very

# 4 Building Resources for Dialectal Arabic

## 4.1 The Shami Dialects Corpus (SDC)

Arabic dialects are usually categorized and classified according to the main geographic region where they are spoken such as: Gulf, Levantine, North Africa, and Egypt. Every region has many sub dialects that in some ways share many similarities among each other, but they are also different.

Levantine dialects are usually considered as one single dialect, assuming mutual intelligibility among people living in that area exists. However, Levantine comprises many different varieties that are spoken in the area. These dialects can be fine-grained to the level of the country or even go deeper to city-level.

In our work, rather than assuming the Levantine dialect to be one, I focus on the country-level to get a more fine-grained categorization of Levantine dialects. Thus, I collect and build resources that contain and concern the four main countries in the Levant: Palestine, Syria, Lebanon, Jordan. Although the Levantine dialects look very similar, and Arabic speakers, in general, cannot distinguish between them, any native speaker of the Levant easily detects the dialects from the accent and the words used. Different terms and expressions are used among the Levantine dialects that make them distinctive; nevertheless, these differences are mostly dependent on the accent. The presence of un-diacritic text or loss of intonation makes the distinction very difficult.

In Arabic, short vowels are represented by a system of diacritics (اَ *a* Fatha, اُ *u* Damma, اِ *i* Kasra). Most, if not all, of the dialectal texts do not use diacritics, and thus, people depend on their knowledge to correctly read and identify the text. The lack of diacritics, intonation and accent increases the difficulty of distinguishing between the dialects in the written form. For example, a word like (كيفك *kyfk* /how are you) is used in all Levantine dialects, but the word pronunciation varies from country to country. Moreover, as a result of the political situation in the region in the past and the forced deportation that the Palestinians were subjected to that led to them moving to neighbouring countries and establishing refugee camps, the Palestinian dialect mixed with many dialects, which makes the Levantine dialects in general difficult to distinguish.

To study the similarities and differences between the Levantine dialects and to help researchers obtain Levantine data that classified into a fine-grained country level, I set out to create a Levantine corpus -Shami Dialects Corpus (SDC) - that was concerned with the four Levantine dialects, Jordanian, Palestinian, Syrian and Lebanese which are spoken in the Levant.

SDC is the first Levantine corpus that concerns the four dialects; in addition, it contains the largest volume of data from each dialect compared to the previous works. Lebanese dialect is introduced significantly for the first time in this corpus, where it is considered a dialect with little presence on the web, as most Lebanese used Latin characters and French to post on social media or other websites.[1] SDC is built from scratch employing manual and automatics approaches for the collection process. The data are gathered from different platforms (personal blogs, Twitter, Facebook, YouTube, and others) and cover various domains. As I have different platforms, I will call the instance by document or sentence.

### 4.1.1 Data Collection

The first step to build a corpus is by collecting data. I approach data collection in two ways, the automatic way and the manual way. I run both ways in parallel, so I can make good use of the time.

I collect part of the SDC manually to be sure that the corpus addresses different topics and domains, and to use in multi-purpose tasks. No constraints are applied, except the data must be in dialectal form - not MSA - and written in Arabic alphabet. For manual collection, I do the following:

- harvest the web and specify some online blogs that publish stories written in Levantine dialects and then collect the stories as well as the readers comments. Collecting stories that are written in dialects helps in gathering much data and in saving time and is easier than collecting posts or comments one by one.

[1] https://www.internetworldstats.com/languages2.htm

- collect posts from public groups and pages on different social media platforms, such as Facebook where the numbers of followers are considered big. I extract the discussion posts in addition to people's opinions and comments.
- address pages of some Levantine celebrities such as actresses and singers, who usually post using their dialect, and collect some data from their discussions.

The dialectal data are only collected to cover three countries: Palestine, Syria and Jordan. I face a problem with the Lebanese dialects according to the fact that most Lebanese are using the Latin character to write Arabic. In addition to that, in Lebanon they have a bilingual education, so Lebanese often use English or French to post and communicate on the web. Because of that, I do not collect any Lebanese data manually.

According to the Twitter policy of data collection and publishing and the available API for developer and researchers, I choose Twitter as a platform to automatically collect data. To speed up the data collection process I rely on the Twitter API streaming library (Tweepy)[2] to collect as many tweets as possible.

I use two ways to collect and retrieve data from Twitter. The first: I address some accounts of celebrities who use dialect while tweeting from each country. I need their corresponding Twitter IDs, so I use then tweeter id[3] for converting purposes. After that, I apply tweepy streaming to collect tweets and replies from these IDs and each streaming run until I reach 9,999 tweets each time, which is the tool's limit.

This way is not a straightforward process, because I have to search and look for active celebrities on Twitter before automatically retrieving their tweets and replies. The other, and faster way, is by depending on the geographical co-ordinates for the four Levantine countries. I run the code to collect data and retrieve all the tweets and replies using this geo-information.

All of the extracted data are then stored in JSON files with the following information: (i) Tweeter ID, (ii) Data and time, (iii) Tweet and replies, (iv) location and (v) number of likes, shares and replies. When I decide to stop streaming and collecting, I convert all JSON files to text files that are cleaned, and only tweets and replies are kept which are collections of dialects, MSA, numbers and some Latin words.

For the annotation process, I apply a very simple rule. For the manual collecting of data, there was no problem detecting the dialects. Regarding the automatic collecting, I label the data depending on the geo-location I used. Thus if the tweets are retrieved by the geo-location of Syria, then

[2] http://www.tweepy.org/

[3] https://tweeterid.com/

they are classified as Syrian. Also, if the tweets are retrieved by tweet-id for a Jordanian celebrity, then they are labelled as Jordanian.

Table 4.1 illustrates the number of sentences (documents) after automatic and manual extraction for each dialect. It is noticeable that I collected more data using the manual approach; it was not easier than the automatic approach, but most of this manual data were in the form of social stories or long posts, especially the Syrian data.

| | Automatic | Manual | Total | Token | Types |
|---|---|---|---|---|---|
| Jordanian | 11,026 | 24,312 | 35,338 | 518,101 | 76,529 |
| Palestinian | 10,149 | 18,280 | 28,429 | 453,716 | 69,954 |
| SSyrian | 13,349 | 43,811 | 57,160 | 834,054 | 83,470 |
| Lebanese | 19,540 | - | 19,540 | 231,692 | 43,292 |
| Total | 54,064 | 86,403 | 140,467 | 2,037,563 | 273,245 |

Table 4.1: Number of sentences for each dialect in Shami

### 4.1.2 Data Preprocessing

In order to introduce SDC to the research community and make it applicable for any NLP task, a special pre-processing treatment must be performed beforehand. Since the collected data are dialects, I need to apply some cleaning steps in order to get a reliable corpus that can be generally used in NLP applications.

As with any dialectal corpora that have been done before, they all share general pre-processing steps, such as removing of diacritics, non-Arabic symbols, numbers, dates and any letters that do not belong to the Arabic alphabet, see 7.3.2 for more details. In the process of normalization I check the writing style and the effect of applying general normalization rules on the text concerning the meaning, so I put and specify some rules that can keep the corpus more reliable and to save the meaning of the text as much as I can. These rules are mentioned in Study 1. In addition to that, I study the phenomena of lengthen word and analyze the origin of repetition in Arabic text. All the previous works removed the repeated characters and keep only two appearances, while according to our algorithm, I set some criteria to save the semantic and the syntax of the text. Figure 7.2 shows the way the algorithm works when the addressed character is conjunction letter و *w* which is and in English. For more details about the preprocessing rules, see Study 1.

In order to present a reliable Levantine corpus with four classified Levantine dialects, I have done a purification process. As I collected part of the data manually, there is no need to check all of them again. I only check the users' replies on some posts, as they are from different countries and the replies may contain mixed dialects. I manually go over the data collected automatically with the help of some friends from the Levant as volunteers, and insure that the classification label or the assigned dialect matches the text. The point behind this step is to obtain a high quality corpus that can be used to conduct any kind of linguistic research and experiments on it. Table 4.2 illustrates the statistics after data purification.[4]

| | Automatic | Manual | Total | Token | Types |
|---|---|---|---|---|---|
| Jordanian | 8,804 | 23,274 | 32,078 | 472,918 | 68,922 |
| Palestinian | 3,566 | 17,698 | 21,264 | 351,814 | 55,942 |
| Syrian | 4,704 | 43,455 | 48,159 | 701,872 | 62,731 |
| Lebanese | 16,304 | - | 16,304 | 177,623 | 35,621 |
| Total | 33,378 | 84,427 | 117,805 | 1,704,227 | 223,216 |

Table 4.2: Number of sentences for each dialect after purification step

This corpus was used to measure the similarities, differences and the overlapping among the Levantine dialects, as in *Study 3*, Moreover, I employ SDC on an NLP task, namely dialect identification, in *Study 1* to check to what degree these dialects are distinguishable and to see if the Machine Learning models are able to classify them or not. Based on SDC, I built Shami-Senti, which is a Levantine Sentiment Analysis corpus, see *Study 4*.

## 4.2 Shami-Senti

In *Study 3*, I employ different algorithms to measure the overlapping and the lexical distances among the dialects and in regarding to MSA as well. From the study, I have seen that Levantine dialects can be seen as being close to MSA in terms of used words and overlap. Then, instead of wasting time and effort to build stand-alone models for every dialect, I suggest fine-tuning off-shelf tools that have been build for MSA-NLP and apply them to dialects. I decide to build a Levantine Sentiment Analysis corpus to check whether MSA Sentiment Analysis models can perform well on Levantine Sentiment Analysis or not. I extended the SDC corpus from *Study 1* by annotating part of it for sentiment. I call the new corpus Shami-Senti.

Shami-Senti is a manually crafted corpus, where the annotators extract any sentence that contains sentimental words, reviews, opinions, feelings

[4] This is a correction table of Table 7.4 at page 109, as I reported the number of characters instead of the number of words, and the method for calculating the number of vocabularies/types was not accurate

or expressions. The annotators all are educated and I condition them to be from the Levant. They are volunteers as friends, relatives and family members. I put constraints on the length and set it at a maximum of 50 words/sentences. As irony and sarcasm are hard to define and classify, and given that the Sentiment Analysis models' performance is usually affected by them, in this corpus I avoid any sentence that might contain them. Using a manual process, I am able to extract 5K sentences from the SDC. After that, an annotation phase begins.

Before I starting the annotation process, I apply some guidelines. They are as follows:

- The sentence is considered positive if it explicitly or implicitly contains any terms or clues that indicate that the speaker is in a positive state, for example: success, happy, etc.
- The sentence is considered negative if it explicitly or implicitly contains any terms or clues that indicate that the speaker is in a negative state, for example: anger, sad, etc.
- The sentence is considered mix if it contains both negative and positive states.
- If the sentence contains a negation for the sentiment terms, then choose the opposite polarity.

For the annotation process, I utilize two methods (i)lexicon-based approach and (ii)human based approach. The first is done by leveraging the available dialectal sentiment analysis lexicons. I run the process: if the sentence contains a positive term from the lexicon, then it's polarity is positive. If it contains any negative terms, then it is considered negative. Any sentences that contain a mixture of positive and negative terms are marked as mix. I depend on three available lexicons: the one provided by LABR [107], which contains negative, positive and negated terms, the Moarlex [130] and the SA lexicon [131], which contains only positive and negative terms. For more details about the lexicons, see Table 10.1 on page 159.

I implement algorithm1 to automatically annotate 1000 sentences from Shami-Senti, exploiting the three aforementioned lexicons. In addition, I commission a Levantine native speaker to annotate the same samples for sentiment. After computing the inter-annotator agreement between the lexicon annotation and the human annotation, the result was not deemed acceptable, as the disagreement was up to 80%. So, for the sake of reliable annotation and in order to build a well qualified Levantine dataset, I chose to manually annotate the corpus.

Following *Study 3*, where the Levantine dialects show big similarities among each other, I hypothesized that I can employ the result to annotate

```
Result: Annotate 1,000 sentences
Build Positive, Negative, Negation lists of words extracted from the
  three lexicons;
Polarity = 0;
for sentence in Shami-Senti do
    count number of positive terms; Then Polarity ++;
    count number of negative terms; Then Polarity −−;
    check if there is a negation,Then Polarity $* - 1$;
    if Polarity > 0 then
        Polarity is Positive;
    else if Polarity < 0 then
        Polarity is negative;
    else
        Polarity is mixed;
    end
end
```

**Algorithm 1:** Lexicon-based annotation of 1,000 Shami sentences

| **Shami-Senti** | | | |
|---|---|---|---|
| Positive | Negative | Mix | **Total** |
| 1,064 | 935 | 243 | **2,242** |

Table 4.3: Number of sentences in Shami-Senti corpus per category

the data for Shami-Senti. I depended on the similarity between the Jordanian and the Palestinian dialects and ask a Palestinian annotator to do the job for these two dialects; while for Syrian and Lebanese, I assign the annotation to a Syrian annotator given the overlap between both dialects. Before starting the manual annotation, I ask both the annotators to annotate the same 533 samples, and assign a label for every sentence. The inter-annotator agreement is computed using Kappa statistics [132] and it returns a $\kappa = 0.838$.

I extracted more than 5,000 sentences for this purpose, and have annotated 2,242 of them so far. I used the term *sentences* as the corpus contains data from different platforms, as I mentioned it is extracted from Shami. Table 4.3 shows the number of sentences per category.

## 4.3 Arabic Tweets Sentiment Analysis Dataset (ATSAD)

Shami-Senti is a manually crafted Levantine corpus for Sentiment Analysis. Even though it is useful to have such a standalone Levantine corpus annotated for the purpose of Sentiment Analysis and Opinion Mining, it is, however, small in terms of size and has been time consuming to create due to manual annotation. Furthermore small corpora are not very effective with Deep Learning models, because these models need a huge amount of data to train and learn. As a consequence, I decided to build an Arabic Tweets Sentiment Analysis Dataset (ATSAD) by leveraging the unique Twitter features that are widely used to express sentiment. I decided not to extend upon the previous Shami-Senti corpus due to many reasons:

- Shami-Senti has been built and based on Shami corpus, which is not only tweets but different text from various social media platforms.
- Shami-Senti needs a huge effort for manual extraction of the data, to annotate it and to define the dialect.
- I wanted to build a Tweet corpus that can be automatically collected and annotated, which saves time and effort.

I employ an automatic approach to build ATSAD and collect a large amount of applicable data, by first constructing a sentiment emojis lexicon. The lexicon includes almost all sentimental positive and negative emojis used to express feelings on Twitter. The lexicon is constructed by collecting the emojis, as well as their indicated polarity, from two available sources, the "Emojis Sentiment Ranking Lexicon" [133], which is available online,[5] and Emojipedia.[6] The total size of the emoji lexicon is 91 negative emojis and 306 positive ones.

Rather than using the popular ways to collect data from Twitter through query terms and hashtags, I utilize the emojis lexicon as the seed for the retrieval process. Therefore, I use the emojis as query terms to retrieve tweets that contain and include these emojis, the tweet language being conditioned to Arabic. I collect $\approx 60K$ tweets using Twitter API in April 2019, and I apply different pre-processing and cleaning steps on the collected tweets and finish up by $\approx 36K$ tweets. For more details about cleaning step see *Study 6* page197. Thus, the number of tweets dropped as a result of duplicated tweets, Tweets not written in dialect, such as supplications, Quran verses, tweets that only contain emojis, and so on. Table 4.4 shows

[5] http://kt.ijs.si/data/Emoji_sentiment_ranking/

[6] https://emojipedia.org/people/emojis

the statistics of the corpus before and after the pre-processing phase. In addition, I do a purification data to check whether the data is dialectal or not; then I make sure that the final 36K tweets are all written in dialects.

| | Positive | Negative | Total | Types | Tokens |
|---|---|---|---|---|---|
| Before | 30,607 | 29,232 | 59,839 | 95,538 | 76,2673 |
| After | 18,173 | 18,695 | 36,868 | 95,057 | 41,8857 |

Table 4.4: Number of tweets in ATSAD before and after pre-processing

To be able to annotate the whole 36k tweets corpus, I apply distant supervision or weak supervision methods on the dataset[134]. The distant supervision approach works by heuristically matching the contents of a database to the corresponding text [135], which in our case is the emojis. I build an algorithm to automatically use the emojis lexicon for the annotation process and weakly label the text according to the contained emojis that appear on it. So, if the tweets are fetched by the positive emojis from the lexicon, then they are weakly labelled as positive, and if they are fetched by the negative lexicon, they are labelled negative.

Two native Arabic speakers are used for annotation purposes, one of them being an NLP expert; the other is a university educated person. The matches between them are 90% for 1% of the data (180 samples); in case of disagreement, I prefer choosing the expert annotation. The annotation procedure is cumulative, in that I pick randomly a sample of 1% of the data every time and ask both of the annotators to annotate it. For every annotation slot, I calculate the number of mismatched labels between the human annotation and the emoji-based annotation (weak labelling). In addition, I compute the accuracy of the emoji-based annotation method by taking the number of right classified instances divided by the total number of the samples. Table 4.5 shows the number of errors (mismatches) and accuracy for annotation samples in the range from 1% to 10% of the corpus. So, I end up with 4k tweets annotated manually. Figure 12.1 at page 199 plots the accuracy results as well.

Having a 77.2% matching accuracy between human annotation and emoji-based annotation is not considered good enough, even though it is less time-consuming compared to manual annotation and it can annotate reasonable amounts of data in a very short time without human effort. As a result, I decided to figure out ways to enhance the corpus and make it more reliable and applicable to NLP tasks.

Since I manually annotated 10% of the dataset, which was 4k tweets, I ask the annotators to continue and annotate another 10% in order to have 8k tweets annotated manually in total. This is a gold standard 8k Sentiment

| Sample % | Samples | #errors | Accuracy |
|---|---|---|---|
| 1% | 360 | 106 | 70.5% |
| 2% | 720 | 200 | 72.2% |
| 3% | 1,080 | 293 | 72.9% |
| 4% | 1,400 | 370 | 74.3% |
| 5% | 1,800 | 450 | 75% |
| 10% | 3,608 | 823 | 77.2% |

Table 4.5: Human annotation accuracy compared to the emoji-based annotation. The first two columns show the percentage and number of the sampled tweets, #errors shows the number of mismatched samples, and the Accuracy column calculates the percentage of the matches between both annotations.

Analysis dataset of which 3,705 of them are classified as positive, 3911 negative and 384 instances are classified as mixed. In our experiments, I exclude the mixed class. In *Chapter 4* I will explain how I apply self-training methods to the corpus in order to improve quality and reliability.

# 5 Dialectal Arabic NLP tasks

## 5.1 Dialect Identification for Dialectal Arabic

Dialect Identification (DI) refers to many things, and can be used on different levels and with different tasks. Some models have been built to check the appearance of dialect terms in an Arabic text and to measure the percentage of dialects in that text[136]. On the other hand, most of the Dialect Identification systems are considered classification systems. Some works build models for a coarse-grained identification level such as the region level[136], other works introduce fine-grained models on the level of the country[81]. Recently, and according to the differences that dialects have to each other, a very fine-grained dialects system has been proposed, which classifies the dialects at the city level[4].

### 5.1.1 SHAMI (SDC) DI

The language identification task itself is considered a solved problem in general. However, in the case of dialectal Arabic where all Arabic dialects are in a diglossic relation with MSA, the task becomes more difficult.

Part of our contribution - as already mentioned - is to build SDC, a Levantine corpus, in order to be able to evaluate the corpus and test its performance in a real NLP task. I choose the Dialect Identification task to do the job of resource evaluation. In addition, in *Study 1*, I want to shed light on the ability of N-gram language models on the process of differentiation between fine-grained dialects like Leventine. In the process of evaluation I use two kind of libraries, including one off-the-shelf library called Langid.py[137] which uses the Naive Bayes classifier (NB) with various character n-gram models. Langid.py supports the developers with many models in order to be able to train the model and build a language identification system based on their languages. Compared to other language identification tools like langdetect[138], TextCat [139] and CDL [140], Langid.py outperformed all of them in terms of accuracy (measured as the proportion of documents from each dataset that are correctly classified) and speed (documents per second). It also supports the character n-grams, and they are very useful and differentiated features for the DI system. On the other hand, I use *scikit-learn*[141] as an open source python library to build some word-grams models and try different machine learning algorithms. *Scikit-learn* is used to estimate the efficiency of word-gram language models. I then compare it with the character gram models and how they work with the dialects.

During the evaluation process, as I build a DI model, I conduct several experiments. They vary between the size of the used data (full or sample), the type of the data (preprocessed or pure), the implemented libraries (langid.py or *scikit-learn*), the techniques (character gram or word gram) and the number of the classified dialects (two, three, four). For each experiment, I calculate the accuracy and the F-score as an evaluation measurement.

I start by using langid.py and build character n-gram models investigating different grams from 4 to 7, then I assign the features to an NB classifier. First, I do an experiment to decide the size of SDC. When I train the DI models by the whole corpus, the accuracy and the F-score are very low, they do not exceed 39% and 55% respectively through all the introduced models. Table 7.10 from *Study 1* shows the results before the filtering process. The fact that Shami is neither a parallel nor a crafted (translated) corpus, makes it very confusing in some of its dialectal sentences. As a result, I decide to purify the corpus with the help of four volunteers one from each Levantine country. Purifying the SDC, and making sure that the annotators extract their own dialect parts, reduces the sparsity of data and also increases and enhances the performance of the classification models.

Generally, to be able to evaluate the SDC, I build some baseline models using two available corpora which both contain Levantine dialects. They are:

- PADIC[58]: a parallel translated corpus which includes MSA, Algerian, Tunisian, Palestinian and Syrian dialectal data. The corpus is collected from Algerian chats and conversations, where they are transcribed by hand to Algerian text and converted to MSA. After that, 20 Tunisian native speakers in addition to two native speakers from Palestine and Syria produce the final parallel corpus from the MSA part.

- Multi-dialect corpus[57]: a parallel translated corpus that covers the dialects of Egyptian, Syrian, Palestinian, Tunisian and Jordanian in addition to MSA. This corpus has been originally built on the Egyptian dialect using an Egyptian-English corpus. It has been translated to the remaining dialects by four native speakers of Palestinian, Syrian, Jordanian and Tunisian.

As my concern is only the Levantine dialects, I only extract the Levantine dialects from these corpora, so I build a Levantine DI model. The data is split into 90% for training purposes and 10% as a test set. Table 5.1 shows the number of sentences/dialects for the corpora concerned.

In tables 7.8, 7.9 and 7.12 the langid.py library calculates the accuracy and F-score for the model employing different n-grams. The model has been trained on all the dialects in the handled corpus and then tested on one dialect every time so that it appears as a single classification test phase, because the test set contains only one class and therefore the precision = 1 all the time. Therefore, in that case the recall is equal to accuracy and this is why there is a gap between Accuracy and F-score in the results. After evaluating every dialect alone, the average for the accuracy and F-score is calculated but for only the Levantine dialects. I will calculate the evaluation matrix in a different way so the results would give an insight into the Levantine data. I will build the model by only extracting the Levantine dialects for every corpus: for example, in PADIC corpus, I will make use of the Palestinian and Syrian dialects only.

In this approach I can also examine the effect of the majority class, especially in an imbalanced dataset like SDC. Accuracy measures how many cases are identified correctly by the model and it is usually used when the data is balanced so that all the classes are equally important; but in the case of an imbalanced dataset, F-score is used more as the harmonic mean of Precision and Recall and it takes into account the data distribution and gives a better measure and assessment of the model's performance.

The model performed better on the two corpora compared to the SDC, as shown in Table 5.2. This is due to many reason, with listed below:

- The nature of the data: both PADIC and the Multi-dialect dataset are parallel and well-crafted corpora, while SDC is not. SDC has been built from people's posts, tweets and blogs, so these are real conversations and discussions not a human translated dataset, while in a parallel corpus the differences among the dialects are more emphasizing.

- The number of the classified Levantine dialects has an effect on the performance and the accuracy of the system. In addition, the size of the corpus plays a role as well. Although the PADIC corpus contains

| | Dialect | Train | test | Total |
|---|---|---|---|---|
| **PADIC** | Palestinian | 5,917 | 501 | 6,418 |
| | Syrian | 5,917 | 501 | 6,418 |
| **Multi-dialect** | Palestinian | 900 | 100 | 1,000 |
| | Syrian | 900 | 100 | 1,000 |
| | Jordanian | 900 | 100 | 1,000 |
| **Shami** | Palestinian | 9,577 | 1,065 | 10,642 |
| | Syrian | 33,983 | 3,776 | 37,759 |
| | Jordanian | 6,316 | 702 | 7,018 |
| | Lebanese | 9,747 | 1,083 | 10,830 |

Table 5.1: Train and test set for the corpora

| | | **PADIC** | | **Multi-dialect** | | **SDC** | |
|---|---|---|---|---|---|---|---|
| | **Techniques** | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score |
| langid.py | 4-gram char | 0.69 | 0.69 | 0.81 | 0.70 | 0.78 | 0.51 |
| | 5-gram char | 0.74 | 0.74 | 0.83 | 0.75 | 0.78 | 0.52 |
| | 6-gram char | 0.75 | 0.74 | 0.82 | 0.74 | 0.78 | 0.52 |
| | 7-gram char | 0.76 | 0.75 | 0.83 | 0.74 | 0.78 | 0.51 |
| Scikit learn | uni-gram word | 0.83 | 0.83 | 0.69 | 0.68 | 0.70 | 0.71 |
| | bi-gram word | 0.84 | 0.83 | 0.69 | 0.96 | 0.70 | 0.70 |

Table 5.2: Test phase evaluation results for all corpora

two dialects and the Multi-dialect corpus contains three dialects, the small data size of the later corpus reduced the conflicts between the dialects and thus the ease of distinguishing between them. While we also find that the SDC is bigger in terms of size and number of dialects , so the results are never considered satisfactory, and the detecting of the dialects is not easy at all.

- Imbalance SDC: PADIC and Multi-dialect datasets are balanced, where each dialect has the same number of instances; however, SDC is not. This is clear in Table 5.1, where Syrian is the majority class and Lebanese is the least. Having an unbalanced dataset confused the classification model, whereby the majority class dominates the results, and this is why the accuracy is considered high compared to the F-score.

As a consequence, I conduct some experiments on the number of the classified dialects and compare them with the baseline; this would make the comparison more reliable as the same number of dialects. Given that PADIC includes the Palestinian and the Syrian dialects, I extract the two dialects from SDC and train the Dialect Identification models on this binary

| | **Techniques** | **Palestinian, Syrian** | | **Jordanian, Lebanese** | |
|---|---|---|---|---|---|
| | | Accuracy | F-score | Accuracy | F-score |
| **langid.py** | 4-gram char | 0.64 | 0.61 | 0.75 | 0.74 |
| | 5-gram char | 0.64 | 0.61 | 0.76 | 0.76 |
| | 6-gram char | 0.65 | 0.62 | 0.76 | 0.75 |
| | 7-gram char | 0.65 | 0.62 | 0.75 | 0.74 |
| **Scikitlearn** | unigram word | 0.87 | **0.85** | 0.90 | **0.90** |
| | bigram word | 0.80 | 0.74 | 0.88 | 0.88 |

Table 5.3: Evaluation of two dialects classification on SDC

| | **Techniques** | **Pal, Jor, Syr** | | **Leb, Jor, Syr** | | **Pal, Leb, Jor** | |
|---|---|---|---|---|---|---|---|
| | | Accuracy | F-score | Accuracy | F-score | Accuracy | F-score |
| langid.py | 4-gram char | 0.74 | 0.58 | 0.79 | 0.60 | 0.73 | 0.60 |
| | 5-gram char | 0.74 | 0.59 | 0.79 | 0.62 | 0.73 | 0.61 |
| | 6-gram char | 0.74 | 0.58 | 0.80 | 0.62 | 0.74 | 0.61 |
| | 7-gram char | 0.73 | 0.57 | 0.79 | 0.61 | 0.74 | 0.61 |
| Scikit learn | uni-gram word | 0.77 | **0.71** | 0.75 | **0.70** | 0.74 | **0.74** |
| | bi-gram word | 0.70 | 0.60 | 0.70 | 0.60 | 0.73 | 0.72 |

Table 5.4: Evaluation of three dialects classification, Pal: Palestinia, Jor: Jordanian, Sy: Syrian, Leb:Lebanese

classification. In the same manner, I also do a binary classification with Jordanian and Lebanese dialects. Table 5.3 shows the results. In the same way, I do three-way classification experiments, taking into consideration that the Multi-dialect corpus includes three dialects (Palestinian, Jordanian and Syrian). I extract the same dialects, and in addition, I run another experiment excluding the Palestinian dialect because it is very mixed with other dialects. I present the results in Table 5.4.

The more dialects that enter into the identification system, the more difficult it is to distinguish between them, and thus the performance decreases. The SDC corpus out performs the PADIC base line when I compare the same dialects at the uni-gram word level. The same applied on the comparison between SDC and the Multi-dialect corpus on the uni-gram word model. The effect of an unbalanced dataset is very clear on SDC where there is a big gap between accuracy and F-score. As accuracy is biased towards the majority class, it achieved higher results compared to F-score as a trade-off between precision and recall.

Generally speaking, it seems to us that if the number of dialects is reduced, the accuracy and F-score increases more, as the dispersion in the data decreases. Also, the clarity of the difference becomes slightly clearer. When I classify between Jordanian and Lebanese, I get the highest results, as there is little similarity between them and they are, to some extent, different. They can then be distinguished by text. From all the previous experiments, I conclude that the Levantine dialects of SDC are very similar in the written texts and that it is difficult to differentiate between them. SDC is not a parallel data set so that textual differences are evident and clear as in PADIC and Multi-Dialect corpora. However, it does address similarities and differences between the four dialects significantly compared to the two previous corpora.

### 5.1.2 MADAR DI

I extend my work on Levantine Dialect Identification and build a machine learning model that is able to distinguish very fine-grained dialects. I participated in the MADAR shared task and my model is ranked first[142]. I build a model that combines different kinds of n-gram models and employ ensemble learning with a voting technique to classify 26 dialects from 25 Arab cities in addition to the MSA. Two corpora are used for the training processes. The first corpus is Madar-6, which includes five dialects from the Arab region as well as the MSA, while the second is Madar-26. Table 5.5 shows the number of sentences/samples per dialect and the total sentences for each data set.

| MADAR | Split | sentences | Total |
|---|---|---|---|
| **Madar-6** | train | 9,000 | 41,600 |
| | dev | 1,000 | 6,000 |
| **Madar-26** | train | 1,600 | 41,600 |
| | dev | 200 | 5,200 |
| | test | 200 | 5,200 |

Table 5.5: Statistics for MADAR datasets

I introduce a Dialect Identification model that was ranked the first in the shared task. It consists of two sub-models as shown in Figure 5.1. The coarse-grained sub-model is responsible for predicting a dialect among six different Arab dialects, followed by a fine-grained sub-model that goes much deeper to classify 26 Arabic dialects. For the purpose of programming and implementing the model, I use the *scikit-learn* python library. I start with a feature engineering process, which is highly dependent on language modelling and try to explore different combinations of n-grams on several levels (words and characters). To combine multiple features, I use the *Feature-Union* class, and in order to emphasize one feature over the other, I use the transformation weight parameter to give a weight for every extracted feature. I try to extract as many discriminated features as possible that can be employed efficiently to distinguish among the desired 6 and 26 Arabic dialects. This is with the help of the language model, which is an informative way to represent the language.

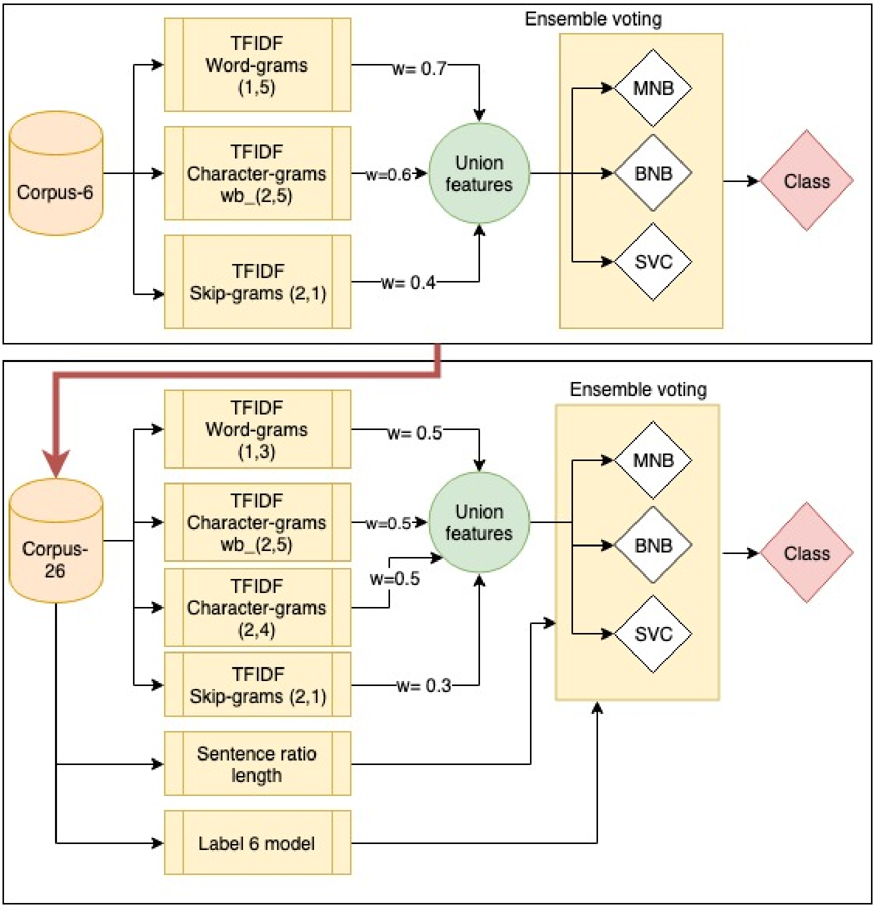


Figure 5.1: ArbDialectID proposed model

For the first sub-model (coarse-grained classifier), the number of dialects - based on the cities where they are spoken - is six. They are (using the respective cities): Beirut (BEI), Cairo (CAI), Doha (DOH), Rabat (RAB), Tunisia (TUN), in addition to (MSA). The differences between this dialect group are reasonably clear, as each group represents a large group of close dialects. For example, BEI includes all the Levantine dialects. As a result, the groups of dialects seem distinguishable by their vocabularies, terms and expressions. Thus, I emphasize the transformation weight of the word-gram level over the character-gram level. The following features are extracted with the transformation weights:

- TF-IDF vectors from the word n-grams ranged from the unigram to 5-grams. I apply 0.7 weight for vector transformation.
- TF-IDF vectors from the character n-grams with boundary consideration ranged from bigrams to 5-grams, and the transformation weight is 0.6.
- Apply skip grams, followed by extraction of the uni-gram words with one-word skipping. I give it the lowest transformation weight of 0.4.

To get the most out of the system, I exploit ensemble learning, where several classifiers can be used and then different voting techniques are applied to the final result. In this model, I build a hard voting ensemble classifier, where it uses predicted class labels for majority rule voting. The following algorithms are used for the ensemble classifier with their updated parameters:

- MultinomialNB (MNB); I set alpha to 0.01
- Linear SVC with l2 penalty and the learning rate sets to 0.0001
- BernoulliNB (BNB); set alpha = 0.01

Using the Madar-6 dataset, I train the model and get an accuracy of 92.7% and a macro F-score of 93%. As a sub-model, I combine the train-set and the dev-set together and rebuild the model again. The predicted output value will be used as an input feature to the second fine-grained sub-model.

The fine-grained model is very challenging due to the similarities among the dialects at the level of the cities and to the fact that this is a written text which has no accents, no intonation and is diacritics free. From Figure 5.1 the fine-grained sub-model looks to some extent similar to the coarse-grained model; however, it is trained on more features with different weights.

This Dialect Identification system is supposed to be able to predict the class of a given sample/sentence among 26 classes/dialects. MADAR corpus covers 25 cities in the Arab countries in addition to the MSA, and they are : Aleppo (ALE), Algeria (ALG), Alexandria (ALX), Amman (AMM), Aswan (ASW), Baghdad (BAG), Basra (BAS), Beirut (BEI), Benghazi (BEN), Cairo (CAI), Damascus (DAM), Doha (DOH), Fes (FES), Jeddah (JED), Jerusalem (JER), Khartoum (KHA), Mosul (MOS), Muscat (MUS), Rabat (RAB), Riyadh (RIY), Salt (SAL), Sana'a (SAN), Sfax (SFX), Tripoli (TRI), Tunisia (TUN) and MSA. In this step the differences are very small between dialects, as most of them are concentrated on the character level like the suffixes, so I emphasize the weight of character n-gram features over the words n-gram features and pay attention to the word boundaries. All the extracted features are chosen empirically after an enormous number of experiments, and they are:

- TF-IDF vectors from the word n-grams with uni-gram, bi-gram and tri-gram words. I apply 0.5 weight for vector transformation.
- TF-IDF vectors from the character n-grams with word boundary consideration ranged from bi-grams to 5-grams and the transformation weight is 0.5.
- Extract another character n-grams but this time without word boundary consideration from bi-grams to 4 grams and the transformation weight is 0.5.
- Again apply skip gram, then I extract the uni-gram words with one work skipping. I assign it 0.3 transformation weight.

Moreover, I add two extra numerical features to the model. I use the sentence length ratio to shed light on the differences between the dialects where some of them use more words to express an idea, and the others use more suffixes. The second feature is the predicted output class fed from the first sub-model. I use the coarse-grained sub-model to predict the class for the input sentence to which one of the six groups it may belong. The same ensemble learning classifiers are used as the first sub-model, but it is trained on Madar-26 this time. I reach 67.29% and 67.32% for accuracy and F-score respectively.

I print out the confusion matrix in figure 8.3, and point out that some dialects are easier to classify by the model, such as the dialects spoken in North-Africa. In contrast, there are other dialects that confused the prediction system and lower the performance, such as the (BAG and BAS), (AMM and JER), (CAI and ASW).

## 5.2 Computational cross-dialectal lexical distance

Diglossia is quite pervasive in the Arabic speaking countries. Diglossia takes place when the spoken language (such as Arabic dialects) is different from the official language (such as MSA). As a consequence of building the previous Levantine resources and proposing Dialect Identification models, I decide to conduct a computational cross-dialectal lexical distance study to measure the lexical distance between the formal MSA language and the informal Arabic varieties.

From the point of view of Arabic speakers, they believe that it is easy to discover the similarities between the dialects and distinguish between them in spoken or written forms, whether they are similar or different. They may state and debate which dialects are the closest to the Classical Arabic or the MSA, while originally such topics need a precise scientific study to determine the similarities, differences and convergence between dialects instead of relying on human intuition. Many differences between dialects and MSA should be highlighted. In Section 2.4, I discussed the qualitative differences in terms of orthography, phonology, morphology, syntax, semantics, and lexical differences. Here in this section, I highlight the lexical quantitative similarities and differences through the use of different computational techniques.

One of the important features to understand the differences between the official language (MSA) and the informal varieties (Dialects) is the lexicon[19]. In order to conduct the lexical distance study, I use different kinds of resources: these include various resources, and parallel, un-parallel and comparable corpora. I try to include as many dialects as possible, so the corpora concerned include the dialects from most of the Arab regions. It is extremely hard to find a parallel corpus that contains all the dialects from the Arabic speaking countries in order to be used for such a comprehensive study. In addition to PADIC[58] and Multi-dialectal[57] - where they are mentioned in 5.1.1 - I employ the following corpora:

1. SDC: Our non-parallel Shami corpus with the four dialects from the Levant. It includes Palestinian, Jordanian, Syrian and Lebanese dialects.

2. WikiDocs Corpus[109]: this is a Comparable dataset, which contains comparable documents from Wikipedia. The documents are in MSA and Egyptian.

### 5.2.1 Lexical Sharing and overlap

I compute the percentage of sharing vocabularies and the overlapping between any pairs of dialects. I employ the Jaccard Index similarity measure on the dataset.

$$JaccardIndex(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|} \tag{5.1}$$

Table 9.8 at *Study 3* presents the overlapping results. The experiment shows that the Palestinian was the closest dialect to MSA in both the parallel datasets. In addition, it shows that the highest percentage of overlap among the Levantine dialects is between Jordanian and Palestinian. The overlap between MSA and Egyptian in the WikiDocs is the lowest.

Table 5.6 presents the whole experimental results when comparing MSA to other dialects in PADIC, while Table 5.7 concerns the Multi-dialect corpus.[1]

### 5.2.2 Vector Space Model (VSM)

Mathematically, a VSM converts all the documents/sentences to vectors in a high dimensional space where each dimension corresponds to a term from the collection. A weighting schema is then used to assign a weight to every term from the dimensional space[143, 13]. In our work, I use TF-IDF for the weighting procedure, followed by computing the cosine similarity between each pair of vectors to indicate the rank of the documents. I index all the words, including the stop-words, since they work as clue features for some dialects. To avoid Out-Of-Vocabulary (OOV) terms, I build a vector for each pair of dialects. The results show that in both of the parallel corpora the Palestinian dialect is the closest to MSA, while the North Africa dialects are the farthest from MSA. The results in Table 9.9 also show a very high similarity across the Levantine dialects, while producing less than half the similarity between MSA and Egyptian in the Wiki-Docs corpus.

### 5.2.3 Latent Semantic Indexing LSI

I employ LSI to be able to represent the concepts that each dialect contains. The key feature of LSI is that it addresses the problem of dealing with synonyms and, in general, polysemy among the terms. It utilizes the Singular Value Decomposition algorithm to reduce the dimensions of the term matrix and extracts the most informative features from the document matrix[13]. In terms of concept similarities, the Palestinian dialect shows a closer similarity to MSA in one of the parallel corpora only, while in the rest of the parallel corpora, it is the Tunisian dialect that gets the closest to MSA over all the parallel corpora. The relation between Jordanian and Syrian dialects is the highest among all the experiments. The similarity between MSA and the Egyptian is high as well, and this is due to the same topic that each pair of documents talk about in the comparable corpus. Table 9.10 represents the results. This method is different from VSM, where LSI cares about the concepts and the meaning behind the terms; however, it is only about the terms and their structure in VSM. Then the nature of the used models affects the degree of similarity and the kind of similarity I am looking for.

[1] For further results, see Study 3.

| | *MSA to:* | | | |
|---|---|---|---|---|
| *Method* | **ALG** | **TN** | **SY** | **PA** |
| **Overlaping** | 0.1 | 0.14 | 0.14 | **0.19** |
| **VSM** | 0.27 | 0.38 | 0.37 | **0.50** |
| **LSI** | 0.68 | **0.75** | 0.69 | **0.75** |
| **HD** | 0.91 | 0.83 | 0.77 | **0.77** |
| **PCC** | 0.76 | **0.92** | 0.67 | 0.85 |

Table 5.6: The relation between the MSA and all the dialects in PADIC corpus

| | *MSA to:* | | | | |
|---|---|---|---|---|---|
| *Method* | **EG** | **JO** | **TN** | **SY** | **PA** |
| **Overlaping** | 0.21 | 0.14 | 0.13 | 0.15 | **0.16** |
| **VSM** | 0.5 | 0.38 | 0.37 | **0.4** | **0.4** |
| **LSI** | 0.72 | 0.37 | **0.75** | 0.4 | 0.41 |
| **HD** | 0.01 | 0.77 | **0.76** | 0.78 | 0.78 |

Table 5.7: The relation between the MSA and all the dialects in the Multi-dialect corpus

### 5.2.4 Hellinger Distance HD

This is a method that is used to measure the differences between two probability distributions[11]. Firstly, I use a bag-of-words model to represent each document as a discrete probability distribution vector and then I apply a Latent Dirichlet Allocation (LDA) to model these vectors[144]. LDA gives us a probability distribution over a specific number of topics - thus, it acts as a soft clustering technique. After that, HD is employed to measure the distance between the topics and the documents. HD is an inverse equation, measuring the extent of the distance, not the similarity, and therefore the greater the value, the less the similarity and vice versa. Since this method is similar to the previous one in that they both work on the topic distribution level, the results are somewhat close, as shown by Table 9.11. The Palestinian dialect appears again as the least distant dialect from MSA in PADIC corpus, while Tunisian is in the lead in the Multi-dialect corpus. The same case applies to the Jordanian and Syrian dialects in the SDC, as their differences are the lowest.

### 5.2.5 Pearson Correlation Coefficient PCC

I build a bag-of-words dataset for all the shared terms included in the dialects with their Term-Frequency (TF) and in addition, I extract the 30 most frequent words for each dialect. They are listed in Table [9.13]. Based on them, I calculate the PCC between the two sets with respect to their frequency. This experiment helps us to address the differences in the usage of frequent words across the dialects. The relation is the highest between MSA and the Tunisian dialect followed by the Palestinian dialect. The relation is also strong between Jordanian and Syrian, as they share many terms. The results are presented in table 9.12.

In all the experiments I have seen so far, it seems that the Palestinian dialect, and sometimes the Tunisian dialect, are the closest to MSA, followed by Syrian. In general, Algerian dialect seems the furthest from MSA. This might be due to the extensive mixing and/or code-switching of Algerian with French and Berber. In the Multi-dialect corpus, the Egyptian dialect shows a high degree of closeness to MSA. However, this corpus has been built by extracting the Egyptian part of the Egyptian-English corpus[65] and then translating it to MSA. Thus, a bias towards MSA as an artifact of the dataset construction process exists. It seems in this respect that the nature of the used corpus and the employed methods play a role in the measurement scores and, furthermore, emphasize the relationships among the varieties.

Overall, I show the degree of convergence between the dialects of the Levant and the linguistic overlap to such an extent that, in some cases, it seems impossible to distinguish between them in writing without the presence of phonological information or without adding diacritics marks. The differences among the dialects in comparison are very clear in the qualitative study while in the quantitative study it is more complicated due to the nature of the corpus and the methods I use.

Palestinian dialect has more sharing and overlapping words with MSA than others, and it is the most similar dialect to MSA when I applied VSM , while the Tunisian and Algerian dialects are furthest from MSA. When I measured the similarities with LSI, I found that Palestinian appears to be close to MSA only in PADIC, whereas the Tunisian dialect shows a close relation to MSA in both corpora. These results show the artefacts of the LSI model which connects the data according to topics and clusters. By exploiting the HD, I could see that Palestinian is less dissimilar from MSA compared to the rest of the dialects in PADIC. Even though in the Multi-dialect corpus the results for the distance of all dialects to MSA is quite close, the Tunisian seems to be the closest to MSA. Again, by computing the correlation coefficient among the varieties, the results show high correlation for the frequent words between the MSA and Tunisian, followed by the

Palestinian dialect in PADIC. This sheds the light on the different usage of frequent words cross dialects. In general over all the experiments I find that on Shami I can demonstrate a high similarity between individual Levantine dialects. These similarities are also very clear in the Levantine dialects which appear on the other corpora as well.

The study also helped us to figure out the differences between the written and spoken forms of the dialects. Some of the Arabic varieties look close to each other in their spoken form, where intonation and other phonological attributes are at play, and the accent and the vowels are very clear. However, this closeness is lost when I move to their written form, because these phonological properties are not present.

## 5.3 Sentiment Analysis for Dialectal Arabic

### 5.3.1 From MSA to Dialectal Arabic

In the previous section, and according to the experiments performed, I saw that some dialects are much closer to MSA than others. Given this finding, the question is whether I am able to employ tools that have been built and used for MSA and adapt them for Dialectal Arabic. More precisely, is it worth employing the already existing tools with minor modifications and fine-tuning, or do I need to build and train specific models for each individual dialect? Moreover, it was mentioned in *Study 3* that Palestinian was one of the closet dialects to MSA.

In *Study 4* I investigate the following research questions

- To what degree can I adapt tools which were trained on MSA and use them on DA?
- Are MSA and DA, in terms of building NLP resources and tools, considered as one language? Or should I consider them as different languages for which I need to build a stand alone tool for every variety?
- In case I cannot adapt MSA models on DA, can I adapt one dialect onto another?

For these reasons, I have built a Levantine corpus in *Study 1*, and I choose the Levantine dialects to be examined in the upcoming study. As a case study task, I chose Sentiment Analysis. In order to evaluate how well the Machine Learning models on MSA perform on Levantine, I have to build a Levantine Sentiment analysis corpus. I build Shami-Senti for this reason, a 2K documents/sentences corpus that is extracted from Shami and are annotated with the help of human annotators. To estimate the performance of SA on Shami-Senti, I firstly have to build and train the models on MSA.

This I do by training on the LABR dataset which is a book-review sentiment analysis dataset[107]. LABR is a large dataset of 63K reviews written mostly in MSA with little dialectal presence. The dataset is available online in different subsets according to the number of the sentiment polarities (two, three, four or five-way classification), and balanced and unbalanced versions. I focus on binary and three-way classification, and, thus, the subsets with two and three classes are chosen.

I employ a number of the well-known Machine Learning algorithms to figure out the most suitable to learn from the MSA data and apply them on DA. The following algorithm are used for the training and the testing purpose:

1. Logistic Regression (LR): is one of the simplest algorithms used in Machine Learning (ML) for binary classification tasks and usually used as a base line. LR is strong in explaining the relationship between one dependent variable and independent variables. Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification[145].

2. Passive Aggressive (PA): a family member of online learning algorithms for large-scale learning. It does not require a learning rate; however, it includes a regularization parameter [146].

3. Linear Support Vector classifier (LinearSVC): like RG it is a discriminative model. LinearSVC is developed from Support Vector Machines algorithms. It is very effective in high dimensional spaces. In addition it is still effective in cases where the number of dimensions is greater than the number of samples [147]. In sklearn, LinearSVC comes with the parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of the penalties and the loss functions and, thus, should scale better to large numbers of samples [141].

4. Stochastic Gradient Descent (SGD): is a linear classifier which implements regularized linear models with stochastic gradient descent (SGD) learning. The term “stochastic” refers to the fact that the weights of the model are updated for each training example [148]. I used it as a simple baseline classifier related to Neural Networks.

5. Multinomial Naive-Bayes (MNB): implements the NB algorithm for multinomially distributed data. The basic idea of NB is to find the probabilities of classes assigned to texts by using the joint probabilities of words and classes. MNB, then, is considered a generative model and is suitable for classification with discrete features (e.g., word counts for

text classification). The multinomial distribution normally requires integer feature counts and it works well for fractional counts such as TF-IDF [149]. Generally speaking, an NB classifier will converge quicker than discriminative models like logistic regression, so one needs less training data.

6. Bernoulli Naive-Bayes (BNB): Like MNB, this classifier is suitable for discrete data. The difference is that while MNB works with occurrence counts, BNB is designed for binary/boolean features [150].

7. Complement Naive-Bayes (CNB): is one of the NB variant, where CNB is particularly suited for imbalanced data sets. Specifically, CNB uses statistics from the complement of each class to compute the model's weights. Further, CNB regularly outperforms MNB (often by a considerable margin) on text classification tasks.[2]

8. Ridge Classifier (RC): Ridge regression (RG) is an extension for linear regression. It's basically a regularized linear regression model. It uses the L2 Regularization technique for a penalty to work as a classifier. A very important fact I need to note about RG is that it will not get rid of irrelevant features but rather minimize their impact on the trained model [151].

9. Perceptron (PR): is a simple classification algorithm suitable for large scale learning and the basic algorithms for neural networks. In sklearn it does not require a learning rate and it is not regularized. The model is updated only on mistakes. This algorithm is faster than SGD using hinge loss [152].

I start the experiments with three-way classification and use the LABR-3 balanced data set with 6,850, 6,578 and 6,580 instances for positive, negative and mixed polarity respectively. The data set is split into 80% for training and 20% for testing. I build the two language models as used by the LABR (baseline): the first is a unigram word model and the second is a combination of unigram and bigram word language models. By employing the same five algorithms that have been used by the baseline I achieve a very low accuracy that does not exceed 60% for any algorithm. Given that the goal is to discover the ability to recognize dialectal Arabic by leveraging MSA models, I build the LABR bi-gram combination model and test it with the Shami-Senti test set. The system's accuracy significantly drops by more than 10%, and achieves 45% accuracy with the SGD-Classifier. In the meantime, when the model is trained and tested in the same dataset Shami-Senti,

[2] https://scikit-learn.org/dev/modules/naive_bayes.html

| **Classifier** | **Accuracy%** | | |
|---|---|---|---|
| | LABR Baseline+ | Test dataset: Shami-Senti | |
| | | **Train:LABRˆ** | Train:Shami-Senti* |
| Logistic Regression | 59 | **46** | 62 |
| Passive Aggressive | 58 | **43** | 64 |
| Linear SVC | 58 | **44** | 64 |
| Bernoulli NB | 34 | **11** | 48 |
| SGD Classifier | 59 | **45** | 65 |

Table 5.8: Applied the baseline on different experiments, + train and test dataset: LABR, t̂rain dataset: LABR, *train dataset: Shami-Senti

the accuracy increases to 65%. Thus, the adaption in this case between the MSA and Levantine dialects fails. Table 5.8 shows the results for the baseline experiments.

Accordingly, I propose a new language model as the previous two models perform very poorly. Our new models focus on character n-gram language models in addition to word n-gram language models. I add 2, 3, 4 and 5 character gram models and also combined all the features for the final model. I use all the aforementioned algorithms with the proposed models and apply them directly on LABR3. The proposed model gets a very slight improvement over the baselines, with a 1% increase. I will further use this model throughout all the upcoming experiments.

The proposed model has been tested on the Shami-Senti as well. The highest accuracy was 50%. On the other hand, I trained the same configuration setting on Shami-Senti and got an accuracy of 71% with the NB classifiers. All the results are shown in Table 5.9. These results indicate that MSA models are not transferable to DA, and, probably, a stand-alone DA model should be used for dialectal Arabic NLP tasks.

The overall accuracy in the previous experiments is not high. Having a mix of neutral classes in LABR confuses the model. The neutral class is a mix of positivity and negativity in both ways, so most of the miss-classified instances are due to confusion. I omitt the third class and conduct the experiments with binary classification. Table 5.10 shows the results for the baseline (MSA experiment), the adaptation model (MSA to DA) experiment and the Dialectal Arabic experiment(Shami-Senti). Despite the improvement in the accuracy when I moved from a three-way classification to a binary classification, however, the adapted MSA models are still unable to classify Levantine dialectal Arabic with good accuracy.

I do a small experiment to check the extent of transfer and adaptation between the dialects. I choose the ASTD corpus[7] for this task. This is an Egyptian dialect corpus collected from Twitter, and it contains 10k

| **Classifier** | **Accuracy%** | | |
|---|---|---|---|
| | LABR model+ | Test dataset: Shami-Senti | |
| | | **Train:LABRˆ** | Train:Shami-Senti* |
| Ridge classifier | 59 | **43** | 69 |
| Logistic Regression | 60 | **46** | 67 |
| Passive Aggressive | 58 | **43** | 68 |
| Linear SVC | 59 | **45** | 69 |
| SGD Classifier | 60 | **50** | 68 |
| Multinomial NB | 59 | **40** | 71 |
| Bernoulli NB | 49 | **44** | 71 |
| Complemtn NB | 59 | **42** | 71 |

Table 5.9: Applied the proposed model on different examples, Applied the baseline on different experiments, + train and test dataset: LABR, t̂rain dataset: LABR, *train dataset: Shami-Senti

| | **Accuracy** | | | | |
|---|---|---|---|---|---|
| | Baseline model | | Proposed model | | |
| | Train:LABR | | | | Train:Shami-Senti |
| **Classifier** | LABR | **Shami-Senti** | LABR | **Shami-Senti** | Shami |
| Ridge classifier | 81 | **54** | 83 | **57** | 73 |
| Logistic Regression | 80 | **56** | 82 | **58** | 74 |
| Passive Aggressive | 81 | **53** | 82 | **56** | 73 |
| Linear SVC | 81 | **55** | 83 | **58** | 73 |
| SGD Classifier | 82 | **54** | 83 | **56** | 73 |
| Multinomial NB | 80 | **53** | 82 | **55** | 74 |
| Bernoulli NB | 76 | **47** | 74 | **48** | 72 |
| Complemtn NB | 80 | **53** | 82 | **55** | 75 |

Table 5.10: The binary classification accuracy results for both the baseline and the proposed SA models. The first four columns refer to the models that LABR (MSA) is used as train dataset , while the last column refers the models when Shami-Senti is used as train and test dataset.

tweets classified into objective, positive subjective, negative subjective and mixed subjective. Our proposed model achieves an accuracy of 83%, while it dropped to 57% when the model is adapted to Levantine dialects.

From the obtained results I can see again that the models fail to classify dialectal Arabic when they are built for MSA purposes. The idea of transferring models or adapting the models from one variety to the other is not applicable as every dialect has different features which seem not adaptable.

It is clear that the performance of the model on Shami increased to be 74%. This is partly due to the accurate human annotation. However, the size of the corpus is small compared to corpora like LABR and ASTD, so the performance increases especially when I ignore the mixed or the neutral class. I examine the effect of adding features to the models for the performance of a small size dataset. As such, I conduct our last experiment by using three Sentiment Analysis lexicons; The LABR lexicon[107], Moarlex[130] and the SA lexicon[131]. I employ them to calculate the percentage of positive terms and negative terms in every document. In addition, I add the feature of negation, if the document contains such terms.

In the implementation, I exploit the *FeatureUnion*, which is an estimator that concatenates results of multiple transformer objects to combine language models in addition to the three features (positivity, negativity, negation). To emphasise the impact of the language model I double its feature weight to 2 and assigned 0.4 for positive term features, 0.2 for negative term features and 0.4 for the negation feature. All of these weights have been chosen after a long number of trials. I run the experiment with 3-way classification and achieve 75.2% by NB classifier (it was 71% at Table 5.9). Thus adding more informative features can help the overall model to learn and predict correctly.

I notice that feature engineering had more effect on DA than MSA, as it adds more values to the small datasets like Shami-Senti while it has minor effects on big datasets like LABR. Therefore, adding more informative features to a small dataset helps the system to learn and predict the correct class. After all experiments, our proposed model outperforms the baseline on both big and small datasets, and gets an accuracy of 83% for MSA and 75.2% for Shami-Senti.

Finally, I can say that MSA models cannot be easily, if at all, used in dealing with DA. There is, thus, a growing need for the creation of computational resources, not only for MSA, but also for DA.

### 5.3.2 Deep Learning For Dialectal Arabic Sentiment Analysis

In this part, I investigate the use of Deep learning for Dialectal Arabic. As in the previous experiments, I tried to adapt the available tools from MSA for DA; however it was not applicable. I also showed how different Machine Learning algorithms work for dialectal resources such as Shami-Senti and ASTD. Hence, I introduce a deep neural network that combines Bi-directional Long Short Term Memory Networks (Bi-LSTM) with Convolutional Neural Networks (CNN) to predict the polarity of an Arabic text and classify it to either positive or negative. I employ the same corpora as before: LABR, ASTD and Shami-Senti. These are different in terms of size, sources and the dialects. The proposed deep-learning model outperforms the state of the art for the ASTD corpus, and produced an improvement for Shami-Senti.

First of all, I represent text by employing the Arabic pre-trained word embeddings AraVec[153]. AraVec is pre-trained on multiple sources such as Twitter and Wikipedia and implemented by the help of Word2Vec[154]. To be able to compare our proposed model I build two different baselines, implemented by *Keras* library. The first baseline consists of an embedding layer followed by two LSTM layers of 128 and 64 output units respectively, then a fully connected layer with a 0.5 dropout and finally a dense Sigmoid layer to classify the input document. I have tried different combinations of LSTM and BI-LSTM models on the three datasets, and I work on both three-way and binary classification. The baseline models accuracy is too low and could not predict the label, see Table 11.3. In some cases, the binary classification gave us, unexpectedly, very high accuracy as shown in Table 11.4. However, this is due to overfitting and data imbalance. Basically, the model was biased towards the majority class. For more details about the network settings and the results, please see Table 11.2.

For the second baseline, I choose the winner model from the Kaggle Sentiment Analysis competition that was developed mainly for English. The model achieved an accuracy of 96%. It consists of an embedding layer followed by a CNN layer (64 filter and 5 kernels), then a max pooling layer of size 2 to feed into an LSTM layer with 30% dropout. The last layer is a softmax layer with only one output as it was a binary classification competition. When I build the model and use it on the three corpora trying both three-way and binary classification, the models fail to correctly predict the polarity. In a similar way to our baseline, this model is also biased towards the majority class in the imbalanced dataset (LABRA and ATSAD), while the accuracy in the other dataset did not exceed the 60%. For all the settings and the result, see *Study 5.*

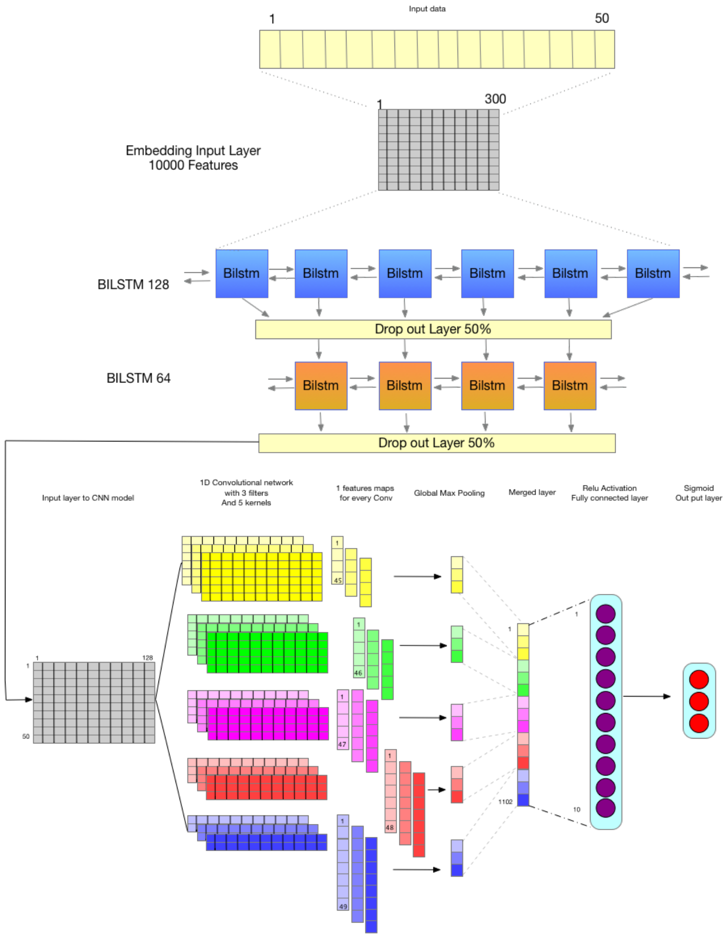


Figure 5.2: Proposed model with BiLSTM and CNN networks

I propose a sophisticated Deep Learning model that combines both LSTM and CNN architectures to retrieve as many features as possible from the datasets. Figure 5.2 shows the structure of the proposed model. I start the model by introducing an embedding layer training with AraVec(300) and set it to be trainable. After that, and in contrast to the baseline, I have two Bi-LSTM layers (128,64 units) followed by several CNN layers. I assume that way that I will extract as many informative representations from the sequential text in both directions. The Bi-LSTIM layers consist of 128 and 64 units respectively, and they feed into several CNN layers with different

| **Parameter** | **Value** |
|---|---|
| Dataset split | 80% train, 10% development, 10% test |
| Max number of features | 15 |
| Embedding size | 300 |
| Embedding model | **CBOW** |
| Embedding trainable | True |
| Max sample length | 70 |
| CNN Filter | [23, 64, 128] |
| CNN Kernel size | [1, 2, 3, 4, 5, 6] |
| Pool size | [1, 2, 3, 4, 5] |
| Batch size | 50 |
| Max epoch | 100 |
| Dropout | 0.5 |
| Optimiser | **Adam**, **RMSprop** |
| Activation function | **Sigmoid**, **Relu** |

Table 5.11: General parameters for the proposed LSTM/CNN model

filters and kernels. After every CNN, I put a max-pooling layer, then a general concatenated layer to merge all the outputs into one dimension vector. In the end, I have a fully connected RELU layer with 10 outputs followed by a Sigmoid layer of three classes for three-way classification or one binary unity for binary classification. Table 5.11 shows the general parameters that have been used to build the model.

The model achieves high accuracy ranging between 80% and 94% for binary classification as shown in table 5.12. The problem of the unbalanced dataset is still presented and the model is unable to predict the right class due to the highly imbalance between the two classes, which is the case for the LABR 2 unbalanced dataset. Getting a high result for small datasets like Shami-Senti is considered an improvement compared to the baselines. I prefer to print out the confusion matrix in Table 5.13 to show the predictions of the model. In the three-way classification, although the model performs better than the baselines, the third category (mix or neutral) has a negative effect on the overall accuracy. Despite that, the proposed model outperforms the state of the art for Deep Learning models for ASTD. I discussed the problem of the third class in *Study 4*.

I performed this study to investigate how Deep Learning models fare with respect to different sizes of datasets in addition to unbalanced corpora. As Dialectal Arabic is considered under-resourced, the size of the corpora decreases the performance of the Deep Learning models, and I know that Deep Learning models need a big amount of data to be trained in, in order

| Corpus | **Three-way Classification** | | | **Binary Classification** | | |
|---|---|---|---|---|---|---|
| | **LSTM-CNN** | Kaggle | LSTM | **LSTM-CNN** | Kaggle | LSTM |
| Shami-Senti | 76.4% | 49% | 53% | 93.5% | 25.3% | 54.5% |
| LABR 2 unbalanced | | | | 80.2% | 80.6% | 55.34% |
| LABR 2 balanced | | | | 81.14% | 53.1% | 81% |
| LABR 3 | 66.42% | 60% | 41.9% | | | |
| ASTD | 68.62% | 59.3% | 53% | 85.58% | 70.7% | 68.5% |

Table 5.12: Accuracy of the proposed model and comparing results from the two baselines

| **ASTD corpus** | | | |
|---|---|---|---|
| | | Predicted | |
| | | POS | NEG |
| Actual | POS | 46 | 18 |
| | NEG | 13 | 138 |

| **Shami-Senti** | | | |
|---|---|---|---|
| | | Predicted | |
| | | POS | NEG |
| Actual | POS | 94 | 4 |
| | NEG | 9 | 93 |

| **LABR2 Balanced** | | | |
|---|---|---|---|
| | | Predicted | |
| | | POS | NEG |
| Actual | POS | 561 | 80 |
| | NEG | 168 | 506 |

Table 5.13: Confusion matrix for the proposed model in the ASTD, Shami-Senti and the LABR 2 balanced corpora.

to achieve reasonable competed results. However, the proposed model could generalized the task and produced good results

### 5.3.3 Distant supervision and Self Training for Dialectal Arabic

As discussed in *Chapter 2*, there is a big need for dialectal Arabic resources. Meanwhile, it takes considerable time and resources to manually build and annotate the corpus for specific tasks. As I have built Shami-Senti manually, and its small size decreased the performance of the models, I therefore decided to build an Arabic Tweets Sentiment Analysis Dataset ATSAD.

I discuss the process of building ATSAD and the annotation procedure in *Chapter 3* where the intrinsic evaluation takes place. Here, I will talk about the corpus extrinsic evaluation and assess it with respect to its impact on a Sentiment Analysis model. To evaluate the corpus I present a method that combines distant supervision with self-training, which I called (double-check approach).

First of all, I compare the quality of ATSAD corpus - which is based on distant supervision labelling - with the same previous corpora (LABR, ASTD, Shami-Senit) in addition to a 40K tweet dataset [106]. The 40k tweets corpus is a dialectal Arabic data set that includes 40,000 tweets classified into positive or negative and mostly from Egyptian dialects. The corpus has been manually annotated and went through a hard pre-processing phase. Given that ATSAD is a binary dataset, then I ignore the third class (Neutral or mix) from all the corpora and implement it as a binary classification task.

The process of improving and enhancing the ATSAD corpus involved a lot of experiments, but before I got into those, I firstly ask whether a sentiment analysis model trained with any of the aforementioned corpora was able to predict the label of ATSAD. For this reason, I conduct an experiment using a simple baseline of a word-gram language model and apply a linearSV classifier. The model is trained on all the datasets separately and then applied on ATSAD to predict the class for every instance. None of the models succeeded with the accuracy not exceeding 60%. Either the domain differences, dialect differences, or both of them, are the reason behind the failure of the prediction model. Generally speaking, a standalone Sentiment Analysis model seems to be needed for ATSAD, which can be also used to improve the corpus.

During all the experiments, I use the same test sets to make the comparison fair. In addition, I use our proposed Sentiment Analysis model, which consists of a combination of word gram language features and character gram language features with and without the consideration of boundaries. The model is discussed in *Study 4*.

ATSAD is a 36k tweets corpus which is split into two subsets. The first is a gold standard of 8K (7,616 tweets and 400 mixed) that are manually annotated, while the rest of 29K (29,252) is an emojis labelling subset (weakly labelling/distant supervision) classified into positive and negative polarities.

Both the baseline and our proposed Sentiment Analysis model are trained on the Gold standard and they achieve an accuracy of 71% and 79% respectively. Based on our work of adapting a trained model for other dialects, see *Study 4*, I exploit the same technique but instead of having two different dialects I have two different labelling or annotation methods. Therefore, I exploit the method by training the models using the Gold standard and letting it predicate the emoji labelling subset (29K). The accuracy was 63% and 76% for the baseline and the Sentiment Analysis model respectively. Here I can say that I got nearly the same result as the agreement between the human annotation and the distant supervision annotation of 76%.

The question is whether I can improve the emojis labelling subset which constitutes most of the ATSAD corpus? In that way, I exploit the gold standard set to enhance and improve the corpus by employing the self-training method. The self-training methods are mostly well known and commonly used in combination with semi-supervised learning, where part of that data is unlabelled[155, 156]. The model is trained by the gold standard train set and incrementally retrained by adding only the most confidently labelled instance as new training data. These new instances are the instances that the emojis labelling matches the prediction label.

Figure 5.3 shows the double-check methods where self-training is em-

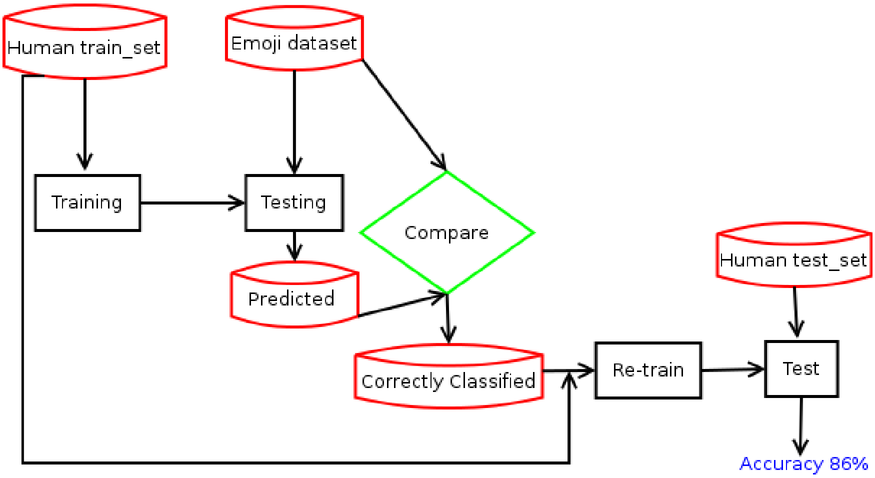


Figure 5.3: Self training (double-check) approach applied on ATSAD

ployed. By employing self-training methods, I end up with 28K tweets (6K from the gold standard and 22 from emojis subset) and a strongly supervised labelling set. This 28K set is used to build the model again and I test the performance on the test set from the gold standard. The model accuracy is the highest, at 77%, for the baseline and 86% for the proposed model. These results mean that increasing the size of the data as well as the quality has a positive impact on the model.

To ensure the reliability of our double-check self-training model, I do an experiment without the check of matching; so when I retrain the model by using the 6k from the gold standard and all the 29k tweets and their labels as predicted from the model, the accuracy drops to 70% and 81% for both the baseline and the proposed model.

General speaking, the weak supervision, in addition to a little human supervision, as well as the self-training techniques, all improve the performance of the model and the final quality of the ATSAD. Also, I have a look at how people use emojis and how these emojis can mislead the prediction and confuse the model. Many emojis are considered tricky where they have a positive expression; however, they are used for negativity too, such as the black smiley face. Moreover, emojis like a smiley face with tears are used for sarcasm, which, unfortunately, badly affects the model.

## 5.4 Feature Engineering or Pre-trained Language models for DI and SA?

In the previous studies, I have tried both traditional ML approaches as well as DL models. The performance of the model and the decision of which model to chose and apply on an NLP tasks depends on many factors, such as: the size of the dataset, the source of the data, the data quality, the balancing between the classes, if the corpus contains MSA or Multi-dialectal Arabic and the number of classes. In this section I compare different approaches; moreover, I use the pre-trained language models that have been recently introduced. To measure and compare among the methods, I apply two NLP tasks: Dialect Identification and Sentiment Analysis. For every task I use three corpora; PADIC, MADAR-6 and SHAMI (SDC) for DI, while for SA I use ATSAD, 40K and ASTD. The choice of the dataset was not random; I chose different datasets with various sizes, sources, balancing, quality and different numbers of classes.

### 5.4.1 BERT for Dialectal Arabic

Google AI Language researchers have recently implemented the Bidirectional Encoder Representations from Transformers (BERT)[157]. The transformer is the main component of BERT; it is an encoder-decoder attention mechanism that has been built to learn the contextual relations between sequences of words in any text and to generate a language model [157]. BERT takes a sequence of words (tokens) as an input layer, converts them to embedded vectors and then goes through the encoder transformer to generate the sequence vectors. To fine-tune BERT for classification tasks, a fully connected classification layer with a soft-max activation function is built on top of the output vectors. As our works concern dialectal Arabic, I conduct some experiments to check the performance of BERT on Dialect Identification and Sentiment analysis; thus I use three different BERT models that support Arabic.

1. Multilingual-BERT:[3] This is the multi-lingual version of BERT, where it contains the top 100 languages with the largest Wikipedia content, including Arabic. I use this multi-lingual BERT where different languages were modelling, so it might make sense for Dialect Identification tasks where the target is to recognize different varieties.

2. Arabic-BERT [82]: Arabic-BERT has been built with 8,2B words from the OSCAR data [158] and the recent data dump from Wikipedia. As

[3] https://github.com/google-research/bert/blob/master/multilingual.md

some of the corpora in the experiments contain MSA, it is good to have a BERT model that has been built for MSA with some dialectal words.

3. AraBERT-Twitter-base [89]: There are two versions of AraBERT, v1 and v2, where they differ in terms of segmentation techniques. AraBERT-Twitter-base is the dialectal version of AraBERTv2. It contains 60M Multi-Dialect Tweets in addition to 200M from the AraBERTv2-base. I recommend this AraBERT models as it works on dialectal Arabic and there is a Twitter version of the model where it contains both MSA and data from twitter. I call it Twitter-Bert through the experiments.

I use the same performance measurements through all the experiments: Accuracy, F-score and Mathews correlation coefficient (MCC). All the experiments' details, the parameters and the settings are mentioned in detail in *Study 7*.

### 5.4.2 Deep Learning LSTM network

To make the comparison as simple as possible, I build an LSTM network and apply it to the corpora with the two tasks. The first layer is a pre-trained Arabic word embeddings (AraVec)[153], followed by an LSTM layer with a dropout of 0.25%. The last two layers are fully connected Dense with 30 nodes and output labels nodes respectively.

### 5.4.3 Feature-Based Classification for Dialectal Arabic

I compare the BERT models, the LSTM deep learning models as well as the traditional Machine learning models on Dialectal Arabic tasks. I build an SVM machine learning model based on *Study 4*, where it applies Word-gram features and character-gram features with and without boundary consideration. I also use the same weights adjustments.

As a consequence, I further investigate the way that the feature engineering based methods affect the classification models. Thus, I place a fully connected classification layer on the top of the language model rather than using a traditional machine learning algorithm such as SVM or NB.

I run all the experiments and write down the results for easy comparison in Table 5.14 for Dialect Identification, and in Table 5.15 for Sentiment Analysis.[4].

The BERT experiments show acceptable and applicable results in terms of accuracy, MCC and F-scores. Twitter AraBERT model outperforms all

[4] All the values in Table 5.14 and Table5.15 are multiplied by 100

| | PADIC | | | SHAMI | | | MADAR-6 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc | MCC | F | Acc | MCC | F | Acc | MCC | F |
| Multilingual-BERT | 72 | 67 | 72 | 88 | 81 | 83 | 89 | 87 | 89 |
| Arabic-BERT | 71 | 66 | 72 | 87 | 78 | 81 | 80 | 76 | 80 |
| Twitter-BERT | **77** | **73** | **77** | **91** | **86** | **86** | **91** | **90** | **91** |
| LSTM | 17 | 0 | 14 | 57 | 0.4 | 18 | 17 | 0 | 29 |
| TFIDF + SVM | 72 | 66 | 72 | 90 | 84 | 86 | 89 | 87 | 89 |
| TFIDF+ Dense | 73 | 68 | 74 | 57 | 0 | 50 | 89 | 87 | 89 |

Table 5.14: Performance measurements for all the experiments on Dialect Identification.

| | ATSAD | | | 40K tweets | | | ASTD | | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc | MCC | F | Acc | MCC | F | Acc | MCC | F |
| Multilingual-BERT | 80 | 60 | 80 | 83 | 66 | 83 | 81 | 51 | 75 |
| Arabic-BERT | 93 | 87 | 93 | 83 | 66 | 83 | 84 | 63 | 81 |
| Twitter-BERT | **97** | **94** | **97** | **91** | **82** | **91** | **88** | **74** | **87** |
| LSTM | 52 | 0 | 34 | 49 | 0 | 33 | 69 | 0 | 41 |
| TFIDF + SVM | 96 | 92 | 96 | 84 | 67 | 84 | 80 | 45 | 71 |
| TFIDF+ Dense | 96 | 91 | 95 | 82 | 63 | 82 | 77 | 49 | 73 |

Table 5.15: Performance measurements for all the experiments on Sentiment Analysis

the mentioned models in both tasks. Regarding the Sentiment Analysis, Twitter-AraBERT outperforms the state of the art results for ASTD and 40K corpora. Employing BERT as a pre-trained language model for dialectal Arabic tasks proves that the model is able to get a high accuracy while saving time and resources, and with a small amount of data compared to deep learning networks like LSTM, CNN and others.

From the results tables, it is quite obvious that the LSTM model is the worst among all the models, with a huge evaluation gap between that and the other models. In order to be able to build an accurate deep learning model, I need a huge amount of data, in addition to a complicated network architecture, to solve many problems that related to Dialectal Arabic such as the OOV problem when using pre-trained word embeddings like AraVec.

Even though BERT models achieve high accuracy, traditional ML models are still able to compete and get reasonably high results within a very short time and with a small amount of resources. It is worth mentioning that a feature-based Machine Learning model which employs the power of character n-grams and word n-grams language models, can compete with the DL models, as well as outperforming them in some cases. The Twitter-AraBERT model and the ML models are close to each other in terms of accuracy, especially for SHAMI, a non-parallel and unbalanced corpus.

Figure 5.4 and Figure 5.5 plot the accuracy for all the models in a graph bar. They show how BERT, in addition to ML models, are competing and achieving higher results compared to DL models. As Multi-lingual BERT contains a lot of languages next to Arabic, its strength is bigger than that of Arabic-BERT in the Dialect Identification task where the goal is to recognize a dialect among multiple dialect labels. However, in Sentiment Analysis, using Multi-lingual BERT is not as powerful as other Arabic-BERT models, where the latter contain many dialectal words that help the sentiment analysis model to handle the classification goal. For corpora such as ATSAD, which have a reasonable amount of documents and are automatically collected, pre-processed and labelled, I find that traditional ML approaches are working well. In general, models that use BERT are considered a reasonable and straightforward solution for applying DL on different corpora. BERT is good in fixing the OOV words that word-embedding suffers from, in addition to the fact, that BERT language model depends on the context of the words.

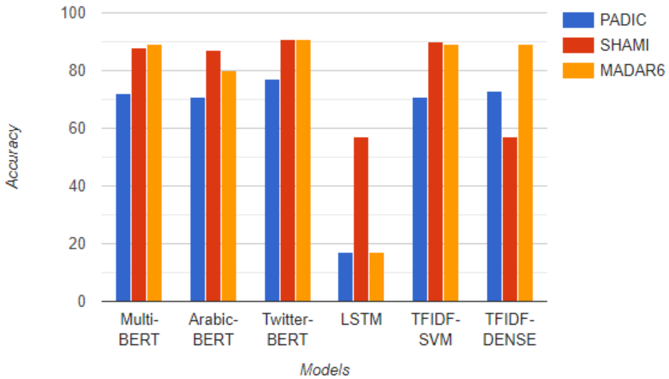


Figure 5.4: Accuracy of different Dialect Identification models

From all the experiments, I find that using a pre-trained language model for dialectal Arabic can be seen as a robust way to build classification models that are able to predict and classify in a high performance way. All the BERT models achieve reasonable accuracy and F-scores. For Dialect Identification and Sentiment Analysis tasks, the Twitter-AraBERT is the best model out of all the proposed models in the study. Implementing the pre-

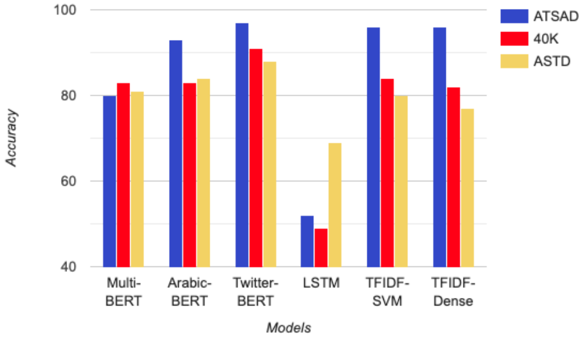


Figure 5.5: Accuracy of different Sentiment Analysis models

trained language models on the top of fully connected layers can save time, resources and even achieve high performance for small and un-balanced corpora, which is the case of the most dialectal Arabic resources. LSTM deep learning networks with pre-trained Arabic word-embeddings AraVec, fails to compete with any other models. A deep learning network can be a good solution for a huge amount of data and complicated structure. AraVec word-embeddings suffers from the OOV problem that decreases the performance of the prediction model, as well as that the word-embedding does not depend on the context of the words, which misleads the results. Feature engineering based methods prove that they are able to compete with pre-trained language models by leveraging the N-gram language models and a very simple machine learning algorithm like SVM.

Many factors play a role in the decision of choosing the best model to be applied on an NLP tasks: the NLP task to be solved, dataset size, the sources and the quality of the data, the data balance, the type of annotation, if the corpus contains MSA or Multi-dialectal Arabic data, and the number of classes, among other factors.

# 6 Conclusion

This thesis aims to investigate a number of important topics in the field of Dialectal Arabic Natural Language Processing. It is shedding light on the written form of Arabic dialects and the differences among them. The thesis addressed three main tasks, which are Dialect Identification and Sentiment Analysis for Arabic dialects and a lexical distance linguistic study. For every task, I investigate the Levantine as a case study, in addition to the Arabic dialects in general.

This thesis has made a number of contributions in different directions. More specifically:

- Resource contributions: The first Levantine corpus (SHAMI) was introduced. This includes data from four dialects of the Levant: Palestinian, Syrian, Lebanese and Jordanian. I then utilized SHAMI to build Shami-Senti, a Levantine sentiment analysis corpus. Both corpora can be used to investigate linguistic aspects of Levantine dialects and to train models for dialectal Arabic NLP tasks. Additionally, and for coarse-grained Sentiment Analysis, I proposed an Arabic Tweets SA Dataset (ATSAD) that includes a gold standard dataset which can be used for evaluation purpose.

- Linguistics contributions: I measured the lexical distance across different dialects and MSA. Such a study can help researchers to fine-tune and transfer knowledge between Arabic varieties, and further understand the relationship between them. I also examined the way MSA tools can be fine-tuned and adapted to process Arabic dialects. Finally, I analyzed several factors that affect the decision of choosing the best model when processing Arabic dialects and handling DANLP tasks. For this reason, I implemented various well-known approaches and used several dialectal datasets for the purpose of comparison.

- Model development contributions: For DI, I proposed various ML models by employing language N-gram models for Levantine DI. I also implemented an ensemble learning model that depends on feature engineering approaches for fine grained DI. Regarding SA, I explored both two-way and three-way sentiment classification by implementing several ML and DL models, as well as different N-gram combinations. I proposed a semi-supervised learning model and applied a distant supervision approach to help NLP researchers build reliable and larger datasets. Moreover, I examined the usage of the pre-trained language model BERT in both DI and SA by exploiting different corpora.

## 6.1 From Questions to Findings

### Dialect Identification

In the context of the Dialect Identification, I conducted Study 1 (page 99) and Study 2 (page 123). I addressed several questions or topics:

1. The unavailability of a Levantine corpus that includes data from the four dialects spoken in the Levant.

2. The feature combinations and an effective model using those combinations that can detect and classify Levantine dialects

3. Which model can be used for fine-grained dialect identification to detect and classify 26 Arabic varieties?

I hypothesise that Levantine dialects, in specifically, and Arabic dialects, in general, are easily distinguishable and detectable by DNNLP models. For this purpose I built the first annotated and classified Levantine SHAMI corpus SDC, where I collected the data both manually and automatically. As the dialects are distinguished by words, I implemented various N-gram language models on the level of words and characters. I also examined the performance of the model on a different number of classification categories.

For the fine-grained DI, the proposed system performed the best in a shared task[4]. I implemented an ensemble learning model to classify 26 Arabic varieties corresponding to 25 Arab cities and MSA. The model focuses on a number of features, such as word-gram, character-gram, as well as skip-gram features. I implemented feature union strategies to combine multi-features and I emphasized the weight of some features upon others.

Additionally, I classified the country of the dialects and used this predicted label as an input feature to the fine-grained model.

A general conclusion that can be drawn from the experiments in studies one and two is that the hypothesis that Arabic dialects are easy to distinguish is incorrect. While it is easy for Levantine speakers to identify the differences between Levantine dialects, the presence of dialects in a written form, where accent and the diacritics are missing, makes the task difficult and complex. Furthermore, the task becomes more complicated and more difficult as the number of dialects increases, going down to the level of cities and provinces. The absence of accents or diacritics in a written form is, thus, one of the biggest obstacles in DI, as the Arabic dialects share many words, but they differ in the way that people pronounce them. Given that classifying data into more categories is usually associated with increasing complexity as the categories increase, I also observed an increasing difficulty in DI the greater the number of dialects were classified. This is especially true when the sentences are so short that even an Arabic-speaking person cannot distinguish between them.

Another conclusion based on the research in DI presented in this thesis, is that the similarity of dialects does not depend on whether they are spoken in the same country. However, the similarity may be due to geographical distances sometimes. For example, I find that the Jerusalem dialect is closer to the dialect of Amman than the Gaza dialect, even though Jerusalem and Gaza belong to the same country. I also notice that some Syrian dialects in the east are closer to the Iraqi dialect than to the Damascus dialect.Additionally, most of the geographically close cities are similar in that their dialects are distinguished from those of the countryside, even when the villages belong to the city.

In general, my research on DI led me to a new hypothesis, i.e. the existence of a new form of dialects: the written forms, that can be similar or different from the characteristics of spoken dialects.

### Cross-dialects lexical distance

Since I study the characteristics of Arabic dialects and focus on written texts, I questioned the level of lexical similarity and divergence between MSA and dialects of Arabic. Therefore, I decided to conduct a cross-dialectal linguistic study on the extent to which these dialects are similar or different from MSA on the one hand, and between them on the other. At the time of Study 3 (page 135), to my knowledge, there was no single corpus that included data from all dialects in addition to MSA. Thus, I compiled more than one dataset, some of which are parallel or semantically comparable, and others containing close dialects. I used different algorithms to measure the distance between the available dialects, e.g. Jaccard Index Vector Space Model, Late Semantic Indexing, Hellinger Distance and Correlation Coefficient. These algorithms measure the distance between dialects in different ways. For example, the Jaccard Index method focuses on the amount of overlapped words between two dialects, while the Late Semantic Indexing method focuses on the semantic presentation of the concepts between dialects and the use of synonyms and polysemy.

Such a linguistic study is considered the first of its kind in the field of Arabic language processing.It showed how close some dialects are to each other, e.g. the similarity of the Jordanian dialect to the Palestinian one. It also presented new findings, such as the convergence of the Tunisian dialect with the MSA , and that the Levantine dialects, in particular the Palestinian ones, are the closest to MSA.

### Sentiment Analysis

In this thesis, we also looked at the task of Sentiment both for Levantine dialects, specifically, and dialectal Arabic, more generally. The main questions we asked were as follows:

1. Based on the findings from study3, what is the possibility of knowledge transfer from MSA to dialects?
2. Are there any available Levantine corpora to be used for the purpose of SA?
3. What is the ability of DL networks to process and handle dialectal SA and handle different kinds of dialectal datasets?
4. Which methods can one use to collect and build a corpus for dialectal Arabic SA, taking into consideration time and resource efficiency?

In Study 4 (page 155), I tested fine-tuning the MSA Sentiment analysis models to adapt to dialects. For this purpose, I utilized the Levantine corpus SDC to build the first Levantine Sentiment Analysis corpus (Shami-Senti) that contains the four Levantine dialects. I built various ML models for MSA using a corpus that contains the MSA and some dialectal words. After that, I used these models to classify Levantine sentiments. The results of the study showed a lack of efficiency and accuracy, and the models' performance was unacceptable. I added other dialects to the experiments, such as the Egyptian dialect, to test the possibility of adapting between dialects. I also reached the same result, which is the inefficiency of models that are trained in a specific language and used for processing and identifying other languages.

To follow the recent developments in the field of NLP and Machine Learning, I conducted Study 5 (page 175). There, I built SA models for each dialect so that the data used in the evaluation was of the same nature as the data I used in training the model. I introduced a complex Deep learning network based on the LSTM network and CNN, both preceded by an Arabic word embedding layer. The model outperformed the state-of-the-art for some corpora and got reasonable and high results for small datasets like Shami-Senti. However, the model suffered from being biased towards the majority class in unbalanced datasets.

Since Shami-Senti is a 2K sentence corpus, I decided to build an Arabic dialect corpus and annotate it for the purpose of sentiment analysis, where all Arab dialects are included and the corpus size is not small. This was done in Study 6 (page 191). Firstly, I collected data from Twitter based on emoji lexicons and then I employed distant supervision techniques to automatically annotate the corpus. Thus, the emojis have been utilized as weak labels. The total size of the corpus is 36K tweets. Based on this corpus, with the help of some annotators, I was able to create a gold standard SA corpus of 8K tweets. To evaluate the rest of the tweets and introduce a reliable corpus, I applied the self-training approach by training an SA model on the gold standard and then adding to the training set all the instances where the predicted label matches the emoji label. Then I retrained the model to improve the classification accuracy and the reliability of the corpus.

From studies 4,5 and 6, I conclude that each dialect needs its own models and applications. Furthermore, datasets of the same dialect should be used to train the models and then employ these models to handle and process the same dialects. I also showed that SA is a semantic task where models need to analyze the context and the feelings behind words. The fact that word-embeddings focus on lexical words, and not on the semantics or the use of words in context, weakens the network's ability to reach the desired results even with large datasets. For the purpose of resource creation, researchers can employ different approaches to collect and build big size resources that are reliable and usable. They can employ distant supervision as weakly labelling techniques in addition to self-training as semi-supervised learning to automatically annotate and evaluate the resources.

### Choosing the optimal model

Finally, in Study 7 (page 211), I addressed the following questions:

1. What are the models that can be used most efficiently for under-resourced dialects?

2. What are the factors that influence the choice of the model?

Throughout all the studies in this thesis, I developed a number of ML models based on feature engineering approaches and word-embeddings, Deep learning networks and, recently, models, which depend on the context of the words using pre-trained language models. In this context, I asked the following question: which is the most efficient of all these approaches? In Study 7, I used pre-trained language models (BERT) for DI ad SA, as these models represent words based on the context of the text. I applied different versions of BERT (Arabic-BERT, Twitter-BERT and multi-lingual BERT). I also implemented feature engineering approaches based on n-gram language models, as well as an LSTM deep learning network with word-embeddings (Ara-Vec). Even though Twitter-BERT achieved the best results in all the experiments, feature engineering approaches also got very competitive results in all settings.

Qualifying the results further, I conclude that there is no single perfect model for a specific NLP task. Several factors play a role in the choice of the best model. Some factors relate to the datasets themselves: corpus size, data balancing, the languages/dialects included in the dataset, whether the corpus is parallel or comparable, the sources where the data has been collected. Other factors relate to the implementation of the model: the training algorithm for the DL network, the choice of the optimizer, the loss function, among other parameters. The used techniques, e.g. the use of the Arabic word-embeddings or the Arabic version of BERT, also play a part in the performance of the proposed model. Finally, performance of the model differs from task to task depending on the nature of the task itself in each case.

## 6.2 Future Works

In this section, I will mention some unresolved issues and future work.

Some questions have not been fully answered through the thesis:

- building resources: the way researchers can build an automatic annotated fine-grained dialectal Arabic resource, which is huge in terms of size and contains only dialectal data without any MSA. How to deal with unbalanced datasets?

- studying linguistics of Arabic dialects: study the differences among the dialects in different aspects such as semantics and syntax in order to transfer the knowledge and fine-tune the tools among the Arabic varieties. Additionally, the effect of detecting dialects in solving the ANLP problems.

- developing NLP models: what if one combines pre-trained language models with feature engineering ML models to solve DI or SA?

Future work to be done includes the following:

- continuing the cross-dialectal linguistic study and addressing the differences among the dialects. Many dialects share the same words; however, these can have different or opposite semantic meanings. This difference among similar terms affects the performance of the models, because the meaning of the words depends on the contexts more than the lexical features.
- study the effect of Dialect Identification on Sentiment analysis. I want to connect both tasks into one application by detecting the dialect first, followed by classifying the sentiment polarity. Investigate the effect of DI on SA and see whether this could produce better performing models.
- Usually, people express their feelings and opinions in a mixed manner. It is possible that one sentence or post contains more than one opinion on the same topic (mixed polarity), or that people discussed different topics with mixed sentiments. So, I plan to investigate the topic of aspect-based sentiment analysis on dialectal Arabic
- Detecting and studying sarcasm in Sentiment analysis. Moreover, how the existence of sarcastic text could mislead the SA models and decrease their efficiency.
- I plan to investigate the topic of detecting and recognizing racism, harassment and cyberbullying in dialectal Arabic by using Levantine as a case study.

# Bibliography

[1] Mustafa Shah. *The Arabic Language*. Routledge, 2008.

[2] Kees Versteegh. *The Arabic Language*. Edinburgh University Press, 2014.

[3] Omar F. Zaidan and Chris Callison-Burch. “The Arabic Online Commentary Dataset: An Annotated Dataset of Informal Arabic with High Dialectal Content”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics. 2011, pp. 37–41.

[4] Houda Bouamor, Sabit Hassan, and Nizar Habash. “The MADAR Shared Task on Arabic Fine-Grained Dialect Identification”. In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Florence, Italy, 2019.

[5] Rehab M Duwairi et al. “Sentiment Analysis in Arabic Tweets”. In: *2014 5th International Conference on Information and Communication Systems (ICICS)*. IEEE. 2014, pp. 1–6.

[6] Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. “SAMAR: Subjectivity and Sentiment Analysis for Arabic Social Media”. In: *Computer Speech & Language* 28.1 (2014), pp. 20–37.

[7] Mahmoud Nabil, Mohamed Aly, and Amir Atiya. “ASTD: Arabic Sentiment Tweets Dataset”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2515–2519.

[8] Manal Abdullah et al. “Emotions Extraction from Arabic Tweets”. In: *International Journal of Computers and Applications* 42.7 (2020), pp. 661–675.

[9] Gilbert Badaro et al. “ArSEL: A Large Scale Arabic Sentiment and Emotion Lexicon”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Ed. by Hend Al-Khalifa et al. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. ISBN: 979-10-95546-25-2.

[10] Samuel Brody and Noemie Elhadad. “An Unsupervised Aspect-Sentiment Model for Online Reviews”. In: *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics.* 2010, pp. 804–812.

[11] Salima Harrat et al. “Cross-Dialectal Arabic Processing”. In: *International Conference on Intelligent Text Processing and Computational Linguistics.* Springer. 2015, pp. 620–632.

[12] Stephen Clark. “Vector Space Models of Lexical Meaning”. In: *Handbook of Contemporary Semantics – second edition.* Ed. by Shalom Lappin and Chris Fox. Wiley – Blackwell, 2015. Chap. 16, pp. 493–522.

[13] Ch Aswani Kumar, M Radvansky, and J Annapurna. “Analysis of a Vector Space Model, Latent Semantic Indexing and Formal Concept Analysis for Information Retrieval”. In: *Cybernetics and Information Technologies* 12.1 (2012), pp. 34–48.

[14] VıCtor GonzáLez-Castro, RocıO Alaiz-RodrıGuez, and Enrique Alegre. “Class Distribution Estimation Based on The Hellinger Distance”. In: *Information Sciences* 218 (2013), pp. 146–164.

[15] Anwar G Chejne. *The Arabic Language: Its Role in History.* U of Minnesota Press, 1968.

[16] Ahmad Al-Jallad. “The Arabic of the Islamic conquests: notes on phonology and morphology based on the Greek transcriptions from the first Islamic century”. In: *Bulletin of the School of Oriental and African Studies* 80.3 (2017), pp. 419–439.

[17] Ahmad Al-Jallad. “Pre-Islamic Arabic”. In: *Arabic and contact-induced change* 1 (2020), p. 37.

[18] Brian Bishop. “A History of The Arabic Language”. In: *Department of Linguistics, Brigham Young University* (1998).

[19] Charles A Ferguson. “Diglossia”. In: *word* 15.2 (1959), pp. 325–340.

[20] MJ Jabbari. "Diglossia in Arabic-a Comparative Study of The Modern Standard Arabic and Colloquial Egyptian Arabic". In: *Global Journal of Human Social Sciences* 12.8 (2012), pp. 23–46.

[21] Abderrahman Zouhir. "Language Situation and Conflict in Morocco". In: *Selected Proceedings of the 43rd Annual Conference on African Linguistics, ed. Olanike Ola Orie and Karen W. Sanders.* 2013, pp. 271–277.

[22] Nizar Y Habash. "Introduction to Arabic Natural Language Processing". In: *Synthesis Lectures on Human Language Technologies* 3.1 (2010), pp. 1–187.

[23] Clive Holes. *Gulf Arabic.* Routledge, 2003.

[24] Kees Versteegh. "Arabic in Europe: From Language of Science to Language of Minority". In: *Lingua e stile* 36.2 (2001), pp. 335–346.

[25] Ernest T Abdel-Massih. *An Introduction to Egyptian Arabic.* MPublishing, University of Michigan Library, 2011.

[26] Muhammad Amara. *Reem Bassiouney: Arabic Sociolinguistics.* 2010.

[27] Terence Frederick Mitchell. *Pronouncing Arabic.* Vol. 2. Oxford University Press, USA, 1990.

[28] Salima Harrat, Karima Meftouh, and Kamel Smaıli. "Maghrebi Arabic Dialect Processing: An Overview". In: *Journal of International Science and General Applications* 1 (2018).

[29] Basem Ibrahim Malawi Al-Raba'a. "Language Attitudes Toward The Rural and Urban Varieties in North Jordan". In: *Al-'Arabiyya* 49 (2016), pp. 67–89. ISSN: 08898731, 23754036. URL: `http://www.jstor.org/stable/26451376`.

[30] Bettina Leitner. "New Perspectives on the Urban–Rural Dichotomy and Dialect Contact in the Arabic glt Dialects in Iraq and South-West Iran". In: *Languages* 6.4 (2021), p. 198.

[31] Mustafa Jarrar et al. "Building a Corpus for Palestinian Arabic: A Preliminary Study". In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP).* 2014, pp. 18–27.

[32] Muhammad Hasan Bakalla et al. *Arabic Culture Through its Language and Literature.* Routledge, 1984.

[33] Elinor Saiegh-Haddad and Roni Henkin-Roitfarb. "The Structure of Arabic Language and Orthography". In: *Handbook of Arabic literacy.* Springer, 2014, pp. 3–28.

[34] Kareem Darwish. “Named Entity Recognition using Cross-Lingual Resources: Arabic as an example”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2013, pp. 1558–1567.

[35] Siamak Rezaie. “Tokenizing an Arabic Script Language”. In: *Arabic language processing: Status and prospects, ACL/EACL* (2001).

[36] Jihad M Hamdan and Shehdeh Fareh. “Acronyms in English and Arabic”. In: *Available at SSRN 3061196* (2003).

[37] Zikrawahyuni Maiza et al. “Orthography and Pronunciation in Arabic and English; A Contrastive Analysis”. In: *BICED 2020: Proceedings of the 2nd EAI Bukittinggi International Conference on Education, BICED 2020, 14 September, 2020, Bukititinggi, West Sumatera, Indonesia*. European Alliance for Innovation. 2021, p. 183.

[38] Ali Ahmed Sabry Farghaly. *Arabic Computational Linguistics*. CSLI Publications, Center for the Study of Language and Information, 2010.

[39] Maryse Maroun. “Diacritics and the Resolution of Ambiguity in Reading Arabic”. PhD thesis. University of Essex, 2018.

[40] Kenneth R Beesley. “Arabic Morphology Using Only Finite-State Operations”. In: *Computational Approaches to Semitic Languages*. 1998.

[41] Nizar Habash, Owen Rambow, and George Anton Kiraz. “Morphological Analysis and Generation for Arabic Dialects”. In: *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*. 2005, pp. 17–24.

[42] Clive Holes. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown University Press, 2004.

[43] Elabbas Benmamoun. “Arabic Morphology: The Central Role of The Imperfective”. In: *Lingua* 108.2-3 (1999), pp. 175–201.

[44] Violetta Cavalli-Sforza, Abdelhadi Soudi, and Teruko Mitamura. “Arabic Morphology Generation using a Concatenative Strategy”. In: *1st Meeting of the North American Chapter of the Association for Computational Linguistics*. 2000.

[45] Ghadah Alwakid, Taha Osman, and Thomas Hughes-Roberts. “Challenges in Sentiment Analysis for Arabic Social Networks”. In: *Procedia Computer Science* 117 (2017), pp. 89–100.

[46] Joseph E Aoun, Elabbas Benmamoun, and Lina Choueiri. *The Syntax of Arabic*. Cambridge University Press, 2009.

[47] Usama Soltan. "On Issues of Arabic Syntax: An Essay in Syntactic Argumentation". In: *Brill's Journal of Afroasiatic Languages and Linguistics* 3.1 (2011), pp. 236–280.

[48] Suzanne Mahmoud Bardeas. "The Syntax of the Arabic DP". PhD thesis. University of York, 2009.

[49] Kristen Brustad. *Spoken Arabic*. Georgetown University Press, 2000.

[50] Usama Soltan. "Standard Arabic Subject-Verb Agreement Asymmetry Revisited in an Agree-based Minimalist Syntax". In: *Agreement systems* 92 (2006), p. 239.

[51] Charles N Li and Sandra A Thompson. *Chinese*. Routledge, 2017.

[52] Derek Rogers and Luciana d'Arcangeli. "Italian". In: *Journal of the International Phonetic Association* 34.1 (2004), pp. 117–121.

[53] Slavomır Čéplö et al. "Mutual Intelligibility of Spoken Maltese, Libyan Arabic, and Tunisian Arabic Functionally Tested: A pilot Study". In: *Folia Linguistica* 50.2 (2016), pp. 583–628.

[54] Nizar Habash, Mona T Diab, and Owen Rambow. "Conventional Orthography for Dialectal Arabic." In: *LREC*. 2012, pp. 711–718.

[55] Arkadiusz Płonka. "Le Nationalisme Linguistique au Liban Autour de Saıd Aql et l'idée de langue libanaise dans la revue" Lebnaan" en nouvel alphabet". In: *Arabica* (2006), pp. 423–471.

[56] Niloofar Haeri. "Sociolinguistic Variation in Cairene Arabic: Palatalization and the Qaf in The Speech of Men and Women". In: (1991).

[57] Houda Bouamor, Nizar Habash, and Kemal Oflazer. "A Multidialectal Parallel Corpus of Arabic." In: *LREC*. 2014, pp. 1240–1245.

[58] Karima Meftouh et al. "Machine Translation Experiments on PADIC: A Parallel Arabic Dialect Corpus". In: *The 29th Pacific Asia conference on language, information and computation*. 2015.

[59] Kareem Darwish, Hassan Sajjad, and Hamdy Mubarak. "Verifiably Effective Arabic Dialect Identification." In: *EMNLP*. 2014, pp. 1465–1468.

[60] Ali Farghaly and Khaled Shaalan. "Arabic Natural Language Processing: Challenges and Solutions". In: *ACM Transactions on Asian Language Information Processing (TALIP)* 8.4 (2009), pp. 1–22.

[61] Khaled Shaalan and Hafsa Raza. "Arabic Named Entity Recognition from Diverse Text Types". In: *International Conference on Natural Language Processing*. Springer. 2008, pp. 440–451.

[62] Wajdi Zaghouani. "Critical Survey of the Freely Available Arabic Corpora". In: *arXiv preprint arXiv:1702.07835* (2017).

[63] Abdulhadi Shoufan and Sumaya Alameri. "Natural Language Processing for Dialectical Arabic: A Survey". In: *Proceedings of the second workshop on Arabic natural language processing.* 2015, pp. 36–48.

[64] Mona Diab et al. "COLABA: Arabic Dialect Annotation and Processing". In: *Lrec workshop on semitic language processing.* 2010, pp. 66–74.

[65] Rabih Zbib et al. "Machine Translation of Arabic Dialects". In: *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies.* Association for Computational Linguistics. 2012, pp. 49–59.

[66] Khalid Almeman and Mark Lee. "Automatic Building of Arabic Multi Dialect Text Corpora by Bootstrapping Dialect Words". In: *Communications, signal processing, and their applications (iccspa), 2013 1st international conference on.* IEEE. 2013, pp. 1–6.

[67] Ryan Cotterell and Chris Callison-Burch. "A Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic." In: *LREC.* 2014, pp. 241–245.

[68] Israa Alsarsour et al. "Dart: A Large Dataset of Dialectal Arabic Tweets". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* 2018.

[69] Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. "You Tweet What You Speak: A City-Level Dataset of Arabic Dialects". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* 2018.

[70] Houda Bouamor et al. "The MADAR Arabic Dialect Corpus and Lexicon". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* 2018.

[71] ElMehdi Boujou et al. "An Open Access NLP Dataset for Arabic Dialects: Data Collection, Labeling, and Model Construction". In: *arXiv preprint arXiv:2102.11000* (2021).

[72] Muhammad Abdul-Mageed et al. "NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task". In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop.* 2020, pp. 97–110.

[73] Muhammad Abdul-Mageed et al. "NADI 2021: The Second Nuanced Arabic Dialect Identification Shared Task". In: *arXiv preprint arXiv: 2103.08466* (2021).

[74] Mohammad Salameh, Houda Bouamor, and Nizar Habash. "Fine-Grained Arabic Dialect Identification". In: *Proceedings of the International Conference on Computational Linguistics (COLING).* Santa Fe, New Mexico, USA, 2018, pp. 1332–1344.

[75] Karima Meftouh et al. "The SMarT Classifier for Arabic Fine-Grained Dialect Identification". In: 2019.

[76] Ahmad Ragab et al. "Mawdoo3 AI at MADAR Shared Task: Arabic Fine-Grained Dialect Identification with Ensemble Learning". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 244–248. DOI: `10.18653/v1/W19-4630`. URL: `https://aclanthology.org/W19-4630`.

[77] Pruthwik Mishra and Vandan Mujadia. "Arabic Dialect Identification for Travel and Twitter Text". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019, pp. 234–238.

[78] Pavel Přibáň and Stephen Taylor. "Zcu-nlp at MADAR 2019: Recognizing Arabic Dialects". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019, pp. 208–213.

[79] Youssef Fares et al. "Arabic Dialect Identification with Deep Learning and Hybrid Frequency Based Features". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019, pp. 224–228.

[80] Sohaila Eltanbouly, May Bashendy, and Tamer Elsayed. "Simple But Not Naive: Fine-Grained Arabic Dialect Identification Using Only N-Grams". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019, pp. 214–218.

[81] Bashar Talafha et al. "Multi-Dialect Arabic BERT for Country-Level Dialect Identification". In: *arXiv preprint arXiv:2007.05612* (2020).

[82] Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. "Kuisail at Semeval-2020 task 12: BERT-CNN for Offensive Speech Identification in Social Media". In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. 2020, pp. 2054–2059.

[83] Abdellah El Mekki et al. "Weighted Combination of BERT and N-GRAM Features for Nuanced Arabic Dialect Identification". In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop*. 2020, pp. 268–274.

[84] Kamel Gaanoun and Imade Benelallam. "Arabic Dialect Identification: An Arabic-BERT Model with Data Augmentation and Ensembling Strategy". In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop*. 2020, pp. 275–281.

[85] Nitin Nikamanth Appiah Balaji and B Bharathi. "Semi-Supervised Fine-grained Approach for Arabic Dialect Detection Task". In: *Proceedings of the Fifth Arabic Natural Language Processing Workshop.* 2020, pp. 257–261.

[86] Ahmad Beltagy, Abdelrahman Wael, and Omar ElSherief. "Arabic Dialect Identification using BERT-Based Domain Adaptation". In: *arXiv preprint arXiv:2011.06977* (2020).

[87] Muhammad Abdul-Mageed et al. "Toward Micro-Dialect Identification in Diaglossic and Code-Switched Environments". In: *arXiv preprint arXiv:2010.04900* (2020).

[88] Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. "ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic". In: *arXiv preprint arXiv:2101.01785* (2020).

[89] Wissam Antoun, Fady Baly, and Hazem Hajj. "Arabert: Transformer-based Model for Arabic Language Understanding". In: *arXiv preprint arXiv:2003.00104* (2020).

[90] Wissam Antoun, Fady Baly, and Hazem Hajj. "Araelectra: Pre-training Text Discriminators for Arabic Language Understanding". In: *arXiv preprint arXiv:2012.15516* (2020).

[91] Badr AlKhamissi et al. "Adapting MARBERT For Improved Arabic Dialect Identification: Submission to the NADI 2021 Shared Task". In: *arXiv preprint arXiv:2103.01065* (2021).

[92] Neil Houlsby et al. "Parameter-Efficient Transfer Learning for NLP". In: *International Conference on Machine Learning.* PMLR. 2019, pp. 2790–2799.

[93] Abdellah El Mekki et al. "Bert-Based Multi-task Model for Country and Province Level MSA and Dialectal Arabic Identification". In: *Proceedings of the Sixth Arabic Natural Language Processing Workshop.* 2021, pp. 271–275.

[94] Anshul Wadhawan. "Dialect Identification in Nuanced Arabic Tweets Using Farasa Segmentation and AraBERT". In: *arXiv preprint arXiv: 2102.09749* (2021).

[95] Gilbert Badaro et al. "A Large Scale Arabic Sentiment Lexicon for Arabic Opinion Mining". In: *Proceedings of the EMNLP 2014 workshop on Arabic Natural Language Processing (ANLP).* 2014, pp. 165–173.

[96] Muhammad Abdul-Mageed and Mona Diab. "Toward Building a Large-Scale Arabic Sentiment Lexicon". In: *Proceedings of the 6th international global WordNet conference.* 2012, pp. 18–22.

[97] Ahmed Oussous, Ayoub Ait Lahcen, and Samir Belfkih. “Improving Sentiment Analysis of Moroccan Tweets Using Ensemble Learning”. In: *International Conference on Big Data, Cloud and Applications.* Springer. 2018, pp. 91–104.

[98] Muhammad Abdul-Mageed, Mona T Diab, and Mohammed Korayem. “Subjectivity and Sentiment Analysis of Modern Standard Arabic”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2.* Association for Computational Linguistics. 2011, pp. 587–591.

[99] Salima Medhaffar et al. “Sentiment Analysis of Tunisian Dialects: Linguistic Resources and Experiments”. In: *Proceedings of the third Arabic natural language processing workshop.* 2017, pp. 55–61.

[100] Nora Al-Twairesh et al. “Sentiment Analysis of Arabic Tweets: Feature Engineering and A Hybrid Approach”. In: *CoRR* abs/1805.08533 (2018).

[101] Rizkallah, Sandra and Atiya, Amir and ElDin Mahgoub, Hossam and Heragy, Momen", editor="Hassanien, Aboul Ella and Tolba, Mohamed F. and Elhoseny, Mohamed and Mostafa, Mohamed. “Dialect Versus MSA Sentiment Analysis”. In: *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018).* Cham: Springer International Publishing, 2018, pp. 605–613.

[102] Ramy Baly et al. “A Characterization Study of Arabic Twitter Data with a Benchmarking for State-of-the-art Opinion Mining Models”. In: *Proceedings of the third Arabic natural language processing workshop.* 2017, pp. 110–118.

[103] Ramy Baly et al. “Comparative Evaluation of Sentiment Analysis Methods Across Arabic Dialects”. In: *Procedia Computer Science* 117 (2017), pp. 266–273.

[104] Hamed Al-Rubaiee, Renxi Qiu, and Dayou Li. “Identifying Mubasher Software Products Through Sentiment Analysis of Arabic Tweets”. In: *2016 International Conference on Industrial Informatics and Computer Systems (CIICS).* IEEE. 2016, pp. 1–6.

[105] Ahmed Mourad and Kareem Darwish. “Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs”. In: *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis.* 2013, pp. 55–64.

[106] Ammar Mohammed and Rania Kora. “Deep Learning Approaches for Arabic Sentiment Analysis”. In: *Social Network Analysis and Mining* 9.1 (2019), p. 52.

[107] Mohamed Aly and Amir Atiya. “LABR: A Large Scale Arabic Book Reviews Dataset”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2013, pp. 494–498.

[108] Salam Khalifa et al. “A large scale corpus of Gulf Arabic”. In: *arXiv preprint arXiv:1609.02960* (2016).

[109] Motaz Saad and Basem O Alijla. “Wikidocsaligner: An Off-the-shelf Wikipedia Documents Alignment Tool”. In: *2017 Palestinian International Conference on Information and Communication Technology (PICICT)*. IEEE. 2017, pp. 34–39.

[110] Nora Al-Twairesh et al. “Suar: Towards Building a Corpus for The Saudi Dialect”. In: *Procedia computer science* 142 (2018), pp. 72–82.

[111] Toshiyuki Takezawa et al. “Multilingual Spoken Language Corpus Development For Communication Research”. In: *International Journal of Computational Linguistics & Chinese Language Processing, Volume 12, Number 3, September 2007: Special Issue on Invited Papers from ISCSLP 2006*. 2007, pp. 303–324.

[112] Imane Guellil et al. “Une Approche Fondée Sur Les Lexiques D’analyse de Sentiments du Dialecte algérien”. In: (2017).

[113] Imane Guellil et al. “Sentialg: Automated Corpus Annotation for Algerian Sentiment Analysis”. In: *International Conference on Brain Inspired Cognitive Systems*. Springer. 2018, pp. 557–567.

[114] Nora Al-Twairesh et al. “Arasenti-tweet: A Corpus for Arabic Sentiment Analysis of Saudi Tweets”. In: *Procedia Computer Science* 117 (2017), pp. 63–72.

[115] Jalal Omer Atoum and Mais Nouman. “Sentiment Analysis of Arabic Jordanian Dialect Tweets”. In: *(IJACSA) International Journal of Advanced Computer Science and Applications* 10 (2019), pp. 256–262.

[116] Ramy Baly et al. “ArSentD-LEV: A Multi-Topic Corpus for Target-Based Sentiment Analysis in Arabic Levantine Tweets”. In: *arXiv preprint arXiv:1906.01830* (2019).

[117] Imane Guellil, Marcelo Mendoza, and Faical Azouaou. “Arabic Dialect Sentiment Analysis with ZERO Effort. Case study: Algerian Dialect”. In: *Inteligencia Artificial* 23.65 (2020), pp. 124–135.

[118] Ibrahim Abu Farha and Walid Magdy. "A Comparative Study of Effective Approaches for Arabic Sentiment Analysis". In: *Information Processing & Management* 58.2 (2021), p. 102438.

[119] Sara Rosenthal, Noura Farra, and Preslav Nakov. "SemEval-2017 Task 4: Sentiment Analysis in Twitter". In: *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*. 2017, pp. 502–518.

[120] AbdelRahim A Elmadany, Hamdy Mubarak, and Walid Magdy. "An Arabic Speech-Act and Sentiment Corpus of Tweets". In: *The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools*. European Language Resources Association (ELRA). 2018.

[121] Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. "Overview of the WANLP 2021 Shared Task on Sarcasm and Sentiment Detection in Arabic". In: *Proceedings of the sixth Arabic natural language processing workshop*. 2021, pp. 296–305.

[122] Ibrahim Abu Farha and Walid Magdy. "From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset". English. In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*. Marseille, France: European Language Resource Association, May 2020, pp. 32–39. ISBN: 979-10-95546-51-1. URL: `https://aclanthology.org/2020.osact-1.5`.

[123] Ines Abbes et al. "DAICT: A Dialectal Arabic Irony Corpus Extracted from Twitter". In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 6265–6271. ISBN: 979-10-95546-34-4. URL: `https://aclanthology.org/2020.lrec-1.768`.

[124] Abdelkader El Mahdaouy et al. "Deep Multi-Task Model for Sarcasm Detection and Sentiment Analysis in Arabic Language". In: *arXiv preprint arXiv:2106.12488* (2021).

[125] Bingyan Song et al. "DeepBlueAI at WANLP-EACL2021 task 2: A Deep Ensemble-based Method for Sarcasm and Sentiment Detection in Arabic". In: *Proceedings of the Sixth Arabic Natural Language Processing Workshop*. 2021, pp. 390–394.

[126] Mohammed Rushdi-Saleh et al. "OCA: Opinion Corpus for Arabic". In: *Journal of the American Society for Information Science and Technology* 62.10 (2011), pp. 2045–2054.

[127] Hady ElSahar and Samhaa R El-Beltagy. “Building Large Arabic Multi-Domain Resources for Sentiment Analysis”. In: *International conference on intelligent text processing and computational linguistics*. Springer. 2015, pp. 23–34.

[128] Ashraf Elnagar and Omar Einea. “BRAD 1.0: Book Reviews in Arabic Dataset”. In: *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE. 2016, pp. 1–8.

[129] Ashraf Elnagar, Yasmin S Khalifa, and Anas Einea. “Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications”. In: *Intelligent Natural Language Processing: Trends and Applications*. Springer, 2018, pp. 35–52.

[130] Mohab Youssef and Samhaa R El-Beltagy. “MoArLex: An Arabic Sentiment Lexicon Built Through Automatic Lexicon Expansion”. In: *Procedia computer science* 142 (2018), pp. 94–103.

[131] Hady ElSahar and Samhaa R El-Beltagy. “A Fully Automated Approach for Arabic Slang Lexicon Extraction from Microblogs”. In: *International conference on intelligent text processing and computational linguistics*. Springer. 2014, pp. 79–91.

[132] Jean Carletta. “Assessing Agreement on Classification Tasks: the Kappa Statistic”. In: *Computational Linguistics* 2.22 (1996), pp. 249–254.

[133] Petra Kralj Novak et al. “Sentiment of Emojis”. In: *PLoS ONE* 10.12 (2015), e0144296. URL: `http://dx.doi.org/10.1371/journal.pone.0144296`.

[134] Limin Yao, Sebastian Riedel, and Andrew McCallum. “Collective Cross-Document Relation Extraction without Labelled Data”. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2010, pp. 1013–1023.

[135] Raphael Hoffmann et al. “Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics. 2011, pp. 541–550.

[136] Omar F Zaidan and Chris Callison-Burch. “Arabic Dialect Identification”. In: *Computational Linguistics* 40.1 (2014), pp. 171–202.

[137] Marco Lui and Timothy Baldwin. “langid.py: An Off-the-shelf Language Identification Tool”. In: *Proceedings of the ACL 2012 system demonstrations*. Association for Computational Linguistics. 2012, pp. 25–30.

[138] Nakatani Shuyo. “Language Detection Library for Java”. In: *Retrieved Jul* 7 (2010), p. 2016.

[139] William B Cavnar, John M Trenkle, et al. “N-gram-based Text Categorization”. In: *Ann Arbor MI* 48113.2 (1994), pp. 161–175.

[140] Michael McCandless. *Accuracy and Performance of Google's Compact Language Detector*. 2011.

[141] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830.

[142] Houda Bouamor, Sabit Hassan, and Nizar Habash. “The MADAR Shared Task on Arabic Fine-Grained Dialect Identification”. In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19)*. Florence, Italy, 2019.

[143] Ray R Larson. “Introduction to Information Retrieval”. In: *Journal of the American Society for Information Science and Technology* 61.4 (2010), pp. 852–853.

[144] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent Dirichlet Allocation”. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[145] Jiashi Feng et al. “Robust Logistic Regression and Classification”. In: *Advances in neural information processing systems*. 2014, pp. 253–261.

[146] Koby Crammer et al. “Online Passive-Aggressive Algorithms”. In: *Journal of Machine Learning Research* 7.Mar (2006), pp. 551–585.

[147] Suresh Kumar and Shivani Goel. “Enhancing Text Classification by Stochastic Optimization method and Support Vector Machine”. In: *International Journal of Computer Science and Information Technologies, 6 (4)* (2015), pp. 3742–3745.

[148] Tobias Günther and Lenz Furrer. “GU-MLT-LT: Sentiment Analysis of Short Messages using Linguistic Features and Stochastic Gradient Descent”. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Vol. 2. 2013, pp. 328–332.

[149] Shuo Xu, Yan Li, and Zheng Wang. “Bayesian Multinomial Naive Bayes Classifier to Text Classification”. In: *Advanced multimedia and ubiquitous engineering*. Springer, 2017, pp. 347–352.

[150] Hiroshi Shimodaira. “Text Classification Using Naive Bayes”. In: *Learning and Data Note* 7 (2014), pp. 1–9.

[151] Harris Drucker et al. “Support Vector Regression Machines”. In: *Advances in neural information processing systems*. 1997, pp. 155–161.

[152] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[153] Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. “AraVec: A Set of Arabic Word Embedding Models for use in Arabic NLP”. In: *Procedia Computer Science* 117 (2017), pp. 256–265.

[154] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[155] David Yarowsky. “Unsupervised Word Sense Disambiguation Rivaling Supervised Methods”. In: *33rd annual meeting of the association for computational linguistics*. 1995, pp. 189–196.

[156] Steven Abney. “Bootstrapping”. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 360–367.

[157] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[158] Pedro Javier Ortiz Suárez, Laurent Romary, and Benoıt Sagot. “A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages”. In: *arXiv preprint arXiv:2006.06202* (2020).

# 7

# Study 1: Towards a Levantine corpus (Shami) and Dialect Identification

*Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis and Simon Dobnik . "Shami: A corpus of levantine Arabic dialects." In:Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). 2018*

Modern Standard Arabic MSA is the official language that used in education and media across the Arab speaking world both in writing, as well as in formal speech. However, it does not constitute the native language for any Arab native speaker. Each country in the Arab World has it's own dialect and sub dialects. With the emergence of social media, the dialectal contents on the Internet are increased and the NLP tools that support MSA are not suited well to Arabic dialects due to the difference between the variants. In this paper, we propose Shami the first Levantine Dialects corpus SDC contains the four dialects which are spoken in (Palestine, Jordan, Lebanon and Syria). We apply preprocessing steps without affecting the semantic meaning of the dialectal data by specifying some rules before cleaning the noise. Dialect Identification is chosen as the task to evaluate SDC and comparing it with two baselines. several experiments are conducted on the SDC by various parameters based on n-grams model and Naive Bayes. SDC outperforms the baseline in terms of size, words and vocabularies in addition we achieved accuracy above 88% when classifying two and three dialects.
**Keywords:** Dialectal Arabic, Levantine Dialects corpus, Dialect Identification

## 7.1 Introduction

Arabic is one of the five most spoken languages in the world; it is spoken by more than 422 million native speakers and used by more than 1.5 billion Muslims [1]. The Arabic language is the Language of the Holy Quran (refer as Classical Arabic) and it is a liturgical language of 1,7 billion Muslims. Arabic Language is a textbook case of diglossia in which the written formal language differs substantially from the spoken vernacular. Modern standard Arabic (MSA), which is based heavily on Classical Arabic, is the official written language used in government affairs, news, broadcast media, books and education. MSA is the lingua franca amongst Arabic native speakers, but it does not have native speakers. The spoken language (collectively referred to as Dialectal Arabic) varies widely across the Arab world. The rapid proliferation of social media resulted in these dialects finding their way to written online social interactions. Dialects of Arabic differ widely among each other and depend highly on the geographic location and the socioeconomic conditions of the speakers.

Arabic Dialects are categorized according to geographic distribution in five dominant groups like:

- Egyptian: spoken in Egypt and some parts of Sudan
- Levantine: spoken in Palestine, Syria, Lebanon and Jordan
- Iraqi: spoken in Iraq…
- Gulf: spoken by Saudi Arabia, Kuwait, Qatar, Bahrain and UAE
- Maghrebi : spoken in Libya, Tunisia, Algeria, Morocco and western Sahara.

Each one of the classifications can be further fine grained to contain sub-dialects or varieties of the dialect; for example Palestinian Dialects which belongs to Levantine group can be also split to Urban, Rural and Bedouin dialects according to social and geographical locations. The situation is depicted in figure 7.1 [1]:

MSA and dialects share a considerable number of lexical, semantic, syntactic and morphological features. However, a number of differences with regard to these features also exist. For example, the word ايش *āyš* in Palestinian which means "what" and it comes from MSA شيء + اي *āy + šy* which means what thing. The word زرابي *zrāby* in Moroccan which means "carpets" and it is a synonym سجاد *sǧād* or بساط *bsāṭ* in MSA. On the other hand, the word وقية *wqyh*in Algerian means 3kg while it means 1/4 kg in

[1] http://www.unesco.org/new/en/unesco/events/prizes-and-celebrations/celebrations/international-days/world-arabic-language-day/

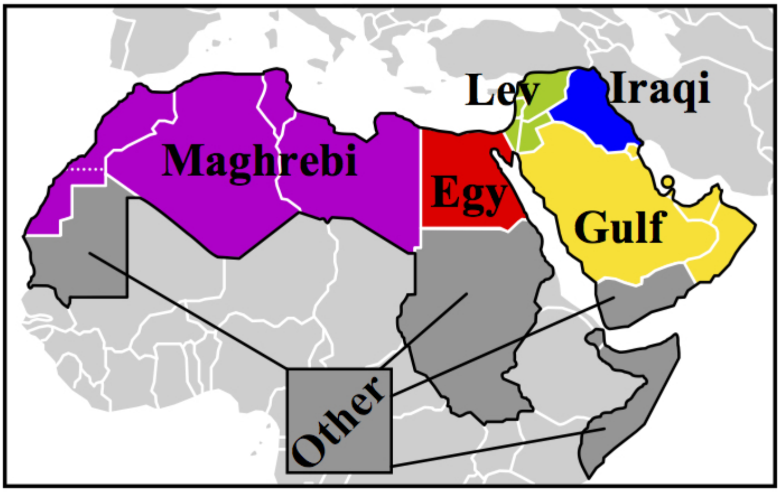


Figure 7.1: Maps of Middle east and North Africa. The regions where the Arabic dialects are spoken are labeled and colored

MSA and Levantine, so it is the same word but it has different meanings in Algerian, Levantine, and MSA.

Most of the Natural language processing (NLP) resources have been developed for MSA. Using these resources for Dialectal Arabic (DA) is considered a very challenging task given the differences between them. Most of the literature on DA that deals with Levantine dialects, treats these as one language despite the different sub-varieties mentioned and also the fact that dialect resources are small in size.

Levantine dialects looks similar for people who speaks different languages or dialects, but they are actually very different. Any Levantine native speaker can distinguish easily the other person's accent if they are Palestinian, Lebanese, Syrian or Jordanian, but sometimes it is difficult or impossible to distinguish between them by reading text without listening to their accent. There are many reasons why discrimination is difficult between Levantine texts:

- Lack of accent in writing. For example, a word like (كيفك *kyfk* /how are you) is used in all dialects, but the word pronunciation varies by country.
- The very similarity between the Palestinian and Jordanian dialects, except in some key words, which we can not find in all sentences.

- As a result of the political conditions experienced by the Levant region, especially Palestine, we find many Palestinians speaking with close accents to Syria and Lebanon, which makes it appear as an intermediate language between the Levantine dialects.
- The Phoneme of (prestige dialect)[2] in sociolinguistics that many of the people change their way of speaking to become more pleasant and clear. For example, many Bedouins or rural people speak like urban to clarify the conversation.

Table 7.1 shows some different examples for the little difference between the Levantine dialects which mostly appear in the pronunciation of the words.

| | |
|---|---|
| الأن أفضل شي أن الشباب كلهم خرجوا عالساحة *ālan afḍl šy an ālšbāb klhm ẖrǧwā ālsāḥh*<br>و تخلصوا من الصمت الذي كانوا يقبعون فيه. *wtẖlṣwā mn ālṣmt ālḏy kānwā yqbwn fyh.*<br>كل التحيات لهم *kl ālthyāt lhm* | MSA |
| Now the best thing is that young people have<br>appeared on the scene and got rid of the silence they were in.<br>All greetings to them | English |
| هلأ احلى شي انو الشباب كلهن ضهروا عالساحي *hla āḥlā šy ānw ālšbāb klhn ḍhrwā ālsāḥy*<br>واتخلصوا من صمتن يلي كانوا فيو. الله يوفئون *wātẖlṣwā mn ṣmtn yly kānwā fyw. āllh ywfywn* | Lebanon |
| هسا احلا شي انه الشباب هلأ كلهم ظهرو علساحه *hsā āḥlā šy ānh ālšbāb hla klhm ẓhrw lsāḥh*<br>واتخلصو من الصمت اللي كانو فيه هه الله محييهم النشامى *wātẖlṣw mn ālṣmt ālly kānw fyh hh āllh mḥyyhm ālnšāmā* | Jordanian |
| هلقيت احلى شي انه الشباب كلهم بينوا عالساحة *hlqyt āḥlā šy ānh ālšbāb klhm bynwā ālsāḥh*<br>واتخلصوا من هالصمت اللي كانوا فيه. الله يحييهم *wātẖlṣwā mn hālṣmt ālly kānwā fyh. āllh yḥyyhm* | Palestinian |
| هلأ أحلى شي انه الشباب كلياتهم بينوا عالساحي *hla aḥlā šy ānh ālšbāb klyāthm bynwā ālsāḥy*<br>واتخلصوا من صمتن اللي كانوا فيه. الله يوفؤن *wātẖlṣwā mn ṣmtn ālly kānwā fyh. āllh ywfwn* | Syrian |

Table 7.1: little differences between the same sentence among Levantine Dialects

In this paper we present the first Levantine dialect corpus, which distinguish among the dialects and to use for multi tasks on NLP and Computational Linguistics such as Language Identification, Sentiment analysis, speak recognition, POS tagging and others. Shami is consider the first Levantine dialects corpus, the name Shami refers to the Levant area in MSA which is called الشام *ālšām* hence a Levantine is called شامي *šāmy* in Arabic Language.

The paper is structured as follows: in section 2, we review some of the most prominent works dialectal Arabic NLP resources. In section 3, we present our corpus. In section 4, discuss Language Identification for Arabic Dialects, while in section 6 we present a preliminary evaluation using a language identification task. Finally, we conclude and discuss our findings in section 7.

[2] http://en.wikipedia.org/wiki/Varieties_of_Arabic

## 7.2 Related Works

In this section we review some works that related to Arabic dialects linguistics resources, like building a Corpus and employed it to some tasks in Natural language processing systems.

The Arabic On line Commentary (AOC) data set presented a monolingual dataset rich with dialectal content [2]. the AOC consists of 52 M words, 108 k words are annotated and 41% are dialectal.Reader commentary from on-line newspapers according to three dialects is extracted (Gulf, Egyptian, Levantine). In the case of Levantine, data are extracted only from Jordanian newspapers. For this reason, Amazon Mechanical Turk (AMT) is used to annotate the collected data using two types of annotations; firstly, the user should identify the extent of dialectal data like: (only MSA, MSA and dialectal, More Dialectal, only Dialectal) and then specify the dialect itself.

Zbib et. al; in [1] built a Parallel Levantine, Egyptian and English corpus by harvesting crowd sourcing. They draw from a large corpus of monolingual Arabic Text, which was collected by the Linguistic Data Consortium (LDC). Then they use the AMT to classify the Dialect sentences like AOC. They hire annotators to translate the Egyptian and Levantine sentences to English. Finally they build Levantine–English and Egyptian–English parallel corpora, consisting of 1.1M words and 380k words, respectively.

The authors of [3] presented PADIC (Parallel Arabic Dialect Corpus), is composed of 6400 sentences for each dialects. It includes five dialects: two from Algerian, one from Tunisia and two dialects from Levantine (Palestinian and Syrian). The Algerian dialects are extracted from recording conversations, movies and TV shows; then this Algerian dialects are translated to MSA and the others are translated manually from MSA corpus to the dialects. The Tunisian corpus was translated by 20 native speakers but there are just 2 speakers from Levantine one for each variant.

A preliminary work on a Corpus of Palestinian dialect was presented in [4] with 5836 Palestinian sentences with 43 K words. They collected data from TV episode scripts, social media and some forums and blogs and then annotated it manually. They exploited existing tools to speed up the annotation process like using MADAMIRA tool for morphological analysis of MSA and Egyptian [5]. They assumed that Palestinian and Egyptian share many orthographic and morphological features.

The COLABA project [6] designed a set of dialects queries per dialect to harvest large quantities of dialectal content from on-line resources like weblogs and forums. Data for Egyptian, Iraqi, Maghrebi and Levantine Arabic were collected making the assumption that Levantine dialects (Palestinian, Syrian, Jordanian, Lebanese) comprise a single Arabic dialect. The data cover three domains only: social issues, politics and religion. They use information retrieval tasks for measuring their ability to properly process dialectal Arabic content .

The five main dialects (Egyptian, Gulf, Levantine, Iraqi and Maghrebi) are also presented in [7]. The data are collected as in the case of AOC, i.e. from local newspaper commentary beside twitter tweets. In the same way they employed AMT for annotating their corpus. The most of Levantine data are Jordanian and they are nearly 6000 sentences.
Bouamor et. al; [8] presented a multi dialectal Arabic parallel corpus the data set consist of 2000 sentences in standard Arabic; Egyptian, Syrian, Palestinian, Tunisian, Jordanian and English. They selected the Egyptian part of Egyptian-English corpus presented by [1], and then asked annotators to translate the sentences to their own dialect. The used methodology relied on the assumption of the familiarity of Egyptian Arabic to most other dialect speakers.
Almeman and Lee [9] built Multi dialect text corpora by bootstrapping dialect words. They categorized the dialects text into four main categories regarding the geographical distribution, which are (Gulf, Levantine, Egyptian and North Africa), resulting in 14.5M, 10.4M, 13M and 10.1M tokens being obtained respectively and The total number of distinct types in all the corpora are 2M types. The key contribution of their proposed approach is language identification by creating words lists for every dialect category. In their methodology they firstly collect data from the web after that they asked six native speaker to extract uniquely dialect words and paraphrase for each dialects into lists, finally using these lists to collect links and download dialectal web pages from the web for every dialects .

## 7.3 Shami Dialects Corpus (SDC)

We set out to create our Shami Dialects Corpus (SDC) that concerned by the four Levantine dialects: Jordanian, Palestinian, Syrian and Lebanese which are spoken in Levant area. Some of the most important features of our SDC are:

- It is the First Levantine Dialects Corpus.
- Contains the largest volume of data from individual Levantine dialects compared to the previous corpora.
- Also it is the first corpus includes a Lebanese dialect significantly.
- It is not a parallel or crafted corpus, it contains real conversations as written without any changing or modification.
- It is not confined to a specific domain, it includes several domains according to the conversations of people such as politics, education, society, health care, house keeping and others.

Unlike previous work we started from scratch by collecting Levantine data depending on 2 approaches: automatic and manual approaches.

### 7.3.1 Data Collection

**Automatic Approaches**

To speed up the data collection process we used automatic methods and relied on the Twitter API streaming library (Tweepy)[3] to collect as many tweets as possible. Firstly, we checked twitter and selected some public figures who use dialect while tweeting from each country, then we used tweeter id [4] to convert all users accounts to their corresponding Twitter IDs. Secondly we used tweetpy streaming to collect tweets and replies from these IDs and each streaming was run until reach 9999 tweets each time. From the other side we used tweetpy to extract data according to the geographical location, so we specified the geographical co-ordinates for the 4 countries and run the code to retrieve all the tweets from these regions. All of the extracted data then stored in JSON files with the following information : (i) Tweeter ID, (ii) Data and Time, (iii) Tweet and replies, (iv) location and (v) number of likes, shares and replies. When we decided to stop streaming and collecting, we converted all JSON files to Text files which were cleaned and only tweets and replies were kept which are a collections of dialects, MSA, numbers and some Latin words.

**Manual Approaches**

As we need various domains and topics for our data, then we worked on collecting part of our SDC manually (the first two authors are Levantine native speakers). We harvested the web and specified some on-line dialectal blogs for public figures from the Levantine countries and also we extracted discussions and stories from some forums with various length for each sentence. Table 7.2 illustrates the number of sentences (documents) after automatic and manual extraction for each dialect.

### 7.3.2 Data Preprocessing

Dialects require special treatment to reprocess the text to be applicable for any task. Given that we collected colloquial data, some preprocessing steps are needed in order to get a reliable corpus that can be generally used in NLP applications. We employed the following processing steps:

[3] http://www.tweepy.org/

[4] https://tweeterid.com/

| | Automatic | Manual | Total |
|---|---|---|---|
| Jordan | 11026 | 24312 | 35338 |
| Palestine | 10149 | 18280 | 28429 |
| Syrai | 13349 | 43811 | 57160 |
| Lebanon | 19540 | 0 | 19540 |
| Total | | | 140467 |

Table 7.2: Number of sentences for each dialect In Shami

- Diacritics removal Arabic text has several diacritics which affect the pronunciation of the words and some time the meaning. we remove these diacritics from the corpus which are : (اّ Tashdid, اَ *a* Fatha, اً *an* Tanwin Fath,اُ *u* Damma,اٌ *un* Tanwin Damm,اِ *i* Kasra,اٍ *in* Tanwin Kasr,اْ Sukun).
- Removal of non-Arabic symbols As we have used automatic tools to collect data so many words and letters are not Arabic text that may contain special characters like (@, !!, ??), number and dates, emotions and Latin script especially in Lebanese dialects which contains a lot of French and most of them write Arabic text using Latin letters.
- Normalization: Arabic Dialects texts suffer from the varieties in orthography due to absence of writing rules. we try to unify the style of writing by normalize letter that may written in different style. Table 7.3 lists the different styles for normalized Arabic letters. Most of the previous works apply full normalization to their corpus which in some time change the meaning of the words, so we put some rules on normalization steps to be more reliable and keep the semantic meaning of the text.

| Written style | Normalized style |
|---|---|
| أ، إ، آ *a* , *i* , *ā* | ا *ā* |
| ة *h* | ه *h* |
| ئ، ؤ *y* , *w* | ء |
| ى *ā* | ي *y* |
| گ *g* | ك *k* |

Table 7.3: Different styles for normalized Arabic letters

In SDC we define the following normalization rules:

- Alef: we only convert أ *a* to ا *ā* if it appears in the beginning of the word because a lot of dialectal text which is written like spoken used Alef styles to emulate the accent. For example we have word ('هلأ *hla*' /now) and we have ('هلا *hlā*'/Hello) so if we normalize the Alef letter then the meaning will be totally changed.
- Alef Maqsora ى *ā* At the end of the word: In most processing steps the letter ى *ā* is converted to a ي *y*, but we did not do so because a lot of words would change the meaning if we unified the characters. An example of this is (على *lā* / on preposition) and (علي *ly* Ali /a name for a male) . If we change the letter ى *ā*, this will affect the context of the sentence
- The remaining characters are changed and standardized as shown in the previous table.

• Lengthen words (Repeated characters): In most colloquial writings we find most people repeat some letters such as repeating the letter 'a' In Waaaaaaw. In previous works, all duplicates are removed and one or two characters are left to appear. In contrast, we have specified some criteria in the system based on the origin of the repetition in the Arabic language as the following:

  - We extracted all words containing repeated characters in texts written in the MSA.
  - We have identified all the characters in which the repetition is basic in word synthesis and we have specified them in a separate list, such as repeating a character ل *l* after the article definition ال *āl* like (الليل *āllyl* /the night) and repeating the letter of ر *r* in the word (مكرر *mkrr* /repeated).
  - All words containing duplicate characters from the previous list are abbreviated to only two characters.
  - The rest of the characters are abbreviated to only one character such as repeating the character و *w* in (مبروووووك *mbrwwwwwk* /congratulation) that is converted to (مبروك *mbrwk*/congratulation), these are just pragmatical one-of spellings to imitate spoken language.
  - We have a special case with letter (و *w* /and) It is the conjunction in the Arabic language. Some people in colloquial dialects

connect it with the next word without entering a space between the letter and the following word. We have made a condition that if the given words begins with more than one و *w* , the first و *w* and the rest of the word are separated and then the original word is treated according to the previous algorithm. Figure 7.2 describes the algorithm of reducing the appearance of repeated characters

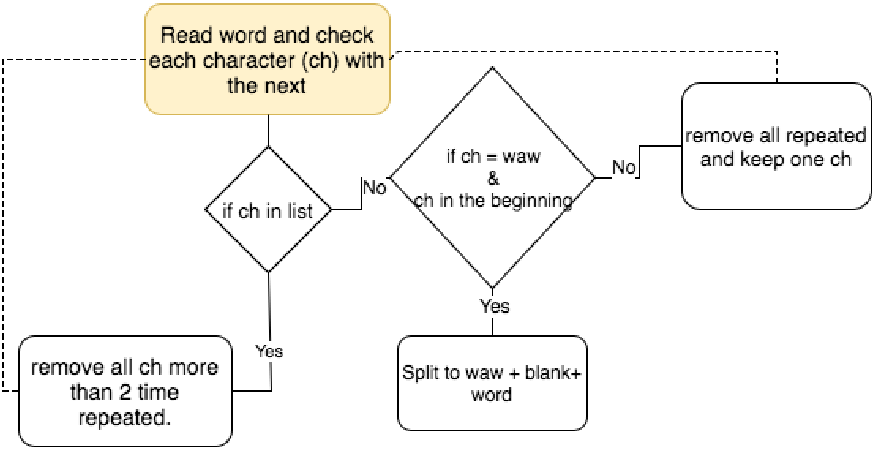

Read word and check each character (ch) with the next
if ch in list
No
if ch = waw & ch in the beginning
No
remove all repeated and keep one ch
Yes
Split to waw + blank+ word
Yes
remove all ch more than 2 time repeated.

Figure 7.2: Algorithm for repeated characters in dialectal words

- Corpus Purification: After the completion of the dialects processing, we have done a purification of the corpus to make sure it is free of some sentences written in the MSA or other dialects. The aim of this step is to obtain unified and separate dialects and because we have previously done automatic data extraction based on geographical location, there is a great confusion between the dialects so that many Palestinians live in Syria and thus we will find some Palestinian sentences in the Syrian data. Table 7.4 illustrates the statistics after filtering all dialects.

### 7.3.3 Comparison to the previous corpora

We mentioned earlier some corpora that included various dialects. Most of the dialectal corpora dealt with the Levantine dialects as if they were only one dialect. In this research, we will compare SDC with Padic Corpus [3] and Multi-dialect Corpus [8] as they separate Levantine dialects such as the Palestinian and Syrian dialects in Padic Corpus and the Palestinian,

| | Automatic | Manual | Total | words | vocabularies |
|---|---|---|---|---|---|
| Jordan | 8804 | 23274 | 32078 | 3684369 | 85383 |
| Palestine | 3566 | 17698 | 21264 | 2789103 | 69378 |
| Syrai | 4704 | 43455 | 48159 | 5268065 | 77918 |
| Lebanon | 16304 | - | 16304 | 1409952 | 44418 |
| Total | 33378 | 84427 | 117805 | 13151489 | 227097 |

Table 7.4: Number of sentences for each dialect after Purification step

Syrian and Jordanian dialects in the Multi-dialect Corpus. Tables 7.5 and 7.6 explain the statistics for each corpus.

| | sentences | words | vocabularies |
|---|---|---|---|
| Palestine | 6418 | 50827 | 22896 |
| Syrai | 6418 | 48701 | 27032 |
| Total | 12836 | 99528 | 49820 |

Table 7.5: Statistics for Padic corpus

| | sentences | words | vocabularies |
|---|---|---|---|
| Palestine | 1000 | 10315 | 8874 |
| Syrai | 1000 | 11586 | 9145 |
| Jordinian | 1000 | 9866 | 8905 |
| Total | 3000 | 31767 | 26924 |

Table 7.6: Statistics for multi-dialects corpus

Shami has been created so that it can be employed in several linguistics fields and NLP, one among them is Dialect Identification. We will use Arabic dialect identification as a test case for the corpus, to demonstrate its usefulness then we will compare it with PADIC corpus and Multi-dialect corpus.

## 7.4 Arabic Dialects Identification

A natural starting place for any Arabic dialectal processing is automatic Dialect Identification (DID) which enables the processing system to automatically classify the input dialect based on previous training or language modeling. Arabic dialect identification tasks refer to two levels of identification:

1. A coarse-grained level to build a learner that builds a learner with the ability to measure the percentage of the dialect contents given a specific Arabic sentence S,

2. A fine-grained level that can exactly classify sentence S to the related dialect in which it belongs.

Zaidan and Callison-Burch [10] extended the work for building a large annotated dataset [2] to train and evaluate automatic classifier for dialect Identification task. They classify the dialect according to the geographical distribution to 5 dominate dialects: Maghrebi, Egyptian, Levantine, Iraqi and Gulf. Their system recognizes the percentage of dialect on the sentence and then in which dialect it is written and they achieved 85.7% accuracy based on word-gram model. They conclude that using n-gram words and characters model is the most suitable methods to distinguish between these dialects.

The work done by [11] proposed sentence level identification and using words as tokens. They present a supervised method using Naive Bayes classifier to recognize the dialectal data and classify it between MSA and Egyptian dialect. They work on two parallel corpora. The first was an Egyptian - Levantine - English corpus of 5M tokenized words of with Egyptian (3.5M) and Levantine (1.5M). that corpus was part of BOLT data. The second was an MSA-English corpus with 57M tokenized words obtained from several LDC corpora. Their system achieved different accuracy regarding some preprocessing steps and the extracted features like percentage of dialect content, perplexity and Meta data. The highest accuracy was 85% on an Arabic online commentary dataset AOC. In [12] this work was extended to include the Iraqi, Levantine and Moroccan dialects.

The work on [13] presented some experiments using the character n-gram, Markov Model and NB classifiers for dialect Identification tasks. Their System has been trained and tested using a data set collected from blogs and forums of different countries with Arabic as an official language. They collected data from the 18 dialects for all countries in the Arab world ,how ever they conducted their experiments on the 6 main dialects based on Geographical Area (Egyptian, Levantine, Gulf, Iraqi, Maghrebi and others like Sudan). they result showed that NB classifier with Bi-gram models performs the best with accuracy of 98%. The Levantine data are the smallest among the dialects and get low accuracy totally. In their work they didn't measuring the work with any baseline.

Arabicized Arabic and dialect Arabic namely (Algerian, Egyptian, Levantine, Gulf, Mesopotamian (Iraqi), Moroccan, Tunisian) are focused on[14].

They introduced a new dialectal corpus and employed existing methods like SVM, Cavnar's text classier and prediction by partial matching to the task of identification. The system showed that machine leaning (ML) models combined with lexicons are well suited for dialects identification as they achieved 93% accuracy when employed on 9 dialects and combining all Levantine dialects together.

There is a lot of work on language identification. In addition, off-the-shelf tools for language identification are available and they are open source like langid.py [15] and langdetect [16]. So Language identification is considered a solved problem but this does not hold for Arabic dialect identification.

Non of Arabic Identification systems which have presented can be generalized to Arabic dialect content according to some points like (i) the data which are trained, mostly have come from the same domain, (ii) most data sets and corpora are small in size,(iii) also it hard due to that each system is trained and tested on different dataset with different parameters (size, domain, preprocessing).

Language identification is a well-known task and given a sufficient amount of resources it can be considered a solved task, however this does not hold for Arabic dialect identification. A lot of off-the-shelf tools for language identification are available and they are open source like langid.py [15] and langdetect [16]. Therefore, it is particularly suited as a task to be applied to verify our new corpus. Secondly, given that the dialects that we are focusing on are very similar, our experiments may also give new insights in respect to language identification.

## 7.4.1 Langid.py for language identification

Lui and Baldwin [15] presented an off-the-shelf tool for language identification called langid.py. In their tool they used Naive Bayes classifier with various n-gram character for training purpose. The tool was trained to identify 97 languages and covered multi-domain language identification corpus of [17]. The tool supports the developer with many modules so they can easily train and build their own language model; by following these steps:

- Indexing all the instances in the Corpus
- Tokenized every sentence in the corpus depending on the number of character grams.
- Choosing the features based on their document frequencies using Aho-Corasick string matching that employs Deterministic Finite Automation (DFA) states for processing every instance.
- Computing the Information Gain (IG) for every domain and every language in the given corpus.
- Selecting the most informative features depending on IG weights.

- Apply NB classifier and train the corpus to generate a language model that can Identify the proposed languages.

When the authors compare their tool with others language identification tool like langdetect, TextCat [18], and CDL [19] they found langid.py is faster and give better accuracy than others, because of that we used this tool to conduct our corpus evaluation on it.

### 7.4.2 Scikit learn tool for machine learning

Scikit-learn [20] is an open source python library that is very simple and efficient tool for data mining, data analysis and machine learning. It contains many modules like classification, regression and clustering beside other modules like preprocessing and feature selection. We used this tool as apart of our work to evaluate the system by word-gram models as langid.py does not support word gram and that are many language can be distinguished by their words.

## 7.5 Evaluation with language identification task

Two popular techniques are used in the literature for Language Identification tasks. One of them concerns with identifying lists of keywords for each language and scoring the text based on these lists [21]. The others employ Machine Learning techniques like Neural Networks [22, 23], Support Vector Machine [24], Hidden Markov Model [25] and n-gram models [26, 27] to distinguish among languages.

Our proposed Dialect Identification system is based on character n-gram and Naive Bayes classifiers. We use these approach because most of the variations between dialects are based on affixation, that can easily defined by language model beside the word features which can be decided by the lexicon.

In the following, we present several experiments in Dialect Identification between all the collected dialects in Shami Corpus. We have conducted several experiments, which varied between the size of the data and the Libraries used and Classification techniques as follows:

- Data Size: firstly we used the full Corpus then we take a sample from the data so the sparsity of the data reduced.
- Libraries: we depend on scikit-learn library and Langid library
- Techniques: we begin with various character grams models and apply Naive Bayes classifier using Langid.py library, then we used the word-gram models using scikit library as it is not supported in langid.py.

For evaluation purpose we measured the accuracy as the truly defined instances while the F-measure as the balance between the Precision and Recall.

### 7.5.1 Baseline system

To properly evaluate our SDC Performance we compare the language model classifiers to two Dialectal Corpus Padic [3] and multi-dialect corpus [8]. In this section we will summarized the experiments that have been carried out in them. Firstly we split the Corpora using a cross-validation technique where in each fold 90% is used for training data and 10% for evaluation purposes. Table 7.7 shows number of sentences(documents) for each fold.

| | Dialect | Train | test | Total |
|---|---|---|---|---|
| PADIC | Palestine | 5917 | 501 | 6418 |
| | Syrian | 5917 | 501 | 6418 |
| Multi dialect | Palestine | 900 | 100 | 1000 |
| | Syrian | 900 | 100 | 1000 |
| | Jordanian | 900 | 100 | 1000 |

Table 7.7: Train and test set for PADIC and Multi-dialect Corpus

We run the first experiment using Langid.py with 4,5,6 and 7 n-character grams to build the language models, then we verify that models using the verification set, while the second experiment was run on scikit learn on unigram and bi-gram word model. The results are shown on Tables 7.8,7.9 respectively for the two corpora.

| | Techniques | Accuracy | F-measure |
|---|---|---|---|
| langid.py | 4-gram char | 0.61 | 0.75 |
| | 5-gram char | 0.64 | 0.78 |
| | 6-gram char | 0.68 | 0.81 |
| | 7-gram char | 0.68 | 0.81 |
| Scikit learn | uni-gram word | 0.83 | 0.83 |
| | bi-gram word | 0.84 | 0.83 |

Table 7.8: Evaluation on PADIC

Generally, it is obvious that 6-gram model works best for language identification in the two corpora, as it appears to be picking out particular phrase. In PADIC, the scikit learn library with word gram model outperforms langid.py because as we mentioned before it is a parallel corpus where

| | Techniques | Accuracy | F-measure |
|---|---|---|---|
| langid.py | 4-gram char | 0.63 | 0.77 |
| | 5-gram char | 0.68 | 0.81 |
| | 6-gram char | 0.70 | 0.83 |
| | 7-gram char | 0.69 | 0.82 |
| Scikit learn | uni-gram word | 0.69 | 0.68 |
| | bi-gram word | 0.69 | 0.69 |

Table 7.9: Evaluation on Multi-dialects Corpus

the differences are greatly clarify when the corpus are built, beside many differences can be observed between Palestinian and Syrian. In contrast to the Multi-dialect Corpus, which encompasses three Levantine dialects. Here the distinction between words becomes harder and accordingly similarities between dialects especially the Palestinian and Jordanian dialects are increased.

### 7.5.2 Dialect Identification with Shami

Firstly we carried out some experiment to decide the size of data that give the highest results and enhance the performance of the model. When we used the full data (Table 7.4) we get low accuracy as shown in Table 7.10.

| Techniques | Accuracy | F-measure |
|---|---|---|
| 4-gram char | 0.36 | 0.52 |
| 5-gram char | 0.38 | 0.53 |
| 6-gram char | 0.38 | 0.55 |
| 7-gram char | 0.39 | 0.55 |

Table 7.10: Evaluation on Shami

As Shami is not considered a parallel corpus and not a crafted corpus as well, many sentences are not very clear to any language it belong. After many experiments we ended up with part of data to reduce the dispersion as shown in Table 7.11. The corpus filtering are done by help of four volunteers one from each Levantine country. We gave them their own dialectal documents, in order to extract only the most familiar sentences of their dialect than others. As a result of filtering, then the performance is increased because the sparsity of data reduced and the documents become more related to the dialect. Table 7.12 explains the results for training and evaluation on the filtered data from Shami.

| Dialect | Train | test | Total |
|---|---|---|---|
| Palestine | 9577 | 1065 | 10642 |
| Syrian | 33983 | 3776 | 37759 |
| Jordanian | 6316 | 702 | 7018 |
| Lebanese | 9747 | 1083 | 10830 |

Table 7.11: Train and test set for Shami after filtering

| | Techniques | Accuracy | F-measure |
|---|---|---|---|
| langid.py | 4-gram char | 0.54 | 0.70 |
| | 5-gram char | 0.65 | 0.71 |
| | 6-gram char | 0.55 | 0.71 |
| | 7-gram char | 0.55 | 0.71 |
| Scikit learn | uni-gram word | 0.70 | 0.71 |
| | bi-gram word | 0.70 | 0.70 |

Table 7.12: Evaluation on Sampling from SDC

Despite the improved performance of the system, it did not increase significantly so that it can not distinguish very accurately among the dialects due to the great similarity among the four dialects and because many sentences did not contain any dialectal key words. To confirm this result, we have done a survey with several sentences without dialectal keywords and asked some Levantine native speakers to identify each sentence to any dialect it belongs. For example, one of the sentences was (الدرس اليوم كان كتير حلو *āldrs ālywm kān ktyr ḥlw* وما حسينا بملل بالمرة. *wmā ḥsynā bmll bālmrt.* ياريت كل يوم يكون هيك *yāryt kl ywm ykwn hyk* ) which in English means (The class today was very nice and interesting and we never felt bored. Hopefully every day be like that). No one can definitely classify this sentence to any Levantine dialect it should belong.

### Minimizing the number of dialects

Due to the greatly similarity between the Levantine dialects therefore, we have conducted several experiments to reduce the number of dialects used in the classification. We used the Palestinian with Syrian languages - such as the PADIC corpus -, Jordanian with Lebanese and then we used three classifications Like Multi-dialect corpus (Palestinian, Jordanian, Syria). Therefore, because of what we have explained about the similarity of the Palestinian with the others, we excluded this dialect and conduct a final classification based on (Jordanian, Syrian, Lebanese). The results are shown in Tables 7.13, 7.14, 7.15 and 7.16 respectively:

| | Techniques | Accuracy | F-measure |
|---|---|---|---|
| langid.py | 4-gram char | 0.73 | 0.83 |
| | 5-gram char | 0.72 | 0.83 |
| | 6-gram char | 0.72 | 0.84 |
| | 7-gram char | 0.72 | 0.83 |
| Scikit learn | uni-gram word | 0.87 | 0.85 |
| | bi-gram word | 0.80 | 0.74 |

Table 7.13: Evaluation on two dialects classification (Palestinian, Syrian)

| | Techniques | Accuracy | F-measure |
|---|---|---|---|
| langid.py | 4-gram char | 0.87 | 0.88 |
| | 5-gram char | 0.89 | 0.89 |
| | 6-gram char | 0.89 | 0.89 |
| | 7-gram char | 0.89 | 0.89 |
| Scikit learn | uni-gram word | 0.90 | 0.90 |
| | bi-gram word | 0.88 | 0.88 |

Table 7.14: Evaluation on two dialects classification (Jordanian, Lebanese)

| | Techniques | Accuracy | F-measure |
|---|---|---|---|
| langid.py | 4-gram char | 0.65 | 0.78 |
| | 5-gram char | 0.65 | 0.79 |
| | 6-gram char | 0.65 | 0.78 |
| | 7-gram char | 0.64 | 0.78 |
| Scikit learn | uni-gram word | 0.77 | 0.71 |
| | bi-gram word | 0.70 | 0.60 |

Table 7.15: Evaluation on three dialects classification (Palestinian, Jordanian, Syrian)

Figures 7.3, 7.4 outline the F-measure for the classification task between two and three dialects comparing with PADIC corpus and Multi-dialect corpus. It seems to us that if the number of dialects is reduced, the Accuracy and F-measure increase better, as the dispersion in the data decreases, also

| | Techniques | Accuracy | F-measure |
|---|---|---|---|
| langid.py | 4-gram char | 0.64 | 0.78 |
| | 5-gram char | 0.81 | 0.82 |
| | 6-gram char | 0.66 | 0.79 |
| | 7-gram char | 0.65 | 0.79 |
| Scikit learn | uni-gram word | 0.75 | 0.70 |
| | bi-gram word | 0.70 | 0.60 |

Table 7.16: Evaluation on three dialects classification (Jordanian, Syrian,Lebanese)

the clarity of the difference becomes slightly clearer. When we classify between Jordanian and Lebanese we get the highest results as there is little similarity between them and they are to some extent different, then can be distinguished by text. From all the previous experiments we conclude the similarity between the Levantine dialects of Shami Corpus in the case of writing and the difficulty of differentiation among them, because the SDC is not a parallel dictionary that can not be clarified textual differences between the sentences as much as in PADIC and Multi-dialect Corpus, However it addresses the similarities and differences among the fours dialects significantly comparing with the two previous corpora

## 7.6 Discussion and Conclusion

In this paper we first presented Shami the first Levantine Dialects corpus containing the dialects from Palestine, Jordan, Syria, Lebanon. Shami is much more varied than the previous existing corpora and introduces new data which previously not available. We have adopted two methods (automatic and manual combination) to collect the Levantine documents, then we employed some pre-processes in the data, in addition to filtering the corpus to make it clearly dialectal and fit the linguistics tasks accurately.

we analyze the impact of language model on the task of Dialect Identification by applying various n-grams model using different library. In the same time we compare SDC with PADIC corpus and Multi-Dialect corpus The best results are achieved when we classify the two dialects (Jordanian and Lebanese) that gives 90% accuracy using uni-gram word model. The result is not surprising because of the little similarity between the 2 dialects on the level of lexical. The worst results we get were when we applying the whole SDC and classify the four dialects that gives 52% accuracy. This is due to the great overlap between the dialect and their dispersion. We

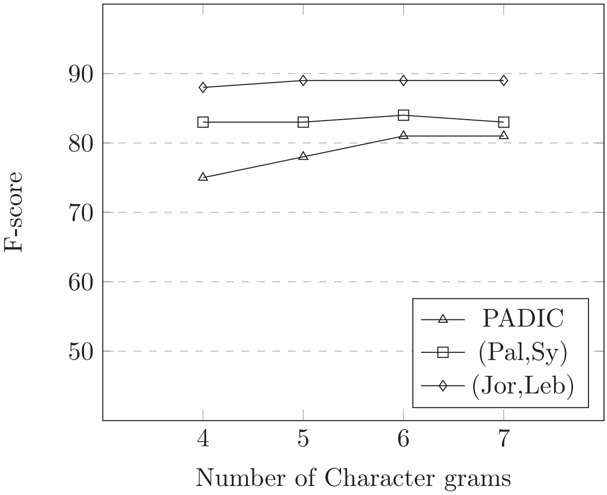


Figure 7.3: F-score on two-way dialect classification

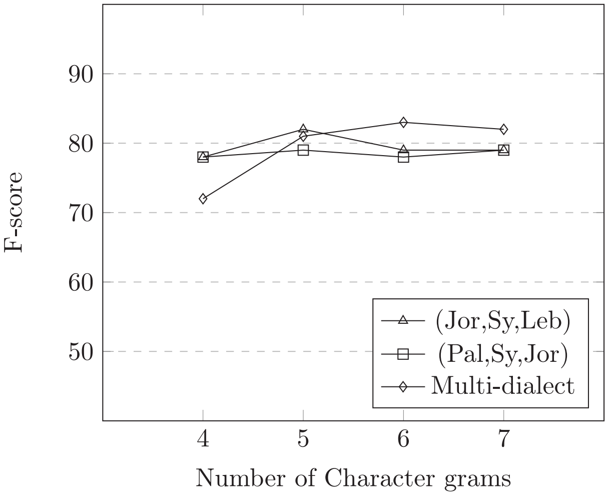


Figure 7.4: F-score on three-way dialect classification

found that SDC outperforms the baseline when comparing with the same dialects as it cover more vocabularies even it is not parallel corpus where the difference are highly visible and easily distinguishable .

Our future work consists in enhancing Shami and extract the dialectal keywords from each dialect , beside that we will measure the extent of convergence and divergence between the Levantine dialects. We will try to merge more than one techniques together to improve the accuracy of language identification.

## 7.7 References

[1] Rabih Zbib et al. "Machine Translation of Arabic Dialects". In: *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics. 2012, pp. 49–59.

[2] Omar F. Zaidan and Chris Callison-Burch. "The Arabic Online Commentary Dataset: An Annotated Dataset of Informal Arabic with High Dialectal Content". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics. 2011, pp. 37–41.

[3] Karima Meftouh et al. "Machine Translation Experiments on PADIC: A Parallel Arabic Dialect Corpus". In: *The 29th Pacific Asia conference on language, information and computation*. 2015.

[4] Mustafa Jarrar et al. "Building a Corpus for Palestinian Arabic: A Preliminary Study". In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*. 2014, pp. 18–27.

[5] Arfath Pasha et al. "MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic." In: *LREC*. Vol. 14. 2014, pp. 1094–1101.

[6] Mona Diab et al. "COLABA: Arabic Dialect Annotation and Processing". In: *Lrec workshop on semitic language processing*. 2010, pp. 66–74.

[7] Ryan Cotterell and Chris Callison-Burch. "A Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic." In: *LREC*. 2014, pp. 241–245.

[8] Houda Bouamor, Nizar Habash, and Kemal Oflazer. "A Multidialectal Parallel Corpus of Arabic." In: *LREC*. 2014, pp. 1240–1245.

[9] Khalid Almeman and Mark Lee. “Automatic Building of Arabic Multi Dialect Text Corpora by Bootstrapping Dialect Words”. In: *Communications, signal processing, and their applications (iccspa), 2013 1st international conference on*. IEEE. 2013, pp. 1–6.

[10] Omar F Zaidan and Chris Callison-Burch. “Arabic Dialect Identification”. In: *Computational Linguistics* 40.1 (2014), pp. 171–202.

[11] Heba Elfardy and Mona T Diab. “Sentence Level Dialect Identification in Arabic.” In: *ACL (2)*. 2013, pp. 456–461.

[12] Wael Salloum et al. “Sentence Level Dialect Identification for Machine Translation System Selection”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2014, pp. 772–778.

[13] Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. “Automatic Identification of Arabic Dialects in Social Media”. In: *Proceedings of the first international workshop on Social media retrieval and analysis*. ACM. 2014, pp. 35–40.

[14] Wafia Adouane et al. “Automatic Detection of Arabicized Berber and Arabic Varieties”. In: *VarDial 3* (2016), p. 63.

[15] Marco Lui and Timothy Baldwin. “langid.py: An Off-the-shelf Language Identification Tool”. In: *Proceedings of the ACL 2012 system demonstrations*. Association for Computational Linguistics. 2012, pp. 25–30.

[16] Nakatani Shuyo. “Language Detection Library for Java”. In: *Retrieved Jul* 7 (2010), p. 2016.

[17] Marco Lui and Timothy Baldwin. “Cross-Domain Feature Selection for Language Identification”. In: *In Proceedings of 5th International Joint Conference on Natural Language Processing*. Citeseer. 2011.

[18] William B Cavnar, John M Trenkle, et al. “N-gram-based Text Categorization”. In: *Ann Arbor MI* 48113.2 (1994), pp. 161–175.

[19] Michael McCandless. *Accuracy and Performance of Google’s Compact Language Detector*. 2011.

[20] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830.

[21] Fred S Richardson and William M Campbell. “Language Recognition with Discriminative Keyword Selection”. In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE. 2008, pp. 4145–4148.

[22] Shawki A Al-Dubaee et al. “Language Identification using Wavelet Transform and Artificial Neural Network”. In: *Computational Aspects of Social Networks (CASoN), 2010 International Conference on*. IEEE. 2010, pp. 515–520.

[23] Javier Gonzalez-Dominguez et al. “Automatic Language Identification using Long Short-Term Memory Recurrent Neural Networks”. In: *Fifteenth Annual Conference of the International Speech Communication Association*. 2014.

[24] Gerrit Reinier Botha et al. “Text-based Language Identification for the South African Languages”. PhD thesis. University of Pretoria, 2008.

[25] Ted Dunning. *Statistical Identification of Language*. Computing Research Laboratory, New Mexico State University, 1994.

[26] Xi Yang and Wenxin Liang. “An N-gram and Wikipedia Joint Approach to Natural Language Identification”. In: *Universal Communication Symposium (IUCS), 2010 4th International*. IEEE. 2010, pp. 332–339.

[27] Ali Selamat. “Improved N-grams Approach for Web Page Language Identification”. In: *Transactions on computational collective intelligence V*. Springer, 2011, pp. 1–26.

# 8

# Study 2: Investigate Language Modelling and Ensemble Learning for Fine Grained Arabic Dialect Identification

*Kathrein Abu Kwaik and Motaz K Saad. "ArbDialectID at MADAR Shared Task 1: Language Modelling and Ensemble Learning for Fine Grained Arabic Dialect Identification." In:ArbDialectID at MADAR Shared Task 1: Language Modelling and Ensemble Learning for Fine Grained Arabic Dialect Identification Proceedings of the Fourth Arabic Natural Language Processing Workshop (2019)*

In this paper, we present a Dialect Identification system (ArbDialectID) that competed at Task 1 of the MADAR shared task, “MADAR Travel Domain Dialect Identification”. We build a coarse and a fine grained identification model to predict the label (corresponding to a dialect of Arabic) of a given text. We build two language models by extracting features at two levels (words and characters). We firstly build a coarse identification model to classify each sentence into one out of six dialects, then use this label as a feature for the fine grained model that classifies the sentence among 26 dialects from different Arab cities, after that we apply ensemble voting classifier on both subsystems. Our system ranked 1st that achieving an f-score of 67.32%. Both the models and our feature engineering tools are made available to the research community.

## 8.1 Introduction

Arabic Language is one of the most spoken languages in the world. Furthermore, Arabic presents us with a special case of Diglossia [1], where the spoken language is different than the formal language. Speakers of Arabic use Modern Standard Arabic (MSA) as the official language in very formal situations like education, religion, media, and politics, while they use an Arabic Dialect (AD) for everyday conversation [2, 3].

With the emergence of social media, speakers of Arabic use their dialects to tweet, post, socialize and express themselves. The Arabic Dialects (AD) do not have a standardized writing and/or orthography, and they do not have a formal grammar. These characteristics make the task of identifying dialects more challenging.

The task of Arabic Dialect Identification (ADI) has recently attracted research attention, building identification systems able to differentiate among the dialects have been attempted. Even though dialects share similar features in term of lexical, syntax, morphology and semantics, they still have many differences which, of course, complicates the identification task.

Many works addressed the problem of dialect identification. They have reported different dialectal divisions, according to the geo-location, the country or, in some cases, on the level of cities. Most of those works used Machine learning classifiers and language modelling and achieved a good accuracy depending on the level of identification and either they explored the coarse grained identification, where the differences between the individual dialects are clear or a fine grained identification, where the differences become hard to detect in text as the dialects look very similar to each others [4, 5, 6, 7, 8].

Other approaches investigated the use of Deep Learning (DL) methods to identify dialects. As such, they tried different DL architectures like LSTMs, CNNs and attention networks, and have employed different word embedding models. elaraby2018deep benchmarked the Arabic Online Commentary (AOC) [9] and tested six different deep learning methods on the ADI task, comparing performance to several classical machine learning models under different conditions (both binary and multi-way classification). Their models reached 87.65% accuracy on the binary task (MSA vs. dialects), 87.4% accuracy on the three-way dialect task (Egyptian, Gulf, Levantine), and 82.45% accuracy on the four-way classification task (MSA, Egyptian, Gulf, Levantine). Similarly, [10] explored the DL methods with different networks structure using AOC on a three-way classification, with LSTM they achieved 71.4% accuracy

This paper presents our participation in MADAR shared task [11]. We participate in Task 1: MADAR travel domain dialect identification, and we ranked 1st in the task with accuracy of 67.3%. We present our proposed model (ArbDialectID) in details and the code is available at GitHub[1].

The rest of this paper is organized as follow: Section 3 discusses the used data and presents our proposed model, we discusses the results in Section 4 and conclude in Section 5.

## 8.2 ArbDialectID: Arabic Dialect Identification System

This section introduces our proposed model which is applied on MADAR corpus for dialect identification shared task. MADAR corpus [12] is a parallel corpus in travel domain, it contains 25 dialects from different Arab cities in addition to the MSA. This corpus has been used for ADI task in [13], where the authors applied language modeling with various combinations of word and character levels and trained the model by MNB classifier. They got 67.9% accuracy for 26 classification task.

Our model consists of two sub models and exploiting two different data set as shown in Figure 8.1. The first model tries to predict the dialect among six different Arab dialects and known as coarse grained level, followed by the second model which goes much deeper and is known as a fine grained level to classify 26 Arabic dialects.

In both of our sub models we use MADAR data set to build and evaluate the models. Table 8.1 shows the number of sentences/samples per dialects and the total sentences for each data set. All of the experiments are implemented by Python and with the help of `Scikit learn` library [14].

| MADAR | Split | sentences | Total |
|---|---|---|---|
| **Corpus-6** | train | 9,000 | 41,600 |
| | dev | 1,000 | 6,000 |
| **Corpus-26** | train | 1,600 | 41,600 |
| | dev | 200 | 5,200 |
| | test | 200 | 5,200 |

Table 8.1: Statistics for MADAR data sets

### 8.2.1 Coarse Grained Dialect Identification

This is the first model where we classify among five different Arab dialects from five Arabic countries, which are covered by MADAR corpus, they are:

[1] https://github.com/motazsaad/ArbDialectID

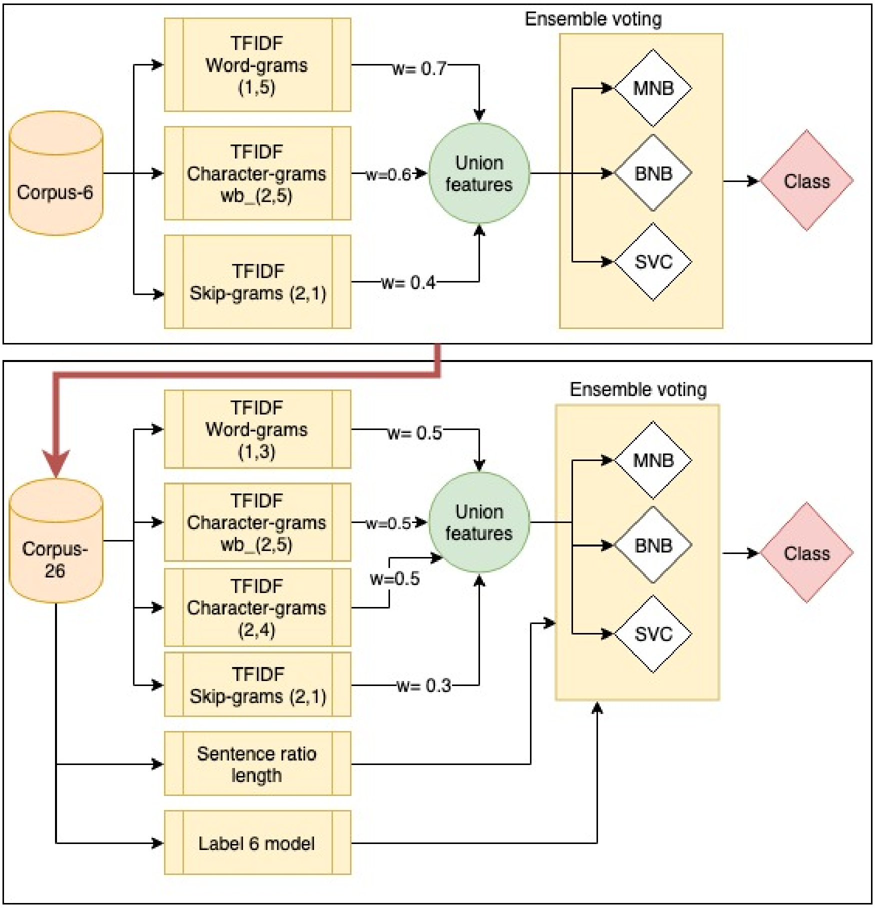


Figure 8.1: ArbDialectID proposed model

Beirut (BEI), Cairo (CAI), Doha (DOH), Rabat (RAB), Tunisia (TUN), In addition to (MSA).

We build a model that depends on the language modelling and exploring different combinations of n-grams in the word level and the character level. We use *FeatureUnion* in *sklearn*, which is an estimator that concatenates results of multiple transformer objects. To build and train the model we extract the following features:

- TF-IDF vectors from the word grams ranged from the unigram to 5-grams. We apply 0.7 weight for vector transformation
- TF-IDF vectors from the character n-grams with word boundary con-

sideration ranged from bigrams to 5-grams and the transformation weight is 0.6
- Apply skip grams , then we extract the uni-gram words with one word skipping. We give it the lowest transformation weight of 0.4

The transformation weight is a weight used in *FeatureUnion* to give a weight for the feature. We choose these weights empirically after many experiments that investigate various weights with many features combinations.

After features extraction process, we build an ensemble voting classifier with hard voting, where it uses predicted class labels for majority rule voting. The ensemble classifiers consists of the following best standalone Machine Learning algorithms:

- MultinomialNB (MNB) , we set alpha to 0.01
- Linear SVC with l2 penalty and the learning rate sets to 0.0001
- BernoulliNB (BNB), set alpha = 0.01

We trained the model using "MADAR corpus-6" train set, and evaluate it by MADAR corpus-6 development set. We reach an accuracy of 92.7% and macro F-score of 93%. Finally, we combine the train and the dev-set together and rebuild the model again. We call it (MADAR model-6). We will use this model later in the second sub model.

### 8.2.2 Fine Grained Dialect Identification

This model is the core of the shared task, where it is going to predict the label for a given sentence and classify it to one of 26 dialects. MADAR corpus covers 25 cities in the Arab countries in addition to the MSA, they are : Aleppo (ALE), Algeria (ALG), Alexandria (ALX), Amman (AMM), Aswan (ASW), Baghdad (BAG), Basra (BAS), Beirut (BEI), Benghazi (BEN), Cairo (CAI), Damascus (DAM), Doha (DOH), Fes (FES), Jeddah (JED), Jerusalem (JER), Khartoum (KHA), Mosul (MOS), Muscat (MUS), Rabat (RAB), Riyadh (RIY), Salt (SAL),Sana'a (SAN), Sfax (SFX), Tripoli (TRI), Tunisia (TUN) and MSA.

In the same manner we build the second model by extracting some features as follow:

- TF-IDF vectors from the word grams with uni-gram, bi-gram and tri-gram words. we apply 0.5 weight for vector transformation
- TF-IDF vectors from the character n-grams with word boundary consideration ranged from bi-grams to 5-grams and the transformation weight is 0.5
- Extract another character n-grams but this time without word boundary consideration from bi-grams to 4 grams and the transformation weight is 0.5

- Again apply skip gram, then we extract the uni-gram words with one work skipping. We assign it 0.3 transformation weight

In addition to theses feature we add another two numerical features, the first is the sentence length ratio for every sentence in the data (train, dev, test) which in turn divides the total number of words appearing in the sentence by the total number of words appearing in the longest sentence. The second features depends on the previous MADAR-model-6. We exploit this model to predict the label for MADAR Corpus-26, so every sentence is combined with a predicted class number with one value from 1 to 6, for example 1 means CAI, 2 is for BEI and so on. So in total we have the TF-IDF vectors features in addition to the two numerical features (the coarse-grained label and the sentence length).

To build the model, we employ ensemble hard voting classifier with the previously mentioned three algorithms (Linear SVC, MNB and BNB). The system is trained on MADAR corpus-26 train set, evaluated by MADAR corpus-26 dev set and finally tested by MADAR corpus-26 test set. Table 8.2 reports the results for the dev set and test set and Figure 8.2 shows the classification report which is produced from the test set .

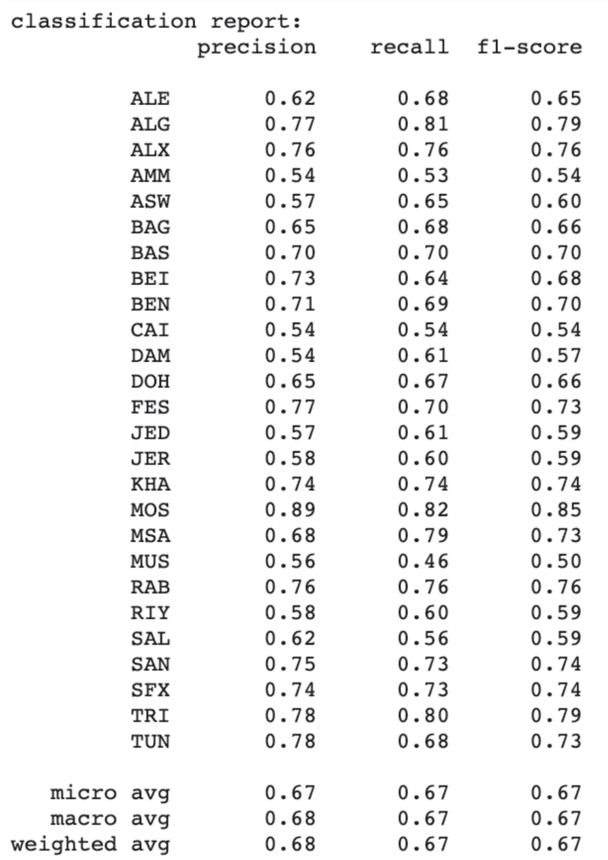
| | Accuracy | macro F-score |
|---|---|---|
| Dev | 68.7 | 69.00 |
| Test | 67.29 | 67.32 |

Table 8.2: Results for 26 dialects Identification system

## 8.3 Discussion

Building a language model for a language or a text is an informative way to describe and represent the language. In this work, we try to extract as many discriminated features as possible that can be employed efficiently to distinguish among the desired 6 and 26 dialects. In the coarse grained dialect identification with MADAR Corpus-6 the task was more flexible, the dialects have a reasonable differences as they represent a large groups of dialects, for example DOH represents dialects from the Arab Gulf, BEI represents the Levantine dialects and so on. Due to the differences on the lexical level between thees dialects we emphasise the word n-grams by using greater weight transformation, and assign a smaller weight value for the character levels n-grams.

For the task of fine grained dialect identification, the task was more tough and we need more extra features and emphasise some of them more.



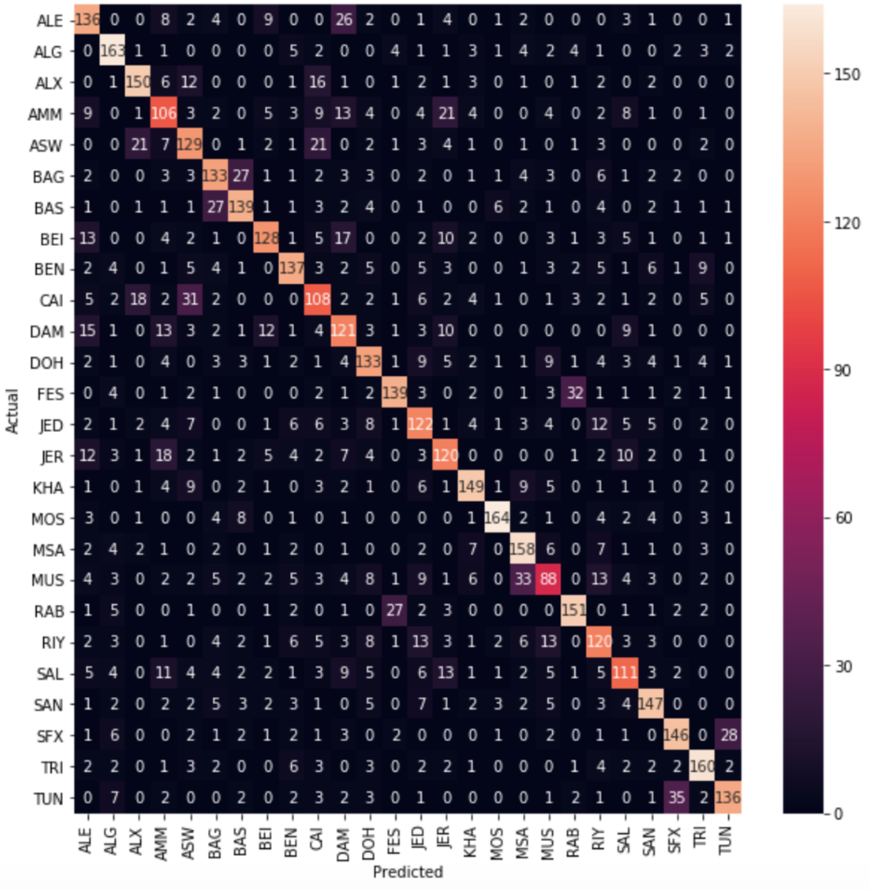
```
classification report:
                 precision    recall  f1-score

          ALE       0.62      0.68      0.65
          ALG       0.77      0.81      0.79
          ALX       0.76      0.76      0.76
          AMM       0.54      0.53      0.54
          ASW       0.57      0.65      0.60
          BAG       0.65      0.68      0.66
          BAS       0.70      0.70      0.70
          BEI       0.73      0.64      0.68
          BEN       0.71      0.69      0.70
          CAI       0.54      0.54      0.54
          DAM       0.54      0.61      0.57
          DOH       0.65      0.67      0.66
          FES       0.77      0.70      0.73
          JED       0.57      0.61      0.59
          JER       0.58      0.60      0.59
          KHA       0.74      0.74      0.74
          MOS       0.89      0.82      0.85
          MSA       0.68      0.79      0.73
          MUS       0.56      0.46      0.50
          RAB       0.76      0.76      0.76
          RIY       0.58      0.60      0.59
          SAL       0.62      0.56      0.59
          SAN       0.75      0.73      0.74
          SFX       0.74      0.73      0.74
          TRI       0.78      0.80      0.79
          TUN       0.78      0.68      0.73

    micro avg       0.67      0.67      0.67
    macro avg       0.68      0.67      0.67
 weighted avg       0.68      0.67      0.67
```

Figure 8.2: Fine Grained Dialect Identification classification report for MADAR corpus-26 test set

Hence, we increase the number of n-grams and emphasise the character n-grams and pay attention to the words boundaries. We employ the first model as another feature to enhance the f-score for the second models. Given that, the corpus contains many short sentence that appears in more one dialects, it makes the models to some extent confused, then we add the length of the sentence as an extract helpful feature where some dialects need more words to express an idea, and the other use more suffixes. It is also impossible for Arabic speakers to detect the dialect from a very short sentence with 100% especially if it does not contain any clue words. In some cases the dialects become very similar to each others when they are spoken by neighbourhood, for instance the Jerusalem dialect and the dialect from Amman where they are considered in some researches in Arabic history as the same dialect [15, 16]. From the classification report in Figure 8.2, it is very clear that some dialects were easier to detect than other, for example, the North Africa dialects gain high f-scores compare to others such as the following dialects: TRI (0.79), SFX(0.74), BEN(0.70), ALG(0.79) and TUN(0.73). The confusion matrix in Figure 8.3 shows the numbers of actual and predicate labels for each dialect. There are some similar pairs of dialects where the system confused like (BAG and BAS), (AMM and JER), (CAI and ASW), (ALE and DAM) and (SFX and TUN).



| | ALE | ALG | ALX | AMM | ASW | BAG | BAS | BEI | BEN | CAI | DAM | DOH | FES | JED | JER | KHA | MOS | MSA | MUS | RAB | RIY | SAL | SAN | SFX | TRI | TUN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ALE | 136 | 0 | 0 | 8 | 2 | 4 | 0 | 9 | 0 | 0 | 26 | 2 | 0 | 1 | 4 | 0 | 1 | 2 | 0 | 0 | 0 | 3 | 1 | 0 | 0 | 1 |
| ALG | 0 | 163 | 1 | 1 | 0 | 0 | 0 | 0 | 5 | 2 | 0 | 0 | 4 | 1 | 1 | 3 | 1 | 4 | 2 | 4 | 1 | 0 | 0 | 2 | 3 | 2 |
| ALX | 0 | 1 | 150 | 6 | 12 | 0 | 0 | 0 | 1 | 16 | 1 | 0 | 1 | 2 | 1 | 3 | 0 | 1 | 0 | 1 | 2 | 0 | 2 | 0 | 0 | 0 |
| AMM | 9 | 0 | 1 | 106 | 3 | 2 | 0 | 5 | 3 | 9 | 13 | 4 | 0 | 4 | 21 | 4 | 0 | 0 | 4 | 0 | 2 | 8 | 1 | 0 | 1 | 0 |
| ASW | 0 | 0 | 21 | 7 | 129 | 0 | 1 | 2 | 1 | 21 | 0 | 2 | 1 | 3 | 4 | 1 | 0 | 1 | 0 | 1 | 3 | 0 | 0 | 0 | 2 | 0 |
| BAG | 2 | 0 | 0 | 3 | 3 | 133 | 27 | 1 | 1 | 2 | 3 | 3 | 0 | 2 | 0 | 1 | 1 | 4 | 3 | 0 | 6 | 1 | 2 | 2 | 0 | 0 |
| BAS | 1 | 0 | 1 | 1 | 1 | 27 | 139 | 1 | 1 | 3 | 2 | 4 | 0 | 1 | 0 | 0 | 6 | 2 | 1 | 0 | 4 | 0 | 2 | 1 | 1 | 1 |
| BEI | 13 | 0 | 0 | 4 | 2 | 1 | 0 | 128 | 1 | 5 | 17 | 0 | 0 | 2 | 10 | 2 | 0 | 0 | 3 | 1 | 3 | 5 | 1 | 0 | 1 | 1 |
| BEN | 2 | 4 | 0 | 1 | 5 | 4 | 1 | 0 | 137 | 3 | 2 | 5 | 0 | 5 | 3 | 0 | 0 | 1 | 3 | 2 | 5 | 1 | 6 | 1 | 9 | 0 |
| CAI | 5 | 2 | 18 | 2 | 31 | 2 | 0 | 0 | 0 | 108 | 2 | 2 | 1 | 6 | 2 | 4 | 1 | 0 | 1 | 3 | 2 | 1 | 2 | 0 | 5 | 0 |
| DAM | 15 | 1 | 0 | 13 | 3 | 2 | 1 | 12 | 1 | 4 | 121 | 3 | 1 | 3 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 1 | 0 | 0 | 0 |
| DOH | 2 | 1 | 0 | 4 | 0 | 3 | 3 | 1 | 2 | 1 | 4 | 133 | 1 | 9 | 5 | 2 | 1 | 1 | 9 | 1 | 4 | 3 | 4 | 1 | 4 | 1 |
| FES | 0 | 4 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 2 | 1 | 2 | 139 | 3 | 0 | 2 | 0 | 1 | 3 | 32 | 1 | 1 | 1 | 2 | 1 | 1 |
| JED | 2 | 1 | 2 | 4 | 7 | 0 | 0 | 1 | 6 | 6 | 3 | 8 | 1 | 122 | 1 | 4 | 1 | 3 | 4 | 0 | 12 | 5 | 5 | 0 | 2 | 0 |
| JER | 12 | 3 | 1 | 18 | 2 | 1 | 2 | 5 | 4 | 2 | 7 | 4 | 0 | 3 | 120 | 0 | 0 | 0 | 0 | 1 | 2 | 10 | 2 | 0 | 1 | 0 |
| KHA | 1 | 0 | 1 | 4 | 9 | 0 | 2 | 1 | 0 | 3 | 2 | 1 | 0 | 6 | 1 | 149 | 1 | 9 | 5 | 0 | 1 | 1 | 1 | 0 | 2 | 0 |
| MOS | 3 | 0 | 1 | 0 | 0 | 4 | 8 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 164 | 2 | 1 | 0 | 4 | 2 | 4 | 0 | 3 | 1 |
| MSA | 2 | 4 | 2 | 1 | 0 | 2 | 0 | 1 | 2 | 0 | 1 | 0 | 0 | 2 | 0 | 7 | 0 | 158 | 6 | 0 | 7 | 1 | 1 | 0 | 3 | 0 |
| MUS | 4 | 3 | 0 | 2 | 2 | 5 | 2 | 2 | 5 | 3 | 4 | 8 | 1 | 9 | 1 | 6 | 0 | 33 | 88 | 0 | 13 | 4 | 3 | 0 | 2 | 0 |
| RAB | 1 | 5 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 1 | 0 | 27 | 2 | 3 | 0 | 0 | 0 | 0 | 151 | 0 | 1 | 1 | 2 | 2 | 0 |
| RIY | 2 | 3 | 0 | 1 | 0 | 4 | 2 | 1 | 6 | 5 | 3 | 8 | 1 | 13 | 3 | 1 | 2 | 6 | 13 | 0 | 120 | 3 | 3 | 0 | 0 | 0 |
| SAL | 5 | 4 | 0 | 11 | 4 | 4 | 2 | 2 | 1 | 3 | 9 | 5 | 0 | 6 | 13 | 1 | 1 | 2 | 5 | 1 | 5 | 111 | 3 | 2 | 0 | 0 |
| SAN | 1 | 2 | 0 | 2 | 2 | 5 | 3 | 2 | 3 | 1 | 0 | 5 | 0 | 7 | 1 | 2 | 3 | 2 | 5 | 0 | 3 | 4 | 147 | 0 | 0 | 0 |
| SFX | 1 | 6 | 0 | 0 | 2 | 1 | 2 | 1 | 2 | 1 | 3 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 146 | 0 | 28 |
| TRI | 2 | 2 | 0 | 1 | 3 | 2 | 0 | 0 | 6 | 3 | 0 | 3 | 0 | 2 | 2 | 1 | 0 | 0 | 0 | 1 | 4 | 2 | 2 | 2 | 160 | 2 |
| TUN | 0 | 7 | 0 | 2 | 0 | 0 | 2 | 0 | 2 | 3 | 2 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | 1 | 35 | 2 | 136 |

Figure 8.3: Fine Grained Dialect Identification confusion matrix for MADAR corpus-26 test set

We investigate the word grams model as well as the character grams model. The best result is obtained when we combine both of these models, given that the differences may occur in terms of lexical words, however there are many differences that occurred on character levels like different clitics, prefixes and suffixes. We try to exploit the best classifier that has been used for ADI and finally end up by ensemble learning that combines the Linear SVC , MNB and BNB with hard voting where the max probability is chosen as the correct class.

## 8.4 Conclusion

We participate in MADAR shared task, Task 1: "MADAR Travel Domain Dialect Identification". We build an ADI system consists of two subsystems. The first is a six dialects classification system, followed by a 26 classification system that classify 26 dialects from 25 cities in the Arab world in addition to MSA. We use different combinations of n-gram models (words, Characters) and skip gram models. In addition to these language modelling features, we compute the ratio length of each input sentence and use the predicted label from the first model. We achieve the best score in the competition with 67.32% f-score and an accuracy of 67.29%.

## Acknowledgements

Kathrein Abu Kwaik is supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg.

## 8.5 References

[1] Charles A Ferguson. “Diglossia”. In: *word* 15.2 (1959), pp. 325–340.

[2] Mustafa Shah. *The Arabic Language*. Routledge, 2008.

[3] Kees Versteegh. *The Arabic Language*. Edinburgh University Press, 2014.

[4] Rabih Zbib et al. “Machine Translation of Arabic Dialects”. In: *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics. 2012, pp. 49–59.

[5] Ryan Cotterell and Chris Callison-Burch. “A Multi-Dialect, Multi-Genre Corpus of Informal Written Arabic.” In: *LREC*. 2014, pp. 241–245.

[6] Omar F Zaidan and Chris Callison-Burch. “Arabic Dialect Identification”. In: *Computational Linguistics* 40.1 (2014), pp. 171–202.

[7] Kathrein Abu Kwaik et al. “Shami: A Corpus of Levantine Arabic Dialects”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[8] Heba Elfardy and Mona T Diab. “Sentence Level Dialect Identification in Arabic.” In: *ACL (2)*. 2013, pp. 456–461.

[9] Omar F. Zaidan and Chris Callison-Burch. “The Arabic Online Commentary Dataset: An Annotated Dataset of Informal Arabic with High Dialectal Content”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics. 2011, pp. 37–41.

[10] Leena Lulu and Ashraf Elnagar. “Automatic Arabic Dialect Classification Using Deep Learning Models”. In: *Procedia computer science* 142 (2018), pp. 262–269.

[11] Houda Bouamor, Sabit Hassan, and Nizar Habash. “The MADAR Shared Task on Arabic Fine-Grained Dialect Identification”. In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19)*. Florence, Italy, 2019.

[12] Houda Bouamor et al. “The MADAR Arabic Dialect Corpus and Lexicon”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[13] Mohammad Salameh, Houda Bouamor, and Nizar Habash. “Fine-Grained Arabic Dialect Identification”. In: *Proceedings of the International Conference on Computational Linguistics (COLING).* Santa Fe, New Mexico, USA, 2018, pp. 1332–1344.

[14] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830.

[15] Jonathan Owens. “Arabic Language History and The Comparative Method”. In: *International Journal of Arabic Linguistics* 1.1 (2015), pp. 1–27.

[16] Brian Bishop. “A History of The Arabic Language”. In: *Department of Linguistics, Brigham Young University* (1998).

# 9

# Study 3: Computational Cross Dialectal Lexical Distance Study

*Kwaik, Kathrein Abu, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnika. "A Lexical Distance Study of Arabic Dialects." Procedia computer science 142, (2018): pp. 2-13.*

Diglossia is a very common phenomenon in Arabic-speaking communities, where the spoken language is different from both Classical Arabic (CA) and Modern Standard Arabic (MSA). The spoken language is characterised as a number of dialects used in everyday communication as well as informal writing. In this paper, we highlight the lexical relation between MSA and Dialectal Arabic (DA) in more than one Arabic region. We conduct a computational cross dialectal lexical distance study to measure the similarities and differences between dialects and MSA. We exploit several methods from Natural Language Processing (NLP) and Information Retrieval (IR) like Vector Space Model (VSM), Latent Semantic Indexing (LSI) and Hellinger Distance (HD), and apply them on different Arabic dialectal corpora. We measure the overlap among all the dialects and compute the frequencies of the most frequent words in every dialect. The results are informative and indicate that Levantine dialects are very similar to each other and furthermore, that Palestinian appears to be the closest to MSA.

**Keywords:** Diglossia; Lexical Distance; Vector Space Model; Latent Semantic Indexing; Hellinger Distance

## 9.1 Introduction

The number of the native Arabic speakers in the world varies from 290 million according to UNESCO[1] to 313 million, according to the Ethnologue[2]. There are three varieties in Arabic language: Classical Arabic, Modern Standard Arabic (MSA), and Arabic dialects (Colloquialism). Classical Arabic (CA) is the form of the Arabic language used in Umayyad and Abbasid literary texts from the 7th century AD to the 9th century AD. The orthography of the Quran was not developed for the standardized form of Classical Arabic [1]. MSA is the official language used for education, news, politics, religion and, in general, in any type of formal setting. Colloquialisms (dialects) are used in everyday communication as well as informal writing, e.g. in social media [2].

As a result of this situation, diglossia, a case where two distinct varieties of a language are spoken within the same speech community [3], is a very common phenomenon in Arabic-speaking communities. In some parts of the Arab speaking world, more than two varieties are spoken within the same community. For example, this is the case in North African communities like Morocco where Arabic, Berber, French, English and Spanish are spoken within the same speech community [4]. In a diglossic situation, the standard formal language assumes the role of the High variety (H), while the other languages or dialects act as the Low variety (L) [5]. MSA is so different from the colloquial dialects that they are in some cases not mutually intelligible. The differences are clearly evident in all linguistic aspects: pronunciation, phonology, morphology, lexicon, syntax and semantics. However, the degree in which the individual dialects differ with respect to these aspects has not been yet quantitatively measured.

In this paper, we focus on measuring the lexical distance between MSA and Arabic dialects using natural language processing techniques, tools and text corpora. We use various distance metrics such as the Vector space model (VSM) based on word distribution over documents as common in Information Retrieval (IR) [6], Latent semantic indexing (LSI) [7] and the Divergence Distance algorithm as Hellinger Distance (HD) [8]. We hope that this study will shed light on similarities and differences between the varieties and therefore inform our future work on building NLP tools and applications for these domains, in particular how these can be ported.

To the best of our knowledge, our work is the most extensive effort to measure the distance or similarity across Arabic dialects using natural

[1] https://en.unesco.org/news/world-arabic-language-day-2017-looking-digital-world

[2] Simons, Gary F. and Charles D. Fennig (eds.). 2018. Ethnologue: Languages of the World, Twenty-first edition. Dallas, Texas: SIL International. Online version: http://www.ethnologue.com.

| Approach Level | Approach Name | Description |
|---|---|---|
| **Character Level** | Longest Common SubString | Measures the length of the longest contiguous sequence of characters existing in the string under comparison |
| | Levenshtein distance | Measure the minimum number of insertions, deletions and substitutions needed to transform one string into another |
| | N-gram models | Can be used in different ways to estimate similarity or dissimilarity. One of the most effective approaches is to build n-gram models for language identification and measure the perplexity of n-grams |
| | Dynamic programming | Used for biological sequence comparison; e.g. the Needleman-Wunsch and Smith-Waterman algorithms |
| **Term Level** | Vector space models | Represent documents as vectors of word frequencies and then apply vector comparison measures to compare vectors of different documents |
| | Cosine Similarity | Measures the cosine angle as a similarity indicator between two vector spaces |
| | Divergence Distance | For example, the Kullback-Leibler distance, Hellinger, Manhattan distance. These are used to measure the divergence between probability distributions |
| | Jacquard similarity | Measures the number of overlapping strings over the number of unique strings between texts to indicate the similarity |
| | Latent Semantic Indexing | Words that are close in meaning will occur frequently in similar positions in the text |

Table 9.1: Summary of the most commonly used approaches for measuring similarity between texts

language processing tools and text corpora.

## 9.2 Related Work

Several approaches have been used to measure the distance between European languages [9, 10], Indian dialects [11], and similar languages [12, 13]. These approaches can be classified according to the type of linguistic representations they investigate: characters, terms and documents. Lexical similarity measures operate on string sequences at a character level and on corpora of texts at a term level. Table 9.1 shows the most popular approaches in the literature.

There is not much research on measuring the lexical closeness and divergence between Arabic and its dialects. Abunasser in [14] compares five Arabic dialects (MSA, Gulf, Levantine, Egyptian and Morocco) in terms of lexical and pronunciation variation. He depends on the Swadesh list [15] and the concept of non-cognate words to measure the amount of linguistic variations between the dialects. As the Swadesh list is a phonological list rather than a lexicon, the author collected the data from two male speakers for each dialect. The Swadesh list has been adapted to the MSA list using two modern Arabic dictionaries (المورد *ālmwrd* [16]andقاموس ابن اياس *qāmws ābn āyās* [17]). To rule out the chance of lexical ambiguity, a context sentence per each lexical item has been provided. Thus, the distance between dialects is measured based on the percentage of non-cognates in the MSA Swadesh list. Moreover, he employs Levenshtein distance to compute the distance between lexical items at the phonemic level based on the IPA transcription of the words in the Swadesh list. He concludes that Gulf and Levantine are the closet dialects to MSA followed by Egyptian, while Morocco is the farthest. The most significant limitation of this experiment is how the data were collected where speakers, gender and the geographical location were limited to two male speakers per dialect only. Also, the two modern dictionaries that are used to translate the Swadesh list to the corresponding MSA list have been authored by Levantine authors which might bias MSA to Levantine to some degree. Finally, with the intention to measure lexical variation, the study uses phonemic representation of words which may also reveal other more subtle non-lexical differences.

Meftouh et al. [18] present PADIC (Parallel Arabic Dialect Corpus). It includes five dialects: two Algerian (from the cities of Algeria and Annaba), one Tunisian and two Levantine dialects (Palestinian and Syrian). The authors present a linguistic analytical study of PADIC where they employ experiments on every pair of dialect and MSA, including:

- identifying the most frequent words in each dialect;
- computing the percentage of common lexical units both at the document and the sentence level to emphasize the relation between the dialects and the MSA; and
- measuring the cross language divergence in terms of the Hellinger distance to measure which language is closer to which one.

The experiments have shown that the Palestinian dialect is the closest dialect to MSA followed by Tunisian and Syrian, whereas Algerian dialects are the most different. The results are expected, as they demonstrate that Tunisian is closer to Algerian than to Palestinian and Syrian. In addition, the closest dialects according to the distance measurements are Algerian dialects on one hand and Palestinian and Syrian on the other hand. Even though the results are reasonable, the corpus has a shortcoming that it has been manually translated from Algerian conversations to MSA and further to other dialects by one native speaker of each dialect which introduces several biases.

Rama et al; [12] present a computational classification of the Gondi dialects which are spoken in central India by applying tools from dialectometry and phylogenetics. They use multilingual word lists for 210 concepts at 46 sites where Gondi is the dominate dialect. They depend on the Glottolog classification as a gold standard to evaluate their results. To be able to compute the aggregate distances, they employ the IPA to convert the word lists to pronunciation data. Levenshtein distance and Long-Short Term Memory neural networks are used as dialectometry methods to measure the distance between every word pair of words on the list. Moreover, they also apply Bayesian analysis on cognate analysis as a phylogentic method. They find that phylogentic methods perform best when compared to the gold standard classification.

Ruette et al; [19] measure the distance between Belgian and Netherlandic Dutch using two similarity measures in the Vector Space Model (VSM). They apply the two methods on a Dutch corpus collected from two registers (quality newspapers and Usenet) and topics related to politics and economy. They also exploit the profile-based approach (where the frequency of preselected words is compared from speakers' data) in addition to the text categorization method. For the profile based approach they implement the City-Block distance as a straightforward descriptive distance measure. On the other hand, text categorisation is using TFxIDF on documents and cosine similarity to measure distance as the complement of cosine similarity.

## 9.3 Qualitative differences between MSA and DA

Arabic is characterized by its rich morphology and vocabulary. For instance the Arabic word وسيعطيك *wsyṭyk* means "and he will give you" in English so one word in Arabic may correspond to 5 words in English [20] and that would make the comparison between languages/dialects challenging. This is true for both MSA as well as dialectal Arabic. However, MSA and DA have a number of differences that make it difficult for one to apply state of the art MSA natural language processing tools to DA. Previous attempts to do so have resulted in very low performance due to the significant difference between the varieties. Habash and Rambow report that over one third of Levantine verbs cannot be analysed using an MSA morphological analyser[21]. The degree of variation between MSA and dialectal Arabic depends on the specific dialect of Arabic. MSA and dialectal Arabic differ to a different degree phonologically, orthographically, morphologically, syntactically, lexically and semantically [22, 23]. In this section we describe some qualitative differences between MSA and the dialects based on our observation of examples.

### 9.3.1 Orthographical and Phonological Differences

Dialectal Arabic (DA) does not have an established standard orthography like MSA. Mostly, Arabic script is used to write DA but in some cases, e.g. in Lebanese, the Latin alphabet is used for writing short messages or posting on social media. For example, كيفك *kyfk* / "how are you" is represented as *Keifk*. Another example is the pronunciation of dialectal words containing the letter ق *q* which depends on the dialect and regions. For instance, the Palestinian speakers from rural and urban regions pronounce it like /’/ glottal stop or /k/ while Bedouin pronounce it as /g/ . The word قال *qāl* /say is pronounced and sometimes written as قال *qāl* , كال *kāl* , ئال *yāl* or جال *ǧāl* [24].

### 9.3.2 Morphological Differences

Dialects, like MSA and other Semitic languages, make extensive use of particular morphological patterns in addition to a large set of affixes (prefixes, suffixes, or infixes) and clitics, and therefore there are some important differences between MSA and dialectal Arabic in terms of morphology because of the way of using these clitics, particles and affixes [25]. Some examples are illustrated in Table 9.2 and 9.3.

| **Example** | Dialect word | Dialect | MSA | **English** |
|---|---|---|---|---|
| **Using multiple words together** | كيفك *kyfk* | Levantine | كيف حالك *kyf ḥālk* | How are you? |
| | معلش *mlš* | Egyptian | لا يهم *lā yhm* | Does not matter |
| **Sharing the stem with different affixes** | بدرسش *bdrsš* | Palestinian | لايدرس *lāydrs* | He does not study |
| | ما بيدرس *mā bydrs* | Syrian | | |
| | ميدرسش *mbydrsš* | Egyptian | | |
| **The future marker** | ح، راح *ḥ, rāḥ* | Palestinian | سوف *swf* | will |
| | حيلعب *ḥylb* | | سوف يلعب *swf ylb* | He will play |
| | راح يلعب *rāḥ ylb* | | | |
| **Clitics** | ب *b* for present | Egyptian | يأكل *yakl* | He is eating |
| | بياكل *byākl* | | | |
| | عم بطبخ *m bṭbẖ* | Syrain | < > | I am cooking |

Table 9.2: Examples for Morphological differences

| MSA | English | Negation | English |
|---|---|---|---|
| أعرف *arf* | I know | لا أعرف *lā arf* | I do not know |
| Palestinian | Jordanian | Syrian | Lebanese |
| مش عارف *mš ārf* | مش عارف *mš ārf* | ما بعرِف *mā brif* | ما بعرِف *mā brif* |
| Egyptian | Algerian | | Tunisian |
| معرفش *mrfš* | مش نعرف *mš nrf* | ملبعاليش *mlbālyš* | منيش عارف *mnyš ārf* |
| Gulf | Iraqi | | |
| مدري *mdry* | ما أدري *mā adry* | | |

Table 9.3: Differences in negation between the dialects

### 9.3.3 Syntactic Differences

Syntactically, MSA and DA are very similar with some differences regarding word order. For example, the OVS and OSV word orders are most commonly used in MSA while in dialects other word order patterns can be found. For example, in Levantine SVO is most commonly used, while in Maghrebi VSO is used to a great extent [26]. Furthermore, in dialectal Arabic it is common to use masculine plural or singular forms instead of dual and feminine plural forms [27].

### 9.3.4 Lexical and Semantic differences

Many DA words are borrowed from a variety of other languages like Turkish, French, English, Hebrew, Persian and others depending on the speaker contact with these languages. Table 9.4 shows some of the borrowed words. New lexical items appear mostly in dialects and not MSA as shown by the example in in Table 9.5. Another thing to note is dialects and MSA share words but with different meanings. For example, the word دول *dwl* means 'these' in Egyptian but “countries” in MSA.

| Word | Original | MSA | English | Word | Original | MSA | English |
|---|---|---|---|---|---|---|---|
| طربيزة *ṭrbyzh* | Turkish | طاولة *ṭāwlh* | Table | بندورة *bndwrh* | Italian | طماطم *ṭmāṭm* | Tomatoes |
| أستاذ *astāḏ* | Persion | مدرس *mdrs* | Teacher | توف *twf* | Hebrew | جيد *ǧyd* | Good |
| أفوكادو *afwkādw* | French | محامي *mḥāmy* | lawyer | تليفون *tlyfwn* | English | هاتف *hātf* | Telephone |

Table 9.4: Examples of borrow words from other languages

| MSA | English | | |
|---|---|---|---|
| الآن *ālān* | Now | | |
| Levantine | Bedouin | Saudi Arabia | Iraqi |
| هلأ، هلقيت *hla, hlqyt* | هلحين *hlḥyn* | دحين *dḥyn* | هالوقت *hālwqt* |
| Libyan | Tunisian | Algerian | Egyptian |
| توا *twā* | توة *twh* | توا *twā* | دلوقتي، دلوقت *dlwqty, dlwqt* |

Table 9.5: Examples for new lexicon in dialects

## 9.4 Quantitative differences between MSA and DA

### 9.4.1 Arabic Corpora

Ferguson [3] was the first to define the term diglossia. He stated and defined the most important features in order to understand the difference between the official languages (H) and the informal varieties (L). One of these features is the lexicon. In his own words: *"A striking feature of Diglossia is the existence of many paired items, one H and one L, referring to fairly common concepts frequently used in both H and L, where the range of meaning of the two items is roughly the same, and the use of one or the other immediately stamps the utterance or written sequence as H or L"*.

In this work, we examine several existing Arabic corpora, so that we can include as many dialectal data as we can. Table 9.6 shows the corpora we use and the dialectal data they contain. Table 9.7 shows the statistics about each corpus where |d| is the number of documents (sentences) in the corpus, |w| is the number of words in the corpus, and |v| is the vocabulary size (number of unique words).

| Corpus Name | Type | Dialects | Description |
|---|---|---|---|
| PADIC<br>(Parallel Arabic Dialect Corpus) | Parallel | MSA, Algerian,<br>Tunisian, Palestinian, Syrian | The corpus is collected from Algerian chats and conversations which are translated to MSA and then to other dialects. |
| Multi-dialectal Arabic<br>parallel corpus | Parallel | MSA, Egyptian, Syrian,<br>Palestinian, Tunisian, Jordanian | This corpus is originally build on Egyptian dialects extracted from Egyptian-English corpus. It has been translated to the remaining dialects by four translators |
| SDC<br>(Shami Dialect Corpus) | Non-parallel | Palestinian, Syrian,<br>Jordanian, Lebanese | The corpus is collected from different sources of social media, blogs, stories and public figures on the Internet. |
| WikiDocs Corpus | Comparable | MSA, Egyptian | It contains a comparable documents from Wikipedia. |

Table 9.6: List of Arabic corpora used to investigate the differences between dialects

The two Algerian dialects are the basis of the Parallel Arabic Dialect Corpus (PADIC), that was collected from daily conversations, movies and TV shows were presented in Annaba and Algeria dialects. The two corpora were transcribed by hand and then translated to MSA. Hence, the MSA is considered the pivot language to construct the Syrian, Palestinian and Tunisian dialects. They adopt Arabic notation to write dialectal words. If the dialectal word does exist in MSA, it is written as MSA without any change, otherwise, it is written as it is uttered. Some consider these rules as drawbacks of the corpus which bias the dialect to the MSA and the translated sentence is subjected to the annotators [18]. The corpus is not considered fully representative for every dialect due to the lack of translators particularly for the Levantine dialects where only 2 translators are involved while for Tunisian they had 20 speakers all of them from the South of Tunisia

| | PADIC | | | | | | SDC | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MSA | PA | AlG | SY | TN | | PA | JO | SY | LB |
| $\|d\|$ | 6.4K | 6.4K | 6.4K | 6.4K | 6.4K | | 21K | 32K | 48K | 16K |
| $\|w\|$ | 51K | 51K | 48K | 49K | 48K | | 0.35M | 0.47M | 0.7M | 0.2M |
| $\|v\|$ | 9.4K | 9.6K | 9.4K | 10K | 10.6K | | 56K | 69K | 63K | 34K |
| | Multi-dialect corpus | | | | | | WikiDocs corpus | | | |
| | MSA | PA | JO | SY | TN | EG | MSA | | EG | |
| $\|d\|$ | 1 K | 1 K | 1 K | 1 K | 1 K | 1K | 459K | | 16K | |
| $\|w\|$ | 11.9K | 10.5K | 9.7K | 11.5K | 10.6K | 10.9K | 83.5M | | 2.18M | |
| $\|v\|$ | 4.4K | 4K | 3.6K | 4K | 3.8K | 4.5K | 4.7M | | 293.5K | |

Table 9.7: Statistics about the used corpora

where their dialect is close to the Standard Arabic.

The Multi-dialectal Arabic parallel corpus is built on the English-Egyptian corpus [28], where the Egyptian sentences have been selected as the starting point for the new parallel corpus. Five translators, one for every dialect, were asked to translate the Egyptian corpus to Palestinian, Jordanian, Syrian and Tunisian dialects, while the Egyptian speakers translated the corpus to corresponding MSA [29]. Using the Egyptian sentences as the pivot dialect makes the corpus heavily influenced and biased by the Egyptian dialects, which is clearly shown in our results in the following sections.

The WikiDocs corpus is extracted from Arabic Wikipedia articles and their corresponding Egyptian Wikipedia articles [30]. It should be noted that a lot of the Egyptian articles are not detailed, as most of these only contain one or two sentences. This is in contrast to the MSA articles, which contain full details on each subject. The Shami Dialect Corpus (SDC) corpus is collected from different domains like social life, sports, house work, cooking, etc. and from resources such as personal blogs, social media public figures posts and stories written in DA. It focuses on public figures from Levantine countries. It is not a parallel corpus, thus the measures are done over the whole corpus and not on every document [23].

In what follows, we exploit various approaches to the lexicon to precisely clarify the difference between MSA and other Arabic dialects in term of lexical distance. The type of corpora affects the way we implement each measure as follows:

- for parallel and comparable corpora: the comparison is at the document (sentence) level, then the average is taken at a corpus level;

- for non-parallel and non-comparable corpora: the comparison is at the corpus level, given that the data belong to the same domain.

In all experiments we have used Python as a programming language to implement the algorithms and used the Gensim library for some methods.

As the corpora are already preprocessed, we did not do any further preprocessing.[3] In the next subsections, we present the measures what we use in our experiments.

### 9.4.2 Lexical Sharing and Overlapping

Jaccard Index is a measure of how similar two data sets are. Given that dialects share many words, we compute the percentage of vocabularies that overlap between these dialects according to Equation 9.1. Table 9.8 presents the similarity overlap across dialects. Palestinian is the most similar to MSA, that coming after the Egyptian dialect, with the highest percentage of vocabulary overlap in both parallel corpora. The measurement on the SDC shows a reasonable overlapping across the Levantine dialects, while in the comparable corpus the overlapping between the MSA and the Egyptian does not exceed the 0.1.

$$JaccardIndex(A, B) = \frac{|A \bigcap B|}{|A \bigcup B|} \tag{9.1}$$

| | **PADIC** | | | | **Multi-dialect corpus** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ALG | TN | SY | PA | | EG | JO | TN | SY | PA |
| MSA | 0.1 | 0.14 | 0.14 | 0.19 | MSA | 0.21 | 0.14 | 0.13 | 0.15 | 0.16 |
| PA | 0.13 | 0.14 | 0.25 | | PA | 0.23 | 0.25 | 0.18 | 0.24 | |
| SY | 0.12 | 0.16 | | | SY | 0.23 | 0.26 | 0.18 | | |
| TN | 0.17 | | | | TN | 0.18 | 0.18 | | | |
| | | | | | JO | 0.21 | | | | |
| | **SDC** | | | | **WikiDocs corpus** | | | | | |
| | LB | JO | SY | | | | EG | | | |
| PA | 0.15 | 0.21 | 0.19 | | MSA | | 0.1 | | | |
| SY | 0.16 | 0.2 | | | | | | | | |
| JO | 0.16 | | | | | | | | | |

Table 9.8: Percentage of vocabulary overlapping between dialects

### 9.4.3 Vector Space Model (VSM)

VSM is broken down into three steps. First, document indexing where each document is represented by the content bearing words which, in turn, are represented as a document-terms vector. VSM represents all documents as

[3] It is possible that the preprocessing techniques that have been used on different corpora might affect their comparison, which is an unfortunate limitation of our approach in terms of the implications for language use in general.

vectors in a high dimensional space in which each dimension of the space corresponds to a term in the document collection [7]. Secondly, term weighting where a weighting schema is used to compute the term weightings for each term in the represented vector (document). The most common weighting schema is to employ the frequency of occurrence expressed as a ration between frequency and inverse document frequency (tf-idf). A similarity coefficient is then computed between each pair of vectors to indicate a ranking of documents [31].

We utilize the VSM to measure the similarity across dialects and MSA by comparing the similarity between the terms in their documents or sentences. Clearly, not all words in a dialect or a document are equally important. Most current approaches remove all the stop words during the preprocessing phase. However, we have decided to index all words as many of the stop words act like function words and therefore are distinguishing of certain dialects. In order to overcome the out-of-dictionary problem we build a vector for each pair of dialects. Therefore, for the first dialect (MSA), we draw a vector model and employ the tf-idf weighting schema. The second dialect is considered as the query vector compared to the first dialect. Spatial closeness corresponds to conceptual similarity (words that are used in the same documents are similar) so we measure the cosine similarity between the main vector model (first-dialect) and the query vector (second-dialect) (what a vector represents in each case depends on the kind of corpora we are comparing as explained above) which is a symmetric measurement. Table 9.9 present the similarity across dialects for all corpora.

| | **PADIC** | | | | **Multi-dialect corpus** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ALG | TN | SY | PA | | EG | JO | TN | SY | PA |
| MSA | 0.27 | 0.38 | 0.37 | 0.5 | MSA | 0.5 | 0.38 | 0.37 | 0.4 | 0.4 |
| PA | 0.38 | 0.47 | 0.63 | | PA | 0.59 | 0.66 | 0.48 | 0.62 | |
| SY | 0.34 | 0.41 | | | SY | 0.63 | 0.7 | 0.5 | | |
| TN | 0.44 | | | | TN | 0.49 | 0.47 | | | |
| | | | | | JO | 0.56 | | | | |
| | **SDC** | | | | **WikiDocs corpus** | | | | | |
| | LB | JO | SY | | | EG | | | | |
| PA | 0.84 | 0.86 | 0.77 | | MSA | 0.4 | | | | |
| SY | 0.81 | 0.9 | | | | | | | | |
| JO | 0.84 | | | | | | | | | |

Table 9.9: Similarity across dialects for all corpora based on VSM

The results show that the Palestinian dialects in both the PADIC and the Multi dialect corpus are closer to MSA, with 0.5 and 0.4 similarity respectively, while the Tunisian and Algerian dialects are furthest from MSA.

Moreover, on SDC we can demonstrate a high similarity between individual Levantine dialects. For example Jordanian is the closest to Palestinian, which seems to coincide with informal observations by native speakers of both dialects.

It is worth mentioning that the Egyptian dialect records the highest relation with MSA in Multi-dialect corpus, as we previously expected. The corpus is biased towards the Egyptian dialect, as Egyptian was the pivot language when the corpus was built. This is reflected in all the measures used here. However, the bias of the pivot language is not reflected between Algerian and MSA in the PADIC corpus as these are the least similar varieties.

### 9.4.4 Latent Semantic Indexing LSI

Unlike VSM and other retrieval methods, LSI can address the problem of synonymy and polysemy among words. It analyzes the documents in order to represent the concepts they contain. LSI tries to map the vector space into a new compressed space by reducing the dimensions of the terms matrix using Singular Value Decomposition (SVD). By using SVD, the main associative patterns and trends are extracted from the document space and the noise is ignored. In other words, it makes the best possible reconstruction of the document matrix with the most valuable information [7]. We exploit the LSI model to measure the similarity between the dialects. We build the model with all the dialects (full corpus) and test it on one dialect in each run. The model outputs the similarity between the test dialect and every dialect used to build the model. Table 9.10 shows the similarities among the Arabic dialects for all corpora.

Palestinian appears to be close to MSA only in PADIC, whereas the Tunisian dialect shows a close relation to MSA in both corpora. In addition to this, it is obvious that the relation between the dialects in the Levantine corpus (SDC) is very strong as well as the relation between the Algerian and Tunisian. These results show the artefacts of the LSI model which connects the data according to topics and clusters.

### 9.4.5 Hellinger Distance

We are interested to measure the divergence between the dialects. Here we will use the Hellinger Distance (HD) that measures the difference between two probability distributions [18]. In this work we use Latent Dirichlet Allocation (LDA) to model a vector of discrete probability distributions of topics to measure the distance between dialects in comparison. LDA is a very common technique used to uncover topics in the data [32]. For

| | **PADIC** | | | | **Multi-dialect corpus** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ALG | TN | SY | PA | | EG | JO | TN | SY | PA |
| MSA | 0.68 | 0.75 | 0.69 | 0.75 | MSA | 0.72 | 0.37 | 0.75 | 0.4 | 0.41 |
| PA | 0.78 | 0.82 | 0.85 | | PA | 0.82 | 0.88 | 0.63 | 0.9 | |
| SY | 0.74 | 0.74 | | | SY | 0.7 | 0.94 | 0.59 | | |
| TN | 0.82 | | | | TN | 0.74 | 0.55 | | | |
| | | | | | JO | 0.73 | | | | |
| | **SDC** | | | | **WikiDocs corpus** | | | | | |
| | LB | JO | SY | | | EG | | | | |
| PA | 0.84 | 0.86 | 0.77 | | MSA | 0.8 | | | | |
| SY | 0.81 | 0.9 | | | | | | | | |
| JO | 0.84 | | | | | | | | | |

Table 9.10: Similarity across dialects for all corpora based on LSI

simplicity, a Bag Of Words (BOW) model is used to represent the data from our corpora. LDA gives us a probability distribution over a specified number of unknown topics. LDA therefore works like a way of soft clustering the documents made up of words. Later HD is then used to measure the distance between these topics and new documents. The greater the distance the less the similarity between the dialects and vice versa.

Table 9.11 shows the distance between the dialects cross all corpora. Palestinian is less dissimilar from MSA compared to the rest of the dialects in PADIC. Even though in the Multi-dialect corpus the results for the distance of all dialects, except of the Egyptian, to MSA is quite close, the Tunisian seems to be the closest to MSA. Considering that the Levantine dialects in SDC are very similar to each other, the Jordanian and the Syrian dialects are the closest to each other, while the Palestinian and the Lebanese dialects are most dissimilar.

### 9.4.6 Frequent words and Correlation Coefficient

This step consists of two parts. At first, we extract the 30 most frequent words in each dialect and then we collect those words that appear in all dialects to calculate the Pearson correlation coefficient among them in respect to their frequency as shown in Table 9.12.

The result shows high correlation for the frequent words between the MSA and Tunisian, followed by the Palestinian dialects in PADIC. This sheds the light on the different usage of frequent words cross dialects. For example Palestinian speakers say في المدرسة *fy ālmdrsh* / "at the school" while the Syrian speakers say بالمدرسة *bālmdrsh*.

For the words that are not shared and have not been included in the

| | **PADIC** | | | | **Multi-dialect corpus** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ALG | TN | SY | PA | | EG | JO | TN | SY | PA |
| MSA | 0.91 | 0.83 | 0.77 | 0.77 | MSA | 0.01 | 0.77 | 0.76 | 0.78 | 0.78 |
| PA | 0.73 | 0.64 | 0.58 | | PA | 0.52 | 0.34 | 0.77 | 0.55 | |
| SY | 0.87 | 0.81 | | | SY | 0.53 | 0.54 | 0.72 | | |
| TN | 0.72 | | | | TN | 0.35 | 0.69 | | | |
| | | | | | JO | 0.51 | | | | |
| | **SDC** | | | | **WikiDocs corpus** | | | | | |
| | LB | JO | SY | | | EG | | | | |
| PA | 0.26 | 0.18 | 0.23 | | MSA | 0.73 | | | | |
| SY | 0.25 | 0.1 | | | | | | | | |
| JO | 0.2 | | | | | | | | | |

Table 9.11: Distance between dialects for all corpora based on Hellinger Distance

| | **PADIC** | | | | **SDC** | | | |
|---|---|---|---|---|---|---|---|---|
| | ALG | TN | SY | PA | | LB | JO | SY |
| MSA | 0.76 | 0.92 | 0.67 | 0.85 | PA | 0.31 | 0.42 | -0.05 |
| PA | 0.97 | 0.95 | 0.86 | | SY | 0.13 | 0.74 | |
| SY | 0.83 | 0.71 | | | JO | 0.47 | | |
| TN | 0.92 | | | | | | | |

Table 9.12: The Person correlation coefficient between dialects in PADIC and SDC

correlation experiment, we have calculated the Term Frequency (TF) as in Equation 9.2.

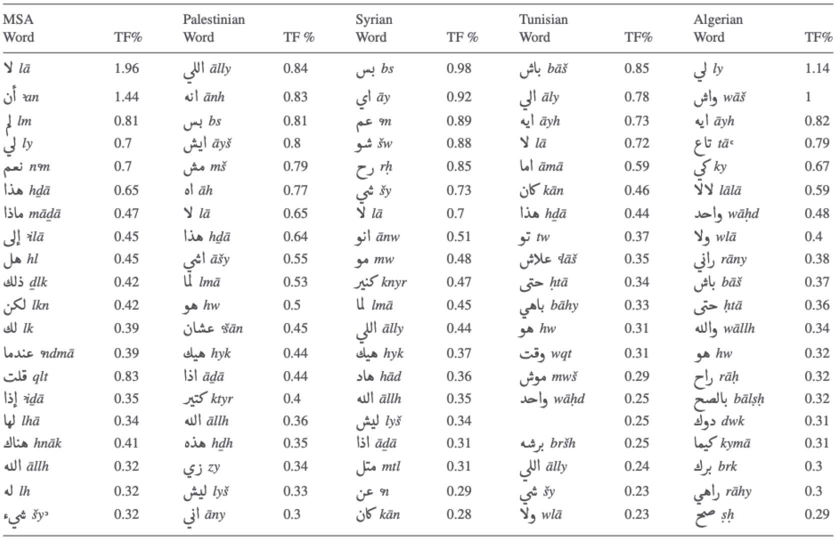
$$TF(t) = \frac{(Number\ of\ times\ term\ t\ appears\ in\ a\ dialect)}{(Total\ number\ of\ terms\ in\ the\ dialect)}. \tag{9.2}$$

As we have already mentioned, we have not eliminated stop words from the corpora as these keywords are discriminative and representative for each dialect and hence can be used to build a dialectal lexicon. Table 9.13shows the 20 most frequent words in PADIC[4].

## 9.5 Conclusion

In this paper, we estimate the degree of similarity and dissimilarity between MSA and DA on one hand, and across dialects of Arabic on the other.

[4] The full tables for all corpora can be found in https://github.com/GU-CLASP/DAdistance



| MSA<br>Word | TF% | Palestinian<br>Word | TF % | Syrian<br>Word | TF % | Tunisian<br>Word | TF% | Algerian<br>Word | TF% |
|---|---|---|---|---|---|---|---|---|---|
| لا *lā* | 1.96 | اللي *ālly* | 0.84 | بس *bs* | 0.98 | باش *bāš* | 0.85 | لي *ly* | 1.14 |
| أن *ʔan* | 1.44 | انه *ānh* | 0.83 | اي *āy* | 0.92 | الي *āly* | 0.78 | واش *wāš* | 1 |
| لم *lm* | 0.81 | بس *bs* | 0.81 | عم *ʕm* | 0.89 | ايه *āyh* | 0.73 | ايه *āyh* | 0.82 |
| لي *ly* | 0.7 | ايش *āyš* | 0.8 | شو *šw* | 0.88 | لا *lā* | 0.72 | تاع *tāʿ* | 0.79 |
| نعم *nʕm* | 0.7 | مش *mš* | 0.79 | رح *rḥ* | 0.85 | اما *āmā* | 0.59 | كي *ky* | 0.67 |
| هذا *hḏā* | 0.65 | اه *āh* | 0.77 | شي *šy* | 0.73 | كان *kān* | 0.46 | لالا *lālā* | 0.59 |
| ماذا *māḏā* | 0.47 | لا *lā* | 0.65 | لا *lā* | 0.7 | هذا *hḏā* | 0.44 | واحد *wāḥd* | 0.48 |
| إلى *ʔilā* | 0.45 | هذا *hḏā* | 0.64 | انو *ānw* | 0.51 | تو *tw* | 0.37 | ولا *wlā* | 0.4 |
| هل *hl* | 0.45 | اشي *āšy* | 0.55 | مو *mw* | 0.48 | علاش *ʕlāš* | 0.35 | راني *rāny* | 0.38 |
| ذلك *ḏlk* | 0.42 | لّما *lmā* | 0.53 | كنير *knyr* | 0.47 | حتى *ḥtā* | 0.34 | باش *bāš* | 0.37 |
| لكن *lkn* | 0.42 | هو *hw* | 0.5 | لما *lmā* | 0.45 | باهي *bāhy* | 0.33 | حتى *ḥtā* | 0.36 |
| لك *lk* | 0.39 | عشان *ʕšān* | 0.45 | اللي *ālly* | 0.44 | هو *hw* | 0.31 | والله *wāllh* | 0.34 |
| عندما *ʕndmā* | 0.39 | هيك *hyk* | 0.44 | هيك *hyk* | 0.37 | وقت *wqt* | 0.31 | هو *hw* | 0.32 |
| قلت *qlt* | 0.83 | اذا *āḏā* | 0.44 | هاد *hād* | 0.36 | موش *mwš* | 0.29 | راح *rāḥ* | 0.32 |
| إذا *ʔiḏā* | 0.35 | كتير *ktyr* | 0.4 | الله *āllh* | 0.35 | واحد *wāḥd* | 0.25 | بالصح *bālṣḥ* | 0.32 |
| لها *lhā* | 0.34 | الله *āllh* | 0.36 | ليش *lyš* | 0.34 |  | 0.25 | دوك *dwk* | 0.31 |
| هناك *hnāk* | 0.41 | هذه *hḏh* | 0.35 | اذا *āḏā* | 0.31 | برشه *bršh* | 0.25 | كيما *kymā* | 0.31 |
| الله *āllh* | 0.32 | زي *zy* | 0.34 | متل *mtl* | 0.31 | اللي *ālly* | 0.24 | برك *brk* | 0.3 |
| له *lh* | 0.32 | ليش *lyš* | 0.33 | عن *ʕn* | 0.29 | شي *šy* | 0.23 | راهي *rāhy* | 0.3 |
| شيء *šyʾ* | 0.32 | اني *āny* | 0.3 | كان *kān* | 0.28 | ولا *wlā* | 0.23 | صح *ṣḥ* | 0.29 |

Table 9.13: The percentage of the most frequent words in PADIC

Different measures have been exploited, such as VSM, LSI, HD as well as simple measures like vocabulary overlap, coefficient correlation and Jaccard similarity. More than one corpus has been used. In particular, PADIC, the Multi-Dialect corpus, SDC and Wiki-Docs were used, that include MSA, Levantine dialects, Egyptian and Dialects from North Africa. This was done in order to minimise the bias of any of the individual corpora and to address the question of the degree of the text representativeness. Most of the measurements used indicate that the Levantine dialects are in general the closet to MSA, while the North African dialects the farthest. Although the results show some differences due to the nature of the corpora, in general, the results are homogeneous. For example, it is expected that the Egyptian dialects appear very close to MSA in the Multi-Dialect corpus. This is, as mentioned earlier, due to a strong bias of the specific corpus towards the Egyptian dialect, given that it was built from an Egyptian corpus and then translated into other dialects and MSA.

We have shown the degree of convergence between the dialects of the Levant and the linguistic overlap to such an extent that in some cases it seems impossible to distinguish between them in writing without the presence of phonological information or without adding accent diacritic marks.

It is very clear that we have a new variety, i.e. an informal writing dialect, which differs from the spoken dialects. Even if some dialects appear close to each other based on the speakers' intuitions, there may be differences in the writing form due to the lack of accent diacritics. The reverse is also true. Some dialects appear closer lexically in their writing form given that a big part of their vocabulary overlaps, but in their spoken form, they are not that close.

This study can be seen as a basis for building Natural Language Processing tools for dialectal processing by adapting what already exists for MSA and focusing on areas of similarity and degrees of difference. The study is the most extensive of its kind concerned with measuring similarities and differences in Arabic and dialectal Arabic, and represents a basis for new similar investigations, focusing on other criteria such as phonological distance, morphological distance and semantic distance. In the future, we plan to employ other methods of measuring similarity and distance based on the semantics of the words, e.g. word embedding techniques with Word2Vec. In this way, one can extract different words in terms of their lexical relatedness, and use them in automatic machine translation tools for the languages and dialects investigated.

## Acknowledgements

Kathrein Abu Kwaik, Stergios Chatzikyriakidis and Simon Dobnik are supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg.

## 9.6 References

[1] Mustafa Shah. *The Arabic Language*. Routledge, 2008.

[2] Kees Versteegh. *The Arabic Language*. Edinburgh University Press, 2014.

[3] Charles A Ferguson. “Diglossia”. In: *word* 15.2 (1959), pp. 325–340.

[4] Abderrahman Zouhir. “Language Situation and Conflict in Morocco”. In: *Selected Proceedings of the 43rd Annual Conference on African Linguistics, ed. Olanike Ola Orie and Karen W. Sanders*. 2013, pp. 271–277.

[5] MJ Jabbari. “Diglossia in Arabic-a Comparative Study of The Modern Standard Arabic and Colloquial Egyptian Arabic”. In: *Global Journal of Human Social Sciences* 12.8 (2012), pp. 23–46.

[6] Stephen Clark. “Vector Space Models of Lexical Meaning”. In: *Handbook of Contemporary Semantics – second edition.* Ed. by Shalom Lappin and Chris Fox. Wiley – Blackwell, 2015. Chap. 16, pp. 493–522.

[7] Ch Aswani Kumar, M Radvansky, and J Annapurna. “Analysis of a Vector Space Model, Latent Semantic Indexing and Formal Concept Analysis for Information Retrieval”. In: *Cybernetics and Information Technologies* 12.1 (2012), pp. 34–48.

[8] VıCtor GonzáLez-Castro, RocıO Alaiz-RodrıGuez, and Enrique Alegre. “Class Distribution Estimation Based on The Hellinger Distance”. In: *Information Sciences* 218 (2013), pp. 146–164.

[9] Barry R Chiswick and Paul W Miller. “Linguistic Distance: A Quantitative Measure of the Distance Between English and Other Languages”. In: *Journal of Multilingual and Multicultural Development* 26.1 (2005), pp. 1–11.

[10] Wilbert Heeringa et al. “Lexical and Orthographic Distances between Germanic, Romance and Slavic Languages and their Relationship to Geographic Distance”. In: *Phonetics in Europe: Perception and Production* (2013), pp. 99–137.

[11] Debapriya Sengupta and Goutam Saha. “Study on Similarity Among Indian Languages using Language Verification Framework”. In: *Advances in Artificial Intelligence* 2015 (2015), p. 2.

[12] Taraka Rama, Çağrı Çöltekin, and Pavel Sofroniev. “Computational Analysis of Gondi dialects”. In: *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial).* 2017, pp. 26–35.

[13] Peter Houtzagers, John Nerbonne, and Jelena Prokić. “Quantitative and Traditional Classifications of Bulgarian Dialects Compared”. In: *Scando-Slavica* 56.2 (2010), pp. 163–188.

[14] Mahmoud Abedel Kader Abunasser. “Computational Measures of Linguistic Variation: A Study of Arabic Varieties”. PhD thesis. University of Illinois at Urbana-Champaign, 2015.

[15] Morris Swadesh. “Salish Internal Relationships”. In: *International Journal of American Linguistics* 16.4 (1950), pp. 157–167.

[16] Munir Baalbaki. “: _ ”. In: (1982).

[17] Elias Antoon Elias and Ed E Elias. “Elias’ Modern Dictionary, Arabic-English”. In: (1983).

[18] Salima Harrat et al. "Cross-Dialectal Arabic Processing". In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer. 2015, pp. 620–632.

[19] Tom Ruette, Dirk Speelman, and Dirk Geeraerts. "Measuring the Lexical Distance Between Registers in National Variaties of Dutch". In: (2011).

[20] Motaz Saad. "Fouille de Documents et D'opinions Multilingue". PhD thesis. Université de Lorraine, 2015.

[21] Nizar Habash and Owen Rambow. "MAGEAD: a Morphological Analyzer and Generator for the Arabic Dialects". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2006, pp. 681–688.

[22] Pradeep Dasigi and Mona T Diab. "CODACT: Towards Identifying Orthographic Variants in Dialectal Arabic." In: *IJCNLP*. 2011, pp. 318–326.

[23] Kathrein Abu Kwaik et al. "Shami: A Corpus of Levantine Arabic Dialects". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[24] Mustafa Jarrar et al. "Building a Corpus for Palestinian Arabic: A Preliminary Study". In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*. 2014, pp. 18–27.

[25] Nizar Habash, Mona T Diab, and Owen Rambow. "Conventional Orthography for Dialectal Arabic." In: *LREC*. 2012, pp. 711–718.

[26] Karima Meftouh et al. "Machine Translation Experiments on PADIC: A Parallel Arabic Dialect Corpus". In: *The 29th Pacific Asia conference on language, information and computation*. 2015.

[27] Kareem Darwish, Hassan Sajjad, and Hamdy Mubarak. "Verifiably Effective Arabic Dialect Identification." In: *EMNLP*. 2014, pp. 1465–1468.

[28] Rabih Zbib et al. "Machine Translation of Arabic Dialects". In: *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics. 2012, pp. 49–59.

[29] Houda Bouamor, Nizar Habash, and Kemal Oflazer. "A Multidialectal Parallel Corpus of Arabic." In: *LREC*. 2014, pp. 1240–1245.

[30] Motaz Saad and Basem O Alijla. “Wikidocsaligner: An Off-the-shelf Wikipedia Documents Alignment Tool”. In: *2017 Palestinian International Conference on Information and Communication Technology (PICICT)*. IEEE. 2017, pp. 34–39.

[31] Ray R Larson. “Introduction to Information Retrieval”. In: *Journal of the American Society for Information Science and Technology* 61.4 (2010), pp. 852–853.

[32] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent Dirichlet Allocation”. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

# 10

# Study 4: The usability of MSA NLP tools for DA

*Chatrine Qwaider, Stergios Chatzikyriakidis, and Simon Dobnik. "Can Modern Standard Arabic Approaches be used for Arabic Dialects? Sentiment Analysis as a Case Study." In Proceedings of the 3rd Workshop on Arabic Corpus Linguistics, pp. 40-50. 2019.*

We present the Shami-Senti corpus, the first Levantine corpus for Sentiment Analysis (SA), and investigate the usage of off-the-shelf models that have been built for Modern Standard Arabic (MSA) on this corpus of Dialectal Arabic (DA). We apply the models on DA data, showing that their accuracy does not exceed 60%. We then proceed to build our own models involving different feature combinations and machine learning methods for both MSA and DA and achieve an accuracy of 83% and 75% respectively.

## 10.1 Introduction

There is a growing need for text mining and analytical tools for Social Media data, for example Sentiment Analysis (SA) tools which aim to distinguish people's views into positive and negative, objective and subjective responses, or even into neutral opinions. The amount of internet documents in Arabic is increasing rapidly [1, 2, 3, 4]. However, texts from Social media are typically not written in Modern Standard Arabic (MSA) for which computational resources and corpora exist. These systems achieve reasonable accuracy on the designated tasks. For example, Abdul-Mageed et. al; achieve an accuracy of 95% on the news domain[2]. On the other hand, research on Dialectal Arabic (DA) in terms of SA is an open research question and presents considerable challenges [5, 1].

The degree to which tools trained on MSA can be used on DA is still also an open research question. This is partly because different dialects differ from MSA to varying degrees[6]. Furthermore, the speakers of Arabic present us with clear cases of Diglossia [7], where MSA is the official language used for education, news, politics, religion and, in general, in any type of formal setting, but dialects are used in everyday communication, as well as in informal writing [8].

In this paper, we examine whether it is possible to adapt classification models that have been trained and built on MSA data for DA from the Levantine region, or whether we should build and train specific models for the individual dialects, therefore considering them as stand-alone languages. To answer this question we use Sentiment Analysis as a case study. Our contributions are the following:

- We systematically evaluate how well the ML models on MSA for SA perform on DA of Levantine;
- We construct and present a new sentiment corpus of Levantine DA;
- We investigate the issue of domain adaptation of ML models from MSA to DA.

The paper is organised as follows: in Section 2, we briefly discuss the task of SA and present related work on Arabic. In Section 3, we describe an extension of the Shami corpus of Levantine dialects [9] annotated for Sentiment, Shami-Senti. In Section 4, we present the experimental setting and results of adapting MSA models to the dialectal domain as well as training specific models. We conclude and discuss directions for future research in Section 5.

## 10.2 Arabic Sentiment Analysis

Manually gathering information about users' opinions and sentiment data is time-consuming. This is why more and more companies and organisations are interested in automatic SA methods to help them understand it. SA refers to the usage of variety of tools from Natural Language Processing (NLP), Text Mining and Computational Linguistics to examine a given piece of text and identify the dominant sentiment subjectivity in it [10, 11]. SA is usually categorised into three main sentiment polarities: Positive (POS), Negative (NEG) and Neutral (NUT). SA is frequently used interchangeably with Opinion Mining [12].

At first glance, Sentiment Analysis is a classification task. It is a complex classification task as if one dives deeper, they are faced with a number of challenges that affect the accuracy of any SA model. Some of these challenges are: (i) Negation terms [13], (ii) Sarcasm [14], (iii) Word ambiguity and (iv) Multi-polarity.

As a result of the rapid development of social media and the use of Arabic dialectal texts, there is an emerging interest in DA. Farra et al., in [15] propose a model of sentence classification (SA) in Arabic documents. They extract sets of features and calculate the total weight for every sentence. A J48 Decision tree algorithm is used to classify the sentences w.r.t. sentiment, achieving an accuracy of 62%.

Gamal et al., collect tweets from different Arabic regions using different keywords and phrases[16]. The tweets include opinions about a variety of topics. They annotate their polarity by checking if they contain positive or negative terms and without considering the reverse polarity in the presence of negation terms. Then, they apply six machine learning algorithms on the data and achieve an accuracy between 82% and 93%.

Oussous et al., build an SA model to classify the sentiment of sentences. The authors construct a Moroccan corpus, where the data are collected from Twitter, and annotate it[17]. Multiple algorithms are used, e.g. Support Vector Machines (SVM), Multinomial Naïve Bayes (MNB) and Mean Entropy (ME). The SVM model achieves an accuracy of 85%. Ensemble learning by majority voting and stacking is also tried. Using the three aforementioned algorithms in the two models, they attain an accuracy of 83% and 84% respectively. Another work using the same classifiers is described in [18]. The dataset covers three domains: education, politics and sports. The resulting accuracy is 80%.

A framework for Jordanian SA is proposed in [19]. The authors create a corpus of Jordanian tweets and build a mapping lexicon from Jordanian to MSA that turns any dialectal word into an MSA word, before classifying the tweet. In order for the tweets to be annotated, crowd-sourcing is used. They further use Rapid Miner for pre-processing, filtering, and classification. Three classifiers are used to evaluate the performance of the proposed framework with 1000 tweets: Naïve Bayes (NB), (SVM) and k-nearest neighbour (KNN). The NB model gets the highest accuracy with 76.78%.

Binary sentiment classification for Egyptian using a NB classifier is investigated in [2]. An accuracy of 80% is achieved. Similarly, the Tunisian dialect is addressed in [20]. Here, the authors create a Tunisian corpus for SA containing 17K comments from social media. Applying Multi-Layer Perceptron (MLP) and SVM on the corpus they get 0.22 and 0.23 error rate respectively. Another line of work addresses the Saudi dialects [21, 22] and some addresses the United Arab Emirates dialects [23, 24].

Several works exploit lexicon-based sentiment classifiers for Arabic. A sentiment lexicon is a lexicon that contains both positive and negative terms along with their polarity weights [25, 26, 27]. The SAMAR system [28] involves two-stage classification based on a sentiment lexicon. The first classifier detects subjectivity and objectivity of documents, which is followed by another classifier to detect the polarity. They employ different datasets and examine various features combinations. Similar work is reported in [4, 29], where both NB and SVM are explored, achieving an accuracy between 73% and 84% .

Abdulla et. al; [30] compare the performance of corpus-based sentiment classification and lexicon-based classification in Arabic. The accuracy of the lexicon approach does not exceed 60%. They conclude that corpus-based methods perform better using SVM and light stemming.

Overall, there is a considerable amount of work on SA and DA but none of these approaches considered the performance of the classifiers across the domains for which limited data exist.

## 10.3 Building Shami-Senti

The question of sentiment analysis has not yet been fully examined for Levantine dialects: Palestinian, Jordanian, Syrian and Lebanese. For this reason, we extend the Shami corpus [9] by annotating part of it for sentiment. We call the new corpus Shami-Senti.

We build Shami-Senti as follows:

1. Manually extract sentences that contains sentiment words, reviews, opinions or feelings from the Shami corpus;
2. Split the sentences and remove any misleading words or very long phrases (set sentences be no longer than 50 words);
3. Try to avoid ironic and sarcastic text where the intended sentiment is reversed. For example, sentences like the following: تصدقوا احنا ناكرين الجميل *tṣdqwā āḥnā nākryn ālǧmyl* الرجال زي الفل *ālrǧāl zy ālfl* “I believe we are ungrateful, this man is perfect” [31], are avoided.

### 10.3.1 Sentiment annotation

Two methods have been used to annotate the corpus, a lexicon-based annotation and human annotation. The sentence is marked as positive if it contains positive terms or negated negative terms. It is considered negative if it contains negative terms or negation of positive terms. Any sentence that contains a mixture of positive and negative terms or no sentiment terms is marked as mixed or neutral.

| Lexicon | Negative | Positive | Negation |
|---|---|---|---|
| LABR | 348 | 319 | 37 |
| Moarlex | 13411 | 4277 | ▨ |
| SA lexicon | 3537 | 855 | ▨ |

Table 10.1: The number of terms in sentiment lexicons

In the lexicon-based annotation, we use three sentiment lexicons: the one provided by LABR [32] which contains negative, positive and negated terms; the Moarlex [33] and the SA lexicon [34] which contain only positive and negative terms. Table 10.1 illustrates the numbers of terms in each lexicon.

First, for the lexicon-based annotation we extracted 1,000 sentences from the Shami corpus and commissioned a Levantine native speaker to annotate them for sentiment. Then, we implemented Algorithm 2 to automatically annotate the same 1,000 sentences. We computed the inter-annotator agreement but the result was very bad, the disagreement was up to 80%. As a result, we did not consider this method as reliable for annotation, hence we chose to annotate the data set manually.

```
Result: Annotate 1,000 sentences
Build Positive, Negative, Negation lists of words extracted from the
  three lexicons;
Polarity = 0;
for sentence in Shami-Senti do
    count number of positive terms; Then Polarity ++;
    count number of negative terms; Then Polarity −−;
    check if there is a negation,Then Polarity $* -1$;
    if $Polarity > 0$ then
        Polarity is Positive;
    else if $Polarity < 0$ then
        Polarity is negative;
    else
        Polarity is mixed;
    end
end
```

**Algorithm 2:** Lexicon-based annotation of 1,000 Shami sentences

For the human annotation method, we asked two native speakers, one from Palestine and another from Syria, to annotate 533 sentences with 1 if these are positive, 0 if negative and -1 if neutral or mixed sentences. Then we calculated the inter-annotator agreement between them using Kappa statistics [35] giving us $\kappa = 0.838$ which is a very good agreement. Since the data was split into separate dialects, we asked the annotators to annotate the parts that they were most familiar with, for example, the Palestinian speaker annotated the sentences in Palestinian and Jordanian, while the Syrian speaker annotated the Syrian and Lebanese sentences. We extracted more than 5,000 sentences/tweets for this purpose, and have annotated nearly 2,000 of them so far. Table 10.2 shows the number of documents per category.

## 10.4 Experiments

In order to estimate the performance of the SA models, which have built on MSA data, on DA evaluation data, we use the following two corpora in our experiments.

- LABR [32]: this is one of the largest SA datasets to-date for Arabic. It consists of over 63k book reviews written in MSA with some dialectal words. LABR is available with different subsets: the authors split it into 2,3,4 and 5 sentiment polarities with balanced and unbalanced divisions. They depend on the user ratings to classify sentences. Thus, 4 and 5 stars ratings are taken as positive, 1 and 2 star ratings are taken as negative and 3 star ratings are taken as mixed or neutral. The fact that LABR is limited to one domain, book reviews, makes it difficult to use it as a general SA model.

- ASTD [36]: it is an Arabic SA corpus collected from Twitter and focuses on the Egyptian dialects. It consists of about 10k tweets, which are classified as objective, subjective positive, subjective negative, and subjective mixed.

Table 10.2 shows the number of instances of each polarity label in different corpora.

In all experiments, we use the same machine learning algorithms that have been used by the LABR baseline. These are:

1. Logistic Regression (LR)
2. Passive Aggressive (PA)
3. Linear Support Vector classifier (LinearSVC)
4. Bernoulli Naive-Bayes (BNB)
5. Stochastic Gradient Descent (SGD)

The choice is motivated as follows. LR is strong in explaining the relationship between one dependent variable and independent variables [37], while PA is suitable for large-scale learning [38]. LinearSVC is effective

| Corpus | NEG | POS | Mix |
|---|---|---|---|
| Shami-Senti | 935 | 1064 | 243 |
| LARB 3 Balanced | 6580 | 6578 | 6580 |
| LABR 2 Balanced | 6578 | 6580 | ▨ |
| ASTD | 1496 | 665 | 738 |

Table 10.2: The number of instances per category in Shami-Senti and other sentiment corpora used in our experiments

in cases where the number of dimensions is greater than the number of samples[39]. BNB is suitable for discrete data[40], and SGD is a linear classifier which implements regularised linear models with stochastic gradient descent (SGD) learning. It is a simple baseline classifier related to neural networks[41].

In addition, we also use some popular linear and probabilistic classifiers. Hence, we use Multinomial Naive-Bayes (MNB), which is suitable for classification of discrete features. The multinomial distribution normally requires integer feature counts and it works well for fractional counts like tf-idf [42]. We further use Complement Naive-Bayes (CNB), which is particularly suited for imbalanced data sets. CNB uses statistics taken from the complement of each class to compute the model's weights.[1] Generally speaking, a NB classifier converges quicker than discriminative models like logistic regression, so one need less training data. The last one is the Ridge Classifier (RC). Its most important feature is that it does not remove irrelevant features but rather minimise their impact on the trained model [43]. All of the algorithms are implemented using the `Scikit learn` library in Python [44] .

### 10.4.1 Three class sentiment classification

We start with the baseline from LABR, and use the 3-class balanced data set. Table 10.3 states the number instances of each polarity class for both training and testing. The baseline method from LABR uses the language model to predict the polarity class. We conduct two experiments: one with unigrams, and one with both unigrams and bigrams. We build the models by transforming the data into a numerical vectors using the Term Frequency vectorize method. First, a Language Model is built by extracting unigrams and bigrams from the dataset and computing their term-frequencies to create the two models, the unigrams, and the combined unigrams and bigrams. Then, every sentence goes through a classifier which produces a probability

[1] https://scikit-learn.org/dev/modules/naive_bayes.html

| | **Positive** | **Negative** | **Mix** |
|---|---|---|---|
| Train | 4936 | 4935 | 4936 |
| Test | 1644 | 1643 | 1644 |

Table 10.3: The number of instances per category in balanced LABR3

| **Classifier** | **Accuracy**<br>TF_wg1 | **Accuracy**<br>TF_wg1+2 |
|---|---|---|
| Logistic Regression | 59 | 59 |
| Passive Aggressive | 54 | 58 |
| Linear SVC | 57 | 58 |
| Bernoulli NB | 35 | 34 |
| SGD Classifier | 59 | 59 |

Table 10.4: Accuracy of the baseline on LABR3 (Tf-wg : is the Term Frequency on Word grams)

of the class the sentence belongs to. Table 10.4 shows the accuracy of the classifiers on the test set trained on the 3-class balanced LABR. The unigram and bigram TF method is doing marginally better than the unigram language model, particularly with the PA classifier. The four classifiers achieve an accuracy between 58% and 59% to classify MSA sentences. BNB is the worst performing classifier with 35% and 34% accuracy respectively. The reason for this might be that we have a large number of features (i.e. individual words) and since BNB models are counting the words that are not present in the document they do not perform well.

MSA has been researched more from an NLP perspective than DA, and therefore several sentiment analysis approaches have been built for it. The question we want to ask, is whether we can apply these NLP approaches directly on DA or new resources and models are required for DA. We, thus, test the reliability of models that are built on MSA data and adapt them to DA data. Here, we test the baseline bigram TF model on the test part of the Shami-Senti corpus. Table 10.5 shows the accuracy from this experiment where we trained the baseline by LABR3 and tested it using Shami-Senti. The accuracy is significantly worse, with a drop of more than 10%. The table also shows the accuracy of the baseline when we trained and tested it on Shami-Senti. The highest accuracy was 65% using SGD classifier.

Given the baseline model's poor performance on DA, we build a new SA model. This model also depends on language modelling, where we use a combination of both word-level and character-level n-grams. After several

| | **Training Dataset** | |
|---|---|---|
| **Classifier** | **LABR3** | **Shami-Senti** |
| Logistic Regression | 46 | 62 |
| Passive Aggressive | 43 | 64 |
| Linear SVC | 44 | 64 |
| Bernoulli NB | 11 | 48 |
| SGD Classifier | 45 | 65 |

Table 10.5: Accuracy of the baseline TF_wg1+2 trained on LABR3 and Shami-Senti and tested on Shami-Senti

| **Classifier** | **Model 1** | **Model 2** |
|---|---|---|
| Ridge Classifier | 57 | 59 |
| Logistic Regression | 59 | 60 |
| Passive Aggressive | 55 | 58 |
| Linear SVC | 57 | 59 |
| SGD Classifier | 59 | 60 |
| Multinomial NB | 57 | 59 |
| Bernoulli NB | 49 | 49 |
| Complement NB | 57 | 59 |

Table 10.6: Accuracy of the proposed model trained and tested on LABR3; Model 1: unigram word level with (2,5) character grams; In Model 2 (unigram,bigrams) word level with (2,5) character grams

experiments, we observe that a language model that combines features of word-level unigrams and bigrams with character-level n-grams from 2 to 5 gives the best accuracy. We test eight different machine learning algorithms to predict sentiment classification.

Table 10.6 shows the accuracy of our model on the LABR 3-class balanced dataset. In Model 1, we test using only unigram words and character grams from 2 to 5, while in Model 2 we add an extra bigram word-level to Model 1. The SGD and LR classifiers give the highest accuracy 60% on Model 2 which is slightly higher than the base line where it was 59%. In all experiments later we will refer to Model 2 as our proposed model. We test this model which was trained on LABR 3 on Shami-Senti. Table 10.7 shows the results. The model is not performing well on DA achieving an accuracy of 50% using the SGD classifier. This indicates that MSA models are not transferable to DA.

We also train the selected classifier configurations on the Shami-Senti

| Classifier | Accuracy |
|---|---|
| Ridge Classifier | 43 |
| Logistic Regression | 46 |
| Passive Aggressive | 43 |
| Linear SVC | 45 |
| SGD Classifier | 50 |
| Multinomial NB | 40 |
| Bernoulli NB | 44 |
| Complement NB | 42 |

Table 10.7: Accuracy of the proposed model trained on LABR3 and tested on Shami-Senti

| Classifier | Accuracy |
|---|---|
| Ridge Classifier | 69 |
| Logistic Regression | 67 |
| Passive Aggressive | 68 |
| Linear SVC | 69 |
| SGD Classifier | 68 |
| Multinomial NB | 71 |
| Bernoulli NB | 71 |
| Complement NB | 71 |

Table 10.8: Accuracy of the proposed model 3-class classification trained and tested on Shami-Senti

corpus (Table 10.8). NB algorithms give the highest accuracy with 71%, while the differences between the classifiers are marginal. We train the model using 1,000 samples and get an accuracy of 69% by MNB which indicates that increasing the size of the data set has a significant impact on the model accuracy.

### 10.4.2 Binary Sentiment classification

The accuracy obtained for the 3-class classification is not very high. This seems to be, at least partly, because the mixed class contains both positive and negative examples which makes the classification task difficult. LABR considers a 3-star rating as a mixed or neutral class. This is not very accurate since, in some cases, users use this rating as negative, while in others as somewhat positive. Table 10.9 shows three samples from the third

| | Sentence | Polarity |
|---|---|---|
| Arabic | بعض الكلمات استوقفتني وجعلتني أفكر.وبعضها الاخر جعلني أبتسم. *bḍ ālklmāt āstwqftny wǧltny afkr.wbḍhā ālāẖr ǧlny abtsm.*<br>والبعض جعلني أغرق في الضحك. اشتقت لهذا الأسلوب في الكتابة *wālbḍ ǧlny aġrq fy ālḍḥk. āštqt lhḏā ālaslwb fy ālktābh* | Positive |
| English | Some words stopped me and made me think.<br>Some of them made me smile. And some made me<br>drowned in laughter !!! I missed this method in writing. | |
| Arabic | الكتاب ليس بسيء ولكنه أثار ضجة اعلانية أكثر من اللازم *ālktāb lys bsy wlknh aṯār ḍǧh ālānyh akṯr mn āllāzm* | Mix |
| English | The book is not bad but it has too much publicity<br>more than it deserves | |
| Arabic | بالكاد اكملتها تفاصيلها كثيرة ومبهمة ومملة وبشعه جدا أشبه بالكوابيس *bālkād ākmlthā tfāṣylhā kṯyrh wmbhmh wmmlh wbšh ǧdā ašbh bālkwābys* | Negative |
| English | Barely completed, the details are many, opaque,<br>boring and very ugly like nightmares | |

Table 10.9: Examples annotated as neutral in LABR3 and the corrected polarity

neutral class in LABR that we consider should potentially belong to different classes.

We reduce the classification to a binary classification task, by focusing on the positive and negative classes only. Using the LABR, we build a baseline with bigram word counts and another model based on term frequency of unigram and bigram words. After that, we build a unigram and bigram TF words model and a (2-5) TF character model (the proposed model) and apply the LABR 2 classes dataset. The accuracy for the three models, in addition the accuracy of the same models tested on Shami-Senti are shown in Table 10.10.

| | **counting 2g** | | **TF_wg 1+2** | | **OUR Model** | |
|---|---|---|---|---|---|---|
| **Classifier** | **LABR** | **Shami** | **LABR** | **Shami** | **LABR** | **Shami** |
| Ridge Classifier | 78 | 53 | 81 | 54 | 83 | 57 |
| Logistic Regression | 80 | 57 | 80 | 56 | 82 | 58 |
| Passive Aggressive | 78 | 53 | 81 | 53 | 82 | 56 |
| Linear SVC | 78 | 55 | 81 | 55 | 83 | 58 |
| SGD Classifier | 80 | 53 | 82 | 54 | 83 | 56 |
| Multinomial NB | 78 | 52 | 80 | 53 | 82 | 55 |
| Bernoulli NB | 76 | 48 | 76 | 47 | 74 | 48 |
| Complement NB | 78 | 51 | 80 | 53 | 82 | 55 |

Table 10.10: Accuracy for binary classifiers with different feature sets trained on the LABR2 dataset and tested on LABR2 and Shami-Senti

We also test the transfer of models between different dialects. We train the classifiers with the proposed configurations to build a model on the ASTD corpus that contains Egyptian dialect data, and test it on both the ASTD and the Shami-Senti corpus. The results are shown in Table 10.11. The proposed model gives an accuracy up to 83% using linear classifiers like SVC and SGD when it is trained and tested on MSA LABR data set, while

| | **Testing Dataset** | |
|---|---|---|
| **Classifier** | **ASTD** | **Shami-Senti** |
| Ridge Classifier | 81 | 55 |
| Logistic Regression | 77 | 55 |
| Passive Aggressive | 82 | 57 |
| Linear SVC | 81 | 56 |
| SGD Classifier | 82 | 56 |
| Multinomial NB | 83 | 57 |
| Bernoulli NB | 82 | 58 |
| Complement NB | 82 | 58 |

Table 10.11: Accuracy of the proposed model on binary classification trained on ASTD and tested on ASTD and Shami-Senti

it gives an accuracy up to 58% when it is tested on Shami-Senti. We also get an accuracy of 83% when we train and test the model on the ASTD corpus and using an MNB classifier and 57% accuracy when we test it on Shami-Senti.

Models which are trained and built on MSA data can not fit well in dialectal data, even though both of them are considered similar languages. The accuracy for any model tested on Shami-Senti does not exceed 60% (Table 10.10 and Table 10.11) in all experiments. Table 10.12 shows that the model works better for binary sentiment classification with 74% accuracy using MNB, when the model is trained and tested on Shami-Senti. The high accuracy could be due to the quality of the data and human performed annotations. The high accuracy achieved (83%) on both LABR and ASTD indicates that increasing the size of the corpus improves the classification task.

### 10.4.3 Feature engineering

In order to improve 3-class sentiment classification, we consider adding more features to the language model. The classifiers with the new features are applied to both the LABR and the Shami-Senti corpus. Based on the three lexicons, (LABR, Moarlex and SA lexicon) we count the number of positive and negative terms in the sentence, and then calculate their probability using Equation 10.1 and 10.2. In addition, we use an additional binary feature to indicate if the sentence contains a negation term or not.

$$P(POS) = \frac{\#pos\_terms\_in\_the\_sentence}{total\_length} \tag{10.1}$$

| **Classifier** | **2 classes** |
|---|---|
| Ridge Classifier | 73 |
| Logistic Regression | 74 |
| Passive Aggressive | 73 |
| Linear SVC | 73 |
| SGD Classifier | 73 |
| Multinomial NB | 74 |
| Bernoulli NB | 72 |
| Complement NB | 75 |

Table 10.12: Accuracy of the proposed model on binary classification trained and tested on Shami-Senti

$$P(NEG) = \frac{\#neg\_terms\_in\_the\_sentence}{total\_length} \tag{10.2}$$

The three extra features and the word and character n-gram features are combined through the FeatureUnion estimator function in scikit-learn [2] to build and train the models. After many trials we chose to specify the weight of the transformer matrix to 0.4 for the positive feature, 0.2 for the negative feature, 0.4 for the negation feature and 2 for the language model features. The weight for the language module feature is doubled in order to increase their impact. Table 10.13 shows the result for the SGD and MNB classifiers on both the MSA and Shami corpus. On the MSA data set we get an accuracy of 58.1% and 58.2% using SGD and MNB respectively, which is not a valuable improvement compared to the results in Table 10.4. On the dialectal data set, the accuracy of the SGD classifier is decreased from 68% in Table 10.8 to 66%. We hypothesise that this is because of the lexicon which includes primarily MSA terms and Egyptian terms rather than Levantine sentiment terms so the probabilities of features are less accurate. Even though, MNB is still able to improve the classification accuracy from 71% to 75.2%.

The effect of feature engineering has more effect on the dialectal data, as the size of the dataset plays an important rule. Adding more informative features to a small dataset help the system to learn and predict the correct class.

### 10.4.4 Deep learning models

Deep learning has emerged as a powerful machine learning technique and has already produced state-of-the-art prediction results for SA [45, 46, 47].

[2] https://scikit-learn.org/0.18/modules/pipeline.html

| | **F.Eng** | |
|---|---|---|
| **Classifier** | **LABR** | **Shami** |
| SGD Classifier | 58.1 | 66 |
| Multinomial NB | 58.2 | 75.2 |

Table 10.13: Accuracy of two classifiers using feature engineering on 3-class classification task

| | **Accuracy** | |
|---|---|---|
| Experiment name | **LABR** | **Shami-Senti** |
| LSTM(100) | 42 | 64.7 |
| BiLSTM(200) | 41.3 | 61.8 |

Table 10.14: Accuracy of deep learning models 3-class LABR and Shami-Senti

In this section, we conduct a small experiment implemented using the Keras library to test two standard deep learning models to classify sentiment in our datasets.

The first model is a Long Short-Term Memory (LSTM) model. It consists of:

1. an embedding layer with max_features (MF) equal to the maximum number of words (7000), weighted matrix which is a 7000 * 100 matrix extracted from Aravec, a pre-trained Arabic word embedding model [48], and max_lenght = 50 as the maximum number of words in each sentence;
2. an LSTM layer with an output of 100 and 50% of dropout rate;
3. a dense layer with an output of 30 followed by a final sigmoid layer with 3 sentiment classes.

The second model, BiLSTM(200), uses a Bidirectional LSTM layer with an output of 200 rather than an LSTM layer with an output of 100. We train the model using the Adam optimiser and a batch size of 50. We train the two models on the LABR3 balanced corpus. In addition, we do the same experiments on Shami-Senti. Table 10.14 shows the results for both datasets.

The test accuracy, in general, is not at the desired level. It is clear that feature-based machine learning classifiers outperform deep learning networks.

## 10.5 Conclusion and future work

In this paper, we have investigated different ML algorithms and built a model for SA that combines word n-grams with character n-grams, in addition to other supportive features. The model outperforms the baseline on both big and small datasets, and gets an accuracy of 83% for MSA and 75.2% for Shami-Senti. What is more important, we have shown that using a model trained on MSA SA data and then testing it on dialectal SA data, does not produce good results. This suggests that MSA models cannot be easily, if at all, used in dealing with DA. There is, thus, a growing need for the creation of computational resources, not only for MSA, but also for DA. The extent of this need, and whether some resources can be re-used up to some point, is something that needs to be further investigated. In the case we have been looking at in this paper, it seems that the existing MSA approaches will not be very usable when thrown at dialectal data. It goes without saying that the same situation holds when one tries to use computational resources used for a specific dialect of Arabic to another one, modulo the closeness (in some computational measure to be defined) between the two varieties.

In the future, we plan to continue our work on the annotation of the Shami-Senti corpus exploiting more automatic ways and aiming at enhancing it in terms of size, quality and distribution. Once this happens, we plan to investigate the application of the same deep learning models used in this paper, as well as more sophisticated ones. On a similar note, we are currently working on using more sophisticated deep learning models for the same sized dataset we have been using in this paper. This is part of a more general question of using deep learning with small datasets: whether such an endeavour is possible, and if yes, what are the techniques and network tweaks that make this possible.

## Acknowledgements

The authors are supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg.

## 10.6 References

[1] Hossam S Ibrahim, Sherif M Abdou, and Mervat Gheith. "Sentiment Analysis for Modern Standard Arabic and Colloquial". In: *arXiv preprint arXiv:1505.03105* (2015).

[2] Muhammad Abdul-Mageed, Mona T Diab, and Mohammed Korayem. "Subjectivity and Sentiment Analysis of Modern Standard Arabic". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics. 2011, pp. 587–591.

[3] Muhammad Abdul-Mageed and Mona T Diab. "Subjectivity and Sentiment Annotation of Modern Standard Arabic Newswire". In: *Proceedings of the 5th linguistic annotation workshop*. Association for Computational Linguistics. 2011, pp. 110–118.

[4] Ahmed Mourad and Kareem Darwish. "Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs". In: *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*. 2013, pp. 55–64.

[5] Gilbert Badaro et al. "A Survey of Opinion Mining in Arabic: A Comprehensive System Perspective Covering Challenges and Advances in Tools, Resources, Models, Applications, and Visualizations". In: *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 18.3 (2019), p. 27.

[6] Kathrein Abu Kwaik et al. "A Lexical Distance Study of Arabic Dialects". In: *Procedia computer science* 142 (2018), pp. 2–13.

[7] Charles A Ferguson. "Diglossia". In: *word* 15.2 (1959), pp. 325–340.

[8] Kees Versteegh. *The Arabic Language*. Edinburgh University Press, 2014.

[9] Kathrein Abu Kwaik et al. "Shami: A Corpus of Levantine Arabic Dialects". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[10] Bing Liu. "Sentiment Analysis and Opinion Mining". In: *Synthesis lectures on human language technologies* 5.1 (2012), pp. 1–167.

[11] Kumar Ravi and Vadlamani Ravi. "A survey on Opinion Mining and Sentiment Analysis: Tasks, Approaches and Applications". In: *Knowledge-Based Systems* 89 (2015), pp. 14–46.

[12] Malak Abdullah and Mirsad Hadzikadic. “Sentiment Analysis on Arabic Tweets: Challenges to Dissecting the Language”. In: *International Conference on Social Computing and Social Media*. Springer. 2017, pp. 191–202.

[13] Umar Farooq et al. “Negation Handling in Sentiment Analysis at Sentence Level”. In: *JCP* 12.5 (2017), pp. 470–478.

[14] Aniruddha Ghosh and Tony Veale. “Fracking Sarcasm using Neural Network”. In: *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*. 2016, pp. 161–169.

[15] Noura Farra et al. “Sentence-level and Document-level Sentiment Mining for Arabic Texts”. In: *2010 IEEE international conference on data mining workshops*. IEEE. 2010, pp. 1114–1119.

[16] Donia Gamal et al. “Opinion Mining for Arabic Dialects on Twitter”. In: *Egyptian Computer Science Journal* 42.4 (2018), pp. 52–61.

[17] Ahmed Oussous, Ayoub Ait Lahcen, and Samir Belfkih. “Improving Sentiment Analysis of Moroccan Tweets Using Ensemble Learning”. In: *International Conference on Big Data, Cloud and Applications*. Springer. 2018, pp. 91–104.

[18] Alaa M El-Halees. “Arabic Opinion Mining using Combined Classification Approach”. In: *Arabic Opinion Mining using Combined Classification Approach* (2011).

[19] Rehab M Duwairi et al. “Sentiment Analysis in Arabic Tweets”. In: *2014 5th International Conference on Information and Communication Systems (ICICS)*. IEEE. 2014, pp. 1–6.

[20] Salima Medhaffar et al. “Sentiment Analysis of Tunisian Dialects: Linguistic Resources and Experiments”. In: *Proceedings of the third Arabic natural language processing workshop*. 2017, pp. 55–61.

[21] Nora Al-Twairesh et al. “Sentiment Analysis of Arabic Tweets: Feature Engineering and A Hybrid Approach”. In: *CoRR* abs/1805.08533 (2018).

[22] Rizkallah, Sandra and Atiya, Amir and ElDin Mahgoub, Hossam and Heragy, Momen", editor="Hassanien, Aboul Ella and Tolba, Mohamed F. and Elhoseny, Mohamed and Mostafa, Mohamed. “Dialect Versus MSA Sentiment Analysis”. In: *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*. Cham: Springer International Publishing, 2018, pp. 605–613.

[23] Ramy Baly et al. "A Characterization Study of Arabic Twitter Data with a Benchmarking for State-of-the-art Opinion Mining Models". In: *Proceedings of the third Arabic natural language processing workshop*. 2017, pp. 110–118.

[24] Ramy Baly et al. "Comparative Evaluation of Sentiment Analysis Methods Across Arabic Dialects". In: *Procedia Computer Science* 117 (2017), pp. 266–273.

[25] Gilbert Badaro et al. "A Large Scale Arabic Sentiment Lexicon for Arabic Opinion Mining". In: *Proceedings of the EMNLP 2014 workshop on Arabic Natural Language Processing (ANLP)*. 2014, pp. 165–173.

[26] Muhammad Abdul-Mageed and Mona Diab. "Toward Building a Large-Scale Arabic Sentiment Lexicon". In: *Proceedings of the 6th international global WordNet conference*. 2012, pp. 18–22.

[27] Gilbert Badaro et al. "ArSEL: A Large Scale Arabic Sentiment and Emotion Lexicon". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Ed. by Hend Al-Khalifa et al. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. ISBN: 979-10-95546-25-2.

[28] Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. "SAMAR: Subjectivity and Sentiment Analysis for Arabic Social Media". In: *Computer Speech & Language* 28.1 (2014), pp. 20–37.

[29] Hamed Al-Rubaiee, Renxi Qiu, and Dayou Li. "Identifying Mubasher Software Products Through Sentiment Analysis of Arabic Tweets". In: *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*. IEEE. 2016, pp. 1–6.

[30] Nawaf A Abdulla et al. "Arabic Sentiment Analysis: Lexicon-based and Corpus-based". In: *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)*. IEEE. 2013, pp. 1–6.

[31] Jihen Karoui, Farah Banamara Zitoune, and Veronique Moriceau. "Soukhria: Towards an Irony Detection System for Arabic in Social Media". In: *Procedia Computer Science* 117 (2017), pp. 161–168.

[32] Mohamed Aly and Amir Atiya. "LABR: A Large Scale Arabic Book Reviews Dataset". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2013, pp. 494–498.

[33] Mohab Youssef and Samhaa R El-Beltagy. "MoArLex: An Arabic Sentiment Lexicon Built Through Automatic Lexicon Expansion". In: *Procedia computer science* 142 (2018), pp. 94–103.

[34] Hady ElSahar and Samhaa R El-Beltagy. “A Fully Automated Approach for Arabic Slang Lexicon Extraction from Microblogs”. In: *International conference on intelligent text processing and computational linguistics*. Springer. 2014, pp. 79–91.

[35] Jean Carletta. “Assessing Agreement on Classification Tasks: the Kappa Statistic”. In: *Computational Linguistics* 2.22 (1996), pp. 249–254.

[36] Mahmoud Nabil, Mohamed Aly, and Amir Atiya. “ASTD: Arabic Sentiment Tweets Dataset”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2515–2519.

[37] Jiashi Feng et al. “Robust Logistic Regression and Classification”. In: *Advances in neural information processing systems*. 2014, pp. 253–261.

[38] Koby Crammer et al. “Online Passive-Aggressive Algorithms”. In: *Journal of Machine Learning Research* 7.Mar (2006), pp. 551–585.

[39] Suresh Kumar and Shivani Goel. “Enhancing Text Classification by Stochastic Optimization method and Support Vector Machine”. In: *International Journal of Computer Science and Information Technologies, 6 (4)* (2015), pp. 3742–3745.

[40] Hiroshi Shimodaira. “Text Classification Using Naive Bayes”. In: *Learning and Data Note* 7 (2014), pp. 1–9.

[41] Tobias Günther and Lenz Furrer. “GU-MLT-LT: Sentiment Analysis of Short Messages using Linguistic Features and Stochastic Gradient Descent”. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Vol. 2. 2013, pp. 328–332.

[42] Shuo Xu, Yan Li, and Zheng Wang. “Bayesian Multinomial Naive Bayes Classifier to Text Classification”. In: *Advanced multimedia and ubiquitous engineering*. Springer, 2017, pp. 347–352.

[43] Harris Drucker et al. “Support Vector Regression Machines”. In: *Advances in neural information processing systems*. 1997, pp. 155–161.

[44] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830.

[45] Lei Zhang, Shuai Wang, and Bing Liu. “Deep Learning for Sentiment Analysis: A Survey”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018), e1253.

[46] Lina Maria Rojas-Barahona. “Deep Learning for Sentiment Analysis”. In: *Language and Linguistics Compass* 10.12 (2016), pp. 701–719.

[47] Duyu Tang, Bing Qin, and Ting Liu. “Deep Learning for Sentiment Analysis: Successful Approaches and Future Challenges”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.6 (2015), pp. 292–303.

[48] Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. “AraVec: A Set of Arabic Word Embedding Models for use in Arabic NLP”. In: *Procedia Computer Science* 117 (2017), pp. 256–265.

# 11

# Study 5: Investigate the performance of Deep Learning methods for Dialectal Arabic Sentiment Analysis

*Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. "LSTM-CNN Deep Learning Model for Sentiment Analysis of Dialectal Arabic." In International Conference on Arabic Language Processing, pp. 108-121. Springer, Cham, 2019.*

In this paper we investigate the use of Deep Learning (DL) methods for Dialectal Arabic Sentiment Analysis. We propose a DL model that combines long-short term memory (LSTM) with convolutional neural networks (CNN). The proposed model performs better than the two baselines. More specifically, the model achieves an accuracy between 81% and 93% for binary classification and 66% to 76% accuracy for three-way classification. The model is currently the state of the art in applying DL methods to Sentiment Analysis in dialectal Arabic.

**Keywords:** Sentiment Analysis, Arabic Dialects, Deep Learning, LSTM, CNN

## 11.1 Introduction

With the emergence of social media, large amounts of valuable data become available online and easy to access. Social media users discuss everything they care about through blog posts or tweets, share their opinions and show interest freely; while they do not actually do it in person. We read about political debates, social problems, questions about a particular product, etc. Companies also use social networks to promote their products and services, and explore people's opinions to improve their products and services, thereby generating a huge amount of data. In this context, the need for an analytical tool that can process the users data and classify them in terms of sentiment polarities is increased and become a necessity.

Sentiment analysis (SA) or Opinion Mining (OM) is the task of determining and detecting the polarity/opinion in a given piece of text and classifying it into positive, negative or neutral and in some fine grained cases also a mixed class. English and other European languages have been explored in the majority SA tools and research; recent efforts extend the focus to other low-resources languages such as Arabic and dialectal Arabic.

Arabic is one of the five most spoken languages in the world, spoken by more than 422 million native speakers[1]. The situation in Arabic is a classic case of diglossia, whereby the written formal language differs substantially from the spoken vernacular [1, 2]. Modern standard Arabic (MSA) is heavily based on Classical Arabic and constitutes the official written language used in government affairs, news, broadcast media, books and education. MSA acts as the lingua franca amongst Arabic native speakers [3]. However, the spoken language (collectively referred to as Dialectal Arabic) widely varies across the Arab world. Moreover, there is neither standard written orthography nor formal grammar for these dialects.

To predict the sentiment of an Arabic piece of text, the majority of the works rely on Machine Learning (ML) algorithms like Linear Support Vector Classification (LinearSVC), Multinomial Naive Bayes (MNB) and others [4, 5, 6, 7, 8, 9]. Even though these classifiers are very easy to implement and achieve good results, they require a lot of feature engineering before applying the data to the classifiers. Therefore, work in Arabic sentiment analysis still depends heavily on the morphological and syntactic aspects of the language, such as POS tagging, word stemming, the sentiment lexicons and other hand-crafted features. It was in these areas that there have been several improvements in detecting sentiment.

After the remarkable improvement brought about by Deep Learning (DL) over the traditional ML approaches, researchers tend to investigate and explore the performance of the deep neural networks in analysing different kinds of Arabic texts and extract features for some NLP tasks such as: Language Identification, Text Summarising, Sentiment Analysis and so on [10, 11, 12].

In this paper we introduce a deep neural network which combines Bidirectional Long-Short Term Memory Networks (Bi-LSTM) with Convolutional Neural Networks (CNN) to predict the polarity of a text and classify

[1] http://www.unesco.org/new/en/unesco/events/prizes-and-celebrations/celebrations/international-days/world-arabic-language-day-2013/

it as either having positive or negative polarity. We exploit some available Arabic sentiment datasets: LABR [13], ASTD [14] and Shami-Senti [15] with different sizes and different dialects. Our system outperforms the state-of-the-art deep learning models for particular datasets like ASTD [16] with improvements on smaller datasets.

The paper is organised as follows: Section 2 gives a brief review of existing work that uses deep learning for Arabic Sentiment Analysis. In Section 3 we briefly discuss the deep learning architectures and we experiment two baselines: a simple LSTM model and the Kaggle model which uses a combination of LSTM and CNN layers, In Section 4, we propose our model and show that it outperforms both baselines, and achieves state-of-the-art results for DL models. Finally, In Section 5 we conclude and discuss directions for future work.

## 11.2 Related Work

Sentiment Analysis is usually considered a supervised classification task, where the texts are classified into two or more sentiments classes by providing a dataset with the text and the sentiment label. The common approach in SA is the use of ML through language modelling and feature engineering.

Most of the SA techniques in Arabic use words and character n-gram features with different representation settings and different classifiers [17, 13, 18, 19]. In some cases, ensemble classifiers are used [20, 21]. Moreover, sentiment lexicons are additional and valuable sources of features that have been used to enrich features for SA [22, 23, 24, 25].

In [26], the authors introduce a subjectivity and sentiment analysis system for Arabic tweets by extracting different sets of features such as the form of the words (Stem, Lemma), POS tagging, the presence of the sentiment adjective and the Arabic form of the tweet (MSA or DA), in addition to other Twitter-specific features such as the userID (person, organization) and the gender of the user. In [8] a language model is built and different machine learning classifiers are used to handle tweets in MSA and Jordanian.

An early deep learning framework for Sentiment Analysis for Arabic is proposed in [27]. The authors explore several network architectures based on Deep Belief networks, Deep Auto Encoder and the Recursive Auto Encoder. The authors there do not mention the range of labels of the polarity classification. They use The Linguistic Data Consortium Arabic Tree Bank (LDC ATB) dataset and show that the model outperforms the state of the art models on the same dataset by around 9% in terms of F-score. They get an accuracy of 74.5%.

Baly et al. [28] build a deep learning model to detect the polarities of tweets in a 5-scale classification that ranges from very negative to very positive. They retrieve tweets from 12 Arab countries in 4 regions (the Arab Gulf, the Levant, Egypt and North Africa). They collect 470K tweets. Their deep learning model consists of an embedding layer followed by an LSTM layer. Pre-trained word embeddings are applied using the skip-gram model from Word2Vec. The authors investigate the performance of their model on different morphological forms (lemma and stem). They achieve an accuracy of 70% for the Egyptian tweets and lemma embeddings while for UAE tweets they get 63.7% accuracy .

Soumeur et al. [29] investigate the Sentiment Analysis in the Algerian users' comments on various Facebook brand pages of companies in Algeria . They collect 100K comments written in Algerian, but they only annotate 25K comments as positive, negative or neutral. They apply a CNN as a feature extractor and transformation network. Their model consists of three type of layers, three CNN layers each with 50 filters and 3 kernel size, followed by pooling layers and the fully connected layers to predict the sentiment of the comment. Their model achieves an 89.5% accuracy.

SEDAT, a sentiment and emotion analyser model, was built in [30] using Arabic tweets. Word and document embeddings in addition to a set of semantic features are used. All the extracted features into CNN-LSTM networks followed by a fully connected layer are applied. The data has been obtained from the public datasets for SemEval2018 (Task 1: Affect in Tweets), which has a size of nearly 7K tweets. The authors further calculate Spearman's correlation coefficient over the baseline models which they outperform with 0.01-0.02 points of difference.

Recently, an ensemble deep learning model was proposed in [16]. There, the authors combine CNN and LSTM models to predict the sentiment of Arabic tweets exploiting Arabic Sentiment Tweets Dataset (ASTD). The model outperforms the state-of-the-art deep learning model's F1-score of 53.6%, as they achieve an accuracy of 65% and an F1-score of 64.46%.

## 11.3 Deep Learning Baselines for Sentiment Analysis

In this section we present two baseline DL systems for dialectal sentiment analysis. But first, we will talk about the word representation and Deep learning network architectures briefly, in the following subsections.

### 11.3.1 Word representation

Although word embedding vectors are easy to train, there are many pre-trained word vectors that were trained on a large amount of textual data. In this work we use Aravec, which is Arabic pre-trained word embeddings [31]. The Aravec are pre-trained using large data from multiple source like Twitter and Wikipedia and implemented by Word2Vec [32]. Each sample/sentence is replaced by a 2D vector representation of dimension $n \times d$, where $n$ is the number of words in the sentence and $d$ is the length of the embedding vector. After many trials we decided to apply the Aravec-CBOW model of dimension $d = 300$.

### 11.3.2 LSTM network

The traditional Continuous Bag of Word model (CBOW) allows to encode arbitrary length of sequence inputs as fixed-size vectors, but this disregards the order of the features in the sequence [32]. In contrast, Recurrent Neural networks (RNN) represent arbitrary-sized sequences in a fixed-size vector as CBOW, while they pay attention to the structure of the input sequence. Special RNNs with gated architecture such as LSTMs have proven very powerful in capturing statistical regularities in sequential inputs [33].

LSTM is the first network that introduces the gating mechanism and is designed to capture the long-distance dependencies and solve the problem of vanishing gradients [34]. While the LSTM is a feed-forward network that reads the sequence from left to right, the Bidirectional LSTM (Bi-LSTM) connects two layers from opposite directions (forward and backward) over the same output. The output layer receives information from both the preceding sequence (backwards) and following sequence (forward) states simultaneously. It is thus very useful when the context of the input is needed, for example when the negation term appears after a positive term [35].

### 11.3.3 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are feature extractor networks that are able to detect indicative local predictors in a large structure [36]. They are designed to combine these predictors to produce a fixed sized vector representation that captures and extracts the most informative local aspects for the prediction task.

For text classification, we use a 1D-CNN, a well known CNN architecture for dealing with sequences. It uses a convolution layer with a window size $k$ that is able to identify the indicative k-grams in the input text, and then act as an n-gram detector [33]. Every CNN applies a nonlinear function called a filter, which transforms a window of size $k$ into scalar values. After applying a multi-filter, the CNN produces $m$ vectors, where each vector corresponds to a filter. Thus, a pooling task is required to combine all of the $m$ vectors into a single $m$ dimension vector. Generally, CNNs focus more on the informative features and disregard their locations in the input text [37, 35].

### 11.3.4 Datasets

We use the following corpora in our experiments (the characteristics of these corpora are presented in Table 11.1):

- LABR [13]: it is one of the largest SA datasets to date for Arabic. The data are extracted from a book review website and consist of over 63k book reviews written mostly in MSA with some dialectal phrases. We use the binary balanced and unbalanced subsets of LABR, in addition to the three-way classification subsets. In LABR, user ratings are used in order to classify sentences. Ratings of 4 and 5 stars are taken by the authors as positive, ratings of 1 and 2 stars are taken as negative and 3 star ratings are taken as neutral. In the binary classification case, 3 star ratings are removed, keeping only the positive and negative labels.

- ASTD [14]: it is an Arabic SA corpus collected from Twitter and focusing on the Egyptian dialect. It consists of approximately 10k tweets which are classified as objective, subjective positive, subjective negative, and subjective mixed.

- Shami-Senti [15]: a Levantine SA corpus. It contains approximately 2k posts from social media sites in general topics, classified as Positive, Negative and Neutral from the four main countries where Levantine is spoken: Palestine, Syria, Lebanon and Jordan.

| Corpus | NEG | POS | Neutral |
|---|---|---|---|
| Shami-Senti | 935 | 1,064 | 243 |
| LABR 3 Balanced | 6,580 | 6,578 | 6,580 |
| LABR 2 Balanced | 6,578 | 6,580 | |
| LABR 2 Un-Balanced | 8,222 | 42,832 | |
| ASTD | 1,496 | 665 | 738 |

Table 11.1: The number of instances per category in the corpora used in our experiments

### Data Preparation

We apply the following pre-processing steps on all corpora:

1. Remove special characters, punctuation marks and all diacritics;
2. Remove all digits including dates;
3. Remove all repeated characters and keep only two repeated characters, using the algorithm from [15];
4. Remove any non-Arabic characters.

We replace every instance sentence with its corresponding word embedding vector from a pre-trained AraVec model [31]. In case of words that occur in the text but do not have an embedding in the pre-trained model, we look for the most similar words, and use them in order to get the corresponding word embeddings vector. More specifically, we look that the distance between the input word and the word in the Aravec model does not exceed two characters either from the beginning or the end of the word. The maximum length of every sentence is fixed to 70 words, thus we apply post-padding with zeros to ensure that all input sentences have the same length.

## 11.3.5 LSTM Baseline

This section describes our LSTM baseline. The Keras library has been used for the implementation of all experiments [38]. After many trials we used the Keras checkpoint function to save the best weight model. The checkpoint function automatically stops training when the validation loss starts increasing. After several experiments we decide to use the Adam optimiser with categorical cross entropy loss function for the multi-classification task, and RMSprop[2] for binary classification. The parameters we used in the baseline model are selected after running a number of experiments playing around with the different parameters as shown in Table 11.2. After many experiments, the parameters that lead to the best result for the baseline are highlighted in bold in the Table 11.2.

The first experiment we conduct uses a simple LSTM network which consists of an Embedding Layer with pre-trained word embedding followed by two LSTM layer with 128 and 64 output units respectively, followed by a fully connected Relu activation layer with 100 output units and a 0.5 dropout layer. Finally, a dense Sigmoid layer to predict the labels is used.

Table 11.3 shows the results for various LSTM-BiLSTM models with different combinations.The LSTM → LSTM experiment is the baseline model

[2] https://keras.io/optimizers/

| **Parameter** | **Value** |
|---|---|
| Dataset split | 80% train, 10% development, 10% test |
| Max number of features | [7K, 10K, **15K**, 25 K, 40K] |
| Embedding size | [100, **300**] |
| Embedding model | **CBOW**, Skip-gram |
| Embedding trainable | **True**, False |
| Max sample length | [50, **70**, 100] |
| Filter | [23, 64, 128] |
| Kernel size | [1, 2, 3, 4, 5, 6] |
| Pool size | [1, 2, 3, 4, 5] |
| Batch size | [32, **50**, 100, 128, 256] |
| Max epoch | 10, 50, **100**, 1000 |
| Dropout | 0.2, **0.5**, 0.7 |
| Optimiser | **Adam**, **RMSprop**, SGD |
| Activation function | Softmax, **Sigmoid**, **Relu** |
| LABR split (train/validation/test) | [70, 10, 20] |
| ASTD and Shami-Senti split (train/validation/test) | [80, 10, 10] |

Table 11.2: General parameters of deep learning models

described above, while in the BiLSTM → LSTM experiment, we change the first layer with a BiLSTM layer. Finally, we try both BiLSTM on the data (BiLSTM → BiLSTM). The model seems to be overfitting the data with the accuracy being very low (less than the 50%). When we apply the baseline model (LSTM → LSTM) on ASTD and ShamiSenti corpora we get a 53% accuracy for both.

| Dataset | Experiment name | Accuracy |
|---|---|---|
| LABR 3 | LSTM → LSTM (baseline model) | 41.9% |
| LABR 3 | BiLSTM → LSTM | 42.3% |
| LABR 3 | BiLSTM → BiLSTM | 40.6% |
| ASTD | LSTM → LSTM | 53% |
| Shami-Senti | LSTM → LSTM | 53% |

Table 11.3: Accuracy of networks with two sequential LSTM/BiLSTM layers for three-way classification

Given the low accuracy on the three class task, we investigate the task of binary sentiment classification using BiLSTM → LSTM model from the second experiment on all of datasets (LABR, ASTD, Shami-Senti) as it produces the highest accuracy among all the previous experiments. In the binary task, we employ RMSprop as an optimiser with binary cross entropy loss function. Table 11.4 shows the results.

The system achieves an unexpected result on the ASTD and LABR 2 unbalanced datasets of 68.5% and 81% accuracy respectively. Table 11.5 shows the confusion matrix for both of these datasets. Since in the ASTD

| Corpus | Test |
|---|---|
| LABR 2 balanced | 55.34% |
| LABR 2 un-balanced | **81** % |
| ASTD | **68.5**% |
| Shami-Senti | 54.5 % |

Table 11.4: Accuracy of the BiLSTM → LSTM model with binary classification task on our corpora

corpus the negative samples are approximately two-thirds the positive ones, the model tends to predict the negative class as an output label more often than the positive label. Similarly, in the LABR 2 unbalanced the model is biased towards the majority class, i.e. the positive class.

| ASTD corpus | | | |
|---|---|---|---|
| | | Predicted | |
| | | Positive | Negative |
| Actual | Positive | 11 | **45** |
| | Negative | 23 | **136** |

| LABR 2 unbalanced | | | |
|---|---|---|---|
| | | Predicted | |
| | | Positive | Negative |
| Actual | Positive | **8036** | 505 |
| | Negative | **1555** | 114 |

Table 11.5: Confusion matrix for the BiLSTM → LSTM model for ASTD and LABR 2 unbalanced corpora.

### 11.3.6 Kaggle Baseline

As a next step, we implement the winner model from the Kaggle sentiment analysis competition which was build for English sentiment analysis and has achieved an accuracy of 96%.[3] They used the Amazon Fine Food Reviews dataset, which includes 568,454 reviews, each review has a score from 1 to 5. The model is illustrated in Figure 11.1 and consists of a CNN layer with max pooling of size 2 and a dropout layer to exclude some features, followed by one LSTM layer, and at the end, a fully connected layer to predict one output class among 3 sentiment classes (Positive, Negative and Neutral).

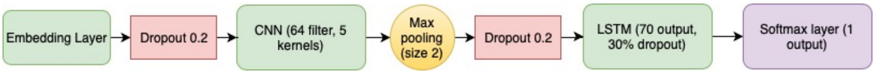


Figure 11.1: Kaggle winner model

We train the model using LABR, ASTD and Shami-Senti and apply both three-way and binary classification. The results are shown in Table 11.6.

[3] https://www.kaggle.com/monsterspy/conv-lstm-sentiment-analysis-keras-acc-0-96

We get a high accuracy for the LABR 2 unbalanced corpus and the ASTD corpus, 80.6% and 70.7% respectively. Taking a look at the confusion matrix in Table 11.7, we see that the model does not learn well. Being biased towards the majority class every time, it is clear that the model is overfitting the training data.

| Corpus | Three-way Classification | Binary Classification |
|---|---|---|
| Shami-Senti | 49% | 52.3% |
| LABR 2 unbalanced | | **80.6%** |
| LABR 2 balanced | | 53.1% |
| LABR 3 | 60% | |
| ASTD | 59.3% | **70.7%** |

Table 11.6: Accuracy of the Kaggle model on three-way and binary sentiment classification

| ASTD corpus | | | |
|---|---|---|---|
| | | Predicted | |
| | | Positive | Negative |
| Actual | Positive | 5 | **51** |
| | Negative | 12 | **147** |

| LABR 2 unbalanced | | | |
|---|---|---|---|
| | | Predicted | |
| | | Positive | Negative |
| Actual | Positive | **8153** | 387 |
| | Negative | **1591** | 78 |

Table 11.7: Confusion matrix for the Kaggle model on the ASTD and LABR 2 unbalanced corpora.

## 11.4 Our Model

In the previous section we have seen that using a combination of LSTM with a CNN enhances the accuracy of the model. Given these results, we propose a more sophisticated model than the one used in the Kaggle experiments that uses several CNN layers employing different filters and kernels to extract as many features as possible. In addition, we use a BiLSTM to extract the features from both directions and keep track of their effects. In contrast to the Kaggle model, in our model the BiLSTM precedes the CNN layers. We assumed that this configuration would provide a more informative representation of the sequential structure of sentences. The results, we get as shown in Table 11.8, seem to justify this assumption as our model performs better than Kaggle in all datasets. Figure 11.2 shows the best performing configuration which consists of an Embedding layer initialised with pre-trained word embedding vectors of size 300 and a max features of 15K, followed by two BiLSTM layers of 128 and 64 output units respectively and 0.5 dropout. The second BiLSTM layer is fed into parallel CNN layers with 5 region sizes (kernels) [2,3,4,5,6] and 3 filters [32,64,128] where we employ *Keras* functional API to build them. Each CNN layer is followed by Global MaxPooling layer. At the end of the CNN network we have a concatenated layer to merger all the outputs into one dimensions vector. This vector feeds into a fully connected Relu layer with 10 output units. Finally, Sigmoid layer with 3 output units for three-way classification and one binary unit for binary classification is used.

| Corpus | **Three-way Classification** | | | **Binary Classification** | | |
|---|---|---|---|---|---|---|
| | **Our Model** | Kaggle | LSTM | **Our Model** | Kaggle | LSTM |
| Shami-Senti | 76.4% | 49% | 53% | 93.5% | 25.3% | 54.5% |
| LABR 2 unbalanced | | | | 80.2% | 80.6% | 55.34% |
| LABR 2 balanced | | | | 81.14% | 53.1% | 81% |
| LABR 3 | 66.42% | 60% | 41.9% | | | |
| ASTD | 68.62% | 59.3% | 53% | 85.58% | 70.7% | 68.5% |

Table 11.8: Accuracy of the proposed model In addition to the comparing results from the two baselines on the three-way and binary sentiment classification

Our model achieves high accuracy results for binary sentiment classification in LABR, ASTD and Shami-Senti. The LABR 2 unbalanced dataset again has a high accuracy of 80.2%, when we look to the confusion matrix it is nearly the same like the one that has shown in Table 11.7. It is very clear that the LABR 2 unbalanced dataset does not learn well due to the data imbalance problem, which misleads the performance of the DL network although it has a reasonable size of training data.

Table 11.9 shows the confusion matrix for the three corpora. Even though the multi-classification results are not very high, our model outperforms the state-of-the-art deep learning models for some corpora like ASTD, where they achieve accuracy of 65% and F-score 64.5% [16]. In our proposed model we get an accuracy of 68.62% and an F-score equal to 69%. Both LABR 3 and ASTD are still suffering from the inaccurate annotation for the third neutral class. They assign the 3 star rating to neutral sentiment which complicates things, given that a 3 star rating might be quite positive or quite negative depending on a number of contextual parameters. This problem makes it hard to achieve very high accuracy when building a multi classification system using these corpora.

## 11.5 Conclusion and Future Work

In this paper we have investigated the use of Deep Learning architectures for dialectal SA. We first started by experimenting with a simple LSTM

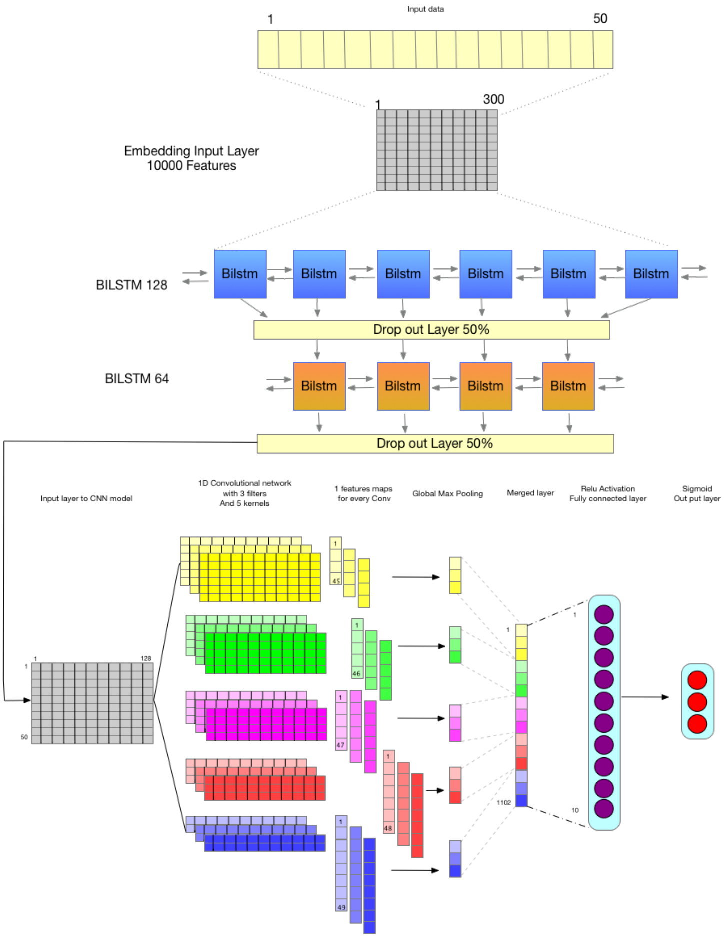


Figure 11.2: Final model with BiLSTM and CNN networks

| ASTD corpus | | | |
|---|---|---|---|
| | | Predicted | |
| | | Pos | Neg |
| Actual | Pos | 46 | 18 |
| | Neg | 13 | 138 |

| Shami-Senti | | | |
|---|---|---|---|
| | | Predicted | |
| | | Pos | Neg |
| Actual | Pos | 94 | 4 |
| | Neg | 9 | 93 |

| LABR2 Balanced | | | |
|---|---|---|---|
| | | Predicted | |
| | | Pos | Neg |
| Actual | Pos | 561 | 80 |
| | Neg | 168 | 506 |

Table 11.9: Confusion matrix for the proposed model in the ASTD, Shami-Senti and the LABR 2 balanced corpora

architecture on three dialectal SA datasets with poor results. We then took an off-the-shelf SA model that uses a combination of an LSTM and a CNN, i.e. Kaggle, and observed a better performance. Finally, we proposed our own model, which is a more elaborate BiLSTM → CNN with more convolutional layers, and obtained state-of-the-art results on the datasets that DL approaches have been previously applied to (i.e. the ASTD). In general, the results are promising but there is definitely room for improvement, especially on the threeway classification task.

One of the things that we would like to try in the future is the use of word embeddings specifically trained for the SA task, as well as even more complex DL architectures, for example those that use an attention mechanism. Another thing we want to do is to increase ShamiSenti's size, so that it is size-wise comparable to LABR3. It will then be possible to check whether the quality of the data will help the model obtain better accuracy scores, and furthermore check the effect of data size on the model's performance.

## Acknowledgements

Kathrein Abu Kwaik, Stergios Chatzikyriakidis and Simon Dobnik are supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg.

## 11.6 References

[1] Kees Versteegh. *The Arabic Language*. Edinburgh University Press, 2014.

[2] Charles A Ferguson. "Diglossia". In: *word* 15.2 (1959), pp. 325–340.

[3] Mustafa Shah. *The Arabic Language*. Routledge, 2008.

[4] Donia Gamal et al. "Opinion Mining for Arabic Dialects on Twitter". In: *Egyptian Computer Science Journal* 42.4 (2018), pp. 52–61.

[5] Ahmed Oussous, Ayoub Ait Lahcen, and Samir Belfkih. "Improving Sentiment Analysis of Moroccan Tweets Using Ensemble Learning". In: *International Conference on Big Data, Cloud and Applications*. Springer. 2018, pp. 91–104.

[6] Noura Farra et al. "Sentence-level and Document-level Sentiment Mining for Arabic Texts". In: *2010 IEEE international conference on data mining workshops*. IEEE. 2010, pp. 1114–1119.

[7] Rehab M Duwairi. "Sentiment Analysis for Dialectical Arabic". In: *2015 6th International Conference on Information and Communication Systems (ICICS)*. IEEE. 2015, pp. 166–170.

[8] Rehab M Duwairi et al. "Sentiment Analysis in Arabic Tweets". In: *2014 5th International Conference on Information and Communication Systems (ICICS)*. IEEE. 2014, pp. 1–6.

[9] Mohamed Elarnaoty, Samir AbdelRahman, and Aly Fahmy. "A Machine Learning Approach for Opinion Holder Extraction in Arabic Language". In: *arXiv preprint arXiv:1206.1011* (2012).

[10] Gheith A Abandah et al. "Automatic Diacritization of Arabic Text using Recurrent Neural Networks". In: *International Journal on Document Analysis and Recognition (IJDAR)* 18.2 (2015), pp. 183–197.

[11] Leena Lulu and Ashraf Elnagar. "Automatic Arabic Dialect Classification Using Deep Learning Models". In: *Procedia computer science* 142 (2018), pp. 262–269.

[12] Mohamed Elaraby and Muhammad Abdul-Mageed. "Deep Models for Arabic Dialect Identification on Benchmarked Data". In: *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. 2018, pp. 263–274.

[13] Mohamed Aly and Amir Atiya. "LABR: A Large Scale Arabic Book Reviews Dataset". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2013, pp. 494–498.

[14] Mahmoud Nabil, Mohamed Aly, and Amir Atiya. "ASTD: Arabic Sentiment Tweets Dataset". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2515–2519.

[15] Kathrein Abu Kwaik et al. "Shami: A Corpus of Levantine Arabic Dialects". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[16] Maha Heikal, Marwan Torki, and Nagwa El-Makky. "Sentiment Analysis of Arabic Tweets using Deep Learning". In: *Procedia Computer Science* 142 (2018), pp. 114–122.

[17] Asmaa Mountassir, Houda Benbrahim, and Ilham Berrada. "An Empirical Study to Address the Problem of Unbalanced Data sets in Sentiment Classification". In: *2012 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE. 2012, pp. 3298–3303.

[18] Amira Shoukry and Ahmed Rafea. “Sentence-Level Arabic Sentiment Analysis”. In: *2012 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE. 2012, pp. 546–550.

[19] Rasheed M Elawady, Sherif Barakat, and Nora M Elrashidy. “Different Feature Selection for Sentiment Classification”. In: *International Journal of Information Science and Intelligent System* 3.1 (2014), pp. 137–150.

[20] Nazlia Omar et al. “Ensemble of Classification Algorithms for Subjectivity and Sentiment Analysis of Arabic Customers’ Reviews”. In: *International Journal of Advancements in Computing Technology* 5.14 (2013), p. 77.

[21] Samar Al-Saqqa, Nadim Obeid, and Arafat Awajan. “Sentiment Analysis for Arabic Text using Ensemble Learning”. In: *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*. IEEE. 2018, pp. 1–7.

[22] Mahmoud Al-Ayyoub et al. “A Comprehensive Survey of Arabic Sentiment Analysis”. In: *Information Processing & Management* 56.2 (2019), pp. 320–342.

[23] Muhammad Abdul-Mageed, Mona T Diab, and Mohammed Korayem. “Subjectivity and Sentiment Analysis of Modern Standard Arabic”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics. 2011, pp. 587–591.

[24] Gilbert Badaro et al. “A Light Lexicon-based Mobile Application for Sentiment Mining of Arabic Tweets”. In: *Proceedings of the second workshop on arabic natural language processing*. 2015, pp. 18–25.

[25] Gilbert Badaro et al. “A Large Scale Arabic Sentiment Lexicon for Arabic Opinion Mining”. In: *Proceedings of the EMNLP 2014 workshop on Arabic Natural Language Processing (ANLP)*. 2014, pp. 165–173.

[26] Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. “SAMAR: Subjectivity and Sentiment Analysis for Arabic Social Media”. In: *Computer Speech & Language* 28.1 (2014), pp. 20–37.

[27] Ahmad Al Sallab et al. “Deep Learning Models for Sentiment Analysis in Arabic”. In: *Proceedings of the second workshop on Arabic natural language processing*. 2015, pp. 9–17.

[28] Ramy Baly et al. “Comparative Evaluation of Sentiment Analysis Methods Across Arabic Dialects”. In: *Procedia Computer Science* 117 (2017), pp. 266–273.

[29] Assia Soumeur et al. "Sentiment Analysis of Users on Social Networks: Overcoming the challenge of the Loose Usages of the Algerian Dialect". In: *Procedia computer science* 142 (2018), pp. 26–37.

[30] Malak Abdullah, Mirsad Hadzikadicy, and Samira Shaikhz. "SEDAT: Sentiment and Emotion Detection in Arabic Text Using CNN-LSTM Deep Learning". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 835–840.

[31] Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. "AraVec: A Set of Arabic Word Embedding Models for use in Arabic NLP". In: *Procedia Computer Science* 117 (2017), pp. 256–265.

[32] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[33] Yoav Goldberg. "Neural Network Methods for Natural Language Processing". In: *Synthesis Lectures on Human Language Technologies* 10.1 (2017), pp. 1–309.

[34] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the Difficulty of Training Recurrent Neural Networks". In: *International conference on machine learning*. 2013, pp. 1310–1318.

[35] Jürgen Schmidhuber. "Deep Learning in Neural Networks: An Overview". In: *Neural networks* 61 (2015), pp. 85–117.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

[37] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *nature* 521.7553 (2015), p. 436.

[38] Antonio Gulli and Sujit Pal. *Deep Learning with Keras*. Packt Publishing Ltd, 2017.

# 12

# Study 6: Investigate Distant supervision and Self training approaches on Dialectal Arabic Sentiment Analysis

*Kathrein Abu Kwaik, Stergios Chatzikyriakidis, Simon Dobnik, Motaz Saad, and Richard Johansson. "An Arabic Tweets Sentiment Analysis Dataset (ATSAD) using Distant Supervision and Self Training." In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 1-8. 2020.*

As the number of social media users increases, they express their thoughts, needs, socialise and publish their opinions. For good social media sentiment analysis, good quality resources are needed, and the lack of these resources is particularly evident for languages other than English, in particular Arabic. The available Arabic resources lack of from either the size of the corpus or the quality of the annotation. In this paper, we present an Arabic Sentiment Analysis Corpus collected from Twitter, which contains 36K tweets labelled into positive and negative. We employed distant supervision and self-training approaches into the corpus to annotate it. Besides, we release an 8K tweets manually annotated as a gold standard. We evaluated the corpus intrinsically by comparing it to human classification and pre-trained sentiment analysis models. Moreover, we apply extrinsic evaluation methods exploiting sentiment analysis task and achieve an accuracy of 86%.

**Keywords:** Sentiment Analysis, Distant Supervision, Self Training

## 12.1 Introduction

Companies and businesses stakeholders reach out to their customers through Social Media not only for advertising and marketing purposes, but also to get customer feedback concerning products or services. This is one of the main reasons that sentiment analysis applications have become increasingly sought out by the industry field. Even though sentiment analysis programs are widely used in the commercial sector, they have many other important uses, including political orientation analysis, electoral programs and decision-making. Sentiment Analysis is the process of automatically mining attitudes, opinions, views and emotions from the text, speech, tweets using Natural Language Processing (NLP) and machine learning [1]. Sentiment analysis involves classifying opinions into different classes like positive, negative, mixed or neutral. It can also refer to Subjectivity Analysis, i.e. the task of distinguishing between objective and subjective text.

There are so many Arabic speakers in the world and they speak different varieties of Arabic depending on the region but with only one variety that is standardised namely, Modern Standard Arabic MSA. Social media is prevalent and it is particularly this domain where the local varieties are used and for which the resources are most limited. The total number of monthly active Twitter users in the Arab region is estimated at 11.1 million in March 2017, generating 27.4 million tweets per day according to weedoo.[1] Arabic, especially dialects, still looking for more efficient resources that can be used for the needs of NLP tasks.

One of the biggest challenges in the construction of Arabic NLP resources is the big variation found in Arabic language where there are Modern Standard Arabic (MSA), Classical Arabic (CA) and the dialects. This has the result, that, in some tasks, it might be necessary to build stand-alone resources for each individual variation where the available tools have been built for MSA can not be adapted for dialects and vice-verse [2]. In addition, building resources requires sufficient time and funding to produce highly efficient resources. Moreover, deep learning NLP methods require a huge amount of data. As a result of the unique Twitter features that are widely used to express opinions, views, thoughts and feelings, we therefore present Arabic Tweets Sentiment Analysis Dataset (ATSAD) contains 36k tweets classified as positive or negative.

The contributions of this paper can be highlighted under two headings: a) resource creation and b) resource evaluation. Regarding resource creation, we introduce a sentiment analysis dataset collected from Twitter, and as for resource evaluation, we introduce a method that combines the distant supervision approach with self-training to build a dataset that satis-

[1] https://weedoo.tech/twitter-arab-world-statistics-feb-2017/

fies the size and quality requirements. In order to annotate a large number of tweets, we employ the distant supervision approach where the emojis are used as a weak noisy label. We manually annotate a subset of 8k tweets of the dataset and offer it as gold standard dataset. In order to improve the quality of the corpus, we apply the self-training techniques on the dataset and combine it with the distant supervision approach as a *double check approach*. Using our proposed double check approach, we achieve an accuracy of 86% on the sentiment analysis task. The dataset is available online for research usage.[2]

The rest paper is organised as follows: Section 2 reviews some related works in term of sentiment analysis and social media resources. In Section 3, the challenges of processing Twitter text are presented and in Section 4, the details of collecting and creating the tweets dataset are presented. We evaluate the dataset in Section 5. Sections 6 and 7 are the conclusion and future work sections respectively.

## 12.2 Related Work

Arabic Sentiment analysis (ASA) has received considerable attention in terms of resource creation [3, 4, 5, 6]. These resources are collected from different sources such as (blogs, reviews, tweets, comments, etc.) and involve a mix of Arabic vernacular and classical Arabic. Furthermore, they have been used extensively in research on SA for Arabic such as [7, 8, 9]. Most NLP work on SA uses machine learning classifiers with feature engineering. For example [10, 11] used machine learning classifiers on polarity and subjectivity classifications. However, recent papers [12, 13, 14] investigated the use of Deep Neural Networks for Arabic sentiment analysis. Most of the datasets are collected from web blogs and customer reviews. Some are manually annotated following a specific annotation guidelines, while other corpora like LABR [4] depend on the stars ratings done by users where the stars are used as polarity labels, the 5 stars denote a high positive, 1 star denotes a high negative and the 3 stars indicate the neutral and mixed label.

In the AraSenTi-tweets corpus [15], many approaches to collect the tweets were adopted, e.g the utilisation of emoticons, sentiment hashtags as well as the sentiment keywords. Then, the authors only keep the tweets that have their location set to a Saudi location. The dataset is manually annotated and sets some annotation guidelines. It contains 17 573 tweets each of which is classified to one of four classes (positive, negative, mixed or neutral). A sentiment baseline is built depending on TFIDF and using SVM with a linear kernel which achieved an F-score of 60.05%.

[2] https://github.com/motazsaad/arabic-sentiment-analysis

In [16], the authors presented the Arabic Sentiment Tweets Dataset (ASTD). It is a dataset of 10,000 Egyptian tweets. It is composed of 799 positive, 1,684 negative, 832 mixed and 6,691 neutral tweets. The authors also conducted a set of benchmark experiments for four way sentiment classification as (positive, negative, mixed, neutral) and two-way sentiment classification as (positive, negative). When focusing on two-way classification, the corpus is unbalanced and small to be useful for the two-way sentiment analysis task.

A corpus for Jordanian tweets is also presented in [17]. The authors collected tweets according to location, and then they filtered them to collect different types of terminologies to identify Jordanian Arabic dialect keywords efficiently. The corpus contains 3,550 Jordanian dialect tweets manually annotated as follows: 616 positive tweets, 1,313 negative tweets, and 1,621 neutral tweets. They conducted several experiments both with and without stemming/rooting applying them to several models with unigrams/bi-grams and trying NB and SVM classifiers. The result shows that the SVM classifier performs better than the NB classifier. The ROC performance reached an average of 0.71, 0.77 on NB and SVM respectively on all experiments. A similar corpus for Levantine dialects is presented in Shami-Senti [2]. It has approximately 2.5k posts from social media sites in general topics classified manually as positive, negative and neutral. The corpus is still under development.

Recently, a 40K tweets dataset is presented in [18]. The authors extracted tweets written in Arabic. After that, they reprocessed the tweets and cleaned them very carefully by two experts, they corrected every misspelling words and removed all the repeating characters, in addition to the normal cleaning steps like normalisation. The total size of the dataset is 40,000 tweets classified into positive and negative equally. The corpus is considered a reliable resource but by manually cleaning all the data, it turns to a very hard crafted corpus where the resulted clean corpus differ than the real tweets, where the goal of cleaning is to normalise text and remove spelling mistakes but keep the style of the author. This has been normalised too much in this corpus and hence important information was lost.

Even though in most of the Arabic tweet corpus creation procedures, the authors used the emoticons to extract as many sentiment tweets as possible such as [15, 19], however none of them using the emojis and the emoticons as a sentiment label. An emoticon is built from keyboard characters that when put together in a certain way represent a facial expression like :) ;) :( and so on, while an emoji is an actual image[3]. The Stanford Twitter Sentiment (STS), is one of the most well-known dataset for English Twit-

[3] https://grammarist.com/new-words/emoji-vs-emoticon/

ter sentiment analysis [20]. The dataset provides training and testing sets. The tweets were collected on the condition to contain at least one emoticon. Then they automatically classified the tweets in regard to the emoticons to positive and negative. The process resulted in a training set of 1.6 million annotated tweets and a test set of 359 manually annotated tweets that are used as a gold standard. The data set has been extensively used for different tasks related to sentiment analysis and subjectivity classification [21, 22, 23, 24]. Refaee and Rieser [19] presented Arabic subsets of tweets using emoticons, hashtags and keywords. They apply distant supervision on the emoticons subset. After the evaluation process, they get an accuracy 95% and 51% for subjectivity analysis and sentiment classification respectively. They comment that emoticons can be used efficiently with subjectivity detection but not for the polarity classification task.

As obvious from the previous discussion, these corpora or dataset have lacked some aspect. They have some limitation in term of the size of the corpus as ASTD, the number of presented dialects as AraSenti and the annotation procedure like LABR. We are looking for Arabic sentiment analysis corpus that concerns the Arabic social media text and that handles multiple dialects in a reasonable number of instances size to conduct experiments and find a way to do the annotation as accurate as possible. In this paper, and similarly to STS [20], we constructed a dataset based on emojis for extracting and classifying tweets. Additionally, we manually annotated 20% of this data, which can then be used as a gold standard for any tweets sentiment analysis task and as the test set for our corpus.

## 12.3 Challenges of processing text from social media

Natural language processing must be adapted to the type of text to be processed (formal, scientific, colloquial), but furthermore, humans differ in the way they write in that specific type of text. This variety in writing style has increased with the advent of social media, where people are using their style of writing and daily conversational language to post, reply, or tweet more often. In addition to specific idiosyncrasies of Arabic in terms of processing, Twitter has unique features that make tweets have different characteristics from other social media [25, 26]. Detecting sentiment in social media text in general and Twitter in particular is a non-trivial task. There are many challenges as follows:

- The short text length is the unique characteristics of tweets, which can be up to 280 characters.

- Due to the constraint on the length of the tweet (280 characters), users tend to employ abbreviations in the tweets to make room for other words.

- Tweets, as well as other social media text, are an example of *User Generated Content*, and contain unstructured language, orthographic mistakes, use of slang words, a lot of ironic and sarcastic sentences, abbreviations and many idiomatic expressions.

- Analysing Arabic tweets in specific is a challenging task due to the use of Arabic dialects in tweets which (due to the lack of standard orthography) results to a lot of spelling inconsistencies. Moreover, the lack of capitalisation and diacritics, as well as the usage of connected words like إنشاالله *inšāāllh* increase the complexity of processing Arabic tweets.

- The extensive of use of misspellings Arabic result in a Data Sparsity, that has an impact on the overall performance of SA systems. Saif et. al; saif2012alleviating propose a semantic smoothing model by extracting semantically hidden concepts from tweets and then incorporate them into supervised classifier training through interpolation to reduce the sparseness in English tweets.

- Many Arabic tweets are verses from the Holy Quran. There prayers to refer to different situations with different meanings are used, for example, ماما بشتاقلك كتير. الله يرحمك ويجمعنا معك في الجنة *māmā bštāqlk ktyr. āllh yrḥmk wyǧmnā mk fy ālǧnh*, which in English means *Mam I miss you a lot. I ask God to have mercy on you and to bring us together in heaven*, even though it ostensibly carries a positive meaning of empathy and paradise, it carries negative feelings of longing and loss due to death.

## 12.4 Arabic Tweets Sentiment Analysis Dataset (ATSAD)

To create and build the sentiment analysis corpus or datasets, we first build a sentiment emoji lexicon. The lexicon contains both positive and negative emojis expressing the feelings corresponding to different sentiment categories. We collect the emojis as well as their indicated sentiment from “Emojis Sentiment Ranking Lexicon" [27] which is available at `http://kt.ijs.si/data/Emoji_sentiment_ranking/` and Emojipedia[4]. Then,

[4] https://emojipedia.org/people/emojis

this lexicon is employed as the seed for the Twitter retrieval procedure. The Lexicon is composed of 91 negative emojis and 306 positive emojis.

Instead of collecting tweets by hashtags or query terms we exploit the emojis and their assigned sentiment and condition the tweet language set to Arabic. We extracted 59k of the tweets using the Twitter API in April 2019. The corpus contains multiple dialects from all over the Arab world as it is not geographically constrained. To automatically annotate the tweets either as positive or negative, we use the emojis as a noisy (weak) label. If the tweet is fetched by the positive emojis from the lexicon like ☺ then it is labelled as positive and the tweets fetched by the negative lexicon are labelled as negative.

More specifically, we perform the following cleaning actions:

1. Remove all metadata generated by Twitter API like tweet_id, username, time, location, RT
2. Remove all special characters but not emojis
3. Remove non-Arabic characters
4. Remove links
5. Remove diacritics from the text
6. Remove duplicated tweets

Table 12.1 shows the statistics of the corpus before and after the pre-processing phase which gives us 36K tweets.

| | Positive | Negative | Total | Vocabs | Words |
|---|---|---|---|---|---|
| Before | 30,607 | 29,232 | 59,839 | 95,538 | 76,2673 |
| After | 18,173 | 18,695 | 36,868 | 95,057 | 41,8857 |

Table 12.1: Statistics of the Twitter sentiment analysis corpus (ATSAD) before and after the pre-processing

## 12.5 Corpus Evaluation

The process of building a resource is not limited to data collection, but it must be checked and verified in order to be trustworthy and used as a resource. In this section, we evaluate the Tweets corpus by introducing two well-known methodologies: Intrinsic and extrinsic evaluations.

In intrinsic evaluation, the corpus is directly evaluated in terms of its accuracy and quality. We check whether the rule-based annotation (simply an emojis annotation) can be used to build a reliable corpus and use it effectively in the desired functionality. On the other hand, in extrinsic evaluation, the dataset is going to be assessed with respect to its impact on an external task which in our case is the sentiment analysis model [28].

To check the quality of the corpus, we have asked two annotators, one an NLP expert, the second an educated native Arabic speaker, to annotate subsets of the corpus. We start with a random sample containing 180 instances (1% of the data) for both positive and negative classes. When the annotation was completed, the two annotators agreed on the 90% of the sample.

In case of disagreement, we choose the expert annotator's choice as the class label. The annotation process is cumulative, in the sense that we pick random samples every time from the corpus and ask the annotators to annotate. For each sample we calculate the number of mismatched labels between the emoji-based annotation and the human annotation, and we also compute the accuracy of the emoji-based annotation by taking the number of right classified instances divided by the total number of the sample. Table 12.2 shows the number of errors (mismatches) and accuracy for annotation samples in the range from 1% to 10% of the corpus. Figure 12.1 plots the accuracy results. It is clear that after manually annotating 10% of the whole corpus, the percentage of matches tweets between the human and the emoji-based annotation is 77.2%.

Obtaining 77.2% is not good enough to use it for a task to predict the real sentiment of the tweets even though it is less time-consuming compared to manual annotation. Therefore, later we are going to present a combination method of self training and distant supervision to improve the quality of the dataset.

Moreover, we check the quality of the corpus with pre-trained sentiment analysis models that have been built and trained on existing datasets. The following datasets are used in our experiments and shown in Table 12.3:

- 40k dataset [18]: as mentioned in the related work section, this is a tweets dataset containing 40,000 instances. It is manually annotated into positive and negative and the tweets are subsequently manually cleaned.

- LABR [4]: a large SA dataset for Arabic sentiment analysis. The data are extracted from a book review website and contain over 63k book reviews written in MSA with some dialectal phrases. Given that our corpus concerns two-way classification, we only use the binary balanced subsets of LABR. LABR can be considered to be a human

| Sample % | Samples | #errors | Accuracy |
|---|---|---|---|
| 1% | 360 | 106 | 70.5% |
| 2% | 720 | 200 | 72.2% |
| 3% | 1,080 | 293 | 72.9% |
| 4% | 1,400 | 370 | 74.3% |
| 5% | 1,800 | 450 | 75% |
| 10% | 3,608 | 823 | 77.2% |

Table 12.2: Human annotation accuracy compared to the emojis based annotation. The first two columns show the percentage and number of the sampled tweets, #_error shows the number of mismatched samples and the Accuracy column calculates the percentage of the matches between both annotations.

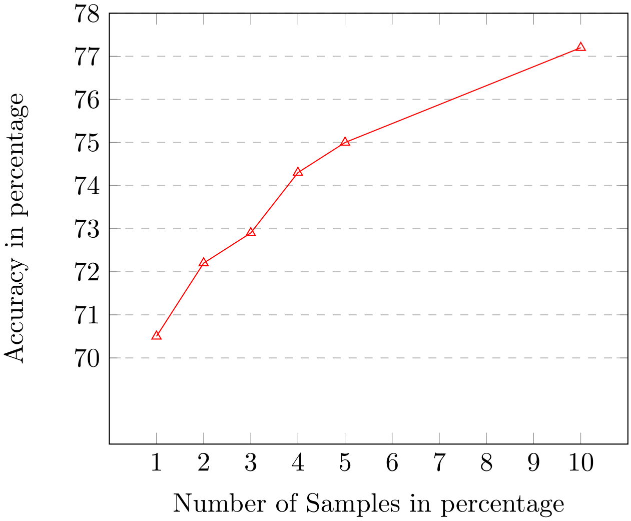


Figure 12.1: Accuracy of dataset comparing to human annotation

annotated corpus, where the users rate books using the stars system (1 to 5).

Ratings of 4 and 5 stars are considered positive, ratings of 1 and 2 stars negative and 3-star ratings are taken as neutral. In the binary classification case, 3-star ratings are ignored, keeping only the positive and negative labels.

- ASTD [16]: an Arabic SA corpus collected from Twitter and focusing on Egyptian Arabic. It consists of approximately 10k tweets which are classified as objective, subjective positive, subjective negative, and subjective mixed. We use only the positive and negative subset.

- Shami-Senti [2]: a Levantine SA corpus. It contains approximately 2.5k posts from social media sites in general topics classified manually as positive, negative and neutral from the four main countries where Levantine is spoken: Palestine, Syria, Lebanon and Jordan.

| Corpus | NEG | POS |
|---|---|---|
| 40k tweets | 20,002 | 19,998 |
| LABR 2 Balanced | 6,578 | 6,580 |
| ASTD | 1,496 | 665 |
| Shami-Senti | 935 | 1,064 |

Table 12.3: The number of instances per category in the corpora used in our experiments

We build a model on each corpus and apply the resulting model to our Twitter corpus. The model uses a combination of (1-3) word grams and a LinearSVC classifier. Table 12.4 shows the accuracy of the models built (trained and tested) on the original datasets, while the ASTAD column shows the accuracy of the trained model when we use it to predict the class on our Twitter dataset. It is clear that none of the models works for this dataset and the accuracy does not exceed 60%. This is an expected result, given that the data are from a very different domain, i.e. book reviews. Even though both ASTD and the ATSAD share the same domain, the ASTD only contains Egyptian dialects. In the case of the Shami corpus, it only contains Levantine dialects with a limited number of examples (2k). The 40k tweets model and ATSAD also share the same domain (tweets) but the manual hard prepossessing and cleaning of the data make it hard to predict real tweets as people post it, also the 40k corpus only has Egyptian dialect.

Summing up, it is clear from the previous discussion that the ATSAD is a challenge for the models trained on the available datasets that are standardised and regularised. Therefore we have to create an ML model that would be successful on this ATSAD. To achieve a good accuracy on the model, then the dataset should be improved in term of the data quality and annotation quality.

| | Same corpus | ATSAD |
|---|---|---|
| 40k tweets | 79% | 60% |
| LABR | 82% | 54% |
| ASTD | 81% | 59% |
| SHAMI-SENTI | 84% | 59% |

Table 12.4: Accuracy of models trained on different SA corpora; the same corpus column indicates the accuracy of the model when the train dataset and the test dataset are both from the same corpus, the last column for the accuracy when we test the models on the ATSAD

## 12.6 Self-training on Distant supervision Corpus

Creating a good resource requires the collection of a big amount of data that are preprocessed and annotated. The annotation is usually done by hiring annotators and specifying annotation rules they have to follow to produce a reasonable annotation agreement. This process is time and money consuming. There is another approach to build a large enough dataset more quickly. The process is called Distant supervision or weak supervision [29].

Distant supervision involves heuristically matching the contents of a database to the corresponding text [30]. In our case, we use the emojis in the tweets to work as weak labels with which we can annotate the 36K tweets automatically. Although this is sometimes not producing high-quality dataset, it works in some tasks.

We annotate the 36k tweets by distant supervision and then extract 4k tweets (10% of the total dataset). We ask the two annotators to label them manually. We compute the number of agreed annotation between the human annotation and the emojis annotation we have an agreement of 77.2%.

To use the human annotation dataset as a gold standard we extract other 4K tweets and also manually annotate them, upgrading the final manually annotated dataset to 8k tweets of which 3705 are classified as positive, 3911 negative and 384 instances are mixed. We exclude the mixed class from our experiments.

We build a baseline with TF-IDF unigram word model and a Linear-SVC classifier. Moreover, we build another complex model -from some previous work - by combining word n-grams (1-5), character n-grams (2-5) with and without word boundary consideration [2]. The models are built for sentiment analysis and the problem is recognised as two-way classification, so every tweet is classified either as positive or negative. Table 12.5 shows the number of tweets per class for the human annotation dataset and the remaining tweets in the emojis dataset which were weakly annotated by the distant supervision.

| | Human annotated | Emojis annotated |
|---|---|---|
| | **Label Distribution** | |
| #Positive | 3,705 | 14,468 |
| #Negative | 3,911 | 14,784 |
| | **Train/Test Distribution** | |
| #Train_set | 6,092 | 23,401 |
| #Test_set | 1,524 | 5,851 |
| #Total_set | 7,616 | 29,252 |

Table 12.5: Statistics of the human annotation subset and the emojis distant supervision subset after subtract the human dataset

We apply both the baseline and the complex model on the manually annotated dataset and we get an accuracy of %71 and %79 respectively. We refer to this experiment as (Manual experiment). To check again the quality of the emojis based dataset we applied the previous model trained on the human labels on the emojis dataset of 29k tweets to predict the label. After testing the two models, the resulted accuracy is %63 and %76 for both the baseline and the complex model respectively (Mixed experiment). The mixed experiment is to some extent similar to the agreement between the manual annotation and the emojis annotation experiment we have done first and got an accuracy of 76% using 4k subset.

To improve the quality of the automatic annotation and therefore the proposed tweets corpus, we will exploit the manual annotation dataset to enhance the entire dataset. Therefore, a self-training approach is to be employed on the data to improve the classification and increase the accuracy of the annotation. Self-training is a commonly used method for semi-supervised learning [31, 32]. The idea of Self-training is to train a classifier with a small amount of labelled data and incrementally retrain the classifier by adding the most confidently labelled instances that were previously unlabelled as a new data. This process continues until most of the unlabelled data becomes labelled [33]. We can implement a self-training technique with little modification of the existing configuration: our dataset is not completely unlabelled but has weak emoji-based annotations. From the mixed model experiments, rather than extracting the instances predicted with the highest confidence, we extract instances where the model prediction label matches the emojis label. This is the case for 22,542 out of 29,252 tweets in the dataset. We add these tweets to the training set

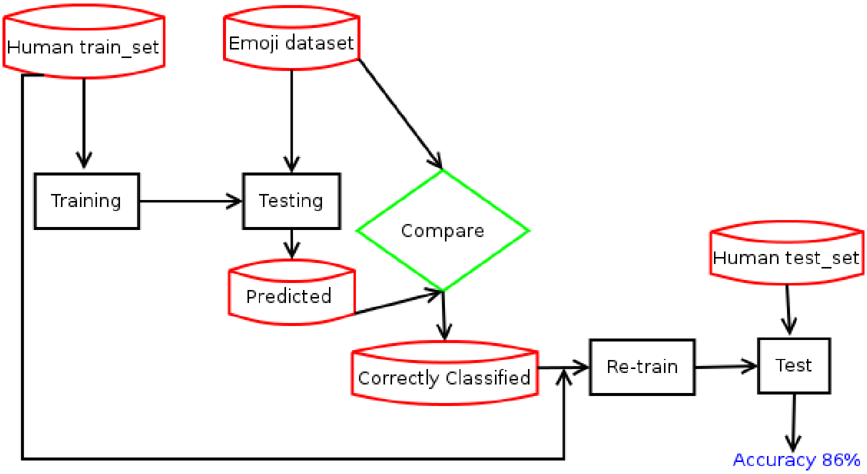


Figure 12.2: Self training (double-check) approach applied on the ATSAD

which consists of the human annotated dataset (6,092). Thus, to re-train the classifier we have a total of 22,542 + 6,092 = 28,634 tweets. We call this experiment (double check) where we combine the self training with distant supervision. The 28K tweets are now a dataset with strong supervised labelling where the small amount of human annotation dataset and distant supervising from emojis helps to annotate more data. We re-build both the baseline and the complex models and retrain them on the dataset we produced from the double check experiment (28k tweets), then apply the model to the test set from the human annotation dataset (1,524 tweets).We use the same dataset across all the experiments in order to allow for the comparison. The baseline and the complex model accuracy increases to 77% and 86% respectively. Figure 12.2 shows the diagram for the self-training approach.

To evaluate our self-training experiment and our method to extract only those instances where the model prediction matches the emojis annotation, we conduct a small experiment of self-training called (Non-check) where we:

1. Use the model from the (mixed experiment) to predict the label for the automatically labelled dataset (29k tweets).

2. Retrain the model with the human annotated training dataset in addition to the predicted labelled dataset (from the previous model). Thus, this re-train dataset consists of 6,092 + 29,252 = 35,341 tweets.

3. Use the manually annotated test set (1524 tweets) and use the model to predict the sentiment.

4. The accuracy of the baseline is 70% and 81% for the complex model.

Consequently, it is clear that (i) using the emojis as a noisy label, (ii) matching with the human annotation and (iii) apply the self training technique to annotate the dataset leads to an improvement of the data. Table 12.6 shows the performance of the models on different datasets. These are represented as plots in Figure 12.3.

| Experiment | #Train | #test | Baseline | Complex |
|---|---|---|---|---|
| Manual | 6,092 | 1,524 | 71% | 79% |
| Mixed | 6,092 | 29,252 | 63% | 76% |
| double-check | 28,634 | 1,524 | **77%** | **86%** |
| Non-check | 35,341 | 1,524 | 70% | 81% |

Table 12.6: The performance of the baseline and complex models on different datasets.

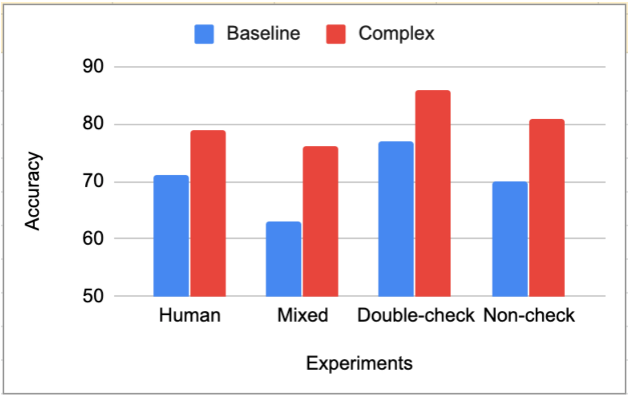
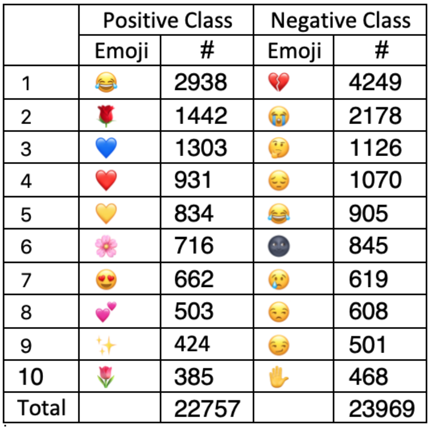


Figure 12.3: plotting the accuracy for all the experiments for both the baseline and complicated models

When we were done with the experiments, we extracted all the emojis and examined the emoji frequencies per category. We found some emojis are shared between the positive and the negative class, such as the smiley face with tears. We also discover that people used the black smiley face to indicate the negative feeling more often than the positive. These emojis are considered tricky emojis and they decrease the quality of the annotation. We modified our conditions by removing all the misleading emojis to collect more accurate data. Up to now we have collected over 200k tweets. Table 12.7 views the number of occurrence for the most 10 frequent emojis per sentiment category.



| | Positive Class | | Negative Class | |
|---|---|---|---|---|
| | Emoji | # | Emoji | # |
| 1 | 😂 | 2938 | 💔 | 4249 |
| 2 | 🌹 | 1442 | 😭 | 2178 |
| 3 | 💙 | 1303 | 🤔 | 1126 |
| 4 | ❤️ | 931 | 😔 | 1070 |
| 5 | 💛 | 834 | 😂 | 905 |
| 6 | 🌸 | 716 | 🌚 | 845 |
| 7 | 😍 | 662 | 😢 | 619 |
| 8 | 💕 | 503 | 😒 | 608 |
| 9 | ✨ | 424 | 😏 | 501 |
| 10 | 🌷 | 385 | ✋ | 468 |
| Total | | 22757 | | 23969 |

Table 12.7: Number of occurrence for the most 10 frequent emojis per category, the last row show the total number of the whole emojis in the dataset per category

## 12.7 Future work

Based on our emojis analysis and the subsequent modification of the data collection and annotation conditions, we are planning to further increase the size of the dataset and use it for different tasks like building custom sentiment word embeddings and to fine-tune deep learning networks.

## 12.8 Conclusion

To extend the limited Dialectal Arabic resources, we collected an Arabic Tweets Sentiment Analysis Dataset (ATSAD). The corpus has been collected from Twitter during April 2019 and employs emojis as seeds for extraction of candidate instances. After the pre-processing, we apply distant supervision using emojis as weak labels to annotate the entire dataset. In addition, we commissioned two annotators to manually annotate a subset of 8k tweets. We evaluate the corpus by comparing the emoji-based annotation with the human annotation and we get an observed agreement of 77.2%. We built a sentiment analysis machine learning model with the unigram features as a baseline and another complex model that utilises word grams and character grams. We exploit the human annotation dataset to help us improve the annotation of the automatically labelled dataset by self-training approaches. Over several experiments we achieve an accuracy of 86%.

Using the distant supervision approaches for automatically data annotation process can saves us a lot of effort, time and money. Distant supervision is a very valuable method to annotate large number of instances automatically, in our case based on emojis to denote the category. The self training approach can be used together with a small number of manually annotated instances to improve the quality of the automatically labelled dataset.

## Acknowledgements

Kathrein Abu Kwaik, Stergios Chatzikyriakidis and Simon Dobnik are supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg. Richard Johansson was funded by the Swedish Research Council under grant 2013–4944.

## 12.9 References

[1] Bing Liu. “Sentiment Analysis and Opinion Mining”. In: *Synthesis lectures on human language technologies* 5.1 (2012), pp. 1–167.

[2] Chatrine Qwaider, Stergios Chatzikyriakidis, and Simon Dobnik. “Can Modern Standard Arabic Approaches be used for Arabic Dialects? Sentiment Analysis as a Case Study”. In: *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics*. 2019, pp. 40–50.

[3] Mohammed Rushdi-Saleh et al. “OCA: Opinion Corpus for Arabic”. In: *Journal of the American Society for Information Science and Technology* 62.10 (2011), pp. 2045–2054.

[4] Mohamed Aly and Amir Atiya. “LABR: A Large Scale Arabic Book Reviews Dataset”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2013, pp. 494–498.

[5] Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. "SAMAR: Subjectivity and Sentiment Analysis for Arabic Social Media". In: *Computer Speech & Language* 28.1 (2014), pp. 20–37.

[6] Ashraf Elnagar, Yasmin S Khalifa, and Anas Einea. "Hotel Arabic-Reviews Dataset Construction for Sentiment Analysis Applications". In: *Intelligent Natural Language Processing: Trends and Applications*. Springer, 2018, pp. 35–52.

[7] Bashar Al Shboul, Mahmoud Al-Ayyoub, and Yaser Jararweh. "Multi-way Sentiment Classification of Arabic Reviews". In: *2015 6th International Conference on Information and Communication Systems (ICICS)*. IEEE. 2015, pp. 206–211.

[8] Islam Obaidat et al. "Enhancing the Determination of Aspect Categories and their Polarities in Arabic Reviews using Lexicon-based Approaches". In: *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*. IEEE. 2015, pp. 1–6.

[9] Nora Al-Twairesh, Hend Al-Khalifa, and Abdulmalik Al-Salman. "AraSenTi: Large-Scale Twitter-Specific Arabic Sentiment Lexicons". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 697–705. DOI: `10.18653/v1/P16-1066`. URL: `https://www.aclweb.org/anthology/P16-1066`.

[10] Aqil M Azmi and Samah M Alzanin. "Aara'–a System for Mining the Polarity of Saudi Public Opinion Through E-newspaper Comments". In: *Journal of Information Science* 40.3 (2014), pp. 398–410.

[11] Samhaa R El-Beltagy et al. "Combining Lexical Features and a Supervised Learning Approach for Arabic Sentiment Analysis". In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer. 2016, pp. 307–319.

[12] Ahmad Al Sallab et al. "Deep Learning Models for Sentiment Analysis in Arabic". In: *Proceedings of the second workshop on Arabic natural language processing*. 2015, pp. 9–17.

[13] Abdelghani Dahou et al. "Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification". In: *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*. 2016, pp. 2418–2427.

[14] Abdulaziz M Alayba et al. “A Combined CNN and LSTM Model for Arabic Sentiment Analysis”. In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer. 2018, pp. 179–191.

[15] Nora Al-Twairesh et al. “Arasenti-tweet: A Corpus for Arabic Sentiment Analysis of Saudi Tweets”. In: *Procedia Computer Science* 117 (2017), pp. 63–72.

[16] Mahmoud Nabil, Mohamed Aly, and Amir Atiya. “ASTD: Arabic Sentiment Tweets Dataset”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2515–2519.

[17] Jalal Omer Atoum and Mais Nouman. “Sentiment Analysis of Arabic Jordanian Dialect Tweets”. In: *(IJACSA) International Journal of Advanced Computer Science and Applications* 10 (2019), pp. 256–262.

[18] Ammar Mohammed and Rania Kora. “Deep Learning Approaches for Arabic Sentiment Analysis”. In: *Social Network Analysis and Mining* 9.1 (2019), p. 52.

[19] Eshrag Refaee and Verena Rieser. “Evaluating Distant Supervision for Subjectivity and Sentiment Analysis on Arabic Twitter Feeds”. In: *Proceedings of the EMNLP 2014 workshop on Arabic natural language processing (ANLP)*. 2014, pp. 174–179.

[20] Alec Go, Richa Bhayani, and Lei Huang. “Twitter Sentiment Classification using Distant Supervision”. In: *CS224N Project Report, Stanford* 1.12 (2009), p. 2009.

[21] Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete. “Combining Strengths, Emotions and Polarities for Boosting Twitter Sentiment Analysis”. In: *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*. ACM. 2013, p. 2.

[22] Hassan Saif, Yulan He, and Harith Alani. “Alleviating Data Sparsity for Twitter Sentiment Analysis”. In: CEUR Workshop Proceedings (CEUR-WS. org). 2012.

[23] Akshat Bakliwal et al. “Mining Sentiments from Tweets”. In: *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*. 2012, pp. 11–18.

[24] Michael Speriosu et al. “Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph”. In: *Proceedings of the First workshop on Unsupervised Learning in NLP*. Association for Computational Linguistics. 2011, pp. 53–63.

[25] Ghadah Alwakid, Taha Osman, and Thomas Hughes-Roberts. "Challenges in Sentiment Analysis for Arabic Social Networks". In: *Procedia Computer Science* 117 (2017), pp. 89–100.

[26] Anastasia Giachanou and Fabio Crestani. "Like It or Not: A Survey of Twitter Sentiment Analysis Methods". In: *ACM Comput. Surv.* 49.2 (June 2016), 28:1–28:41. ISSN: 0360-0300. DOI: `10.1145/2938640`. URL: `http://doi.acm.org/10.1145/2938640`.

[27] Petra Kralj Novak et al. "Sentiment of Emojis". In: *PLoS ONE* 10.12 (2015), e0144296. URL: `http://dx.doi.org/10.1371/journal.pone.0144296`.

[28] Philip Resnik and Jimmy Lin. "11. Evaluation of NLP Systems". In: *The handbook of computational linguistics and natural language processing* 57 (2010).

[29] Limin Yao, Sebastian Riedel, and Andrew McCallum. "Collective Cross-Document Relation Extraction without Labelled Data". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2010, pp. 1013–1023.

[30] Raphael Hoffmann et al. "Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics. 2011, pp. 541–550.

[31] David Yarowsky. "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods". In: *33rd annual meeting of the association for computational linguistics*. 1995, pp. 189–196.

[32] Steven Abney. "Bootstrapping". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 360–367.

[33] Wei Gao et al. "Semi-Supervised Sentiment Classification with Self-training on Feature Subspaces". In: *Workshop on Chinese Lexical Semantics*. Springer. 2014, pp. 231–239.

# 13

# Study 7: Comparing the Pre-trained language models and the feature based approaches on Dialect Identification and Sentiment Analysis

*Kathrein Abu Kwaik, Stergios Chatzikyriakidis, Simon Dobnik*

*"Pre-trained models or feature engineering? The case of Arabic Dialectal Identification and Sentiment Analysis"*

Recently with the increasing usage of the Internet and social media platforms especially in the Arab world, the work on Arabic Natural Language Processing (ANLP) has been increasing. Various types of resources have been introduced and built. In addition, researchers investigating different ANLP tasks, as there is a great trend towards processing Arabic dialects where dialects are the spoken and un-official language among Arabs. Lots of models and algorithms have been utilized for the purpose of Dialectal Arabic Natural Language Processing (DANLP), so in this paper we conduct a comparison study between some of the well-known and most commonly used methods in NLP and how they perform in different Arabic corpora and in two NLP tasks: Dialect Identification and Sentiment Analysis. We introduce different Bidirectional Encoder Representations from Transformers (BERT) models such as (Multilingual-BERT, Ara-BERT, and Twitter-Arabic-BERT). Moreover, we also study the performance of traditional Machine Learning approaches with features and the usage of pre-trained word-embeddings in Deep Learning networks like Long Short Term memory (LSTM). We compare the results with BERT models. We find that using feature-based classification can compete BERT models. On the other hand, using BERT is a good solution for dialectal Arabic that can save time and be applied on small amounts of data.

## 13.1 Introduction

The last decade has seen not only the emergence and development of social media platforms, but also, and relating to the latter, an increased interest in the automatic processing of Arabic dialects. A number of researchers have investigated several tasks related to Dialectal Arabic (DA) Natural Language Processing (NLP) that range from purely theoretical issues of syntax and morphology [1, 2] to more applied tasks like language generation and machine translation [3, 4, 5].

Notwithstanding the interest in processing the dialects, this is still in its developing stage and the lack of significant and valuable resources is well-known. Nowadays, much of the NLP research handles the problem of Dialectal Arabic by introducing and building different kind of resources, e.g. lexicons, corpora, tree-banks and others that are usually focused on a specific task they are addressing [6]. Dialectal Arabic resources are still suffering from the lack of available data that would enable a full investigation of the newly introduced Deep Learning (DL) networks on it.

Furthermore, the available works that support DA differ in terms of the tasks they are concerned with and the datasets used, a fact that leads to different results that are hard to compare. Some researchers and developers still support the use of traditional ML techniques in Arabic NLP given the limited size of available corpora [7], while others try to overcome and fine-tuned complex DL networks[8]. In case of limited size corpora, feature based ML approaches still give better results than DNNs where they mainly depend on huge amount of data[9]. In this paper, we investigate the performance of different approaches on DA using two NLP tasks: Dialect Identification (DI) and Sentiment Analysis (SA). We explore various datasets that have different sizes, balanced and imbalanced, hand-crafted and user-generated. In addition, we employ several features such as N-gram language models, pre-trained word embeddings, and the recently introduced pre-trained language model BERT [10]. For classification tasks we try traditional ML algorithms like Support Vector Machine (SVM), Fully connected dense layers and Long Short Term Memory LSTM-DL networks. We outperform SOTA for Sentiment Analysis corpora. In addition, it is considered one of the little researches that applied BERT on Dialect Identification tasks.

The paper is organised as follows: Section 2 discusses the recently related works in DI and SA, while Section 3 introduces the datasets used throughout the experiments. Section 4 presents the experiments, settings and, results. Discussion Section is presented in 5, while conclusions can be

found in Section 6.
Place licence statement here for the camera-ready version. See Sectionof the instructions for preparing a manuscript.

## 13.2 Related work

As mentioned, in this paper we focus on two kinds of NLP tasks: Dialect Identification and Sentiment Analysis. Three main approaches are presented through this research: (i) traditional ML with feature engineering, (ii) straightforward LSTM DL architectures and (iii) pre-trained language models.

As regards Dialect Identification, the vast majority of research uses traditional ML with feature engineering [11, 12, 13]. Recently, due to the introduction of the MADAR corpus [14], which covers 25 Arabic dialects, a good amount of research followed. Salameh et. al ; bouamor2019madar presented a fine grained Dialect Identification model, where they use a character-gram language model with Multinomial Naive Bayes (MNB) classifier to identify the label of 25 dialects. At the MADAR shared task , the top five ranked systems were focusing on traditional character feature classification [15, 16, 17, 18]. All papers conclude that neural methods did not do as well which is likely result of the limited training data. Other researchers have turned towards straightforward DL architectures. For example, in [19] the authors propose a deep learning CNN network based on character feature extraction to distinguish among MSA and dialects, while de Francony et al. de2019hierarchical, comparing two approaches for Arabic fine-grained Dialect Identification, use an RNN (BLSTM, BGRU) with hierarchical classification and the voting classifier approach based on NB and Random Forest. In the same track, [20] try different combinations of deep learning networks with different kinds of features on the MADAR corpus. Both works conclude that traditional machine learning algorithms outperform deep learning networks arguing that this might be because of the small size of the used corpus.

Even though most works for Dialect Identification are feature-based machine learning models, there are some works recently that investigate the use of pre-trained language models such as BERT. In [21, 22, 23], different Dialect Identification models based on BERT were introduced for MADAR [24] and NADI shared tasks[1].

Sentiment Analysis is a supervised classification task where any proposed model should be able to classify a sentence into two or more sentiments classes. The dominant approach for Arabic Sentiment Analysis in

[1] https://sites.google.com/view/nadi-shared-task

the last couple of years has been the feature-based and language modelling approach using machine learning classification algorithms like SVM, Multinomial Naive Bayes Classifier and others [25, 26, 27, 28, 29]. Some works used linguistics features such as the Stem, Lemma, POS tagging, in addition to the Arabic variety (MSA,dialect) and some used more specific features depending on the kind of the dataset, e.g. the userID (person, organisation) and the gender of the user found in datasets that use Twitter data[7, 30]. However, most of the works used the language models by extracting words and character n-grams and investigating different machine learning classifiers [31].

Recently, researchers and developers started using the deep learning networks for Sentiment Analysis with word embeddings and pre-trained language models. A CNN feature extractor and transformation network was proposed in [32] to determine the sentiment of Algerian users' comments on various Facebook brand pages of companies in Algeria, while [33] presented an LSTM network with pre-trained word embeddings to build a 5-scale Sentiment Analysis model for 4 Arab dialects.A combination of word and document embeddings in addition to a set of semantic features were used in [34] for Arabic tweets. The features are applied into CNN-LSTM networks followed by a fully connected layer. Moreover; Heikal et al., heikal2018sentiment proposed an ensemble DL model that combines LSTM with CNN to predict the sentiment class of Arabic tweets exploiting Arabic Sentiment Tweets Dataset (ASTD). More recently, deep LSTM-CNN networks were presented in [35]. They introduced their 40K-tweets dataset which was collected from Twitter focusing on Egyptian dialects. Likewise, [36] proposed a DL model that uses AraVec word embeddings with two Bi-LSTMs followed by 15 parallel CNN layers.

After introducing BERT, many works built and train their dialectal models by applying Arabic BERT as a first layer on the model instead of word-embeddings layer. In [37], they proposed the Transformer-based Model for Arabic Language Understanding called AraBERT. In their work they applied AraBERT on different Dialectal Arabic NLP tasks such as Sentiment Analysis and Question Answering tasks. Some works built their own dialectal BERT to be able to train their models, such as DziriBERT for Algerian dialects [38] and ARabiziBERT where Arabizi is a written form of spoken Arabic, relying on Latin characters and digits [39].

Despite the large number of works presented in the field of ANLP on Dialect Identification and Sentiment Analysis, it is still difficult to establish whether using old-fashion machine learning algorithms with hard feature engineering is better than using more sophisticated deep learning networks with pre-trained models. This is because the accuracy or the F-score between works varies according to the corpora in use. The used datasets differ in terms of size or the covered dialects, classification methods, or even the quality of the dataset. In this paper we make a comparison using the same corpora and by applying different approaches to make a reasonable conclusion about which is better for Arabic dialect NLP at the present moment and with the present datasets available.

## 13.3 Dataset

We will use the following existing corpora that were used in the development of Arabic NLP systems and compare the performance of systems using them. For the task of Dialect Identification we use three well-known corpora as follow:

- PADIC [4]: a Parallel Arabic Dialect Corpus (PADIC), that was collected from Algerian telephone conversations, transcribed and then translated to the other dialects. It is composed of 6,400 sentences for each dialects. The corpus contains five dialects where two of them present Algerian dialects (Algeria, Annaba), one from Tunisia and two dialects from Levantine (Palestine, Syria), in addition to MSA.
- SHAMI [40]: a Levantine dialect corpus, includes 66,251 documents which were collected from different domains such as sports, social life, cooking, and others and it covers the four Levantine dialects. The corpus is unbalanced in term of number of documents per dialect with 10,830, 37,760, 10,643, 7,018 for Lebanese, Syrian, Palestinian and Jordanian respectively.
- MADAR-6 [14]: is a parallel corpus in the travel domain and covers in addition to MSA five different Arab dialects from five Arabic cities: Beirut (BEI), Cairo (CAI), Doha (DOH), Rabat (RAB), Tunisia (TUN), therefore it is called MADAR-6. The corpus is composed of 10,000 documents for each dialect.

For dialectal Arabic Sentiment Analysis task we focus on binary classification where the document is classified as either positive polarity or negative polarity. The three used corpora are:

- ATSAD [41]: it is an Arabic Tweets Sentiment Analysis Dataset (multi-dialects). The corpus has been collected from Twitter during April 2019 and employs emojis as seeds for extraction of candidate instances. It is a balanced binary corpus which was partly annotated by human experts and then self training techniques were applied to annotate the rest of tweets.

- 40-K tweets [35]: An Egyptian binary balanced corpus where all the tweets were pre-processed and cleaned manually by two experts. The total size is 40,000 tweets.

- ASTD [42]: it is an Arabic Sentiment Tweets Dataset focusing on the Egyptian dialect. The corpus is composed of 10k tweets classified for objective and subjective sentiment. It is un-balanced dataset since there are 1681 negative documents and 818 positive ones.

## 13.4 Experiments

In this section we describe our experiments and the proposed model for both Dialect Identification and Sentiment Analysis on dialectal Arabic. For both tasks, we make use of three corpora as shown in the previous section. We split the datasets into 90% for training set and 10% for testing. The 90% training part is further split into 80% for training and 20% for validation. Tables 13.1 and 13.2 show the total size of the corpora in concern alongside the number of sentences for every set: training, validation, and testing.

| **Dataset** | **# Dialects** | **Total size** | **Train_set** | **Val_set** | **Test_set** |
|---|---|---|---|---|---|
| **PADIC** | 6 | 33,502 | 25,560 | 6,391 | 3551 |
| **Shami** | 4 | 66,251 | 47,699 | 11,925 | 6,626 |
| **MADAR-6** | 6 | 60,000 | 43,200 | 10,800 | 6,000 |

Table 13.1: Corpora statistics for the Dialect Identification task

| **Dataset** | **Total size** | **Train_set** | **Val_set** | **Test_set** |
|---|---|---|---|---|
| **ATSAD** | 22,542 | 16,229 | 4,058 | 2,255 |
| **40-K tweets** | 40,000 | 28,800 | 7,200 | 4,000 |
| **ASTD** | 2499 | 1,799 | 450 | 250 |

Table 13.2: Corpora statistics for the Sentiment Analysis task

We investigate the performance and the differences among various approaches. Many experiments have been done, however we tried to make the experiments as simple as possible and not to introduce a sophisticated models. first of all we apply BERT as a pre-trained language model followed by a classification layer. Then we compare it with a pre-trained word embeddings (AraVec) [43] and an LSTM network. On the other hand we investigate the performance of feature extraction language model on both old-fashion machine learning algorithms like SVM and on a fully connected

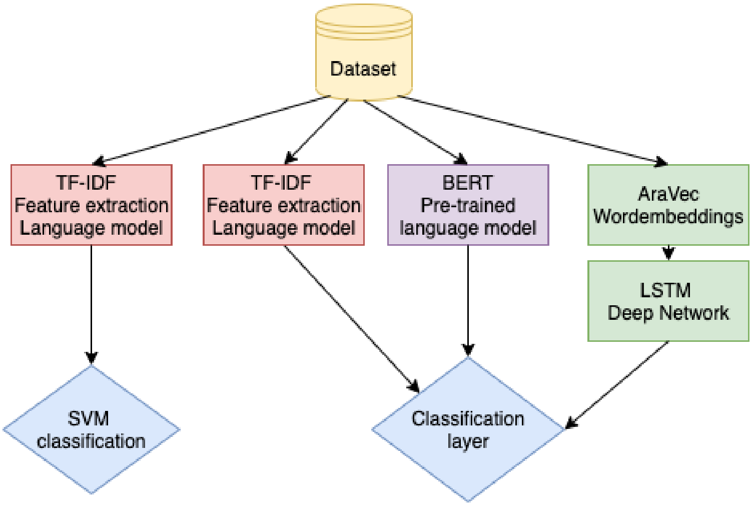

Dataset
TF-IDF Feature extraction Language model
TF-IDF Feature extraction Language model
BERT Pre-trained language model
AraVec Wordembeddings
LSTM Deep Network
SVM classification
Classification layer

Figure 13.1: Fours different approaches have been done for the classification tasks through the experiments

classification layer. Figure13.1 shows a diagram summarising all the experiments.

In order to evaluate the performance of the proposed models; we use the accuracy in addition to the the following two measures:

- Mathews correlation coefficient (MCC): It is used in machine learning to measure the quality of classification model [44]. It is a balanced measure which could also be used for imbalanced classification problem [45]. The MCC has a value between -1 (total disagreement between prediction and observation) to +1 (perfect predication), 0 value indicate random prediction. MCC is calculated from the confusion matrix according to equation 13.1

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{13.1}$$

- F-score measure: is a well-known measure for classification success in machine learning. The F-score is the arithmetic mean between the precision and the recall [46].

### 13.4.1 BERT for Dialectal Arabic

BERT or Bidirectional Encoder Representations from Transformers (BERT), has recently been introduced by Google AI Language researchers [10]. The main component of BERT is the Transformer which is an encoder-decoder attention mechanism that has been build to learn the contextual relations between sequence of words in any text and generate a language model.[2]

Figure 13.2 explains the high-level architecture of BERT. It takes a sequence of words (sub-words) as an input layer. These tokens are embedded into vectors and then go through the transformer encoder. The output of BERT is a sequence of vectors, where each vector presents an input token. To apply fine-tuning on BERT for any classification task or language generation task, a fully connected classification layer with soft-max activation function is built on top of the output vectors. Figure 13.3 shows how a simple Sentiment Analysis classification model can be built using BERT.

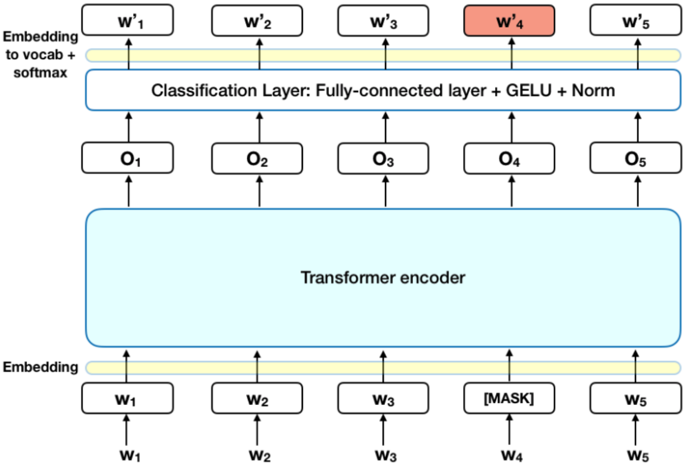


Figure 13.2: A high-level architecture of BERT [3]

As we work on Dialectal Arabic, then we use the Arabic versions of BERT. In the top of BERT model we add a classification layer for our two tasks which are trained separately. We use the following BERT models:

1. Multilingual-BERT[6]: This is the multi-lingual version of BERT, where

[2] For more technical details see [10].

[3] Source:https://mc.ai/bert-explained-state-of-the-art-language-model-for-nlp/

[5] Source:http://jalammar.github.io/illustrated-bert/

[6] https://github.com/google-research/bert/blob/master/multilingual.md

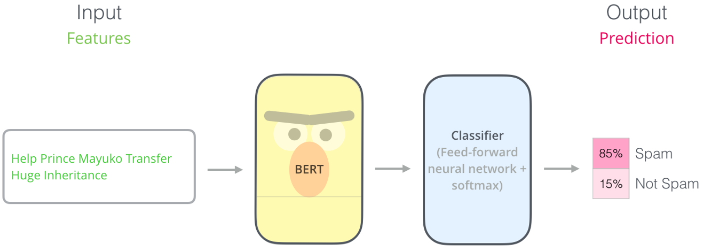

Input
Features
Help Prince Mayuko Transfer Huge Inheritance
BERT
Classifier
(Feed-forward neural network + softmax)
Output
Prediction
85%
Spam
15%
Not Spam

Figure 13.3: Using Bert for machine learning classification tasks like Sentiment Analysis [5]

it contains the top 100 languages with the largest Wikipedia content, including Arabic.

2. Arabic-BERT [47]: consists of 4 models of different sizes (Large,Base,Medium, and Mini), we use the base model for the experiments. Arabic-BERT has been built with 8,2B words from the OSCAR data [48] and the recent data dump from Wikipedia.
3. AraBERT-Twitter-base [37]: AraBERT is an Arabic pre-trained language model based on Google's BERT architecture as well as it uses the same BERT-base configuration. There are two versions of AraBERT v1 and v2 where they differ in term of segmentation techniques. AraBERT-Twitter-base is the dialectal version of AraBERTv2. It contains 60M Multi-Dialect Tweets in addition to 200M from the AraBERTv2-base.

The same major settings are used through all the experiments in order to get a reasonable comparison. We used Adam optimizer where learning rate is 5e-5 and epsilon equal to 1e-8 through all the BERT models. We set the batch size to be 32 for multi-lingual BERT and 16 for both Arabic BERT and Twitter-AraBERT models. The preferred number of epochs for fine tuning Multi-BERT is between 2 to 4 epochs [10], in our case 4 epochs were the best for Sentiment Analysis, while for Dialect Identification the best number of epochs was 10 epochs. For Arabic BERT and Twitter-AraBERT the number of epochs is between 8 to 10 and we employ early stopping and save the best performed epoch. We also explore some max sequence lengths for both tasks and end up using 77 for Sentiment Analysis and 130 for Dialect Identification using the Mutli-lingual BERT, 280 and 256 for ArabicBERT and Twitter-AraBERT respectively.

We build the first model by employing the Multi-Lingual BERT as a basic layer, and then we have a soft max fully connected layer for classification purpose. Table 13.3 and table 13.4 present the output results for the Accuracy, MCC and F-score for the two tasks.For Dialect Identification the accuracy ranges between 0.72 to 0.89 with 10 epochs and very short training time compared to end-to-end neural network. In case of Sentiment Analysis we get an accuracy between 0.8 to 0.83 where the model outperforms the state of the art result for ASTD corpus for example in the field of deep learning [8, 36].

| **Data_set** | **Accuracy** | **MCC** | **F-score** |
|---|---|---|---|
| **Padic** | 0.72 | 0.67 | 0.72 |
| **Shami** | 0.88 | 0.81 | 0.83 |
| **MADAR-6** | 0.89 | 0.87 | 0.89 |

Table 13.3: Results of applying Multilingual-BERT on Dialect Identification task

| **Data_set** | **Accuracy** | **MCC** | **F-score** |
|---|---|---|---|
| **ATSAD** | 0.80 | 0.6 | 0.80 |
| **40-k Tweets** | 0.83 | 0.66 | 0.83 |
| **ASTD** | 0.81 | 0.51 | 0.75 |

Table 13.4: Results of applying Multilingual-BERT on Sentiment Analysis task

As Multi-Lingual model is not a model that was only built for the purpose of Arabic-NLP, thus we implement the second model by the help of Arabic-Bert. We used the basic version of Arabic-BERT and then the same soft-max classification layer to solve the task. The three test measurements (Accuracy, MCC, F-scores) for Dialect identification and Sentiment Analysis are presented in tables 13.5 and 13.6 respectively. The accuracy for DI models have been ranged from 0.71 to 0.80 which is less than the previous Multi-lingual BERT model. This is may be because that the later model was built and trained using different languages so it is easier to fine-tune it to a task where the purpose is identifying or classifying languages or dialects. In sentiment Analysis , the models performed better than Multilingual-BERT where the accuracy is in range of 0.83 to 0.90.

Both Multilingual BERT and Arabic-BERT have been trained on MSA data collected mainly from news websites and Wikipedia documents. We conduct a third experiment to measure the performance of the model when we use a dialectal BERT, so it is Twitter-AraBERT. Table 13.7 shows the output test accuracy for Dialect Identification task. It is clear the the model is the best among the BERT models have used previously. The accuracy now

| Data_set | Accuracy | MCC | F-score |
|---|---|---|---|
| **Padic** | 0.71 | 0.66 | 0.72 |
| **Shami** | 0.87 | 0.78 | 0.81 |
| **MADAR-6** | 0.80 | 0.76 | 0.80 |

Table 13.5: Results of applying Arabic-BERT on Dialect Identification task

| Data_set | Accuracy | MCC | F-score |
|---|---|---|---|
| **ATSAD** | 0.93 | 0.87 | 0.93 |
| **40-k Tweets** | 0.83 | 0.66 | 0.83 |
| **ASTD** | 0.84 | 0.63 | 0.81 |

Table 13.6: Results of applying Arabic-BERT on Sentiment Analysis task

is in the range of (0.77 and 0.91). In table 13.8 the accuracy for Sentiment Analysis models ranged from (0.88 to 0.97). The model outperforms the SOTA for 40K tweets dataset [35] where they achieved an avg-accuracy of 81 using LSTM models. In addition it outperforms the SOTA for ASTD corpus [8]. Among the three BERT models, Twitter-AraBERT is the best performing models where the data is mostly dialectal where the BERT-base it is mostly MSA data.

| Data_set | Accuracy | MCC | F-score |
|---|---|---|---|
| **Padic** | 0.77 | 0.73 | 0.77 |
| **Shami** | 0.91 | 0.86 | 0.87 |
| **MADAR-6** | 0.91 | 0.90 | 0.91 |

Table 13.7: Results of applying Twitter-AraBERT on Dialect Identification task

| Data_set | Accuracy | MCC | F-score |
|---|---|---|---|
| **ATSAD** | 0.97 | 0.94 | 0.97 |
| **40-k Tweets** | 0.91 | 0.82 | 0.91 |
| **ASTD** | 0.88 | 0.74 | 0.87 |

Table 13.8: Results of applying Twitter-AraBERT on Sentiment Analysis task

### 13.4.2 LSTM Baseline

In order to evaluate our work, we build a simple LSTM baseline and apply it to the corpora for the two tasks. In the LSTM baseline we employ the AraVec which is a pre-trained Arabic word embeddings as a first layer [43], followed by an LSTM layer with 70 nodes and a dropout of 0.25%. This is followed by a fully connected dense layer with 30 nodes. The last layer is also a fully connected dense layer where the output depends on the number of classes in each task. For Dialect Identification, there are 6, 6 and 4 output classes for PADIC, MADAR-6 and SHAMI respectively. For the Sentiment Analysis task, it is a binary classification task. Table 13.9 shows the LSTM baseline settings.

| | |
|---|---|
| **Max_length** | 130 (DI), 77 (SA) |
| **Optimiser** | Adam (DI), RMSprop(SA) |
| **Word_embeddings** | AraVec 300 |
| **LSTM_nodes** | 70 |
| **Drop_out** | 0.25 |
| **Dense_nodes** | 30 |
| **Activation_function** | Sigmoid |
| **Loss** | Categorial_crossentrapy (DI),<br>Binary_crosentrapy (SA) |
| **Batch_size** | 32 |
| **Epochs** | up to 100, Eaarly_stopping |

Table 13.9: LSTM baseline network settings

We use the loss with a minimum value to monitor the model and to save the best performance weights. Tables 13.10 and 13.11 show the results of applying the baseline into the corpora in concern. It is clear that a baseline LSTM with Arabic pre-trained word embeddings is not able to perform well with dialectal Arabic NLP tasks. The accuracy does not exceed 0.6 in any corpus. Moreover, the MCC shows zero values through all the corpora which means that the classifier is not able to correctly classify the documents and it is no better than a random prediction. For Shami corpus the accuracy is high (comparing to other datasets) while the F-score is equally low 0.18, which suggests that Shami is more imbalanced and the model is not doing well on recall on minority classes.

| **Data_set** | **Accuracy** | **MCC** | **F-score** |
|---|---|---|---|
| **PADIC** | 0.17 | 0 | 0.14 |
| **Shami** | 0.57 | 0.004 | 0.18 |
| **MADAR-6** | 0.17 | 0 | 0.29 |

Table 13.10: Results of applying LSTM baseline on Dialect Identification task

| **Data_set** | **Accuracy** | **MCC** | **F-score** |
|---|---|---|---|
| **ATSAD** | 0.52 | 0 | 0.34 |
| **40-k Tweets** | 0.49 | 0 | 0.33 |
| **ASTD** | 0.69 | 0 | 0.41 |

Table 13.11: Result of applying LSTM baseline on Sentiment Analysis task

### 13.4.3 Feature-Based Classification for Dialectal Arabic

In addition to BERT and LSTM experiments we also investigate the performance of traditional Machine Learning on Dialectal Arabic. An SVM machine learning model have been built which was mainly proposed in [9] for dialectal Arabic Sentiment Analysis. We employ the same approach for both tasks. The models apply various n-gram features as follows:

- Word-gram features with uni-gram, bi-grams and tri-grams, the transformation weight is 0.8.
- Character-gram features with word boundary consideration from bigrams to 5-grams and the transformation weight of 0.5
- Character-gram features without word boundary consideration from bi-grams to 5-grams and the transformation weight of 0.4.

The results after training and testing the model are presented in Tables 13.12 and 13.13.

| **Data_set** | **Accuracy** | **MCC** | **F-score** |
|---|---|---|---|
| **PADIC** | 0.72 | 0.66 | 0.72 |
| **Shami** | 0.90 | 0.84 | 0.86 |
| **MADAR-6** | 0.89 | 0.87 | 0.89 |

Table 13.12: Results of applying the ML model on Dialect Identification task

| Data_set | Accuracy | MCC | F-score |
|---|---|---|---|
| **ATSAD** | 0.96 | 0.92 | 0.96 |
| **40-k Tweets** | 0.84 | 0.67 | 0.84 |
| **ASTD** | 0.80 | 0.45 | 0.71 |

Table 13.13: Results of applying the ML model on Sentiment Analysis task

We further investigate the effect of feature based approach by placing a fully connected classification layer on the top of the language model rather than using a traditional machine learning algorithm such as SVM or NB. The model seems like BERT, but instead of the pre-trained language model layers we use the feature extraction language model discussed before, followed by a classification layer. Table 13.14 and Table 13.15 show the results for the experiment.

| Data_set | Accuracy | MCC | F-score |
|---|---|---|---|
| **PADIC** | 0.73 | 0.68 | 0.74 |
| **Shami** | 0.57 | 0 | 0.50 |
| **MADAR-6** | 0.89 | 0.87 | 0.89 |

Table 13.14: Results of the feature-based model with fully connected classification layers on Dialect Identification task

| Data_set | Accuracy | MCC | F-score |
|---|---|---|---|
| **ATSAD** | 0.96 | 0.91 | 0.95 |
| **40-k Tweets** | 0.82 | 0.63 | 0.82 |
| **ASTD** | 0.77 | 0.49 | 0.73 |

Table 13.15: Results of the feature-based model with fully connected classification layers on Sentiment Analysis task

## 13.5 Discussion

It is quite obvious that the LSTM model is the worst among all the models with a huge evaluation gap between it and the other models. The low performance of the LSTM network might be due to the usage of the AraVec pre-trained word embeddings. The percentage of OOV words is high (from 30% to 70%). This makes the LSTM network does not perform well even when the word embeddings layer is set to be trainable. The network also biases to the majority class and that was very clear in the case of SHAMI which is the most unbalanced dataset among the others.

As we see from the experiments that feature-based classification methods are competing the approach using pre-trained language models followed by a fully connected layer. It is worthy mentioned that the features engineering approach can still have the ability to compete deep learning models as well as outperforming in some case. Figure 13.4 plots the accuracy for the Dialect Identification models as well as the used corpora, while we ignore the LSTM experiment as it is the worst and the MCC was 0. Although the results were close to each other in some cases, however, the Twitter-AraBERT outperforms all the models on all corpora. The Twitter-AraBERT model and the ML models are close to each other in terms of accuracy especially for SHAMI where it is an un-parallel corpus as well as unbalanced. It is clear that the size of the corpus has an effect on the performance of the DI task, for example, SHAMI and MADAR both are doing better applying ML than PADIC, Moreover for a corpus of reasonable size, even with unbalanced data like SHAMI, ML algorithm (SVM) has the ability to compete the pre-trained language model. Regarding well structured and human annotation corpora as PADIC and MADAR, therefore both feature-based approaches do nearly the same regardless they use SVM or classification layer. Both corpora have been handcrafted that increase the power of N-grams language models.

Figure13.5 plots the accuracy for the Sentiment Analysis and the applied corpora. Twitter-Arabert is the best also over all the corpora. Sentiment Analysis is a task that does not depend on the language as much as it depends on the context where feelings are expressed. On the ATSAD corpus which is a corpus that used the emojis as weakly labels to collect and annotate it, then Twitter-AraBERT performs very high. Twitter-AraBERT is also able to deal efficiently with the problem of un-balancing datasets like ATSD, however, it is a small size corpus. When it comes to the number of dialects in the corpus, 40k-tweets as well as ASTD both are Egyptian corpora, then Twitter-BERT is considered better than ML methods. In contrast, on the multi-dialect corpus for the purpose of Sentiment Analysis like ATSAD, then Feature-based approach with ML model is also a good choice as much as Twitter-BERT where they achieve very close results.

Generally Speaking, It is therefore clear that applying pre-trained language model on dialectal Arabic NLP tasks lead to reasonable results in terms of saving time, resources and achieves a good accuracy and F-score values with small amount of data. Many factors play a role on the decision of chose the best model to apply on an NLP tasks such as: The size of the dataset, the sources and the quality of the data, the data balance, if the corpus contains MSA or multi-dialectal Arabic and the number of classes.

In case of under-resource languages, if the traditional machine learning approaches performs well and competing, what is the need for very complicated as well as time and resource consuming deep learning networks?? Some times very simple approaches can do better and outperform the DL networks.

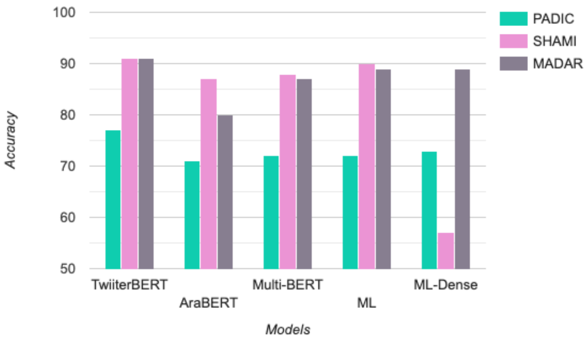


Figure 13.4: Accuracy of different Dialect Identification models

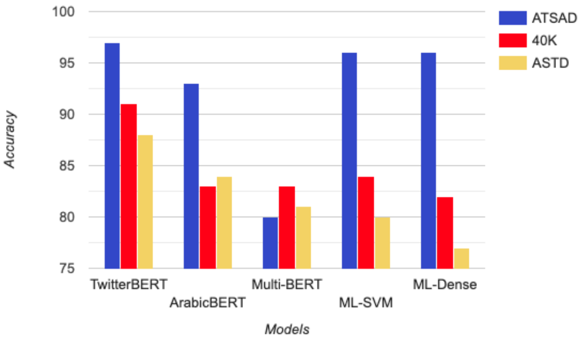


Figure 13.5: Accuracy of different Sentiment Analysis models

## 13.6 Conclusion

In this study we aim to model the way of choosing the best methods for a Dialectal Arabic NLP tasks, taking into consideration the differences among the resources. We implement various approaches from traditional ML to the most recent trendy approaches like BERT, using different corpora. We want to measure the worthy of applying complicated models and the performance of feature engineering methods. Firstly we propose the usage of a pre-trained language model like BERT into Dialectal Arabic. Two DA-NLP tasks were used in this study (Dialect Identification and Sentiment Analysis), in addition to six different corpora (3 for each task) were also explored. Fine-tuning BERT for DA cam produce an acceptable result for all the corpora. Using BERT that supports the Arabic language saves the effort and time to build deep learning models from scratch that need a huge amount of data and resources for training in order to be able to give reasonable and trusted results.

The second part of the study was to investigate other approaches and compare them to BERT. We build an LSTM baseline with the support of the pre-trained AraVec which unfortunately does not perform well. The usage of AraVec with a large OOV dialectal words does not facilitate the model in being retrained and fine-tuned for DA. On the other hand, we also build feature engineering approaches either with the SVM machine learning algorithm or with a fully connected neural network layer. The usage of a tailor-made feature extractor can compete sophisticated solutions like BERT and DL approaches. In summary, after investigating traditional and modern machine learning approaches, we can say that building a deep learning models from scratch do not considered a right way for modeling. BERT appears to be a good an reasonable solution to apply for dialectal Arabic tasks as it compete the tradition machine learning models and outperforming the deep learning models. However usage of new-modern proposed techniques does not necessarily mean getting better results all the time. Sometimes, as in our experiments, the use of simple traditional methods like ML svm algorithm does impressively well even compared to BERT and produce competitive results.

## 13.7 References

[1] David Chiang et al. “Parsing Arabic dialects”. In: *11th Conference of the European Chapter of the Association for Computational Linguistics*. 2006.

[2] Nizar Habash, Owen Rambow, and George Anton Kiraz. “Morphological Analysis and Generation for Arabic Dialects”. In: *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*. 2005, pp. 17–24.

[3] Rabih Zbib et al. "Machine Translation of Arabic Dialects". In: *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies.* Association for Computational Linguistics. 2012, pp. 49–59.

[4] Karima Meftouh et al. "Machine Translation Experiments on PADIC: A Parallel Arabic Dialect Corpus". In: *The 29th Pacific Asia conference on language, information and computation.* 2015.

[5] Mona Diab and Nizar Habash. "Natural Language Processing of Arabic and its Dialects". In: *Proceedings of the conference on empirical methods in natural language processing (EMNLP). Doha, Qatar.* Citeseer. 2014, p. 10.

[6] Imane Guellil et al. "Arabic Natural Language Processing: An Overview". In: *Journal of King Saud University-Computer and Information Sciences* (2019).

[7] Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. "SAMAR: Subjectivity and Sentiment Analysis for Arabic Social Media". In: *Computer Speech & Language* 28.1 (2014), pp. 20–37.

[8] Maha Heikal, Marwan Torki, and Nagwa El-Makky. "Sentiment Analysis of Arabic Tweets using Deep Learning". In: *Procedia Computer Science* 142 (2018), pp. 114–122.

[9] Chatrine Qwaider, Stergios Chatzikyriakidis, and Simon Dobnik. "Can Modern Standard Arabic Approaches be used for Arabic Dialects? Sentiment Analysis as a Case Study". In: *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics.* 2019, pp. 40–50.

[10] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Ridouane Tachicart et al. "Automatic Identification of Moroccan colloquial Arabic". In: *International Conference on Arabic Language Processing.* Springer. 2017, pp. 201–214.

[12] Ossama Obeid et al. "ADIDA: Automatic Dialect Identification for Arabic". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations).* Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 6–11. DOI: `10.18653/v1/N19-4002`. URL: `https://www.aclweb.org/anthology/N19-4002`.

[13] Omar F Zaidan and Chris Callison-Burch. "Arabic Dialect Identification". In: *Computational Linguistics* 40.1 (2014), pp. 171–202.

[14] Houda Bouamor et al. "The MADAR Arabic Dialect Corpus and Lexicon". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[15] Kathrein Abu Kwaik and Motaz K Saad. "ArbDialectID at MADAR Shared Task 1: Language Modelling and Ensemble Learning for Fine Grained Arabic Dialect Identification". In: *ArbDialectID at MADAR Shared Task 1: Language Modelling and Ensemble Learning for Fine Grained Arabic Dialect Identification* Proceedings of the Fourth Arabic Natural Language Processing Workshop (2019).

[16] Karima Meftouh et al. "The SMarT Classifier for Arabic Fine-Grained Dialect Identification". In: 2019.

[17] Ahmad Ragab et al. "Mawdoo3 AI at MADAR Shared Task: Arabic Fine-Grained Dialect Identification with Ensemble Learning". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019, pp. 244–248.

[18] Pruthwik Mishra and Vandan Mujadia. "Arabic Dialect Identification for Travel and Twitter Text". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019, pp. 234–238.

[19] Mohamed Ali. "Character Level Convolutional Neural Network for Arabic Dialect Identification". In: *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. 2018, pp. 122–127.

[20] Youssef Fares et al. "Arabic Dialect Identification with Deep Learning and Hybrid Frequency Based Features". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019, pp. 224–228.

[21] Chiyu Zhang and Muhammad Abdul-Mageed. "No Army, No Navy: Bert Semi-supervised Learning of Arabic Dialects". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. 2019, pp. 279–284.

[22] Bashar Talafha et al. "Multi-Dialect Arabic BERT for Country-Level Dialect Identification". In: *arXiv preprint arXiv:2007.05612* (2020).

[23] Ahmad Beltagy, Abdelrahman Wael, and Omar ElSherief. "Arabic Dialect Identification using BERT-Based Domain Adaptation". In: *arXiv preprint arXiv:2011.06977* (2020).

[24] Houda Bouamor, Sabit Hassan, and Nizar Habash. "The MADAR Shared Task on Arabic Fine-Grained Dialect Identification". In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*. Florence, Italy, 2019.

[25] Asmaa Mountassir, Houda Benbrahim, and Ilham Berrada. “An Empirical Study to Address the Problem of Unbalanced Data sets in Sentiment Classification”. In: *2012 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE. 2012, pp. 3298–3303.

[26] Mohamed Aly and Amir Atiya. “LABR: A Large Scale Arabic Book Reviews Dataset”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2013, pp. 494–498.

[27] Nazlia Omar et al. “Ensemble of Classification Algorithms for Subjectivity and Sentiment Analysis of Arabic Customers’ Reviews”. In: *International Journal of Advancements in Computing Technology* 5.14 (2013), p. 77.

[28] Rasheed M Elawady, Sherif Barakat, and Nora M Elrashidy. “Different Feature Selection for Sentiment Classification”. In: *International Journal of Information Science and Intelligent System* 3.1 (2014), pp. 137–150.

[29] Samar Al-Saqqa, Nadim Obeid, and Arafat Awajan. “Sentiment Analysis for Arabic Text using Ensemble Learning”. In: *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*. IEEE. 2018, pp. 1–7.

[30] Amira Shoukry and Ahmed Rafea. “Sentence-Level Arabic Sentiment Analysis”. In: *2012 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE. 2012, pp. 546–550.

[31] Rehab M Duwairi et al. “Sentiment Analysis in Arabic Tweets”. In: *2014 5th International Conference on Information and Communication Systems (ICICS)*. IEEE. 2014, pp. 1–6.

[32] Assia Soumeur et al. “Sentiment Analysis of Users on Social Networks: Overcoming the challenge of the Loose Usages of the Algerian Dialect”. In: *Procedia computer science* 142 (2018), pp. 26–37.

[33] Ramy Baly et al. “Comparative Evaluation of Sentiment Analysis Methods Across Arabic Dialects”. In: *Procedia Computer Science* 117 (2017), pp. 266–273.

[34] Malak Abdullah, Mirsad Hadzikadicy, and Samira Shaikhz. “SEDAT: Sentiment and Emotion Detection in Arabic Text Using CNN-LSTM Deep Learning”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 835–840.

[35] Ammar Mohammed and Rania Kora. “Deep Learning Approaches for Arabic Sentiment Analysis”. In: *Social Network Analysis and Mining* 9.1 (2019), p. 52.

[36] Kathrein Abu Kwaik et al. “LSTM-CNN Deep Learning Model for Sentiment Analysis of Dialectal Arabic”. In: *International Conference on Arabic Language Processing.* Springer. 2019, pp. 108–121.

[37] Wissam Antoun, Fady Baly, and Hazem Hajj. “Arabert: Transformer-based Model for Arabic Language Understanding”. In: *arXiv preprint arXiv:2003.00104* (2020).

[38] Amine Abdaoui et al. “Dziribert: a Pre-trained Language Model for the Algerian Dialect”. In: *arXiv preprint arXiv:2109.12346* (2021).

[39] Gaétan Baert et al. “Arabizi Language Models for Sentiment Analysis”. In: *Proceedings of the 28th International Conference on Computational Linguistics.* Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 592–603. DOI: `10.18653/v1/2020.coling-main.51`. URL: `https://aclanthology.org/2020.coling-main.51`.

[40] Kathrein Abu Kwaik et al. “Shami: A Corpus of Levantine Arabic Dialects”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* 2018.

[41] Kathrein Abu Kwaik et al. “An Arabic Tweets Sentiment Analysis Dataset (ATSAD) using Distant Supervision and Self Training”. In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection.* 2020, pp. 1–8.

[42] Mahmoud Nabil, Mohamed Aly, and Amir Atiya. “ASTD: Arabic Sentiment Tweets Dataset”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* 2015, pp. 2515–2519.

[43] Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. “AraVec: A Set of Arabic Word Embedding Models for use in Arabic NLP”. In: *Procedia Computer Science* 117 (2017), pp. 256–265.

[44] Brian W Matthews. “Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme”. In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), pp. 442–451.

[45] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. “Optimal Classifier for Imbalanced Data using Matthews Correlation Coefficient Metric”. In: *PloS one* 12.6 (2017), e0177678.

[46] Leon Derczynski. “Complementarity, F-score, and NLP Evaluation”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16).* 2016, pp. 261–266.

[47] Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. “Kuisail at Semeval-2020 task 12: BERT-CNN for Offensive Speech Identification in Social Media”. In: *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. 2020, pp. 2054–2059.

[48] Pedro Javier Ortiz Suárez, Laurent Romary, and Benoıt Sagot. “A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages”. In: *arXiv preprint arXiv:2006.06202* (2020).

**Chatrine Qwaider** (Kathrein Abu Kwaik) is a researcher in the field of Natural language processing (NLP) and Computational Linguistics. She is interested in Arabic Language processing and building Machine Learning models. She has many publications addressing Dialectal Arabic NLP tasks such as character recognition, text summarization, dialect identification, sentiment analysis and cross-dialectal studies.

Nowadays, she is an NLP research engineer at the Data Science Research Engineers (DSRE) at the Chalmers University of Technology in Sweden.

Department of Philosophy,
Linguistics and Theory of Science

ISBN: 978-91-8009-803-8 (PRINT)
ISBN: 978-91-8009-804-5 (PDF)