
DEPARTMENT OF PHILOSOPHY,
LINGUISTICS AND THEORY OF SCIENCE

# AUTOMATIC DETECTION OF UNDER-RESOURCED LANGUAGES

## Dialectal Arabic Short Texts

**Wafia Adouane**


| | |
|---|---|
| Master's Thesis: | 30 credits |
| Programme: | Master's Programme in Language Technology |
| Level: | Advanced level |
| Semester and year: | Spring, 2016 |
| Supervisor: | Richard Johansson, Nasredine Semmar and Alan Said |
| Examiner: | Staffan Larsson |
| Report number: | (number will be provided by the administrators) |
| Keywords: | Under-resourced languages, Discrimination between similar languages and language varieties, Arabic varieties, Linguistic resource building |

# Abstract

Automatic Language Identification (ALI) is the first necessary step to do any language-dependent natural language processing task. It is the identification of the natural language of the input content by a machine. Being a well-established task in computational linguistics since early 1960's, various methods have been successfully applied to a wide range of languages. The state-of-the-art automatic language identifiers are based on character *n*-gram models trained on huge corpora. However, there are many natural languages which are not yet automatically processed. For instance, minority languages or informal forms of standard languages (general purpose languages used only in media/administration and taught at schools). Some of these languages are only spoken and do not exist in a written format. The use of social media platforms and new technologies have facilitated the emergence of written format for these spoken languages based on pronunciation. These new written languages are under-resourced, hence the current ALI tools fail to properly recognize them.

In this study, we revisit the problem of ALI with the focus on discriminating under-resourced similar languages. We deal with the case of dialectal Arabic (informal Arabic varieties) used in social media, and we consider each Arabic dialect/variety as a stand-alone language. Our main purpose is to investigate the performance of the ALI standard methods, namely machine learning and dictionary-based methods, on distinguishing Arabic varieties. Given the fact that discriminating between Arabic varieties is a nontrivial linguistic task because of the absence of any clear-cut borderlines between the variants, we can conclude that machine learning models are well suited for Arabic dialects identification. Support vector machines, namely the LinearSVC method combining the character-based 5-6-grams with dialectal vocabulary as features, outperforms all the other methods. The dictionary-based method suffers mainly from the shortage in the vocabulary coverage.

# Acknowledgements

I would like to thank very much my supervisors, Richard Johansson, Nasredine Semmar and Alan Said, for accepting to work on this topic and for their invaluable guidance and useful feedback. I would like also to thank Victoria Bobicev for her help in explaining and implementing the Prediction by Partial Matching (PPM) method. Special thanks go to all my lovely friends who collected the dialectal data and annotated it for free. Well that's friendship! Sorry, for not naming you, one by one, but I'm sure you all know whom I'm talking about. My thanks go also to the students of the Arabic linguistics and literature department at Boumerdès University for accepting to do the annotation for us. This project would not exist without you 'Anjad, choukran ktir ilkon'. I would like also to thank all my teachers and classmates at the MLT program. Last but not least, I would like to express my deepest gratitude to my parents 'la raison de mon existence et ma source d'insperation', my brothers and sisters and my husband for their support despite the long distance.

# Contents

# 1 Introduction

Automatic Language Identification (ALI), also called language recognition, is a task of identifying the natural language[1] an input text is written in. It is the first step for any language-dependent Natural Language Processing (NLP) application. Being a well studied field in computational linguistics, ALI is considered to be a solved problem since years given the successful achievements for many languages. ALI is commonly framed as a categorization[2] problem. However, the rapidly growth and wide dissemination of social media platforms and new technologies have contributed to the emergence of written forms of some varieties which are either minority languages or colloquial forms of general purpose (standard) languages. These languages were not written before social media and mobile phone messaging services, and they are typically under-resourced. The state-of-the-art automatic language identification tools fail to recognize them and represent them by a unique category; standard language. For instance, whatever is written in French is considered as French even though there are many French varieties which are considerably different from each other. They also fail to properly identify social media content written in well-resourced languages. The reason is that social media typically uses informal[3] languages. In this study, we deal with the case of Arabic varieties including Modern Standard Arabic (MSA) and colloquial variants. We consider only the seven (7) most popular Arabic dialects, based on the geographical classification, plus MSA. There are many local dialects due to the linguistic richness of the Arab world, but it is hard to deal with all of them for two raisons: it is hard to get enough data, and it is hard to find reliable linguistic features as these local dialects are very similar.

## 1.1 Motivation

The vast majority of the world's languages, particularly informal[4]ones, are under-resourced. Therefore, it is hard to analyze and process them automatically using the standard ALI methods which require huge corpora for training (Benajiba & Diab, 2010). Furthermore, available automatic language identifiers perform well for long documents as they rely on character/word *n*-gram models and statistics using large training corpora to identify the language of an input text (Zampierri & Gebre, 2012). New technologies and social media platforms, nevertheless, use short texts for technical reasons. In addition to this serious weakness, current language identification tools always return an output language even for unseen languages in the training dataset. This causes the classification of unknown languages as unrelated languages, which leads to misleading information and wrong analysis. For instance Berber written in Arabic script, which is an unknown language, is classified as Arabic.

Arabic varieties[5] are a case of under-resourced and unknown languages to the available automatic language identifiers, despite their widespread use on the Web. Current automatic language identifiers classify all of them in one class, namely Arabic, which refers to Modern Standard Arabic (MSA).

1 Any language spoken naturally by humans compared to artificial languages.

2 Assigning a predefined category to a given text based on the presence or absence of some features.

3 Languages which do not adhere to the grammar or the orthography of their standard form.

4 The same as in note 3: languages which do not adhere to the grammar or the orthography of their standard form.

5 A collection of written Arabic varieties which are basically spoken and informal languages.

Arabic variants, written in the Arabic script[6], do share lots of vocabulary and morpho-syntactic structures with each other and MSA as well. Therefore, they are a perfect example of the challenging tasks o f Discriminating Similar Languages (DSL) and Discriminating Language Varieties (DLV). DSL deals with identifying similar languages from each other, for instance discriminating between Bosnian, Croatian and Serbian languages. DLV is a special task of DSL which deals with discriminating between varieties of the same language, for instance Brazilian Portuguese and European Portuguese. Both DSL and DLV tasks are sub-tasks of ALI. Very limited research has been done for both automatic identification of Arabic dialects and DSL/DLV tasks (Zaidan, 2012; Saâdane, 2015). Our research question, in this study, is to investigate whether standard ALI methods, namely statistics using *n*-gram models and dictionary-based methods will be able to discriminate between Arabic varieties in the context of the social media domain, which poses significant challenges to Natural Language Processing in general.

## 1.2 Goals and contributions

Automatic processing of informal languages has recently attracted the attention of the research community. This is also our goal in this thesis which seeks, more specifically, to fill a serious gap in the automatic processing of under-resourced languages in the context of social media. Our main goal is twofold:

- Design an automatic language identifier for the most popular Arabic dialects which is able to discriminate between these similar languages.
- Build linguistic resources for Arabic dialects to overcome the issue of resource scarceness.

The main contributions of this project are:

- We provide an automatic language identifier which distinguishes properly between Arabic and Arabicized Berber which is not an Arabic variant but coexists with Arabic and which is still misclassified as Arabic by the state-of-the-art automatic language identifiers.
- Most of the works done before focus on distinguishing between Modern Standard Arabic (MSA) and dialectal Arabic (DA), where the latter is regarded as one class and which consists mainly of Egyptian Arabic. Further, Zaidan (2012) in his PhD distinguishes between four Arabic varieties (MSA, Egyptian, Gulf and Levantine dialects) using *n*-gram models. Saâdane (2015) in her PhD classifies Maghrebi Arabic (Algerian, Moroccan and Tunisian dialects) using morpho-syntactic information. To the best of our knowledge, this is the first work which distinguishes between eight (8) high level Arabic variants (Algerian, Egyptian, Gulf, Levantine, Mesopotamian, Moroccan, Tunisian dialects and MSA).
- Limited work has been done to automatically process dialectal Arabic mainly because of the lack of data, let alone annotated data. The linguistic resources built in this project would help to mitigate this serious issue. The dialectal lexicons will be soon available online.
- As a minor contribution, we show that Arabicized Berber, which is also an under-resourced language, is easily separated from Arabic even though there is a considerable overlap between them.

6 They are also written in Latin script or what is known as Romanized Arabic or Arabizi.

## 1.3 Thesis organization

We start by giving a general overview of Automatic Language Identification, Discriminating Similar Languages, Discriminating Language Varieties, Arabic Natural language processing related work, and the potential applications of dialectal Arabic identification in Chapter 2. We continue by describing the linguistic landscape of Arabic and its variants followed by their main characteristics based on modern Arabic dialectology in Chapter 3. Then we will describe the process of building the linguistic resources used in this study and motivate the choice of the used approaches in Chapter 4. We will describe the experiments and analyze the results in Chapter 5, and then conclude with the findings of our study and give avenues for the future research in Chapter 6.

# 2 Background

## 2.1 Automatic Language Identification

As introduced in Chapter 1, Automatic Language Identification (ALI) is a crucial NLP task which consists in identifying the natural language of an input text by a machine. It is the first text processing to properly deal with language-based NLP task. ALI for written texts is a well-established task in computational linguistics since early 1960's. Mustonen (1965) applied statistical methods using syllable information to distinguish between English, Finnish and Swedish. Some researchers argue that ALI can be traced back as early as 1967 to the experiments of E. Mark Gold. "Language identification was arguably established as a task by Gold (1967), who construed it as a close class problem: given data in each of a predefined set of possible languages, human subjects were asked to classify the language of a given test documents" (Baldwin & Lui, 2010). Other researchers report that early ALI approaches started in 1980 with Norman Ingle's work where he used stop word frequency, applying Zipf's law, as features to recognize a language. "Ingle applied Zipf's law distribution to order the frequency of stop words in a text and used this information for language identification" (Zampierri & Gebre, 2012).

Since then, various methods have been used to approach the task of automatic language identification. The simplest method is using the language special characters or diacritical marks to distinguish it from other languages with different character set. Several mathematical models, using statistics and probabilities, have been applied to written language identification task as well. These methods, abundantly discussed in the literature, are using some information as features. "The main idea was to create distributions of specific 'elements' for a number of languages and, subsequently, to compare these to the distribution of the same elements obtained from a given text" (Hornik et al., 2013). Among these methods, we list: using syllables (Mustonen, 1965), unique letters, words or combinations (Newman, 1987), orthography (Beesley, 1988), word-based *n*-grams (Batchelder, 1992), morpho-syntactic characteristics (Ziegler, 1992), character sequence prediction (Dunning, 1994), most frequent character-based *n*-grams (Cavnar & Trenkle, 1994; Combrinck & Botha, 1994), estimating the *n*-gram likelihood (Padró & Padró, 2004), Prediction by Partial Matching using character/word as features (Bratko et al., 2006), support vector machines (SVMs) with both character and word *n*-grams (Yan Deng, 2008), and POS distribution (Zampieri et al., 2013). Many studies comparing different methods have been published for instance Grefenstette (1995) and Padró & Padró (2004). In addition, dictionary-based methods have been used (Řehůřek & Kolkus, 2009). All these methods, as well as others we have not mentioned, are reported to perform very well for standard languages.

Current available language identifiers rely on character/word *n*-gram models and statistics using large training corpora to identify the language of an input text (Zampierri & Gebre, 2012). They are mainly trained on standard languages and not on the varieties of each language. For instance, current language identification tools can easily distinguish Arabic from Persian, Pashto and Urdu based on the character sets and topology. However, they fail to identify Arabic varieties from each other.

## 2.2 Discriminating Similar Languages and Language Varieties

As described above, Discriminating Similar Languages (DSL) and Discriminating Language Varieties (DLV) are one of the serious bottlenecks[7] of the current automatic language identification tools. They are a big challenge for under-resourced languages. DLV is a special case of DSL where the languages to distinguish are very close. DSL and DLV are even harder for the social media domain which uses short texts written in informal languages. These tasks have recently attracted the intention of the research community, for instance the organization of the DSL Shared Task since 2014 (Goutte et al., 2016). DSL can be simply defined as a specification or a sub-task of automatic language identification (Tiedemann & Ljubešić, 2012). Many of the standard methods used for the ALI have been applied to the DLS and DLV tasks for some languages. Goutte et al., (2016) give a comprehensive bibliography of the recently published papers dealing with these tasks.

## 2.3 Arabic Natural Language Processing

Most of the Arabic NLP tools are MSA-based because of the data availability. “The fact is that most of the robust tools designed for the processing of Arabic to date are tailored to MSA due to the abundance of resources for that variant of Arabic” (Benajiba & Diab, 2010). However, the considerable differences between Arabic varieties and MSA makes it unpractical to apply the MSA-based NLP tools to process written dialectal Arabic. The results are simply incomprehensible outputs. “In fact, applying NLP tools designed for MSA directly to dialectal Arabic (DA) yields significantly lower performance, making it imperative to direct the research to building resources and dedicated tools for DA processing” (Benajiba & Diab, 2010).

Little work have been done for written dialectal Arabic. Available NLP tools for dialectal Arabic deal mainly with Egyptian Arabic such, as MADAMIRA, which is a morphological Analyzer and dis-ambiguator for Modern Standard Arabic (MSA) and Egyptian Arabic (Pasha et al., 2014), and opinion mining/sentiment analysis for colloquial Arabic (Egyptian Arabic) (Hossam et al., 2015). Eskander et al., (2014) presented a system for automatic processing of Arabic social media text written in Arabizi[8]. For written dialectal Arabic, there are some works as well, namely automatic identification of some Arabic dialects (Egyptian, Gulf and Levantine) Elfardy & Diab (2013) identified MSA from Egyptian at a sentence level, Tillmann et al., (2014) proposed an approach to improve classifying Egyptian and MSA at a sentence level, and Saâdane (2015) built a morpho-syntactic analyzer for Maghrebi Arabic (Algerian, Moroccan and Tunisian dialects).

The lack of data does not apply to spoken dialectal Arabic as there are sufficient phone and TV program recordings which are easy to transcribed based on the need. “The problem is somewhat mitigated in the speech domain, since dialectal data exists in the form of phone conversations and television program recordings, but, in general, dialectal Arabic data sets are hard to come by” (Zaidan & Callison-Burch, 2014). Akbacak et al., (2009), Akbacak et al., (2011), Lei & Hansen (2011), Boril et al., (2012), and Zhang et al., (2013) are some works done for spoken dialectal Arabic.

'Real Arabic' is the Arabic used by people in their daily interactions; a language which has a communicative function. This is dialectal Arabic and not MSA (Benajiba & Diab, 2010). Consequently, to be able to understand the Arabic social media content and build useful NLP application according to the needs of users, it is necessary to process dialectal Arabic. “Any serious attempt at processing real Arabic has to account for the dialects” (Ibid).

7 Among others like they fail to properly identify uncontrolled languages which are varieties of general purpose languages.

8 Arabic written in Latin script

## 2.4 Applications of dialectal Arabic identification

Identifying Arabic is important to analyze it and automatically process it. Being able to properly discriminate between its varieties will avoid the risk of mixing meanings because of the big differences between these variants and the considerable amount of false friends between them. Recently, there is a considerable interest by both research and industry to automatically process social media content, sentiment analysis, opinion mining, event and information extraction, authorship recognition, machine translation, etc. All the mentioned applications are language-dependent and require the identification of the Arabic variety at hand to accurately handle content, and wrong identification will provide misleading information.

On top of this, identifying Arabic dialects and building linguistic resources for each dialect separately will help both to adapt the existing resources built originally for MSA, and to trustfully build new applications. Correctly distinguishing between Arabic variants will also be very useful in information retrieval, cross language information retrieval and user-based search applications. In case a user is interested in a particular content, it would be possible to filter the search by the desired Arabic variants. In the context of information security, language variety recognition might be useful in determining the origin of spams and online threats via authorship analysis given that users can change their locations but hardly change their linguistic identity. In general, the correct detection of each variant will help reducing ambiguity and improving language-dependent NLP applications such as machine translation.

# 3 Arabic variants

Arabic is a Semitic language written in Arabic script from right to left. It is the 5th world largest language in terms of the number of speakers[9]. Linguistically, the origin of Arabic is still not proved because Arabic existed well before Islam. The major issue is that the pre-Islamic Arabic is not documented, so only few things are known about that period (Rabin, 1951). Despite the fact that it is the official language of the Arab world, Arabic is a mixture of varieties and not just one language (Hassan R.S., 1992). These varieties can be divided into two classes: Modern Standard Arabic and dialects.

## 3.1 Modern Standard Arabic

Modern Standard Arabic (MSA) is the only formal and standardized written variety, which makes Arabic a monocentric[10] language. It is the official language used in media and schools in all Arabic-speaking countries. In many cases, MSA is used as a lingua franca, namely between speakers from Middle East and North Africa because their dialects are not mutually intelligible. Scholars consider MSA as the reference as it preserves the ancient properties (grammar, morphology and orthography, etc.) of Classical Arabic, called also Quranic Arabic. MSA is not a dialect as it has no native speakers.

## 3.2 Arabic languages / dialects / varieties

In this thesis, we will use the terms 'language', 'variety' and 'dialect' interchangeably. The reason is that we could not find any linguistic difference between the three terms. By definition, a dialect is a variety of a language which is different from other varieties of the same language in terms of morpho-syntactic structures, phonology and vocabulary. It is a native language of a group of people in a given region or social class. We can say that a variety is a dialect which has a standard form (codified) compared to a dialect which does not have a standard form. In Arabic, Modern Standard Arabic (MSA) is a mixture of languages, and it does not have native speakers, so it is not a dialect. But it has a standard form, so it is considered as an Arabic variety only which is hardly, if at all, used outside school, media, official communication or administration. Arabic dialects have their own varieties, for instance the Arabic spoken in Cairo in Egypt is different from the one spoken in Alexandria, etc. These Arabic dialects are different from each other, and they have their own morphology, phonology and syntax, but since a long time they were not allowed to be written for political reasons. Based on all of this, Arabic NLP community considers MSA to be the only standard Arabic variant and the remaining variants as informal languages (because of the absence of standard orthography and grammar). They are simply referred to as Egyptian Arabic, Moroccan Arabic, etc.

Modern Arabic dialectology considers each variant as a stand-alone language because they have all the criteria other languages would have (native speakers, morphology, syntax, phonology, semantics,

9 More than 295 millions according to https://en.wikipedia.org/wiki/World_language#cite_note-27 retrieved on April 22nd, 2016.

10 A language with only one standardized version

and they have their own variants) the only missing part is that they are not documented (Palva, 2006). However, with the rise of social media and new technologies, these colloquial languages have acquired some written form based on pronunciation. There is no clear-cut decision, which is linguistically well motivated, to whether a 'language' is a dialect, variety or a language. For instance, Swedish, Danish and Norwegian are considered both dialects of the same language and stand-alone languages even though they are very similar. Also Spanish and Italian are very similar, yet they are considered as stand-alone languages. However, Cantonese and Mandarine are very different Chinese variants (a speaker of one variety does not understand a speaker of the other variety) which are considered as dialects. This categorization is based on the fact that these languages are spoken in different countries (nation). Arabic varieties, which are also considerably different from each other, are used in different countries. Based on this, it does not really matter if we consider Arabic variants as languages or dialects. In terms of usage, Arabs use their dialects in their daily interactions. Therefore Arabic dialects are the 'real Arabic' used for communicative purpose. There are many varieties where each Arabic-speaking country has its own national varieties with their typical syntactic, morphological and lexical characteristics. Moreover, based on the fact that each national variety has regional and local varieties, each variety can be considered as a stand-alone language. These overlapping varieties are spoken colloquial languages[11] which are still not codified[12] despite their wide popularity for political reasons and not for any linguistic reason (Hassan R.S., 1992).

Concerning the origins of Arabic dialects, scholars say that "Arabic dialects appeared after the expansion of the Arabs, which began after the death of the Prophet Muhammad in 632 C.E." (Palva, 2006)[13]. This means that colloquial varieties, which are all purely spoken languages, originated from the contact of the Arabic spoken in Arabia with other languages outside that region. This applies to modern Arabic dialects, which still exist nowadays as well as to those which disappeared like Andalusian and Sicilian dialects. Contrary to the Romance languages which had developed from Latin, MSA has acquired its present form from the various varieties it had contact with. Hassan R.S. (1992) explains "...one may argue that the varieties of Arabic are not necessarily deviations from a norm, but rather a norm (let it be the standard variety in the past and the various koines[14] developing at present) has evolved or is evolving from the wide range of existing spoken varieties..."

## 3.3 Dialectal Arabic classification

Hassan R.S. (1992) describes the task of classifying Arabic dialects by saying "...these varieties are not difficult to recognize, but are impossible to describe as they are full of unpredictability and hybridization. They can be better described as geographical, cultural or social varieties rather than national norms." He suggests to classify Arabic varieties into geographically contiguous and culturally related blocs which had similar colonialism pattern. Geographically speaking, Arabic dialects are classified in two main blocs, namely Middle East (Mashriqi) and North Africa (Maghrebi) dialects. These two main blocs contain very different dialects. Therefore, it is better to narrow the space to the national level instead, i.e. subdivide these two blocs into national level where each group of close dialects is a norm for a country. This is hard to control because "national borders are not necessarily

11 Except Maltese which is the only official dialectal Arabic variety written in Latin script in its standard form.

12 Arabic dialects are not documented and do not have a standard orthography or grammar.

13 There were many varieties in Arabian Peninsula but just little is known about them.

14 A dialect of a region that becomes a standard form of a larger area

the most fitting framework for linguistic studies" (Taine-Cheikh, 2012). Figure 3.1 gives an idea about Arabic dialects linguistic borders according to modern dialectology[15].

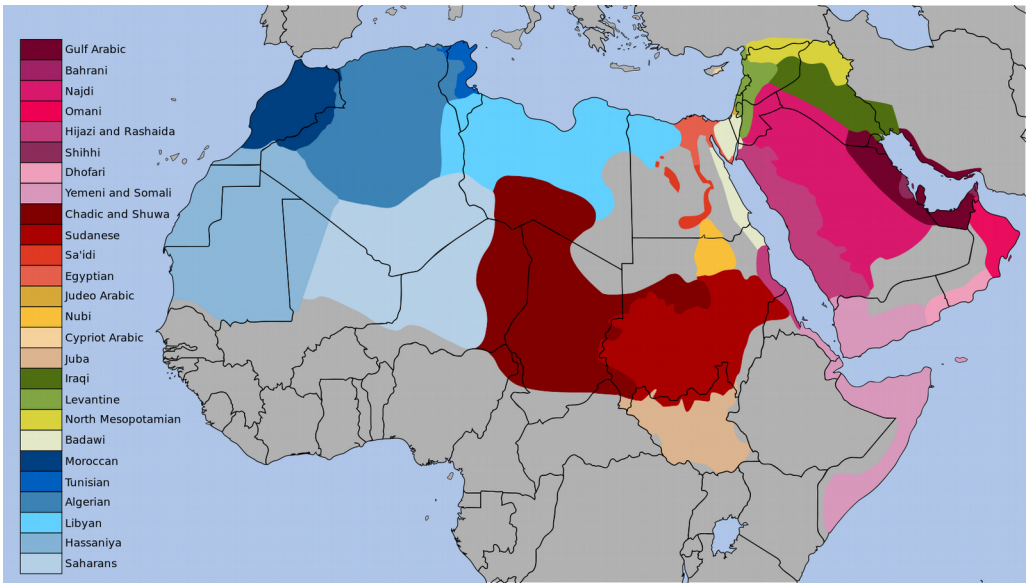



Figure 3.1: Dialectal Arabic linguistic borders map

Classifying Arabic varieties into national levels taking into account the linguistic borders is more accurate than dividing them into two main blocs; east and west. Nevertheless, this classification assumes that there exists only one Arabic variety within 'linguistic borders'. That is not the case as there are many regional and local dialects coexisting in the same area. Referring to dialects using the name of the countries where they are spoken is common among linguists and dialectologists. Palva (2006) justifies the use of generalizing labels for dialects "… they are used for the sake of convenience, although in fact they often refer to the dialects of the capital cities." He continues "this is not merely a simplification but, in a sense, it is also justified because of the ongoing trend toward regional standard dialects with the dialects of the urban centers as the models." For instance, Egyptian Arabic is in fact predominated by the Cairene dialect.

In this respect, Palva (2006) suggests that dialect boundaries should be defined by isoglosses[16]. "Drawing isoglosses on a map normally exhibits border areas in which a number of isoglosses lie close enough together to constitute bundles of isoglosses marking boundaries between different dialect areas. The bundles normally reveal the focal area of a dialect, and between the focal areas there are transitional areas in which the isoglosses do not tally with the bundles and in which contrasting items may be used interchangeably." Likewise, it will be possible to identify groups of close dialects.

Other traditional classifications were suggested as well, e.g. sociologically-based classification which takes into account the social environment where a dialect is spoken, and classifies it either as Bedouin (badwyn) or Sedentary (HaDari) dialect. Further, a religious affiliation-based classification has been suggested as there are considerable differences between Christians, Jewish and Muslims. "Also, among the same religious community, there are clear differences, the best example is between Shia-Sunni in Bahrain" (Palva, 2006). These are just some high level dialect classification. Further

15 The map is retrieved from Wikipedia on April 22nd, 2016.

16 The geographic boundary of a certain linguistic feature, such as the pronunciation of a vowel, the meaning of a word, or the use of some syntactic feature.

subdivisions were suggested as well. Palva explains that it is hard to find a dialectal clear-cut boundary, which causes a classification problem between some neighboring dialects. For instance some Egyptian dialects share lots of vocabulary with Maghrebi dialects.

The above-mentioned classifications are based on extralinguistic variables, simply because it is very hard to find a general valid linguistic classification which assumes the existence of a strong feature set. “We have to realize here that no generally accepted linguistic variables are available to serve for a linguistic classification of the Arabic dialects” (Behnstdt & Woidich, 2013). It would be useful to use the linguistic information of individual Arabic variant as discriminative features at least in clustering regional groups. This can be done by applying statistics. Behnstdt & Woidich (2013) share the idea with us by saying “using linguistic variables in this way as discriminants is possible for smaller regions”. Likewise, it would be possible to group Arabic dialects into regional clusters and find some of their interrelations. In practice, however, the biggest challenge is how to weight the importance of the linguistic features. This is still an unsolved issue.

It is necessary to decide how to cluster Arabic variants in order to be able to properly analyze and process them automatically. Nonetheless, it is hard to distinguish each variant from another based on the classification of Figure 3.1 because of the considerable lexical overlap and similarities between them. Moreover, it is very hard and expensive to collect data for each single variant given the fact that some are rarely used on the Web. Based on the fact that people of the same region tend to use the same vocabulary and have the same pronunciation, Habash (2010) has suggested to group Arabic dialects in six main groups, namely Egyptian (which includes Egyptian, Libyan and Sudanese), Levantine (which includes Lebanese, Jordanian, Palestinian and Syrian), Gulf (including Gulf Cooperation Council Countries), Iraqi, Maghrebi (which includes Algerian, Moroccan and Tunisian) and the rest are grouped in one class called 'Other'.

We suggest a further division based on isoglosses where each Maghrebi variant is counted as a separate language and which includes an additional Gulf/Mesopotamian[17] dialect group. So for the Mesopotamian Arabic, we include some local variants of Iraqi, Kuwaiti, Qatari and Emirati spoken Arabic. We group the rest of regions in the Gulf Arabic. Recent works consider all spoken Arabic in Gulf Cooperation Council Countries as Gulf Arabic. Our motivation to do so is that these two broad regional dialectal groups (Maghrebi and Gulf) include a wide variety of languages which are easily distinguished by humans. Therefore, machines should be also able to discriminate between these varieties. In this study, we consider eight (8) high level groups which are: Algerian (ALG), Egyptian (EGY), Gulf (GUL), Levantine (LEV), Mesopotamian (KUI), Moroccan (MOR), Tunisian (TUN) dialects plus MSA. In all cases, we will focus on the language of the indigenous populations and not on the Pidgin Arabic[18].

## 3.4 Characteristics of Arabic dialects

Nowadays, there is no single absolute classificatory criterion. Even isogloss criteria are not valid anymore because of the diglossic situation in all Arabic-speaking societies[19] (Enam El-Wer, 2013).

17 There is no clear-cut dialectal borderlines between the Arabic varieties spoken in the Arabian Peninsula, namely between Gulf Arabic and Mesopotamian Arabic. Qafisheh (1977) gives a thorough morpho-syntactic analysis of the Gulf Arabic including Bahraini, Emirati, Qatari, Kuwaiti and some regions of Saudi Arabia and excluding the Arabic dialects spoken in the rest of the Gulf countries. However, we do not have any morpho-syntactic parser, if it exists at all, to take all the grammars into account. We will base our dialect clustering on some common linguistic features, for instance the use of 'ch' instead of 'k', see (Palva, 2006) for more details.

18 Simplified language varieties created by foreigners living in Arabic-speaking countries to make communication easier.

19 Therefore considering the diglossia assumes that all the Arabic-speaking societies have invariant or uniform linguistic structure.

Instead, modern dialectology considers some prominent typological, structural and functional features. Arabic varieties are spoken informal languages with no standardized or normalized forms. Therefore, they adhere perfectly to the 'write-as-you-speak' principle and transcribe foreign words. None of them strictly adheres to the MSA grammar or orthography. People usually use the Arabic script, but in many cases when the available communication tools do not support Arabic script people use other scripts, for example the Latin script (Romanized Arabic). Some Arabic varieties are written in other scripts such as the Hebrew script (Judeo-Arabic) or Greek script (Cypriot Arabic).

Most Arab societies are multilingual or at least bilingual. North African countries for instance use a mixture of Berber, French, English, Spanish, Arabic which is itself a mixture of languages and lots of words of unknown origins. There is also extensive language mixing[20] between MSA and dialectal Arabic. The linguistic situation is reflected on the data available on the Web. For instance, the following sentence: بونجوغ عليكم لميسيو نتاع هاد السيمانة والو ماعجبتنيش قاع [bwnjwg Elykm lmysyw ntAE hAd AlsymAnp wAlw mAEjbtny$ qAE][21] which means [hello, the show of this week is bad, I did not like it at all] has at least three languages بونجوغ [bwnjwg] and لميسيو [lmysyw] (French), السيمانة [AlsymAnp] (Spanish), [qAE] قاع of unknown origin, عليكم [Elykm] (MSA) and the remaining words are Maghrebi Arabic. This is just an example of language mixing, which is heavily used. The use of mix-languages either in the Arabic script or another script is part of the informality of the dialectal Arabic for historical reasons.

Arabic dialects are under-resourced languages and the available automatic language identifiers classify all of them as MSA. There are some available data collections of folk songs and colloquial proverbs as well as some dialectal word lists (glossaries), but they are outdated and useless for our purpose or any other computational linguistic purpose as they are available only on paper. Another common characteristic of dialectal Arabic on the web, and Arabic in general, is that texts are unvoweled. For MSA, it is argued that the use of Arabic vowels (which are written as diacritics) causes visual disturbance for readers. Also, it causes an extra typing effort because each vowel is a separate character. Commonly, vowels are not used and readers still can understand the meaning in most cases. However, for dialects, vowels are not used and the reader can not understand the meaning of words. For machines, this is an extra ambiguity source because vowels act as a disambiguator in many cases for Arabic in general. Therefore, it is hard to guess the meaning of many words without giving their precise context.

While modern Arabic dialects share a significant number of distinctive features and some of them overlap significantly in vocabulary with each other, they have also considerable differences, particularly in the vocabulary (lexical items and semantics). The differences are not easily identified, i.e. no systematic repeated differences to catch as features, i.e. the meaning of words depends on the variety they are used in. In many cases, the Arabic variety determines the intended meaning. For instance, consider the following sentence: مابي أكل اليوم في البيت خلينا نروح نشتري الجاهز [mAby >kl Alywm fy Albyt xlynA nrwH n$try AljAhz]. Any Arabic speaker can clearly see that the sentence is not in MSA. It is either in Gulf dialect or in Mesopotamian (Iraqi/Kuwaiti Arabic). The meaning of the sentence, however depends on the Arabic variety. In Gulf Arabic, it means [I do not want to eat home food today, let's go and buy ready food] and in Mesopotamian Arabic, it means [there is no food at home today, let's go and buy ready food]. Let's take another example: جان الشعب من مؤيدي هاي الرئيس [jAm Al$Eb nm m&ydy hAy Alr}ys]. In Mesopotamian dialect, the sentence means [people were

20 This term refers to the use of more than one language in a single interaction. The classic code-switching framework does not apply to Arabic for many complex reasons which are out of our scope. Researchers like Sankoff (1998) suggested to classify the use of mixed languages in Arabic as a separate phenomenon and not code-switching. Others termed it 'mixed Arabic', see (Davies et al.,2013). We will use 'language mixing' to refer to the 'code-switching' phenomenon.

21 To make it easy to read for non-Arabic speakers, we use Buckwalter Arabic transliteration scheme. The complete chart is shown in Appendix A.

among the supporters of this president]. In all other Arabic dialects, it means [people are crazy because of the supporters of this president]. Another sentence: بدى اروح [bdY ArwH]. In Levantine Arabic, it means [I want to go] whereas in Maghrebi Arabic it means [he/it starts to vanish/go]. These examples give an idea of the false friends between Arabic variants.

In terms of vocabulary, there are two types of words. Firstly, there are words which exist in both MSA and dialectal Arabic. These may keep their MSA meaning, and if so they are counted as vocabulary overlap. They may also keep only their word form and acquire new meanings depending on the Arabic dialect they are used in. For instance, the word الشعب [Al$Eb] which means [people] has the same meaning in MSA and dialectal Arabic. However, the word أبي [>by] means in MSA [my father which is a noun and in Gulf Arabic it means [I want] which is a verb. Secondly, there are words which are typically dialectal, i.e., they do not exist in MSA. In NLP applications such as event extraction, sentiment analysis/opinion mining and machine translation, it is important to know what the intended meaning of a given word is. Unfortunately, the context of the words is typically not enough for disambiguation. Only knowing the Arabic variant will determine the intended meaning. Grammatical, phonological and morphological differences do not hinder mutual understanding even though they are useful features in distinguishing some dialects from others. We will not focus on any syntactic or structural differences between MSA and other varieties because there is no syntactic parser or morpho-syntactic analyzer which supports a wide range of dialects[22]. If there is any, they should be language dependent, i.e. we need first to know the language at hand to be able to analyze it properly. This is not our case because our goal is to detect the language itself.

22 There is MADAMIRA which, for now, supports only MSA and Egyptian. We choose not to use it because we want to have the same treatment for all variants.

# 4 System implementation

## 4.1 Linguistic resources

New technologies play a considerable role in preserving many marginalized and under-resourced languages, basically spoken languages, from disappearing[23] by providing them with platforms to document them, i.e. preserve them in written texts. Likewise, dialectologists and linguists will find considerable material to study. The use of Arabic dialects on the Web is a quite recent phenomenon characterized by the absence of freely available linguistic resources[24] which allow us to perform any automatic processing. The deficiency of linguistic resources for dialectal Arabic (DA) is caused by two factors "a lack of orthographic standards for the dialects, and a lack of overall Arabic content on the web, let alone DA content. These lead to a severe deficiency in the availability of computational annotations for DA data" (Diab et al., 2010). This is not surprising because Arabic dialectology is not even considered as an academic field in Arabic-speaking countries. Most of the studies done for Arabic dialectology are conducted by non-Arab researchers, mainly Europeans. Our task requires annotated data. To overcome this serious hindrance, we built linguistic resources from scratch consisting of dataset and lexicons for each dialect considered in this study. The following sections, in this chapter, describe the procedures of building these resources.

### 4.1.1 Dataset

In this subsection, we will describe in details how we collected the data and annotated it. We will also explain the measures used to evaluate the quality of the annotation along with the raw data pre-processing.

#### 4.1.1.1 Dataset building and annotation

Usually Arabic dialects are used to communicate in different social media platforms and to express opinions on forums or blogs[25]. They are also widely used in commenting on events or news on news agencies' websites. We have collected manually around 100 to 150 documents for each dialect using its dialectal vocabulary[26]. Table 4.1 gives an idea on how the very first dialectal lexicons look like. We have compiled a list of popular websites which contain dialectal content covering a wide range of topics such as popular TV shows/event in the corresponding Arabic-speaking countries. Next, we have

23 This was not the case of many minority spoken languages which disappeared without being documented, i.e. no written trace has been left, for instance Arabic varieties used in the pre-Islamic period.

24 There are some collections by individuals but unfortunately not digitalized or do not respect corpus linguistics annotation conventions. These collections were used first by dialectologists.

25 There is no statistics done in this direction but only compared to the content of other websites such as news, official orga - nization and institute where only MSA is used.

26 Based on our dialectal Arabic knowledge, we compiled manually a list of special dialectal vocabulary for each dialect. This contains mainly prepositions, question words, personal pronouns, verbs and adjectives.

asked native speakers, two people for each dialect, to collect more data using the already compiled collection as seeds and the list of websites as a start. Of course, they are encouraged to collect data from other websites given that the data is clearly dialectal. Our purpose in doing this is to provide some guidelines and show what kind of data we are aiming at collecting. Ideally we would have looked at just some data resources and harvest content as much as possible either manually or by a script. But given the fact that data depends on the platform it is used in[27] and our goal that is to build

| Dialect | ALG | EGY | GUL | KUI | LEV | MOR | TUN |
|---|---|---|---|---|---|---|---|
| Dialectal Vocabulary | واش | بتوع | ذلحين | ماكو | منشان | ديال | باش |
| | راك | ده | تبي | هسه | شو | واش | فماش |
| | كيما | كده | ليش | هيج | متل | بزاف | زوز |
| | باش | دى | وش | احنه | هيك | راه | علاش |
| | راه | عشان | عشان | اكو | بدي | باش | باهي |
| | بصح | زى | الحين | كولش | بدك | غادي | توا |
| | راني | مينفعش | مب | شلون | مافى | وخا | برشا |
| | خطرة | ومعيش | فديتك | يحجي | كمان | دابا | كيفاش |
| | راهو | كدة | مهوب | علمود | لسى | فاش | دبوزة |
| | علاش | عايز | ويش | جان | الحكي | هادشي | شنوا |
| | راهم | دلوقتي | مدري | شكد | منيح | ديالك | تنجم |
| | بزاف | مفيش | يبون | هذوله | ليش | داك | فمة |
| | نتاع | عاوز | وهاذول | عليج | هون | مزيان | هكا |
| | راكم | علشان | شلون | لويش | هلا | علاش | شنوة |
| | راهي | محدش | زين | ينطيج | عنجد | داكشي | قداش |
| | شكون | هتبقى | شنو | هيجي | مبارح | عفاك | هذاكا |
| | كيفاش | ازاي | موب | عندج | مابدى | زوين | كرهبة |
| | تهدر | كويس | ربعي | شكو | هيدا | كيفاش | عالخ |
| | شحال | دلوقتى | انزين | مالته | لعند | قاليها | ماهياش |
| | شياتين | النهارده | الريال | وايد | ضلي | كنبغيك | ماريتش |
| | ..... | ..... | ..... | .... | ..... | ..... | ..... |

Table 4.1: Dialectal vocabulary snippet used to retrieve data

a general system which will be able to handle various domain/topic independent data, we have used various data domains dealing with quite varied topics like cartoons, cooking, health/body care, movies, music, politics and social issues. We made sure to include content from all the topics for each dialect. We also give some instructions such as:

- Collect only what is clearly written in your dialect, i.e. texts containing at least one clear dialectal word and you can easily understand it and reproduce the same in your daily interactions.
- Keep the original texts without any editing.
- Include only texts and not the user's information.

Here we feel the need to explain further the first instruction. As pointed in Chapter 3, Arabic-speaking societies are multilingual and Arabic itself is a mixture of languages[28], we consider multilinguality as a main characteristic of dialectal Arabic. “A prominent aspect of Arabic is that it is in contact not only with other languages, the situation underlying codeswitching, but also as it were, with itself” (Davies

27 For instance the use of some special markers in some platforms and the allowed length of the texts. S horter text means more abbreviations.

28 Plus the extensive borrowing of expression/word from almost all languages

et al., 2013). This means that it is very rare, if at all, to find texts written only in one of the high level seven (7) dialectal groups considered in this study. A dialectal Arabic text is a mixture between MSA and any of the Arabic dialects. Given the fact that MSA overlaps with all Arabic varieties and that some European and other indigenous languages are commonly used for historical reasons depending on the region, we have allowed mixed data following some priorities. For instance, if a text contains MSA and dialectal vocabulary, the entire text is considered to be in the corresponding dialect. If a text contains dialectal vocabulary and some words in European languages either in Arabic script or another script, then the document is in dialectal Arabic. In case a text contains dialectal words along with words in an indigenous language[29], then it is not dialect Arabic. In all cases, we ignore the Named Entities (NE) such as people, organization, product, company and country names.

For now, the hardest case to deal with is a mixed data between clearly two or more Arabic dialects like when quoting someone. This is still an unsolved issue for the automatic language identification task. Instead of deciding in which language a text is written, researchers are talking about 'language mixing' detection. An even more refined solution would be to introduce a mixed-category, for instance say 'the text is written in Algerian and Tunisian dialects' or to segment instead of classify. This is out of our scope in this project because it would require tremendous efforts for collecting and annotating data. In our case, it does not matter if a mixed text is classified in either dialects as long as it contains clear vocabulary in that dialect. We have allowed such mixed data even though it causes some noise in our data particularly between very close dialects like Maghrebi or Gulf/Mesopotamia dialects. The reason is that the real data is mixed so there is no point in picking out only clear-cut cases.

We have also used a script with the dialectal vocabulary, shown in Table 4.1, as keywords to collect more data. We have collected 1 000 documents for each dialect, roughly published between 2012-2016 in various platforms (micro-blogs, forums, blogs and online newspapers) from all over the Arab world. The same native speakers have been asked to clean the data following the same set of instructions. We ended up with an unbalanced corpus of between 2 430 – 6 000 documents for each dialect.

In terms of data source distribution, the majority of the content comes from blogs and forums where users are trying to promote their dialects; roughly 50%, around 30% from popular TV-show YouTube channels and the rest is collected from Twitter and Facebook. The selection of the data sources is based on the quality of the dialectal content, in other words, we know that the content of the selected forums and blogs is dialectal which is used to teach or promote some dialects between users. Further, it is easy to collect content from these platforms without signing up or knowing in advance some particular users account. The dialectal data collection took us in total two months. The included documents are short, between 2 and 250 tokens, basically product reviews, comments and opinions on quite varied topics. However, whatever the data source the dialectal content is the same except for the allowed text lengths and some platform special markers[30]. This is not an issue as we take care of it in the pre-processing step.

As described above, the data collection has been done separately, each dialect content has been separately collected. This comes in handy for the annotation process which is seen as a categorical classification, i.e. attribute a given label from a pre-defined set of labels to some document. We picked up 2000 documents for each dialect and assigned them the corresponding label. In addition to this dialectal corpora, we have added 2 000 documents written in MSA which we have collected from a freely available book (a collection of short stories) and various newspapers websites. We made sure that we included various topics. We assume that any educated Arabic-speaker can easily spot MSA

29 This refers to any indigenous language depending on the country, for instance, Iraq ( Kurdish, Assyrian, Armenian, Chaldean, Ashuri and Turkoman), Lebanon (Armenian), North Africa (Berber), Oman (Balochi), Syria (Kurdish, Armenian, Aramaic and Circassian).

30 Such as #, @, 'follow', 'retweet', 'posted originally by', 'reply', 'like'

from dialectal Arabic because MSA is the only formal/normalized variety which is orthographically, syntactically and stylistically different from dialectal Arabic. It is important to mention that MSA is also used in social media but commonly for limited topics particularly religion.

In North Africa, Berber or Tamazight [31], which is widely used, is also written in Arabic script mainly in Morocco, Algeria and Libya. Arabicized Berber or Berber written in Arabic script is an under-resourced language and unknown to all available automatic language identification tools which misclassify it as Arabic (MSA)[32]. Arabicized Berber does not use special characters and it coexists with Maghrebi Arabic where the dialectal contact has made it hard for non-Maghrebi people to distinguish it from local Arabic dialects[33]. For instance each word in the Arabicized Berber sentence أحمل ساقول ماشي دول كان [>Hml sAqwl mA$y dwl kAn] which means [love is from heart and not just a word] has a false friend in MSA and all Arabic dialects. In MSA, the sentence means literally [I carry I will say going countries was] which does not mean anything. This motivates us to add it a separate category referred to as 'BER'. We collected 503 documents from north African countries mainly from forums, blogs and Facebook. For more data, we have selected varied texts from Algerian newspapers and segment them. Originally the news texts are short, 1 500 words each. So we have considered each paragraph as a document (maximum 178 words). Then, we have added 1 497 documents to the ones collected from social media to get in total 2 000 documents tagged as 'BER' which we added to our dialectal Arabic collection. We could have collected all the Arabicized content from forums and blogs, etc. However because of time limitation, we used newspapers content instead. Another motivation to do so is that Berber also has many varieties, likewise we made sure to include content from most of them. In total, we have collected 18 000 documents, 2 000 for each category (the 7 dialects, MSA and BER).

As a side task, we want to make difference between Arabic and non-Arabic[34] texts written in Arabic script plus those written in any other script[35]. Ideally, we want to say that any text written in a non-Arabic script is written in an unknown language. To be able to do so, we have created a dataset of 2 000 documents containing short texts in different languages and scripts which we tagged as 'UKN' and added them to the previous collection (18 000 documents). We choose not to include Arabicized Berber in the 'UKN' category because as a minor goal, we want to build a language identifier for Arabicized Berber as well. Another motivation is that BER is still unknown language to the available automatic language identifiers unlike Pashto, Persian and Urdu (languages using Arabic script) for which language identifiers exist.

Table 4.2 shows some statistics about the entire dataset. In the presented figures as well as in the next experiments, we do not count unimportant words including punctuation, emoticons, any word occurring in the MSA data more than 100 times (prepositions, verbs, common nouns, proper nouns, adverbs, etc.) and Named Entities (NE). Removing unimportant words is motivated by the fact that these words are either prevalent in all Arabic varieties or they do not carry any important linguistic

31 Berber or Tamazight is an Afro-Asiatic language widely spoken in North Africa and different from Arabic. It has 13 varieties and each has formal and informal forms. It has its unique script called Tifinagh but for convenience Latin and Arabic scripts are also used. Using Arabic script to transliterate Berber has existed since the beginning of the Islamic Era, see (L. Souag, 2004) for details.

32 Among the freely available language identification tools, we tried Google Translator, Open Xerox language and Translated labs at http://labs.translated.net.

33 In all polls about the hardest dialect to learn, Arabic speakers mention Maghrebi Arabic which has Berber, French and words of unknown origins which is not the case of other Arabic dialects.

34 Pashto, Persian and Urdu for instance.

35 Arabic dialect is also written in Latin script or what is known as Romanized Arabic (RA). The removal of Latin script words will filter also any potential RA words. We assume that this is not an issue since RA is mainly used because of the non-availability of the Arabic keyboard. It does not make any sense to mix scripts. If it does exist, then it should be rare.

information like emoticons and punctuation. This makes them very weak discriminants, hence it is better to remove them for the sake of saving memory and making the system faster. The choice of removing NE is motivated by the fact that they are either dialect (region) specific or prevalent; i.e. they exist in many regions so they are weak discriminants. Moreover, we want the classifier to be robust and effective by learning the language variety and not heuristics about a given region. We would have presented the documents by topic distribution as well, but we did not keep track of that as it requires more human effort. Also the mixed topic texts make it hard to give accurate figures. Given the total number of documents is #Documents and the total number of tokens per document is #Tokens, the document average length (Av. Length) is computed as follows:

$$Av.\,Length = \frac{\#\text{Tokens}}{\#\text{Document}}$$

| Language | ALG | BER | EGY | GUL | KUI | LEV | MOR | MSA | TUN | UKN |
|---|---|---|---|---|---|---|---|---|---|---|
| **#Document** | 2 490 | 2 320 | 2 430 | 2 519 | 6 000 | 2 673 | 3 800 | 9 884 | 2 864 | 2 000 |
| **#Tokens** | 31320 | 69850 | 42071 | 83240 | 94702 | 69792 | 44928 | 235818 | 79749 | 65349 |
| **#Types** | 17382 | 25183 | 21007 | 35065 | 34856 | 28568 | 21541 | 81791 | 32004 | 29616 |
| **Av. Length** | 12.57 | 30.10 | 17.31 | 33.04 | 15.78 | 26.10 | 11.82 | 23.85 | 27.84 | 32.67 |

Table 4.2: Dataset statistics

Arabicized Berber (BER) and the unknown category (UKN) have both long documents (big Av. Length) because the removal of all words occurring more than 100 times in the MSA data does not have a big effect as they have different vocabulary.

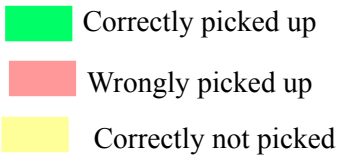
#### 4.1.1.2 Evaluation of the dataset annotation

To assess the reliability of the annotated data, we have conducted a human evaluation. As a sample, we have picked up randomly 100 documents for each language from the collection (18 000 documents) removed the labels, shuffled and put all in one file (900 unlabeled documents in total). We asked two native speakers for each language, not the same ones who collected the original data, to pick out what s/he thinks is written in his/her dialect, i.e. can understand easily and can produce the same in his/her daily life. All the annotators are educated, either have already finished their university or still students. This means that all of them are expected to properly distinguish between MSA and dialectal Arabic. Our two MSA annotators are both pupils at secondary school. The results are collected in Table 4.3 which is read as follows: from the 900 documents the first Algerian (ALG) annotator picked out correctly all the Algerian documents in the collection plus 3 Egyptian, 24 Moroccan and 31 Tunisian documents. The second Algerian annotator correctly picked out 93 Algerian documents only.

Arabicized Berber is completely different from Arabic that is why it is easily spotted, namely for educated Berber speakers (it is thought at school). Likewise, MSA is easy to detect from other varieties that is the reason why none of the speakers picked it out as a dialect. For all Arabic dialects, except Egyptian, there is a difference between the two annotators because all of them are from different regions and also each one has his/her individual variation. The difference is clearly seen for both GUL and LEV. As mentioned in Chapter 3, these dialects are actually a group of regional close dialects. Also, not all the annotators picked out 100 documents in their dialects. This is due to the fact that there are many local dialects using different vocabulary. For instance in Algeria, a person from western part understands another from the eastern part even though they use different words.

| | | Picked up documents/language | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | ALG | EGY | GUL | KUI | LEV | MOR | TUN | BER | MSA |
| Native speakers of each language | ALG | 100 | 3 | 0 | 0 | 0 | 24 | 31 | 0 | 0 |
| | | 93 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | EGY | 2 | 100 | 17 | 0 | 6 | 0 | 0 | 0 | 0 |
| | | 1 | 100 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| | GUL | 0 | 0 | 96 | 21 | 13 | 0 | 0 | 0 | 0 |
| | | 0 | 6 | 83 | 0 | 2 | 0 | 0 | 0 | 0 |
| | KUI | 0 | 0 | 27 | 100 | 0 | 0 | 0 | 0 | 0 |
| | | 4 | 0 | 5 | 97 | 0 | 0 | 0 | 0 | 0 |
| | LEV | 0 | 0 | 19 | 0 | 100 | 0 | 0 | 0 | 0 |
| | | 0 | 7 | 11 | 0 | 89 | 0 | 0 | 0 | 0 |
| | MOR | 10 | 0 | 0 | 0 | 0 | 100 | 14 | 0 | 0 |
| | | 9 | 2 | 0 | 0 | 0 | 92 | 9 | 0 | 0 |
| | TUN | 34 | 0 | 0 | 0 | 0 | 13 | 98 | 0 | 0 |
| | | 16 | 0 | 0 | 0 | 0 | 7 | 96 | 0 | 0 |
| | BER | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 |
| | MSA | 0 | 5 | 9 | 0 | 0 | 0 | 0 | 0 | 100 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

Table 4.3: Native speaker annotation




The vocabulary difference is a good indicator of the region/location of a person. It is hard to assume that everyone is familiar with all the dialects spoken in his/her country/region. That is one possible explanation why some annotators missed some documents written in their high level dialect group. More or less, the 2nd annotators confuse less there dialects with others compared to the 1st annotators.

The reason is that the 2nd group of annotators are trained linguists (Master's students of Arabic linguistics and literature) we hired them because they are familiar with many dialects. The confusion is mainly between very close dialects like ALG, MOR and TUN (Maghrebi dialects), GUL and Mesopotamian dialects. The results are expected and perfectly reflect the linguistic situation of Arabic varieties particularly neighboring ones. The task is even harder for short documents in the absence of typical dialectal vocabulary.

To interpret the results, we compute the inter-annotator agreement for each language to see how often the annotators agree. Since we have two annotators per language, we compute the Cohen's kappa coefficient (κ) which is a standard metric used to evaluate the quality of a set of annotations in classification tasks by assessing the annotators' agreement. " κ measures pairwise agreement among a set of coders making category judgments, correcting for expected chance agreement" (J. Carletta, 1996). It is computed as follows:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \qquad (1)$$

where P(A) is the estimated probability of agreement and P(E) is the chance agreement, i.e. what is the probability that the two independent annotators agree by chance.

We take Algerian dialect as an example to show how we compute the κ[36]. We convert the classification into a binary categorization, i.e. is it ALG or other (including all the other dialects). We get Table 4.4.

| | | **Annotator # 1** | | |
|---|---|---|---|---|
| | | **ALG** | **OTHER** | |
| **Annotator # 2** | **ALG** | 93 | 0 | 93 |
| | **OTHER** | 65 | 742 | 807 |
| | | 158 | 742 | 835 |

Table 4.4: ALG binary classification

First, we compute the probability that the two annotators agree (both say either ALG or OTHER) by summing the number of times they agreed on and divided it by the total number of documents in the dataset.

$$P(A) = \frac{93 + 742}{900} = 0.927$$

Second, we sum the columns and the rows and multiply them for each case (ALG and OTHER), and then we divide all by the total number of documents in the dataset.

$$P(E) = \left(\frac{93 * 158}{900} + \frac{807 * 742}{900}\right) * \frac{1}{900} = 0.757$$

36 We followed the method explained in http://epiville.ccnmtl.columbia.edu/popup/how_to_calculate_kappa.html

Finally, we substitute the values in the equation (1):

$$\kappa(Algerian)=\frac{0.927-0.757}{1-0.757}=0.70$$

For the Precision and Recall, we take the average of the tow annotators:

$$Precision(Algerian)=(\frac{100}{158}+\frac{93}{93})*\frac{1}{2}=0.816$$

$$Recall(Algeria)=(\frac{100}{100}+\frac{93}{100})*\frac{1}{2}=0.965$$

Likewise, we compute the kappa, Precision and Recall for the rest of languages. The results are shown in Table 4.5.

| Dialect | ALG | BER | EGY | GUL | KUI | LEV | MOR | MSA | TUN |
|---|---|---|---|---|---|---|---|---|---|
| **Kappa (%)** | 70 | 100 | 89 | 72 | 83 | 81 | 83 | 92 | 78 |
| **Precision (%)** | 81.60 | 100 | 88.54 | 82.52 | 85.12 | 83.60 | 81.39 | 93.85 | 73.83 |
| **Recall (%)** | 96.50 | 100 | 100 | 89.50 | 98.50 | 94.50 | 96.00 | 100 | 97.00 |

Table 4.5: Kappa/Precision/Recall for each language

The results in Table 4.3, which are reflected in Table 4.5, indicate that the annotators confuse mainly between Algerian, Moroccan and Tunisian dialects which belong to the same regional group, namely Maghrebi Arabic. They confuse also between Gulf and Mesopotamian dialects. This is related to the fact that the Gulf dialect category contains actually a wide variety of dialects with no clear-cut borderlines between them. Another reason is that the name 'Gulf' itself is misleading and has different interpretations: does it refer geographically to the Arabian Peninsula which includes parts of Iraq and Jordan, or does it refer only to the political and economic alliance called Gulf Cooperation Council (GCC) which excludes both Iraq and Jordan. Still, there is no a satisfactory answer to this question and the absence of linguistic information makes it even harder to properly classify these dialects. Egyptian and Levantine Arabic share some syntactic structures such are the use of 'b' to mark the verb progressive mode. Algerian, Egyptian and Mesopotamian mainly share the way the negation is expressed in addition to some common vocabulary. However the confusion with MSA is mainly caused by false friends.

Generally, we can say trustfully that the data quality is 'satisfactory' for Algerian, Gulf and Tunisian dialects by interpreting the kappa metric which is between 0.6 – 0.8. The quality of the rest of the dialectal data is 'really good', kappa 0.8 – 1. These are the conventional Kappa ranges which are kind of arbitrary but commonly used in measuring the quality of the annotated data. The credibility of the kappa itself is questionable when it comes to nontrivial linguistic tasks like discriminating between Arabic dialects which are themselves a group of varieties. For instance, an Arabic native speaker from Baghdad is not necessary familiar with all the other dialects spoken in Iraq. We need to have ensure that all the annotators are in the same condition, i.e. from the same region and speak exactly the same language. This is hard to ensure because there are many local dialects and individual variations.

Uebersax (1987) explained why it is inappropriate to rely on the kappa and the necessity to look for new methods for measuring inter-annotator agreement.

#### 4.1.1.3 Data pre-processing

As discussed in Chapter 3, 'real Arabic' or dialectal Arabic is any informal language which does not adhere to the MSA grammar and uses non-standardized orthography and language mixing in both Arabic and Latin scripts mainly. Further, social media language is characterized by the use of short texts, particularly in micro-blogs for technical reasons, the use of emoticons and special markers[37]. To optimize the classification task, we have pre-processed the collected data, i.e. introducing tokenization, normalization rules and removal of unimportant tokens[38].

- **Tokenization:** the default tokenization is whitespace-based which includes punctuation and other tokens with words. To keep only the Arabic words, we introduce a space between punctuation, digits, emoticons, special markers, Latin script characters and tokenize on whitespace again.

- **Normalization:** users use emphatic lengthening to emphasis their opinions by doubling characters. For instance in سييت لروسات نتاع لتارت استرو جات منيفيييييك [syyt lrwsAt ntAE ltArt Astrw jAt mnyfyyyyyk] which means [I tried the recipe of lemon pie and it was greaaaaat]. We reduced all the adjacent repeated characters into two occurrences.

- **Token filtering:** we remove all non Arabic words to keep only the Arabic content. This concerns digits, emoticons, non-Arabic script, special markers and NE.

- **Vowel removal:** most of the collected data is unvoweled, i.e. missing the diacritics or Arabic vowels, however there are some partially voweled documents in some dialects. We think it would be biased if we keep them as distinguishing features for the dialects they do occur in since they should occur in all dialects depending on the user. To unify the data, we remove all vowels. For instance, the fully voweled sentence أَنَا سَعِيدٌ بِقُدُومِ الرَّبِيعِ [<naA saEiydo biquduwmi Alra~biyEi] which means [I'm happy with coming of spring] means the same thing as the unvoweled sentence أنا سعيد بقدوم الربيع [>nA sEyd bqdwm AlrbyE] and does not cause any understanding problems.

### 4.1.2 Dialectal lexicons

As mentioned in the data building section, we have manually compiled a list of dialectal words, based on the author of this thesis knowledge, which we have used to retrieve some data using a script[39]. At this level, we have used the remaining data, after removing the 18 000 documents (2 000 for each language without counting the UKN), and extracted all the unique vocabulary for each dialect, using a script, to build dialectal lexicons. We have also added some dialectal lexicons we collected from some exchange forums where individual users were trying to promote their culture and dialects. The reason

37 Such as #, @, 'follow', 'retweet', 'posted originally by', 'reply', 'like'

38 It would be even better to introduce some spelling correction rule for the most frequent misspellings such as the 'أ' [hamza] letter which is spelt as a simple 'ا' [alif] and the 'ة' [ta marbota] which is a final feminine marker misspelt usually as 'ه' [h].

39 We used the words of the compiled lists as keywords in our research to access some Web platforms using the Client library for the Recorded Future API available at https://github.com/recordedfuture/api/wiki/RecordedFutureAPI

we have done so is the desperate lack of digitalized dialectal lexicons[40] and the few available ones are outdated word lists in paper format. For MSA, we have used the content of two freely available books. We would have also used some MSA dictionary, but this would need more effort as the freely available dictionaries are not designed to be easily used for any computational purpose. As pointed out before, we filtered unimportant words. Table 4.6 shows the size of each dialectal lexicon plus MSA.

| **Language** | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
|---|---|---|---|---|---|---|---|---|---|
| **#Types** | 9 980 | 22 738 | 6 134 | 10 620 | 10 564 | 10 450 | 100 230 | 12 344 | 13 693 |

Table 4.6: Sizes of lexicons

In order to have even more refined lexicons, as a second lexicon processing step, we use Term Frequency-Inverse document Frequency (TF-IDF) to measure the importance of each word to each dialect. TF-IDF, a word/ feature importance weighting scheme, is computed as follows:

$$TF = \frac{T}{N} \qquad (1)$$

In equation (1), T is the number of times s term appears in a document and N is the total number of words in the document. Equation (1) means if a word appears frequently in a document, it is important so assign it a high score.

$$IDF = \log\left(\frac{ND}{Nt}\right) \qquad (2)$$

In equation (2), ND is the total number of documents and Nt is the number of documents containing a term t. Equation (2) means if a word (term) appears in many documents, then it is no more informative, so we assign it a low score.

$$TF - IDF = TF * IDF \qquad (3)$$

Equation (3) scales up frequent words appearing in a single document or few documents and scales down words appearing in many documents. Therefore, only informative words will be considered which means that most misspellings will be discarded as they may occur in all varieties. Still, some misspellings are seen only in some variants. In this case, we have manually checked the lexicons and removed all misspellings taken the MSA orthography as reference. Table 4.7 gives some statistics about the dialectal lexicons after applying TF-IDF.

| **Language** | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
|---|---|---|---|---|---|---|---|---|---|
| **#Types** | 9 172 | 21 786 | 5 979 | 10 349 | 10 272 | 9 969 | 88 361 | 11 879 | 13 101 |

Table 4.7: Statistics of lexicons applying TF-IDF

40 “For many regions no substantial dictionaries are available. We have reasonable dictionaries for Levantine, Algerian and Iraqi, but these are sometimes outdated and need to be replaced or updated” (Behnstdt & Woidich, 2013).

We want also to emphasis the fact that Arabic variants use an extensive language mixing, particularly in north African societies. We have included frequent arabicized French words. We think they are good discriminants between Maghrebi and Mashriqi (North Africa and Middle East) blocs. Actually, some of these words may also occur in Mashriqi Arabic, but less frequently, like ميرسي [myrsy] which is [thank you] in French, merci.

## 4.2 Approaches

We will experiment with both supervised and dictionary-based methods. For the supervised approach, which requires annotated data for training, we will use machine learning methods (Cavnar's method, a variety of classifiers implemented in Scikit-learn package and Prediction by Partial Matching (PPM) method).

### 4.2.1 Machine learning methods

Cavnar's Text Categorization Character-based *n*-gram method is one of the automatic language identification (ALI) statistical standard methods. It is a collection of the most common character-based *n*-grams[41] used as a language profile (Cavnar and Trenkle, 1994). For each language, we create a character-based *n*-gram profile (including different lengths of *n*-gram where the value of *n* ranges between 2-5), sort it and consider only the most common 300 *n*-grams. This choice is for practical reason which is explained by the fact that at some point, the frequency of some *n*-grams is more or less the same for all languages. Therefore, they are not informative, i.e. do not really represent a given language and cannot be used as distinctive features to distinguish each language from others. The distance between language models is defined as the sum of all the out-of-place scores[42]. At the end, the language with the minimum distance from the source text will be the identified language.

We will experiment with different classifiers from Scikit-learn package (F. Pedregosa et al., 2011)[43], namely the KNeighbors classifier (KN) implementing the k-nearest neighbors vote, Decision Tree classifier (DT), Naive Bayes classifier for multinomial models (NB), Logistic Regression (LR) and Support Vector Machines SVM, namely LinearSVC (SVM). These methods as well as Cavnar's method require a category label for each document in the training set. The category labels are the nine (9) languages we are interested in: ALG (Algerian), BER (Berber), EGY (Egyptian), GUL (Gulf), KUI (Mesopotamian), LEV (Levantine), MSA, MOR (Moroccan) and TUN (Tunisian). For features[44], we will use character-based *n*-grams and word-based *n*-grams of different length, the number of dialectal vocabulary (count how many dialectal words are included in each document) using the compiled dialectal lexicons and finally combine all the features. Character-based *n*-gram means taking small pieces (letters) of the text including the white space. Word-based *n*-gram is a bit longer piece of text where we take the entire words (sequence of letters between two white spaces) and combining different sequences of words as information units. The dialectal vocabulary feature means considering only the dialectal words in the text using the compiled dialectal lexicons. For measuring the

41 A sequence of *n* characters from a given sequence of text where *n* is an integer.

42 Computing the distance between the ranking of the *n*-gram lists. The out-of-place score of an *n*-gram which keeps its ranking is zero. Otherwise, the out-of-place score is the difference between the two rankings.

43 For more information see: http://scikit-learn.org/stable/.

44 Piece of information useful for prediction.

importance of the features, we will experiment with both simple term frequency (TF) and TF-IDF schemes. Our purpose is to investigate how these different settings will perform in distinguishing Arabic varieties from each other.

Prediction by Partial Matching (PPM) is a lossless compression algorithm which has been applied to various tasks for instance text classification (Teahan & Harper, 2003; Zippo, 2012) and language identification (Bobicev, 2015). The core idea of the PPM method is to encode all the symbols of the training data within their context (for each symbol or character, a context is a sequence of preceding symbols of different lengths[45]). The symbols can be either words or characters. Therefore, it is a language independent method which does not require any prior data pre-processing or feature selection. Moreover, it considers the entire text as a single string where all characters are in lower case. It uses a simple blending strategy called 'escape event' to create the probability distribution of each symbol by combining all its context predictions. Each symbol probability is estimated from the probabilities of its context in a descending order (the propriety is given to longer contexts). PPM uses various blending mechanisms depending on the weighting of the 'escape event'. The simplest one is to assign a uniform low probability for all unseen characters and consider the probability of the seen ones. To simplify things, take an example. Assume that we have the string 'dialectal'. The probability of the symbol 'c' in the 6$^{th}$ position in maximum context length of 4 is computed as follows:

$$P('c') = \lambda_4 * P('c'|'iale') + \lambda_3 * P('c'|'ale') + \lambda_2 * P('c'|'le') + \lambda_1 * P('c'|'e') + \lambda_0 * P('c')$$

where $\lambda_0$, $\lambda_1$, $\lambda_2$, $\lambda_3$ and $\lambda_4$ are assigned normalization weights (the longer the context the higher the weight). In case a symbol in a new document is not seen in the training data, an 'escape event' probability is assigned. In this study, we will implement the benchmark escape method called 'C' (Moffat,1990) and take the maximum context of 5 symbols. With the help of Bobicev, we will implement a character-based method called PPMC5 as described in (Bobicev, 2015). Once the prediction models are built in the training phase, the per-symbol cross-entropy is measured to compute the similarities between the texts (documents used in the training dataset for each language). Intuitively, the lower the cross-entropy (less new information between the two texts) the more similar the texts are. After computing the cross-entropies between all languages, the language of the text with lower score wins.

### 4.2.2 Dictionary-based method

This method consists in using some language specific words as a lexical representation for the given language. It is based on the vocabulary overlap, i.e. compute the sum of the overlap words after a dictionary lookup for each language and the language with more word overlap will be returned. It would be easier to use blacklisted[46] words as discriminants as has been done for some close languages (Ljubešić, 2007). However, this does not apply to Arabic variants because there are no blacklisted words. The easiest possible way is to use the compiled dialectal lexicons as lexical profiles for each Arabic variant. But some words belong to several varieties and it is hard to find strong indicative words for each variety, as discussed in Chapter 3. Therefore, the main challenge is how to define the relevance of each dialectal word. "The problem of whether some isoglosses or variables should be given more weight than others in this procedure is still unsolved in theory and the researchers follow their own intuitions" (Behnstdt & Woidich, 2013).

45 Many previous works have reported that taking the context of 5 characters is the best maximum context length. This makes a perfect sense because the same long matches are less frequent or what is called data sparsity.

46 A list of forbidden words in some countries, regions or communities

De Jong (2000) applied a statistical method called 'step-method' to distinguish between some Arabic variants in Egypt. He computed the distance between dialects based on the vocabulary overlap, i.e. dialects sharing more dialectal vocabulary are close and vice versa. The results were good but the method needs to be applied to a wide range of varieties for an objective evaluation. A better refined method is the one suggested by dialectometry[47] where all variables have the same importance, therefore they are attributed the same weight. In our case, since we do not have really a good way to measure the importance of each dialectal word, so we will use a simple quantitative approach. We will divide the dialectal words into two sets: strong dialectal features which consist of the few words existing only in one variant, and the remaining words are included in the weak dialectal features. Assuming that the total number of correctly identified documents is TP, and the total number of documents misidentified for each language is FP, and the total number of documents which are not labeled as belonging to a correct given category is FN. Then Precision, Recall and F-score are computed as follows:

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

$$Recall = \frac{TP}{(TP + FN)}$$

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

47 A branch of linguistics which studies a dialect quantitatively and computationally using statistics.

# 5 Experiments and result analysis

In this chapter, we describe the different experimental setups and analyze the results. For machine learning, we use a balanced[48] dataset containing 18 000 documents (2 000 X 9). In order to investigate the effect of the data pre-processing, stop words[49] and vowel removal, we pre-processed the 18 000 documents in different ways. Table 5.1 shows the four pre-processing types.

| Dataset | Pre-processing |
|---|---|
| **A** | No pre-proccesing (raw data) |
| **B** | Full pre-processing |
| **C** | Partial pre-processing: keep vowels |
| **D** | Full pre-processing + stop words removal |

Table 5.1: Data pre-processing types

Each dataset, regardless of the pre-processing type, is divided into two sets:

- 80% or 14 400 documents (1 600 X 9) for training
- 20% or 3 600 documents (400 X 9) for testing

## 5.1 Cavnar's Text Categorization Character-based *n*-gram

We use Cavnar's classifier, as described in Chapter 4, with the four datasets in Table 5.1. Table 5.2 summarizes the classification results where we limited the maximum length of the texts to classify to 200 characters, and we used character-based 3-grams as feature. These parameters have been randomly chosen for the sake of showing the differences.

| Dataset | Accuracy (%) |
|---|---|
| **A** | 56.11 |
| **B** | 54.19 |
| **C** | 53.88 |
| **D** | 59.50 |

Table 5.2: Cavnar's classifier with different dataset pre-processing

48 Containing the same number of documents for each category or language.

49 All words occurring in the MSA dataset more than 100 times.

Comparing the classification results using dataset A and B shows that full pre-processing has slightly decreased the Cavnar's classifier accuracy. This might be caused by removing punctuation and emoticons which are used inconsistently in different dialects, i.e. used more in some dialects and rarely or not at all in others. Comparing B and C shows that keeping or removing vowels does not really matter maybe because they are not frequent. Comparing B and D shows that removing stop words has a good effect on the classification.

In the rest of the experiments, we will use dataset D (with full pre-processing and stop words removal) as a default dataset because it keeps the actual language words.

Now, we investigate the effect of the text length (number of characters) and *n*-gram length. Classification results are in Figure 5.1.

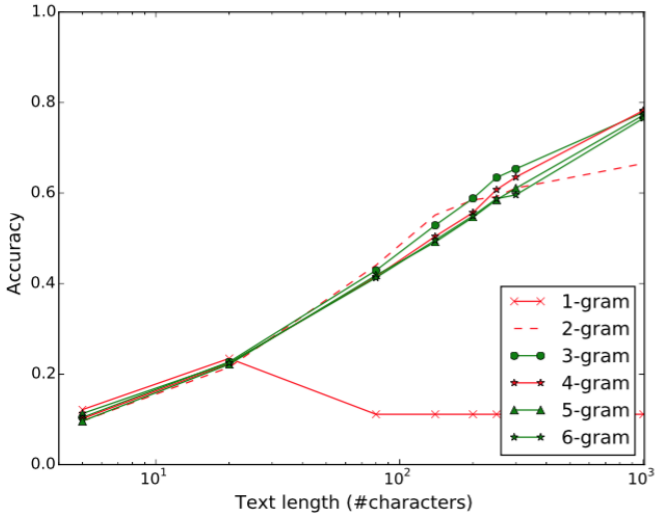



Figure 5.1: Effects of the *n*-gram and text lengths

For text length between 5 and 20 characters, all the *n*-grams have the same performance curve; the accuracy increases with the increasing of the text length. For text length between 20-200 characters, the accuracy also increases except for unigram where it gradually decreases and stabilizes at text length of 80 characters. Using bigrams as feature outperforms all the rest, 58.47% accuracy. After that, the classification accuracy increases with the increase of the text length, and 4-grams outperform the rest *n*-grams, 77.77% accuracy with maximum text length of 1 000 characters, and all *n*-grams greater than 2 perform almost the same.

The poor performance of the unigram is related to the fact that all the concerned languages use the same character set[50], i.e. no special characters with almost the same distribution. Error analysis shows that all the 3 600 testing documents are classified as MSA when the text length exceeds 20 characters.

To show the Cavnar's classification performance per language, we use dataset D, 3-grams as feature and maximum text length equals to 140 characters because it is the maximum length of a tweet. Table 5.3 shows the classification results.

50 Arabic alphabet contains 28 characters.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **ALG** | 41.34 | 37.00 | 39.05 |
| **BER** | 98.43 | 94.00 | 96.16 |
| **EGY** | 56.20 | 38.50 | 45.70 |
| **GUL** | 32.69 | 50.50 | 39.69 |
| **KUI** | 47.05 | 53.75 | 50.18 |
| **LEV** | 46.23 | 36.75 | 40.95 |
| **MOR** | 57.14 | 48.00 | 52.17 |
| **MSA** | 63.28 | 81.00 | 71.05 |
| **TUN** | 39.71 | 34.25 | 36.78 |

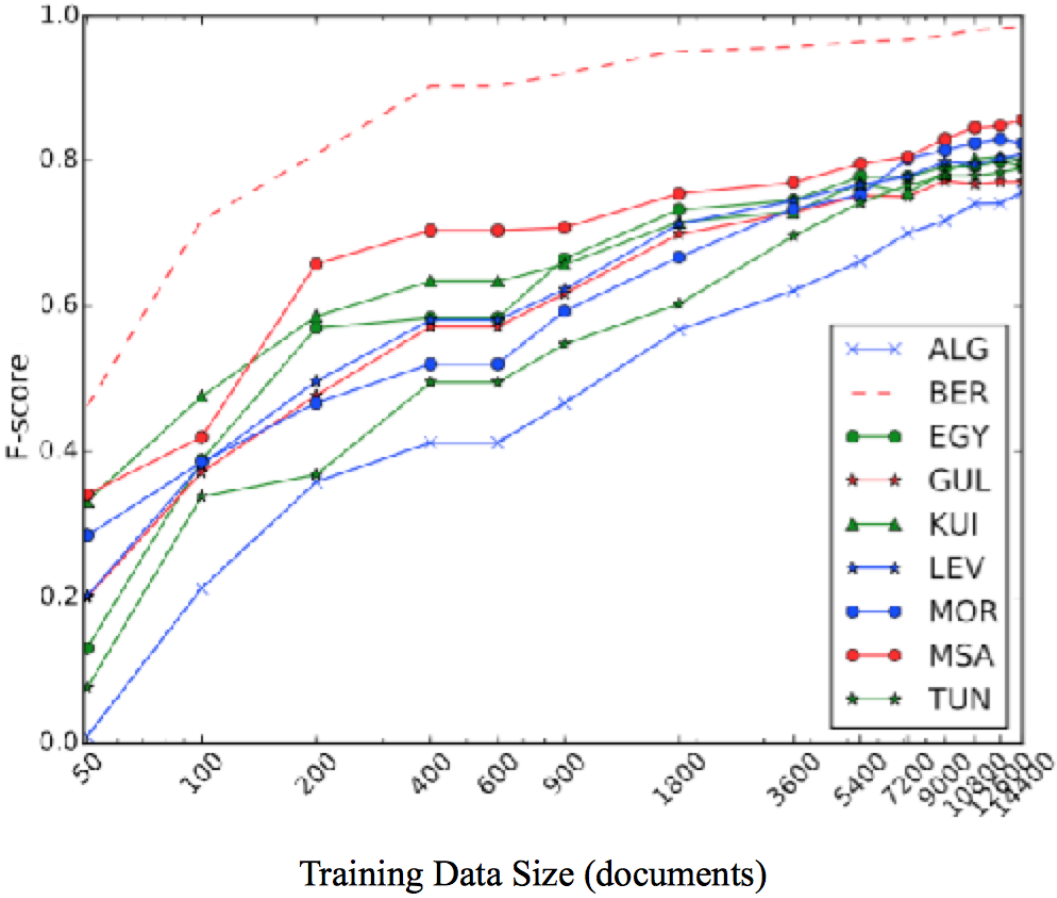
Table 5.3: Cavnar's classifier performance per language

The macro-average F-score of the Cavnar's classifier (using the mentioned settings) is 52.41%. Except for Arabicized Berber, the classifier finds it hard to distinguish dialects from each other even though it performs better in distinguishing MSA from dialectal Arabic. Table 5.4 shows the confusion matrix of the Cavnar's classifier for the same settings.

| | | **Misclassified languages** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Correct languages** | | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
| | **ALG** | 148 | 1 | 8 | 59 | 51 | 12 | 26 | 45 | 50 |
| | **BER** | 0 | 376 | 3 | 5 | 2 | 0 | 10 | 2 | 2 |
| | **EGY** | 20 | 2 | 154 | 62 | 35 | 41 | 49 | 9 | 28 |
| | **GUL** | 18 | 0 | 12 | 202 | 68 | 42 | 19 | 25 | 14 |
| | **KUI** | 18 | 0 | 7 | 110 | 215 | 21 | 13 | 3 | 13 |
| | **LEV** | 25 | 1 | 47 | 68 | 29 | 147 | 29 | 12 | 42 |
| | **MSA** | 4 | 2 | 10 | 20 | 15 | 4 | 324 | 2 | 19 |
| | **MOR** | 52 | 0 | 11 | 35 | 20 | 27 | 23 | 192 | 40 |
| | **TUN** | 73 | 0 | 22 | 57 | 22 | 24 | 19 | 46 | 137 |

Table 5.4: Cavnar's confusion matrix

We can see that Cavnar's classifier often confuses between Maghrebi dialects (ALG, MOR, TUN) and between GUL and KUI dialects. That is expected and accepted because, as discussed in Chapter 3, there are no dialectal clear-cut borderlines between neighboring dialects. In more details, there are more MOR and TUN documents (52, 73 respectively) confused with ALG ones compared to to the ALG documents confused with MOR or TUN documents, 45 and 50 respectively. This is applicable for KUI documents confused with GUL ones, 110 and 68 respectively. Figure 3.1 explains the reasons where it is impossible in practice to draw the dialectal borderlines.




The confusion between Maghrebi, Egyptian and Levantine dialects is related to the fact that some Levantine dialects (southern Syria and some parts of Lebanon, including Beirut) share the use of split-morpheme negations with Egyptian and North African dialects (Palva, 2006). It is also important to notice that while BER is rarely confused, MSA is confused with all dialects including BER.

All in all, for short texts of 140 characters or less, Cavnar's character-based classifier is not efficient in distinguishing Arabic varieties from each other, particularly the very close ones like Maghrebi dialects. Nevertheless, it performs better in discriminating between MSA and dialectal Arabic. Also, it distinguishes Arabicized Berber fairly well from Arabic.

## 5.2 Scikit-learn classifiers

In this section, we will experiment with various methods (classifiers) from Scikit-learn package along with different model construction and feature weighting. In all cases, we will use the default parameters[51]. Furthermore, we will use the binary classification setting as opposed to the 9-class classification. For instance, 'is a document written in MSA or something else (other Arabic variety)' as opposed to 'is a document written in MSA, ALG, BER, EGY, GUL, LEV, KUI, MOR or TUN.' Both classification settings will return only one label as an output because each classifier is implemented as

51 The default parameters for each classifier are detailed in http://scikit-learn.org/stable/.

a group of classifiers, and the category or the label with the highest prediction score will be returned each time.

### 5.2.1 Character-based *n*-gram

First, we will experiment with five classifiers. As a reminder, we will use KNeighbors classifier (KN), Decision Tree classifier (DT), Naive Bayes classifier (NB), Logistic Regression (LR) and Support Vector Machines SVM, namely LinearSVC which we will refer to simply by SVM.

We will use the dataset D, and as features we will use character-based 3-grams with simple frequency (TF) for a maximum text length of 140 characters. Table 5.5 shows the overall accuracy of the mentioned classifiers.

| Classifier | Accuracy (%) | Training time (sec) |
|---|---|---|
| **KN** | 55.38 | 2.500 |
| **DT** | 50.83 | 21.228 |
| **NB** | 74.50 | 2.584 |
| **LR** | 75.77 | 5.607 |
| **SVM** | 84.55 | 4.575 |

Table 5.5: Classifiers accuracies

The SVM classifier is faster and outperforms the other classifiers. Thus, we will use it in the rest of this subsection.

Now, we will investigate the data pre-processing effect along with the feature importance. We will use the same settings used with Cavnar's and see what will happen. Results are shown in Table 5.6.

| Dataset | Accuracy with TF (%) | Accuracy with TF-IDF (%) |
|---|---|---|
| **A** | 83.49 | 85.61 |
| **B** | 83.44 | 85.83 |
| **C** | 83.52 | 85.97 |
| **D** | 84.55 | 86.27 |

Table 5.6: SVM using different datasets and weighing schemes

Overall, data pre-processing has a very slight effect on the SVM classifier using TF, but still fully pre-processed data with stop words removal (dataset D) scores the best. However, data pre-processing has a slight effect on the SVM performance when using TF-IDF. This is expected as TF-IDF scales down frequent unimportant words and scales up important less frequent words. Also, it is clear that using TF-IDF is better than simply using TF.

Figure 5.2 summarizes the effect of the character-based *n*-gram and the text lengths on the SVM performance using dataset D and TF-IDF.

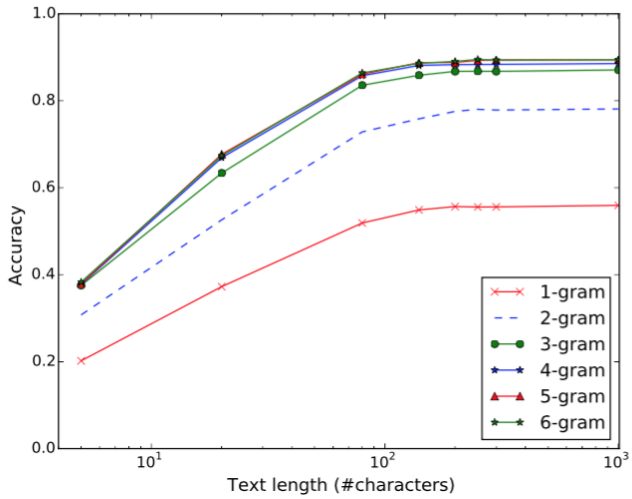



Figure 5.2: Effects of the *n*-gram and text lengths on SVM

Using character-based 4-grams scores the best in all cases as well as 5-grams or 6-grams but slower. Increasing the size of the text increases the classification accuracy in all cases as well. The classification performance stabilizes when text length equals 80 characters.

In another experiment, we combine the *n*-grams using the same experimental setups. We report some results in Table 5.7 for text maximum length of 140 characters.

Combining 5-grams and 6-grams has a positive effect on the SVM classifier accuracy. Other combinations decreases the classification accuracy or do not have any effect.

| ***n*-gram** | **Accuracy (%)** | **Combination accuracy (%)** |
|---|---|---|
| **2-gram** | 75.83 | 84.94 |
| **3-gram** | 86.27 | |
| **4-gram** | 88.11 | 88.33 |
| **5-gram** | 88.74 | |
| **5-gram** | 88.74 | 89.02 |
| **6-gram** | 88.61 | |

Table 5.7: SVM accuracy with *n*-gram combinations

To show the performance of the SVM classifier per language, we will use the same settings again and report the results, in Table 5.8 for text maximum length of 140 characters using the combination of 5-grams and 6-grams as feature.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **ALG** | 87.57 | 82.75 | 85.09 |
| **BER** | 100 | 99.50 | 99.75 |
| **EGY** | 94.74 | 81.00 | 87.33 |
| **GUL** | 82.38 | 86.50 | 84.39 |
| **KUI** | 89.61 | 92.75 | 91.15 |
| **LEV** | 88.95 | 76.50 | 82.26 |
| **MOR** | 89.76 | 92.00 | 90.86 |
| **MSA** | 85.90 | 99.00 | 91.99 |
| **TUN** | 84.99 | 92.00 | 88.36 |

Table 5.8: SVM performance per language

The macro-average F-score of the SVM classifier (in Table 5.8) is 89.02%. The classifier performs very well for BER like with Cavnar's classifier. It identifiers MOR and TUN better than ALG. Likewise, it recognizes KUI better than GUL. MSA is also well distinguished from other varieties with a high recall of 99.00% and F-score of 91.99%.

Table 5.9 shows the confusion matrix of the SVM using the same settings.

The SVM classifier often confuses between Maghrebi dialects (ALG, MOR and TUN) and between GUL and KUI. The explanations are the same as mentioned before. Still MSA is confused with all other languages mainly because of the false friends, i.e., SVM seems to overgenerate MSA and undergenerate the other variants.

| | | **Misclassified languages** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Correct languages** | | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
| | **ALG** | 331 | 0 | 2 | 1 | 1 | 1 | 6 | 21 | 37 |
| | **BER** | 0 | 398 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| | **EGY** | 8 | 0 | 324 | 14 | 6 | 17 | 21 | 3 | 7 |
| | **GUL** | 1 | 0 | 6 | 346 | 22 | 15 | 6 | 3 | 1 |
| | **KUI** | 1 | 0 | 0 | 23 | 371 | 3 | 1 | 0 | 1 |
| | **LEV** | 9 | 0 | 6 | 33 | 9 | 306 | 21 | 6 | 10 |
| | **MSA** | 1 | 0 | 0 | 0 | 1 | 1 | 396 | 0 | 1 |
| | **MOR** | 12 | 0 | 1 | 3 | 1 | 1 | 6 | 368 | 8 |
| | **TUN** | 15 | 0 | 3 | 0 | 3 | 0 | 2 | 9 | 368 |

Table 5.9: SVM confusion matrix

Finally, we will validate our model using the 10-fold cross-validation technique which consists in splitting the entire dataset into 10 equal folds. Each time, we preserve one fold for testing and train on the rest 9 folds. This will give us an idea on how the model is dataset independent. We will use the same settings above. The accuracy values are close to each other for all cross-validation folds, and close to the overall accuracy which is 89.11%. This means that the model is not an overfit, it is actually doing something.

As a summary for this subsection, SVM classifier using combination of character-based 5-6-grams outperforms all the other classifiers and settings and performs reasonably well in distinguishing dialectal Arabic. Filtering unimportant words and stop words along with using TF-IDF scores better. The hardest part is to properly distinguish between very similar dialects like Maghrebi dialects.

### 5.2.2 Word-based *n*-gram

We will redo the same experiments as with character-based *n*-gram, but now we will use word-based *n*-gram (entire words) as features. Table 5.10 shows the performance of the previously selected classifiers using word-based unigram (1-gram) with text maximum length of 11 words (the shortest document average length in our dataset)[52]. Since we are dealing with words, it makes sense to use TF-IDF to weight the importance of each word and use dataset D because we want to deal with the words of the language and not false heuristics such as punctuation, emotions, vowels, etc.

In Table 5.10, the maximum text length (11 words) is shorter than the one used in Table 5.5 (140 characters). This makes it hard to compare the results, particularly for the SVM classifier. However, even with less text in Table 5.10, NB classifier scores better, 82.25% accuracy compared to 74.50% accuracy with 140 characters (Table 5.5). NB has slightly outperformed the SVM classifier. For effectiveness reason (fast and best performance), we will use the NB classifier in the rest of this part.

| Classifier | Accuracy (%) | Training time (sec) |
|---|---|---|
| **KN** | 69.55 | 0.542 |
| **DT** | 57.11 | 14.989 |
| **NB** | 82.25 | 0.569 |
| **LG** | 75.77 | 1.099 |
| **SVM** | 81.02 | 0.942 |

Table 5.10: Classifiers accuracies with word-based unigram

Next, we will experiment with different word-based *n*-grams and text lengths. The results are shown in Figure 5.3.

52 We choose to use text maximum length of 11 words just for illustration. The assumption is that if the classifiers perform well for short texts, then they should perform better for longer ones.

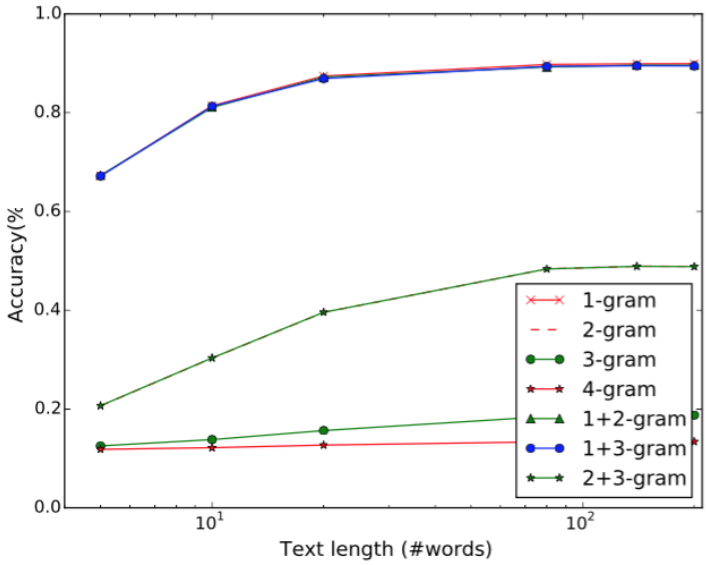



Figure 5.3: *n*-gram and text length effect on NB

Using word-based unigram, combining unigram with bigram and unigram with trigram perform almost the same, 82.25%, 82.33% and 82.16% accuracy respectively for maximum text length of 11 words. Other *n*-grams and combinations have not worked well because of the data sparsity; i.e. it is hard to find always the same word combination when dealing with different users and topics. Also, increasing the text length, more than 20 words does not improve the classification with all the *n*-grams, except for bigrams and the combination of bigrams and trigrams for which the accuracy has improved from 39.55% (20 words) to 48.36% (80 words). This is related to the fact that the dataset documents are short and it is less likely that some *n*-gram combinations occur frequently.

For detailed NB performance per languages, see Table 5.11 where we use word unigram with the same settings mentioned above.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **ALG** | 71.01 | 79.00 | 74.79 |
| **BER** | 98.72 | 96.75 | 97.73 |
| **EGY** | 79.856 | 79.25 | 79.55 |
| **GUL** | 73.74 | 78.25 | 75.79 |
| **KUI** | 81.46 | 86.75 | 84.02 |
| **LEV** | 79.16 | 79.75 | 79.45 |
| **MOR** | 84.07 | 85.75 | 84.90 |
| **MSA** | 96.75 | 74.50 | 84.18 |
| **TUN** | 81.27 | 80.25 | 80.75 |

Table 5.11: NB performance per language

The macro-average F-score of the NB classifier is 82.35%. NB can identify MOR and TUN with higher precision than ALG. Also, it detects MSA with a high precision, 96.75%, compared to the other varieties. Table 5.12 shows that it is still not easy to distinguish between Maghrebi dialects, KUI and GUL and between EGY, LEV and GUL. Also, BER is confused with ALG, MOR, LEV, GUL and EGY. For ALG and MOR, it is expected given the close contact between these languages.

| | | **Misclassified languages** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Correct languages** | | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
| | **ALG** | 316 | 0 | 8 | 7 | 6 | 3 | 3 | 30 | 27 |
| | **BER** | 8 | 387 | 1 | 1 | 0 | 2 | 0 | 1 | 0 |
| | **EGY** | 12 | 0 | 317 | 22 | 10 | 27 | 3 | 3 | 6 |
| | **GUL** | 5 | 0 | 16 | 313 | 35 | 24 | 0 | 3 | 4 |
| | **KUI** | 3 | 0 | 5 | 35 | 347 | 4 | 0 | 2 | 4 |
| | **LEV** | 11 | 0 | 19 | 26 | 8 | 319 | 2 | 8 | 7 |
| | **MSA** | 15 | 3 | 22 | 14 | 9 | 15 | 298 | 8 | 16 |
| | **MOR** | 32 | 1 | 2 | 2 | 3 | 5 | 2 | 343 | 10 |
| | **TUN** | 43 | 1 | 7 | 6 | 8 | 4 | 0 | 10 | 321 |

Table 5.12: NB confusion matrix

However for MSA, it is mainly caused by false friends, for instance the sentence: أين اروح أسا؟ [>yn ArwH >sA] which means in Berber [why he went/left today] but because of false friends with MSA, it may be read as [where I go sorrow]. It does not have any meaning but the classifier does not know it. MSA is also still confused with other Arabic varieties. For instance, in Egyptian, the sentence رحت بالعربية للجامعة [rHt bAlErbyp lljAmEp] means [I went to university by car] and [I went Arabic to university] in MSA. The word 'Arabic' is ambiguous between Egyptian Arabic (meaning car) and MSA (meaning Arabic language). It is hard to deal correctly with this kind of situation without introducing some word knowledge. This is a simple example why we need to correctly identity Arabic varieties and treat them as separate languages with their own resources.

We can conclude that the NB classifier using word-based unigrams is good at discriminating between Arabicized Berber and Arabic. Compared to the performance of the SVM in the experiment above (using character-based 5-6-grams), NB is less effective in distinguishing between MSA and dialectal Arabic. It even underperforms when it comes to discriminating between Arabic dialects. This can not be generalized because with NB we limit the maximum text length to 11 words and with SVM to 140 characters. If the word average length is 5 characters, then 11 words mean 55 characters which is a big difference. Still our main purpose is to experiment with different settings. We also assume that if NB performs as reported for very short texts (11 words or less), it should perform better for longer documents. It would be better to use the same text length for comparing the classifiers. However, it is hard to estimate how many words are included in 140 characters (each word has a different number of characters). We will do a full-length-document experiment at the end.

### 5.2.3 Dialectal vocabulary

The idea is that dialectal words[53] (unigrams) and some word combinations (bigrams and trigrams) are strong informative features. We will use the entries of the compiled dialectal lexicons as features. To do so, we have created a method which extracts the unique vocabulary of an input text and computes the overlap with each dialectal lexicons. We train NB and SVM classifiers with the dataset D using the dialectal vocabulary as features weighted using TF-IDF and text maximum length of 11 words. Overall, SVM classifier performs better than NB, a macro-average F-score of 70.12% and 55.67% respectively. For illustration, we will report in Table 5.13 the performance of the SVM per language.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **ALG** | 74.36 | 72.50 | 73.42 |
| **BER** | 96.95 | 95.25 | 96.09 |
| **EGY** | 61.02 | 38.75 | 47.40 |
| **GUL** | 57.03 | 54.75 | 55.87 |
| **KUI** | 62.33 | 58.75 | 60.49 |
| **LEV** | 66.05 | 62.25 | 64.09 |
| **MOR** | 82.41 | 82.00 | 82.21 |
| **MSA** | 59.44 | 84.25 | 69.70 |
| **TUN** | 76.52 | 88.00 | 81.86 |

Table 5.13: SVM performance using dialectal vocabulary

The classifier's macro-average F-score is 70.12%. Table 5.14 shows the confusion matrix of the SVM (for the same settings as in Table 5.13).

| | | Misclassified languages | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
| **Correct languages** | **ALG** | 290 | 0 | 4 | 7 | 12 | 11 | 21 | 14 | 41 |
| | **BER** | 1 | 381 | 3 | 2 | 0 | 0 | 12 | 0 | 1 |
| | **EGY** | 12 | 2 | 155 | 36 | 39 | 55 | 62 | 19 | 20 |
| | **GUL** | 14 | 1 | 34 | 220 | 55 | 27 | 30 | 10 | 9 |
| | **KUI** | 10 | 0 | 22 | 68 | 232 | 22 | 27 | 5 | 14 |
| | **LEV** | 8 | 0 | 23 | 25 | 17 | 250 | 60 | 6 | 11 |
| | **MSA** | 5 | 7 | 9 | 13 | 12 | 4 | 337 | 8 | 5 |

53 The entries of the compiled lexicons

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **MOR** | 31 | 2 | 3 | 10 | 4 | 6 | 8 | 328 | 8 |
| | **TUN** | 19 | 0 | 1 | 5 | 2 | 3 | 10 | 8 | 352 |

Table 5.14: SVM confusion matrix using the dialectal vocabulary

The classifier finds it hard to distinguish properly between close dialects like GUL and KUI, ALG and TUN or MOR and ALG. This is expected, as explained before, particularly for short texts. However, confusion between MSA and EGY and between BER and MSA is mainly caused by false friends. The poor performance of the classifier is related also to the fact that the compiled dialectal lexicons are not of a wide coverage, i.e. there are still many dialectal words which are not covered, given that we have not used neither the training nor the testing datasets in building the dialectal lexicons.

### 5.2.4 Feature combination

Now we will combine some features, namely word-based unigram combined with dialectal vocabulary and the character-based 5-6-grams combined with dialectal vocabulary. We will use document maximum length of 140 characters (for character-based *n*-gram) and maximum length of 11 words (for word-based *n*-gram).

#### 5.2.4.1 Combining word-based unigram with dialectal vocabulary

We will experiment with both NB and SVM classifiers. Figure 5.4 shows the performance of both.

It is clear that SVM performs better than NB. Comparing to the results of Table 5.11, we see that combining word-based unigram with dialectal vocabulary has improved the performance of SVM, accuracy of 86.36% compared to 81.02% when using only word unigram for maximum text length of 11 words. The performance of NB has however slightly decreased, accuracy of 82.02% compared to

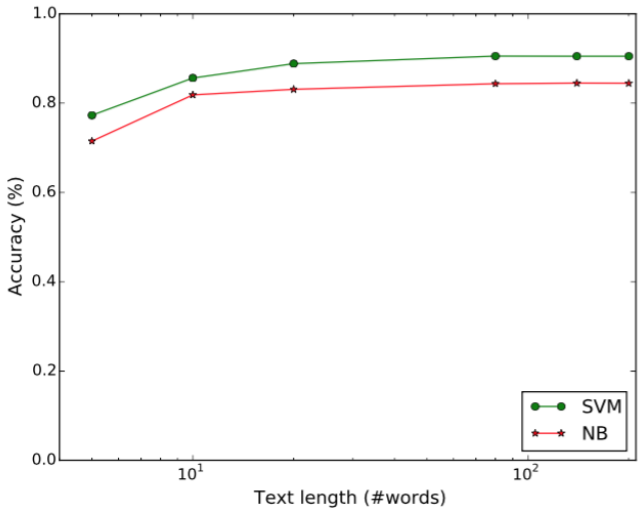



Figure 5.4: NB and SVM with feature combination

82.25% with only word unigram using the same experimental setup. To have a close look at the effect of combining features, we will show in Table 5.15 the performance of SVM per language using dataset D and text maximum length of 11 words.

The macro-average F-score of the SVM (using the mentioned settings) is 86.31%. Comparing the results in Table 5.15 and Table 5.11 (the macro-average F-score is 82.35%), it is evident that the feature combination has positive effect on the classification for all languages. Ideally, we would have compared the SVM performance using word unigram. But we think the comparison is still valid because the scores of the SVM using word unigram as features are less than the results shown in Table 5.11 using NB.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **ALG** | 81.22 | 86.50 | 83.78 |
| **BER** | 98.51 | 99.00 | 98.75 |
| **EGY** | 86.08 | 75.75 | 80.59 |
| **GUL** | 79.95 | 80.75 | 80.35 |
| **KUI** | 83.00 | 84.25 | 83.62 |
| **LEV** | 85.52 | 79.75 | 82.54 |
| **MOR** | 89.93 | 91.50 | 90.71 |
| **MSA** | 84.80 | 89.25 | 86.97 |
| **TUN** | 88.51 | 90.50 | 89.49 |

Table 5.15: SVM performance using word-based unigram and dialectal vocabulary

As shown in Table 5.16, the classifier still has the same error types: confusion between Maghrebi dialects, GUL/KUI and confusion with MSA and BER because of false friends. MSA is confused with all languages.

| | | **Misclassified languages** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Correct languages** | | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
| | **ALG** | 346 | 0 | 4 | 0 | 3 | 2 | 8 | 18 | 19 |
| | **BER** | 3 | 396 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | **EGY** | 12 | 2 | 303 | 23 | 12 | 20 | 14 | 6 | 8 |
| | **GUL** | 5 | 0 | 12 | 323 | 40 | 13 | 5 | 1 | 1 |
| | **KUI** | 5 | 0 | 9 | 33 | 337 | 10 | 2 | 3 | 1 |
| | **LEV** | 5 | 0 | 13 | 18 | 5 | 319 | 30 | 5 | 5 |
| | **MSA** | 8 | 4 | 7 | 3 | 6 | 7 | 357 | 2 | 6 |
| | **MOR** | 21 | 0 | 1 | 2 | 1 | 0 | 2 | 366 | 7 |
| | **TUN** | 21 | 0 | 3 | 2 | 2 | 2 | 2 | 6 | 362 |

Table 5.16: SVM confusion matrix using word-based unigram and dialectal vocabulary

To end up this subsection, we validate the classifier using the same mentioned settings (as in Table 5.15). For all cross-validation folds, the accuracies of the SVM classifier are close to each other. The macro-average accuracy is 89.54%. This means that the classifier has learned general patterns.

#### 5.2.4.2 Combining character-based 5-6-grams with dialectal vocabulary

We now use the combination of character-based 5-6-grams with the dialectal vocabulary. We use both SVM and NB classifiers. For a document maximum length of 140 characters, the SVM classifier outperforms the NB classifier, a macro-average F-score of 92.94% and 86.26% respectively. The performance of the SVM classifier per language is shown in Table 5.17. As a reminder, the reported results are for a maximum text length of 140 characters.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **ALG** | 91.79 | 92.25 | 92.02 |
| **BER** | 100 | 100 | 100 |
| **EGY** | 95.63 | 82.00 | 88.29 |
| **GUL** | 86.92 | 89.75 | 88.31 |
| **KUI** | 91.20 | 93.25 | 92.21 |
| **LEV** | 91.71 | 88.50 | 90.08 |
| **MOR** | 93.84 | 95.25 | 94.54 |
| **MSA** | 93.46 | 100 | 96.62 |
| **TUN** | 92.98 | 96.00 | 94.46 |

Table 5.17: SVM performance using character-based 5-6-grams and dialectal vocabulary

In Table 5.18, we show the confusion matrix of the SVM classifier using the same settings as in Table 5.15.

| | | **Misclassified languages** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Correct languages** | | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
| | **ALG** | 369 | 0 | 1 | 0 | 0 | 1 | 2 | 12 | 15 |
| | **BER** | 0 | 400 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **EGY** | 6 | 0 | 328 | 15 | 6 | 19 | 10 | 8 | 8 |
| | **GUL** | 1 | 0 | 5 | 359 | 24 | 7 | 3 | 0 | 1 |
| | **KUI** | 1 | 0 | 1 | 21 | 373 | 4 | 0 | 0 | 0 |
| | **LEV** | 7 | 0 | 7 | 16 | 3 | 354 | 10 | 1 | 2 |
| | **MSA** | 0 | 0 | 0 | 0 | 0 | 0 | 400 | 0 | 0 |
| | **MOR** | 9 | 0 | 0 | 2 | 1 | 1 | 3 | 381 | 3 |
| | **TUN** | 9 | 0 | 1 | 0 | 2 | 0 | 0 | 4 | 384 |

Table 5.18: SVM confusion matrix using character-based 5-6-grams and dialectal vocabulary

For all cross-validation folds, the accuracies of the SVM classifier (using the same settings as in Table 5.15) are close to each other. The macro-average accuracy is 92.18%. This indicates that the classifier has learned general patterns.

We can say that using only the dialectal words covered by the compiled dialectal lexicons as discriminants is not enough to distinguish Arabic variants from each other. It is not even efficient in detecting MSA from dialectal Arabic. This is because the lexicons are limited in terms of coverage. However, combining the word-based unigrams with the dialectal vocabulary has improved the performance of the SVM classifier for all languages (a macro-average F-score of 86.31% compared to 82.35% respectively). Moreover, combining dialectal vocabulary with the character-based 5-6-grams has increased the performance of the SVM, a macro-average F-score of 92.94% compared to 89.02% using only the character-based 5-6-grams (see Table 5.8). The results are all for short texts, maximum length of 140 characters or 11 words.

### 5.2.5 Regional dialect grouping

The classifiers we have tried so far, in different experimental setups, confuse mainly between Maghrebi and between GUL/KUI dialects. We will group these dialects in higher level categories (regional dialect) and see its effect. We will use dataset D and remove some documents to have a balanced test dataset and we keep the same training dataset. We keep only 400 documents grouping ALG, MOR and TUN (133, 133 and 134 documents respectively) labelled as 'MGR'. The same for GUL and KUI for which we keep only 400 documents (200 for each) labeled as 'KGI'. Using the same settings as in Table 5.15 (word unigram/dialectal vocabulary combination and maximum text length of 11 words), the SVM and NB macro-average F-scores are 85.38% and 56.16% respectively.

We cross validate the SVM classifier grouping the regional dialects. The macro-average accuracy is 92.25% which is close to the accuracy of each fold. Overall, grouping close dialects in regional category has a positive effect on the SVM performance for BER, EGY, LEV and MSA for which the precision has improved even though the recall has dropped for some, see Table 5.19. This is maybe because lots of hard cases were merged. For instance ALG, MOR and TUN are now grouped in the Maghrebi category, so the problem of classifying a given document as ALG, MOR or TUN is not more posed.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **BER** | 99.24 | 98.50 | 98.887 |
| **EGY** | 91.86 | 67.75 | 77.99 |
| **KGI** | 78.19 | 95.00 | 85.78 |
| **LEV** | 92.91 | 73.50 | 80.88 |
| **MGR** | 73.64 | 75.50 | 83.20 |
| **MSA** | 88.10 | 83.25 | 85.60 |

Table 5.19: SVM performance with dialect grouping

The SVM confusion matrix, Table 5.20, shows that the classifier still confuses between MSA and other languages mainly because of false friends. MGR, KGI, LEV and EGY are still difficult to distinguish from each other because they share some frequent properties like the use of the same negation particles along with some false friends.

| | | Misclassified languages | | | | | |
|---|---|---|---|---|---|---|---|
| | | **BER** | **EGY** | **KGI** | **LEV** | **MGR** | **MSA** |
| **Correct languages** | **BER** | 394 | 0 | 0 | 0 | 5 | 1 |
| | **EGY** | 1 | 271 | 49 | 18 | 47 | 14 |
| | **KGI** | 0 | 5 | 380 | 3 | 11 | 1 |
| | **LEV** | 0 | 12 | 33 | 302 | 25 | 28 |
| | **MGR** | 0 | 2 | 5 | 1 | 391 | 1 |
| | **MSA** | 2 | 5 | 19 | 2 | 39 | 333 |

Table 5.20: SVM confusion matrix with dialect grouping

### 5.2.6 Learning curves

To investigate the impact of the training dataset size on the classification, we use the SVM classifier trained on the dataset D combining the word-based unigram with dialectal vocabulary as features. Figure 5.5 shows the learning curve per language for text maximum length of 11 words. We use the same test dataset and change the size of the training dataset.

For all languages, increasing the size of the training dataset improves the classifier performance because new features are considered and learned from the new train dataset. Consequently, adding more training samples is beneficial for getting better classification performance.

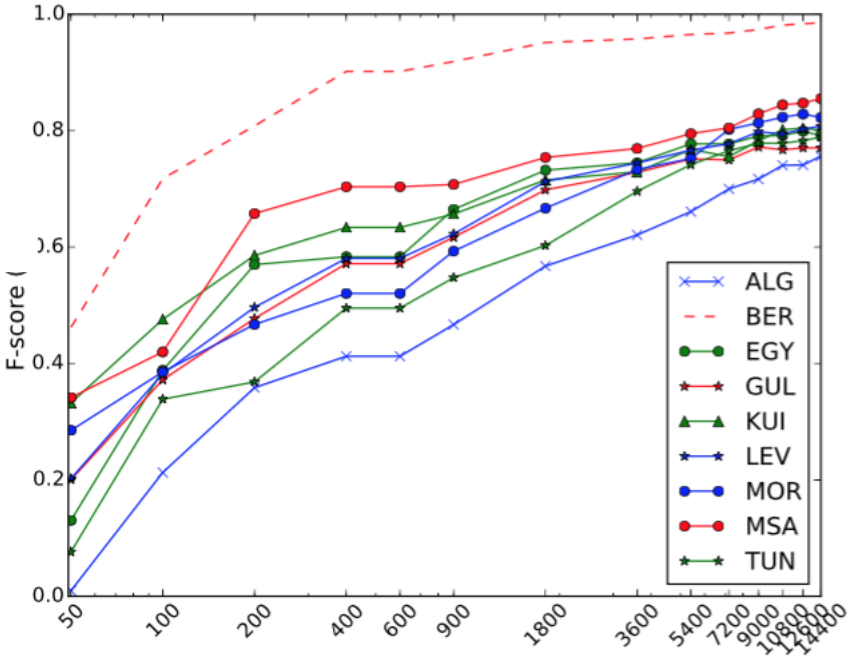



Figure 5.5: Learning curve for each language using SVM

### 5.2.7 Introducing the 'Unknown' category

Currently, we assume that all input texts will be written in Arabic script and belong to one of the Arabic varieties we are dealing with. This is far from being the case because we do not even cover all the existing Arabic varieties. So what if the text is not written in Arabic script? To deal with such case, we will add the 'UKN' language category, discussed in Chapter 4. One can argue that it is enough to set a threshold and consider all scores below it to be unknown language. But we find it hard to set a threshold which will be dataset independent. We think the easiest way is to introduce the 'UKN' category as done in (Řehůřek & Kolkus, 2009). We train the SVM classifier with the dataset D (plus 'UKN' documents having the same pre-processing) and combining the word-based unigram with dialectal vocabulary as features. Table 5.21 shows the classifier's performance per language for text maximum length of 11 words.

The classifier's macro-average F-score is 87.11%. The classifier distinguishes the 'UKN' category fairly well from the rest. This is expected because there are many languages using different scripts which are easy to distinguish from the Arabic script. However, still there are some confusions with the languages using the Arabic script like Pashto, Persian and Urdu. Our purpose is to return 'UKN' category instead of something else.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **ALG** | 80.51 | 86.75 | 83.51 |
| **BER** | 98.51 | 99.00 | 98.75 |
| **EGY** | 87.76 | 71.25 | 79.28 |
| **GUL** | 80.10 | 80.50 | 80.30 |
| **KUI** | 83.62 | 84.25 | 83.94 |
| **LEV** | 85.48 | 84.25 | 83.94 |
| **MOR** | 90.57 | 91.25 | 90.91 |
| **MSA** | 84.06 | 87.00 | 85.50 |
| **TUN** | 88.51 | 90.50 | 89.49 |
| **UKN** | 93.11 | 98.00 | 95.49 |

Table 5.21: SVM performance with 'UKN' category

### 5.2.8 Using the full-length-document

Until now, all the reported results are for short texts (maximum length of 140 characters or 11 words). Table 5.22 shows the results for the SVM classifier using the entire data text length both using the word unigram/dialectal vocabulary combination and the character-based 5-6-grams/dialectal vocabulary combinations. The classifier achieves a macro-average F-score of 93.40% using the character-based 5-6-grams/dialectal vocabulary combination and 90.48% using the word unigram/dialectal vocabulary combination both wighted with TF-IDF. Further, the NB classifier's best score is 88.54% and 84.78% macro-average F-score using the character-based 5-6-grams/dialectal vocabulary and the word unigram/dialectal vocabulary combination respectively. The macro-average F-score of the SVM classifier using only the character-based 5-6-grams is 89.65% and 87.78% using the word-based unigram only. Cavnar's classifier macro-average F-score is 81.57% using character 3-grams. Regardless of the text length and the used classifier, the classification errors are all of the same type; confusion between Maghrebi dialects, GUL and KUI, LEV and Maghrebi, LEV and EGY. The confusion with MSA and BER is mainly caused by false friends.

| | Word unigram/dialectal vocabulary | | | Character 5-6-grams/dialectal vocabulary | | |
|---|---|---|---|---|---|---|
| **Language** | **Precision (%)** | **Recall (%)** | **F-score (%)** | **Precision (%)** | **Recall (%)** | **F-score (%)** |
| **ALG** | 85.30 | 88.50 | 86.87 | 91.85 | 93.00 | 92.42 |
| **BER** | 99.26 | 100 | 99.63 | 99.50 | 100 | 99.75 |
| **EGY** | 93.48 | 82.50 | 87.65 | 97.12 | 84.25 | 90.23 |
| **GUL** | 85.36 | 86.00 | 85.68 | 88.53 | 88.75 | 88.64 |
| **KUI** | 88.41 | 91.50 | 89.93 | 90.07 | 95.25 | 92.59 |
| **LEV** | 90.22 | 83.00 | 86.46 | 92.65 | 88.25 | 90.40 |
| **MOR** | 92.61 | 94.00 | 93.30 | 93.66 | 96.00 | 94.81 |
| **MSA** | 89.12 | 96.25 | 92.55 | 92.99 | 99.50 | 96.14 |
| **TUN** | 91.63 | 93.00 | 92.31 | 95.29 | 96.00 | 95.64 |

Table 5.22: SVM performance with full-length-document

## 5.3 Prediction by Partial Matching method

We implement the PPMC5 method, as described in Chapter 4, using the same dataset D with the same split (train on 80% and test on 20%). Table 5.23 shows the confusion table of the PPMC5 method. Here, we use the full-length-document.

| | | **Misclassified languages** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Correct languages** | | **ALG** | **BER** | **EGY** | **GUL** | **KUI** | **LEV** | **MSA** | **MOR** | **TUN** |
| | **ALG** | 334 | 0 | 0 | 5 | 4 | 1 | 7 | 13 | 36 |
| | **BER** | 0 | 400 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | **EGY** | 8 | 0 | 323 | 26 | 8 | 14 | 11 | 4 | 6 |
| | **GUL** | 1 | 0 | 11 | 316 | 41 | 19 | 5 | 2 | 5 |
| | **KUI** | 1 | 0 | 2 | 18 | 377 | 1 | 1 | 0 | 0 |
| | **LEV** | 8 | 1 | 16 | 48 | 20 | 284 | 15 | 4 | 4 |
| | **MSA** | 1 | 0 | 1 | 3 | 2 | 0 | 391 | 0 | 2 |
| | **MOR** | 27 | 0 | 1 | 1 | 1 | 0 | 3 | 359 | 8 |
| | **TUN** | 21 | 0 | 0 | 0 | 3 | 0 | 6 | 6 | 364 |

Table 5.23: Confusion table of the PPMC5 classifier

The method reaches a macro-average Precision of 87.66%, macro-average Recall of 87.44%, a macro-average F-score of 87.55% and a micro-average F-score of 87.44%. In this part, we report the PPMC5

method performance using the macro-average measures (average of each measure) because we are interested in its over all performance across the test dataset. More or less, the PPMC5 classifier finds it hard to accurately distinguish between Maghrebi dialects, GUL and KUI, EGY and GUL and between LEV and GUL. It identifies Arabicized Berber accurately. It is also good at detecting MSA from dialectal Arabic.

## 5.4 Dictionary-based method

We use the entries of the dialectal lexicons as discriminants for each Arabic variety. We divide them into two categories: strong and weak features. Strong features consist of a short list of words occurring only in one variety. Some words are shown in Table 4.1. Weak dialectal features include words occurring in three varieties at most, for instance the words 'علاش' [ElA$] which means [why] and 'بزاف' [bzAf] which means [lots of/many] in Maghrebi Arabic. As mentioned before, there is no a reasonable way to set the minimum threshold for each document to belong to a certain dialect. Therefore, we use a simple statistical approach which gives more importance (weight) for strong dialectal features and treats all the weak dialectal features the same and attributes the same weight for all of them. If a text contains a strong feature, then it is classified in the corresponding dialect. Otherwise, it is classified in the dialect which has more weak feature overlap. In case two or more dialects have the same overlap, then the text belongs to any one of them if none of them is MSA. If one of them is MSA, then the priority is given to the dialect. In case there is no overlap at all, the text is not Arabic and 'UKN' is returned because if it is Arabic there should be at least one overlap with MSA. In this approach we keep all MSA words including prepositions, conjunctions and coordination, etc. We think that they are good discriminants between MSA and dialectal Arabic. We use the entire fully pre-processed dataset, 18 000 (2 000 documents for each language). We report in Table 5.24 the Precision, Recall and F-score for each language.

| Language | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|
| **ALG** | 86.58 | 67.15 | 75.63 |
| **BER** | 87.31 | 99.45 | 92.98 |
| **EGY** | 80.05 | 80.05 | 80.05 |
| **GUL** | 73.30 | 69.35 | 71.27 |
| **KUI** | 76.70 | 88.40 | 82.13 |
| **LEV** | 81.51 | 83.55 | 82.51 |
| **MOR** | 79.72 | 91.05 | 85.00 |
| **MSA** | 90.84 | 90.35 | 90.60 |
| **TUN** | 79.95 | 77.20 | 78.55 |

Table 5.24: Performance of the dictionary-based classifier

The macro-average F-score of the dictionary-based method is 82.08%. This method performs reasonably well in distinguishing between MSA and dialectal Arabic as well as distinguishing Arabicized Berber (BER). However, the method, as implemented, is still not effective in discriminating between Arabic dialects. Analyzing some classification errors indicates that the majority of the misclassifications are mainly caused by two problems. First, the shortage in the coverage of the dialectal lexicons where there are many new, unseen, dialectal words in the documents which are not yet covered in the lexicons. This is confirmed by the fact that the 'UKN 'category is returned 32 times, i.e., no vocabulary match. The second problem is the fact that we attribute the same weight, no priority, for all weak dialectal words then return the dialect with the maximum overlap.

This is not the best method to classify Arabic dialect using vocabulary, especially that the dialectal lexicons are not all of the same size (see Table 4.6). But given the fact that still there is no sound theoretical basis for any particular method, the suggested statistical method, as simple as it is, is reasonable. An error analysis shows that documents which contain strong dialectal words are rarely misidentified. Also, filtering all the clearly MSA words might be problematic because of the considerable amount of false friends between MSA and dialectal Arabic. We have not included the data of the 'UKN' category in this experiment simply because we do not have lexicons for the included languages, particularly Pashto, Persian and Urdu. These languages have many false friends with Arabic, so using only Arabic lexicons is biased. Another thing is that this method works better with fairly long documents. This is expected because then there are more dialectal words potentially covered by the lexicons. Since there is no a way to decide whether a word belongs to one of the dialects, some extralinguistic information are needed.

## 5.5 Summary of the results

To sum up this Chapter, we will report the main results for each method. In all cases, we will use the fully pre-processed dataset and all features are weighted using TF-IDF. Also, as a reminder, we use the binary classification setting. Table 5.25 summarizes the main results which are a simplified version (report only the macro-average F-score) to be able to compare between methods, i.e. each method performs differently for each language.

| Method | Features | Maximum Text Length | Macro-average F-score (%) |
|---|---|---|---|
| **Cavnar** | Character 3-grams | 140 characters | 52.41 |
| | Character 3-grams | Full length | 81.57 |
| **SVM** | Character 5-6-grams | 140 characters | 89.02 |
| | Character 5-6-grams | Full length | 89.65 |
| | Word unigram | 11 words | 80.96 |
| | Word unigram | Full length | 87.78 |
| | Dialectal vocabulary | 11 words | 70.12 |
| | Dialectal vocabulary | Full length | 64.81 |
| | Character 5-6-grams + Dialectal vocabulary | 140 characters | 92.94 |
| | Character 5-6-grams + Dialectal vocabulary | Full length | 93.40 |
| | Word unigram + Dialectal vocabulary | 11 words | 86.31 |
| | Word unigram + Dialectal vocabulary | Full length | 90.48 |
| **PPM** | No features | Full length | 87.55 |
| **Dictionary** | No features | Full length | 82.08 |

Table 5.25: Summary of the main results

In all cases, increasing the text length improves the performance of the classifier, except for SVM using only dialectal vocabulary as features. This is related to the limited coverage of the compiled lexicons, i.e. increasing the text length increases the chance of unseen words (not covered by the lexicons). Combining the character-based 5-6-grams with the dialectal vocabulary outperforms all the other methods. Cavnar's method performs poorly for short texts (maximum of 140 characters)

# 6 Conclusions

## 6.1 General findings

In this study, we deal with the task of discriminating between Arabic varieties which are very close languages. We consider eight (8) high level varieties (Algerian (ALG), Egyptian (EGY), Gulf (GUL), Levantine (LEV), Mesopotamian (KUI), Moroccan (MOR), Tunisian (TUN) dialects plus Modern Standard Arabic (MSA)) which are the most popular Arabic variants. The task is challenging at many levels. First, except MSA, Arabic dialects are under-resourced and undocumented languages (spoken languages). Second, dialectal Arabic is mostly used in social media and mobile phone messages. This makes the task harder since this genre allows only short texts. Moreover, colloquial Arabic is usually used in a multilingual societies with an extensive use of 'mixed languages' and non standardized orthography.

To start, with the help of native speakers of each dialect, we collected a data from various social media platforms (micro-blogs, forums, blogs and online newspapers) including a wide variety of topics (cartoons, cooking, health/body care, movies, music, politics and social issues). We added Arabicized Berber which is an under-resourced-language coexisting with Arabic in North Africa. We removed 18 000 (2 000 document for each language) to be used in different experiments and extracted unique vocabulary from the rest to compile dialectal lexicons. We also used some extra-sources (forums content) to collect the dialectal words. We used the Cohen's kappa coefficient (κ) to measure the data annotation quality. Overall, the inter-annotator agreement is satisfactory even though the use of the κ in such nontrivial linguistic task is questionable.

The task is seen as a categorization problem for short texts written in very similar languages. Our purpose is to apply the automatic language identification standard methods to Arabic varieties. We have used machine learning (Cavnar's text classification method, some classifiers from Scikit-learn and the Prediction by Partial Matching method) and dictionary-based methods. We have used different experimental settings to investigate the effect of some parameters, particularly the data pre-processing, the weighting of feature importance. We have used both character and word-based language models as features. To select the most informative features, we have experimented with both simple frequency counting (TF) and TF-IDF schemes. In all cases, we find that using fully pre-processed data with unimportant words removal scores the best. Furthermore, TF-IDF weighting performs better than the simple TF. This makes perfect sense as there are many vocabulary overlap between Arabic varieties. This makes the prevalent dialectal words useless discriminants. Also, using the fully pre-processed data allows the classifiers to learn the actual linguistic features of each variant rather than some language/region specific heuristics like Named Entities.

We find also that Cavnar's method using small pieces of text (letters/characters) as features does not help in discriminating between Arabic varieties. The reason is that all the varieties use the same character set with almost the same distribution. The Cavnar's classifier scores the best, 52.41% macro-average F-score, using 3-grams with short texts (maximum length of 140 characters). Increasing the length of the *n*-gram does not improve the classifier's performance. However, increasing the length of the text has a good effect, 81.57% macro-average F-score using text full-length. For the SVM classifier, increasing the *n*-gram length improves the performance, 88.61% accuracy using 6-grams compared to 85.86% using 3-grams. Also, combining long character-based *n*-grams (5-6-grams) scores even better, 89.02% accuracy. This can be explained by the fact that character-based 5-6-grams are mainly words and these long matches do not occur frequently by chance. Another important finding is

that character-based language model is very effective in discriminating between Arabicized Berber and Arabic even for short texts of 140 characters maximum. The Cavnar's classifier got an F-score of 96.16% (Table 5.3) and the SVM got an F-score of 99.75% (Table 5.3). The same is noticed in discriminating between MSA and dialectal Arabic where the Cavnar's classifier performed with an F-score of 71.05% (Table 5.3) and the SVM classifier performed with an F-score of 91.99% (Table 5.8). Using the SVM classifier, we found that character-based *n*-gram language model is slightly better than the word-based *n*-grams, where we got a macro-average F-score of 89.65% and 87.78 % respectively using the full-length-document (Table 5.25). Word unigram/bigram combination outperforms the rest of the *n*-grams and the *n*-gram combinations. As mentioned before, the reason is the data sparsity.

Using only the dialectal words works better with SVM compared to NB, where we got a macro-average F-score of 70.12% and 55.67% respectively for short texts. Compared to the SVM performance using unigram/bigrams combination, it is evident that using only the dialectal vocabulary is not efficient in distinguishing between Arabic varieties. This might be related to the shortage in the coverage of the dialectal lexicons. It is not even able to properly detect MSA from dialectal Arabic (Table 5.13). Nevertheless, combining the word-based unigrams with the dialectal vocabulary and the combining the character-based 5-6-grams with the dialectal vocabulary have improved the performance of the SVM classifier with the same experimental setup. We also noticed that using the full text length (no maximum length limitation) has a positive effect on the classification. Error analysis shows that all the errors, whatever the method, are of the same type; confusion between very similar languages.

Likewise, the PPMC5 classifier is good at distinguishing Arabicized Berber from Arabic and MSA from dialectal Arabic. It performs almost like the SVM classifier using only word unigrams (Table 5.25). The same is applied to the dictionary-based method. All in all, the used methods are effective in detecting Arabicized Berber from Arabic and MSA from dialectal Arabic. However, all of them fail to accurately distinguish between Arabic varieties. The hard cases are distinguishing between very close dialects like the Maghrebi (Algerian, Moroccan and Tunisian) dialects, Gulf and Mesopotamian varieties, Levantine and Egyptian and Maghrebi variants. The absence of strong hallmarks besides the lexical similarities make the task hard even for native speakers. The main reason is that these dialects are actually a group of different dialects and coming from a given region does not necessary mean being familiar with all the spoken dialects in that region. Given the fact that it is hard to find suitable linguistic resources, we can conclude that machine learning models are well suited for the identification of Arabic varieties.

Grouping the very close dialects into one regional group has a positive effect on the classification as many hard cases are merged. However, still there are some prevalent features and vocabulary similarities between these regional groups (Maghrebi, Gulf/Mesopotamian, Levantine and Egyptian). This makes discriminating between dialectal Arabic a nontrivial linguistic task since it is hard to find a dialectal clear-cut borderlines. The good thing with this study is that the findings do not conflict with the findings of the modern standard Arabic dialectology, rather they confirm them. For instance, Palva (2006) states 'the differences between the Bedouin dialects in the whole Western dialect area are relatively slight'. This explains perfectly the confusion of the classifiers between the Maghrebi dialects. He also says 'the Mesopotamian gilit[54] dialects had the same changes as the Gulf Arabic while other dialects preserve the Bedouin dialects. Moreover, the Muslim dialect in Baghdad developed due to its contact with the qeltu[55] dialects from which many changes were adopted'.

As a side task, we have introduced the 'Unknown' category which is assumed to take care of the cases where the input text is not written in any of the Arabic varieties. For the rest of the Arabic dialects which we have not dealt with, there is a big chance that they will be classified as one of the

54 It is an Arabic dialect spoken in Iraq. The name refers to the way the phrase 'I said' is pronounced.

55 It is another Arabic dialect spoken in Iraq where people pronounce the phrase 'I said' as 'qeltu'.

eight (8) dialects due to the big similarities between them. Likewise, we do not assume that we cover all the existing natural languages. Of course, this is not the best solution. For instance, a better solution is to set a minimum threshold, for each language, that each text has to score to belong to that language. Yet, how to set a minimum threshold for each language which is not data biased? This is applicable to all languages and not just Arabic. We would like to investigate the performance of each method based on the data source. However, the time limitation did not allow us to do. Still, we believe that the applied methods should perform the same because the used pre-processing filters all the platform special markers and region specific words.

## 6.1 Future directions

We have applied some of the automatic language identification standard methods to discriminate between Arabic varieties which are under-resourced dialects (languages). We found that machine learning models are well suited for our task to a large extent. This should be a good start to automatically process dialectal Arabic. Still, there are some points we want to explore further like applying the two step classification process which consists in first identifying the regional dialectal group, for instance Maghrebi, then apply some different feature weighting to identify the dialect itself. It would be also possible to analyze the misspellings which seem to be consistent within the same variant because the orthography is based on the pronunciation. This could help improving the dialectal Arabic identification. Another worth exploring way is to include some user metadata (extralinguistic information) like the location.

Most of the current NLP applications which support Arabic are rule-based systems. They are based mainly on the morpho-syntactic analysis of parsers designed for Modern Standard Arabic (MSA). However, using these tools, as they are, to automatically process dialectal Arabic content does not output any accurate analysis. This is because of the considerable differences and false friends between MSA and colloquial Arabic. At the same time, building new rule-based NLP tools for each Arabic variety is extremely expensive because of the complexity of the Arabic morphology. Moreover, applying statistical methods is impossible for now due to the lack of dialectal data and even MSA-dialectal Arabic parallel corpora. We believe that the best way is to adapt the already existing MSA tools to take into account the properties of each Arabic variant. Analyzing the compiled dataset would be beneficial to extract morphosyntactic information of the Arabic variants. It is also necessary to consider each variant as a stand-alone language because all the variants are different and they have a considerable amount of false friends. With some human effort, it could be also used as start to build parallel corpora for Arabic variant by translating all the content to MSA and to other dialects.

It is also worth investigating ways to support under-resourced languages. The data deficiency makes it hard to apply the standard methods used for general purpose languages. Therefore, finding new ways to build linguistic resources for these languages is itself an interesting research topic. In the context of today's globalization, it is important to build multilingual and cross-lingual NLP applications which are able to process the real data written in 'non standardized languages'. From the business perspective, in order to have a communicative effect for any NLP application, it should take into account the targeted user needs. We have just mentioned some interesting areas as a motivation to process low-density languages and take the Arabic varieties as an example.

Last but not least, one of the NLP challenges is to properly deal with mixed language inputs (text or speech). As mentioned before, Arabic in general is a mixture of various unrelated languages (for historical reasons) written either in Arabic script or others. Further, most Arabic-speaking societies are multilingual. For instance, it is hardly possible to find any NLP tool which is able to correctly analyzed the sentence: bf زعما تاوعو ليزامي مايجوتيكش علاش حابس ياخي [yAxy Habs ElA$ mAyjwtyk$

lyzAmy tAwEw zEmA bf] which means [How stupid is he, why does not he add you to his friends, supposedly he is a best friend] taken from North Africa let alone correctly translating it into any other language. It contains words in Maghrebi Arabic and French words (ajouter [to add] and les amis [friends]) adapted to Maghrebi morphology and syntax and an English common social media abbreviation (bf). The complexity and the linguistic richness of Arabic makes it a good study case of many NLP challenges.

# References


Abd-El-Jawad, Hassan R.S. (1992). Is Arabic a pluricentric language?. I*n Clyne, Michael G. Pluricentric Languages: Differing Norms in Different Nations. Contributions to the sociology of language 62*. Berlin & New York: Mouton de Gruyter. pp. 261–303

Akbacak, M., Franco, H., Frandsen, M., Hasan, S., Jameel, H., Kathol, A., Khadivi, S., Lei, X., Mandal, A., Mansour, S., Precoda, K., Richey, C., Vergyri, D., Wang, W., Yang, M. & Zheng, J. (2009). Recent advances in SRI's IraqCommtm Iraqi Arabic-English speech-to-speech translation system. *In Proceedings of IEEE ICASSP, (Taipei)*, pp. 4809 – 4813.

Akbacak, M., Vergyri, D., Stolcke, A., Scheffer, N. & Mandal, A. (2011). Effective Arabic dialect classification using diverse phonotactic models. *In INTERSPEECH'11*, Florence, Italy.

Baldwin, T. & Lui, M. (2010). Language Identification: The Long and the Short of the Matter. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pp. 229–237. Los Angeles, California.

Batchelder, E.O. (1992). A Learning Experience: Training an Artificial Neural Network to Discriminate Languages. *Technical report*, City University of New York.

Beesley, K. (1988). Language identifier: A computer program for automatic natural language identification on on-line text. *Languages at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association*, pp. 47-54.

Behnstdt, P. & Woidich, M. (2013). Dialectology. *In the Oxford Handbook of Arabic Linguistics, Dialectology*, pp. 300 – 323.

Bobicev, V. (2015). Discriminating between similar languages using ppm. *In Proceedings of the LT4VarDial Workshop*, Hissar, Bulgaria.

Bobicev, V. (2007). Comparison of Word-based and Letter-based Text Classification. *RANLP V*, Bulgaria, pp. 76–80.

Boril, H., Sangwan, A. & Hansen, J. H. L.(2012). Arabic dialect identification – Is the secret in the silence? and other observations. *In INTERSPEECH 2012*, Portland, Oregon.

Bratko, A., Cormack, G. V., Filipic, B., Lynam, T. R. & Zupan, B. (2006). Spam filtering using statistical data compression models, *Journal of Machine Learning Research 7*:2673–2698.

Buckwalter, Tim. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0. *Linguistic Data Consortium, University of Pennsylvania. LDC Catalog No.: LDC2002L49.*

Canasai Kruengkrai, Prapass Srichaivattana, Virach Sornlertlamvanich & Hitoshi Isahara. (2005). Language identification based on string kernels. I*n Proceedings of the 5th International Symposium on Communications and Information Technologies (ISCIT- 2005)*, pp. 896–899, Beijing, China.

Carletta, Jean. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2), pp. 249–254.

Catherine Taine-Cheikh. (2012). On the usefulness and limits of a geographic perspective in dialectology: Arabic and Berber examples. *STUF, Akademie Verlag, (Issue 1)*, pp. 26-46.

Christoph Tillmann, Yaser Al-Onaizan & Saab Mansour. (2014). Improved Sentence-Level Arabic Dialect Classification. *In Proceedings of the 1st Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects,* pp. 110 –119, Dublin, Ireland.

Cleary, J., Teahan, W. & Witten, I. (1995). Unbounded length contexts for PPM. *Data Compression Conference.*

Combrinck, H. & Botha, E. (1994). Automatic language identification: Performance vs. complexity. *In Proceedings of the 6th Annual South Africa Workshop on Pattern Recognition.*

Cyril Goutte, Serge Léger, Shervin Malmasi & Marcos Zampieri. (2016). Discriminating Similar Languages: Evaluations and Explorations. *In Proceedings of Language Resources and Evaluation (LREC)*. Portoroz, Slovenia.

Davies, E., Bentahila A. & Owens, J. (2013). Codeswitching and related issues involving Arabic. *In the Oxford Handbook of Arabic Linguistics, Sociolinguistics,* pp. 326-348.

de Jong & Rudolf, E. (2000). A grammar of the Bedouin dialects of the Northern Sinai littoral: Bridging the linguistic gap between the Eastern and Western Arab world. *Leiden: Brill.*

Dunning, T. (1994). Statistical Identification of Language. *Technical Report MCCS*, pp. 94-273, Computing Research Lab (CRL), New Mexico State University.

Enam El-Wer. (2013). Dialectology. *In the Oxford Handbook of Arabic Linguistics, Sociolinguistics*, pp. 249-250.

Fadi Biadsy, Julia Hirschberg & Nizar Habash. Spoken Arabic dialect identification using phonotactic modeling. (2009). *In Proceedings of the EACL Workshop on Computational Approaches to Semitic Languages*, pp. 53–61.

Gregory Grefenstette. (1995). Comparing two language identification schemes, JADT 1995, 3r*d International conference on Statistical Analysis of Textual Data*, Rome.

Habash, N. (2010). Introduction to Arabic Natural Language Processing. *Morgan & Claypool Publishers.*

Heba Elfardy & Mona Diab. (2013). Sentence Level Dialect Identification in Arabic. *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-13)*, Sofia, Bulgaria.

Hornik K., Mair P., Geiger W., Rauch J., Buchta C., Feinerer I. (2013).The textcat package for N-Gram based text categorization in R. *Journal of Statistical Software*, 52 (6), pp. 1-17.

Houda Saadane. (2015). *Le traitement automatique de l'arabe dialectalisé: aspects méthodologiques et algorithmiques*. PhD thesis, Université Grenoble Alpes.

Ljubešić, N., Kranjčić, D. (2014). Discriminating between VERY similar languages among Twitter users. 9th *Language Technologies Conference Information Society – IS.*

Ljubešić, N., Nives Mikelic & Damir Boras. (2007). Language identification: How to distinguish similar languages? *In Proceedings of the 29th International Conference on Information Technology Interfaces.*

M. Zampierri & B. G. Gebre. (2012). Automatic Identification of Language Varieties: The Case of Portuguese. *In Proceedings of KONVENS 2012 (Main track: poster presentations)*, Vienna.

Marcos Zampieri, Binyam Gebrekidan Gebre & Sascha Diwersy. (2013). N-Gram Language Models and POS Distribution for the Identification of Spanish Varieties. *In Proceedings of TALN*, Sables d'Olonne, France, pp. 580-587.

Moffat, A. (1990). Implementing the PPM data compression scheme. *IEEE Transactions on Communications*, 38(11), pp. 1917-1921.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy & Yassine Benajiba. (2010). COLABA: Arabic dialect annotation and processing. *In Proceedings of the LREC Workshop on Semitic Language Processing*, pp. 66 –74.

Muntsa Padró & Llus Padró. (2004). Comparing methods for language identification. *Procesamiento del Lenguaje Natural*, 33, pp. 155–162.

Mustonen, S. (1965). Multiple discriminant analysis in linguistic problems. *In Statistical Methods in Linguistics*. Skriptor Fack, Stockholm.

Newman, P. (1987). Foreign language identification – first step in the translation process. *In K. Kummer (editor), Proceedings of the 28th Annual Conference of the American Translators Association,* pp. 509-516.

Nizar Y. Habash, Owen C. Rambow, David Chiang, Mona Diab, Rebecca Hwa, Khalil Sima'an, Vincent Lacey, Roger Levy, Carol Nichols & Safiullah Shareef. (2006). Parsing Arabic Dialects, *Columbia University Academic Commons*, http://hdl.handle.net/10022/AC:P:20935.

Palva, H. (2006). *Encyclopedia of Arabic languages and linguistics, v.1, A-Ed. Leiden: Brill*, pp. 604 613.

Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander R., Habash, N., Pooleery, M., Rambow, O. & Roth, R.M. (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. *In Proceedings of the 9th LREC*, Iceland.

Pedregosa, F., Varoquaux, G., Gramfort, G., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, pp. 2825–2830.

Qafisheh, H. A. (1977). A short reference grammar of Gulf Arabic. Tucson: University of Arizona Press.

Rabin, Chaim. (1951). Ancient West-Arabian. *London: Taylor's Foreign Press.*

Radim Řehůřek & Milan Kolkus. (2009). Language Identification on the Web: Extending the Dictionary Method. *In Computational Linguistics and Intelligent Text Processing, 10th International Conference, CICLing Proceedings,* pp. 357-368. Mexico City, Mexico.

Sankoff, D. (1998). The production of code-mixed discourse. *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, New Brunswick, NJ: ACL Press, pp. 8-21.

Souag, Lameen. (2004). Writing Berber Languages: a quick summary. *L. Souag. Archived* from http://goo.gl/ooA4uZ, Retrieved on April 8th, 2016.

Teahan, William J. & Harper, David J. (2003). Using compression-based language models for text categorization. In Language Modeling and Information Retrieval, pp. 141-165.

Tiedemann, J. & Ljubešić, N. (2012). Efficient Discrimination Between Closely Related Languages. *In Proceedings of COLING*, pp. 2619-2634.

Uebersax, J.S. (1987). Diversity of decision-making models and the measurement of inter-rater agreement. *Psychological Bulletin*, vol: 101, No:1, pp. 140 –146.

William B. Cavnar and John M. Trenkle. 1994. N- gram-based text categorization. *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 161–175, Las Vegas, US.

Yan Deng. (2008). Automatic Language Identification using support vector machines and phonetic N-gram, *IEEE*.

Yassine Benajiba & Mona Diab. (2010). A web application for dialectal arabic text annotation. *In Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages*: Status, *Up- dates, and Prospects.*

Yun Lei & John H. L. Hansen. (2011). Dialect classification via text-independent training and testing for Arabic, Spanish, and Chinese. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1), pp. 85–96.

Zaidan Omar F. & Callison-Burch Chris. (2014). Arabic dialect identification. *Computational Linguistics*, 40(1), pp. 171-202.

Zaidan Omar F. (2012). *Crowdsourcing Annotation for Machine Learning in Natural Language Processing Tasks.* PhD thesis, Johns Hopkins University.

Zhang, Q., Boril, H. & Hansen, J. H. L. (2013). Supervector Pre-Processing for PRSVM-based Chinese and Arabic Dialect Identification. *IEEE ICASSP'13*, pp. 7363-7367, Vancouver, Canada.

Ziegler, DV. (1992). The automatic identification of languages using linguistic recognition signals. *State University of New York at Buffalo*, Buffalo, NY.

Zippo, AG. (2012). Text Classification with Compression Algorithms. http://arxiv.org/abs/1210.7657.

# 7 Appendix A: Buckwalter Arabic transliteration scheme

Buckwalter Arabic transliteration scheme is developed by Tim Buckwalter in 1990's. He used it in his Arabic Morphological Analyzer. Since then, it is commonly used by the research community. It is a simplified letter-to-letter mapping between Arabic and Latin Alphabet. The full mapping chart is shown in Figure A.1.

| | | |
|---|---|---|
| ء ' | ذ * | ل l |
| آ \| | ر r | م m |
| أ > | ز z | ن n |
| ؤ & | س s | ه h |
| إ < | ش $ | و w |
| ئ } | ص S | ى Y |
| ا A | ض D | ي y |
| ب b | ط T | ً F |
| ة p | ظ Z | ٌ N |
| ت t | ع E | ٍ K |
| ث v | غ g | َ a |
| ج j | ـ _ | ُ u |
| ح H | ف f | ِ i |
| خ x | ق q | ّ ~ |
| د d | ك k | ْ o |

Figure A.1: Buckwaalter Arabic transliteration scheme