DOCTOR OF PHILOSOPHY

Categorisation of Arabic Twitter Text

Altamimi, Mohammed Hamed R

Award date:
2020

Awarding institution:
Bangor University

DOCTOR OF PHILOSOPHY

Categorisation of Arabic Twitter Text

Altamimi, Mohammed Hamed R

Award date:
2020

PRIFYSGOL
BANGOR
UNIVERSITY

School of Computer Science and

Electronic Engineering

Categorisation of Arabic Twitter Text

Mohammed Hamed Altamimi

Submitted in partial satisfaction of the requirements for the
Degree of Doctor of Philosophy
in Computer Science

Supervisor Dr. William J. Teahan

February 2020

Acknowledgements

First of all, I would like to express my profound and sincere appreciation to my supervisor, Dr. William J. Teahan, to whom I am very thankful for the ongoing guidance and advice which has been invaluable during this doctoral study as well as for my career and which is very much appreciated.

A special thanks to my parents for having faith in me and being there for me at all times. I would also like to thank all of my brothers and sisters for their continuous support and encouragement during my journey. Last but not least, I would like to express my gratitude to my beloved wife Sumayyah for her continuous support during the last six years that we have spent together. I am grateful to her for taking care of our precious kids Hamad and Anas, and still being here at the end.

Abstract

The shortage of Arabic language resources in the field of corpus linguistics compared to other popular languages such as English, Chinese and Spanish inspired this work. The research in the field of dialectal Arabic is still limited due to the relative unavailability of resources and the time-consuming nature of the task needed to create and process these corpora.

This thesis introduces the Bangor Twitter Arabic corpus (BTAC) that was created specifically using Arabic Twitter text. The corpus contains over 122K tweets. The tweets were annotated manually into five main dialects, Egyptian, Gulf, Iraqi, Maghrebi, and Levantine, in addition to Modern Standard Arabic and Classical Arabic. The resource has also identified written code-switching in single tweet which occurs between Modern Standard Arabic and Arabic dialects.

This thesis evaluates various methods for categorisation of Arabic Twitter text. The categorisation is performed on three main categorisation tasks: authorship attribution; gender categorisation; and dialect identification. The experiments are performed using the Prediction by Partial Matching (PPM) character-based text compression approach. Furthermore, well known algorithms were selected to perform the comparison using character-based and feature-based approaches such as Multinomial Naïve Bayes (MNB), K-Nearest Neighbours (KNN), and Support Vector Machine (SVM).

The results show that PPM outperforms traditional feature-based classifiers in most cases in terms of accuracy, precision, recall and F-measure. The results reported for classifying author multiple tweets achieved an accuracy of 88% for gender categorisation, an accuracy of 96% for authorship attribution, and an accuracy of 87% for dialect identification. In terms of single-tweet text categorisation, the results achieved an accuracy of 76% for gender categorisation, an accuracy of 77% for authorship attribution, and an accuracy of 74% for dialect identification. Further optimization using concatenated author models as the secondary class type improved

the classification accuracy for both the gender and dialect experiments, achieving an accuracy of 97% for gender categorisation and an accuracy of 98% for dialect identification.

We also investigated code-switching that often occurs in text acquired from social media. In this study we investigated code-switching between two variant linguistic systems from one language (Modern Standard Arabic and Arabic dialects). The purpose of the experiment was to detect the switch at the character level. An accuracy of 81.2% for detecting code-switching was obtained using 5-fold cross-validation on the full BTAC dataset.

Table of Content

List of Tables

List of Figures

List of Abbreviations

NLP	Natural Language Processing
ML	Machine Learning
PPM	Prediction by Partial Matching
MNB	Multinomial Naïve Bayes
SVM	Support Vector Machine
KNN	K-Nearest Neighbours
MSA	Modern Standard Arabic
CA	Classical Arabic
DA	Dialectal Arabic
BTAC	Bangor Twitter Arabic Corpus

Chapter 1

Introduction and Motivation

1.1. Background of the study

Text mining is the task of obtaining meaningful information from collections of text. Text mining has received significant attention in recent years due to the rapid growth of data available on the web over the past two decades. There are several text mining tasks and techniques including text pre-processing, categorisation and clustering. The goal of text mining is to convert this data into useful knowledge with the help of various techniques and algorithms in order to discover information (Allahyari *et al.*, 2017).

Text categorisation is the process of assigning documents to predefined categories. Recently, text categorisation has become popular due to the tremendous volumes of data available online and especially social media. Because of the global reach of social media, there are large amounts of data available to be categorised into many different languages. The motivation behind the research is to examine the text categorisation problem of data specifically from Twitter written text in the Arabic language.

Twitter is an online social network which people use to post their messages called “tweets”. A common characteristic of communication on online social networks, specifically Twitter, is that it occurs using short messages called “micro-blogging”, often using non-standard language variations. This type of text is considered challenging as stated, with many non-standard text variations such as acronyms, emoticons, and misspellings. Not only is the complexity of Twitter text challenging but also the linguistic structure of the Arabic language introduces several challenges such as inflection and diacritics. These characteristics make this type of text a challenging text for natural language processing (NLP).

Generally, since the Arab Spring in 2010, social media is increasingly used in the Middle East to express feelings in a range of topics such as political, social, financial and so forth. However, some of the users misuse the services for one reason or another; for example, it has been wrongly used to spread fake news and rumour to escalate the political crisis in the Middle East (Ritzen, 2019). Those users hide their true identity by using false names and posting misleading profile pictures. Furthermore, they hide their location, and sometimes use bots which are used to post from different locations to ensure further anonymity.

A motivation behind this research is to examine the text categorisation problem of the Arabic language through the context of Twitter text. We are also specifically interested in studying three different classification tasks using text collected from Twitter: gender categorisation; authorship attribution; and dialect identification. The goal for performing twitter text classification is to discover information about users by examining their Twitter timeline. To achieve this goal, a ground truth dataset is needed to train various models and evaluate their effectiveness. Although there are large numbers of Classical and Modern Standard Arabic language resources, the field of dialectal resources is still limited. Nevertheless, dialectal resources have witnessed growth recently by using web-based textual sourced data from social media websites. This massive increase of dialectal resources has provided the incentive to produce this work.

One area of interest is the role of gender in online communication. People may not expose their true identity by deliberately assigning a different gender or by hiding their gender information for malicious reasons. For example, recently there were a number of times where Twitter accounts were used by males pretending to be a female in order to participate in crucial debates in favour of feminism (Ulman, 2017). Furthermore, a well-known case that illustrates the need to study gender-related issues is the account of Sara Ibrahim. This account claimed to be for a teenage girl who was receiving treatment for leukaemia. The account posted daily information about Sara's case using photos of an American girl fighting bone cancer. The account drew the sympathy of

the public community, and was used to collect donations for the non-real Sara (Bakhsh, 2015).

We are also interested in studying authorship attribution. From an idiolect point of view, each person has their own style of language. Studying the author's writing style is a fundamental task used in forensic analyses. Authorship analysis contributes to verifying or rejecting the assumption that the given person is the author of the text, among other candidates. This could help forensic analysis involving the study of the writing style of a suspect. For example, in January 2010, the New York Daily News reported on a series of Twitter messages exchanged between two young friends. These messages directed one to murder the other. The exchanged Twitter messages between the two young friends were considered important evidence in the trial (Silva *et al.*, 2011).

We are also motivated to study the problem of dialect identification. The only present way to identify the location of the tweet is to use the geolocation of the tweet (latitude and longitude). This heavy reliance on the location of the tweet will overlook the language used in the tweet (the dialect) that will often be a better indicator of its origin. In January 2017, the BBC posted large networks of fake accounts found on Twitter. These accounts included three subtle features: tweets were posted from places where nobody lived; tweets were produced from a single source; and tweets were posted for a specific topic.

A primary motivation of this research is to investigate the effectiveness of classifying Arabic text acquired from social media using prediction by partial matching (PPM). PPM is investigated because of its excellent overall ability at both compression and classification. Previous research using English text that applies the character-based compression approach has shown promising results. This has provided part of the motivation to apply Arabic text categorisation using a character-based compression approach for this thesis and compare it against the machine learning approach.

1.2. Research aim and objectives

The aim of this research is to study the performance of a compression character-based approach against the current leading machine learning feature-based approach. A secondary aim is to demonstrate that compression character-based approach should be considered in all future comparative studies within the field of text categorisation and NLP. In order to achieve this aim, the specific objectives are as follows:

Objective 1:

To create a dialectal corpus for Arabic using Twitter text. The corpus will be annotated according to the five main Arabic dialects – Gulf, Egyptian, Levantine, Maghrebi, and Iraqi – in addition to Modern Standard Arabic and Classical Arabic. Also, code-switching or (mixed dialects) will be tagged for further text analysis.

The sub-objective is to review the current dialectal corpora of Arabic existing under specific criteria.

Objective 2:

To adapt the character-based approach based on compression employing Prediction by Partial Matching (PPM) for Arabic text categorisation (using the corpus that was created for Objective 1) in different applications such as authorship attribution, gender categorisation, and dialect identification.

Objective 3:

To evaluate the effectiveness of the method used (PPM) and compare it with other feature-based approaches using Machine learning algorithms.

The sub-objectives of the research can be broadly stated as follows:

- To investigate text categorisation using various N-gram features such as word-based N-grams (unigram, bigram, and trigram), and character-based N-grams (1-6).

- To explore text categorisation using single tweet vs. multiple author tweets. Single tweet categorisation classifies each tweet separately, whereas multiple tweet categorise several tweets from the same author.
- To investigate which order K of PPM is best for the Arabic text, where K is the number of preceding symbols used for prediction.

1.3. Research question

The overarching research question associated with this thesis is as follows:

How does the effectiveness of the character-based compression approach for text categorisation using Prediction by Partial Matching compare to that of commonly used machine learning algorithms in classifying Arabic Twitter text according to gender, authorship, and dialects?

1.4. Thesis contributions

This thesis has achieved several contributions in the fields of text categorisation and NLP. These contributions have highlighted the effectiveness of the Prediction by Partial Matching method in classifying Arabic twitter text. The following list summarises these contributions.

- We have developed the BTAC corpus, which contains over 122K annotated tweets for five Arabic dialects – Gulf, Egyptian, Levantine, Maghrebi, and Iraq – in addition to Modern Standard Arabic and Classical Arabic. The corpus is also labelled according to the gender of the user who wrote the tweets and the genre of the tweets e.g. Social, Economy, Greetings, Cultural, Religious, Sport, and Political. The corpus is freely available for download (see section 3.4.3.3). This corpus represents a valuable and rich resource for NLP applications targeting Arabic dialects and code-switching research.

- Effective Authorship attribution for Arabic Twitter text has been achieved using the PPM compression method. Experimental results showed that this method was very effective reporting a high accuracy of 96% and 0.96 F-measure.
- Effective gender categorisation using the PPM compression method has been achieved when applied to Arabic Twitter text. Experimental results showed that this method was very effective compared with well-known machine learning algorithms producing an accuracy of 88% and F-measure of 0.87.
- Significant results for dialect identification of Arabic Twitter text has also been achieved using the PPM compression method. Experimental results showed that these methods were very effective reporting an accuracy of 87% and 0.87 F-measure.
- A novel method for significantly improving results for both gender categorisation and dialect identification has been devised that uses models trained on data organised by secondary class type (authorship) rather than the primary class type (gender or dialect).
- Effective detection of code-switching at the character level has been achieved. An accuracy of 81.2% was obtained using 5-fold cross-validation on the full BTAC corpus.

1.5. Publications

This research has produced three publications in the fields of Computer Science and Computational Linguistics.

- Gender and Authorship Categorisation of Arabic Text from Twitter Using PPM. This paper was published in AIRCC's International Journal of Computer Science and Information Technology (IJCSIT).

  Altamimi, M., and Teahan, W.J., "Gender and Authorship Categorisation of Arabic Text from Twitter using PPM." International Journal of Computational Science and Information Technology (IJCSIT) 9(2) (2017): 131-140.

- BTAC: A Twitter Corpus for Arabic Dialect Identification.

  The 6th Conference on CMC and Social Media Corpora was held in Antwerp, Belgium on 17-18 September 2018. It was hosted by the CLiPS research center (Computational Linguistics and Psycholinguistics) at the University of Antwerp:

  Altamimi, M., Alruwaili, O. and Teahan, W.J., "BTAC: A Twitter Corpus for Arabic Dialect Identification." In the 6th Conference on Computer-Mediated Communication (CMC) and Social Media Corpora (CMC-corpora 2018) (p. 5) 17-18 September 2018, University of Antwerp, Belgium.

- Arabic Dialect Identification of Twitter text Using PPM Compression.

  This paper has been accepted for publication in the International Journal of Computational Linguistics (IJCL):

  Altamimi, M., and Teahan, W.J., "Arabic Dialect Identification of Twitter text Using PPM Compression." International Journal of Computational Linguistics (IJCL) 10(4) (2019): 47-59.

1.6. Thesis Outline

The rest of the thesis begins with chapter two examining the background and related work including an introduction to the field of text categorisation along with an introduction to the main method used in this thesis, the compression-based approach Prediction by partial matching (PPM). Next, chapter three discusses the design and creation of the Bangor Twitter Arabic Corpus (BTAC), which includes overviews of the previous Arabic dialectal corpora. Then, chapter four explores authorship attribution tasks using PPM. Following this, chapter five explores the gender text categorisation problem using PPM. Chapter six covers the identification of dialect using PPM. Chapter seven presents code-switching detection of Arabic twitter text using character-based, and chapter eight concludes the thesis and sets out directions for future work.

Chapters	Title
Ch. 2	Background and Related Work.
Ch. 3	Creating the Bangor Twitter Arabic Corpus (BTAC).
Ch. 4	Authorship attribution of Arabic Tweets using PPM.
Ch. 5	Gender categorisation of Arabic Tweets using PPM.
Ch. 6	Dialect identification of Arabic Tweets using PPM.
Ch. 7	Character-based Identification of Code-Switching in Arabic tweets.
Ch. 8	Conclusion and Future Work.

Chapter 2

Background and Related Work

The purpose of this chapter is to provide the background and related work to the text categorisation applications used in this thesis. This chapter also introduces and explains the main method adopted for this research Prediction by Partial Matching (PPM), which is described along with other standard machine learning algorithms that have been used for comparison purposes. Furthermore, this chapter also reviews the basic fundamentals of Arabic language structure, with an emphasis on several challenges for text categorisation.

Thus, the chapter is summarised as follows. Section 2.2 provides a background of the Arabic language with an overview of the Arabic language structure; section 2.3 lists the related work for Arabic textual resources and its dialects; section 2.4 describes several text categorisation applications; section 2.5 explains approaches for text categorisation with an emphasis on the machine learning algorithms used in this thesis; section 2.6 provides an overview of the compression algorithms; section 2.7 explains compression techniques for text classification; section 2.8 describes the main method used, Prediction by Partial Matching; section 2.9 explains how PPM compression-based is used for language modelling; section 2.10 explains how PPM is used for text categorisation with an example on the use of PPM for the authorship attribution problem; section 2.11 demonstrates the classification procedure used in this study; section 2.12 reports metrics that are used for evaluation purposes; and finally, section 2.13 concludes of the chapter.

2.1. Introduction

With the increase of text available in social media, an efficient process is required for filtering those text data and converting them into useful information. Text mining is the process of acquiring knowledge from linguistic sources (Knight, 1999). The benefit of text mining is achieving semantic analysis without the need to understand the text, this ensures that relevant information is returned within a short period of time. Text categorisation is the process of assigning text to predefined categories. Recently, text categorisation has become popular due to the rapid growth of data available on the web over the past two decades. As a result, there are large amounts of data available to be categorised. Text categorisation becomes fundamental where it could be used in many applications such as dialect identification, authorship attribution, and gender categorization.

2.2. Arabic language background

Arabic is spoken by over 300 million people. It is the fifth most widely spoken language in the world after Chinese, Spanish, English, and Hindi. It is widely used in all of the Middle Eastern countries as a first or second language. There are three forms of Arabic: Classical Arabic (CA); Modern Standard Arabic (MSA); and Dialectal Arabic (DA). CA represents an older style of Arabic. It was used more among people in the sixth and seventh centuries (pre-Islam) and continued beyond that (Holes, 2004). CA is used in religious books such as the Qu'ran, Hadith (the speech of the prophet Muhammad, peace be upon him), and also some traditional books such as poetry and history. MSA is more formally used by all the Arabic-speaking people. It started to become popular with the increase of Arabic media in the second half of the nineteenth century and its popularity continues until now. It is used more in modern life media sources such as newspapers, magazines, and formal TV programmes.

However, MSA is used much less frequently in daily conversations where people use dialects to communicate. Dialects are widespread and divided by geographical region and are used in daily life communication between people from the same location see

Figure 2.1. However, it can often be difficult for Arabic-speaking people to understand each other when they use dialects specifically from north African regions (Harrat *et al.*, 2015).

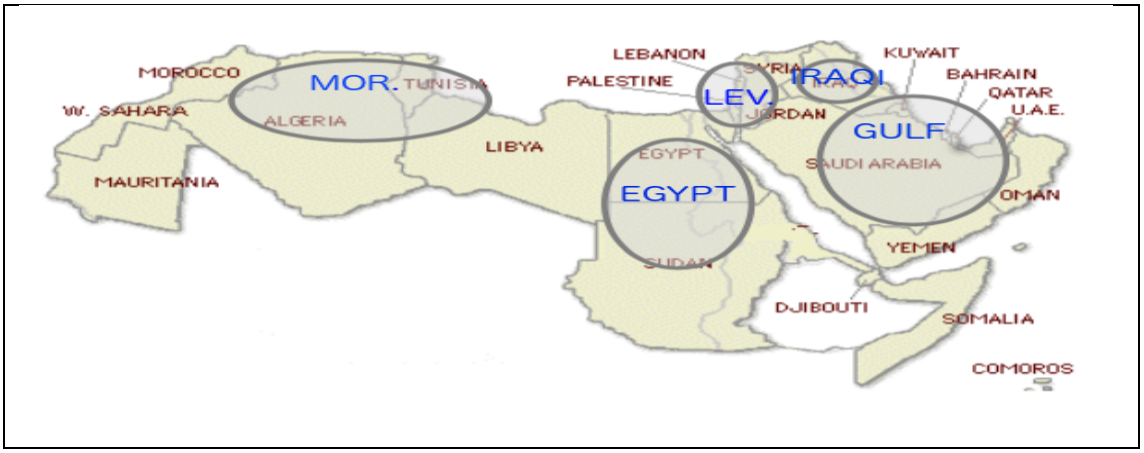

Figure 2.1 Map of the Middle East showing the spread of the five main dialects (based on Alshutayri & Atwell 2017). MOR. indicates Moroccan dialect; LEV. is for the Levantine dialect; EGYPT is for the Egyptian dialect; GULF indicates Gulf dialect; and IRAQI is for the Iraqi dialect.

2.2.1. Overview of Arabic language structure

The Arabic language is a language with a complex morphology. It notably differs from most of the other popular languages. In general, Arabic differs from the English language in three cases. First, Arabic has two genders – masculine and feminine. Second, Arabic has three forms of number – singular, dual, and plural. Third, Arabic has three grammatical cases – nominative, accusative, and genitive. However, words in Arabic can be categorised into three main parts-of-speech: nouns; verb; or particle. First, nouns describe a person, thing, or idea and it can be originated from other nouns or verb. Also, nouns has three nominative cases: when it is the subject; it could be the object of a verb; or it could be the object of a preposition (Zrigui *et al.*, 2012). Second, similar to English, verbs are divided into perfect, imperfect, and imperative forms. Finally, particle includes adjective, adverbs, pronouns, conjunctions, prepositions, interrogatives, and interjections (Khoja, 2001). This shows how Arabic has a complex linguistic structure which means that one root of the word can generate multiple hundreds of words having different or the same meanings. This increase in

morphological variations introduces several challenges for text categorisation tasks such as: inflection; diacritics; and word length and synonyms.

Form	Translation	Form	Translation
feminine	مؤنث	object	مفعول
masculine	مذكر	object of a preposition	مجرور بحرف جر
singular	مثنى	perfect	صيغة الفعل التام
plural	جمع	imperfect	صيغة الفعل الناقص
nominative	الرفع	imperative	صيغة الامر
accusative	النصب	pronouns	الضمائر
genitive	الجر	adjectives	الصفات
particles	ادوات	adverbs	الاحوال
nouns	اسماء	conjunctions	العطف
verbs	افعال	prepositions	حروف الجر
adjectives	صفات	interjections	صيغة التعجب
adverbs	ظروف	interrogatives	علامات الاستفهام
subject	فاعل		

Table 2.1 Overview of the Arabic language grammar structure based on (Ayedh *et al.*, 2016)

2.2.1.1. Inflection

In English, prefixes and suffixes are added to the beginning and the end of the root. Arabic is a highly diverse language. Infliction can be seen in various forms such as prefixes, infixes, and suffixes. This results in adding affixes to stems; for instance for the root علم the variations include the following: prefix: معلوم; infix: عالم; suffixes: علمي (De Roeck and Al-Fares, 2000). This increase in language variation expands the number of word variations in the Arabic language. An expected problem with inflections is that extracting lexical features due to variations of a word will be difficult (Abbasi and Chen, 2005b; Ayedh *et al.*, 2016).

2.2.1.2. Diacritics

Diacritics are special marks placed below or above letters to represent short vowels. Diacritics usage changes both the meaning and pronunciation of words. Arabic diacritics include Fathah (ـَ–), Dama (ـُ–), Kasra (ـِ–), Double Dama (ـٌ–), Double Fathah (ـً–), Double Kasra (ـٍ–), Hamza (ء), Shada (ـّ–), and Sukon (ـْ–), where – signifies a single letter (Duwairi, 2006). For instance, the word (علم) could mean ‘science’ (عِلْمُ), or

it could mean the verb ‘teach’ (عَلَّمَ), or it could mean the noun ‘flag’ (علمْ) depending on the diacritics (Nwesri *et al.*, 2006). Diacritics are rarely used in writing because the readers are expected to infer the missing short vowels using their semantic knowledge of the language. However, it is therefore difficult for feature extraction programs to use this knowledge directly (Abbasi and Chen, 2005b).

2.2.1.3. Word length and synonyms

Arabic words are considered short; therefore, this might reduce the effectiveness of lexical features, such as word length distribution (Abbasi and Chen, 2005b). Also, most Arabic letters are written concatenated with each other, few letters written separated from each other. In addition, nouns in Arabic do not start with capital letters such as names, cities, acronyms, and abbreviations as is the case in English, making the nouns more difficult to identify. Furthermore, Arabic is diverse with synonym variations; examples of synonyms in Arabic are انظر, شوف, شاهد, اطلع, ناظر, بص which all mean ‘look’. These variations are all derived from the classical Arabic, modern standard Arabic, and dialectal Arabic.

2.3. Resources for Arabic language

This section lists the related work for textual Arabic resources. First, the work related to Modern Standard Arabic resources is listed, followed by discussion of the work achieved for Classical Arabic, before moving on to mention Arabic dialect corpora resources. In the final subsection, summary of existing dialectal Arabic corpora are listed in Table 2.4 with different criteria such as the type of data, source, size, number of dialects, and annotation method.

2.3.1. Resources for Modern Standard Arabic

Most of the contributions on Arabic have focused on producing MSA corpora due to the availability of online media sources at the beginning of the twenty first century. Maamouri et al. (2004) created the Penn Arabic Treebank (PATB) which is derived mainly from newswires from diverse regional sources in the Arab region. The corpus

contains over 542K tokens and is considered the most widely used annotated MSA resource due to the informative linguistics content that the corpus contains such as parts-of-speech and morphological annotation.

The corpus developed by Abdelali et al. (2005) focused on a MSA text obtained through crawling online published newspapers from various Arabic countries. They collected 107 daily issues from 11 Arabic publications. The corpus of contemporary Arabic (CCA) created by Al-Sulaiti and Atwell (2006) collected a range of online written text with various topics such as fiction, arts, science, and business while the spoken part was taken from TV and radio podcasts. The corpus contains over 1M words purposely designed as a learner corpus (teaching Arabic as a foreign-language) and for other language processing research which is helpful for information retrieval studies and Arabic machine translation.

Smrž and Hajic (2006) developed the Prague Arabic Dependency Treebank (PADT). It is primarily intended for building models for statistical parsing. It consists of 100K tokens annotated morphologically and analytically from newswire including plain text from the Arabic Gigaword corpus (Graff, 2003). It varies considerably from that of the Penn Arabic Treebank (PATB) as it uses morphological and syntactic information. Moreover, Alansary et al. (2007) created the International Corpus of Arabic (ICA) containing 80M words. The collection of samples is written in MSA selected from a wide range of newspapers sources. Habash and Roth (2009) created the Columbia Arabic Treebank (CATiB). The corpus is manually annotated with parts-of-speech and syntactic analyses of 228K words.

Moreover, the resource Essex Arabic Summaries Corpus (EASC) introduced by El-Haj et al. (2010) is intended for Arabic text summarisation. The corpus contains diverse articles in different areas such as education, finance, health, politics, environment, art, music, science, religion, sport and tourism. The corpus contains 153 Arabic articles manually summarised by humans.

Saad and Ashour (2010) created an open source Arabic corpora (OSAC) which contains approximately 18 million words collected from newswire websites such as CNN, the BBC and other publications, focusing on various topics such as economics, education, religion, sport, health, astronomy, stories, law and cooking recipes.

There are also a few corpora that contain a mixture of both CA and MSA text. The Bangor Arabic Compression corpus (BACC) contains 31 million words designed for performing compression experiments on Arabic text. The text files are written with various genres, such as sports, culture, economics and so forth which are collected from many sources such as magazines, books, and websites (Teahan and Alhawiti, 2013). Also, the King Abdulaziz City for Science and Technology (KACST) corpus contains 700 million words from the pre-Islamic era to the present day. The corpus contains a variety of text mainly from newspapers, magazines, books and old manuscripts (Al-Thubaity, 2015).

Reference	**Name**	**Size**	**Source**
Maamouri et al. (2004)	PATB	542K tokens	Newswires
Abdelali et al. (2005)	Not mentioned	107 issues	Arabic newspaper publications
Al-Sulaiti and Atwell (2006)	CCA	1M words	TV and radio podcasts
Smrž and Hajic (2006)	PADT	100K tokens	Newswire
Alansary et al. (2007)	ICA	80M words	Newspaper publications
Habash and Roth (2009)	CATiB	228K words	Newswire
El-Haj et al. (2010)	EASC	153 articles	Newspaper
Saad and Ashour (2010)	OSAC	18M words	Newswire websites
Alhawiti and Teahan (2013)	BACC	31M words	Websites, magazines, and books
Al-Thubaity (2015)	KACST	700M words	Newspapers, magazines, and books

Table 2.2 Summary of existing Modern Standard Arabic corpora.

2.3.2. Resources for Classical Arabic

Classical Arabic (CA) corpora have derived less attention from the research community. The reason for this is that classical corpora do not represent the majority of written text in Arabic; CA represents specific periods of time even though it is still used nowadays as quoted text from the Holy Qu'ran and prophet supplications. CA resources are required for training purposes to combine with other resources such as MSA and DA rather than by themselves in order to build more effective NLP systems. Most of the content available corpora refer back to the pre-Islamic era and subsequent eras.

Both modern standard Arabic corpora mentioned previously, BACC and KACST, cover partly classical content. Moreover, The King Saud University Corpus of Classical Arabic (KSUCCA) is considered a resource that was created mainly for classical text. The goal of creating this corpus was to support research in both linguistics and computational linguistics such as studying the lexical semantics of the Holy Qu'ran. The corpus contains over 50 million words which are organised according to categories such as religion, science, sociology, linguistics, literature, and biography (Alrabiah *et al.*, 2013). The Shamela corpus is another large-scale historical Arabic corpus. It consists of one billion words across a fourteenth-century time span. The corpus includes Hadith collections, biographies, jurisprudence (Fiqh), and popular religious writings (Belinkov *et al.*, 2016).

Reference	Name	Size	Source
Alrabiah et al. (2013)	KSUCCA	50M words	Holy Qu'ran, Hadith, and old books
Belinkov et al. (2016)	The Shamela	1B words	Hadith, biographies, and jurisprudence

Table 2.3 Summary of existing Classical Arabic corpora.

2.3.3. Resources for Dialectal Arabic

Corpora that are specifically created for dialectal Arabic are more challenging to design than MSA and CA corpora as they require more time to build and need

linguistics experts to validate the dialectal text. Gadalla et al. (1997) created the first Dialectal Arabic (DA) corpus called CALLHOME. The corpus focused on the Egyptian dialect mainly collected from phone conversations. The corpus is designed for research related to speech recognition rather than research related to text analysis. Omar and Callison (2011) produced a corpus mainly from newspaper commentary sections and Twitter data to investigate dialect and genre categorisation. The corpus is considered the first actual attempt to create a dialect text corpus called the Arabic online commentary (AOC). The corpus consisting of a total of 108K sentences was labelled for MSA and three dialects: Levantine; Gulf; and Egyptian.

Salama et al. (2014) created a corpus for dialectal Arabic collected mainly from YouTube commentaries. The corpus contains multiple dialects such as Egyptian, Gulf, Iraqi, Maghrebi and Levantine. Over 600K sentences were collected and annotated according to the geolocation of the comments' authors. Harrat et al. (2014) collected a corpus comprising 4K sentences mainly from Algerian dialects. The corpus is used for machine translation research between modern standard Arabic and Algerian dialect.

Almeman and Lee (2013) automatically built a dialectal corpus by bootstrapping Arabic web pages with a total of 1043 dialect words. The corpus that was created comprised 48 million tokens mainly distributed across four dialects – Gulf, Levantine, Egyptian and Maghrebi. The text was obtained from various resources such as forums, blogs and comments. The Gumar corpus (Khalifa *et al.*, 2016) was built from over 100 million words of the Gulf dialects collected from 1,200 novel forums. The corpus is classified for the Gulf sub-dialects at the document level.

The corpus created by Saad (2017) comprises text in both Egyptian and MSA dialects collected from Wikipedia articles. Harrat et al. (2017) created an Arabic parallel corpus that is built for Maghrebi, Tunisia, Algerian, Palestine and Syrian dialects. It consists of 6400 sentences in each of the five dialects in addition to MSA. The data were taken from recorded movies and shows and then transcribed by hand.

The Curras corpus was created by Jarrar et al. (2017) and is considered the first morphological corpus of the Palestinian dialect. The corpus contains 56K tokens collected form social media, blogs, forums, and stories. The project COLABA produced by Diab et al. (2010) harvested weblogs with the focus on both Egyptian and Levantine dialects. Unfortunately, this corpus is not publicly available.

Recently, Twitter has provided a rich resource for collecting dialectal text. Many researchers have taken advantage of this and used its API to collect texts in the form of tweets. Mubarak and Darwaish (2014) collected over 175 million Arabic tweets. Those tweets are filtered according to the user location to determine the dialects resulting in 6.5 million dialectal tweets. The corpus was created mainly to perform Arabic dialect categorisation. The research by Alshutayri et al. (2016) also explores Twitter as a source of Arabic dialects. Over 200K tweets were collected according to 35 unique words from the main five dialects. Alshutayri and Atwell (2018) added 10K comments from newspapers, and 812K comments from Facebook along with previous corpora from Twitter. The corpus is annotated for five groups of Arabic dialects – Gulf, Iraqi, Egyptian, Levantine, and North African. Twenty-four-thousand tweets and comments were randomly annotated online through a game by 1,575 participants designed specifically for this task.

The Arap-Tweet corpus (Zaghouani and Charfi, 2018) was collected mainly for dialect Arabic categorisation and it was annotated according to the geographical location. The Shami corpus created by Abu Kwaik et al. (2018) is designed specifically for Levantine dialects. Over 117K tweets were collected from four Levantine dialects. The corpus is created to aid dialects identification research. Furthermore, the dataset DART has about 25K tweets that are annotated via crowdsourcing over five main Arabic dialects: Egyptian; Gulf; Levantine; Maghrebi; and Iraqi (Alsarsour *et al.*, 2018).

Finally, two dialectal resources are created for the MADAR project (which means ‘orbit’ in Arabic). The first dataset is the MADAR travel domain dialect identification dataset (Bouamor *et al.*, 2018). It is a parallel corpus consisting of sentences covering the dialects of 25 cities form the Middle East. The corpus consists of selected 2,000

sentences translated from Basic Traveling Expression Corpus (BTEC) (Takezawa *et al.*, 2007).

The second dataset is the MADAR Twitter user dialect identification dataset (Bouamor *et al.*, 2019). The corpus contains 2,980 Twitter user profiles from 21 different countries in the Middle East. A total of 100 tweets are collected from each user. Then the user's profile is annotated manually by three Arabic native speakers according to the location.

2.3.4. Discussion and limitations of dialectal Arabic corpora

This section reviews various aspects about dialectal Arabic corpora. The discussion investigates the corpora in three different contexts: data annotation; variety of dialects; and size of the annotated text.

2.3.4.1. Data annotation

The method used for the annotation process differs from one corpus to another. Some corpora crowdsource their annotation tasks to other service providers. The only drawback to these corpora is the quality of the annotation specifically when the number of annotators increases. For instance, the Arabic Online Commentary (AOC) corpus (Zaidan and Callison-Burch, 2011) outsourced the annotation task by using Amazon's Mechanical Turk service employing 454 annotators. According to Darwish, Sajjad, and Mubarak (2014), they did not use the AOC corpus in their study as the annotation quality of the corpus was not reliable.

Other research presumes that each tweet belongs to a dialect according to the geographical information of the tweet (latitude and longitude) (Mubarak and Darwish, 2014; Salama *et al.*, 2014; Abu Kwaik *et al.*, 2018; Zaghouani and Charfi, 2018). However, that is not an accurate assumption for several reasons: tweets might be written by a user in a location speaking a different dialect; tweets might involve modern standard Arabic and classical Arabic; and tweets might also contain code-switching.

More recently, some Twitter accounts might be used by bots. These bots use specific locations to post tweets automatically often for malicious reasons. In January 2017,

the BBC published a large list of fake accounts found on Twitter (2017). One of the features of those accounts was that they involved tweets posted from places where nobody lives. Fake accounts and bots are becoming more prevalent in the Middle East. These accounts are used to post fake news, propaganda, and hate speech from certain locations to escalate the Middle East crises in the region and create further tensions between the nations. For this reason, it seems that trusting the location of the tweet for annotation purposes is not an always accurate assumption.

Lately, other corpora are annotated according to specific keywords from each dialect. This idea is more precise than relying on a tweet's location, but the only downside of this approach is code-switching might occur in some of the tweets. In addition, building a corpus using key words requires a generalised quantity of keywords that could be relied on. However, language used on social media differs from other standard language in that it is very common to encounter new words.

However, other researchers annotated their corpora manually using volunteers or experts. This method is more accurate than all the other methods above. The annotation task is done by known people. The downside of this method, though, is that it is challenging because it takes more time and effort to produce this kind of annotation.

2.3.4.2. Variety of dialects

The number of dialects that some corpora include varies according to the aim of the research. For example, some researchers focused only on creating a mono-dialect corpus where the focus was to create a corpus for one specific dialect: Egyptian (Gadalla *et al.*, 1997; Saad, 2017); and Palestinian (Jarrar *et al.*, 2017). Other corpora aimed to create a corpus for multiple dialects from the main dialects in the Middle East (Zaidan and Callison-Burch, 2011; Mubarak and Darwish, 2014; Salama *et al.*, 2014; Alsarsour *et al.*, 2018; Alshutayri and Atwell, 2018). Other corpora are created from multiple dialects from specific regions; for example, Abu Kwaik et al. (2018) uses four Levantine dialects, Khalifa et al. (2016) use six Gulf dialects, and Harrat et al. (2017) uses five Maghrebi and Levantine dialects.

However, adding more than the five main dialects is confusing and challenging. The annotation task becomes more complex as it requires experts from each dialect. Zaghouani and Charfi (2018) built a corpus that contained 11 dialects and found similarity between some dialects such as the Moroccan and the Algerian dialects and also between the Qatari dialect and some other Gulf dialects.

2.3.4.3. Size of the annotated file

It is believed that a consistent smaller corpus that adheres to a high-quality design is far more valuable than a larger corpus (Granger, 1993). Most of the corpora are now much larger due to better hardware and software that exist currently. This allows researchers to collect data more easily. Moreover, the existence of various APIs from social media platforms allow research to collect larger volumes of data; however, The AOC corpus annotated 110K sentences out of 3.1 million collected sentences. Mubarak and Darwish (2014) collected 175 million tweets, then extracted 6.5 million tweets using the aforementioned list. After that 100 tweets from each dialect were evaluated by three different judges. Those judges were from the countries associated with the dialect to confirm whether the tweet belongs to a dialect or not. Alshutayri and Atwell (2018) annotated 8K tweets according to a seed word out of 210K collected tweets. Moreover, they annotated 24K documents out of the 1.1 million documents collected from Twitter, Facebook and website comments, relying on 1,575 users who participated in a game designed for annotating documents.

The table below summarises existing dialectal Arabic corpora contributions.

Reference	Type	Source	Size	#Dialects	Annotation method
Altamimi et al. (2018)	Dialects Genre Gender	Twitter	122K tweets	5 dialects, MSA and CA	Manually by dialects, genre, gender, and authorship
Gadalla et al. (1997)	Dialects	Phone conversation	120 telephone conversations	Egyptian dialects	Not mentioned
Khalifa et al. (2016)	Dialects	Text- novel	100M words	6 Gulf dialects	Document-based annotation
Saad (2017)	Dialects	Wikipedia	Not mentioned	Egyptian dialects	Not mentioned

Harrat et al. (2017)	Parallel dialects	Movies and shows	6400 sentences	5 dialects	Manual translation
Jarrar et al. (2017)	Dialects	Text-social media, blogs, forums, and stories	56K tokens	Palestinian dialects	Manual annotation
Omar and Callison (2011)	Dialects	News commentary and Social media	108K sentences	4 dialects, MSA	Amazon's Mechanical Turk service
Salama et al. (2014)	Dialects	YouTube commentary	630K sentences	5 dialects	Geographical location of the author
Mubarak and Darwaish (2014)	Dialects	Twitter	6.5M tweets	5 dialects	According to the location of the tweet
Alshutayri et al. (2016)	Dialects	Twitter	210K tweets	5 dialects	According to common dialectal word
Alshutayri and Atwell (2018)	Dialects	Text-Twitter Facebook Comments	226K tweets 812K Facebook comments 9K online comments	5 dialects	According to user location and common dialectal word
Abu Kwaik et al. (2018)	Dialects	Twitter	117K tweets	4 Levantine dialects	According to the location of the tweet
Zaghouani and Charfi (2018)	Dialects Gender Age	Twitter	2M tweets	11 dialects	According to the location of the tweet
Alsarsour *et al.* (2018)	Dialects	Twitter	25K tweets	5 dialects	Manually via crowdsourcing
(Bouamor *et al.*, 2019)	Dialects	Twitter	297 tweets	21 dialects	Manually according to the user profile

Table 2.4 Summary of existing dialectal Arabic corpora.

2.4. Applications of text categorisation

In this section, a number of text categorisation applications are listed where such techniques are applied. Each application is supported with popular research using different approaches such as feature-based and character-based compression.

2.4.1. Authorship attribution

There are two prominent surveys mainly directed at authorship attribution studies (Juola, 2006; Stamatatos, 2009). This section, however does not review the traditional studies of authorship attribution that address how to solve disputed authorship. Instead, the focus is on more recent contributions to authorship attribution, highlighting the classification methods and the types of features used in the research, and the reported results. Following this, most of the contributions made by researchers for the Arabic language are explored, with a focus on the limitations of the research for that specific language.

In the studies on authorship attribution, to identify the author of a text, researchers in the past have focused on attributing authors who used a single unique type of textual feature. This assumption was derived from the belief that each author has a specific writing style. For example, the study by Mendenhall (1887) used word length frequencies to attribute text to Marlowe, Bacon or Shakespeare; also the research by Brinegar (1963) used the same method to attribute the Quintus Curtius Snodgrass Letters which were 10 letters published in the New Orleans Daily Crescent in 1861. However, Williams (1975) investigated the work by Mendenhall and concluded that the distinctions in word-length distributions found in the study were more attributed to the style of text (poetry and prose), rather than between different authors.

An earlier study by Yule (1939) suggested using sentence length as a possible method for distinguishing authors. Koppel et al. (2009) stated that this method did not yield reliable results if used alone, but it did open the way to figure out new methods such as statistical techniques and machine learning algorithms which were explored later.

Statistical features have often been used to attribute authorship. Mosteller and Wallace (1963) used Bayesian statistical analysis to discover the actual author of Federalist Papers (a number of political newspaper essays written by John Jay, Alexander Hamilton, and James Madison; both Hamilton and Madison claimed that they wrote 12 of these essays). The study uses a word count based on function words such as *upon, of, and* to measure the distinction among candidate authors. This study

encouraged researchers to discover other types of features and techniques to apply to the authorship problem.

Researchers made many efforts to define features for capturing the authors' writing "fingerprints" (Holmes, 1994; 1998); for example, sentence length, word length, word frequencies, character frequencies, and vocabulary richness. However, most of the work undertaken in the beginning of 1990s lacked satisfactory evaluation measures. Stamatatos (2009) state that much of the research was mainly directed to solving disputed authorship for large textual data, hence the number of candidate authors (two to three authors) was often too small. This caused most of the studies to report just the name of the potential author candidate, rather than report a measure of accuracy.

With the increase of available electronic text, authorship attribution research was reanimated using statistical techniques and machine learning algorithms. Khmelev and Tweedie (2001) performed a study based on Markov chains that were used for attributing 45 authors from the Gutenberg project, achieving an accuracy of 74.42%. Also, many researchers explored compression-based approaches. Benedetto et al. (2002) used off-the-shelf compression programs such as GZIP to perform classification. Their study mainly focused on an authorship attribution task for Italian text, achieving an accuracy of 93.3%. However, their method was criticised by Goodman (2002) for taking too long to execute.

Additionally, Khmelev and Teahan (2003) used the prediction by partial matching (PPM) text compression scheme to attribute authors from the RCV1 dataset (Lewis *et al.*, 2004). They achieved an accuracy of 89%. This was similar to Hunnisett and Teahan (2004) who achieved similar results using higher-order PPM models. With the same dataset, Marton et al. (2005) explored other compression methods such as RAR, LZW, and GZIP, achieving accuracy of 78%, 66%, and 79%, respectively.

Furthermore, various machine learning methods have been used for authorship attribution such as the research by Hoorn et al. (1999) which studied three classifiers – Naïve Bayes (NB), K-nearest neighbour (KNN), and neural networks – to attribute

authors of poetry. Using trigrams, they found that the neural network outperformed other algorithms, achieving 80-90% accuracy when identifying two authors of poetry.

Khmelev and Teahan (2003) also performed classification using SVM, reporting an accuracy of 85%. Moreover, Argamon and Levitan (2005) measured the use of function words for authorship attribution using SVM, and achieved 99% accuracy. Their research involved using the most frequent words to distinguish between eight authors. Later, Zhao (2007) compared various classifiers such as Naïve Bayes, KNN, 3-NN, Decision Tree, and Bayesnet. They found that the Bayesnet classifier is the most effective method using function words to distinguish between authors.

In contrast, instead of using function words, Tan and Tsai (2010) used both lexical and syntactic features for authorship attribution. They reported an accuracy of 90% using the Naïve Bayes classifier. Their research also found that for large datasets, *syntactic* features such as common and function words are effective, whereas *lexical* features such as number of sentences and punctuations are efficient for small datasets.

Pillay and Solorio (2010) combined supervised and unsupervised learning approaches to their authorship attribution task on web forum posts. They first performed clustering to help identify relevant features which can assist the classification process. Next, they performed classification using four algorithms: Bayesian Networks; Naïve Bayes; and Decision Trees (C4.5). They reported that using Bayesian Networks performed well with an accuracy of 90.80% when used to distinguish between five authors.

Concerning non-English studies, Coyotl-Morales et al. (2006) were able to use a set of word sequences to identify the authorship between five Mexican poets using the Naïve Bayes classifier. They studied the effect of identifying text using frequent words sequences that combine stylistic and topic features of the authors. The research achieved an accuracy of 77.30%.

Türkoğlu et al. (2007) applied various classifiers such as Naïve Bayes, SVM, KNN, Random Forest and Multi-Layer Perceptron (MLP) classification methods to identify authorship on Turkish text. Ten different feature vectors were obtained from authorship

attributes including N-grams and various combinations of these features. SVM outperformed other algorithms with an average accuracy of 84.8%.

Pavelec et al. (2007) applied SVM to the Portuguese language using four kernel functions: Linear; Polynomial; Gaussian; and Tangent Hyperbolic. Their best result was achieved using a linear kernel which resulted in a 75.1% accuracy. Reicher et al. (2010) performed authorship classification for the Croatian language for multiple sources such as articles, blogs and books. The classification implemented using SVM achieved accuracies of 93%, 91%, and 98% for each source. The study reported that the accuracy performed best when using combinations of function words, punctuation marks, word length, and sentence length frequencies.

2.4.1.1. Previous work on Arabic authorship attribution

For the Arabic language, Abbasi and Chen (2005a) used SVM and C4.5 with lexical, syntactic, structural and content-specific features. They used the clustering algorithm of De Roeck and Al-Fares (2000) to extract roots of Arabic words. These roots were used to help identify authors with SVM using features based on root morphological similarities in Arabic scoring 85.43% accuracy. Later, Abbasi and Chen (2005b) used SVM and C4.5 decision trees on political and social Arabic web forum messages from Yahoo groups. They performed authorship analysis achieving an accuracy of 94.83% with SVM using a combination of lexical, syntactic, structural, and content-specific features.

In a further attempt, Abbasi and Chen (2006) examined both SVM with a Writeprints technique which uses a dynamic feature-based sliding window algorithm. They created an authorship visualisation tool which generates specific writing patterns. These patterns can be automatically identified according to the authors' writing styles. As a result, the Writeprints technique performed well against SVM when the messages were grouped by single authors. However, SVM outperformed Writeprints when identifying the author of a single message.

Shaker and Corne (2010) used function words for Arabic authorship attribution of 12 Arabic Books. They applied linear discriminant analysis (LDA) using 104 common Arabic conjunctions and prepositions reflecting English function words produced by Mosteller and Wallace (1963). The best performance achieved was 87.63% accuracy using 54 function words.

Furthermore, Alwajeeh et al. (2014) explored authorship identification of five authors' news articles using both the SVM and NB classifiers, with both classifiers performing well in terms of the accuracy – 99.8% and 99.4%. Altheneyan and Menai (2014) undertook an extensive study in authorship identification using NB classifier models: Multinomial Naïve Bayes; Multivariate Bernoulli Naïve Bayes (MBNB); and Multivariate Poisson Naïve Bayes. MBNB performed well among all the NB models with an average accuracy of 97%.

Albadarneh et al. (2015) conducted author identification in Twitter data. They focused on many challenges such as dealing with the large scale of Arabic tweets, the short text length of tweets, and the lack of available Arabic NLP tools. The results produced an accuracy of 61.6% using the NB classifier to perform the experiment using the Hadoop platform for Big Data analysis.

Later, Rabab'ah et al. (2016) explored authorship identification on Twitter posts. The study achieved the highest accuracy of 68.67% using SVM, which was obtained by a combination of bag-of-words stylometry features. Altakrori et al. (2018, p. 35) also studied Arabic authorship attribution extensively on Twitter posts. The experiment involved studying the effect of increasing the number of tweets to the testing set and increasing the number of authors per experiment. They did not achieve a satisfactory result due to the large scale of candidate authors (20 authors).

2.4.1.2. Limitations of previous authorship attribution research

From the previous studies in the literature, it seems that machine learning algorithm methods have been studied thoroughly to attribute authors. It is considered a new trend in research due to the increase in the range of software that implements the

algorithms and the fast-processing hardware that runs these algorithms. Moreover, Arabic authorship attribution is also well studied for most of the machine learning algorithms such as Naïve Bayes, SVM, and KNN (Shaker and Corne, 2010; Altheneyan and Menai, 2014; Alwajeeh *et al.*, 2014; Albadarneh *et al.*, 2015; Rabab'ah *et al.*, 2016). However, a different statistical approach such as using a compression character-based approach is a less studied area for Arabic. According to Teahan and Harper (2003) compression-based classification using character features can outperform the feature-based approach. Character-based features have proved to be capable of detecting "fingerprints" of an author, when used in various forms such as character unigram, bigram and trigram features. Characters or sequences of characters can effectively capture the author's style of writing; in particular when the author inserts specific characteristic identifiers when writing such as punctuation, underscores and commas. In contrast, word-based approaches are language-dependent and could rely on the topic of the text more than on the author's style, which might cause confusion during the classification process.

In terms of the method used, Prediction by Partial Matching (PPM) is a compression-based method which outperforms other compression methods. In English, using Reuters-10 author articles, Teahan et al. (2001) found that order 3 PPMD performed well with an accuracy of 91%. On the same dataset, Marton et al. (2005) showed that the RAR, LZW, and GZIP compression methods did not perform as well, achieving 87%, 84%, and 83%, respectively. Moreover, on the RCV10-dataset, Khmelev and Teahan (2003) reported an accuracy of 89.2% using order 5 PPMD. On the same dataset, Marton et al. (2005) reported worse results for RAR, LZW, and GZIP achieving an accuracy of 78%, 66% and 79%, respectively.

To date no specific study in Arabic has investigated the effect of compression methods on the authorship attribution problem. However, the study by Ta'amneh et al. (2014) examined genre categorisation using compression methods such as RAR, LZW, and GZIP on the BBC Arabic dataset (Saad and Ashour, 2010) achieving accuracy of 84%, 76%, and 62%, respectively. Prior to the research conducted by Ta'amneh study, no

work has been done to examine the effects of using compression-based methods such as specifically for the Arabic authorship attribution problem.

This current research is conducted using the new Twitter corpus – BTAC – for attributing authors. Previously, authorship attribution studies relied on a small number of candidate authors to evaluate the results using large amounts of training text. However, this research investigates the problem as a real-life authorship attribution scenario. In addition, further investigation is conducted on the effect of various text sizes such as: short texts in the form of single tweets; and long texts in the form of multiple tweets. Also, we test the performance of having a small set of 5, 10, 15 and 20 candidate authors, and a more complicated scenario with a large number (101) of candidate authors.

In previous studies of authorship attribution in Arabic from the Twitter dataset Rabab'ah et al. (2016) explored authorship identification on 12 users. Another study by Albadarneh et al. (2015) performed an authorship study for 20 Twitter users, whereas the study by Alwajeeh et al. (2014) attributed articles to just five authors. Table 2.5 below lists the recent Arabic authorship studies with classifier used, accuracy reported, type of text, and the total number of authors.

Citation	Classifier	Accuracy	Text Source	#Authors
Alwajeeh et al. (2014)	SVM	99.8%	News	5
Altheneyan and Menai (2014)	MBNB	97.0%	Books	10
Abbasi and Chen (2005b)	Writeprints	94.8%	Al-Hayat Newspaper	100
Shaker and Corne (2010)	LDA	87.6%	Books	12
Rabab'ah et al. (2016)	SVM	68.6%	Twitter	12
Albadarneh et al. (2015)	NB	61.6%	Twitter	20

Table 2.5 Recent results for Arabic authorship attribution.

2.4.2. Gender categorisation

There are three main aspects that determine the differences for gender categorisation research: first, the method used for categorisation; second, the features used to search for similar styles between both genders; and third, the accuracy produced by the categorisation method. This section addresses those aspects for most of the gender categorisation publications with an emphasis on research that used data from social media.

Gender categorisation is the process of identifying the gender of the author of the text and has been applied to many different fields, from speech understanding (Trudgill, 1972; Ritchie Key, 1975; Labov, 1990; Eckert, 1997) to informal writing such as student essays (Mulac *et al.*, 1990; Mulac and Lundell, 1994) and electronic messaging (Herring, 1996). Also, gender categorisation is seen in image recognition in the field of forensic analyses (Jain *et al.*, 2005). However, in this review, the main interest is in research related to gender categorisation in written text.

Gender categorisation research started with the investigation of language differences between genders. Lakoff (2004) found that a list of lexical, syntactic and pragmatic features differentiates the language style used by women as well as the use of specific wordlists, expletives, and questions asked. Mulac et al. (1988) found there are certain questions that tend to be asked by women; for instance, “Does anyone want to get some food?”, whereas “Let’s go get some food” is more likely to be found in men’s conversations. On the other hand, a number of studies have argued against the existence of any clear differences in men’s and women’s language (Bradley, 1981; Weatheral, 2002). For example, Thomson and Murachver’s (2001) study of email messages found that men and women equally tend to ask questions, offer compliments, and apologies.

With the development of NLP tools, automated gender categorisation has been increasingly investigated in gender studies. Koppel et al. (2002) were able to predict authors' gender with approximately 80% accuracy, analysing a large corpus of formal written texts (fiction and non-fiction) from the British National Corpus (BNC). Their

study used 42,000 words as training data. They found that the best performance was achieved when combining both features involving function word distributions and parts-of-speech N-grams using the Exponential Gradient (EG) algorithm of Kivinen and Warmuth (1996). Furthermore, Argamon et al. (2003) used writing style features to help to categorise the gender. They used the BNC and applied EG algorithms to categorise gender. They showed that males used more nouns then females, whereas females used more pronouns. Doyle and Keselj (2005) used distance measurement techniques for automatic categorisation of author gender based on N-grams profiles achieving 81% accuracy.

Another study by Cheng et al. (2009) investigated the gender categorisation problem by using the Enron email dataset (Enron, 2005). Results showed that the Support Vector Machine classifier (SVM) performs better than the decision tree method, achieving an accuracy of 82.20%. They observed that when combining a list of psycho-linguistic features introduced in their previous work, word-based features (such as function words) contributed more in categorisation of the gender. More recently, Cheng et al. (2011) addressed the problem of gender categorisation using psycho-linguistic features by analysing the generic writing styles of men and women. Experimental results showed that SVM outperforms Bayesian-based logistic regression and AdaBoost decision tree for identifying the author's gender. The experiment was performed using the Reuters and Enron corpora achieving accuracy of 76.75% and 82.23%, respectively.

Lately, the attention of research has shifted towards text generated from social media. This attention was necessary to address the problems of gender information being hidden in social media.

2.4.2.1. Gender categorisation studies on social media

The study of gender categorisation in social media is applied mostly using machine learning algorithms. Rao et al. (2010) investigated various classification tasks such as age, gender, regional origin, and genre for Twitter text. Their study produced a collection of large datasets representing all the classification tasks. They used SVM

with different feature models such as sociolinguistic feature models, N-grams feature models, and stacked models which combined both N-grams features and sociolinguistic features along with their prediction weights. The results showed that stacked models that combine both models using the SVM classifier achieved the best accuracy of 72.33%; better than the sociolinguistic feature model and N-grams alone with accuracy of 71.76% and 68.70%, respectively.

Burger et al. (2011) used statistical models to detect the gender of data collected from Twitter. Machine learning algorithms were applied such as SVM and NB, along with the Winnow2 algorithms (Littlestone, 1988). The results show that balanced Winnow2 achieves the best accuracy of 74.0%, over SVM and NB, 71.8%, 67.0%, respectively using word unigram features.

A more recent study by Liu and Ruths (2013) investigated the relationship between first names and text in detecting gender tweets, assuming this could improve the accuracy of gender detection. The study involved collecting a large body of data of tweets that were classified using SVM, achieving an accuracy of 87.1%. Marquardt et al. (2014) investigated how to increase the predictive performance of detecting users according to age and gender attributes. They used text from Twitter, blogs, and hotel reviews in the English and Spanish languages. SVM was used to detect gender attributes with context-based and stylistic features. They used two models to evaluate their accuracy – powerset transformation (LP) model, and classifier chains (CC) model – both using SVM as the underlying learning algorithm. The accuracy of detecting gender on the Twitter dataset was 71.15% using LP models, and 69.15% using CC models.

Modak and Mondal (2014) also studied gender classification using machine learning. Several classification algorithms were applied such as Naïve Bayes, maximum entropy and decision tree. They applied the username of the author's tweet as a feature to classify gender of the tweet. The results showed that maximum entropy performed with the highest accuracy in comparison to the other classifiers. Deitrick et al. (2012b) studied gender categorisation of English tweets. They used a neural

network with different N-grams features to classify the text and achieved 82% accuracy using the entire set of features.

Sap et al. (2014) studied age and gender categorisation for predicting gender words from a dataset of Facebook users. They used regression and classification based on word features, and achieved 91.9% accuracy in gender detection. Meanwhile, Ugheoke and Saskatchewan (2014) studied gender detection for authors of tweets using the SVM classifier and achieved an accuracy of 86.8%. To optimise the results, they manually annotated the gender by comparing the user name of the account with the US Social Security publicly available gender names. This method increased the accuracy of the results, achieving 95.3%.

Mikros (2012) showed that the accuracy of gender categorisation achieved 82.6% using the SVM classifier. The study investigated gender detection and author profiling using data collected from various Greek blogs. The author focused on two features of text content: *classical stylometric* features; which depend on the vocabulary variation such as word length and word frequencies; and *N-grams* features which depend on character bigrams and word unigrams.

The study by Volkova et al. (2013) undertook an analysis of the important differences between male and female in Twitter text for three languages – English, Russian and Spanish. They investigated whether gender differences in subjective language can effectively be used to improve sentiment analysis. Their result shows that incorporating gender leads to major improvements for sentiment analysis which can help to improve subjectivity and polarity classification in all three languages.

2.4.2.2. Gender classification studies for Arabic text

Alsmearat et al. (2014) performed an extensive study on the Arabic gender of formal online news text containing 250 articles for both genders. They find that SVM achieved a high accuracy comparing several machine learning algorithms using bag-of-words approach for feature selection. They also studied the effect of stemming on the

performances of the selected classifier reporting that it decreases the accuracy in some experiments when applying an Arabic light stemmer.

Furthermore, Alsmearat et al. (2015) investigated the impact of emotion analysis on identifying the author's gender. Their work included analysing bag-of-words computing features related to sentiment. The goal was to find out if there is a specific distinct style of writing between Arabic males and females. They collected their dataset manually from Arabic news websites and also collected Modern Standard Arabic text for both genders. They used Naïve Bayes, KNN and SVM to evaluate their experiments. The results showed that SVM achieved a high accuracy of 86.4% using the bag-of-words approach. However, when applying feature selection techniques, they could not confirm that there is a distinct style of writing between males and females based on the dataset used.

Estival et al. (2007) developed a tool which can detect author attributes or other information such as name, age, gender, and level of education. They used a collection of emails collected from both genders. Many machine learning classifiers were used in their experiments such as SVM, KNN and decision trees (J48). In gender detection, the results showed that SVM achieved 81.15% accuracy over other classifiers. This result was achieved using a combination of four features – character, morphological, lexical and named entity which includes language-independent name features such as URLs.

Moreover, AlSukhni and Alequr (2016) investigated the use of machine learning algorithms in detecting the gender of Arabic tweets. They collected a Twitter dataset for both genders which mainly comprised text written in the Jordanian dialect. They tested the ability of many machine learning classifiers, such as J48, KNN, Naïve Bayes, MNB and SVM. Overall, MNB performed well against other classifiers with an accuracy of 59.91% after pre-processing and 60.72% without pre-processing. However, the results showed significant improvement after adding the author's first name as a feature, when the accuracy of J48, MNB and SVM classifiers achieved above 98%.

Alrifai et al. (2017) presented an approach for author profiling using twitter data. They used the dataset provided on the PAN2017 (Rangel *et al.*, 2017) consisting of 24000 tweets collected from both gender, and the test set consist of 1600 authors. An accuracy of 66.38% was reported with character n-gram feature using SVM classifier as a training algorithm.

Most of the contributions to the field on gender categorisation used machine learning algorithms, particularly SVM. This algorithm excels in most of the research due to its suitability for binary classification. One problem with the machine learning approach is that it requires pre-processing steps such as feature extraction, stemming, and normalisation. This is performed to reduce the complexity of Arabic language morphology by decreasing the number of features. However, that is not always effective as it may change the word meaning or gender specifically in this study, knowing that Arabic words have two genders, masculine and feminine. For this reason, performing stemming and normalisation can in fact lead to negative performance as reported by Wahbeh et al. (2011).

For this reason, using PPM which is based on a character-based approach is preferable for gender categorization problems where it is confirmed that the most representative features need to be considered, whereas also there are no pre-processing or higher level of modification required for the examined text. The following summarises the advantages of using PPM over other algorithms that require the features-based approach:

- It does not require features selection to be performed or stop word extraction.
- It does not require heavy morphological analyses such as stemming and normalisation which may hinder the categorisation process.
- It performs at the character level, so it does not require specifying word boundaries or word segmentation.
- Syntactic features such as periods, colons and comma and the repeat of these features are considered helpful for recognition.

Reference	Classifier	Features	Acc.	Material Used
Alsmearat et al. (2015)	SVM	bag-of-words	86.04%	500 articles
Estival et al. (2007)	SVM	combination of four features	81.15%	8,028 emails
Alrifai et al. (2017)	SVM	character n-gram	66.38%	1600 authors
AlSukhni and Alequr (2016)	MNB	first name	60.72%	8000 tweets

Table 2.6 Recent contributions to gender categorisation in Arabic language.

2.4.3. Dialect identification

There are two definitions of dialect. It can refer to the accent used by a speaker (Chambers and Trudgill, 1998) or it can refer to the language vocabulary used by a specific region or community (Liu and Hansen, 2011). The first definition refers to the pronunciation variations of language, whereas the second definition refers to the structure of the language such as lexicon (vocabulary) or grammatical variations (morphology and syntax). In this section, the focus is on the second definition with an emphasis on the text variation of the language among the regional community of native speakers. Many papers and research studies address the challenge of dialect identification (Chambers and Trudgill, 1998); more recently a survey was undertaken based specifically on dialect identification (Etman and Beex, 2015).

The early studies of dialects were introduced by social and linguistics scientists. The aim of most of this early research was to find dialectal separation between two regions or to answer general questions about the existence of dialect. For instance, Bailey (1968) and Davis and Houck (1992) focused on finding word variations between the north and south of the U.S.A. surrounding the Ohio river and whether or not the Midland dialects exist. Labov (1972) demonstrated that the New York accent variations after the “r” letter occur after the vowel (postvocalic). Other studies focused on words

and lexical sets to classify English pronunciation according to some specific features that represent each accent (Wells, 1982); and more recently (Nagy *et al.*, 2006). Kessler (1995) focused on string distance methods in Irish Gaelic which combined studying lexical set methods with phonetic analysis. However, these kinds of studies proved to be expensive as they required time and experts to identify the differences and perform the identification. The deeper the approach to the study of features, the harder it was to solve the problem and the more time was needed to achieve results.

Computational linguistics was not applied to study dialect identification in the early research stages for a few reasons. First, most of the main languages are standardised and, even with the existence of dialects, differences do not affect people understanding each other. For instance, English, which has received much consideration generally from a computational perspective, has two main variations: American English and British English (Stein and Quirk, 1995; Trudgill and Hannah, 2013). In French, the language follows the standard approach. In India, most of the variations are related to language differences. In Arabic, dialectal variations exist between most of the Middle Eastern countries, but the language is standardised when used formally.

Second, other research areas were sparked in the field of computational linguistics, as differences were more highly obvious in those areas such as language identification, authorship attribution, gender identification, and topic categorisation. Third, the study of dialects was considered challenging often initially due to lack of resources, as dialects are not used formally. For instance, no dialects are used in media sources such as newspapers and formal TV programmes as they are considered too informal to use in most of the formal communication. However, with the growth of the internet, people started to communicate more in forums, newspaper commentary sections, in email exchanges, and instant private messages such as MSN, Paltalk, and Yahoo messengers. All these tools helped to generate more interaction between people speaking different dialects. In the early stages, differences were highly noticeable, which attracted researchers to work in the field of dialect

identification taking the advantage from the data generated from the various online forms of communication. Therefore, dialect identification has emerged in recent times as an important sub-field within computational linguistics, because it is now considered less expensive to develop resources. This in turn populated the field of dialect research by either supporting dialectal studies or offering better representation of a dialectal feature that can be distinguished by various methods and tools.

Modern research of dialect contains four factors that determine the differences of most research: type of feature; method used; number of dialects; and size of data. It is unfair to mention the results of any study without stating those four factors. Performance results improve when using a fewer number of dialects. In addition, results improve when studies are performed between non-closely related languages, where differences are highly notable, whereas other studies performed at the dialect level have often not produced as good results where differences can be difficult to identify even by humans.

In the case of the English language, Lui and Cook (2013) investigated cross-domain three-way national dialect classification between Australian, British and Canadian English. Their results demonstrated that there are lexical and syntactic characteristics of each national language variation that exist across several data sources such as web data, web government pages, and tweets. They found that the SVM classifier using bag-of-words features generally outperformed features based on syntax or character sequences when differentiating between Australian, British and Canadian English.

Ljubesic et al. (2007) used a character N-gram model in combination with a most frequent words list to distinguish between Croatian, Serbian and Slovenian-related languages using 13 thousand documents. Their study achieved high accuracies of over 99%. This research led to further work by Tiedemann and Ljubešić (2012) on investigating Bosnian, Croatian, and Serbian-related languages using a total of 600 documents. They performed an experiment using a Naïve Bayes classifier with word unigram features, achieving accuracies of 95%. More recently, Ljubešić and Kranjčić

(2015) distinguished Twitter users by language using very similar South-Slavic languages – Bosnian, Croatian, Montenegrin and Serbian. They applied the supervised machine learning approach by annotating a subset of 500 users from an existing Twitter account collected by user language. They showed that by using a simple bag-of-words model, and the univariate feature, they were able to achieve a 98% user classification accuracy using Multinomial Naïve Bayes.

Zampieri and Gebre (2012) explored computational techniques for automatic dialect identification of two variations of the Portuguese language – Brazilian Portuguese and European Portuguese. A character-based model that used 4-grams was reported to perform best compared to character N-grams of other lengths from 1-6. They used data collected from newswire containing one thousand documents divided between the two variations, and reported an accuracy of 99.8%. Later, Zampieri et al. (2013) applied an N-gram language model on four Spanish variations, Espagne, Argentine, Mexique and et Pérou, with a total of one thousand documents from newswires. They found that word 2-grams outperformed character N-grams of any length from 1 to 5. They also found that binary classification settings achieved significantly better results reporting an accuracy of 96.9% whereas, in comparison to the 4-way classification, this achieved an F-measure of 0.876.

In the Chinese context, Xu et al. (2017) performed 6-way, 3-way, and 2-way classifications in various greater Chinese dialects such as Mainland China, Hong Kong, Taiwan, Macao, Malaysia, and Singapore. They found that character bi-grams and segmented words work much better in Chinese then character unigrams do. This indicates that such longer units are more meaningful in Chinese and can better reflect the characteristics of a dialect. They performed 6-way classification via the linear kernel support vector machine using the LIBLINEAR library, achieving an accuracy of 82% on a total of 15 thousand text sentences collected from newswires.

In the Indian language, Kumar et al. (2018) identified Indian variations of Modern Standard Hindi (MSH), Braj, Awadhi, Bhojpuri, and Magahi using 10 thousand sentences of each variation. The study demonstrated that character N-gram were

more effective than word N-gram features. However, combining both character and word n-grams led to better results, achieving an accuracy of 96.4%.

2.4.3.1. Arabic dialect identification research

Arabic dialect identification is a crucial topic for most Arabic NLP research because of the diversity of Arabic dialects and the fact that Arabic is spoken in 20 different countries in the Middle Eastern region. Most of the early work on Arabic focused only on Modern Standard Arabic. Recently, there has been an increase in studies focusing on Arabic dialect identification due to the availability of NLP tools and resources that support the Arabic language.

The early work on Arabic was started by Zaidan and Burch (2011) which is considered one of the first attempts to investigate the Arabic dialects in depth. They created an Arabic Online Commentary dataset with a total of 108 thousand sentences which were labelled for MSA and three dialects – Levantine, Gulf, and Egyptian. The study reported an accuracy of 69.4% from a 4-way classification. However, for a 2-way classification using character-based N-gram and word-based N-gram features between Egyptian Arabic and MSA, the accuracy reached 87.9% using word-based unigrams (Zaidan and Callison-Burch, 2014). Likewise, the study by Elfardy and Diab (2013) performed 2-way classification of the Egyptian dialect and MSA using the AOC dataset. They applied the Naïve Bayes classifier using tokenisation to the sentence level, scoring an accuracy of 85.5%.

Darwish et al. (2014) performed a study of 2-way classification on Egyptian dialects and MSA. The study included a range of lexical and morphological features to classify a total of 700 tweets annotated evenly between the two variations. The accuracy reached 95% using the Random Forest classifier. Moreover, the research by Malmasi et al. (2015) examined a 6-way classification task using two thousand sentences of a multidialectal Arabic dataset (Bouamor *et al.*, 2014). Various character-based N-gram and word-based N-gram features were examined. The best result showed that by using the LIBLINEAR SVM classifier combining all character and word features, an accuracy of 74% was achieved.

Furthermore, Sadat et al. (2014) conducted an experiment using Markov models. Their result showed that the Naïve Bayes classifier performs better than the character N-gram Markov models for most Arabic dialects. The experiment was performed using data collected from 18 Middle Eastern countries with a total of 63 thousand sentences. They reported an accuracy of 98% at distinguishing among all the dialectal datasets. The study by Alshutayri and Atwell (2017) reported a classification accuracy of 79%. They performed the classification using Multinomial Naïve Bayes (MNB) using the *WordTokenizer* feature in Weka. Their training data contained 8,090 tweets, and testing on 1,764 tweets divided unequally between the five main Arabic dialects.

El Haj et al. (2018) presented experimental results from automatically identifying dialects. They performed 5-way classification tasks with a total of 16 thousand sentences using SVM. The study used subtractive bivalency profiling features combined with grammatical and stylistic features. The results showed that their classification methods can reach more than 76% accuracy using 10-fold cross validation.

Most of the work on dialect identification performed binary classification to identify two dialects (Elfardy and Diab, 2013; Darwish *et al.*, 2014; Zaidan and Callison-Burch, 2014). Other studies were performed the study with more dialects by using 4-way and 5-way classifications (Zaidan and Callison-Burch, 2011; El Haj *et al.*, 2018). However, according to Katakis et al. (2008), the more labels there are to categorise, the more complicated the identification task becomes.

There are two experimental settings used in most dialect studies: the first involves identifying short text represented by a single sentence or tweet. The second setting involves large text represented by paragraphs, multiple sentences, or multiple tweets. Most previous work on identifying Arabic dialects has involved classification of short text represented by sentences or tweets (Zaidan and Callison-Burch, 2011; Elfardy and Diab, 2013; Darwish *et al.*, 2014; Sadat *et al.*, 2014; Malmasi *et al.*, 2015; El Haj *et al.*, 2018).

Studies have performed experiments using different data sizes; for example, Darwish et al. (2014) performed the experiments using a total of 700 tweets. Malmasi et al. (2015) used sentences collected from the multidialectal parallel corpus of Arabic (MPCA). A total of 2000 sentences were translated by native speakers into five Arabic dialects. The studies by El Haj et al. (2018) and Sadat et al. (2014) used a total of 16,000 tweets and 63,000 sentences, respectively, to perform their experiments. Other researchers (Zaidan and Callison-Burch, 2011; Elfardy and Diab, 2013) used the AOC dataset, which is the closest dataset to the corpus of this current study, consisting of 108 thousand sentences collected from the commentary sections in popular Arabic newspapers.

However, the experiments in this chapter investigated a 7-way classification task including five main Arabic dialects in addition to Modern Standard Arabic and Classical Arabic. In addition, two experimental settings were used in this study: identification of short text represented by single tweet; as well as large text represented by multiple tweets composed by the same author. Furthermore, this study used over 112 thousand tweets from BTAC as a training set, and also performed the testing on an unseen test set consisting of over 6500 tweets. Table 2.7 below reports the recent publications for Arabic dialect identification.

Reference	Classifier	No. of dialects	Features	Accuracy	Data size
Sadat et al. (2014)	Naïve Bayes	18	Word N-gram	98.0%	63K sentences
Darwish et al. (2014)	Random Forest	2	Character N-gram	95.0%	700 tweets
Elfardy and Diab (2013)	Naïve Bayes	2	Tokenization sentence-level	85.5%	108k sentences
El Haj et al. (2018)	SVM	5	All features	76.2%	16K sentences
Malmasi et al. (2015)	linear SVM	6	All features	74.0%	2,000 sentences
Zaidan and Callison-Burch (2011)	Kneser-Ney	4	Word N-gram	69.4%	108k sentences

Table 2.7 Recent publications for Arabic dialect identification.

2.4.3.2. Studies on code-switching

Code-switching is the shift in spoken or written text between two languages or dialects. It also has been defined as the shift between two linguistic variations at the same moment (Scotton and Ury, 1977). It has been studied previously using data acquired from utterances as this is a common phenomenon among bilingual people in Arabic (Al-Rowais, 2012; Rivera, 2019). However, more recently, code-switching has been seen to emerge with textual data specifically in social media websites. In this review, we are focusing on studies that use textual data specifically for the Arabic language.

Several studies showed promising results for code-switching detection methods using two languages. Yeong and Tan (2010) used N-gram-based approaches to identify Malay-English vocabulary. The experiment results reported that using the syllable structure method was able to achieve 93.73% accuracy on 10,000 testing vocabularies. Oco and Roxas (2012) examined code-switching that occurs between Tagalog-English languages using pattern matching refinements (PMRs). They achieved an accuracy of 94.51% using a dictionary-based approach.

Lignos and Marcus (2013) detected code-switching between Spanish-English in Twitter data which occurred in 11% of the text. They achieved an accuracy of 96.9% using ratio list models which label each word according to dominant language models. Piergallini et al. (2016) used the Naïve Bayes algorithm to predict code-switching that occurs in Swahili-English languages, achieving an accuracy of 96.9%.

In the Arabic language, Samih and Maier (2016) proposed token- and text-level code-switching detection between MSA and Moroccan Arabic (Darija). Both variations share similar descriptive levels but differ in phonology, syntax, and lexicon (Ennaji *et al.*, 2004). They achieved 91.4% of accuracy at the token level using the Conditional Random Fields (CRF) classifier. Tarmom et al. (2018) detected the Egyptian Arabic dialect and English language that exists in Facebook pages. The study achieved a high accuracy of 99.8% using the compression-based approach. Furthermore, they achieved an accuracy of 97.8% in identifying the shift between the Saudi dialect, Egyptian dialect, and English language.

However, other studies performed in-depth code-switching prediction between two linguistics variations of a single language. Elfardy et al. (2014b) used the language model approach with a back off to a morphological analyser to handle out-of-vocabulary words to recognise code-switching points in Arabic. The approach yielded an F-measure of 0.86, and 0.20 for both token-level and tweet-level code-switching. Furthermore, Elfardy et al. (2014) examined the code-switching that occurs between Egyptian dialect and modern standard Arabic. They performed an experiment using a Naïve Bayes classifier and achieved an accuracy of 51.9%. The study used the AOC dataset of Zaidan and Callison-Burch (2011) containing 26k sentences comprising both Egyptian dialect and MSA.

Shrestha (2016) reported an F-measure of 0.34 for identifying code-switching using Conditional Random Fields (CRF). The experiment was performed on data consisting of 1,262 tweets where code-switching occurred in 214 tweets between Arabic dialect and MSA. El Haj et al. (2018) detected code-switching and bivalency that occurs in text. They used subtractive bivalency profiling to extract code-switching content as pre-processing to the dataset being used. This was achieved by examining dialect identification for the Arabic language. Overall, they achieved an accuracy of 76% for identifying the dialect on 16,494 sentences using the Support Vector Machine classifier. Moreover, they examined testing on a completely unseen test set consisting of 7,073 sentences and reported an accuracy of 66%.

2.5. Approaches for text categorisation

There are two main text categorisation approaches that are related to the approaches adopted in this thesis; *feature-based* approach which is based on the selection of features used by most of the machine learning algorithms, and the compression-*based* approach which is adopted by most of the character-based compression algorithms. Figure 2.2 shows a general overview of text categorisation techniques that are explained in this chapter.

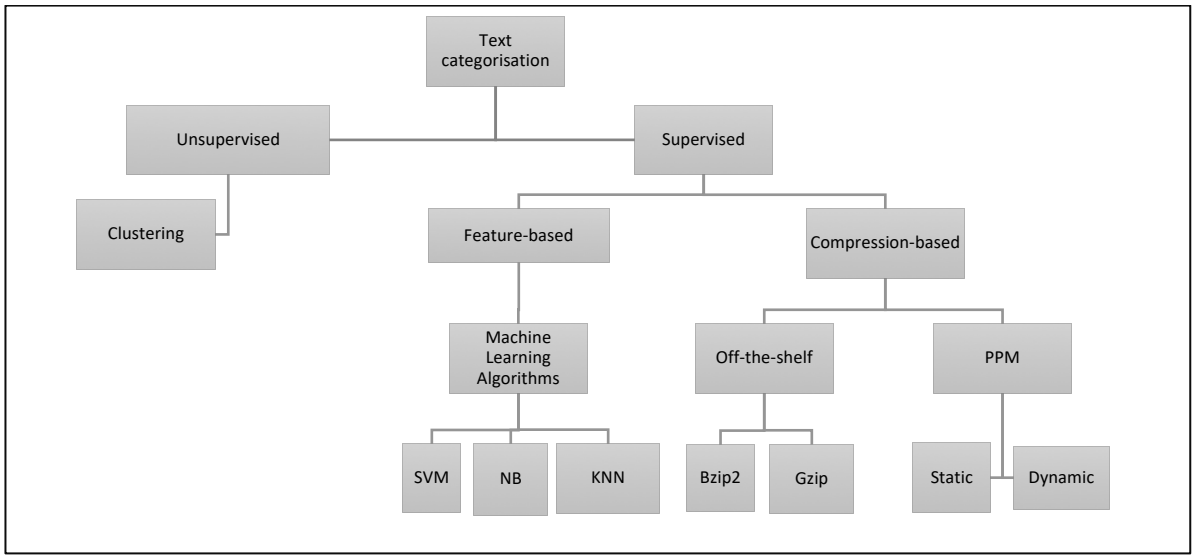

Figure 2.2 Overview of text categorisation techniques that are explained in this chapter.

2.5.1. Feature-based approach for text categorisation

The feature-based approach is based on traditional machine learning. It relies heavily on selecting a sequence of words and pre-processing the sequence to build the models. However, the feature-based approach usually requires pre-processing steps before applying the machine learning algorithms such as stemming, tokenisation, removal of stop words, and the calculation of word frequencies to help build word vector lists. In addition, feature selection is subsequently applied to determine the most important features in a text (Ta'amneh *et al.*, 2014).

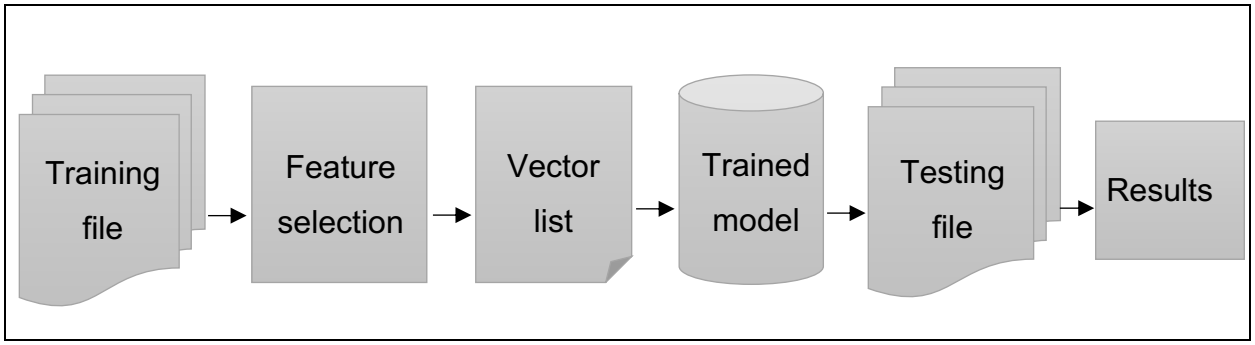

Figure 2.3 Workflow of feature-based approach used for text categorisation.

This thesis adopts the feature-based approach for comparison purposes only; three popular algorithms were selected that were used previously in text classification

research (Cavnar *et al.*, 1994; Koppel *et al.*, 2002; Sebastiani, 2002; Eyheramendy *et al.*, 2003; Fung, 2003; Duwairi, 2006; Mesleh, 2007; Alsaleem, 2011): K-nearest neighbours as a representation of similarity function and close distances approach; Multinomial Naïve Bayes which is based on the frequency count and probabilistic approach; and the Support Vector Machine which is based on statistical-based approach, which implements the "one-against-one" approach for multiclass classification.

2.5.1.1. K-nearest neighbours (K-NN)

The K-nearest neighbours algorithm is used for classification and regression. It is widely used in text classification tasks due to its robustness (Jiang *et al.*, 2007). The algorithm bases its theory on the closest distances measured among all observations. It categorises the problem depending on the number of votes of its neighbours; for instance, when a new document *(d)* is ready to be classified, a number of training documents are retrieved to predict the new documents. The algorithm finds the *K* nearest documents in the training set and assign the most common class to the new document. The distance between the new document *d* and the most common training document is calculated according to the standard Euclidian distance function (Mitchell, 1997). It is defined as follows:

$$d(x,y) = \sqrt{\sum_{i=1}^{N} \left(a_i(x) - a_i(y)\right)^2}$$

where x and y are two points in N-dimensional space and a_i is the value of the i_{th} attribute.

The algorithm is considered one of the lazy learning algorithms as it does not rely heavily on the training data to produce the result, where all the work happens in the testing phase when the prediction is requested (Altman, 1992). The K-nearest neighbours algorithm is used in this thesis as a representation of the instance-based

approach and compares its performance against character-based compression techniques.

2.5.1.2. The support vector machine (SVM)

The Support Vector Machine (SVM) was first introduced by Boser et al. (1992) and Cortesand Vapnik (1995) although the basis for SVMs has been around since the 1960s. It is a machine learning approach that is used for classification and recognition. It transforms the original training data into higher dimension data by using a nonlinear mapping then creates the best linear separating hyperplane known as the "decision boundary". Data from two classes can be divided by a hyperplane which can be used for categorisation (Han *et al.*, 2011). Given a set of N linearly separable points $S = \{x_i \in R^n | \ i = 1,2, \dots, N\}$, each point x_i belongs to one of the two classes, labelled as $\mathrm{y_i} \in \{-1, 1\}$. A separating hyper-plane divides S into two classes, each class containing points with the similar class label only. The separating hyper-plane can be identified by the pair (w, b) that satisfies

$$w.x + b = 0$$

$$\text{and } \begin{cases} w.x_i + b \geq +1 \text{ if } y_i = +1 \\ w.x_i + b \leq -1 \text{ if } y_i = -1 \end{cases}$$

for $i = 1,2, \dots, N$; where the dot operation $(.)$ is defined by:

$$w.x = \sum w_i \, x_i$$

for vectors w and x. Thus, the aim of the SVM learning is to find the best separating hyperplane that maximize the margin to nearest vector of both classes. This can be formularized as:

$$\text{minimize } \frac{1}{2} w.w$$

subject to

$$\begin{cases} w.x_i + b \geq +1 \text{ if } y_i = +1 \\ w.x_i + b \leq -1 \text{ if } y_i = -1 \end{cases} \text{ for i} = 1,2,\ldots,N$$

For the classification task, SVM makes a decision according to the hyperplane instead of going through the whole training set. It classifies the new test set according to location of the new vectors out of which side of hyperplane it lands on (Yang and Liu, 1999). Moreover, SVM is not only compatible with linear data but also works well with non-linear data. It uses kernel methods to avoid the explicit mapping of non-linear data by replacing its features by kernel functions such as Polynomial, RBF and, Sigmoid (Gharib *et al.*, 2009).

LibSVM is used in this thesis, a Java library for support vector machines (Chang and Lin, 2011). The library is currently one of the most widely used SVM software. From 2000 to 2010, there were more than 250,000 downloads of the library. LibSVM implements the “one-against-one” approach (Knerr *et al.*, 1990) for multiclass classification. It is also quick and flexible when it is used with text classification. For these reasons, this SVM package is used in this thesis as a representation of feature-based and statistical techniques and compared against the character-based compression techniques.

2.5.1.3. Naïve Bayes (NB)

Naïve Bayes is a simple probabilistic classifier based on applying Bayes' theorem which was named after Thomas Bayes (1702–1761). It assumes that features are independent given the class. Given a test example t, based on the feature values, Naïve Bayes allocates the test example to the class with the highest probability (Mitchell, 1997).

The probability that the test example t belongs to a specific class C can be estimated as follows:

$$P(C|t) = \frac{P(C).P(t|C)}{P(t)}$$

where $P(C)$ is the probability of a class calculated from the number of documents in the category divided by number of documents in all categories. $P(t)$ is the probability of a test document and is a constant that can be ignored. $P(t|C)$ is the probability of the test document given the class. Documents can be represented by a set of words:

$$P(t|C) = \prod_i P(word_i|C).$$

This can be rewritten as:

$$P(C|t) = P(t) \prod_i P(word_i|C)$$

where $P(word_i|C)$ is the probability that a given word occurs in all documents of class C, and this can be computed as follows:

$$P(word_i|t) = \frac{W_{ct} + 1}{N_c + |V|}$$

Where W_{ct} is the number of times that the word occurs in class C, N_c is the total number of words in class and V is the size of the vocabulary. Finally, 1 is added to the W_{ct} to avoid zero probability.

In this thesis, Multinomial Naïve Bayes is specifically used as another representative of feature-based and probabilistic techniques and compared against the compression character-based techniques. It works well when words are represented in terms of their occurrences (frequency count). It is also found to be almost uniformly better than the multi-variate Bernoulli Naïve Bayes model (McCallum and Nigam, 1998). Also, it outperforms other Naïve Bayes models such as Poisson and Bernoulli (Eyheramendy *et al.*, 2003).

2.5.2. Character-based compression approach for text categorisation

An alternative approach to the feature-based method is to adopt an information theoretic approach instead and apply a character-based approach based on compression. Character-based compression models process each character in the text sequentially. Using the character-based approach can help to avoid the aforementioned pre-processing steps faced in the feature-based approach such as feature selection, stemming, and tokenisation (Teahan and Harper, 2003).

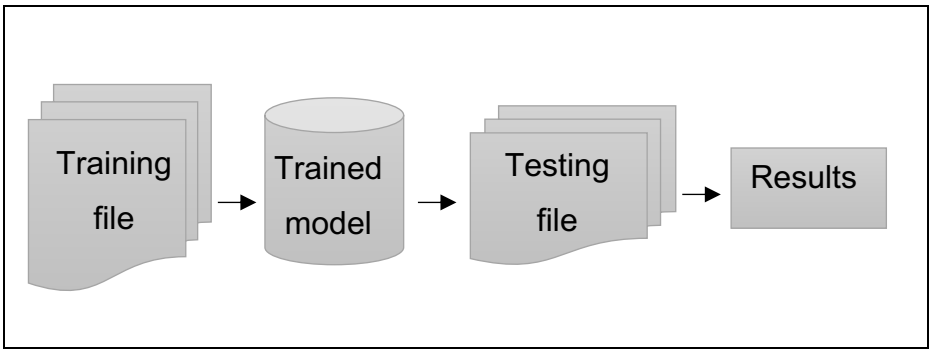

Figure 2.4 Workflow of character-based approach used for text categorisation.

This research uses the character-based approach to sidestep the characteristics related to Arabic text such as inflection and diacritics. Many researchers have only relied on the main 28 Arabic letters and ignored the rest of the characteristics because of their effect on the feature extraction process (Abbasi and Chen, 2005b). However, these characteristics are an important and unique aspect of Arabic text, and should be considered when performing categorisation, and should not apply normalisation or stemming during tokenisation as it may lead to mis-identification of the author of the text (Alwajeeh *et al.*, 2014). The character-based compression approach is discussed in more details below in section 2.5 as it is the main method adopted and relates to the research questions / objective(s). The next section lists the most important applications for text classification.

2.6. Overview of compression algorithms

The idea of compression is to encode a symbol to reduce its size in order to transmit the message more efficiently. Thus, the encoded symbol is transmitted to the receiver which is then decoded in order to retrieve its original form of the symbol (Nelson and Gailly, 1996). There are two ways to compress data – lossless and lossy compression. *Lossless* compression ensures that any data being compressed is not lost. It is used for text compression and for binary data files, such as documents, database files, and computer applications. However, *lossy* compression loses some of the non-essential data during the process of compression. Lossy compression has become increasingly popular particularly with the need to compress multimedia data such as images, videos and audio data (Memon *et al.*, 1999).

Lossless compression algorithms can be categorised into three main methods – dictionary-based, context-based and transform-based – as illustrated in Figure 2.5.

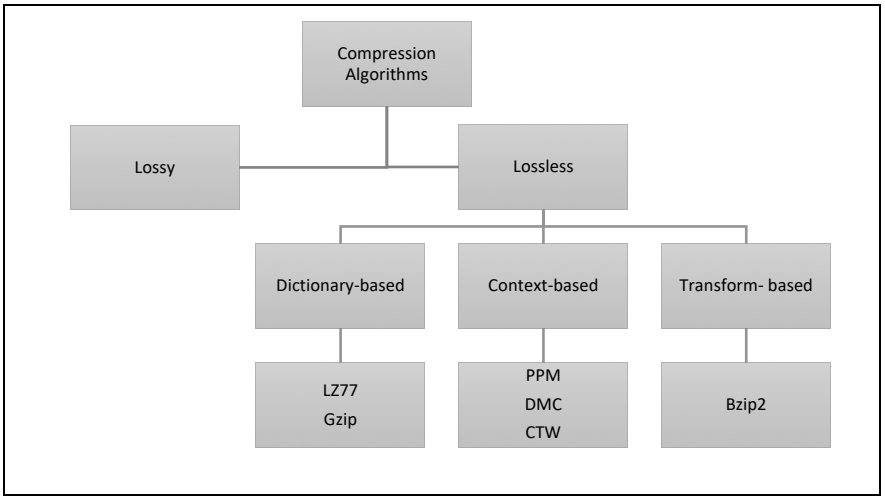

Figure 2.5 Overview of compression-based algorithms.

2.6.1. Dictionary-based lossless compression methods

A dictionary-based approach is an adaptive approach based on building a dictionary. The encoding part works when reading in the input stream by looking for repeated groups of symbols that appear in a dictionary. If a symbol match is found, a pointer or index into the dictionary can be sent instead of the code of the symbol. The decoding

part works simply by replacing the dictionary pointers with the original symbol according to the entered stream (Nelson and Gailly, 1996). A good example of a dictionary-based approach is LZ77 (Ziv and Lempel, 1977). It uses a sliding window technique in which it consists of two segments – a *search* buffer and a *look-ahead* buffer. The encoder attempts to read symbols from the look-ahead buffer and find a match in the search buffer. When the match is found, the encoder produces a list of seen occurrences with a reference containing the following: 'offset'; 'match length'; and 'next symbol'. Unseen symbols being encoded are included in the compressed stream as "literals". Another method, the LZ78 algorithm, replaces a reference in the dictionary instead of adding pointers (Ziv and Lempel, 1978). A variant of the LZ78 algorithm known as LZW (Welch, 1984) initialises the dictionary to the set of all input characters, which ensures a match in the dictionary for any input. Therefore, it eliminates the need for adding "literals" in the compressed stream (Bratko, 2012).

2.6.2. Context-based lossless compression methods

In contrast to dictionary methods, the context-based approach encodes single characters one at a time using the probability of a character's appearance. Each code is constructed in such a way that more probable symbols are represented with fewer bits so that the compression code length is minimised. One of the context-based lossless compression methods is prediction by partial matching (PPM). The basic idea of PPM is to use the last few characters in the input to predict the upcoming one. Other methods such as dynamic Markov compression predict the symbol like PPM except they encode the predicted input one bit at a time rather than one byte at a time the way that PPM does (Cormack and Horspool, 1987).

In general, the dictionary-based approaches, such as Ziv-Lempel, are known for speed and memory efficiency whereas a context-based approach is known for better accuracy in compression (Bell *et al.*, 1989).

2.6.3. Transform-based lossless compression method

The transformation-based approach is more recent than the dictionary and context-based approaches. A popular example of this method is the Burrows-Wheeler transform (Burrows and Wheeler, 1994). It uses a block-sorting compression that works by re-ordering a group of symbols together into runs of similar symbols. The algorithm tends to put the same characters of the original sequence next to each other making it easier for the algorithm to compress the data. The decoding part of the algorithm works by sorting the last column of the table generated by the algorithm and retrieve the original sequence. The cost of using this method is the computational sort-based process used when encoding and decoding (Salomon, 2004). A popular software package, bzip2, is a free and open-source file compression program that uses the Burrows-Wheeler algorithm.

2.7. Compression techniques for text classification

It is hard to determine who first suggested using compression for classification. The compression-based approach is considered a non-standard approach for classification, but it is adopted for classification to overcome problems with the feature-based approach (Marton *et al.*, 2005). Assuming the features are generally words, problems arise with the feature-based approach such discarding part of the text by feature selection, specifying the feature boundary, ignoring the morphological variants of the features, and neglecting other non-textual features such as numbers and symbols (Frank *et al.*, 2000).

Compression-based text classification methods help to overcome the problems above. By adopting a character-based approach, an overall compression of an article can be compared with other training models to find which model compresses the article best and then choosing the class associated with the model. This approach has several advantages such as: avoiding all the pre-processing steps performed by feature selection; ensuring that part of the text is not discarded since the whole text consists of sequences of the characters being considered; automatically dealing with non-word features of a document such as number and symbols; and automatically dealing with

different types of documents such as non-arbitrary files in a computer system (Marton *et al.*, 2005).

Many studies have investigated compression methods for classification. Marton et al. (2005) tested three compression methods – RAR, Gzip, and LZW – on the English language applied to the problems of genre categorisation and authorship attribution. RAR always performed much better than LZW and Gzip. On the other hand, PPM was compared with a combination of other compression algorithms including RAR, Gzip, and Bzip2 for authorship attribution. PPM was reported to perform better than the other compression methods (Khmelev and Teahan, 2003; Thomas, 2011).

On the Arabic language, Ta'amneh et al. (2014a) tested three compression algorithms – RAR, Gzip, and LZW on genre categorisation. RAR always produced a more accurate classification than either LZW or Gzip. Alhawiti (2014) compared PPM with other compression methods, Gzip, ABC2.4, and Bzip2. PPM outperformed other methods by using bi-graph substitution for PPM (BS-PPM) designed specifically for the Arabic language. In this thesis, PPM is used for its excellent overall ability at both compression and classification compared with the other compression-based techniques.

2.8. Prediction by Partial Matching (PPM)

The Prediction by Partial Matching (PPM) text compression technique for lossless data is based on the adaptive context modelling family which uses a fixed number of preceding characters according to a selected maximum fixed order to predict the coming character. For example, if the selected maximum order is three, the prediction of the following character will be based on the previous three characters. PPM moves from the maximum highest order down to lower orders using the escape mechanism whenever a previously unseen symbol is encountered. This process will be continued until the lowest default order of -1 is reached, where all character probability are equiprobable. It has shown excellent performance in many NLP tasks, such as text correction and language identification (Teahan and Cleary, 1997).

PPM has gone through many developments with variations such as PPMA and PPMB (Cleary and Witten, 1984), PPMC (Moffat, 1990), PPMD (Howard, 1993), PPM* (Cleary and Teahan, 1997) and PPMO (Wu and Teahan, 2008). For PPMC, the probability P_{PPMC} for the next character φ is given by:

$$P_{PPMC}(\varphi) = \frac{c_d(\varphi)}{T_d}$$

where the currently used coding order is specified by d, the total amount of times that the current context $c_{i-5} \cdots c_{i-1}$ has occurred is indicated by $T_d(c_{i-5} \cdots c_{i-1})$. $c_d(c_i|c_{i-5} \cdots c_{i-1})$ represents the total number of occurrences for the symbol φ in the current context. The estimation of the escape probability E by PPMC is as follows:

$$E_{PPMC} = \frac{t_d}{T_d}$$

where the total number of times that a unique character has occurred following the current context is represented by t_d.

In this thesis, PPMD is used instead of PPMC as it produces better compression results with Arabic (Alhawiti, 2014; Aljehane, 2018). Also PPMD is improved variation of PPMC invented by Howard (1993) which often results in better compression. It is similar to PPMC except that it makes use of new symbol by adding ½ instead of 1 to both escape and counts for each new symbol. The formula for estimating the probability P for the next character φ is given by:

$$P_{PPMD}(\varphi) = \frac{2c_d(\varphi)-1}{2T_d}, \quad (1)$$

and the escape probability is estimated as follows:

$$E_{PPMD} = \frac{t_d}{2T_d}. \quad (2)$$

Many researchers have investigated PPM. Khmelev and Teahan (2003) applied the compression-based method to the authorship attribution problem in English; the goal was to find duplicated documents and plagiarism by comparing several compression

algorithms used for identification purposes. PPMD order 5 performed well achieving 89.2%. Teahan and Harper (2003) performed text categorisation using PPMD achieving 91.1% compared to 84.8% using Naïve Bayes classifier.

Bobicev (2007) undertook a comparative experimental study of two PPM-based text categorisation methods – a comparison of word-based and character-based methods. The results showed that the word-based method is not better than the character-based method even though the differences are very small.

In the study by Bratko et al. (2006), the character-based PPM models were used for spam detection in a binary classification: *spam* versus *valid email*. The models they created were applied to the spam-filtering task and showed better results than other machine learning approaches, demonstrating that data-compression models are well suited to the spam-filtering problem.

Frank et al. (2000) undertook extensive experiments on the use of compression models for categorisation. They produced some promising results, but they found that other techniques such as machine learning should be considered, and more evaluation was needed of the performance between compression-based methods and the state-of-the-art machine learning methods.

2.9. PPM compression-based language model

PPM compression encodes a text while building a model for it. Each single symbol is encoded within the context provided by the previous symbols appearing in the document. For a given text, T , we can achieve the best compression for a model p_M for document D by observing that

$$H(T, p_M, D) =$$

$$= -\frac{1}{n}\log_2 p_M(D), \qquad D = x_{1n}$$

$$= -\frac{1}{n}\log_2 \prod_{i=1}^{n} p_M(x_i|context_i) \quad \text{[by Chain Rule]}$$

$$= \frac{1}{n}\sum_{i=1}^{n} -\log_2 p_M(x_i|context_i),$$

where $context_i = x_1, x_2, \ldots, x_{i-1}$. Thus, each symbol is encoded according to its information contained within the context provided by all the previous symbols. In practice, this is generally not possible so PPM applies the Markov assumption by assuming a fixed-order context.

2.10. Using PPM for categorisation

PPM is used for classification by simply selecting the class related with the model that best compresses the text. The main idea is to predicate the correct class of text T using the formula:

$$\hat{\theta}(T) = argmin_c H(T|S_c)$$

where $H(T|S)$ is some approximation of relative entropy of text T with respect to text S and the class c is chosen from the model with the minimum value. In this case, it is estimated using the PPM compression scheme i.e. for an order five model, it is calculated using the following formula:

$$H(T|S) = -\sum_{i=1}^{n} \log_2 P(c_i|c_{i-5} \cdots c_{i-1}) \quad (3)$$

where n is the length of the text and the probabilities for each character are calculated using the PPM Markov-based modelling method which estimates the probability of the next character (see formulas (1) and (2) for PPMD) based on the context of the previous five characters.

2.10.1. Example of PPM

Table 2.8 below shows an example of how the PPMC processes the string “I have a dream. I have a dream. I ha” using different orders K= 2, 1, 0 and -1, where K means the prediction of the upcoming character will be estimated based on the (number of K) preceding characters. Usually, each character will be encoded arithmetically with the probability estimated by the model (Witten *et al.*, 1987). Although for the purposes of

classification, the arithmetic coding step can be eliminated since the physical process of writing to a file on disk is not required and only the modelling step is required.

Order K=2			**Order K=1**			**Order K=0**			**Order K= -1**		
Prediction	c	p	Prediction	c	p	Prediction	c	p	Prediction	c	p
l_ →h	3	3/4	l →_	3	3/4	→l	3	$\frac{3}{46}$	→ A	1	$\frac{1}{\lvert A\rvert}$
→esc	1	1/4	→esc	1	1/4						
_h →a	3	3/4	_ →h	3	3/13	→_	9	$\frac{9}{46}$			
→esc	1	1/4									
_a →_	2	2/3	→a	2	2/13	→h	3	$\frac{3}{46}$			
→esc	1	1/3									
_d →r	2	2/3	→d	2	2/13	→a	7	$\frac{7}{46}$			
→esc	1	1/3									
_l →_	2	2/3	→l	2	2/13	→v	2	$\frac{2}{46}$			
→esc	1	1/3									
ha →v	2	2/3	→esc	4	4/13	→e	4	$\frac{4}{46}$			
→esc	1	1/3									
av →e	2	2/3	h →a	3	3/4	→d	2	$\frac{2}{46}$			
→esc	1	1/3	→esc	1	1/4						
a_ →d	2	2/3	a →v	2	2/9	→r	2	$\frac{2}{46}$			
→esc	1	1/3									
am →.	2	2/3	→_	2	2/9	→m	2	$\frac{2}{46}$			
→esc	1	1/3									
ve →_	2	2/3	→m	2	2/9	→.	2	$\frac{2}{46}$			
→esc	1	1/3	→esc	3	3/9						
e_ →a	2	2/3	v →e	2	2/3	→esc	10	$\frac{10}{46}$			
→esc	1	1/3	→esc	1	1/3						
ea →m	2	2/3	e →_	2	2/6						
→esc	1	1/3									
dr →e	2	2/3	→a	2	2/6						
→esc	1	1/3	→esc	2	2/6						
re →a	2	2/3	d →r	2	2/3						
→esc	1	1/3	→esc	1	1/3						
m. →_	2	2/3	r →e	2	2/3						
→esc	1	1/3	→esc	1	1/3						
._ →l	2	2/3	m →.	2	2/3						
→esc	1	1/3	→esc	1	1/3						
			. →_	2	2/3						
			→esc	1	1/3						

Table 2.8 The generation of PPMC model after processing the string "*I have a dream. I have a dream. I ha*" using maximum order 2. (The space is represented by _ in this table). esc refers to escape; c indicates count; p refers to the probability.

Imagine three scenarios where two subsequent letters – "ve", "te", and "rm" – are encountered after the sentence "I have a dream. I have a dream. I ha" has already been seen (see Table 2.9). First, for encoding "ve" following "ha", using maximum order of two, in this situation the probability is estimated as $\frac{2}{3}$ for each letter ('v' and

'e'), since the context and predictions, ha→v, and av→e are found in order two (K=2) context (see Table 2.8). This requires 1.21 bits $[-\log_2\left(\frac{2}{3}\times\frac{2}{3}\right)]$ to encode. (As stated, PPM normally uses arithmetic coding to physically encode the probabilities which results in the code length being close to the theoretical optimum which is $-\log_2 p$ where p is the probability being encoded. However, when using PPM for text classification purposes, there is no need to physically encode the probabilities and instead, PPM computes the theoretical code lengths directly and uses that as the categorisation measure.)

However, if "te" needs to be encoded following "ha", the escape probability of $\frac{1}{3}$ will be encoded from order two because the letter "t" was not seen in that context after following the "ha". Then the process will move down to order one, and the escape probability $\frac{3}{9}$ will need to be encoded again because the letter "t" was also not seen in order one after following "a". Next, the escape probability $\frac{10}{46}$ will be encoded a third time because the letter "t" was also not seen in order zero. Finally, the process will move down to order -1 where the letter "t" is found, so the encoded probability will be $\frac{1}{A}$ where A is the alphabet size (256 for a standard byte-based encoding 8 bits). Moreover, the second letter "e" will be encoded with probability $\frac{1}{3}$ after escaping because the context "at" was not seen in order two. After that, the escape probability $\frac{3}{9}$ will be encoded again because the letter "e" was also not seen in order one after following the "a". Next, the letter "e" is seen in order zero where the encoded probability will be found $\frac{4}{46}$.

The total probability for encoding the letter "t" is $(\frac{1}{3}(esc)\times\frac{3}{9}(esc)\times\frac{10}{46}(esc)\times\frac{1}{A})$, and the letter "e" is $(\frac{1}{3}(esc)\times\frac{3}{9}(esc)\times\frac{4}{46})$. which requires 20.2 bits to encode both letters (see Table 2.9).

Finally, If the aim is encoding the two letters “rm” after seeing “ha”, then the escape probability will be $\frac{1}{3}$ for order two and $\frac{3}{9}$ for order one because the letter “r” was not seen in both orders, before the letter is found in order zero where the encoded probability will be found $\frac{2}{46}$. Similarly, the letter “m” will result in the encoding of an escape probability of $\frac{1}{3}$ for order two, $\frac{3}{9}$ for order one, until the letter is seen in order zero where the encoded probability will be found $\frac{2}{46}$. The total probability to encode the letter “r” is $(\frac{1}{3}(esc) \times \frac{3}{9}(esc) \times \frac{2}{46})$, and the letter “m” is $(\frac{1}{3}(esc) \times \frac{3}{9}(esc)\ \frac{2}{46})$ which requires 15.5 bits to encode both characters (see Table 2.9).

Text	Subsequent letters	Codelength being used
..dream I ha..	ve	$-\log_2\left(\frac{2}{3}\times\frac{2}{3}\right) = 1.21$ bits
	te	$-\log_2\left((\frac{1}{3}\times\frac{3}{9}\times\frac{10}{46}\times\frac{1}{\mathrm{A}})(\frac{1}{3}\times\frac{3}{9}\times\frac{4}{46})\right) = 20.2$ bits
	rm	$-\log_2\left((\frac{1}{3}\times\frac{3}{9}\times\frac{2}{46})(\frac{1}{3}\times\frac{3}{9}\times\frac{2}{46})\right) = 15.5$ bits

Table 2.9 Encoding three sample characters using PPMC.

2.10.2. Example of using PPM for authorship identification

The following example shows how the PPM compression scheme can be used to identify authorship of two popular speeches:

Martin Luther King speech, “I have a dream” (King, 1963):

> *“I have a dream that my four little children will one day live in a nation where they will not be judged by the color of their skin but by the content of their character. I have a dream. I have a dream that one day on the red hills of Georgia, the sons of former slaves and the sons of former slave owners will be able to sit down together at the table of brotherhood.”*

John F. Kennedy. “We choose to go to the moon" (Kennedy, 1962)”

"We choose to go to the moon. We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard, because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one which we intend to win, and the others, too."

Testing	Martin L. King Model (Codelength)	John F. Kennedy Model (Codelength)	Expected author
I have a dream that one day this nation will rise up and live out the true meaning of its creed.	267.694 bits (2.788 bits/char)	368.914 bits (3.843 bits/char)	Martin Luther King
The Mariner spacecraft now on its way to Venus is the most intricate instrument in the history of space science.	494.961 bits (4.419 bits/char)	468.522 bits (4.183 bits/char)	John F. Kennedy

Table 2.10 Example of predicting authorship using PPMD order 5.

The table above shows the codelength compression of both sample speeches of Martin L. King and John F. Kennedy. In the first row, the codelength shows that the expected author is Martin L. King based on the minimum codelength of 267.6 bits. This value is achieved after building an order 5 model by training on speech samples from Martin L. King and calculating probability of each letter occurring as shown in Table 2.8. Then the codelength is calculated using formula (3) as shown in Table 2.9. Similarly, in the second row, the codelength shows that the expected author is John F. Kennedy based on the minimum codelength of 468.5 bits.

2.11. Procedure used for compression-based classification experiments

Marton et al. (2005) provided an overview for three compression-based classification procedures: standard MDL (minimum description length); approximate MDL (AMDL); and the best-compression neighbour (BCN) procedures. MDL and AMDL perform classification on models trained using all the training text concatenated for each class.

On the other hand, BCN performs classification using separate models for each training document (i.e. a separate non-concatenated text for each document). The study by Marton et al. (2005) produced results for two procedures – AMDL and BCN. They found that AMDL produced the best result when using RAR compression techniques while BNC produced the best result using the GZIP compressor.

Thomas (2011) introduced four procedures which he called "protocols", as shown in Table 2.11. Three of these protocols were mentioned by Marton et al. (2005). However, Thomas separated these protocols according to concatenated *versus* non-concatenated models and dynamic *versus* static models. Concatenated models are trained on all text available for each class concatenated together into a single training text used to train a single class model. Non concatenated models use separate models trained on each training document. Dynamic models update continuously during the testing process whereas the static models remain fixed once the training process is completed.

Protocols	Static Models	Dynamic Models
Concatenation of training texts for each class	SMDL (Protocol I)	AMDL (Protocol II)
Non-concatenation of training texts for each class	Protocol III	BCN (Protocol IV)

Table 2.11 Protocols for text categorisation. Source: Thomas (2011).

Thomas (2011) examined all four protocols. He found that models using concatenated training texts outperformed non-concatenated models in nearly every experiment for all corpora. In addition, concatenated models require less processing time due to the large amounts of calculations required to perform for each separate non-concatenated model. Moreover, he found that dynamically concatenated models slightly improve the accuracy over statically concatenated models.

In this thesis, after labelling each tweet according to the gender, authorship and dialect, all the tweets were concatenated into separate files for each corresponding class during the training process. The setup for each experiment is described in each

of chapter 4, 5, and 6. (Two new protocols that use training text organised according to a secondary class type are introduced in Chapter 5).

2.12. Evaluation Metrics

Evaluation techniques are required in order to measure the success of classification-based research. It is difficult to compare results with other research without having the same metric. These metrics are used in the experimental evaluation in chapters 4, 5, 6 and 7.

- **Contingency table**

A contingency table is used to measure the relationship between two or more variables when performing a classification task. The table below shows how a contingency table is created. Each entry in the table (clockwise from top left) represents the following: the number of true positive items which are the number of cases where the prediction matches the correct label; the false positive items which are the number of cases where the prediction matches the incorrect label as positive; true negative items which are the number of cases where the prediction matches the correct negative label; and false negative items which are the number of cases where the prediction matches the incorrect negative label.

	Correct	**Not correct**
Selected	True positives	False positives
Not Selected	False negatives	True negatives

Table 2.12 Contingency table.

The following evaluation measures described below are calculated from the confusion matrix.

- **Precision**

Precision indicates the number of selected items that are correctly predicted as positive. It consists of all the true positive items divided by all the items that are predicted positive. It is defined as:

$$\frac{\#\ \text{True positives}}{\#\ \text{True positives} + \#\ \text{False positives}}.$$

- **Recall**

Recall indicates the proportion of actual classes correctly categorised as positive. It consists of all the true positive items divided by all the items that are actually positive. It is defined as:

$$\frac{\#\ \text{True positives}}{\#\ \text{True positives} + \#\ \text{False negatives}}.$$

- **Accuracy**

Accuracy is the measurement of all items that have been classified correctly. It consists of all the true positive and negatives items divided by all the items being classified. It is defined as:

$$\frac{\#\ \text{True positives} + \#\ \text{True negatives}}{\#\ \text{True positives} + \#\ \text{False positives} + \#\ \text{False negatives} + \#\ \text{True negatives}}.$$

- **F1-Measure**

F1-Measure is the combination of both precision and recall. It is defined as:

$$\frac{2\ Recall\ \times\ Precision}{Recall + Precision},$$

2.13. Conclusion

To conclude, the field of text categorisation is full of interesting challenges. We have first reviewed several tasks of text categorization. Many researchers have tackled this

field in order to solve some of the text categorisation tasks. Some of these tasks are fully studied with various techniques while others still need further investigation, specifically using text compression for Arabic text classification.

We have reviewed the basic fundamentals of Arabic language structure, with an emphasis on several challenges for text categorisation tasks such as inflection, diacritics, and word length. The main question in this thesis is whether applying techniques on the character-based approach is effective for Arabic text, knowing the complex structure of Arabic.

We have also reviewed the current dialectal corpora of Arabic existing under specific criteria. We also set the background for previous work on authorship attribution with an emphasis on work involving the Arabic language. We provide the background for previous work on gender categorisation with an emphasis on data taken from social media. In addition, we set the background and related work on dialect identification with an emphasis on Arabic dialects studies.

This chapter has also reviewed the difference between the types of data compression, lossless and lossy, and the uses of both techniques. Then, the main adopted method Prediction by Partial Matching (PPM) is reviewed and explained in detail. In addition, other machine learning approach which have been used for comparison purposes are discussed such as K-nearest Neighbours, Multinomial Naïve Bayes, and the Support Vector Machine.

Chapter 3

Creating the Bangor Twitter Arabic Corpus

3.1. Introduction

Corpora refer to sets of sorted text or speeches saved structurally in machine-readable form (McEnery and Wilson, 2003; Kennedy, 2014). Corpora have been used by many natural language researchers to facilitate tagging, segmentation, categorisation, and compression; for example, the application of corpora in NLP has increased in the last 20 years due to the significant improvements in hardware and software used to manage the task. Twitter text is considered a unique style of writing that is relatively new compared to texts found in other more traditional corpora. The peculiar nature of Twitter text is the short length of tweets, the availability of data, and the type of language used. Therefore, a corpus containing Arabic Twitter text in particular, which is under-resourced, would provide an interesting resource for researchers. All of these characteristics helped to motivate the work described in this chapter.

This chapter is an extension of the paper published as follows:

> *Altamimi, M., Alruwaili, O. and Teahan, W.J., "BTAC: A Twitter Corpus for Arabic Dialect Identification." In the 6th Conference on Computer-Mediated Communication (CMC) and Social Media Corpora (CMC-corpora 2018) (p. 5) 17-18 September 2018, University of Antwerp, Belgium.*

The rest of the chapter is organised as follows: section 3.2 discusses the purpose of creating BTAC corpus; section 3.3 provides background to the Arabic language dialects; section 3.4 describes creating the Bangor Twitter Arabic Corpus (BTAC) which involves the collection process, the pre-processing steps, and the annotation process of the corpus; section 3.5 analyses the corpus after the dialect and genre annotation; section 3.6 reports the methods used for evaluating the quality of the

corpus along with the corresponding experimental results; and finally section 3.7 concludes the chapter.

3.2. Purpose of creating the corpus

As stated, the shortage of Arabic language resources in the field of corpus linguistics compared to other popular languages such as English, Chinese and Spanish inspired this work. Many Arabic researchers have worked hard to establish resources in the last decade. The research in the field of dialectal Arabic is still limited due to the relative unavailability of resources and the time-consuming nature of the task needed to create and process these corpora (Jarrar *et al.*, 2017).

In this chapter, the focus is on creating a dialectal Arabic corpus for the research community. Newspaper commentaries sections and forums were used in the past as sources for Arabic dialectal corpora. However, dialectal resources have witnessed growth recently because of the ready availability of web-based resources in the form of textual data from social media websites, unlike in the past. This massive increase of dialectal resources has provided the incentive to produce this work. The corpus is planned to support various Arabic studies that depend on authentic data in addition to text analysis areas such as dialect identification, code-switching and other classification tasks such as gender categorisation, authorship attribution, and genre categorisation.

3.3. Arabic dialects background

There are many Arabic dialects that are spread throughout the Middle East. However, there are five main dialects that are widely spread: the Gulf; the Egyptian; the Maghrebi; the Levantine; and the Iraqi. The Gulf dialect is found in countries such as Saudi Arabia, Kuwait, Bahrain, Qatar, Emirates, and Oman. The Egyptian dialect is widely spoken only in Egypt. The Maghrebi dialect includes dialectal variations from countries such as Morocco, Tunisia, Libya, and Algeria. The Levantine dialect is found in countries such as Syria, Lebanon and Jordan, and in Palestine. Finally, the Iraqi

dialect is spoken only in Iraq. Other dialects such as Sudanese, Somali, Yamani, and Mauritani are not included in our research due to the lack of use in social media for these dialects (Salem, 2017).

Sometimes, countries are affected by dialects according to the geographical region. The region south of Iraq and Jordan is affected by the Gulf dialects. West of Iraq is close to the Levantine dialects. In addition, dialectal Arabic sometimes overlaps within MSA whereas differences can be highly noticed in other variations. For example, the verb "انظر" [B: "anzr" E: "look"] is widely used differently in most of the Arabic dialects, MSA, and CA. The dissimilarity can be seen more than once with each dialect according to the graphical region of the country. Also, similarity can be seen with the same verb "شوف" [B: "shwf", E: "look"] in Egyptian, Iraqi, and Gulf dialects – see Table 3.1 above for each verb with the corresponding dialects and transliteration.

	MSA		CA		Egyptian		Iraqi		Gulf		Levantine		Moroccan	
Word	انظر شاهد اطلع	anzr shahd atl'e	رِ	ـرَ	بص شوف	bs shwf	شوف باوع دحق عاين	shwf baw'e dhq 'eayn	راع شوف ناظر عاين تفرج	ra'e shwf nazr 'eayn tfrj	لد إطلع طل	ld etl'e tl	اشبح هكشف بهت	ashb hhks hf bht

Table 3.1 Example of the verb "look" which shows differences among Arabic dialects.

3.4. Creating the Bangor Twitter Arabic Corpus (BTAC)

In the BTAC corpus, the focus was to create a corpus that contains text from social media. Twitter in the form of tweets is considered challenging for many reasons: tweets are written using only 280 characters making it more difficult to classify them; text is written informally; and the frequency of misspelling and slang in twitter text is high (Çoban *et al.*, 2015). Over 122K tweets were annotated manually according to the five main dialects in addition to Modern Standard Arabic and Classical Arabic. Code-switching is also identified when mixed dialects occur for further research analyses. In this research the focus is on the change from one dialect to another within

each tweet. Code switching occurs often in Arabic conversations as people shift from a dialect to MSA or CA.

Also, the corpus was manually annotated according to the genre along with the dialects. In addition, the tweets were checked by an expert for further evaluation (see section 3.4.3 below on annotation process of BTAC). Several processing steps were performed on the tweets (see section 3.4.2 on processing steps). It is not assumed the tweets belong to a specific dialect by the tweet or username location. Each tweet is checked to assess which dialect it belongs to. Table 3.2 lists Bangor Twitter Arabic Corpus characteristics.

Size	Medium (From 1 to 3 million words)
Text Languages	Multi-dialectal
The language of Texts	Arabic
Mode	Written text
Nature of Data	Specialised (dialect)
Nature of Application	Categorisation
Authorship	Multi-author
Annotation	Manual annotation according to the dialect and genre
Access	Free

Table 3.2 Bangor Twitter Arabic Corpus characteristics.

The corpus employs a judgment (e.g. purposive) sampling technique rather than collecting a large number of tweets. This is achieved by four criteria: selection of the dialect; selection of the gender; selection of the interest; and collection of data according to specific requirements (see section 3.4.1.1). The corpus contributions can be summarised as follow:

- Over 200K tweets were collected and used to build a new dialectal corpus for Arabic tweets.
- After cleaning, over 122K tweets were labelled into five dialects in addition to MSA and CA.

- The corpus includes other labels for each tweet such as gender, authorship, and genre to allow researchers to perform other types of text analyses. Also, mixed dialects are tagged for further code-switching research.
- The corpus is checked and evaluated by an expert.

3.4.1. Collection process

The aim is to create a corpus that contains high-quality ground truth data. In Twitter, accounts are created using name, location, profile pictures, and biography. The selection process used these information to identify over 101 users from different locations in the Middle East. The selection process involved users from the five main dialects: Egyptian; Gulf; Iraqi; Maghrebi; and Levantine. Also, the selection process involved users with different interests such as religion, culture, politics, sport, and general. The selection process involved both genders to perform gender categorisation. To verify the gender selection, the corpus contained tweets from verified accounts from known people. The entire selection process was based on user information such as account location, profile picture, bio information, and Twitter verification star.

The corpus are collected using the Tweepy library (Tweepy, 2009). Tweepy is a Python package that interacts with the Twitter API for collecting data. Also, certain hashtags were crawled separately and added to the training set afterward. The reason for this was to expand the size for the Iraqi, Levantine, and Maghrebi dialects. Table 3.4 shows the list of hashtags used.

3.4.1.1. Training set

A total of 101 users were selected in the corpus – 20 users for each main dialect, and 21 users for the Maghrebi dialect, as we found less tweets collected for this dialect. Over two thousand tweets were collected from each user. The reason for this was to balance the selection process. The collection process did not collect tweets from more users as it was intended to annotate the data manually. The selection process involved both genders. Out of the 101 users collected, 40 users were females, and 61 users

were males. The selection process could not balance between the two genders as it was very hard to find females interested in some genres such as religion and politics.

In order to create the training dataset, 2K tweets were collected from each account. The tweets were collected between December 2016 and January 2017. The selected user's account is listed in Table 3.3. In order to select the Twitter account, the selection was verified with the following requirements:

- an actual Twitter account was selected (e.g., not a media source or an organisational account);
- the account was publicly accessible;
- the account should belong to a verified user with Twitter blue star verification if possible;
- the account should have at least posted not less than 5K tweets from the time of starting their account;
- the majority of the account should be written text (e.g., not a picture or links);
- the majority of the text should be written in Arabic.

Dial.	General	Sport	Politics	Religion	Culture
Gulf	**@EbMi_** **@MeMeAlaid** @IIYaserSh @AmarapAmmar	**@nasraweia** **@Jawaher_ALsaif** @Meshari_ @I_mohdiary	**@imankais1** **@SameeraAbd** @DrHAKEM @aAltrairi	**@Nawal_Al3eed_** @salman_alodah @MohamadAlarefe @a_alemran	**@alarabeya8** **@Kaw993** @majeedtimraz0 @MS_Holaiby
Levantine	**@3omariReem** **@Marwa_101** @moh_akkad92 @Anasal10	**@RaghoodaSa11** **@lilianetannoury** @HusseinY22 @sabbah_ashraf	**@EHSANFAKEEH** **@sourayaassi1** @YZaatreh @Omar_Madaniah	@moalhasan @DrZaineddin @shmhd1 @1977Lababidi	**@loubabah** **@nardeen_abunaba** @Zahiwehbe @AymanOtoom
Egyptian	**@ablaFahita** **@sasosall** @mohamedelshrafa @loaiomran	**@ManarSarhan** **@aya_elmshnb** @IbrahimsaidAdam @Nsoo7y	**@Nadiaglory** **@AzzaElGarf** @MohamedAbuHamed @ashrafrady	@amrkhaled @DrAliGomaa @alqaradawy @raghebelsergany	**@RadwaAboAlam** **@FatimaNaoot** @sultanhaggar @alikhiry000
Maghrebi	**@Unknowngirl1990** **@ferferdaous** @BiiiGGGG @imedbhri	@BadiBenJemaa @derradjihafid @DhiabTarak @Chaouali1970 @Benayadachraf	**@Ania27El** **@hassinaouch** @anwarmalek @mohamedzitout	@hassan_kettani @SOHAIB_SOUNI @Abou3issa_Lotfi @salafisenna	**@amira27277** **@nouaramechta** @Abou_Yaareb @WacinyAlAraj
Iraqi	**@KaramAlhafidh** **@Ola_Zngna** **@rtto98901** @HaydarAlmudafar	**@Qi3iQ** **@sarrrrh2044** @EyEyad @Rawanalnahi	**@NerminGaga** @zyaad_alsenjary @alhashimi_Tariq @akklaph	@alduferi1969 @aliqaradaghi @HadiAlModarresi @zaman_alhasnawi	**@DIJLA85** **@alia_jaber97** @a_k_omari @ZaidHamdany

Table 3.3 All the users selected for BTAC. Text in bold font refers to the female users.

3.4.1.2. Hashtags

The aim of the hashtag collection was to add more tweets to certain dialects. These hashtags are considered part of the training set. It was noticed that some dialects contained less tweets than the others. Therefore, further tweets were collected using Iraqi, Levantine and Maghrebi hashtags to expand the corpus for these dialects. These hashtags were chosen according to the geolocation from people speaking those dialects. Table 3.4 below lists the hashtags used for the corpus.

Dialects	Hashtags
Iraqi	#شعر_عراقي #غزل_عراقي #اشعار_ #يوميات_مگرود #دارميات #شعر_شعبي_عراقي
Levantine	#لو_بتحبني #انك_لبناني_يعني #النقل_المشترك #بلد_الظلام
Maghrebi	#محرز_في_ليفربول #غرد_بالداريجه #غرد_بالامازيغيه #تونس_المزيانة

Table 3.4 List of Hashtags used in BTAC.

3.4.1.3. Testing set

The study attempted to collect multiple test sets over different periods of time. This is designed to represent the exact nature of the twitter streaming feed. The intention was to designate specific training and testing splits rather than use a cross-fold validation process. This is because designating specific splits is more representative of the categorisation task in this case due to the dynamic streaming nature of twitter data which changes over time. This will mean that other researchers will be able to directly compare future experimental results avoiding possible inconsistent processing of the data by explicitly designating the specific training and testing splits used herein in order to aid future research.

The testing set was collected from the same selected users in Table 3.3 at three different time periods in March, April, and July, 2017. This was done to avoid any possible overlap between the testing and training sets. The testing sets were processed the same way as the training set was processed. However, this time only the top 50 tweets were collected from the same usernames in the training set employed in this research. The number of tweets including training and testing data before and after processing is shown in Table 3.5.

File	# tweets before processing	# tweets after processing
Training	218,835	115,985
Testing	15,000	6,890
Combined	233,835	122,875

Table 3.5 Size of the corpus before and after processing.

3.4.2. Processing steps

In order to clean the text, the following processing steps were applied to all the tweets. A sample tweet before and after processing is shown in Table 3.6.

- Retweeted tweets were removed. This was to ensure that the tweets were collected for a specific username and were not tweeted by another person.
- HTTP links, usernames, images, and non-Arabic tweets were also removed as the focus was only on Arabic text, and also to ensure that tweets did not contain spam and other non-relevant data that would not help when performing classification.
- Hashtags, emojis, stop words and special characters such as underscores, and quotes were retained. This information was retained to provide the text in its original form and might aid identification when classification experiments are performed in the future.

Label	Tweet
Before processing	@RyBookFair: أوقات زيارة #معرض_الرياض_الدولي_للكتاب https://t.co/A2lWzgBtq7
After processing	أوقات زيارة #معرض_الرياض_الدولي_للكتاب

Table 3.6 A sample tweet before and after processing.

3.4.3. Annotation process of BTAC

As stated, the corpus contains five main dialects – Egyptian, Gulf, Iraqi, Maghrebi and Levantine in addition to MSA and CA. Two Arabic native speakers (postgraduates with experience in NLP) independently annotated the corpus manually. Details concerning

the two annotators are shown in Table 3.7. The first annotator was the author of this thesis. The second annotator, Osama Alruwaili, is an expert in linguistics studies. This collaboration produced the paper mentioned at the beginning of the chapter. To ease the annotation process for both annotators, a website was created to enable the annotation process.

File	Annotator 1	Annotator 2
Name	Mohammed Altamimi	Osama Alruwaili
Qualification	PhD Student in Computer Sciences	PhD Student in Linguistics Studies
Time spent	2 months full time	4 months part time

Table 3.7 Qualifications of the two annotators.

3.4.3.1. Annotation tool for the BTAC

A website was created to help the annotation process using the Google site service. The service allows for the creation of high-quality sites for teams, projects or events. The main page in the website contained links for each author tweets, whereas author timeline tweets was uploaded using a Google Excel spreadsheet. The advantage of using the Google spreadsheet was to provide synchronization between the two annotators, and accessibility from any device used by the annotators.

Each spreadsheet contains author timeline tweets. For each tweet, a dropdown list was attached for each line containing three columns: main dialect; second dialect (in case of code-switching excites); and tweet genre. For dialect annotation, eight labels were included within each dropdown list containing Egyptian, Gulf, Iraqi, Maghrebi, Levantine, MSA, CA, and Unknown. In addition to dialect annotation, the corpus was annotated by genre. A total of 12 genre were included – Social, Economy, Greetings, Cultural, Religious, Sport, Tourism, Political, Art, Conversation, Information, and Unknown. These genres were chosen according to other studies who used similar topics (El-Haj *et al.*, 2010; El-Haj and Koulali, 2013). For example, In the second row of the spreadsheet in the Figure 3.1, the tweet " أشكرك لأنك التقطت الفكرة التي حاولت طرحها وغابت عن البعض ورفضها." translated as "Thank you for picking up the idea that I tried to explain

and rejected it by the other." has been labelled as general statements for its genre, and labelled as modern standard Arabic for the type of dialect used. More sample tweets are shown in Table 3.8.

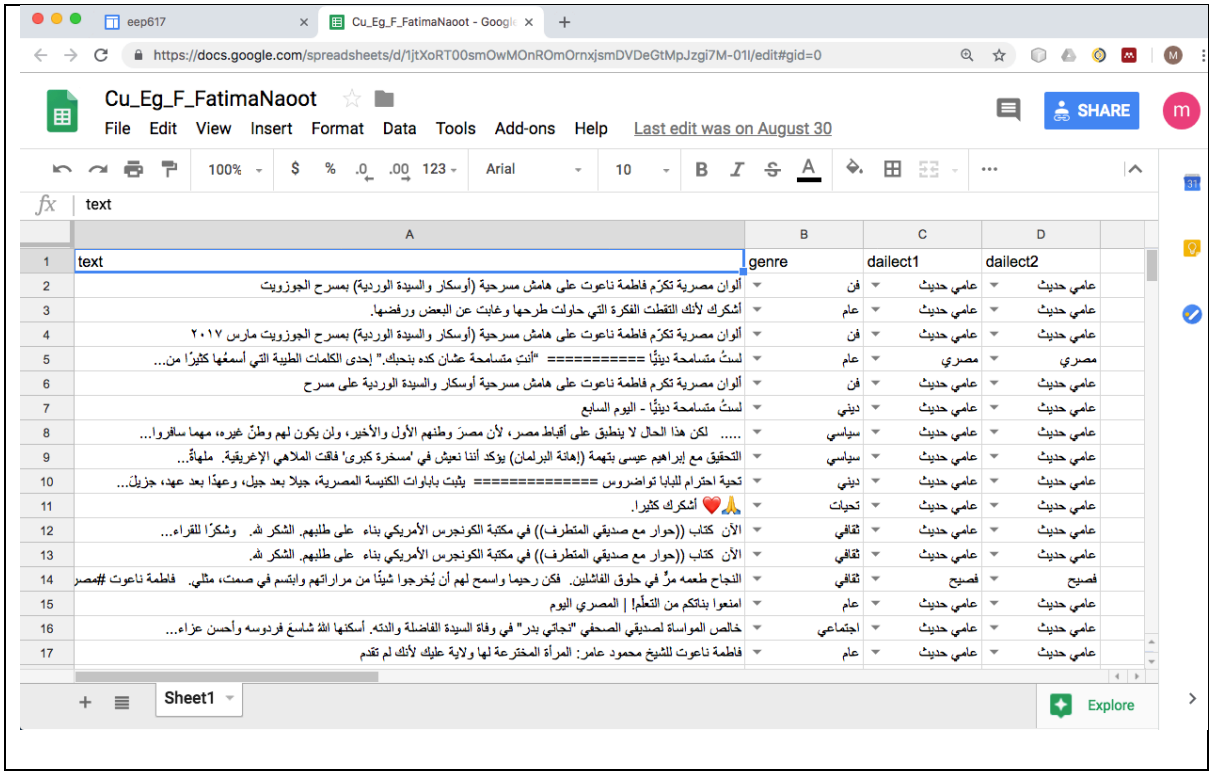

Figure 3.1 Copy of the Google spreadsheet used to annotate the corpus.

3.4.3.2. Annotation labelling

The goal of the annotation was to identify whether the tweet is written using one of the dialects, MSA or CA. Tweets that could not be assigned to one of the dialects were marked as unknown so that they could be identified and excluded from both the training and testing sets if needed. Moreover, code-switching was also identified for tweets that were written in mixed dialects (for example, a tweet that is written in the Egyptian dialect followed by MSA text or CA).

In order to assure the annotators followed the same annotation steps, the following annotation labels was used:

- Dialect: This was for tweets that are written in one of the dialects; these should be annotated under the name of the dialect.
- Classical Arabic: This included tweets that contain old writing styles such as the Qu'ran, Hadith, Prayers, or Poetry.
- Modern Standard Arabic. This was for tweets written in a modern style of Arabic such as found in newspapers, magazines or TV programmes.
- Unknown: This was for tweets that used an unknown set of multiple dialects or tweets that were not meaningful or contained only symbols such as emojis or undetermined text.
- Mixed: This was for tweets that contained two dialects. The name of the second dialect is mentioned in this case.

3.4.3.3. Data availability

The corpus is presented in three different formats: TXT for easy processing; XML for data manipulation; and CSV for supporting tables. The entire corpus is available to download freely[1]. The main corpus includes three main folders: train; test; and code-switching. Each tweet was labelled in all three folders with various labels such as main dialects, second dialects, genre, gender, and authorship. Code-switching content is excluded and stored in a different folder.

[1] *https://github.com/Maltamimi01/BTAC*

Tweet	Main Dialect	Second Dialect
وفي ناس بتهرب منك وقت الجد وناس بتشوف فيها كل حد English: There are people evading in some serious situations and there are people around	Egyptian	
جيبوه بلكي يصعدنا لكاس العالم... English: Bring him it might lead us to the World Cup	Lev.	
أهالي سوسة يحتفلون بالجزائريين بتظاهرة جميلة جدا عرفنانا منهم بإنقاذهم الموسم السياحي جو مزيان بين الشعبين ما حلاها سوسة وناسها #احنا_واحد English: The people of Sousse celebrated the Algerians in a very beautiful demonstration in recognition for saving the tourist season a lovely environment between the two peoples #we_are_together	MSA	Maghrebi
لا نقول الا الحمد لله على كل حال والعوض عألله لكن اي شخص بيقول عنهن شعب طيب رح اسمعو كلام يسم بدنو English: We do not say, but praise is to Allah. In any case, the reward is for him, but anyone says they are good people, I will poison him	CA	Lev.

Table 3.8 Some sample tweets from the BTAC and their English translations. Text in red colour shows code-switching content which will be examined in chapter 7.

3.5. BTAC analysis

The tweets are analysed after manually annotating the tweets according to both dialect and genre. The results from this analysis are presented in this section.

3.5.1. Dialect tweet analysis

The tweets were distributed unevenly among the dialects. Out of the 122K tweets collected, it was found that the majority of the tweets were written in Modern Standard Arabic and Classical Arabic styles. Most of the tweets that were written in Modern Standard Arabic style (37%) with political influence due to the current situation in the Middle East (Ritzen, 2019). However, tweets written in Classical Arabic (27%) were religious, historical, and cultural tweets. The rest of the tweets were written in dialectal form ordered by the percentage of tweets as follows: Gulf (8%); Egyptian (8%); Levantine (7%); Maghrebi (3%); and Iraqi (2%). The Gulf and Egyptian dialects are highly used in social media especially on Twitter. The Maghrebi, Levantine and Iraqi dialects accounted for the lowest number of tweets out of those collected. This is due to Twitter not being very popular in those countries compared to Facebook, which is

mostly used in these countries (Salem, 2017). In addition, countries such as Morocco, Tunisia and Algeria also use other languages to communicate such as French (Harrat *et al.*, 2015).

Dialect	Number of tweets	Dialect	Number of tweets
MSA	46,053	Mixed	8,492
Classic	32,490	Levantine	8,221
Gulf	9,769	Maghrebi	4,147
Egyptian	9,624	Iraqi	1,970

Table 3.9 Breakdown of the annotated tweets.

Undetermined tweets were labelled as Mixed. Those tweets contained emojis, numbers and undecided dialectal text. Lastly, code-switching occurs in 1% of the entire corpus. Most of these tweets are mixed with either CA or MSA. Only three tweets were found to have a mixture of dialects as it was found to be extremely rare to have people using multiple Arabic dialects in the same tweet.

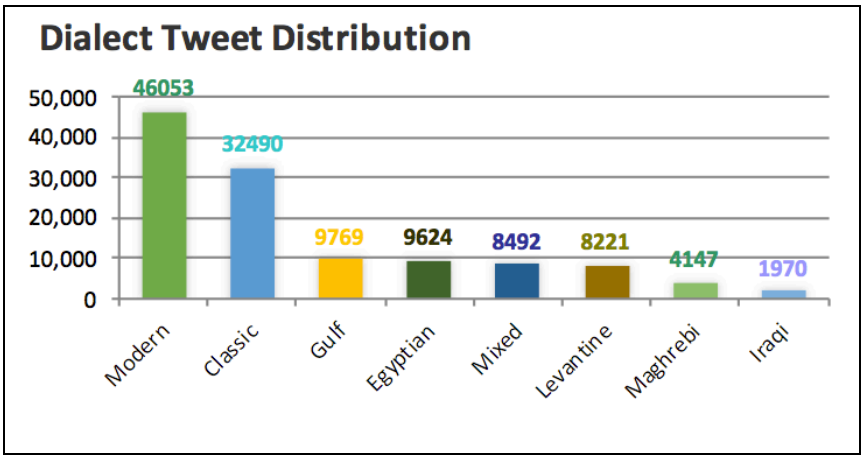

Figure 3.2 Dialect percentage of the annotated tweets.

3.5.1.1. Unigram distribution

Unigram samples taken from the tweets show the obvious distinctions between all of the Arabic dialects. Table 3.10 shows the top 20 unigrams for BTAC. This table is generated by removing the stopwords using the list produced by Alrefaie (2016). Buckwalter transliteration is provided in the first column for each dialect to illustrate the differences between the subset in Latin script.

	Classic		Modern		Gulf		Egyptian		Levantine		Maghrebi		Iraqi	
Top 20 unigrams	Allhm	اللهم	AlErAq	العراق	Ally	اللي	m$	مش	Em	عم	twns	تونس	mw	مو
	Al$yx	الشيخ	mSr	مصر	w$	وش	Ally	اللي	$w	شو	Ally	اللي	dArmy	دارمي
	AlnAs	الناس	swryA	سوريا	$y	شي	dh	ده	hyk	هيك	rby	ربي	byh	بيه
	Abn	ابن	AlmwSl	الموصل	mw	مو	dy	دي	Ally	اللي	bA$	باش	hAy	هاي
	AlHyAp	الحياة	lyrAn	إيران	ybArk	يبارك	Ayh	ايه	m$	مش	m$	مش	Any	اني
	AlIslAm	الإسلام	AlEAlm	العالم	tslm	تسلم	mSr	مصر	$y	شي	br$A	برشا	ErAqy	عراقي
	mHmd	محمد	Hlb	حلب	E$An	عشان	E$An	عشان	Anw	انو	$y	شي	rwHy	روحي
	$Ahd	شاهد	AlErbyp	العربية	AlnSr	النصر	AlzmAlk	الزمالك	knt	كنت	ly	لي	$nw	شنو
	AlImAm	الإمام	dAE$	داعش	bEmrk	بعمرك	kdh	كده	rH	رح	$kwn	شكون	ly$	ليش
	AlslAm	السلام	AlsEwdyp	السعودية	yArb	يارب	rbnA	ربن	HdA	حدا	ky	كي	$lwn	شلون
	$y'	شيء	AlErb	العرب	fyk	فيك	HDrtk	حضرتك	ly$	ليش	mHrz	محرز	hyj	هيج
	rHmh	رحمه	trkyA	تركيا	yqwl	يقول	Hd	حد	bdy	بدي	ElA$	علاش	AlnAs	الناس
	wmA	وما	rwsyA	روسيا	xlAS	خلاص	lyh	ليه	kmAn	كمان	$A'	شاء	nAs	ناس
	llh	لله	Al$Eb	الشعب	Tyb	طيب	dA	دا	mw	مو	HAjp	حاجة	byk	بيك
	rmDAn	رمضان	mbArAp	مباراة	wAnA	وانا	HAjh	حاجه	AzA	ازا	mw$	موش	$y	شي
	Ohl	أهل	AlmdY	المدى	xyr	خير	bAllh	بالله	ktyr	كتير	AljzA}r	الجزائر	Akw	اكو
	Alnby	النبي	trAmb	ترامب	ly	لي	zy	زي	mtl	متل	Anty	انتي	wAnt	وانت
	flA	فلا	mHmd	محمد	yAllh	ياالله	knt	كنت	Tyb	طيب	mbrwk	مبروك	AlErAq	العراق
	tEAlY	تعالى	AyrAn	ايران	knt	كنت	Aqsm	اقسم	hAd	هاد	flAwn	فلاون	byhA	بيها
	Alqr|n	القرآن	mdynp	مدينة	Any	اني	mAt$	ماتش	lAzm	لازم	wA$	واش	mnk	منك

Table 3.10 Top 20 unigrams for BTAC.

3.5.2. Genre tweet analysis

On the other hand, the annotation of genre was somewhat challenging due to the various texts that were collected. The highest number of tweets were those that contain conversations (29%). It was hard to determine the genre of these tweets as they contain responses or chat text. Political tweets were next with (21%); this was expected due to the current situation in the Middle East. Religious tweets followed, at 16%; these tweets contained some Qu'ran verses, prophet supplications, prayers, and a few explanations of religious problems. Sport accounted for (15%) of the tweets; these contained discussion about various European and local football matches and

sports news. In this section, it was noticed that some tweets contained strong language and hate language, specifically during rival matches. Cultural tweets accounted for (12%) of the total. These tweets were quoted from books, poems and speeches. A small percentage (3%) of tweets contained various types of greetings. Social, information, and unknown tweets accounted for 1% of the tweets. Finally, genres such as travel, health, media, art and economy formed less then 1% of the tweets.

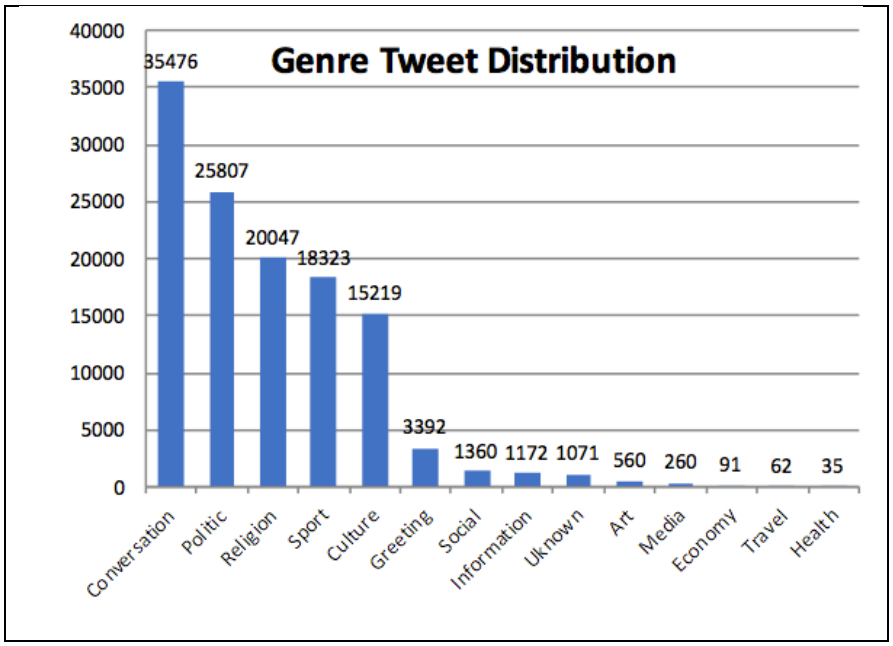

Figure 3.3 Genre percentage for the BTAC corpus after the annotation.

Genre	# Tweets	Percentage	Genre	# Tweets	Percentage
Conversations	35,476	29.00%	Information	1,172	1.00%
Politics	25,807	21.00%	Unknown	1,071	1.00%
Religion	20,047	16.00%	Art	560	0.50%
Sport	18,323	15.00%	Media	260	0.20%
Culture	15,219	12.00%	Economics	91	0.17%
Greeting	3,392	3.00%	Travel	62	0.08%
Social News	1,360	1.00%	Health	35	0.05%

Table 3.11 Breakdown of genre tweets after the annotation.

3.5.3. Challenges during annotation

As Arabic speakers, the annotators were able to understand the text as they were readily identifiable as belonging to a specific dialect, but it was noted that it was difficult to understand the meaning of some of the tweets. The annotation task was not as easy as expected; this was due to the time-consuming and challenging nature of the task. This corpus aimed to be manually checked to study the variations in the language and to rely on an accurately annotated corpus produced by known annotators. Both annotators kept track of the difficulties faced during the annotation process. They collected all the challenging tweets and excluded them from each dialect and labelled them as unknown. The list below summarises the challenges faced when annotating:

- Short tweets: tweets that are less than three words. This kind of tweet was difficult to annotate as it is pretty difficult to identify the exact dialect that it belongs to.
    - Example: للضحك فقط😂😂😂😂
    - Translation: Just for laughing😂😂😂😂
- Acronyms: It was difficult to understand some of the tweets that contained acronyms. Those tweets were kept under unknown category.

  Example*:* اي اي اي🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣

  Translation: AY AY AY 🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣
- Tweets that used unknown dialects: Some of the tweets were considered dialectal but it was very hard to label them to a specific dialect. This kind of tweet was kept under the Mixed category.

  Example: والله إنتو الفخر كله

  Translation: You are full of pride

  Example*:* وعيدك مبارك سيدي

  Translation: Eid Mubarak sir

  Dialects Gulf, Levantine, and Iraqi

- Tweets using local dialects: These tweets existed in some variation of the Maghrebi dialects. Tweets were written using a local dialect such as *Amazigh* (a Berber language) which Arabic native speakers consider hard to understand. Example: 🎙Lip challenge سمشنحمثاس زيث إثمشيشث ما ؤرسمشحمثتارا سمشحمثاست #غرد_بالامازيغيه
- Sensitive tweets: Some of the tweets were found to contain offensive language. These tweets were encountered mostly in the sport and political topics. These tweets were removed as they are not suitable for public consumption.

3.6. Corpus evaluation

This section describes the evaluation of the corpus by applying various experimental analysis: First, the annotation quality was evaluated by applying a inter-annotator agreements (IAA) analysis. Second, cross-corpus classification experiments were used to compare the classification result of this corpus with other existing corpora. This helped to verify the dialectal subset of this research by comparing it against existing dialectal corpora. And finally, an N-grams evaluation was applied with the aim to compare the classical and modern standard Arabic subsets of the corpus by verifying it against other existing classical and Modern Arabic corpora.

3.6.1. Annotation evaluation

In order to evaluate the quality of the annotation, the Kappa coefficient, κ (Cohen, 1960) for measuring inter-annotator agreements (IAA) between the two annotators was used, as follows:

$$\kappa = \frac{P_{\mathrm{O}} - P_e}{1 - P_e},$$

where P_{O} is the actual agreement between annotators, and P_e is the expected agreement between annotators obtained if they randomly assign tags while annotating. $\mathrm{P_e}$ is calculated as:

$$P_e = \sum_q \frac{n_{A1q}}{i} \times \frac{n_{A2q}}{i} = \frac{1}{i^2} \sum_q n_{A1q} \times n_{A2q},$$

where n_{Axq} is the number of tweets to which annotator Ax assigned tag q. i is the total number of annotated tweets.

The Kappa coefficient was measured for the total of 122K tweets that were annotated by the two annotators who took part in this research. The obtained Kappa value was 0.864 for all the MSA, CA and Dialects tweets as shown in Table 3.12. The obtained Kappa result shows substantial agreement according to the measurement of observer agreement for categorical data suggested by Landis and Koch (1977). This reflects the correlation agreement of the two annotators.

After checking the disagreement of the annotated tweets between the two annotators, the cause of the difference was found to be one of the following two reasons:

- More than two dialects could be assigned to the tweets. To solve this disagreement, an annotation label was added that indicated that code-switching had occurred to the other dialect.
- Human error was the reason for disagreement. To overcome this, the two annotators modified the annotation label to reflect the accurate dialect after they reached an agreement.

File	Agreement	Disagreement	Observed Agreement	Kappa
MSA	46,197	6,617	94.6	0.888
CA	33,000	15,323	87.5	0.723
DA	34,098	828	99.3	0.983

Table 3.12 Inter-annotator agreement, disagreement and Kappa values for the BTAC analysis.

3.6.2. Cross-corpus evaluation

The goal in this experiment was to evaluate the corpus by comparing it with another existing corpora that shares similar characteristics. In order to perform this evaluation, corpora that are designed for the same purpose (dialectal Arabic) have to be

compared. The first dataset was selected from the AOC corpus (Zaidan and Callison-Burch, 2011) which contains over 130K annotated sentence labelled in five dialects including MSA consisting of 27K newspaper comments and 105K tweets. The corpus was annotated manually using Amazon's Mechanical Turk service. The second dataset was taken from the AOCD corpus (Mubarak and Darwish, 2014), which contains 139K tweets was selected from the dataset representing the main five dialects. The tweets are annotated according to the geo-location.

In contrast, BTAC corpus contains over 105K tweets including five dialects in addition to MSA and CA. The interesting thing to note is that all corpora differ entirely by the size and annotation method. In each experiment, training is undertaken using one corpus and testing is done against the other two corpora. The experiment employed the Multinomial Naïve Bayes (MNB) algorithm using the WEKA toolkit (Hall *et al.*, 2009).

		Test		
	Corpora	BTAC	AOC	AOCD
Train	BTAC	-	56.0%	44.1%
	AOC	49.6%	-	51.4%
	AOCD	16.1%	34.8%	-

Table 3.13 Cross-corpus evaluation using MNB.

The result shows that BTAC has identified 56% of the AOC corpus and 44% of the AOCD corpus whereas AOC identified 51% of the AOCD corpus and 49% of the BTAC corpus. Lastly, AOCD only identified 16% of the BTAC corpus and 34% of the AOC corpus. This result shows how the BTAC corpus identified both corpora, which indicates that the BTAC is more generalised then the AOC and AOCD corpora despite the differences in size and content. Table 3.13 shows the evaluation results.

3.6.3. N-gram feature-based approach for evaluation

Another method to assist in evaluating the quality of the corpus is to compare it against another existing corpus that shares similar characteristics. The initial way to evaluate

this is to use token analyses (Rayson, 2012), but this method only shows count information. An alternative approach is to examine the corpus by applying the N-grams feature-based approach (Alhawiti, 2014). The corpus evaluation metric is based on the relative entropy calculated using the following formula,

$$M_{C_1,C_2} = \frac{\sum_{i=1}^{n} D_{C_1,C_2}(g_i)}{n},$$

where M is the average code length difference between the two corpora C_1 and C_2. n is the number of N-grams of the same length that occur in the two corpora C_1 and C_2. D is the code length difference for an N-gram defined as the absolute difference between compression codelengths when encoding the N-gram using two different models and calculated as follows:

$$D_{C_1,C_2}(g) = \left|H_{C_1}(g) - H_{C_2}(g)\right| = \left|log_2 P_{C_1}(g) - log_2 P_{C_2}(g)\right|.$$

If the value of M_{C_1,C_2} is equal to zero, then the two corpora would be exactly the same. If the $\mathrm{M}_{\mathrm{C}_1,\mathrm{C}_2}$ value is less than 1, then the corpus being compared has much in common with the reference corpus. If the M_{C_1,C_2} score is over 1, it would specify that the corpus being compared is quite different from the reference corpus. In general, it is impossible to have M_{C_1,C_2} equal to zero if entirely different corpora are being compared, but low codelength values signify that the two corpora being compared have shared characteristics and high values indicate more differences.

In this evaluation, the aim was to compare three popular corpora against BTAC corpus to find out how similar they are. The Holy Qu'ran and Hadith from the King Saud University Classical Arabic (KSUCCA) corpus (Alrabiah *et al.*, 2013) was selected as a representation of classical Arabic text. In addition, corpus A from (Alkahtani and Teahan, 2016) and the TED corpus from (Kulkarni, 2016) was selected as a representation of modern standard Arabic text. *Table* 3.14 shows the result of the comparison.

Average codelength differences (M_{C_1,C_2})	Holy Qu'ran (CA)		Hadith (CA)		Corpus A (MSA)		TED corpus (MSA)	
	BTAC (CA)	BTAC (MSA)	BTAC (CA)	BTAC (MSA)	BTAC (CA)	BTAC (MSA)	BTAC (CA)	BTAC (MSA)
Unigrams	2.291	2.658	2.331	2.414	2.248	2.090	1.922	1.786
Bigrams	2.721	3.183	2.401	2.279	2.008	1.804	1.928	1.716
Trigrams	2.804	3.397	2.589	2.307	2.804	1.741	1.996	1.735

Table 3.14 Codelength differences calculated using formula (1) for BTAC (Classic) and (MSA) when comparing it with: Holy Qu'ran; Hadith; Corpus A; and TED corpus.

Since the Holy Qu'ran is a classical Arabic text, the average code length indicates a strong similarity between this text and the classical Arabic part of BTAC. The average M value of unigrams, bigrams, and trigrams is 2.605 compared to 3.079 for the modern standard Arabic part of BTAC. Recall that, low codelength values signify that the two corpora being compared have shared similar features. Similarly, the Hadith classical corpus which is composed of prophet Muhammad supplication shows an average 2.331 code length for unigrams for the classical Arabic part of BTAC. This is compared to the average 2.414 code length of unigrams for the MSA part.

However, as Corpus A and the TED corpus are modern standard Arabic, the average code length of the Modern standard Arabic part of BTAC is less than the classical Arabic part of BTAC. For instance, in Corpus A, the average code length of unigrams, bigrams, and trigrams for the classical Arabic part of BTAC is 2.353. This is compared to the average 1.878 code length of the Modern standard Arabic part of BTAC. Also, in the TED Corpus, the average code length of unigrams, bigrams, and trigrams for the classical Arabic part of BTAC is 1.946. This is compared to the average 1.745 code length of the Modern standard Arabic part of BTAC.

To summarise, as Corpus A and the TED corpus are modern corpora of Arabic text, the average codelength of BTAC (MSA) is less then BTAC (Classic) for unigrams, bigrams and trigrams, thus indicating similarities between these corpora. Unlike the Holy Qu'ran and Hadith, the average codelength difference of BTAC (Classic) is less than for BTAC (MSA), which indicates noticeable similarities between these corpora for unigrams, bigrams and trigrams.

3.7. Conclusion

This chapter explored the creation of BTAC which contains texts from social media as a reference for Arabic dialects. The collection process involved over 122K tweets manually annotated according to the main five Arabic dialects – Egyptian, Gulf, Iraqi, Maghrebi, and Levantine, in addition to the two main Arabic styles – Classical Arabic and Modern Standard Arabic. Also, the annotation process involved annotation of the tweets according to genre such as information, politics, religion, art, sport, media, culture, economics, greeting, travel, social, and health.

This corpus represents a valuable and rich resource for NLP applications targeting Arabic dialects. The annotation also highlights some tweets that contained code-switching. The evaluation of the annotators' performance is shows substantial agreement according to the measurement of observer agreement for categorical data. Cross-corpus evaluation with other existing corpora was performed and the results show that BTAC is more generalised despite the differences in size and content. Furthermore, the study invstigated the corpus using an N-grams evaluation, which indicates noticeable similarities between the classical and MSA datasets of BTAC corpus with other popular classical and MSA corpora.

Chapter 4

Authorship Attribution of Arabic Tweets Using PPM

4.1. Introduction

Authorship attribution addresses the problem of determining the author of an anonymous text from a set of nominated authors based only on the internal features of the text. It is considered a text classification problem where each author represents a class (Koppel *et al.*, 2009). Authorship attribution has achieved some degree of success in many applications concerning well-known historical text such as the Federalist Papers (Baayen *et al.*, 1996; Khmelev and Tweedie, 2001; Oakes, 2004), 15th book of OZ (Binongo, 2003), and The Pickett letters (Holmes *et al.*, 2001). However, modern research of authorship attribution involves many varied applications such as detecting plagiarism (Chaski, 2005) and identifying writers of messages involving harassment (Abbasi and Chen, 2005b). This research has usually involved formal text from various publications such as articles, essays and emails. However, very limited studies to validate the different techniques have been undertaken specifically for social networking messages such as Twitter. Such research would be relevant in a broad range of applications including computer forensics, criminal and civil law, and cybercrime investigation.

One reason to study authorship attribution for social media such as Twitter is to reveal the true persona of the source of the text among potential suspects. In the age of online surveillance, it is important to identify hidden name accounts by exposing the real person behind an account. The Arab Social Media Report series (Salem, 2017) has reported that, based on a regional survey concerning false information that people usually provide on social media, around 61% of those respondents said that they provide a false name. Those accounts are used for several reasons including online

sexual harassment, illegal trading, cyberstalking, extreme religious, and extreme feminism.

Therefore, this study aims to examine the effect of prediction by partial matching (PPM) in recognising authorship text from social media. The key goal is to adopt a character-based approach to capture the “fingerprints” of the author of the text on social media. In addition, the results are compared with traditional machine learning algorithms and various features such as characters, and words are employed to aid in authorship attribution.

Part of this work in this chapter has been published in the International Journal of Computer Science and Information Technology (IJCSIT):

> *Altamimi, M., and Teahan, W.J., "Gender and Authorship Categorisation of Arabic Text from Twitter using PPM." International Journal of Computational Science and Information Technology (IJCSIT) 9(2) (2017): 131-140.*

The chapter is structured as follows. Section 4.2 describes the experimental setup and dataset breakdown; section 4.3 reports the results of single tweet authorship attribution, with a focus on results using various word and character (N-grams) features; section 4.4 reports the results of multiple tweets authorship attribution and discuss the mis-classified authors results; section 4.5 discusses the findings of the study; and section 4.6 concludes the chapter.

4.2. Experimental Setup

In this experiment, two different datasets were examined – single tweet and multiple-tweets authorship attribution – single tweet categorisation classifies each tweet from each author separately, whereas multiple tweet categorisation classifies several tweets from each author. given a set of candidate authors from Twitter composing relatively short text. Is it possible to identify a set of sample writings for each one of the candidate authors? Figure 4.1 shows the experimental design for both approaches. The training phase uses the concatenation of training text available from each class (i.e. Author 1 and Author 2 and so forth) using Protocol I (see section 2.11).

The testing phase remains as discussed earlier examining single tweet and multiple-tweets authorship attribution.

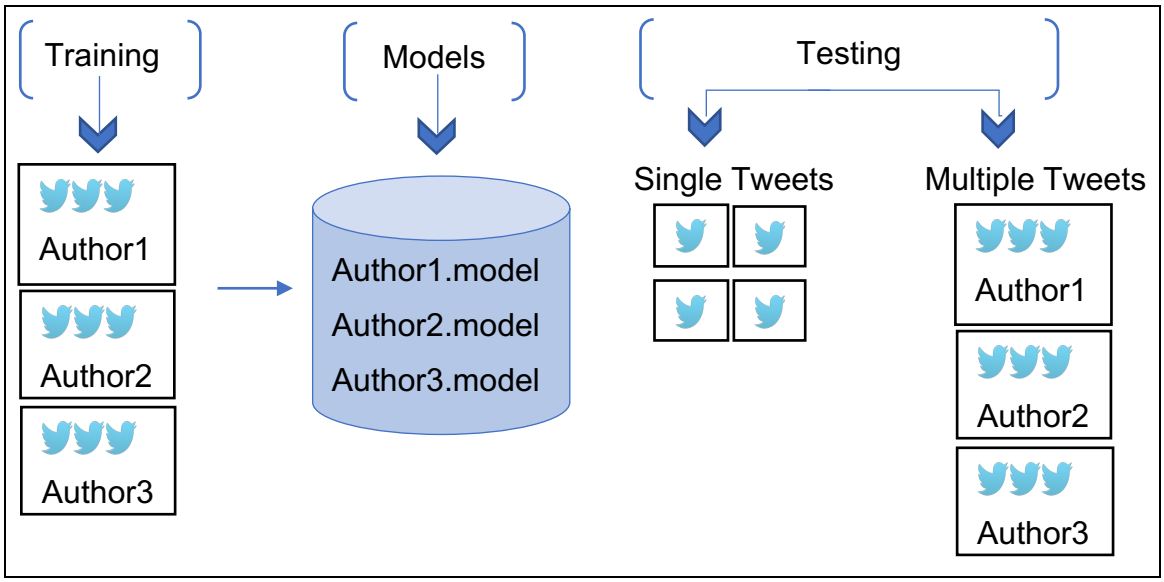

Figure 4.1 Experimental design for authorship attribution experiments.

4.2.1. Dataset

The dataset for this research contains 101 users to perform authorship attribution experiments on, with each user representing a set of sample writings for each one of the candidate authors. The authors represent different dialects from the Middle East (Gulf, Egyptian, Levantine, Maghrebi, Iraqi). The intention is to identify the authors using different dialects and styles of writing. Also, to add variation to the sample of author, the authors' selection includes authors with different interests such as religion, culture, politics, sport, and general. The total number of tweets obtained for each author is listed in Table 4.1 below. The training set composed of 2000 tweets was collected from each author back in January 2017. The testing set composed of 50 tweets was collected form the same authors in three different time periods (March, April, and July 2017). The table shows that the total number of training and testing tweets vary for each author. This is because several processing steps were performed to make sure that tweets follow the same standard in each of the experiments performed in this thesis. [Section 3.4.1 lists all the processing steps performed]

#	Authors	#Training	#Testing	#	Authors	#Training	#Testing
1	FatimaNaoot	679	47	52	akklaph	1536	56
2	RadwaAboAlam	989	63	53	alhashimi_Tariq	1904	122
3	alikhiry	886	72	54	zyaad_alsenjary	1627	110
4	sultanhaggar	1635	101	55	EHSANFAKEEH	754	71
5	AzzaElGarf	806	49	56	sourayaassi	1483	98
6	ablaFahita	1464	21	57	Omar_Madaniah	1749	116
7	sasosall	845	67	58	YZaatreh	1532	102
8	loaiomran	568	22	59	AniaEl	1680	106
9	mohamedelshrafa	275	1	60	hassinaouch	576	102
10	Nadiaglory	563	35	61	anwarmalek	553	41
11	mohamdAbuHamed	1396	94	62	mohamedzitout	1814	111
12	ashrafrady	1064	65	63	loubabah	1143	58
13	DrAliGomaa	1752	68	64	alia_jaber	125	7
14	alqaradawy	1823	119	65	AymanOtoom	1495	93
15	amrkhaled	1345	96	66	Zahiwehbe	827	45
16	raghebelsergany	908	44	67	Nawal_Aleed_	971	78
17	ManarSarhan	1007	64	68	MohamadAlarefe	1203	88
18	aya_elmshnb	1566	103	69	a_alemran	1528	80
19	IbrahimsaidAdam	1616	116	70	salman_alodah	1556	109
20	Nsooy	1288	88	71	HadiAlModarresi	1455	88
21	alarabeya	556	34	72	alduferi	1027	56
22	MS_Holaiby	930	63	73	aliqaradaghi	1115	57
23	majeedtimraz	922	20	74	zamanalhasnawi	1407	85
24	DIJLA	1606	111	75	moalhasan	1039	38
25	EbMi_	1857	114	76	DrZaineddin	1229	59
26	MeMeAlaid	1334	91	77	Lababidi	491	71

27	AmarapAmmar	798	39	78	shmhd	1041	29
28	IIYaserSh	1511	100	79	Abouissa_Lotfi	632	9
29	KaramAlhafidh	321	39	80	SOHAIB_SOUNI	699	44
30	Ola_Zngna	212	11	81	hassan_kettani	1101	98
31	rtto	260	3	82	salafisenna	1036	84
32	HaydarAlmudafar	350	9	83	amira	782	36
33	Marwa_	1444	91	84	nouaramechta	780	57
34	omariReem	1429	77	85	Abou_Yaareb	1990	121
35	Anasal	1332	60	86	WacinyAlAraj	765	4
36	moh_akkad	1727	114	87	Jawaher_ALsaif	806	81
37	Unknowngirl	1129	21	88	nasraweia	697	17
38	ferferdaous	323	11	89	I_mohdiary	1631	101
39	BiiiGGGG	509	44	90	Meshari_	1403	98
40	imedbhri	1161	73	91	QiiQ	1416	72
41	Kaw	952	63	92	sarrrrh	394	83
42	nardeen_abunaba	1124	63	93	EyEyad	1641	61
43	ZaidHamdany	798	58	94	rawanalnahi	1573	120
44	a_k_omari	1521	83	95	RaghoodaSa	991	7
45	SameeraAbd	1211	52	96	lilianetannoury	1315	58
46	imankais	1099	80	97	HusseinY	1774	112
47	DrHAKEM	1910	105	98	sabbah_ashraf	767	74
48	aAltrairi	1078	75	99	BadiBenJemaa	872	68
49	NerminGaga	1172	20	100	Benayadachraf	1465	98
50	derradjihafid	694	74	101	Chaouali	450	96
51	DhiabTarak	475	82				

Table 4.1 Total number of training and testing tweets used in the BTAC.

4.3. Single tweets authorship attribution

In this experiment, the attribution task for identifying the authorship of single tweets was investigated for 101 authors. Each tweet was split into a single file with a total of 6890 tweets tested. The task was investigated using different orders of PPMD from order 2 to order 13. Overall, the best accuracy is obtained by applying PPMD using order 5 with accuracy 57.2%. This initial result did not match expectations but was anticipated due to a large number of tweets which caused much confusion. Also, classifying single tweets that contain fewer author features made the classification task more difficult. Table 4.2 reports these results of authorship attribution of single Arabic tweets using PPMD.

Orders	**Accuracy**	**Recall**	**Precision**	**F-measure**
Order 2	46.9%	0.447	0.432	0.439
Order 3	52.4%	0.486	0.473	0.479
Order 4	56.1%	0.513	0.509	0.511
Order 5	**57.2%**	0.522	0.524	0.523
Order 6	56.9%	0.519	0.523	0.521
Order 7	57.0%	0.522	0.526	0.524
Order 8	56.9%	0.521	0.523	0.522
Order 9	56.6%	0.518	0.520	0.519
Order 10	56.6%	0.518	0.520	0.519
Order 11	56.6%	0.520	0.519	0.520
Order 12	56.4%	0.515	0.517	0.516
Order 13	56.4%	0.515	0.516	0.515

Table 4.2 Authorship attribution of single Arabic tweets using PPMD.

In order to compare these results with other classifiers using WEKA (Hall *et al.*, 2009). Machine learning algorithms were applied using the same testing set used for PPM. We used *string-to-word-vector* filter to build our vector list. We did not undertake further pre-processing of the data such as stemming, tokenisation or removal of stop

word. however, only the top 100 most frequently used words from each author. The default setting in WEKA retains up to 1000 words from each class. However, processing of these words required substantial execution time and was found to be unmanageable for a total of 101 class of authors. Results showed that none of the three classifiers performed well. This was expected due to the lack of contextual information from each author, as the experiment is performed on single tweets for each author.

Test	Accuracy	Recall	Precision	F-measure
MNB	15.2%	0.152	0.329	0.163
LibSVM	14.7%	0.147	0.330	0.158
KNN1	2.46%	0.000	0.004	0.002

Table 4.3 Authorship attribution of single tweets using machine learning algorithms.

However, a study by Luyckx and Daelemans (2011) suggested that the problem of authorship attribution is relatively less complicated when the number of authors is small. Therefore, an experiment with random authors to attribute their single tweets was investigated. However, we wanted to test the performance of having various sets of 5, 10, 15 and 20 candidate authors. The authors were selected from one dialectal region (Egyptian) where the authors use much common context as they are from one region. The authors selected are the top 5, 10, 15 and 20 authors from Table 4.1.

Table 4.4 shows that the result improved when performing with a smaller number of authors, as was expected. The best result was obtained with five authors achieving an accuracy of 77.7%. A total of 332 tweets were tested in this experiment. This represents the scenario when the intention is to identify a text in a forensic investigation among potential suspects. In addition, the accuracy of attributing single tweets of 20 authors achieved an accuracy of 73.04%. This result is better then attributing single tweets of 10 and 15 authors achieving accuracy of 71% and 70%, respectively.

This result was obtained using order 4 of PPMD which shows that authorship is better identified using short sequences of characters. Note that, order 4 of PPMD is

equivalent to two Arabic letters as the compression is performed at the byte level. However, further experiments are required using multiple tweets dataset to confirm this assumption.

#Authors	#Tweets	Accuracy	Recall	Precision	F-measure
5	332	**77.7%**	0.766	0.797	0.781
10	478	71.0%	0.631	0.638	0.634
15	920	70.0%	0.629	0.642	0.635
20	1355	73.0%	0.674	0.677	0.676

Table 4.4 Authorship attribution of single tweets using PPMD performed on 5, 10, 15, 20 authors.

With the same dataset produced by five authors, the experiment was performed on single tweets with the machine learning algorithms. This time, the top 1000 most frequent words were used from each author, as the aim was to expose as many features as possible from each author's text. The results reported in Table 4.5 show that LibSVM generated the best result with an accuracy of 60.8%, slightly better than MNB with an accuracy of 59.3%. KNN did not perform well on this experiment compared to the other classifiers, achieving an accuracy of just 31.3%.

Test	Accuracy	Recall	Precision	F-measure
MNB	59.3%	0.593	0.626	0.590
LibSVM	60.8%	0.608	0.604	0.605
KNN1	31.3%	0.313	0.114	0.166

Table 4.5 Authorship attribution of single Arabic tweets performed on five authors.

As mentioned, one of the aims of this work was to investigate authorship attribution using different feature sets. Using the single tweets dataset produced by the five authors, various word-based (unigram and bigram), and character-based features from one to six characters. We could not add more word n-grams as this would result in low frequency. Overall, Table 4.6 shows that LibSVM achieved the best results with character 6-grams with an accuracy of 63.8%; this was slightly better than MNB which

achieved an accuracy of 62.9% also using character 6-grams. KNN achieved the best accuracy with word unigrams obtaining 31.3%.

This result shows that using machine learning algorithms to identify authorship was found to perform best using a higher number of characters such as 6-grams. However, further experiments using multiple tweets dataset is needed to confirm this assumption.

Classifiers		**MNB**				**LibSVM**				**KNN 1**			
Features		**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**	**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**	**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**
Word	**Unigrams**	59.3	0.59	0.62	0.59	60.8	0.60	0.60	0.60	**31.3**	0.31	0.14	0.16
	Bigrams	48.1	0.48	0.64	0.48	54.8	0.54	0.61	0.54	14.7	0.14	0.02	0.03
Character	**Unigrams**	24.0	0.24	0.73	0.20	22.5	0.22	0.74	0.19	30.0	0.30	0.11	0.16
	Bigrams	14.1	0.14	0.45	0.10	18.6	0.18	0.46	0.12	31.0	0.31	0.22	0.25
	Trigrams	50.6	0.50	0.67	0.44	50.6	0.50	0.66	0.45	9.33	0.09	0.03	0.05
	4-grams	61.4	0.61	0.64	0.57	62.3	0.62	0.62	0.58	14.1	0.14	0.04	0.06
	5-grams	59.6	0.59	0.63	0.59	61.1	0.61	0.63	0.60	19.5	0.19	0.19	0.12
	6-grams	**62.9**	0.63	0.66	0.62	**63.8**	0.63	0.63	0.62	14.1	0.14	0.16	0.04

Table 4.6 Results for single tweets authorship attribution using different word and character features.

However, we performed the experiment using PPMD for various orders 2 to 12. The experiment was performed with 332 single tweets for 5 authors as shown by the best result in table 4.4. The best classification reported on Table 4.7 was when using order 4 and order 6 with an accuracy of 77%. This result shows that PPM outperforms machine learning algorithms using various word and character features.

Orders	Accuracy	Recall	Precision	F-measure
Order 2	76.0%	0.75	0.75	0.75
Order 3	74.0%	0.72	0.74	0.73
Order 4	**77.0%**	0.76	0.79	0.78
Order 5	76.0%	0.74	0.76	0.75
Order 6	**77.0%**	0.75	0.77	0.76
Order 7	75.0%	0.74	0.76	0.75
Order 8	74.0%	0.73	0.75	0.74
Order 9	75.0%	0.74	0.77	0.76
Order 10	76.0%	0.76	0.78	0.77
Order 11	76.0%	0.76	0.78	0.77
Order 12	76.0%	0.75	0.78	0.76
Order 13	76.0%	0.75	0.78	0.76

Table 4.7 Results for single tweets authorship attribution using a different order of PPM.

4.4. Authorship attribution for multiple tweets

Unlike testing on single tweets, in this experiment the aim was to investigate the accuracy when combining multiple tweets from the same author. We applied the experiments to the Twitter data that was described in Chapter 3. The number of authors in Test I, Test II, and Test III are 94, 96, and 95, respectively. Table 4.8 reports the accuracy for various orders: Test I achieved the best accuracy of 93.6% for order 4, Test II achieved the best accuracy of 94.8% for order 2, and Test III achieved the best accuracy of 89.5% for order 3. Overall, the lower orders of PPMD achieved better accuracy. This is due to the dataset containing short sequences of characters, such as syntactic features, that helped identify the authors.

Orders	Test I (March)	Test II (April)	Test III (July)
Order 2	90.4	**94.8**	88.4
Order 3	89.4	92.7	**89.5**
Order 4	**93.6**	88.5	87.4
Order 5	92.6	87.5	88.4
Order 6	90.4	87.5	87.4
Order 7	90.4	87.5	86.3
Order 8	92.6	89.6	87.4
Order 9	90.4	89.6	87.4
Order 10	91.5	89.6	87.4
Order 11	91.5	89.6	88.4
Order 12	91.5	89.6	88.4
Order 13	91.5	89.6	88.4

Table 4.8 Accuracy of authorship attribution in three test sets using PPM.

For authorship attribution using machine learning algorithms, we used *string-to-word-vector* filter to build our vector list. we did not undertake further pre-processing of the data such as stemming, tokenisation or removal of stop word. However, only the top 100 most frequent words were retained from each author for the vector list. Again, it was found that the default setting of 1000 words took a significant amount of time to process for a total of 101 class of authors. It was found that MNB and LibSVM produced a better result than KNN1 as shown in Table 4.9. However, further investigation is required using various word and characters N-gram features.

Tests	**Test I (March)**				**Test II (April)**				**Test III (July)**			
Measure	**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**	**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**	**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**
MNB	**74.4**	0.74	0.67	0.69	**81.2**	0.81	0.75	0.77	**78.9**	0.78	0.71	0.73
SVM	**74.4**	0.74	0.67	0.69	80.2	0.80	0.75	0.76	77.8	0.77	0.69	0.72
KNN1	2.12	0.02	0.01	0.01	12.0	0.12	0.09	0.09	3.15	0.03	0.02	0.02

Table 4.9 Authorship attribution of author tweets using machine learning classifiers for the different test sets.

Also, the effect of adding word and character features were examined to the attribution process. The goal of this was to evaluate which feature gives the best performance in the attribution process. Word N-grams and characters using MNB, LibSVM, and KNN1 were employed to find the best features to use for the machine learning classifiers. However, all the testing sets (Test I, Test II, and Test III) were combined into one testing set. Table 4.10 reports the experimental results. Overall, the best performance was obtained by using unigram word features, achieving the same accuracy of 93% for both MNB and LibSVM classifiers. KNN achieved the best result with word unigrams obtaining 57.4% of accuracy.

This result shows that using machine learning algorithms to identify authorship performs best when using a higher number of characters. However, in this experiment, word unigrams achieved the best result in identifying authorship features. Note that in comparison with character N-grams, word unigrams still rely on a higher number of characters, as the average Arabic word length is 5 characters (Alotaiby *et al.*, 2009)

Classifiers		**MNB**				**LibSVM**				**KNN 1**			
Features		**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**	**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**	**Acc. (%)**	**Rec.**	**Prec.**	**F-Meas.**
Word	**Unigrams**	**93.0**	**0.93**	**0.89**	**0.90**	**93.0**	**0.93**	**0.89**	**0.90**	**57.4**	**0.57**	**0.53**	**0.54**
	Bigrams	86.0	0.84	0.82	0.83	81.0	0.80	0.80	0.80	1.9	0.02	0.01	0.01
Character	**Unigrams**	59.4	0.58	0.55	0.56	57.4	0.57	0.53	0.54	6.9	0.06	0.06	0.06
	Bigrams	36.6	0.36	0.29	0.31	35.6	0.35	0.29	0.31	11.8	0.11	0.08	0.09
	Trigrams	17.8	0.17	0.14	0.14	16.8	0.16	0.12	0.13	32.6	0.32	0.24	0.26
	4-grams	42.5	0.42	0.32	0.34	45.5	0.45	0.35	0.38	45.5	0.45	0.37	0.39
	5-grams	76.2	0.76	0.67	0.70	78.2	0.78	0.69	0.72	49.5	0.49	0.43	0.45
	6-grams	88.1	0.88	0.82	0.84	89.1	0.89	0.84	0.85	42.5	0.42	0.39	0.39

Table 4.10 Results for multiple tweets authorship attribution using different word and character features.

To compare the results obtained from machine learning classifiers, Table 4.11 shows classification using PPM performed using various orders from 2 to13. Overall, the lower orders achieved better accuracy using PPM, with the best result obtained when

using order 2 with an accuracy of 96%. This supports the earlier observation in Table 4.4 and Table 4.8 which shows that authorship is better identified using short character sequences such as syntactic features. This result is the best accuracy obtained for the experiments described in this chapter outperforming the other character and word-based approaches used by the machine learning algorithms.

Orders	Accuracy	Recall	Precision	F-measure
Order 2	**96.0**	**0.96**	**0.94**	**0.96**
Order 3	92.0	0.92	0.89	0.88
Order 4	93.0	0.93	0.89	0.90
Order 5	90.0	0.90	0.85	0.87
Order 6	89.0	0.89	0.84	0.86
Order 7	91.0	0.91	0.88	0.87
Order 8	92.0	0.92	0.89	0.88
Order 9	92.0	0.92	0.89	0.88
Order 10	92.0	0.92	0.89	0.88
Order 11	92.0	0.92	0.89	0.88
Order 12	92.0	0.92	0.89	0.88
Order 13	92.0	0.92	0.89	0.88

Table 4.11 PPMD results for author tweets attributions using different orders

As seen, combining all three testing sets into one set offers better insight for each author as seen in Table 4.10. The results in Table 4.12 show summary of the best experiment results obtained for all of the classifiers. PPMD with order 2 is found to be the best with only four mis-classifications out of the 101 authors. Next, MNB and SVM using word unigram feature achieved the second best result with seven authors being mis-classified.

Classifiers	Incorrect	Accuracy	Recall	Precision	F-measure
MNB- Unigrams	7	93.06	0.931	0.896	0.908
SVM- Unigrams	7	93.06	0.931	0.896	0.908
KNN1- Unigrams	43	57.42	0.574	0.537	0.546
PPMD order 2	4	**96.00**	0.960	0.940	0.950

Table 4.12 Authorship attribution of author tweets for different classifiers.

4.4.1. Studying mis-classified authors

A total of four authors were found to be mis-classified using PPMD. Three were incorrectly classified due to relatively few tweets found in their testing set. For example the first three authors in Table 4.13 have in total less then 10 tweets for each author. This is due to most of their testing set were retweeted content which was removed as pre-processing step for the corpus. This was to ensure that the testing set contained only tweets for each specific author and not to have comprised other tweets written by other authors.

However, the last author *Sohaib_Souni* was found to have less than 300 bytes codelength difference between the actual and incorrect author. This author was mis-classified to an author who had similar interests and dialect (which were religion and the Maghrebi dialect). Table 4.13 below shows the codelength and codelength difference between the mis-classified author and the actual author.

Incorrect author	Testing Tweets	Codelength (bits)	Classified to	Codelength (bits)	Codelength Difference (bits)
RaghoodaSa	7	615.380	SameeraAbd	588.643	26.737
Abouissa_Lotfi	9	2134.416	salman_alodah	2057.448	76.968
mohamedelshrafa	1	507.148	MohamedAbuHamed	450.341	56.807
Sohaib_Souni	44	18961.330	salafisenna	18685.416	275.914

Table 4.13 Codelength and codelength difference between the incorrectly and correctly classified authors.

4.5. Discussion and findings

In this chapter, a number of Arabic authorships attribution experiments has been presented. The main aim was to perform the experiment using Prediction by Partial Matching (PPM) to see how well it would perform in Arabic. Also, the results were compared with other machine learning algorithms using both word- and character-based approaches for the same dataset.

First authorships attribution for 101 authors was performed on a total of 6890 single tweets. This experiment did not achieve good results for PPM with an accuracy of 57.2%, or machine learning algorithms which produced very low accuracy of 15.5% and 14.7% using MNB and LibSVM. This was expected due to the lack of contextual information from each author, as the experiment was performed on single tweets for each author.

Another experiment was performed when testing single tweets of only five authors with a total of 332 tweets. This was preformed to simplify the problem and represents the scenario when investigating text among potential suspects. This experiment achieved better results using PPM with accuracy of 77.7% using order 4, compared to 60.84% and 59.3% of accuracy using LibSVM and MNB. This shows that authorship is better identified using short sequences of characters using order 4 which is equivalent to two Arabic letters.

Lastly, a further experiment was performed to identify single tweets of only five authors with a total of 332 tweets using the machine learning algorithms. The experiment was studied with various word-based features (unigram and bigram), and character-based features from one to six characters; the result shows that character-based features of 6-grams produced the best results, yielding accuracies of 62.9% and 63.8% using MNB and LibSVM classifiers.

Another experiment involved authorship attribution of multiple tweets for 101 authors; this experiment achieved better results with an accuracy of 96% using PPMD order 2. This result is much higher authors than other Arabic attribution studies. For instance,

Abbasi and Chen (2005b) achieved 94.8% accuracy using 100 authors of newspapers articles, Shaker and Corne (2010) reported 87.6% accuracy using 12 books authors, Albadarneh *et al.* (2015) achieved 61.6% accuracy using 20 authors from Twitter, and Rabab'ah *et al.* (2016) reported 68.6% accuracy using 12 authors from Twitter. It is important to note that our experiment involved a much larger number of twitter authors.

Finally, multiple tweets authorship attribution was investigated using word-based (unigram and bigram), and character-based features from one to six characters. This was performed for the machine learning algorithms to compare their results with PPM. However, the best results for machine learning algorithms were obtained by MNB and LibSVM using unigram word features which achieved the same accuracy – 93% – for both classifiers. In contrast, character-based PPM using order 2 produced the best results overall with an accuracy of 96%. This result demonstrates that authorship is better identified using short character sequences, which interestingly corresponds to other work that produced similar findings for the Greek language (Mikros, 2012).

4.6. Conclusion

This chapter has investigated the authorship attribution problem using two main approaches: character-based using PPM and feature-based using various machine learning classifiers, such as KNN, MNB and LibSVM. Overall, both approaches were applied with a focus on the Arabic language which is still largely understudied specifically on short text. The experiments were performed with Twitter posts which represent additional challenges due to the limitation in their sizes. The findings showed that PPM generates superior results compared to the machine learning algorithms. Results were not as good when trying to determine authorship for single tweets. This was expected due to the lack of the contextual information within the short single tweets. However, the second experiment involved attributing authorship when concatenating multiple tweets from the same source. This produced the best result achieving an accuracy of 96%.

Chapter 5

Gender Categorisation of Arabic Tweets Using PPM

5.1. Introduction

The aim of this chapter is to investigate the problem of gender categorisation for the Arabic language. Unlike the traditional gender analysis which involves formal text such as books, articles or email messages, this research investigates the gender categorisation problem for informal text collected from social media. This type of text is considered challenging with many non-standard text variations such as acronyms, emoticons, and misspellings. However, in online communication, people may not expose their true identity for one reason or another by deliberately assigning a different gender or by hiding their gender information for malicious reasons. A well-known case to illustrate the need to study the gender categorisation problem is the suicide of the teenage Megan Meier after exchanging emails with Lori Drew (female) who pretended to be a teenage boy (called Josh). Megan killed herself due to cyberbullying. Josh abruptly ended their friendship, telling Megan that she was cruel (Wikipedia Contributors 2018). This example shows the urgent need for applications to check users' accounts regularly and flag suspicious accounts for further investigation.

Part of this work in this chapter has been published in the International Journal of Computer Science and Information Technology (IJCSIT):

> *Altamimi, M., and Teahan, W.J., "Gender and Authorship Categorisation of Arabic Text from Twitter using PPM." International Journal of Computational Science and Information Technology (IJCSIT) 9(2) (2017): 131-140.*

The chapter is structured as follows. Section 5.2 discusses the aims of this chapter; section 5.3 describes the experimental setup for both single tweet gender categorisation and author gender profiling. Next, sections 5.4 and 5.5 report the

experimental results, section 5.6 discusses an optimisation using a new classification protocol to improve the categorisation results further, section 5.7 discusses the research findings, and lastly, section 5.8 concludes the chapter.

5.2. Research aims for this chapter

There are a number of aims that we want to achieve in this chapter. Examining the effect of PPM in recognising Arabic text using the BTAC; in particular, we aim to look specifically at the problem of gender categorisation using a statistical approach such as PPM. The results generated here are compared with those from other machine learning algorithms. In addition, we aim to employ various features such as characters and words to aid in identifying gender. Finally, we analyse the incorrectly classified tweets and discuss further improvements. These aims can be summarised as follows:

- apply PPM to the gender categorisation problem;
- compare the results with other machine learning algorithms;
- apply various features such as characters and words;
- examine the performance of identifying multiple tweets versus single tweets;
- investigate further improvements.

5.3. Experimental Setup

In this investigation, in order to see how effective PPM is at gender categorisation of Arabic tweets, two datasets were examined: single tweet gender categorisation; and multiple tweet author gender profiling. Single tweet gender categorisation classifies each tweet separately, whereas multiple tweet gender profiling categorises gender based on several tweets from the same author.

Figure 5.1 below shows the experimental design for the study. Each tweet is pre-processed as described in section 3.4.2. Then, the tweets are annotated and split according to the author's gender. After that, two categorisation experiments are applied for this study – single tweet gender categorisation, and multiple tweet author

gender profiling. For each approach, various classifiers have been applied including Prediction by Partial Matching (PPM) and machine learning algorithms for comparison.

For Prediction by Partial Matching (PPM), static PPMD models were created by training on each class of text. WEKA (Hall *et al.*, 2009) was used for the other machine learning algorithms for a selection of well-performing classifiers such as Multinomial Naïve Bayes (MNB), K-Nearest Neighbours (KNN), and an implementation of Support Vector Machines (LIBSVM).

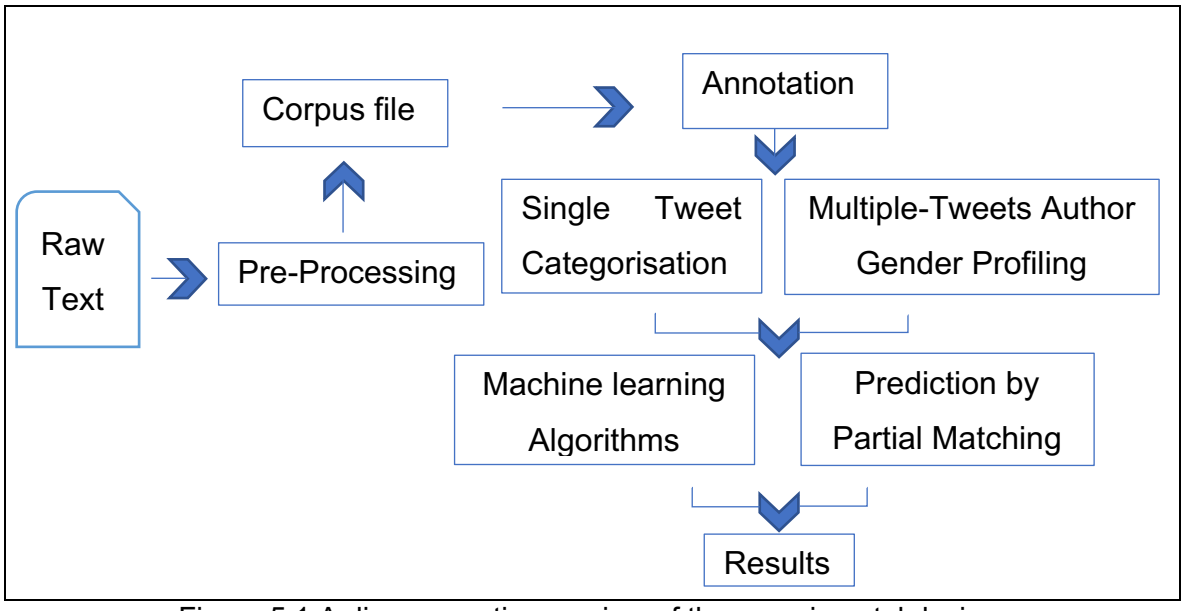

Figure 5.1 A diagrammatic overview of the experimental design.

5.3.1. Collection process

The experiments used the data in the BTAC corpus (Chapter 3) to perform the gender evaluation. The corpus contains 122 thousand tweets written by 101 authors. The corpus is designed to contain mixed gender texts to allow the study of gender categorisation. The selected gender accounts were based on the username, profile picture, and account description. To verify the selection, accounts were chosen that adhered to Twitter policy. Figure 5.2 below shows a screenshot of three different verified profile accounts.

We also checked the account description field for obvious indicators of gender; for instance in Figure 5.2, the description "كاتبة وأديبة مصرية" translated as “Female Egyptian

Writer and Novelist" which indicates that description is written by female since the suffixes "ـة" have been added after each word; whereas a description like "الأمين العام لمؤتمر الأمة", translated as "Secretary-General of National Council", confirms the description is written by a male since no suffixes have been added.

Figure 5.2 Three screenshots for Twitter accounts.

5.3.2. Size of the dataset

The author selection for the corpus (BTAC) involved both genders from various dialects in the Middle East. The reason for this is to acquire gender variation among several dialects. It was decided to use an explicit testing set for testing purposes rather than split the training set, as the intention was to study gender variation within the time frame. The training set was collected for 101 accounts and the test was collected for the same users for three different periods of times in March, April, and July 2017. Table 5.1 below shows the size of the training and testing sets used for the experimental dataset.

Data	Male	Female
Training	73,160	38,900
Test I (March)	954	527
Test II (April)	1809	846
Test III (July)	1928	926

Table 5.1 Total number of tweets collected for gender categorisation experiment.

Gender	Tweet
Male	اتمنالك كل خير يبو طارق وتشوف ثمرة تعبك 😢 فيلم_بلال كان نفسي احضر#
Female	😍 بتعرفي مبارح استفقدتك قلت وينها الها 4 ايام غايبة
Male	نسبة الأخطاء والتركيز وإستغلال الفرص عوامل مؤثره تحدد كثيراً نتيجة الكلاسيكو!
Female	أنا وبلقيس بنحب نفس الچاكتة ونفس الراجل.. #أختي_ومرات_حبيبي

Table 5.2 Sample of tweets representing each gender.

5.3.3. Procedure used for gender categorisation experiment

In this study, after labelling each tweet according to the gender, all the tweets were concatenated into two files (Male/Female) in the training process. Therefore, the outcome of this procedure is two models representing each gender. The testing process remains as discussed in section 5.3 where the aim is to examine the performance of identifying multiple tweets versus single tweets. The following diagram demonstrates the classification procedure. The training phase uses the concatenation of training text available for each class (i.e. Male and Female) using Protocol I (see section 2.11).

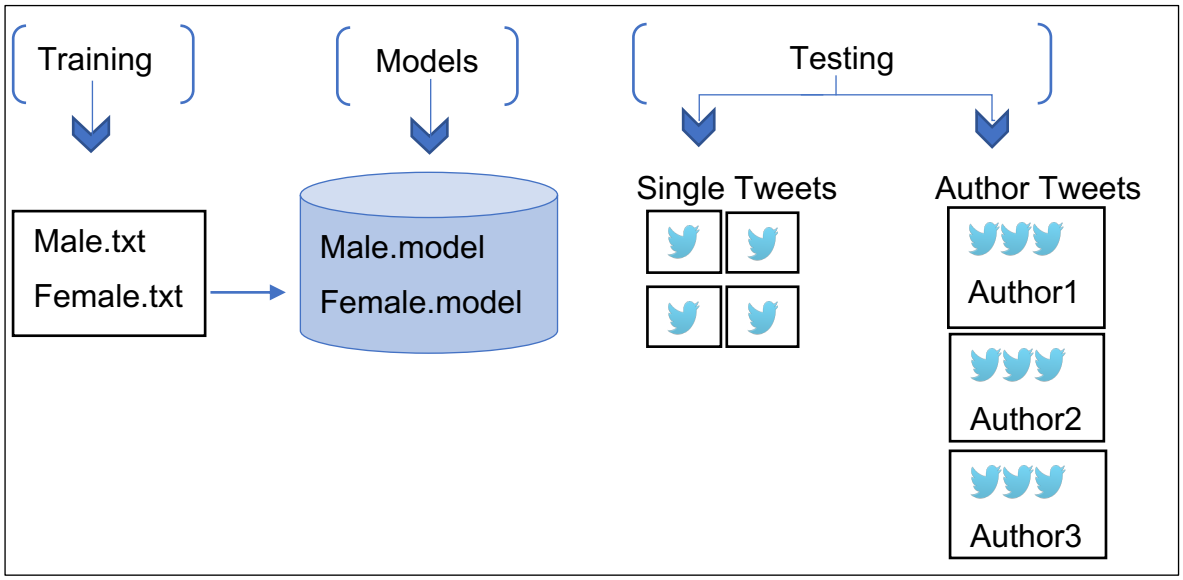

Figure 5.3 Overview of the classification procedure used.

5.4. Single Tweet Gender Categorisation

In this experiment, a binary-classification task for identifying the gender for single tweets was performed – i.e. is this tweet written by a male or female? Each tweet was split into a single file with over 6890 tweets (i.e. files) being tested in three different test sets. Investigation was carried out using different orders of PPMD from order 3 to order 13. We started with order 3 as lower orders such as 2 performed worse. Overall, the best results were obtained by applying PPMD using order 11 with an average accuracy of 76.3%. It has been found that the results increased up to order 12 but then decreased subsequently as shown in Table 5.3. The best result is shown in bold font for each test set. Results for order 11 for Test I, Test II and Test III datasets (see Table 5.2 above) are similar in terms of recall and precision ranging from 0.72 to 0.75.

Orders	Test I (March)				Test II (April)				Test III (July)			
	Acc.	Rec.	Prec.	F-Meas.	Acc.	Rec.	Prec.	F-Meas.	Acc.	Rec.	Prec.	F-Meas.
3	68%	0.68	0.67	0.68	69%	0.68	0.66	0.67	66%	0.68	0.66	0.67
4	72%	0.72	0.71	0.71	71%	0.71	0.68	0.70	68%	0.68	0.66	0.67
5	73%	0.73	0.71	0.72	73%	0.72	0.70	0.71	70%	0.70	0.68	0.69
6	73%	0.71	0.71	0.71	76%	0.74	0.72	0.73	71%	0.71	0.69	0.70
7	76%	0.74	0.74	0.74	**77%**	0.74	0.73	0.74	73%	0.72	0.71	0.71
8	75%	0.72	0.73	0.72	**77%**	0.73	0.73	0.73	73%	0.71	0.70	0.70
9	76%	0.72	0.74	0.73	76%	0.73	0.73	0.73	74%	0.71	0.71	0.71
10	76%	0.73	0.75	0.74	76%	0.72	0.73	0.73	75%	0.71	0.72	0.72
11	**77%**	0.73	0.75	0.74	**77%**	0.73	0.73	0.73	**75%**	0.72	0.72	0.72
12	**77%**	0.74	0.75	0.75	**77%**	0.74	0.74	0.74	74%	0.71	0.71	0.71
13	**77%**	0.73	0.75	0.74	**77%**	0.73	0.74	0.73	73%	0.70	0.70	0.70

Table 5.3 Single tweet gender categorisation of Arabic using PPMD for different orders.

In order to compare these results with other classifiers, several machine learning algorithms from WEKA (Hall *et al.*, 2009) were used. Training and testing sets for various WEKA classifiers need to run through a *string-to-word-vector* filter. The filter for this research was using the 'frequency-inverse document frequency' *(tf-idf)* measure. For gender categorisation, the top 1000 words to appear in the filter from each category (female/male) were chosen. This ensures that the classifiers were exposed to more contexts for both genders. We did not undertake further pre-processing of the data such as stemming, tokenisation or removal of stop word, as the intention was to mimic the same approach taken for the PPM-based experiments. Three different algorithms – MNB, KNN, and LibSVM – were applied.

Table 5.4 shows that the LibSVM classifier outperforms the other machine learning classifiers with an average accuracy of 70%. MNB performed slightly less efficiently than LibSVM achieving an average accuracy of 68%. Lastly, the accuracy of the KNN classifier is the lowest of the three machine learning algorithms with an average accuracy of 64%. Results are shown in Table 5.4 below. Overall, compared with the

previous PPMD experiment, it can be seen that PPMD with order 11 outperforms all the machine learning algorithms that were tested with an average accuracy of 76.3%.

Tests	**Test I (March)**				**Test II (April)**				**Test III (July)**			
Measures	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
MNB	68%	0.67	0.67	0.67	70%	0.70	0.70	0.70	67%	0.67	0.67	0.67
libSVM	69%	0.69	0.69	0.69	72%	0.72	0.72	0.72	70%	0.70	0.69	0.69
KNN 1	64%	0.64	0.64	0.64	66%	0.66	0.66	0.66	63%	0.63	0.63	0.63
PPMD 11	**77%**	0.73	0.75	0.74	**77%**	0.73	0.73	0.73	**75%**	0.72	0.72	0.72

Table 5.4 Results for single tweet gender categorisation using machine learning classifiers.

As mentioned earlier, one of the objectives of this work was to investigate the use of different feature sets such as word N-grams features (unigram, bigram, and trigram), and character N-grams (1-6). We added word trigrams this time to examine a wider range of word features. However, we performed the experiment with a combination of all three test sets combined into one test set (see Table 5.1 above).

Classifiers		**MNB**				**LibSVM**				**KNN 1**			
Features		**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
Word	**Unigrams**	**68.8**	0.68	0.68	0.68	71.0	0.71	0.70	0.70	64.9	0.65	0.64	0.64
	Bigrams	66.9	0.66	0.63	0.64	66.7	0.66	0.44	0.53	67.4	0.67	0.64	0.60
	Trigrams	67.7	0.67	0.68	0.67	66.6	0.66	0.44	0.53	**67.6**	0.67	0.64	0.60
Character	**Unigrams**	67.8	0.67	0.65	0.65	69.9	0.70	0.69	0.69	64.0	0.64	0.64	0.64
	Bigrams	67.8	0.67	0.68	0.68	**71.9**	0.71	0.71	0.71	46.8	0.46	0.69	0.43
	Trigrams	67.9	0.67	0.68	0.68	71.2	0.71	0.69	0.70	45.0	0.45	0.66	0.40
	4-grams	67.2	0.67	0.67	0.67	70.0	0.70	0.68	0.69	51.7	0.51	0.64	0.51
	5-grams	67.2	0.67	0.66	0.67	70.3	0.70	0.70	0.70	57.0	0.57	0.64	0.58
	6-grams	66.3	0.66	0.65	0.65	69.1	0.69	0.68	0.68	64.2	0.64	0.63	0.63

Table 5.5 Results for single tweet gender categorisation using different features.

The best results for MNB occurred when using word unigrams achieving an accuracy of 68.8%, slightly better than when using character trigrams with accuracy of 67.9%. In contrast, LibSVM achieved the best results with character bigram and trigram

features with accuracy of 71.9% and 71.2%, respectively. KNN achieved the best results with word trigram features with an accuracy of 67.6%.

However, the same combinations of test sets were applied to PPMD which outperformed the other machine learning classifiers. PPMD performed well with order 12 achieving an accuracy of 76% with an F-measure of 0.73. This shows that gender is better identified using long sequences of characters. Note that order 12 is equivalent to 6 Arabic letters as the compression is performed at the byte level. However, further experiments using multiple tweets dataset is needed to confirm this assumption.

Orders	Accuracy	Recall	Precision	F-measure
Order 3	68%	0.68	0.66	0.67
Order 4	70%	0.70	0.68	0.69
Order 5	72%	0.71	0.69	0.70
Order 6	73%	0.72	0.71	0.71
Order 7	75%	0.73	0.72	0.73
Order 8	75%	0.72	0.72	0.72
Order 9	75%	0.72	0.72	0.72
Order 10	**76%**	0.72	0.73	0.72
Order 11	**76%**	0.72	0.73	0.73
Order 12	**76%**	0.73	0.73	0.73
Order 13	**76%**	0.72	0.73	0.72

Table 5.6 Results for single tweets gender categorization using different orders of PPMD.

Referring to the confusion matrix, the LibSVM classifier using character bigrams mis-classified female tweets more than male tweets. In contrast, the MNB classifier using word unigrams identifies male tweets more than female tweets, but it also identified female tweets twice as much as the LibSVM classifier similar to the KNN using word unigrams features. Finally, The total number of correctly classified instances for PPMD was 5277 including 3834 instances tweets categorised for males and 1443 for tweets categorised for females (see Table 5.7 for the confusion matrix). Although PPMD

identified gender in single tweets best in our experiment, it still mis-classified 37% of the female tweets and 16% of the male.

Confusion matrix	PPMD order 12		MNB		LibSVM		KNN1	
Gender	Male	Female	Male	Female	Male	Female	Male	Female
Male	3834	758	3564	1027	4354	237	3388	1203
Female	856	1443	1120	1179	1756	543	1211	1088

Table 5.7 Confusion matrix for single tweet gender categorisation using PPMD, MNB, LibSVM, and KNN.

5.5. Author Gender Profiling

In this section, the results for author gender profiling were reported where multiple tweets for each author were classified. A total of 101 authors were categorised according to their gender and performed the experiment again on the three test sets. Overall, the accuracy of identifying gender when combining multiple tweets for each author was found to be higher than identifying the gender of a single tweet. We investigated using different orders of PPMD from order 3 to order 13. Overall, the best accuracy was obtained for the Test I set using order 11 with an accuracy of 88%. Test II and Test III achieved best results using order 9. Table 5.8 reports the results.

Orders	Test I (March)				Test II (April)				Test III (July)			
	Acc.	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
3	77%	0.77	0.76	0.76	80%	0.78	0.79	0.79	79%	0.79	0.78	0.78
4	77%	0.77	0.76	0.76	81%	0.80	0.80	0.80	78%	0.77	0.76	0.77
5	80%	0.80	0.80	0.80	79%	0.78	0.78	0.78	77%	0.76	0.75	0.75
6	79%	0.78	0.79	0.78	83%	0.81	0.83	0.82	78%	0.77	0.76	0.77
7	85%	0.82	0.86	0.84	83%	0.81	0.83	0.82	77%	0.75	0.75	0.75
8	86%	0.83	0.89	0.86	**85%**	0.82	0.86	0.84	80%	0.77	0.79	0.78
9	87%	0.84	0.89	0.87	**85%**	0.82	0.86	0.84	**84%**	0.81	0.84	0.83
10	86%	0.83	0.89	0.86	84%	0.81	0.85	0.83	83%	0.80	0.84	0.82
11	**88%**	0.86	0.89	0.88	84%	0.81	0.85	0.83	82%	0.79	0.81	0.80
12	87%	0.85	0.88	0.86	84%	0.81	0.85	0.83	82%	0.79	0.81	0.80
13	87%	0.85	0.88	0.86	**85%**	0.83	0.85	0.84	83%	0.80	0.84	0.82

Table 5.8 Author gender profiling of Arabic tweets using PPMD.

Machine learning algorithms were also applied to author gender categorisation on the three different testing sets. MNB achieved the best results with an average accuracy of 75.3%. LibSVM achieved an average of 70.0% across all testing sets. Lastly, KNN1 performed lowest results achieving an accuracy of 62.6%. Overall, from the previous experiment, PPMD performed better compared to the other machine learning algorithms with an average accuracy of 85.3%. However, further investigation is required using various word and characters N-gram features.

Test	Test I (March)				Test II (April)				Test III (July)			
Measures	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
MNB	75%	0.75	0.75	0.75	73%	0.74	0.74	0.74	78%	0.78	0.80	0.79
libSVM	69%	0.69	0.68	0.68	70%	0.70	0.70	0.70	71%	0.71	0.71	0.71
KNN 1	63%	0.63	0.67	0.65	63%	0.63	0.69	0.66	62%	0.62	0.57	0.59
PPMD 9	**87%**	0.84	0.89	0.87	**85%**	0.82	0.86	0.84	**84%**	0.81	0.84	0.83

Table 5.9 Experimental results for author gender profiling of Arabic tweets.

As an additional comparison, further experiments were performed with other features such as word N-grams (unigram, bigram, and trigram), and character N-grams (1-6). All the test sets for each author were combined into one test set. MNB identified gender best using high number of characters 5-gram features with an accuracy of 78%. While LibSVM identified gender best with character unigram features with an accuracy of 76%. KNN identified gender best with word trigrams features with an accuracy of 70%. The results are reported in Table 5.10.

Classifiers		**MNB**				**LibSVM**				**KNN 1**			
Features		**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
Word	**Unigrams**	75%	0.75	0.75	0.75	71%	0.71	0.70	0.70	62%	0.62	0.76	0.49
	Bigrams	76%	0.76	0.76	0.76	60%	0.60	0.36	0.48	62%	0.62	0.62	0.62
	Trigrams	77%	0.77	0.77	0.77	60%	0.60	0.36	0.48	**70%**	0.70	0.71	0.70
Character	**Unigrams**	68%	0.68	0.70	0.68	**76%**	0.76	0.76	0.76	68%	0.68	0.67	0.67
	Bigrams	72%	0.72	0.71	0.71	68%	0.68	0.79	0.61	41%	0.41	0.76	0.26
	Trigrams	71%	0.71	0.71	0.71	62%	0.62	0.63	0.52	40%	0.40	0.76	0.24
	4-grams	75%	0.75	0.76	0.75	68%	0.68	0.71	0.63	40%	0.40	0.76	0.24
	5-grams	**78%**	0.78	0.78	0.78	67%	0.67	0.66	0.65	40%	0.40	0.76	0.24
	6-grams	75%	0.75	0.76	0.75	74%	0.74	0.74	0.74	62%	0.62	0.67	0.51

Table 5.10 Results for author gender profiling using different features.

Also, a combination of three test set were applied for gender profiling using PPM. The results shown in Table 5.11 show that PPMD identified gender best with order 13 achieving an accuracy of 88%. Overall, the gender categorisation experimental results, it can be seen that using PPMD in categorising author gender outperforms other machine learning classifiers. Furthermore, the result shows clearly that gender can be predicted best with higher order using PPMD.

Orders	Accuracy	Recall	Precision	F-measure
Order 3	78%	0.77	0.77	0.77
Order 4	79%	0.78	0.78	0.78
Order 5	80%	0.79	0.79	0.79
Order 6	82%	0.81	0.81	0.81
Order 7	85%	0.85	0.83	0.84
Order 8	86%	0.87	0.83	0.85
Order 9	87%	0.89	0.84	0.87
Order 10	86%	0.88	0.83	0.85
Order 11	86%	0.88	0.83	0.85
Order 12	87%	0.89	0.84	0.87
Order 13	**88%**	0.90	0.85	0.87

Table 5.11 Results for author gender profiling using different orders of PPMD.

5.5.1. Mis-classified instances

For each of the incorrect instances from the author profiling experiment, further investigation was undertaken to gain a better insight into the effectiveness of the classification. The confusion matrix is reported as shown in Table 5.12.

Confusion matrix	Male	Female
Male	60	1
Female	11	29

Table 5.12 Confusion matrix for all test sets using PPM order 13.

From the confusion matrix, it can be seen that only one male user was classified as female from a total of 61 male users. The PPM codelength difference between the female and male models was relatively low, being less than 10.16 bits (see Table 5.13).

User	Original	Classified	Female (bits)	Male (bits)	Difference (bits)
HaydarAlmudafar	Male	Female	1495.082	1505.242	10.016
Ola_Zngna	Female	Male	2122.946	2112.374	10.572
NerminGaga	Female	Male	5078.898	4679.036	399.862
Nawal_Aleed	Female	Male	22336.309	21901.215	435.094
AniaEl	Female	Male	26182.217	25965.645	216.572
Amira	Female	Male	10527.591	10392.348	135.243
moalhasan	Female	Male	10608.572	9714.538	894.034
loubabah	Female	Male	14806.273	14511.586	294.687
lilianetannoury	Female	Male	11059.908	10922.140	137.768
hassinaouch	Female	Male	34979.020	32719.201	2259.819
ferferdaous	Female	Male	2192.403	2142.364	50.039
EHSANFAKEEH	Female	Male	28075.094	27726.158	348.936

Table 5.13 Codelength and differences between the incorrectly classified author genders.

In contrast, 11 female users were classified as male. Eight authors out of 11 were classified as male with a codelength difference less than 400 bits. Only three authors were classified with over 400 bits difference in codelength. A possible reason for the mis-classification is that the cost of compression in some tweets is affected by the topic. For instance, the author Nawal_Aleed mostly tweets about religion. This affects the gender classification as most of the authors who tweet about religion are males. Similarly, the authors moalhasan and hassinaouch mostly tweet about politics. This shows how tweet topics can effect the prediction of the user's gender.

5.6. Improving gender classification

Following the classification procedure in section 5.4.3, the training model was previously trained on the concatenation of training text available for each class (i.e. male and female) using Protocol I (see section 2.9). In this current experiment, we wanted to apply a new protocol to see if this would lead to any improvement.

The idea behind the new protocol is as follows. Often there is insufficient training text for Protocols III and IV which uses training text which is too specific or limited to each training document (i.e. each tweet when classifying Twitter data), whereas there might be too much training text for Protocols I and II which might lead to mis-classifications because models are too general. Perhaps an alternative protocol might be more effective if it used more training models where there would be less (but more specific) training text to train on.

One way to achieve this is to create models for another secondary class type different to that used for the primary categorisation task. This is possible as long the primary class information can be deduced once the secondary class was identified. This class type might also be easier to recognize. For example, for gender classification, the primary class type is gender. A secondary class type might be authorship. Once a specific author was identified, then the gender could readily be deduced if this information was available during training. Using authorship as the secondary class type might also lead to improved performance because previously published text categorisation results for this class type using PPM have been better (in terms of accuracy, recall and precision, for example) compared to the gender class type. A key insight is that even though the authorship for the test documents may not be completely known, the classification process will guess the gender from a known author found in the training set which has the most similar properties in the test set in terms of the language used. This suggests that the following two new protocols shown in Table 5.14 could be added to the Protocols Table that was previously provided in Table 2.11.

Protocols	Static Models	Dynamic Models
Concatenation of training texts for each primary class	SMDL (Protocol I)	AMDL (Protocol II)
Non-concatenation of training texts for each primary class	Protocol III	BCN (Protocol IV)
Concatenation of training texts for another secondary class type	Protocol V	Protocol VI

Table 5.14 Amended Protocols for text categorisation.

The number of training models required for each protocol is as follows: C_1 for Protocols I and II where C_1 is the number of classes for the class type used for the primary categorisation task (e.g. two for Gender i.e. “Male” and “Female”); T for Protocols III and IV where T is the number of training documents and where T is usually much greater than C_1; and C_2 for Protocols V and VI where C_2 is the number of classes for a class type different to the primary categorisation task and where C_2 would be greater than C_1, but still much less than T (e.g. if an authorship secondary class type is used, then this would be equal to 101 for BTAC).

For the gender classification experiment reported in this section (and later on for dialect identification in section 6.6.2 of Chapter 6), only Protocol V has been investigated since Protocol VI (using dynamic models rather than static models) requires substantially more resources to perform the classification procedure. This is because dynamic models update continuously during the testing process whereas static models remain fixed once the training process is completed.

Figure 5.4 demonstrates the gender classification procedure that uses Protocol V. This shows that the training process uses authorship files as a secondary class type. Note that the primary class (gender) can be detected since it is available during the training process.

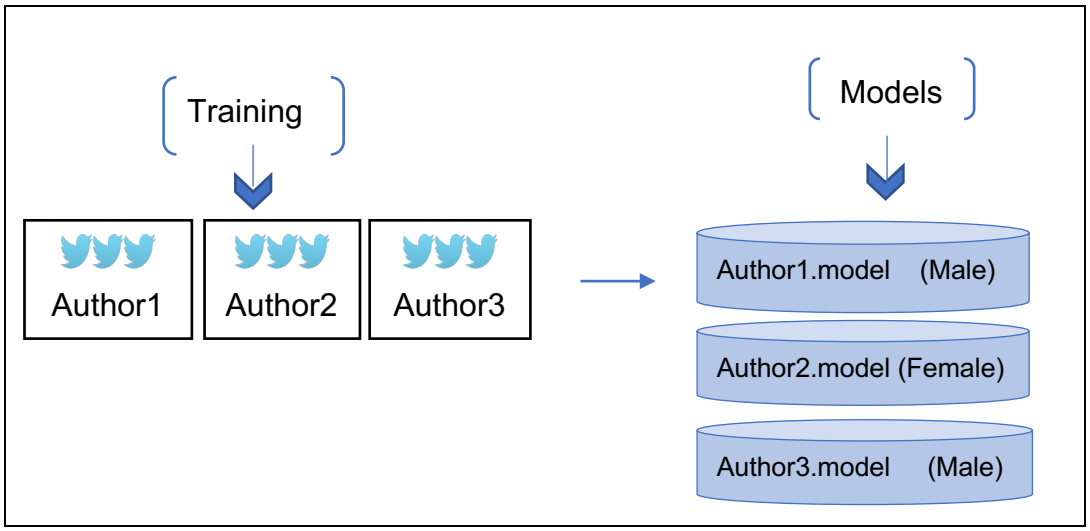

Figure 5.4 Training process for the new concatenated author models (Protocol V).

Table 5.15 shows the experimental results when using PPMD using Protocol V on higher orders. It shows the best results for all the experiments in this chapter. It can be seen that the accuracy of identifying single tweets using the concatenated model was 76%, whereas using the concatenated author models yielded an accuracy of 81%. Similarly, the accuracy of the author profiling experiment increased to 97% compared to 88% using the concatenated model. This shows that the use of the new protocol helps to improve the results of gender categorisation for both single tweet and multiple tweets.

Tests	**Single Tweets**				**Author profiling**			
Orders	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
Order 11	**81%**	0.770	0.780	0.775	**97%**	0.976	0.962	0.969
Order 12	**81%**	0.776	0.783	0.779	**97%**	0.976	0.962	0.969
Order 13	**81%**	0.777	0.783	0.778	**97%**	0.976	0.962	0.969

Table 5.15 Results for gender categorisation using Protocol V.

One of the benefits of using this new protocol for gender categorisation is that it helps to identify gender using text from multiple tweets written by (possibly) other authors with the same gender. There was a downside of using concatenated models (Protocol I) when female authors were classified as male, as shown in Table 5.13. That was because, in the case of some authors, the cost of compression in some tweets is affected by the topic. However, the use of the new protocol helps to categorise gender by comparing with other authors who have the same gender. Table 5.16 shows the miss-classified author genders after using the new protocol. It clearly shows that the use of author-trained models also helps to decrease the average codelengths between the authors compared to the results in Table 5.13.

Users	Original	Classified	Female (bits)	Male (bits)	Difference (bits)
hassinaouch	Female	Male	38646.023	38146.188	499.835
Ola_Zngna	Female	Male	2288.389	2261.728	26.661
Raghooda	Female	Male	578.810	565.202	13.608

Table 5.16 Codelengths and differences between the incorrectly classified author genders using Protocol V.

5.7. Discussion and findings

In this chapter, we examined the performance of the selected classifiers in gender categorisation of single tweets and authors' gender profiling for multiple tweets. As shown in Table 5.6, PPMD outperforms other machine learning classifiers as it achieved better accuracy, F-measure, recall, and precision in all three test sets, at 76.4%, 0.73, 0.73, and 0.73, respectively. Out of the three test sets with a total of 6990 tweets (male 4691 and female 2299), 5277 tweets were correctly classified (3834 tweets for males and 1443 for females). This compares well with results reported by other researchers – for example in English, the study by Burger et al. (2011) which reported an accuracy of 74% accuracy and the study by Rao et al. (2010) which reported an accuracy of 72.33%. Marquardt et al. (2014) used SVM to detect gender on the Twitter dataset and reported an accuracy of 71.15%, while AlSukhni and Alequr (2016) achieved 60% accuracy in identifying the gender of 8000 Arabic tweets using MNB. Furthermore, the results of this current research were improved by using a new protocol that achieved accuracy, F-measure, recall, and precision of 81.6%, 0.77, 0.77, and 0.78, respectively.

Also, the research investigated author gender profiling – reported in Table 5.11 – which we found to significantly improve results for accuracy, F-measure, recall, and precision, achieving 88.1%, 0.87, 0.85, and 0.90, respectively. These results compare well with other researchers results in Arabic on different datasets such as the studies by Alrifai et al. (2017) which achieved 66.38% of accuracy, and Alsmearat et al. (2014) which reported 86.4% of accuracy. In English, Estival et al. (2007) which achieved 81.15% accuracy, Liu and Ruths (2013) which reported 87.1%, Marquardt et al. (2014)

which reported 71.1%, Ugheoke and Saskatchewan (2014) which reported 86.8%, and Mikros (2012) which yielded 82.6% for the Greek language. In addition, the results of this current research were improved using a new protocol achieving accuracy, F-measure, recall, and precision of 97.0%, 0.96%, 0.97, and 0.96, respectively.

Moreover, the research tested the effect of using word features such as word N-grams and character N-gram features. In single tweet categorisation, LibSVM performed best with character bigrams with an accuracy of 71.9%. In contrast, MNB achieved the best result with word unigrams features achieving an accuracy of 68.8%. However, in the author gender categorisation experiment, LibSVM performed best with character unigram achieving an accuracy of 76%. MNB achieved the best result with word character 5-grams features achieving an accuracy of 78%. Similarly, it has been found that PPMD with higher orders such as 12 and 13 worked best. This finding corresponds to those of other researchers who found that higher orders of N-grams are most effective for gender categorisation (Deitrick *et al.*, 2012b, 2012a; Mikros, 2012) using English and Greek language respectively, and (Alrifai *et al.*, 2017) using Arabic language.

To understand why gender was identified best with higher orders, Table 5.17 presents a selection of prominent gender-specific features produced using character 6-grams that represent each gender. This shows that the noun "#رواية" translated as "#novel" is a common expression for females, whereas "الكتاب" translated as "Book" is a popular expression used by males. Moreover, writing "Thanks" is different between both genders; females write it repeating the last Arabic letter "ا" as "شكرااا"; instead. In contrast, males write the word using double diacritics fathah as "شكراً".

Female			Male		
Feature	**Translation**	**Count**	**Feature**	**Translation**	**Count**
😂😂😂😂😂😂	-	100	شكراً	Thanks	169
حبيبتي	Love	85	مباراة	Match	161
#رواية	#novel	85	الشباب	The guys	75
فاطمة	Fatimah	79	ميلان	Milan	73
شكرااا	Thanks	68	تسلملي	Appreciate	62
رواية_	_novel	62	الكتاب	Book	55
فها_حب	Contains_love	62	اليوفى	Juventus	44
.....	-	60	😄😄😄😄😄😄	-	36
🌹💝😍	-	18	😂😂😂😂😂	-	27

Table 5.17 A selection sample of prominent gender-specific features.

5.8. Conclusion

In this chapter, we presented the results of gender categorisation in Arabic using Prediction by Partial Matching (PPM) on both single tweets achieving an accuracy of 76.4%, and multiple tweets (author profiling) achieving an accuracy of 88.1% using concatenated models (Protocol I). We compared these results with other machine learning algorithms. In addition, various features were applied such as character N-grams and word N-grams. Overall, the results show that PPM significantly outperforms other mainstream machine learning methods for gender categorisation of Twitter data. Furthermore, the results of this current research were improved by using concatenated author models to identify a secondary class (Protocol V) on both single tweets gender categorisation achieving an accuracy of 81.6%, and multiple tweets (author profiling) achieving an accuracy of 97.0%. This work would be useful for such applications where user's profile is examined and needs to be checked regularly based on the text content in order to detect false profiles for further investigation.

Chapter 6

Dialect Identification of Arabic Tweets Using PPM

6.1 Introduction

Arabic dialects refer to the spoken variations of Arabic language which differ according to the geographical region. Although the written form of Arabic is still considered the standard form of the language which is used formally, a dialect is more often used unofficially among people from a specific region on a daily basis. Recently, it has been noticed that the written form of dialects is being used more frequently for informal written communication on the web (Hamdi *et al.*, 2015). However, most of the early research on languages has involved identification between the main languages. Dialect identification in social media is a recently emerging phenomenon in the field of text categorisation.

Meanwhile, the main way to identify the *source* of a tweet is to trace the *location* of the tweet (latitude and longitude). The problem is that researchers rely heavily on the location of the tweet rather than on identifying the dialect from the content of the text. With the existence of many fake accounts and bots in social media which use hidden locations to spread rumours or start political propaganda, it is becoming vital to study the language used besides other elements to identify the actual source of the tweets rather than relying on misleading location information.

Part of this chapter is based on a paper accepted for publication in the International Journal of Computational Linguistics (IJCL):

> *Altamimi, M., and Teahan, W.J., "Arabic Dialect Identification of Twitter text Using PPM Compression." International Journal of Computational Linguistics (IJCL) 10(4) (2019): 47-59.*

The rest of the chapter is organised as follows: section 6.2 discusses the goals of this chapter; section 6.3 describes the experimental setup for both single tweet's dialect identification and multiple tweets author-based dialect identification; and sections 6.4 and 6.5 report the results of both experiments with a focus on the incorrectly classified instances and the improvement of the results using the new classification protocol (Protocol V). Next, section 6.6 discusses the findings from both experiments, and lastly section 6.7 concludes the chapter and suggests future work.

6.2 Goals for the investigation

This section describes the set of goals to be achieved in this chapter. It first examines the effect of PPM in recognising Arabic dialects text using the BTAC corpus. In particular, it aims to identify the dialects in a single tweet and the author's dialects with multiple tweets, and then compare the results with those obtained from using the machine learning algorithms. In addition, it employs various features such as characters and word features to help identify the dialects. Finally, it analyses the mis-classified tweets in both experiments (single tweets, and author tweets). These analyses will give better insight into the task to provide some feedback to improve the results.

This chapter's aims are summarised as follows:

- to use PPM to identify Arabic tweets for single tweets and for multiple authors' tweets;
- to compare the results with the benchmark machine learning algorithms;
- to apply various variants such as character- and word-based approaches;
- to analyse the incorrectly classified instances in order to find any improvements.

6.3 Experimental Setup

This section describes the setup for the experiments that were conducted for this study. These experiments adopted a supervised classification approach for this task by applying both PPM and machine learning algorithms to identify Arabic dialects

using the BTAC corpus. An explicit testing set was used to identify which dialect each tweet belongs to. Moreover, the experiments attempted to identify each author's dialects from the 101 authors collected from the corpus. Two feature types were employed for these tasks, as described below.

6.3.1 Character-based classification approach

PPM and machine learning algorithms were applied on sub-word features using constituent characters. PPM classification using different orders were performed from 3 to 13. After finding the best order, these results were compared with those from machine learning algorithms using character N-grams (1-6). The character-based approach allowed for implicitly capturing various sub-lexical features such as single letters, suffixes, prefixes, and morphemes.

6.3.2 Feature-based classification approach

For the machine learning based approaches, word N-grams were also investigated to compare these with the results generated from the character-based approach. Each feature (i.e. unigram, bigrams and trigrams) was extracted with its frequency distribution. The features' frequencies are weighted using the *tf-idf* weighting scheme, as this proved to be effective in previous research.

6.3.3 Dataset

The dataset used Bangor Twitter Arabic corpus (BTAC) which is purposely designed for dialect research (see chapter 3). The corpus covers five major Arabic dialect groups: Gulf; Egyptian; Levantine; Maghrebi; and Iraqi; in addition to Modern Standard Arabic (MSA) and Classical Arabic (CA). The corpus contains over 120K tweets annotated according to the different Arabic dialects. The corpus is collected from 101 authors from the Middle East, with all the tweets manually annotated according to the dialects and independently verified by two experts. An explicit testing set has been created for testing purposes rather than splitting the training set. The test set was collected from the same users for three different time periods to verify that there is no

overlap between the training and testing set, and to reflect a real-world data collection scenario. The total number of tweets collected for each dialect including training and testing sets are listed in Table 6.1. The training phase uses the concatenation of training text available for each class (i.e. Gulf; Egyptian; Levantine; Maghrebi; Iraqi; MSA; and CA) using Protocol I (see section 2.11).

Dialects	Train	Test I (March)	Test II (April)	Test III (July)
All Tweets	114,956	1,481	2,655	2,754
MSA	42,658	757	1,276	1,362
CA	31,006	301	590	593
Gulf	9,148	128	260	233
Egyptian	9,057	98	221	248
Mixed	8,343	49	64	36
Levantine	7,857	70	150	144
Maghrebi	3,980	50	47	70
Iraqi	1,884	12	29	45
emojis	1,023	16	18	23

Table 6.1 Breakdown of the tweets used for the dialect identification experiments.

The first experiment is applied to the task of identifying the dialect for single tweets with there being three different explicit testing sets. This experiment is designed in order to find out whether the system would be able to identify the dialect from a single tweet, knowing that a single tweet is written in 140 characters or less. This is considered challenging as the system is being provided with minimal dialectal context.

In the second experiment, tweets are combined for each author in order to investigate dialect identification for each author. All the three test sets for each author were combined to one file; then each author is labelled with the highest number of tweets generated from the author's training file (Modern, Classic, Gulf, Egyptian, Levantine, Maghrebi, or Iraqi). The goal of this experiment is to identify the dialect used by each author. This dataset provides better insight for each author as multiple tweets from each author are collected. Also, it can be considered less challenging than the first dataset as the system is being exposed to more data.

Figure 6.1 shows the actual dialect used for each author before and after the manual annotation. Before the annotation, each author was chosen based on geolocation with a total of 101 authors for all dialects. However, after the annotations, we can see that the majority of authors were added to both MSA and CA style instead of their main dialect. This is because each author was labelled according to the most used dialects found in the training sets.

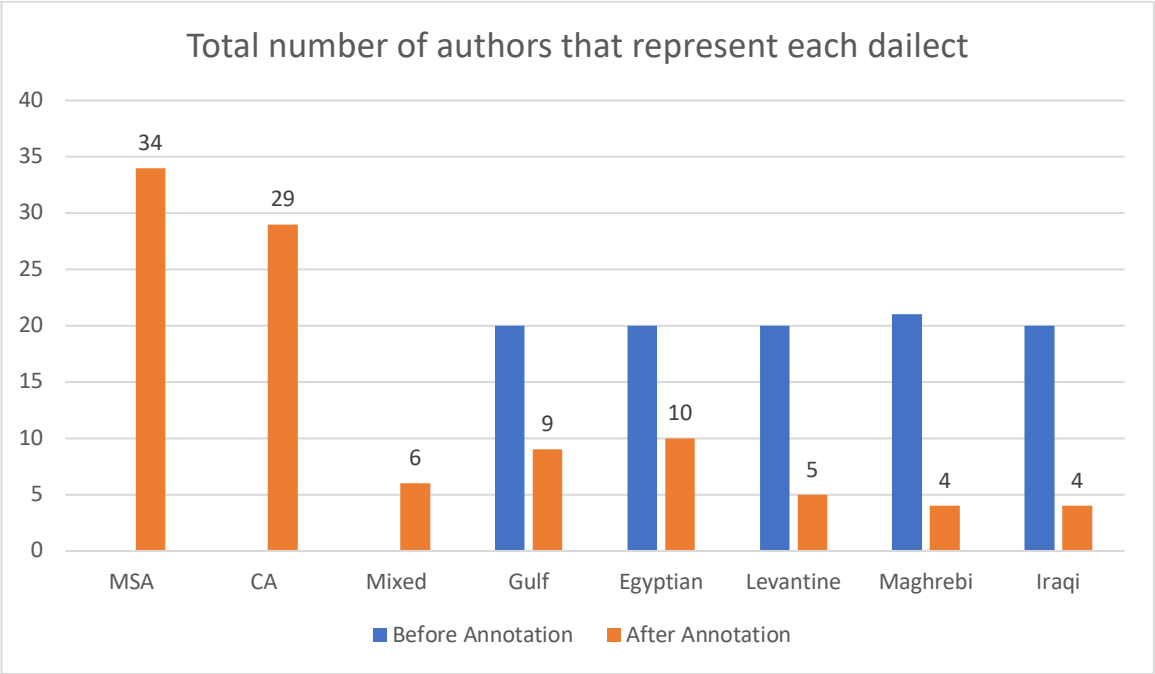

Figure 6.1 Number of users that represent each dialect before and after the annotation.

6.4 Dialect identification of tweets: Experimental Results

In this experiment, a multiclass classification (7-way) task for identifying each single tweet's dialects (Modern, Classic, Gulf, Egyptian, Levantine, Maghrebi, Iraqi). This was investigated using different orders of PPMD from order 3 to order 13. We started with order 3 as lower orders such as 2 performed worse. Each tweet is split into a single file; over 6573 tweets were tested in three different test sets. While other research experiments perform dialects classification using machine learning

algorithms, this experiment is novel in its approach to dialect classification for Arabic text as it investigates the use of the character-based text compression scheme PPM. Each order experiment, below, was tested on three different test sets. Overall, order 6 performed better in terms of F-measure with an average of 0.63, while order 11 performed better in terms of accuracy with an average of 74%.

Orders	Test I (March)				Test II (April)				Test III (July)			
	Acc.	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
3	66%	0.61	0.50	0.55	66%	0.58	0.49	0.53	65%	0.60	0.50	0.55
4	69%	0.66	0.54	0.60	70%	0.61	0.53	0.56	70%	0.67	0.56	0.61
5	71%	0.64	0.56	0.60	72%	0.63	0.56	0.59	72%	0.68	0.59	0.63
6	72%	0.65	0.57	0.61	**74%**	0.64	0.59	0.61	**74%**	0.68	0.62	0.64
7	73%	0.65	0.58	0.61	**74%**	0.63	0.59	0.61	**74%**	0.66	0.62	0.64
8	72%	0.64	0.59	0.61	**74%**	0.61	0.60	0.60	73%	0.64	0.62	0.63
9	73%	0.65	0.60	0.62	**74%**	0.61	0.60	0.60	73%	0.63	0.60	0.62
10	73%	0.63	0.58	0.60	**74%**	0.60	0.60	0.60	73%	0.63	0.60	0.63
11	**74%**	0.63	0.59	0.61	**74%**	0.59	0.59	0.59	**74%**	0.65	0.63	0.64
12	73%	0.63	0.58	0.61	**74%**	0.60	0.59	0.59	73%	0.64	0.61	0.62
13	73%	0.64	0.59	0.61	**74%**	0.60	0.60	0.60	73%	0.64	0.61	0.62

Table 6.2 Dialect identification of Arabic single tweets using PPMD.

However, three machine learning classifiers were also investigated using Weka (Hall *et al.*, 2009): Support Vector Machines specifically the LIBSVM package (Chang and Lin, 2011); Multinomial Naïve Bayes (MNB); and k-nearest neighbours (KNN). In order to classify text for machine learning algorithms using Weka, training and testing sets need to be run through a *string-to-word-vector* filter. The filter for this experiment was built using the common term frequency-inverse document frequency *(tf-idf)* measure. Multiple tokenisers were also used such as word N-grams and character N-grams; however, no further pre-processing of the data was carried out, such as stemming, tokenisation, and removal of stop words, as the intent was to mimic the same approach used for PPM.

Test	Test I (March)				Test II (April)				Test III (July)			
Measures	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
MNB	65%	0.65	0.69	0.66	69%	0.69	0.70	0.69	69%	0.70	0.72	0.70
LibSVM	66%	0.66	0.66	0.66	65%	0.66	0.67	0.66	69%	0.69	0.69	0.69
KNN 1	53%	0.54	0.54	0.54	53%	0.54	0.53	0.53	52%	0.52	0.52	0.52
PPMD6	**72%**	0.65	0.57	0.61	**74%**	0.64	0.59	0.61	**74%**	0.68	0.62	0.64

Table 6.3 Experimental results for dialect identification of Arabic single tweets using Machine learning classifiers and PPM.

Best results are obtained using the Multinomial Naïve Bayes classifier with an average accuracy of 67.6%. LibSVM performed less then MNB with an average accuracy of 66.6%. Finally, KNN performed worse with an average accuracy of 52.6%. The experimental results are reported in the Table 6.3. When comparing all methods, we can see that PPM performs better in terms of accuracy while MNB and LibSVM performs better in terms of F-measure.

Classifiers		MNB				LibSVM				KNN 1			
Features		**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**	**Acc.**	**Rec.**	**Prec.**	**F-Meas.**
word	**Unigrams**	**68.8**	0.69	0.68	0.68	67.3	0.64	0.67	0.65	53.4	0.53	0.53	0.53
	Bigrams	57.6	0.54	0.57	0.56	50.8	0.34	0.50	0.43	**56.4**	0.52	0.56	0.54
Character	**Unigrams**	56.2	0.54	0.56	0.55	58.5	0.54	0.58	0.55	48.5	0.49	0.48	0.50
	Bigrams	64.0	0.64	0.64	0.64	69.1	0.68	0.69	0.68	20.2	0.28	0.20	0.64
	Trigrams	68.3	0.68	0.68	0.68	**72.8**	0.72	0.73	0.72	20.2	0.22	0.20	0.60
	4-grams	67.9	0.68	0.67	0.67	71.8	0.71	0.71	0.72	26.2	0.30	0.26	0.56
	5-grams	66.4	0.66	0.66	0.66	68.2	0.66	0.68	0.68	37.2	0.41	0.37	0.56
	6-grams	64.2	0.64	0.64	0.64	63.2	0.58	0.63	0.65	53.3	0.52	0.53	0.52

Table 6.4 Dialect identification of Arabic single tweets using different features.

As mentioned earlier, one of the goals of this work is to apply different feature sets; the word N-grams features (unigram and bigram), and character N-grams features (1-6). All three tests were combined into one test set resulting in a total of 6573 single tweets. Table 6.4 reports the experiment results. In general, character trigrams

features identified single dialect tweets best using LibSVM achieving an accuracy of 72.8%. MNB identified tweets best with the word unigram feature, achieving an accuracy of 68.8%. However, the KNN1 classifier found word bigram performed best with an accuracy of 56.4%. Overall, in terms of F-measure, LibSVM was found to perform well with results as high as 0.73.

Orders	**Accuracy**	**Recall**	**Precision**	**F-measure**
Order 3	66.2	0.60	0.50	0.54
Order 4	70.2	0.65	0.55	0.59
Order 5	72.2	0.66	0.57	0.61
Order 6	74.0	0.66	0.60	0.63
Order 7	**74.1**	0.64	0.60	0.62
Order 8	73.8	0.63	0.60	0.62
Order 9	73.8	0.63	0.60	0.61
Order 10	74.0	0.63	0.60	0.61
Order 11	74.0	0.63	0.60	0.61
Order 12	73.7	0.62	0.60	0.61
Order 13	73.5	0.61	0.60	0.61

Table 6.5 Dialect identification of Arabic single tweets using different orders of PPMD.

Table 6.6 shows the confusion matrix for PPMD order 7 which achieved the highest score when compared with other the orders. The confusion matrix shows a close relation between CA and MSA as both are considered formal. However, there are a few uses of words interchangeably which leads to some tweets being mis-classified. Also, the results show more mis-classifications of MSA tweets with other dialects; this shows that there is some overlap between MSA and other dialects which makes it difficult to classify.

The table also shows less confusion between the Egyptian, Levantine and Maghrebi dialects which may be due to the clear features that make it easier for the classifier to

distinguish between these dialects. On the other hand, more confusion for both Iraqi and Gulf dialects can be seen; this is due to the overlap between Gulf and Iraqi dialects as south of Iraq are influenced by the gulf dialects which also lead to confusion between the two dialects. Finally, it is important to note that the classifier used in this experiment has managed to distinguish between dialects despite the data being imbalanced.

PPMD 7	CA	Egyptian	Gulf	Levantine	MSA	Iraqi	Maghrebi
CA	1220	8	12	3	206	6	6
Egyptian	8	418	24	23	73	5	13
Gulf	21	36	363	59	91	22	21
Levantine	11	18	39	228	37	7	18
MSA	507	89	121	33	2503	24	51
Iraqi	3	7	13	16	16	25	4
Maghrebi	4	10	6	13	12	7	113

Table 6.6 Confusion matrix for single tweets dialect identification using PPMD order 7.

6.4.1 Incorrectly classified instances

To improve the identification result, the tweets that were incorrectly classified were analysed further. It was noticed that some of the tweets are not correctly identified due to insufficient context to distinguish the dialect when the tweet is too short. In contrast, very long tweets are likely to have more features that facilitate correct classification. Four further classification experiments were performed by removing tweets that contained fewer than 20, 40, 50, and 60 bytes. A total of 1174 tweets were removed from the testing sets. Assuming the average Arabic token is five characters, the average characters within a 20-byte tweets are just *two tokens*. This because each Arabic character is allocated two bytes. However, the average characters within 40-byte tweets are *five tokens*. In contrast, the regular tweets containing 140 characters require roughly 250 bytes each.

PPMD 7	# of tweets removed	Accuracy (%)	Recall	Precision	F-measure
All test data	Non	74.1	0.649	0.607	0.627
< 20	114	74.9	0.651	0.625	0.638
< 40	522	77.3	0.655	0.680	0.667
< 50	287	78.3	0.655	0.705	0.679
< 60	251	79.1	0.655	0.721	0.686

Table 6.7 Results of dialect identification of Arabic single tweets using PPMD7 after removing tweets containing fewer than 20, 40, 50 and 60 bytes.

From Table 6.7, using PPMD 7 with the best order from the previous experiment, it can be seen that by removing tweets that are less than 20, 40, 50, and 60 bytes, the results gradually improve. Overall, there is a slight rise in the recall by 1.4%. However, there is a dramatic increase of 11.4% with precision from 0.607 to 0.721. This increase improves the F-measure by 5.9%, which also increases the accuracy to reach 79.1%.

MNB word-unigrams	# of tweets removed	Accuracy (%)	Recall	Precision	F-measure
All test data	Non	68.8	0.688	0.713	0.692
< 20	114	69.3	0.694	0.719	0.698
< 40	522	71.5	0.715	0.740	0.719
< 50	287	72.6	0.726	0.750	0.730
< 60	251	73.1	0.732	0.756	0.736

Table 6.8 Results of dialect identification of Arabic single tweets using MNB after removing tweets containing fewer than 20, 40, 50 and 60 bytes.

On the other hand, the same dataset of removed tweets was applied to MNB – see Table 6.8. The table shows that there is a moderate rise in the recall and precision with 4.4%, and 4.3% respectively. This led to a total of 4.4% rise in the F-measure, and the accuracy also increased to reach 73.1%.

Furthermore, Table 6.9 shows the effect of removing tweets for the LibSVM classifier with character trigrams features which performed best previously. Overall, there was

a smaller rise in the recall and precision, both by 3.6%. This led to a 3.7% rise in the F-measure, and also an increase in accuracy to reach 76.4%. This shows that PPM produces better accuracy when tweets that contain less text are removed. The accuracy, recall, and precision after removing 20, 40, 50, and 60 bytes are reported in the tables below.

LibSVM character-trigrams	# of tweets removed	Accuracy (%)	Recall	Precision	F-measure
All test data	Non	72.8	0.728	0.732	0.722
< 20	114	73.4	0.734	0.737	0.728
< 40	522	75.1	0.751	0.755	0.745
< 50	287	75.7	0.758	0.762	0.752
< 60	251	76.4	0.764	0.768	0.759

Table 6.9 Results of dialect identification of Arabic single tweets using LibSVM after removing tweets containing fewer than 20, 40, 50 and 60 bytes.

However, it was noted that a few of the longer tweets were also mis-classified. These tweets were then analysed in more depth to help better understand how difficult the task is. This showed that identifying dialects within tweets can be complicated even for a native speaker, for the following reasons:

- There are no defined boundaries between dialects and modern standard Arabic when dealing with text; for instance, some tweets are influenced by the standard modern Arabic regardless of the dialects used.
- The classification is affected by the topic bias of the tweets; for instance, the tweet might be classified by the tweet's topic regardless of the dialects being used.
- The cost of encoding the text can be dominated by people's names or location when classifying tweets.
- Classifying single tweets is more challenging due to the fact that some tweets contain less dialect content.

6.5 Author dialect identification

As well as classifying single tweets, classification of multiple tweets from the same author were also investigated. Each author's tweets from the three test sets were combined into single files; each file represents tweets from the same author. Each author had their dialect labelled according to the most used dialect found in the training sets, and this dataset was then used for experiments using PPMD and the machine learning algorithms. This classification task is different to the problem of authorship attribution, as the intention is to classify text according to the dialect or dialects used by the author, not to classify text according to which author the tweets belongs to.

Table 6.10 shows the results of classifying a total of 101 authors according to their dialect using MNB, LibSVM, KNN, and PPMD. The best result is achieved using PPMD order 5 reporting an accuracy of 87.1%, slightly better than MNB achieving an accuracy of 86.1%. LibSVM reported an accuracy of 71.2%. Finally, KNN 1 achieved an accuracy of 34.6%. However, further investigation is required using various word and characters N-gram features.

Measures	Correctly classified	Accuracy (%)	Recall	Precision	F-measure
MNB	87	86.1	0.861	0.870	0.862
LibSVM	72	71.2	0.713	0.663	0.672
KNN 1	35	34.6	0.347	0.402	0.190
PPMD 5	88	**87.1**	0.840	0.915	0.876

Table 6.10 Results of author dialect identification using machine learning algorithms and PPMD.

However, various N-grams word and character features were also applied to the machine learning classifiers in Table 6.11. The results show that MNB produces the best result using word unigram features with an accuracy of 86.1%. In contrast, LibSVM achieved the best result using character 4-grams features, with an accuracy

of 78.2%. Also, KNN 1 achieved best results using character unigram features achieving 49.5%.

Classifiers		MNB				LibSVM				KNN 1			
Features		Acc.	Rec.	Prec.	F-Meas.	Acc.	Rec.	Prec.	F-Meas.	Acc.	Rec.	Prec.	F-Meas.
Word	Unigrams	**86**	0.86	0.87	0.86	71	0.71	0.66	0.67	34.6	0.34	0.40	0.19
	Bigrams	80	0.80	0.81	0.79	36	0.36	0.40	0.22	40.0	0.40	0.59	0.29
Character	Unigrams	52	0.52	0.64	0.49	54	0.54	0.58	0.51	**49.5**	0.49	0.57	0.48
	Bigrams	68	0.68	0.73	0.68	60	0.60	0.63	0.55	4.09	0.05	0.28	0.02
	Trigrams	76	0.76	0.78	0.75	73	0.73	0.70	0.70	9.09	0.09	0.29	0.03
	4-grams	80	0.80	0.82	0.79	**78**	0.78	0.74	0.75	9.09	0.09	0.29	0.03
	5-grams	80	0.80	0.81	0.80	75	0.75	0.74	0.72	9.09	0.09	0.29	0.03
	6-grams	80	0.80	0.83	0.80	76	0.76	0.75	0.74	34.6	0.34	0.40	0.19

Table 6.11 Results for dialect identification of Arabic author using different features.

Orders	Accuracy (%)	Recall	Precision	F-measure
Order 2	76.2	0.686	0.792	0.735
Order 3	84.2	0.807	0.861	0.833
Order 4	83.2	0.793	0.867	0.828
Order 5	**87.1**	0.840	0.915	0.876
Order 6	84.2	0.773	0.890	0.827
Order 7	84.2	0.773	0.886	0.826
Order 8	81.2	0.718	0.869	0.786
Order 9	81.2	0.718	0.869	0.786
Order 10	82.2	0.738	0.878	0.802
Order 11	82.2	0.738	0.863	0.796
Order 12	82.2	0.738	0.878	0.802

Table 6.12 Results for dialect identification of Arabic author using PPM with different orders.

To compare the results with PPM, an experiment was performed using various orders of PPMD. Table 6.12 shows that order 5 PPMD outperformed other orders, achieving an accuracy of 87.1%. This result shows that identifying dialect within multiple tweets is better using short sequences of characters. This is probably due to words with two or three characters containing more distinguishable dialect features. This was similar to other Arabic studies that found that character N-grams identifies Arabic dialects best (Darwish *et al.*, 2014; Sadat *et al.*, 2014).

Table 6.13 shows the confusion matrix for the PPMD order 5 classifier which achieved the highest scores in all the experiments as shown in Table 6.12. The confusion matrix shows that there were some confusions between CA and MSA due to the overlap of some features they both use. This supports the earlier informal observation when classifying single tweets. Also, similar to previous experiments performed using single tweets, less confusion among Egyptian, Levantine, and Maghrebi dialects was observed. In contrast, a close relationship can be seen between the Iraqi and Gulf dialects as three Iraqi authors were classified as using the Gulf dialect, so this was examined further below.

PPMD 5	**CA**	**Egyptian**	**Gulf**	**Levantine**	**MSA**	**Mix**	**Iraqi**	**Maghrebi**
CA	26	1	0	0	2	0	0	0
Egyptian	0	8	0	0	0	0	0	0
Gulf	0	0	7	0	0	0	1	0
Levantine	0	0	0	5	0	0	0	0
MSA	3	1	1	0	31	1	0	2
Mix	0	0	0	0	0	5	0	0
Iraqi	0	0	0	0	0	0	3	0
Maghrebi	0	0	0	0	1	0	0	3

Table 6.13 Confusion matrix for author dialect identification using PPMD order 5.

6.5.1 Incorrectly classified instances

After checking the 13 inaccurately classified authors, it appeared that eight of the authors changed their type of writing from one style to the other. For instance: three of the authors changed their style of writing from CA to MSA in the testing set; one author changed their form of writing from MSA to CA; another author used mostly Moroccan dialects instead of MSA; a further one did the opposite, switching from Moroccan to MSA; another changed from writing in Egyptian to write in MSA; and one author changed from writing in Egyptian to write in CA. In the cases of the remaining five authors, one author had less than 10 tweets in their testing set. This because most of their tweets were retweets or repeated. Note that all retweets or duplications were deleted when cleaning the BTAC corpus. Table 6.14 shows the codelength differences between training and classified labels for the rest of the four authors which PPM mis-classified.

PPMD 5	**Author1 Omran**	**Author2 Kareem**	**Author3 Saban**	**Author4 Meshari**
Training label	MSA	Iraqi	Mix	Gulf
2nd Training label	CA	Mix	Levantine	MSA
Classified label	CA	Gulf	MSA	MSA
CA	**19475.064**	5417.531	20494.516	22959.967
Egyptian	22794.061	4912.527	19130.512	23266.092
Gulf	22659.961	**4715.622**	19271.420	**23407.367**
Levantine	23053.801	4835.546	19132.461	24736.404
MSA	**19989.648**	5011.298	**18836.396**	**21314.678**
Maghrebi	23433.289	5058.061	20975.969	25037.535
Iraqi	24787.527	**4881.765**	21935.494	26231.883
Mix	21188.887	4773.661	**18861.479**	22069.014
Difference (bits)	**514.584**	**166.143**	**25.083**	**2092.689**

Table 6.14 Codelengths for the mis-classified authors.

The codelength differences among the first three authors (shown in the final row of the table) is quite low. For example, the codelength difference between classical and

modern style for the first author is 514.584 bits. In contrast, the fourth author is classified to the modern style which is the highest training subset in the corpus and therefore has a better contextual coverage compared to the other styles and dialects. After analysing this author's tweets, it was found that the codelength difference was quite high (2092.689 bits). This is due to this author having the second highest number of tweets written in the modern style. Furthermore, it was found that this particular author is a journalist, and this clarifies the classification results as the author used modern standard Arabic style in most of his tweets.

6.5.2 Improving author dialect identification

Following the classification improvement for gender categorisation in chapter 5, the experiment was repeated to see if better performance was possible using Protocol V. That is, instead of concatenating the training data according to the dialects comprising eight labels – Modern, Classic, Mix, Gulf, Egyptian, Levantine, Maghrebi and Iraqi – the training data would be concatenated according to authorship instead. This generated 101 models since 101 authors were collected for BTAC. Each author is labelled according to the most used dialect found from the training sets. Figure 6.2 demonstrates the author dialect identification procedure that uses Protocol V. This shows that the training process uses authorship files as a secondary class type. Note that the primary class (dialect) can be detected since it is available during the training process.

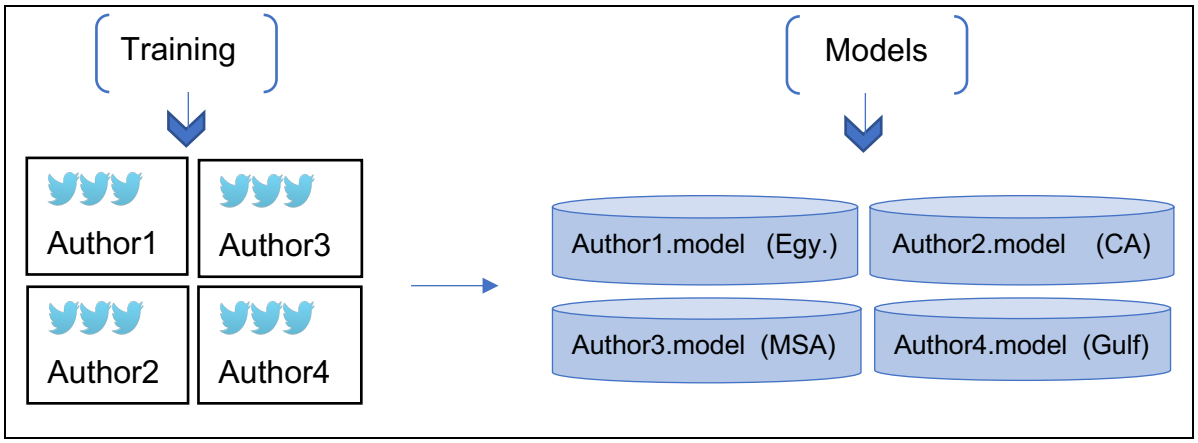

Figure 6.2 Dialect identification procedure for Protocol V.

The classification performed using orders 2 to 6 as these were shown to produce the best results on the dialect experiments in this chapter. The best result was obtained with order 2, achieving an accuracy of 98%, with two authors being mis-classified (see Table 6.15). The reason for the misclassification is that both authors had few tweets in their testing sets. This result reflects the results that were found for authorship attribution in Chapter 4, which obtained the highest accuracy with order 2. The result obtained confirms the ability of the system to capture the correct dialect with the aid of the author writing style. Orders 3 and 4 achieved the second best result with accuracy of 94.1%, with six authors being mis-classified. By using author-trained models, other factors besides dialect were considered in the identification process such as each author's possibly different writing styles (e.g., hash, comma, and dots) and interests (e.g. football, hobbies, politics) instead of relying heavily on the dialect most used by an author in the testing set as shown in section 6.6.1.

Orders	**Accuracy (%)**	**Recall**	**Precision**	**F-measure**
Order 2	**98.0**	0.983	0.966	0.974
Order 3	94.0	0.953	0.885	0.919
Order 4	94.0	0.953	0.885	0.919
Order 5	93.1	0.950	0.881	0.915
Order 6	92.1	0.944	0.869	0.906

Table 6.15 Dialect identification of Arabic authors using Protocol V.

The mis-classified authors from orders 2 and 3 were analysed. Table 6.16 shows that the difference in codelengths between the lowest and correct models is low. The average codelength differences average in order 2 is 20.56 bits for two authors, while the average codelength in order 3 is 29.81 bits in total for the six authors. This compares to the average codelength with the concatenated models produced in Table 6.14, which resulted in an average of 699.624 bits for the four authors. This shows that using concatenated author models also reduces the cost of compression, which also helps to accurately identify the dialect for each author.

	PPMD2		PPMD3					
Author	**Author1 MohAshr**	**Author2 Raghooda**	**Author3 Abouiss**	**Author4 Lababidi**	**Author5 rttoi**	**Author6 olZngna**	**Author7 Karamfdl**	**Author8 MohAsh**
Training label	Egy	Lev	CA	MSA	Iraqi	Iraqi	Iraqi	Egy
Classified to	CA	Gulf	MSA	CA	Gulf	Gulf	Egy	MSA
Lowest Codelength	455.83	589.14	1987.8	9849.8	680.15	2315.2	5288.6	410.44
Codelength for correct classification	469.54	616.55	1998.4	9935.6	683.03	2324.6	5319.5	449.88
Difference (bits)	**13.71**	**27.41**	**10.56**	**85.76**	**2.88**	**9.39**	**30.86**	**39.44**

Table 6.16 Codelengths for the mis-classified authors produced by concatenated author models using Protocol V.

6.6 Discussion and findings

Experimental results for a number of Arabic dialect identification experiments have been presented using prediction by partial matching (PPM) employing the character-based approach. These results have also been compared with those from other machine learning algorithms using character-based and word-based approaches. The results showed that the PPM classifier provided competitive performance.

The findings demonstrated the utility of the selected corpus BTAC for experiments for Arabic dialect identification. Single tweets identification achieved an accuracy of 74% and a F-measure of 0.630 for PPM. Over 6500 tweets were examined using the 7-way classification task with a total of 112 thousand annotated tweets as the training set. This dataset is more comprehensive compared to others used for Arabic dialect identification studies performed on single tweets. For instance, the study by Zaidan and Callison-Burch (2011) used a dataset size of over 108 thousand sentences. They reported an accuracy of 69.4% performed on a 4-way classification task. Also, the research by Abu Kwaik et al. (2018) yielded an accuracy of 52% performed on a 4-way classification task.

Although other researchers have produced better results than those from this study, for this study we trained and tested on a larger dataset. El Haj et al. (2018) performed their study using 16 thousand tweets. Their study achieved an accuracy of 76.2% and an F-measure of 0.78. They performed a 5-way classification task using SVM. Moreover, Sadat et al. (2014) performed Arabic dialect identification on 18 Arabic variations. A total of 63 thousand sentences were used for training data, with the testing set consisting of 100 sentences for each dialect. They reported an overall F-measure of 0.80 and an accuracy of 98% using the character bi-gram model. Finally, Malmasi et al. (2015) obtained similar results to this current study achieving an accuracy of 74% from a total dataset consisting of just two thousand sentences. The study by Alshutayri and Atwell (2017) reported 5-way classification accuracy of 79%. Their training data contained 8,090 tweets, and testing was done on 1,764 tweets.

In addition, this work has also investigated classification by dialects according to each author yielding an accuracy of 87%. When comparing this with other machine learning algorithms, MNB performed the best with an accuracy of 86%. In addition, the inaccurately classified authors were highlighted, and it was found that the mis-classification was due to either fewer tweets by the author or that some authors changed their style of writing in the testing set. We also found out that the classification of the text can be dominated by people's names or location or the tweet's topic.

Furthermore, the research results were improved by using concatenated author models (Protocol V). Instead of concatenating the models according to the available training text for each class (i.e. dialects) using Protocol I. The results improved to an overall accuracy of 98%. To the best of our knowledge, no other studies for Arabic have achieved such a good result using similar approaches.

In general, it was also found that character N-grams identify Arabic dialects best, similar to the results reported by (Darwish *et al.*, 2014; Sadat *et al.*, 2014). Specifically, we found that short sequence of characters such as orders 5, 6, and 7 capture the dialect features best in Arabic. This is in contrast to other experiments on multiple

tweets classification which reported that word-based approaches identified Arabic dialects best (Zaidan and Callison-Burch, 2014; Harrat *et al.*, 2017; Abu Kwaik *et al.*, 2018). However, the above-cited studies used machine learning algorithms which are known for their ability to perform well with word-based approaches.

6.7 Conclusion

In this chapter, Prediction by Partial Matching was employed to identify Arabic text for two scenarios: single tweets and multiple (author) tweets. The research achieved accuracies of 74% and 87% on both experiments using concatenated models. The results were also compared with the benchmark machine learning algorithms. In addition, various features such as character-based and word-based approaches were applied, and the incorrectly classified authors were assessed, in order to find improvements. It was also found that by using the new concatenated author models, the results of author dialect identification improved to an overall accuracy of 98%.

There are a number of possible directions for future work in this chapter. The accuracy can be improved by increasing the size of the training data for both the Maghrebi and Iraqi dialects. In this regard, the relatively high classification accuracy of the compression-based approach is reassuring, given the restricted amount of training data available. Furthermore, the generalizability of the system needs to be investigated with a much greater number of authors in the author dialect identification experiment in order to determine how well the system scales up with real case scenarios.

Chapter 7

Character-based Identification of Code-switching in Arabic Tweets

7.1. Introduction

The purpose of this chapter is to identify code-switching in Arabic Twitter text. Code-switching is a phenomenon that often occurs in text acquired from social media. This is because the use of informal written text increases the chance that there is a shift between two linguistic systems. Many researchers have proposed the need to identify code-switching that occurs between two main languages. However, in this chapter we investigate code-switching between two variant linguistics systems from one language. The motivation for this came from observing in BTAC code-switching occurrences in Twitter for the Arabic language, which usually took place between dialectical and modern standard Arabic content, and vice versa.

This chapter examines the code-switching dataset in BTAC and detects code-switching which occurs in both dialect and MSA content. Furthermore, for validation, a further experiment applying five-fold cross-validation was applied on the entire data of the corpus which did not contain code-switching content. Finally, we analysed the code-switching outcomes and demonstrate results samples. The remainder of the chapter is structured as follows: Section 7.2 describes the method used for character-based segmentation; section 7.3 explains the experimental setup for both the experiment and data used in the study; section 7.4 reports the experimental results; section 7.5 discusses the outcomes of the study and analyses sample of the results; and section 7.6 concludes the chapter.

7.2. Character-based segmentation using PPM

The method used to perform the segmentation is based on the character-based approach. Prediction by Partial Matching (PPM) is used to find the most likely true segmentation by switching between the compression-based language model used to encode each character. This is achieved by calculating the lowest compression codelength for the annotated character sequence. The searching process for finding the best switching uses the Viterbi-based algorithm which was implemented using the Tawa toolkit (Teahan, 2018).

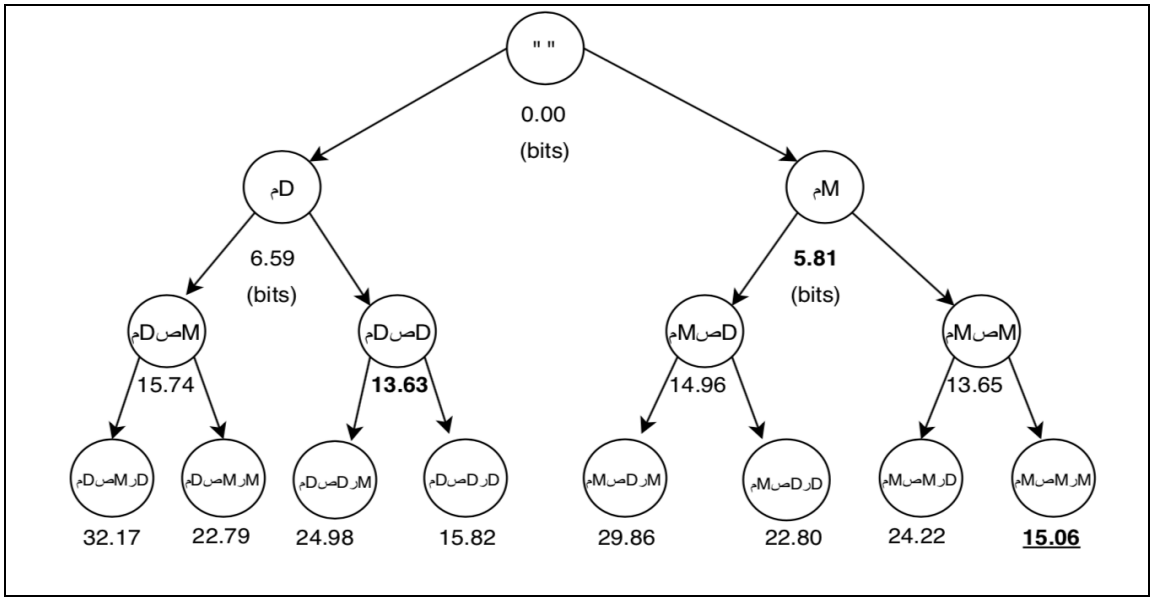

Figure 7.1 Search tree for the sample "مصر" (translation: “Egypt”).

The search process for the segmentation problem which shows how the search was applied is shown in Figure 7.1. Two language variations were used in this example – Modern Standard Arabic and Arabic Dialect. For each node, the transformed letter indicated which model was used to code it – the abbreviation ‘M’ if the Modern Standard Arabic model was used and the abbreviation ‘D’ if the Arabic Dialect model was used instead. If the process decided to shift from one model to the other, then a special sentinel symbols (like an end-of-file character) was encoded to signify the switch which added major costs to the encoding process. Under each node the codelength (i.e. cost of compression) is calculated based on a static order 5 PPMD

trained using the MA and DA datasets from BTAC (see the following section – experimental setup). The lowest codelength for each level of the tree is shown in bold font.

The example shows the process of segmentation of the word "مصر" (translation: “Egypt”). In the first level the lowest codelength for encoding the first letter, 5.81 bits, was assigned to the MA model. In the second level, the lowest codelength of 13.63 bits was assigned to the DA model. Finally, the third level returned back to the correct MA model as it received the lowest codelength in the third level of 15.06 bits. Note that the alternative segmentations which involved code-switching did not produce the best result in this short sequence of three characters. The Viterbi-based search algorithm effectively makes the search tractable for longer sequences by pruning poorly performing segmentations which use the same predictive contexts.

7.3. Experimental setup

In the BTAC corpus (see chapter 3), code-switching has been manually identified and extracted for this and further research. Code-switching occurs among less than 1% of the entire corpus. 713 tweets were identified to contain code-switching mostly between dialectal and MSA content. Each tweet is processed and tagged with labels according to the trained model (DA and MSA). Each tweet split is tagged with the corresponding model. Then the file is post-processed to count the number of characters that were correctly classified to calculate the accuracy. The accuracy is calculated according to the percentage of correct characters based on the ground-truth judgments made for the code-switching data in BTAC.

Code-switching dataset	
#Word	12,812
#Characters	61,426
#Testing tweets	713
#Training tweets	105,583

Table 7.1 Number of tweets, words, and characters for the code-switching dataset in BTAC.

7.4. Experimental Results

The majority of code-switching exists in BTAC between dialect and MSA content. The models were built using a training set comprised of dialectal Arabic and MSA subsets taken from non-code-switching data found in BTAC. The MSA subset combines both modern and classical Arabic text containing 73,663 tweets, whereas the dialect Arabic subsets combines content from the five main dialects in one model (Egyptian, Gulf, Iraqi, Maghrebi, and Levantine), containing 31,922 tweets. Segmentation was performed using order 5 PPMD models as this was the order which achieved the best accuracy in identifying dialect tweets previously.

The first experiment result reported in Table 7.2 involved using just the 713 tweets in BTAC that had been identified as containing a mixture of DA and MSA are examined. A total number of 327 tweets were identified as containing code-switching, whereas 386 tweets were identified as having no code-switching existing between them. The accuracy obtained for identifying code-switching in the tweets at a character level using the PPM-based method was 62.4%.

Data	**Results**
Number of tweets where single code-switching occurs	327
Number of tweets where multiple code-switching occurs	68
Number of correct characters	38,377
Number of correct tweets	446.1
Accuracy using order 5 PPMD models	62.4%

Table 7.2 The results for code-switching performed on 713 tweets.

Additionally, we performed another experiment for validation purposes using five-fold cross-validation. This time, the experiments included the entire corpus containing a total of 106,298 tweets including the code-switching dataset. Then the tweets are split into five folds, each fold consisted of 21,260 tweets. Similarly, a training set was used which comprised dialectal Arabic in one model, and combination of modern standard Arabic and classical Arabic in another model. Every fold is used once for testing while using the others folds for training. The experiment achieved an accuracy of 81.2%

when the full dataset was taken into consideration. Note that the segmentation method was able to identify most of the non-code-switching tweets correctly by identifying a single language variation for each tweet. The following table reports the experimental result.

Data	**Five-fold cross-validation 106k tweets**				
	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**
#Correct tweets	16,352	16,478	16,418	16,263	16,501
#Total character	1,729,467	1,732,827	1,736,996	1,738,553	1,725,308
#Correct character	1,403,759	1,412,557	1,413,560	1,394,846	1,413,412
Accuracy	81.1%	81.5%	81.3%	80.2%	81.9%
Avg. Accuracy of PPM order 5	81.2%				

Table 7.3 The results for code-switching performed using five-fold cross-validation.

The confusion matrix shows the total numbers of actual and predicted characters. (The “No” label refers when no code-switching occurred content, and the “Yes” label refers to the existence of code-switching content). Out of 8,601,725 characters for the tweets that contain no code-switching, the system predicted 81.4% of the total number of characters correctly. However, when using the dataset that contains code-switching which consists of 61,426 characters, the system predicted 52.8% of the total number of characters that contains code-switching correctly.

Confusion Matrix	**Predicted “No”**	**Predicted “Yes”**
Actual “No”	7,003,979	1,597,746
Actual “Yes”	28,964	32,462

Table 7.4 Confusion matrix for the code-switching experiment performed on the entire corpus at the character level using five-fold cross-validation.

Confusion Matrix	**Predicted “No”**	**Predicted “Yes”**
Actual “No”	82,465	23,488
Actual “Yes”	345	368

Table 7.5 Confusion matrix for the code-switching experiment performed on each tweet in the entire corpus using five-fold cross-validation.

7.5. Discussion of experimental results

As mentioned above, predicting code-switching in text acquired from Twitter is the key motivation of the research conducted for this chapter. Most of the studies mentioned in the related work section performed code-switching between two languages. For example, Yeong and Tan (2010) identified the switch between Malay-English vocabulary, Oco and Roxas (2012) examined code-switching that occurs between Tagalog-English languages, Lignos and Marcus (2013) detected code-switching between Spanish-English, and Piergallini et al. (2016) identified code-switching that occurs in Swahili-English languages. These studies reported higher results due to the clear differences between the languages.

Moreover, other studies performed code-switching between two linguistic variations of a single language. Elfardy et al. (2014) examined the code-switching that occurs between the Egyptian dialect and modern standard Arabic, and Shrestha (2016) detected code-switching that occurs between Arabic dialects and MSA. These studies produce very low accuracy due to the fact that the differences are not highly noticeable when identifying two linguistic variations from the same language.

However, the reported result in this research is an accuracy of 62.4% which shows that the accuracy of identifying code-switching is complicated. This result compares with other studies such as Elfardy et al. (2014) who examined the code-switching that occurs between the Egyptian dialect and modern standard Arabic, achieving an accuracy of 51.9%. Elfardy et al. (2014b) used a language modelling approach with back-off to a morphological analyser and reported an F-measure of 0.20 for Twitter data. Shrestha (2016) reported an F-measure of 0.34 for identifying code-switching using Conditional Random Fields (CRF).

Further analyses were applied to the segmented tweets. Most of the incorrectly segmented tweets fall into six erroneous cases:

- Case one: The number of characters of the MSA part of the tweet is larger than the DA part. This occurs in 18.3% of the tested tweets (131).

Original:

<MA> نظام الاقطاع بالاسلام، <DA><\MA> وحكو عنو كثير علماء، الشافعي والماوردي وغيرهن.. مم تشغيل فلاحين تحت نظام او سلطة معينة<\DA>

Segmented:

<MA> نظام الاقطاع بالاسلام، وحكو عنو كثير علماء، الشافعي والماوردي وغيرهن.. مم تش<\MA><DA>غيل فلاحين تحت نظام او سلطة معينة<\DA>

- Case two: The number of characters of the DA part of the tweets is larger than the MSA part. This occurs in 18% of the tested tweets (130).

  Original:

  <MA> ما أكثر أسئلة التحية عندنا <\MA><DA>!كيفك ؟ شلونك ؟ عساك طيب ؟ ... ولا جواب <\DA>

  Segmented:

  <MA> ما أكثر أ<\MA><DA>!سئلة التحية عندنا كيفك ؟ شلونك ؟ عساك طيب ؟ ... ولا جواب <\DA>

- Case three: One label was assigned to the tweets instead of segmentation. This occurs in 53% of the tested tweets (386).

  Original:

  <DA>لا ياخي غلط فيه حديث <MA>"لقنوا موتاكم لا إله الا الله"<\MA> وتراه مستحب تلقين المحتضر<\DA>

  Segmented:

  <DA>لا ياخي غلط فيه حديث "لقنوا موتاكم لا إله الا الله" وتراه مستحب تلقين المحتضر<\DA>

- Case four: Number of characters in both segmentations are equal. This occurs in a total of 66 tweets accounting for around 9% of the tested tweets.

  Original:

  <MA>اندلاع حريق ضخم داخل مصافي تكرير النفط بمدينة حيفا المحتلة<\MA><DA>تاني خبر حلو النهاردة <\DA>

  Segmented:

  <MA>اندلاع حريق ضخم داخل مصافي تكرير النفط بمدينة حيفا المحتلة<\MA><DA>تاني خبر حلو النهاردة <\DA>

- Case five: Tweets that are wrongly assigned to contain code-switching. This often occurred specifically in the second experiment. The total number of splits that occurs is 3212 tweets, accounting for around 3% of the entire corpus.

Original:

<DA> 🏚 يا بنتي البارح برشا توانسة يشجعوا في الجزائر .. هاذم أغبياء ولا شنو وضعهم؟ <\DA>

Segmented:

<DA> يا بنتي البارح برشا توانسة يشجعوا في الجزائر .. هاذم <\DA><MA>🏚 أغبياء ولا شنو وضعهم؟ <\MA>

- Case six: Tweets that are wrongly tagged with the opposite label. This often occurred specifically in the second experiment. The total number of tweets is 20,276 tweets, accounting for around 19% of the entire corpus.

  Original:

  <MA> من هو ؟!! أمير ملك ثلث مساحة إفريقيا ورفض #خلافة #المسلمين !! #شاهد<\MA>

  Segmented:

  <DA> من هو ؟!! أمير ملك ثلث مساحة إفريقيا ورفض #خلافة #المسلمين !! #شاهد<\DA>

- Case seven: Tweets that were correctly not segmented, which represents the majority of the tweets in the second experiment. The total number of tweets is 82,465 tweets, accounting for around 77% of the entire corpus. Table 7.6 summarises all the seven cases.

	Case	# Tweets	Percentage
Experiment 1	1	131	18%
	2	130	18%
	3	386	54%
	4	66	9%
Experiment 2	5	3,212	3%
	6	20,276	19%
	7	82,465	77%

Table 7.6 Summary of the six cases with number of tweets and percentage for both experiments.

7.6. Conclusion

In this chapter, Prediction by Partial Matching was employed to identify code-switching in Arabic Twitter text. Two experiments were employed in this chapter. First, a mixture of dialectal and MSA tweets were examined in this experiment. The accuracy obtained for identifying code-switching in these tweets was found to be 62.4%. Note that this result is produced from a small sample of 713 tweets. Second, for validation purposes, a further experiment using five-fold cross-validation was applied on the entire data of the corpus which did not contain code-switching content. The obtained accuracy for identifying code-switching in this experiment achieved 81.2%. Note that the method was able to identify most of the non-code-switching tweets correctly by identifying a single language variation for each tweet. Finally, we analysed the code-switching output and found that the incorrectly segmented tweets fell into six erroneous cases.

There are a number of possible directions for future work from this chapter. We plan to perform further analysis using other corpora that contain code-switching content. Also, we plan to extend our dialect dataset to include segmenting individual dialects instead of combining all the dialectal text together into one model. This would be required for the system to be used in the future for automatically annotating data.

Chapter 8

Conclusion and Future Work

8.1. Discussion

There were two motivations that inspired the work within this thesis. The first was that although there are a significant volume of classical and modern standard resources in the Arabic language community, the field of dialectal resources is still limited. Dialectal resources have recently witnessed major growth because of the availability of web-based resources in the form of textual data from social media websites, unlike in the past. This increase in the availability of dialectal text has provided the incentive to produce this work.

Therefore, a new corpus was created called the Bangor Twitter Arabic Corpus (BTAC). The corpus was planned to support various Arabic studies that depend on authentic data in addition to text analysis in areas such as dialect identification, code-switching and other classification tasks such as gender categorisation, authorship attribution, and genre categorisation. The corpus contains text from social media as a reference for Arabic dialects. The process involved the collection of over 122K tweets that were manually annotated according to the main five Arabic dialects – Egyptian, Gulf, Iraqi, Maghrebi, and Levantine, in addition to the two main Arabic styles – Classical Arabic and Modern Standard Arabic. The annotation also highlighted for further studies some of the tweets that contained code-switching. The corpus also involved genre annotation of each tweet such as information, politics, religion, art, sport, media, culture, economics, greeting, travel, social, and health. This corpus represents a valuable and rich resource for NLP applications targeting Arabic dialects research.

The second motivation was to perform text categorisation experiments in order to evaluate the PPM character-based approach and compare with the feature-based approach. The aim was to investigate Arabic Twitter text in three main classification tasks: gender categorisation; authorship attribution; and dialects identification. The potential for categorisation of text using the character-based approach has been underestimated in various studies (Frank *et al.*, 2000). This underestimation had been supported by published experimental results for the word-based approach adopted by many machine learning algorithms. In the past, various studies have employed the character-based approach and performed categorisation experiments for various tasks involving English and Chinese texts. However, to the best of our knowledge, no study involving Twitter text using PPM has been performed before this one involving specifically Arabic language.

In term of single tweets categorisation, the experimental results reported in this thesis that using PPM has produced an accuracy of 76% for gender categorisation, an accuracy of 77% for authorship attribution, and an accuracy of 74% for dialect identification. When compared to the machine learning classifiers where the best results were achieved by the LIBSVM classifier with an accuracy of 71% for gender categorisation, an accuracy of 63% for authorship attribution, and an accuracy of 72% for dialect identification.

In term of the author tweets categorisation, the experimental results reported in this thesis that using PPM has produced an accuracy of 88% for gender categorisation, an accuracy of 96% for authorship attribution, and an accuracy of 87% for dialect identification for Arabic Twitter text using PPM. In contrast, when compared to the machine learning classifiers, the best results were achieved by the MNB classifier with an accuracy of 78% for gender categorisation, an accuracy of 93% for authorship attribution and an accuracy of 86% for dialect identification.

Further optimisation was achieved for gender and dialect experiments using a classification procedure that used secondary class authorship information. The procedure concatenated all the data for each author as the training set which produces

separate models for each author. This improved classification accuracy in both gender and dialect experiments. The results significantly improved with gender author categorisation achieving an accuracy of 97%, and dialect author identification achieving an accuracy of 98%.

We also investigated code-switching that often occurs in text acquired from social media. In this study we investigated code-switching between two variant linguistic systems from one language (Modern Standard Arabic and Arabic dialectal text). The purpose of the experiment was to detect the shift at the character level. An accuracy of 81.2% for detecting code-switching was obtained using 5-fold cross-validation on the full BTAC dataset.

8.2. Review of aim and objectives

The aim and objectives of this thesis summarised in section 1.2 have been effectively accomplished in this research. The character-based approach using PPM has been successfully applied to the real-world problem of text categorisation in the Arabic language. The results have been compared with well-known machine learning algorithms.

The following lists summarises the achieved objectives:

- *Create a dialectal corpus for Arabic language using Twitter text.*

This objective was achieved as described in chapter 3 by creating the BTAC which contains over 122K tweets manually annotated according to the five main Arabic dialects.

- *Apply character-based approach based on compression using Prediction by Partial Matching (PPM) for Arabic twitter text categorisation.*

This objective was accomplished as described in chapters 4, 5, 6, and 7, with an accuracy of 88% obtained for gender categorisation, an accuracy of 96% for authorship attribution, and an accuracy of 87% for dialect identification.

- *Evaluate the effectiveness of the method used (PPM) and improve the results.*

The results were improved using models trained on data organised by secondary class type (authorship) rather than the primary class type (gender or dialect) achieving an accuracy of 97% for gender categorisation, and an accuracy of 98% for dialect identification.

- *Explore text categorisation using single tweet vs multiple tweets.*

Experimenting using a multiple tweets dataset resulted in significantly better accuracy than the performance using the single tweets dataset. However, the results improved from 76% to 88% for gender categorisation, from 77% to 96% for authorship attribution, and from 74% to 87% for dialect identification.

- *Investigate which order of PPM is best for Arabic text.*

We investigated which order was best order for Arabic text categorisation. We found that the best order varied in each experiment that we performed. For instance, orders 7-13 performed best for gender categorisation. The reason for this is that each Arabic letter is represented by two orders of PPM as the compression is performed at the byte level where each Arabic letter is allocated two bytes. This means that order 12 in PPM represent a total of six Arabic letters. This is similar to other researchers who found that longer sequences of characters identify gender best (Deitrick *et al.*, 2012b, 2012a; Mikros, 2012). Furthermore, for dialects identification, orders 5, 6 and 7 performed best in identifying dialects tweets, similar to other studies that reported similar findings using Arabic (Darwish *et al.*, 2014; Sadat *et al.*, 2014).

In contrast, order 2 performed best for authorship attribution. This was expected as low sequences of characters best identify the author's fingerprint which corresponds to other work with similar findings using Greek text (Mikros, 2012).

8.3. Review of research question

The main research question in this thesis has been addressed:

How does the effectiveness of the character-based compression approach for text categorisation using Prediction by Partial Matching compare to that of commonly used machine learning algorithms in classifying Arabic Twitter text according to gender, authorship, and dialects?

As shown in the experiments of chapters 4, 5, 6, and 7, character-based compression using the Prediction by Partial Matching classifier has successfully been applied to the problem of Arabic Twitter text categorisation. The PPM classifier performs better than other traditional word-based classifiers (Multinomial Naïve Bayes, Support Vector Machine, and k-nearest neighbors) in precision, recall and F-measure at various categorisation tasks on Arabic: authorship attribution; gender categorisation; and dialect identification.

8.4. Limitations of the work

We have encountered a few limitations while performing the experiments. Some limitations are discussed below:

- The classification for some tweets is affected by the topic. For instance, when classifying authors according to gender, there are some topics mostly tweeted by males such as religion and sport. This affects the gender classification when female authors tweet about topics which are mostly covered by males.
- In the dialect identification experiment, the modern standard Arabic class often dominates other classes. This was seen with a few tweets where the classification was assigned to the wrong class as shown in the confusion matrix for single tweets dialect identification (see Table 6.6 in section 6.4).
- There was limited availability of training text for certain dialects in Twitter such as Iraqi and Maghrebi dialects as Twitter is less popular social media platform in some cultures.

8.5. Future work

There are a number of potential directions for future work in this thesis as follows:

- On BTAC, we are looking to expand the size of the corpus even further. We would also like to collaborate with other Arabic researchers to improve the resources to the research community. One idea is to use our corpus as ground truth data and apply it to help annotate much larger new corpora according to dialects, gender, or genre. This would be the first step before manually annotating the corpus.
- PPM produces superior results at identifying author related features. Various authors' information can be predicted such as gender, dialect, and authorship. However, further investigation is needed on different fields aside from authorship analyses. This could involve classification tasks such as genre classification. Furthermore, the system can be developed to predict ongoing criminal activities in social media. For instance, theft, online sexual harassment, piracy, illegal trading, cyber stalking, extreme religious, and extreme feminism.
- Comparison with other classification methods need to be investigated such as: neural networks and deep learning algorithms.
- The generalizability of the system needs to be investigated with a much greater number of authors in the author dialect identification experiment in order to determine how well the system scales up with real case scenarios.
- Further analysis using machine learning algorithms is needed to examine how well they perform at identifying code-switching content.
- Further processing of the text involving morphological analyses or parts-of-speech tagging may yield improved results for the classification or segmentation tasks.

References

Abbasi, A. and Chen, H. (2005a) ‘Applying Authorship Analysis to Arabic Web Content’, in *International Conference on Intelligence and Security Informatics*. Springer, pp. 183–197.

Abbasi, A. and Chen, H. (2005b) ‘Applying Authorship Analysis to Extremist-Group Web Forum Messages’, *IEEE Intelligent Systems*. IEEE, 20(5), pp. 67–75.

Abbasi, A. and Chen, H. (2006) ‘Visualizing Authorship for Identification’, in *International Conference on Intelligence and Security Informatics*. Springer, pp. 60–71.

Abdelali, A., Cowie, J. and Soliman, H. (2005) ‘Building a Modern Standard Arabic Corpus’, in *Workshop on Computational Modeling of Lexical Acquisition*, pp. 25–28.

Abu Kwaik, K., Saad, M. K., Chatzikyriakidis, S. and Dobnik, S. (2018) ‘Shami: A Corpus of Levantine Arabic Dialects’, in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. European Language Resources Association (ELRA).

Al-Rowais, H. H. (2012) ‘Code Switching Between Arabic and English: Social Motivations and Structural Constraints’, *Masters Thesis, Ball State University*.

Al-Sulaiti, L. and Atwell, E. S. (2006) ‘The Design of a Corpus of Contemporary Arabic’, *International Journal of Corpus Linguistics*. John Benjamins Publishing Company, 11(2), pp. 135–171.

Al-Thubaity, A. O. (2015) ‘A 700M+ Arabic corpus: KACST Arabic Corpus Design and Construction’, *Language Resources and Evaluation*. Springer, 49(3), pp. 721–751.

Alansary, S., Nagi, M. and Adly, N. (2007) ‘Building an International Corpus of Arabic (ICA): Progress of Compilation Stage’, in *7th International Conference on Language Engineering, Cairo, Egypt*, pp. 5–6.

Albadarneh, J., Talafha, B., Al-Ayyoub, M., Zaqaibeh, B., Al-Smadi, M., Jararweh, Y.

and Benkhelifa, E. (2015) ‘Using Big Data Analytics for Authorship Authentication of Arabic Tweets’, in *Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on*. IEEE, pp. 448–452.

Alhawiti, K. M. (2014) *Adaptive Models of Arabic Text*. Ph.D thesis, Bangor University.

Aljehane, N. O. M. (2018) *Grammar-based Preprocessing for PPM Compression and Classification*. Ph.D thesis, Bangor University.

Alkahtani, S. and Teahan, W. J. (2016) ‘A New Parallel Corpus of Arabic/English’, in *Proceedings of the Eighth Saudi Students Conference in the UK*. World Scientific, pp. 279–284.

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B. and Kochut, K. (2017) ‘A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques’, *arXiv preprint arXiv:1707.02919*.

Almeman, K. and Lee, M. (2013) ‘Automatic Building of Arabic Multi Dialect Text Corpora by Bootstrapping Dialect Words’, in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*. IEEE, pp. 1–6.

Alotaiby, F., Alkharashi, I. and Foda, S. (2009) ‘Processing large Arabic text corpora: Preliminary analysis and results’, in *Proceedings of the second international conference on Arabic language resources and tools*. Citeseer, pp. 78–82.

Alrabiah, M., Al-Salman, A. and Atwell, E. S. (2013) ‘The Design and Construction of the 50 Million Words KSUCCA’, in *Proceedings of WACL’2 Second Workshop on Arabic Corpus Linguistics*. The University of Leeds, pp. 5–8.

Alrefaie, M. (2016) *Arabic Stop Words*. Available at: https://github.com/mohataher/arabic-stop-words/blob/master/list.txt (Accessed: 20 April 2018).

Alrifai, K., Rebdawi, G. and Ghneim, N. (2017) ‘Arabic Tweeps Gender and Dialect Prediction.’, in *Conference and Labs of the Evaluation Forum (CLEF) (Working Notes)*.

Alsaleem, S. (2011) ‘Automated Arabic Text Categorization Using Support Vector Machine and Naïve Bayes’, *International Arab Journal of E-Technology*, 2(2), pp. 124–128.

Alsarsour, I., Mohamed, E., Suwaileh, R. and Elsayed, T. (2018) ‘DART: A Large Dataset of Dialectal Arabic Tweets’, in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association.

Alshutayri, A. and Atwell, E. (2018) ‘Creating an Arabic Dialect Text Corpus by Exploring Twitter, Facebook, and Online Newspapers’, in *Proceedings of OSACT’2018 Open-Source Arabic Corpora and Processing Tools*. European Language Resources Association.

Alshutayri, A., Atwell, E., Alosaimy, A., Dickins, J., Ingleby, M. and Watson, J. (2016) ‘Arabic Language WEKA-Based Dialect Classifier for Arabic Automatic Speech Recognition Transcripts’, in *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2016)*. COLING 2016, pp. 204–211.

Alshutayri, A. O. O. and Atwell, E. (2017) ‘Exploring Twitter as a Source of an Arabic Dialect Corpus’, *International Journal of Computational Linguistics (IJCL)*. CSC Journals, 8(2), pp. 37–44.

Alsmearat, K., Al-Ayyoub, M. and Al-Shalabi, R. (2014) ‘An Extensive Study of The Bag-of-words Approach for Gender Identification of Arabic Articles’, in *Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on*. IEEE, pp. 601–608.

Alsmearat, K., Shehab, M., Al-Ayyoub, M., Al-Shalabi, R. and Kanaan, G. (2015) ‘Emotion Analysis of Arabic Articles and Its Impact on Identifying the Author’s Gender’, in *Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of*. IEEE, pp. 1–6.

AlSukhni, E. and Alequr, Q. (2016) ‘Investigating the Use of Machine Learning

Algorithms in Detecting gender of the Arabic Tweet Author', *International Journal of Advanced Computer Science and Applications*. The Science and Information (SAI), 7(7), pp. 319–328.

Altakrori, M. H., Iqbal, F., Fung, B., Ding, S. H. H. and Tubaishat, A. (2018) 'Arabic Authorship Attribution: An Extensive Study on Twitter Posts', *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*. ACM, 18(1), p. 5.

Altamimi, M., Alruwaili, O. and Teahan, W. J. (2018) 'BTAC: A Twitter Corpus for Arabic Dialect Identification', in *the 6th Conference on Computer-Mediated Communication (CMC) and Social Media Corpora (CMC-corpora 2018)*, p. 5.

Altamimi, M. and Teahan, W. J. (2017) 'Gender And Authorship Categorisation Of Arabic Text From Twitter Using PPM', *International Journal of Computational Science and Information Technology (IJCSIT)*, 9(2), pp. 131–149.

Altamimi, M. and Teahan, W. J. (2019) 'Arabic Dialect Identification of Twitter text Using PPM Compression', *International Journal of Computational Linguistics (IJCL)*, 10(4), pp. 47–59.

Altheneyan, A. S. and Menai, M. E. B. (2014) 'Naïve Bayes Classifiers for Authorship Attribution of Arabic Texts', *Journal of King Saud University-Computer and Information Sciences*. Elsevier, 26(4), pp. 473–484.

Altman, N. S. (1992) 'An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression', *The American Statistician*. Taylor & Francis, 46(3), pp. 175–185.

Alwajeeh, A., Al-Ayyoub, M. and Hmeidi, I. (2014) 'On Authorship Authentication of Arabic Articles', in *Information and Communication Systems (ICICS), 2014 5th International Conference on*. IEEE, pp. 1–6.

Argamon, S., Koppel, M., Fine, J. and Shimoni, A. R. (2003) 'Gender, Genre, and Writing Style in Formal Written Texts', *Text The Hague Then Amsterdam Then Berlin*. Walter De Gruyter & Co, 23(3), pp. 321–346.

Argamon, S. and Levitan, S. (2005) ‘Measuring the Usefulness of Function Words for Authorship Attribution’, in *Proceedings of the 2005 ACH/ALLC Conference*.

Ayedh, A., Tan, G., Alwesabi, K. and Rajeh, H. (2016) ‘The Effect of Preprocessing on Arabic Document Categorization’, *Algorithms*. Multidisciplinary Digital Publishing Institute, 9(2), p. 27.

Baayen, H., Van Halteren, H. and Tweedie, F. (1996) ‘Outside the Cave of Shadows: Using Syntactic Annotation to Enhance Authorship Attribution’, *Literary and Linguistic Computing*. Oxford University Press, 11(3), pp. 121–132.

Bailey, C. J. N. (1968) ‘Is There a" Midland" Dialect of American English?’, in. ERIC Clearinghouse.

Bakhsh, S. (2015) *Saudi Con Artists Use Photos of Sick American Girl to Solicit Donations*, *BBC News*. Available at: Saudi con artists use photos of sick American girl to solicit donations (Accessed: 8 August 2018).

BBC (2017) *Massive Bot Networks Found on Twitter*, *BBC News*. Available at: https://www.bbc.co.uk/news/technology-38724082 (Accessed: 1 May 2018).

Belinkov, Y., Magidow, A., Romanov, M., Shmidman, A. and Koppel, M. (2016) ‘Shamela: a Large-Scale Historical Arabic Corpus’, *arXiv preprint arXiv:1612.08989*.

Bell, T., Witten, I. H. and Cleary, J. G. (1989) ‘Modeling for Text Compression’, *ACM Computing Surveys*, 21(4), pp. 557–591. doi: 10.1145/76894.76896.

Benedetto, D., Caglioti, E. and Loreto, V. (2002) ‘Language Trees and Zipping’, *Physical Review Letters*. APS, 88(4), p. 48702.

Binongo, J. N. G. (2003) ‘Who Wrote the 15th Book of Oz? An Application of Multivariate analysis to Authorship Attribution’, *Chance*. Taylor & Francis Group, 16(2), pp. 9–17.

Bobicev, V. (2007) ‘Comparison of Word-based and Letter-based Text Classification’, *Recent Advances in Natural Language Processing V.* Amsterdam: John Benhamins

Publishing Company, 7, pp. 76–80.

Boser, B. E., Guyon, I. M. and Vapnik, V. N. (1992) ‘A Training Algorithm for Optimal Margin Classifiers’, in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, pp. 144–152.

Bouamor, H., Habash, N. and Oflazer, K. (2014) ‘A Multidialectal Parallel Corpus of Arabic.’, in *Proceedings of the Ninth International Conference on Language Resources and Evaluation*. European Language Resources Association, pp. 1240–1245.

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., Obeid, O., Khalifa, S., Eryani, F. and Erdmann, A. (2018) ‘The MADAR Arabic dialect corpus and lexicon’, in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Bouamor, H., Hassan, S. and Habash, N. (2019) ‘The MADAR shared task on Arabic fine-grained dialect identification’, in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 199–207.

Bradley, P. H. (1981) ‘The Folk-Linguistics of Women’s speech: An Empirical Examination’, *Communications Monographs*. Taylor & Francis Group, 48(1), pp. 73–90.

Bratko, A. (2012) ‘Text Mining Using Data Compression Models’. Ph.D thesis, University of Ljubljana.

Bratko, A., Cormack, G. V, Filipič, B., Lynam, T. R. and Zupan, B. (2006) ‘Spam Filtering Using Statistical Data Compression Models’, *Journal of Machine Learning Research*, 7(12), pp. 2673–2698.

Brinegar, C. S. (1963) ‘Mark Twain and the Quintus Curtius Snodgrass letters: A statistical test of authorship’, *Journal of the American Statistical Association*. Taylor & Francis, 58(301), pp. 85–96.

Burger, J. D., Henderson, J., Kim, G. and Zarrella, G. (2011) ‘Discriminating Gender on Twitter’, *Proceedings of the Conference on Empirical Methods in Natural Language*

Processing. Association for Computational Linguistics, pp. 1301–1309.

Burrows, M. and Wheeler, D. J. (1994) 'A Block-Sorting Lossless Data Compression Algorithm'.

Cavnar, W. B., Trenkle, J. M. and Mi, A. A. (1994) 'N-Gram-Based Text Categorization', *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 161–175.

Chambers, J. K. and Trudgill, P. (1998) *Dialectology*. Cambridge University Press.

Chang, C.-C. and Lin, C.-J. (2011) 'LIBSVM: a Library for Support Vector Machines', *ACM Transactions on Intelligent Systems and Technology (TIST)*. ACM, 2(3), p. 27.

Chaski, C. E. (2005) 'Who's at the keyboard? Authorship Attribution in Digital Evidence Investigations', *International Journal of Digital Evidence*, 4(1), pp. 1–13.

Cheng, N., Chandramouli, R. and Subbalakshmi, K. P. (2011) 'Author Gender Identification From Text', *Digital Investigation*. Elsevier, 8(1), pp. 78–88.

Cheng, N., Chen, X., Chandramouli, R. and Subbalakshmi, K. P. (2009) 'Gender Identification From E-mails.', *IEEE Symposium on Computational Intelligence and Data Mining*. IEEE, 9, pp. 154–158.

Cleary, J. G. and Teahan, W. J. (1997) 'Unbounded Length Contexts for PPM', *The Computer Journal*. IEEE, 40(2 and 3), pp. 67–75.

Cleary, J. and Witten, I. (1984) 'Data Compression Using Adaptive Coding and Partial String Matching', *IEEE Transactions on Communications*. IEEE, 32(4), pp. 396–402.

Çoban, Ö., Özyer, B. and Özyer, G. T. (2015) 'A Comparison of Similarity Metrics for Sentiment Analysis on Turkish Twitter Feeds', in *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*. IEEE, pp. 333–338.

Cohen, J. (1960) 'A Coefficient of Agreement for Nominal Scales', *Educational and Psychological Measurement*. Sage Publications Sage CA: Thousand Oaks, CA, 20(1), pp. 37–46.

Contributors, W. (2018) *Suicide of Megan Meier*, *Wikipedia, The Free Encyclopedia.* Available at: https://en.wikipedia.org/w/index.php?title=Suicide_of_Megan_Meier&old

id=893886481 (Accessed: 4 November 2018).

Cormack, G. V and Horspool, R. N. S. (1987) 'Data Compression Using Dynamic Markov Modelling', *The Computer Journal*. The British Computer Society, 30(6), pp. 541–550.

Cortes, C. and Vapnik, V. (1995) 'Support-Vector Networks', *Machine learning*. Springer, 20(3), pp. 273–297.

Coyotl-Morales, R. M., Villaseñor-Pineda, L., Montes-y-Gómez, M. and Rosso, P. (2006) 'Authorship Attribution Using Word Sequences', in *Iberoamerican Congress on Pattern Recognition*. Springer, pp. 844–853.

Darwish, K., Sajjad, H. and Mubarak, H. (2014) 'Verifiably Effective Arabic Dialect Identification', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 1465–1468.

Davis, L. M. and Houck, C. L. (1992) 'Is There a Midland Dialect Area?—Again', *American Speech*. JSTOR, pp. 61–70.

Deitrick, W., Miller, Z., Valyou, B., Dickinson, B., Munson, T. and Hu, W. (2012a) 'Author Gender Prediction in an Email Stream Using Neural Networks', *Journal of Intelligent Learning Systems and Applications*. Scientific Research Publishing, 4(03), p. 169.

Deitrick, W., Miller, Z., Valyou, B., Dickinson, B., Munson, T. and Hu, W. (2012b) 'Gender Identification on Twitter Using the Modified Balanced Winnow', *Communications and Network*. Scientific Research Publishing, 4(3), pp. 189–195.

Diab, M., Habash, N., Rambow, O., Altantawy, M. and Benajiba, Y. (2010) 'COLABA: Arabic Dialect Annotation and Processing', in *The Language Resources and Evaluation workshop on semitic language processing*, pp. 66–74.

Doyle, J. and Keselj, V. (2005) ‘Automatic Categorization of Author Gender via N-gram Analysis’, in *The 6th Symposium on Natural Language Processing, SNLP*, pp. 1–5.

Duwairi, R. M. (2006) ‘Machine Learning for Arabic Text Categorization’, *Journal of the American Society for Information Science and Technology*. Wiley Online Library, 57(8), pp. 1005–1010.

Eckert, P. (1997) ‘Age as a Sociolinguistic Variable’, *The Handbook of Sociolinguistics*. Wiley Online Library, pp. 151–167.

El-Haj, M. and Koulali, R. (2013) ‘KALIMAT a multipurpose Arabic Corpus’, in *Second Workshop on Arabic Corpus Linguistics (WACL-2)*, pp. 22–25.

El-Haj, M., Kruschwitz, U. and Fox, C. (2010) ‘Using Mechanical Turk to Create a Corpus of Arabic summaries’. European Language Resources Association.

Elfardy, H., Al-Badrashiny, M. and Diab, M. (2014a) ‘A Hybrid System for Code Switch Point Detection in Informal Arabic Text’, *XRDS: Crossroads, The ACM Magazine for Students*. ACM, 21(1), pp. 52–57.

Elfardy, H., Al-Badrashiny, M. and Diab, M. (2014b) ‘AIDA: Identifying Code Switching in Informal Arabic Text’, in *Proceedings of The First Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics, pp. 94–101.

Elfardy, H. and Diab, M. (2013) ‘Sentence Level Dialect Identification in Arabic’, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2, pp. 456–461.

Ennaji, M., Makhoukh, A., Es-saiydi, H., Moubtassime, M. and Slaoui, S. (2004) ‘A Grammar of Moroccan Arabic’. Faculté des Lettres et des Sciences Humaines, Université Sidi Mohamed Ben Abdellah.

Enron (2005) *Enron E-mail Dataset*. Available at: http://www.cs.cmu.edu/~enron/ (Accessed: 18 March 2019).

Estival, D., Gaustad, T., Pham, S. B., Radford, W. and Hutchinson, B. (2007) 'Tat: an Author Profiling Tool with Application to Arabic Emails', in *Proceedings of the Australasian Language Technology Workshop 2007*, pp. 21–30.

Etman, A. and Beex, A. A. L. (2015) 'Language and Dialect Identification: A Survey', in *SAI Intelligent Systems Conference (IntelliSys), 2015*. IEEE, pp. 220–231.

Eyheramendy, S., Lewis, D. D. and Madigan, D. (2003) 'On the Naïve Bayes Model for Text categorization'.

Frank, E., Chui, C. and Witten, I. H. (2000) 'Text Categorization Using Compression Models'. University of Waikato, Department of Computer Science.

Fung, G. (2003) 'The Disputed Federalist Papers: Support Vector Machine Feature Selection via Concave Minimization', in *Proceedings of the 2003 Conference on Diversity in Computing*. ACM, pp. 42–46.

Gadalla, H., Kilany, H., Arram, H., Yacoub, A., El-Habashi, A., Shalaby, A., Karins, K., Rowson, E., MacIntyre, R. and Kingsbury, P. (1997) 'CALLHOME Egyptian Arabic Transcripts', *Linguistic Data Consortium, Philadelphia*.

Gharib, T. F., Habib, M. B. and Fayed, Z. T. (2009) 'Arabic Text Classification Using Support Vector Machines.', *International Journal of Computer Applications*, 16(4), pp. 192–199.

Goodman, J. (2002) 'Extended Comment on Language Trees and Zipping', *arXiv preprint cond-mat/0202383*.

Graff, D. (2003) 'Arabic Gigaword Corpus', *Philadelphia: Linguistic Data Consortium*.

Granger, S. (1993) 'English Language Corpora: Design, Analysis and Exploitation', *The International Corpus of Learner English. In J. Aarts, P. de Haan, & N. Oostdijk (Eds.)*. Rodopi, p. (pp. 57–69).

Habash, N. and Roth, R. M. (2009) 'Catib: The Columbia Arabic treebank', in *Proceedings of the ACL-IJCNLP 2009 conference short papers*. Association for

Computational Linguistics, pp. 221–224.

El Haj, M., Rayson, P. E. and Aboelezz, M. (2018) 'Arabic Dialect Identification in the Context of Bivalency and Code-Switching', *Proceedings of the 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan.* European Language Resources Association.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009) 'The WEKA Data Mining Software: an Update', *ACM SIGKDD Explorations Newsletter*. ACM, 11(1), pp. 10–18.

Hamdi, A., Nasr, A., Habash, N. and Gala, N. (2015) 'POS-Tagging of Tunisian Dialect Using Standard Arabic Resources and Tools', in *Workshop on Arabic Natural Language Processing*. Association for Computational Linguistics, pp. 59–68.

Han, J., Pei, J. and Kamber, M. (2011) *Data Mining: Concepts and Techniques*. Elsevier.

Harrat, S., Meftouh, K., Abbas, M., Jamoussi, S., Saad, M. and Smaili, K. (2015) 'Cross-dialectal arabic processing', in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 620–632.

Harrat, S., Meftouh, K., Abbas, M. and Smaili, K. (2014) 'Building Resources for Algerian Arabic Dialects', in *15th Annual Conference of the International Communication Association Interspeech*.

Harrat, S., Meftouh, K. and Smaili, K. (2017) 'Creating Parallel Arabic Dialect Corpus: Pitfalls to Avoid', in *18th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING)*.

Herring, S. C. (1996) *Computer-Mediated Communication: Linguistic, Social, and Cross-Cultural Perspectives*. John Benjamins Publishing.

Holes, C. (2004) *Modern Arabic: Structures, Functions, and Varieties*. Georgetown University Press.

Holmes, D. I., Gordon, L. J. and Wilson, C. (2001) 'A Widow and Her Soldier: Stylometry and the American Civil War', *Literary and Linguistic Computing*. Oxford University Press, 16(4), pp. 403–420.

Hoorn, J. F., Frank, S. L., Kowalczyk, W. and van Der Ham, F. (1999) 'Neural Network Identification of Poets Using Letter Sequences', *Literary and Linguistic Computing*. Oxford University Press, 14(3), pp. 311–338.

Howard, P. G. (1993) 'The Design and Analysis of Efficient Lossless Data Compression Systems'.

Hunnisett, D. S. and Teahan, W. J. (2004) 'Context-based Methods for Text Categorisation', in *Proceedings of the 27th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pp. 578–579.

Jain, A., Huang, J. and Fang, S. (2005) 'Gender Identification Using Frontal Facial Images', in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, pp. 4-pp.

Jarrar, M., Habash, N., Alrimawi, F., Akra, D. and Zalmout, N. (2017) 'Curras: an Annotated Corpus for the Palestinian Arabic Dialect', *Language Resources and Evaluation*. Springer, 51(3), pp. 745–775.

Jiang, L., Cai, Z., Wang, D. and Jiang, S. (2007) 'Survey of Improving K-Nearest-Neighbor for Classification', in *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*. IEEE, pp. 679–683.

Juola, P. (2006) 'Authorship Attribution', *Foundations and Trends in Information Retrieval*. Now Publishers Inc., 1(3), pp. 233–334.

Katakis, I., Tsoumakas, G. and Vlahavas, I. (2008) 'Multilabel Text Classification for Automated Tag Suggestion', in *Proceedings of the ECML/PKDD*.

Kennedy, G. (2014) *An Introduction to Corpus Linguistics*. Routledge.

Kennedy, J. F. (1962) ‘We Choose to Go to the Moon’, *speech presented at Rice University on the Nation’s Space Effort in Rice University, Houston*.

Kessler, B. (1995) ‘Computational Dialectology in Irish Gaelic’, in *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*. Morgan Kaufmann Publishers Inc., pp. 60–66.

Khalifa, S., Habash, N., Abdulrahim, D. and Hassan, S. (2016) ‘A large Scale Corpus of Gulf Arabic’, *arXiv preprint arXiv:1609.02960*.

Khmelev, D. V and Teahan, W. J. (2003) ‘A Repetition Based Measure for Verification of Text Collections and for Text Categorization’, in *Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in Informaion Retrieval*. ACM, pp. 104–110.

Khmelev, D. V and Tweedie, F. J. (2001) ‘Using Markov Chains for Identification of Writer’, *Literary and Linguistic Computing*. Oxford University Press, 16(3), pp. 299–307.

Khoja, S. (2001) ‘APT: Arabic Part-of-Speech Tagger’, in *Proceedings of the Student Workshop at NAACL*, pp. 20–25.

King, M. L. (1963) ‘I Have a Dream’, *Speech. Lincoln Memorial*. Washington, D. C.

Kivinen, J. and Warmuth, M. K. (1996) ‘Additive Versus Exponentiated Gradient Updates for Linear Prediction’, in *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*. ACM, pp. 209–218.

Knerr, S., Personnaz, L. and Dreyfus, G. (1990) ‘Single-layer Learning Revisited: a Stepwise Procedure for Building and Training a Neural Network’, in *Neurocomputing*. Springer, pp. 41–50.

Knight, K. (1999) ‘Mining online text’, *Communications of the ACM*, 42(11), pp. 58–61.

Koppel, M., Argamon, S. and Shimoni, A. R. (2002) ‘Automatically Categorizing Written Texts by Author Gender’, *Literary and linguistic computing*. Oxford University

Press, 17(4), pp. 401–412.

Koppel, M., Schler, J. and Argamon, S. (2009) ‘Computational Methods in Authorship Attribution’, *Journal of the American Society for information Science and Technology*. Wiley Online Library, 60(1), pp. 9–26.

Kulkarni, A. (2016) *TED-Multilingual-Parallel-Corpus*. Available at: https://github.com/ajinkyakulkarni14/TED-Multilingual-Parallel-Corpus/tree/master/Bilingual_Parallel_Corpus (Accessed: 10 November 2018).

Kumar, R., Lahiri, B., Alok, D., Ojha, A. K., Jain, M., Basit, A. and Dawer, Y. (2018) ‘Automatic Identification of Closely-related Indian Languages: Resources and Experiments’, *arXiv preprint arXiv:1803.09405*.

Labov, W. (1972) *Sociolinguistic Patterns*. University of Pennsylvania Press.

Labov, W. (1990) ‘The Intersection of Sex and Social Class in the Course of Linguistic Change’, *Language variation and change*. Cambridge University Press, 2(2), pp. 205–254.

Lakoff, R. and Lakoff, R. T. (2004) *Language and Woman’s Place: Text and Commentaries*. Oxford University Press, USA.

Landis, J. R. and Koch, G. G. (1977) ‘The Measurement of Observer Agreement for Categorical Data’, *Biometrics*. JSTOR, pp. 159–174.

Lewis, D. D., Yang, Y., Rose, T. G. and Li, F. (2004) ‘Rcv1: A New Benchmark Collection for Text Categorization Research’, *Journal of Machine Learning Research*, 5(Apr), pp. 361–397.

Lignos, C. and Marcus, M. (2013) ‘Toward Web-scale Analysis of Codeswitching’, in *Proceedings of annual meeting of the Linguistic Society of America*.

Littlestone, N. (1988) ‘Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm’, *Machine Learning*. Springer, 2(4), pp. 285–318.

Liu, G. A. and Hansen, J. H. L. (2011) ‘A Systematic Strategy for Robust Automatic

Dialect Identification', in *Signal Processing Conference, 2011 19th European*. IEEE, pp. 2138–2141.

Liu, W. and Ruths, D. (2013) 'What's in a Name? Using First Names as Features for Gender Inference in Twitter.', in *the Association for the Advancement of Artificial Intelligence In Spring Symposium: Analyzing Microtext*, pp. 10–16.

Ljubešić, N. and Kranjčić, D. (2015) 'Discriminating Between Closely Related Languages on Twitter', *Informatica*, 39(1).

Ljubesic, N., Mikelic, N. and Boras, D. (2007) 'Language Indentification: How to Distinguish Similar Languages?', in *Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on*. IEEE, pp. 541–546.

Lui, M. and Cook, P. (2013) 'Classifying English Documents by National Dialect', in *Proceedings of the Australasian Language Technology Association Workshop 2013 (ALTA 2013)*, pp. 5–15.

Luyckx, K. and Daelemans, W. (2011) 'The Effect of Author Set Size and Data Size in Authorship Attribution', *Literary and linguistic Computing*. Oxford University Press, 26(1), pp. 35–55.

Maamouri, M., Bies, A., Buckwalter, T. and Mekki, W. (2004) 'The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus', in *NEMLAR conference on Arabic language resources and tools*. Cairo, pp. 466–467.

Malmasi, S., Refaee, E. and Dras, M. (2015) 'Arabic Dialect Identification Using a Parallel Multidialectal Corpus', in *International Conference of the Pacific Association for Computational Linguistics*. Springer, pp. 35–53.

Marquardt, J., Farnadi, G., Vasudevan, G., Moens, M.-F., Davalos, S., Teredesai, A. and De Cock, M. (2014) 'Age and Gender Identification in Social Media', in *Proceedings of CLEF 2014 Evaluation Labs*, pp. 1129–1136.

Marton, Y., Wu, N. and Hellerstein, L. (2005) 'On Compression-Based Text Classification', *Advances in Information Retrieval*, 3408, pp. 300–314.

McCallum, A. and Nigam, K. (1998) ‘A Comparison of Event Models for Naïve Bayes Text Classification’, in *the Association for the Advancement of Artificial Intelligence-98 workshop on learning for text categorization*, pp. 41–48.

McEnery, T. and Wilson, A. (2003) ‘Corpus Linguistics’, *The Oxford handbook of computational linguistics*. Oxford University Press Oxford, pp. 448–463.

Memon, N., Kong, X. and Cinkler, J. (1999) ‘Context-based Lossless and Near-lossless Compression of EEG Signals’, *IEEE Transactions on Information Technology in Biomedicine*. IEEE, 3(3), pp. 231–238.

Mendenhall, T. C. (1887) ‘The Characteristic Curves of Composition’, *Science*. JSTOR, 9(214), pp. 237–249.

Mesleh, A. (2007) ‘Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System’, *Journal of Computer Science*, 3(6), pp. 430–435.

Mikros, G. K. (2012) ‘Authorship Attribution and Gender Identification in Greek Blogs’, *Methods and Applications of Quantitative Linguistics*. Academic Mind University of Belgrade, 21, pp. 21–32.

Mitchell, T. M. (1997) ‘Machine Learning. 1997’, *Burr Ridge, Illinois: McGraw Hill*, 45(37), pp. 870–877.

Modak, S. and Mondal, A. C. (2014) ‘A Comparative study of Classifiers’ Performance for Gender Classification’, *International Journal of Innovative Research in Computer and Communication Engineering*, 2(5), pp. 4214–4222.

Moffat, A. (1990) ‘Implementing the PPM Data Compression Scheme’, *IEEE Transactions on communications*. IEEE, 38(11), pp. 1917–1921.

Mosteller, F. and Wallace, D. L. (1963) ‘Inference in an Authorship Problem: A Comparative Study of Discrimination Methods Applied to the Authorship of the Disputed Federalist Papers’, *Journal of the American Statistical Association*. Taylor & Francis, 58(302), pp. 275–309.

Mubarak, H. and Darwish, K. (2014) ‘Using Twitter to Collect a Multi-Dialectal Corpus of Arabic’, in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*. Association for Computational Linguistics, pp. 1–7.

Mulac, A. and Lundell, T. L. (1994) ‘Effects of Gender-Linked Language Differences in Adults’ Written Discourse: Multivariate Tests of Language Effects’, *Language & Communication*. Elsevier, 14(3), pp. 299–309.

Mulac, A., Studley, L. B. and Blau, S. (1990) ‘The Gender-Linked Language Effect in Primary and Secondary Students’ Impromptu Essays’, *Sex Roles*. Springer, 23(9–10), pp. 439–470.

Mulac, A., Wiemann, J. M., Widenmann, S. J. and Gibson, T. W. (1988) ‘Male/female language Differences and Effects in Aame-Sex and Mixed-Sex Dyads: The Gender-linked Language Effect’, *Communications Monographs*. Taylor & Francis, 55(4), pp. 315–335.

Nagy, N., Zhang, X., Nagy, G. and Schneider, E. W. (2006) ‘Clustering Dialects Automatically: A Mutual Information Approach’, *University of Pennsylvania Working Papers in Linguistics*, 12(2), p. 12.

Nelson, M. and Gailly, J.-L. (1996) *The Data Compression Book*. M&T Books New York.

Nwesri, A. F. A., Tahaghoghi, S. M. M. and Scholer, F. (2006) ‘Capturing Out-of-Vocabulary Words in Arabic Text’, in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 258–266.

Oakes, M. P. (2004) ‘Ant Colony Optimisation for Stylometry: The Federalist Papers’, in *Proceedings of the 5th International Conference on Recent Advances in Soft Computing*, pp. 86–91.

Oco, N. and Roxas, R. E. (2012) ‘Pattern Matching Refinements to Dictionary-Based Code-Switching Point Detection’, in *Proceedings of the 26th Pacific Asia Conference*

on Language, Information, and Computation, pp. 229–236.

Pavelec, D., Justino, E. and Oliveira, L. S. (2007) 'Author Identification Using Stylometric Features', *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. Asociación Española para la Inteligencia Artificial, 11(36).

Piergallini, M., Shirvani, R., Gautam, G. S. and Chouikha, M. (2016) 'Word-Level Language Identification and Predicting Codeswitching Points in Swahili-English Language Data', in *Proceedings of the Second Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics, pp. 21–29.

Pillay, S. R. and Solorio, T. (2010) 'Authorship Attribution of Web Forum Posts', in *ECrime Researchers Summit (ECrime), 2010*. IEEE, pp. 1–7.

Rabab'ah, A., Al-Ayyoub, M., Jararweh, Y. and Aldwairi, M. (2016) 'Authorship Attribution of Arabic Tweets', in *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*. IEEE, pp. 1–6.

Rangel, F., Rosso, P., Potthast, M. and Stein, B. (2017) 'Overview of the 5th Author Profiling Task at Pan 2017: Gender and Language Variety Identification in Twitter', *Working Notes Papers of the CLEF*.

Rao, D., Yarowsky, D., Shreevats, A. and Gupta, M. (2010) 'Classifying Latent User Attributes in Twitter', in *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*. ACM, pp. 37–44.

Rayson, P. (2012) 'Wmatrix Corpus Analysis and Comparison Tool', *Lancaster University*.

Reicher, T., Krišto, I., Belša, I. and Šilić, A. (2010) 'Automatic Authorship Attribution for Texts in Croatian Language using Combinations of Features', in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, pp. 21–30.

Ritchie Key, M. (1975) 'Male/Female Language', *New York*.

Ritzen, Y. (2019) *The fake Twitter Accounts Influencing the Gulf Crisis*, *Aljazeera.com*. Available at: https://www.aljazeera.com/news/2019/07/fake-twitter-accounts-influencing-gulf-crisis-190717052607770.html (Accessed: 4 August 2019).

Rivera, G. C. (2019) *Automatic Detection of Code-Switching in Arabic Dialects*. Masters Thesis, Massachusetts Institute of Technology.

De Roeck, A. N. and Al-Fares, W. (2000) 'A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots', in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 199–206.

Saad, M. (2017) *Egyptian Comparable Wikipedia Corpus*. Available at: https://www.kaggle.com/mksaad/arb-egy-cmp-corpus (Accessed: 4 April 2018).

Saad, M. K. and Ashour, W. M. (2010) 'OSAC: Open Source Arabic Corpora', *Conference: 6th ArchEng International Symposiums, EEECS'10 the 6th International Symposium on Electrical and Electronics Engineering and Computer Science*, 10.

Sadat, F., Kazemi, F. and Farzindar, A. (2014) 'Automatic Identification of Arabic Dialects in Social Media', in *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*. ACM, pp. 35–40.

Salama, A., Bouamor, H., Mohit, B. and Oflazer, K. (2014) 'YouDACC: the Youtube Dialectal Arabic Commentary Corpus'. Figshare.

Salem, F. (2017) 'Social Media and the Internet of Things towards Data-Driven Policymaking in the Arab World: Potential, Limits and Concerns', *The Arab Social Media Report, Dubai: MBR School of Government*, 7.

Salomon, D. (2004) *Data Compression: the Complete Reference*. Springer Science & Business Media.

Samih, Y. and Maier, W. (2016) 'Detecting Code-Switching in Moroccan Arabic Social Media', *SocialNLP@ IJCAI-2016, New York*.

Sap, M., Park, G., Eichstaedt, J., Kern, M., Stillwell, D., Kosinski, M., Ungar, L. and Schwartz, H. A. (2014) 'Developing Age and Gender Predictive Lexica over Social Media', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1146–1151.

Scotton, C. M. and Ury, W. (1977) 'Bilingual Strategies: The Social Functions of Code-Switching', *International Journal of the Sociology of Language*. Walter de Gruyter, Berlin/New York, 1977(13), pp. 5–20.

Sebastiani, F. (2002) 'Machine Learning in Automated Text Categorization', *ACM Computing Surveys (CSUR)*. ACM, 34(1), pp. 1–47.

Shaker, K. and Corne, D. (2010) 'Authorship Attribution in Arabic using a Hybrid of Evolutionary Search and Linear Discriminant Analysis', in *Computational Intelligence (UKCI), 2010 UK Workshop on*. IEEE, pp. 1–6.

Shrestha, P. (2016) 'Codeswitching Detection via Lexical Features in Conditional Random Fields', in *Proceedings of The Second Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics, pp. 121–126.

Silva, R. S., Laboreiro, G., Sarmento, L., Grant, T., Oliveira, E. and Maia, B. (2011) '"twazn me!!! ;(" Automatic Authorship Analysis of Micro-Blogging Messages', in *International Conference on Application of Natural Language to Information Systems*. Springer, pp. 161–168.

Smrž, O. and Hajic, J. (2006) 'The Other Arabic Treebank: Prague Dependencies and Functions', *Arabic Computational Linguistics: Current implementations. CSLI Publications*, 104.

Stamatatos, E. (2009) 'A Survey of Modern Authorship Attribution Methods', *Journal of the American Society for information Science and Technology*. Wiley Online Library, 60(3), pp. 538–556.

Stein, G. and Quirk, R. (1995) 'Standard English', *The European English Messenger*,

4(2), pp. 62–63.

Ta'amneh, H., Keshek, E. A., Issa, M. B., Al-Ayyoub, M. and Jararweh, Y. (2014) 'Compression-based Arabic Text Classification', in *Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on*. IEEE, pp. 594–600.

Takezawa, T., Kikui, G., Mizushima, M. and Sumita, E. (2007) '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*, pp. 303–324.

Tan, R. H. R. and Tsai, F. S. (2010) 'Authorship Identification for Online Text', in *Cyberworlds (CW), 2010 International Conference on*. IEEE, pp. 155–162.

Tarmom, T., Teahan, W., Atwell, E. and Alsaka, M. (2018) 'Compression vs Traditional Machine Learning Classifiers to Detect Code-switching in Varieties and Dialects: Arabic as a Case Study', *Natural Language Engineering*, 1(1), pp. 1–20.

Teahan, W. (2018) 'A Compression-Based Toolkit for Modelling and Processing Natural Language Text', *Information*. Multidisciplinary Digital Publishing Institute, 9(12), p. 294.

Teahan, W. J. and Alhawiti, K. M. (2013) *Designcompilation and Preliminary Statistics of Compression Corpus of Written Arabic*. Bangor University, Tech. Rep., 2013.[Online]. Available: http://pages. bangor. ac. uk/eepe04/index. html.

Teahan, W. J. and Cleary, J. G. (1997) 'Models of English text', in *Proceedings DCC'97. Data Compression Conference*. IEEE, pp. 12–21.

Teahan, W. J. and Harper, D. J. (2003) 'Using Compression-Based Language Models for Text Categorization', *Language Modeling for Information Retrieval*. Springer, pp. 141–165.

Thomas, D. (2011) *An Empirical Study of Stream-Based Techniques for Text Categorization*. Ph.D Thesis, Bangor University.

Thomson, R. and Murachver, T. (2001) ‘Predicting Gender from Electronic Discourse’, *British Journal of Social Psychology*. Wiley Online Library, 40(2), pp. 193–208.

Tiedemann, J. and Ljubešić, N. (2012) ‘Efficient Discrimination Between Closely Related Languages’, *Proceedings of COLING 2012, the 24th International Conference on Computational Linguistics*. The COLING 2012 Organizing Committee, pp. 2619–2634.

Trudgill, P. (1972) ‘Sex, Covert Prestige and Linguistic Change in the Urban British English of Norwich’, *Language in society*. Cambridge University Press, 1(2), pp. 179–195.

Trudgill, P. and Hannah, J. (2013) *International English: A Guide to the Varieties of Standard English*. Routledge.

Türkoğlu, F., Diri, B. and Amasyalı, M. F. (2007) ‘Author Attribution of Turkish Texts by Feature Mining’, in *International Conference on Intelligent Computing*. Springer, pp. 1086–1093.

Tweepy (2009) *Tweepy*, *Tweepy.org*. Available at: Tweepy.org (Accessed: 5 March 2016).

Ugheoke, T. O. and Saskatchewan, R. (2014) ‘Detecting the Dender of a Tweet Sender’, *M. Sc. Project Report, Department of Computer Science, University of Regina, Regina*.

Ulman, A. (2017) *Saudi Arabia: A Different Kind of Feminism*. Available at: Feministcampus.org (Accessed: 29 July 2018).

Volkova, S., Wilson, T. and Yarowsky, D. (2013) ‘Exploring Demographic Language Variations to Improve Multilingual Sentiment Analysis in Social Media’, in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1815–1827.

Wahbeh, A., Al-Kabi, M., Al-Radaideh, Q., Al-Shawakfa, E. and Alsmadi, I. (2011) ‘The Effect of Stemming on Arabic Text Classification: an Empirical Study’, *International*

Journal of Information Retrieval Research (IJIRR). IGI Global, 1(3), pp. 54–70.

Weatheral, A. (2002) 'Towards Understanding Gender and Talk-in-Interaction', *Discourse & Society*. Sage Publications Sage UK: London, England, 13(6), pp. 767–781.

Welch, T. A. (1984) 'Technique for High-Performance Data Compression', *Computer*. IEEE, (52).

Wells, J. C. (1982) *Accents of English*. Cambridge University Press.

Williams, C. B. (1975) 'Mendenhall's Studies of Word-Length Distribution in the Works of Shakespeare and Bacon', *Biometrika*. Oxford University Press, 62(1), pp. 207–212.

Witten, I. H., Neal, R. M. and Cleary, J. G. (1987) 'Arithmetic Coding for Data Compression', *Communications of the ACM*. ACM, 30(6), pp. 520–540.

Wu, P. and Teahan, W. J. (2008) 'A New PPM Variant for Chinese Text Compression', *Natural Language Engineering*. Cambridge University Press, 14(03), pp. 417–430.

Xu, F., Wang, M. and Li, M. (2017) 'Sentence-Level Dialects Identification in the Greater China Region', *arXiv preprint arXiv:1701.01908*.

Yang, Y. and Liu, X. (1999) 'A Re-Examination of Text Categorization Methods', in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, p. 99.

Yeong, Y.-L. and Tan, T.-P. (2010) 'Language Identification of Code Switching Malay-English Words Using Syllable Structure Information', in *Spoken Languages Technologies for Under-Resourced Languages*.

Yule, G. U. (1939) 'On Sentence-Length as a Statistical Characteristic of Style in Prose: With Application to Two Cases of Disputed Authorship', *Biometrica*, 30, pp. 363–390.

Zaghouani, W. and Charfi, A. (2018) 'Arap-Tweet: A Large Multi-dialect Twitter Corpus for Gender, Age and Language Variety Identification', *arXiv preprint arXiv:1808.07674*.

Zaidan, O. F. and Callison-Burch, C. (2011) ‘The Arabic Online Commentary Dataset: an Annotated Dataset of Informal Arabic with High Dialectal Content’, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2, pp. 37–41.

Zaidan, O. F. and Callison-Burch, C. (2014) ‘Arabic Dialect Identification’, *Computational Linguistics*. MIT Press, 40(1), pp. 171–202.

Zampieri, M. and Gebre, B. G. (2012) ‘Automatic Identification of Language Varieties: The Case of Portuguese’, in *KONVENS2012-The 11th Conference on Natural Language Processing*. Österreichischen Gesellschaft für Artificial Intelligende (ÖGAI), pp. 233–237.

Zampieri, M., Gebre, B. G. and Diwersy, S. (2013) ‘N-gram Language Models and POS Distribution for the Identification of Spanish Varieties (Ngrammes et Traits Morphosyntaxiques pour la Identification de Variétés de l’Espagnol)[in French]’, *Proceedings of TALN 2013*, 2, pp. 580–587.

Zhao, Y. (2007) ‘Effective Authorship Attribution in Large Document Collections’.

Ziv, J. and Lempel, A. (1977) ‘A Universal Algorithm for Sequential Data Compression’, *IEEE Transactions on Information Theory*. IEEE, 23(3), pp. 337–343.

Ziv, J. and Lempel, A. (1978) ‘Compression of Individual Sequences via Cariable-Rate Coding’, *IEEE transactions on Information Theory*. IEEE, 24(5), pp. 530–536.

Zrigui, M., Ayadi, R., Mars, M. and Maraoui, M. (2012) ‘Arabic Text Classification Framework Based on Latent Dirichlet Allocation’, *Journal of Computing and Information Technology*. SRCE-Sveučilišni računski centar, 20(2), pp. 125–140.