Romanized Berber and Romanized Arabic Automatic Language Identification Using Machine Learning

Wafia Adouane[1], Nasredine Semmar[2], Richard Johansson[3]
Department of FLoV, University of Gothenburg, Sweden[1]
CEA Saclay – Nano-INNOV, Institut CARNOT CEA LIST, France[2]
Department of CSE, University of Gothenburg, Sweden[3]
wafia.gu@gmail.com, nasredine.semmar@cea.fr
richard.johansson@gu.se

Abstract

The identification of the language of text/speech input is the first step to be able to properly do any language-dependent natural language processing. The task is called Automatic Language Identification (ALI). Being a well-studied field since early 1960's, various methods have been applied to many standard languages. The ALI standard methods require datasets for training and use character/word-based n-gram models. However, social media and new technologies have contributed to the rise of informal and minority languages on the Web. The state-of-the-art automatic language identifiers fail to properly identify many of them. Romanized Arabic (RA) and Romanized Berber (RB) are cases of these informal languages which are under-resourced. The goal of this paper is twofold: detect RA and RB, at a document level, as separate languages and distinguish between them as they coexist in North Africa. We consider the task as a classification problem and use supervised machine learning to solve it. For both languages, character-based 5-grams combined with additional lexicons score the best, F-score of 99.75% and 97.77% for RB and RA respectively.

1 Introduction

Social media and new technology devices have facilitated the emergence of new languages on the Web which are mainly written forms of colloquial languages. Most of these languages are under-resourced and do not adhere to any standard grammar or orthography. Romanized Arabic (RA) or Arabic written in Latin script (called often Arabizi) is an informal language. However, Romanized Berber (RB) is one of the Berber or Tamazight[1] standard forms. Both RA and RB are under-resourced and unknown languages to the available language identification tools [2]. To be able to automatically process and analyze content in RA and RB, it is necessary to properly recognize the languages. Otherwise, there is a large risk of getting misleading information. Moreover, it is crucial to be able to distinguish between them. The reason is that RA and RB coexist in North Africa, which is a rich multilingual region, and they share a considerable amount of vocabulary due to the close contact between them. Undoubtedly, this type of tool will help to build NLP applications for both. There is some work done to automatically transliterate RA into Arabic script (Al-Badrashiny et al., 2014). However, this is very limited because RA perfectly adheres to the principle 'write as you speak', i.e. there is no standardized orthography. Furthermore, the Arabic Chat Alphabet (ACA), designed for Romanized Arabic used in social media, is just a suggested writing system and not necessarily a Natural Language Processing (NLP) tool for RA. To overcome the various challenges faced when dealing with RA automatic processing, namely the use of non-standardized orthography, spelling errors and the lack of linguistic resources, we believe that it is better to consider RA as a stand-alone language and try to find better ways to deal with it instead of using only transliteration. RB is already a stand-alone language. It is important to clarify that considering both

[1] An Afro-Asiatic language widely spoken in North Africa. It is a minority language compared to Arabic.

[2] Among the freely available language identification tools, we tried Google Language Identifier, Open Xerox language, langid.py (M. Lui and T. Baldwin, 2012) and Translated labs at http://labs.translated.net.

Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects,
pages 53–61, Osaka, Japan, December 12 2016.

RA and RB as a stand-alone languages does not suggest, at any point, that the use of the Latin alphabet is a sufficient criteria to define them as such. Our main motivation is to make their automatic processing easier.

We start the paper with a general overview about the work done for informal Arabic NLP in Section 2. We then give some brief information about RA and RB in Section 3. Next, in Section 4, we describe how we proceed to build the linguistic resources used to build our system. In Section 5, we explain the used methods and describe the experiments and discuss the results. We conclude by general findings and future directions.

2 Related Work

Arabic NLP is mainly Modern Standard Arabic (MSA) based. Recently, the automatic processing of informal Arabic or dialectal Arabic in general has attracted more attention from the research community and industry. However, the main issue is the absence of freely available linguistic resources[3] which allow for automatic processing. The deficiency of linguistic resources for dialectal Arabic written in Arabic script (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 applied only to the written DA because there are available resources for spoken DA or at least it is easy to create them, for instance by recording TV-shows. However, for dialectal Arabic written in Latin script or RA, the only available resources are a few datasets that have been individually built for specific projects.

In general, only some work has been done for dialectal Arabic written in Arabic script, among others, automatic identification of some Arabic dialects (Egyptian, Gulf and Levantine) using word and character n-gram[4] models (Cavnar and Trenkle, 1994) and automatic identification of Maghrebi Arabic (Algerian, Moroccan and Tunisian) using morpho-syntactic information (Saâdane, 2015), Arabic dialect identification using a parallel multidialectal corpus (Malmasi et al., 2015) and identification of the most popular Arabic dialects using various automatic language identification methods (Adouane, 2016). However, the work done so far for RA deals mainly with Egyptian Arabic. For instance, Eskander et al., (2014) presented a system for automatic processing of Arabic social media text written in Arabizi by detecting Arabic tokens, Egyptian words, and non-Arabic words, mainly English words. They used a supervised machine learning approach to detect the label of each input token (sounds, punctuation marks, names, foreign words or Arabic words) and transliterated it into Arabic script. Darwish (2014) also presented an Arabizi identification system using word and sequence-level features to identify Arabizi that is mixed with English and reported an identification accuracy of 98.5%. This does not generalize to other RA content since it did not consider a broader range of data, i.e. there are many other Arabic dialects which are considerably different from Egyptian Arabic, for instance Arabic dialects used in North Africa, Levant region, Gulf countries and Iraq. Moreover, the mixed language used with Romanized Arabic is not always English[5].

To our knowledge, there has not been much work done to process RA (NLP applications like language identification, sentiment analysis/opinion mining, machine translation, Part-of-Speech tagging, etc.) as a stand-alone language. Furthermore, none of the automatic language identification standard methods have been applied to a wide range of Arabic dialects written in Latin script. As mentioned, the main challenge is the absence of data. RB is also unknown to the current language identifiers. It is an under-resourced language and a minority language compared to Arabic. There has been some work done for Berber automatic language identification, for instance Chelali et al. (2015) created a Berber speaker identification system using some speech signal information as features. Also Halimouche et al. (2014) have used prosodic information to discriminate between affirmative and interrogative sentences in Berber. Both sets of work were done at the speaker level. There are also some other applications which assume

[3]For dialectal Arabic written in Arabic script, there are some collections by individuals but unfortunately not digitalized or do not respect corpus linguistics annotation conventions (Behnstdt and Woidich, 2013).

[4]A sequence of n characters from a given sequence of text where n is an integer.

[5]We collected a dataset written in Romanized Arabic (including various Arabic dialects) and found various mixed languages, namely Berber, French, German, Italian, Spanish, Swedish and English.

that the input is always in RB. Both RA and RB are unknown languages to the available automatic language identifiers. The main motivation of this paper is to create an automatic language identifier which is able to detect RA and RB and at the same time is able to distinguish between them.

3 Characteristics of RA and RB

By informal languages, we mean the kind of written or spoken languages that do not adhere strictly to some standard spelling and grammar. The informality can be manifested in the form of ungrammatical sentences, misspellings, newly created words and abbreviations or even using unusual scripts as in the case of RA which has existed since the 20th century in North Africa. During the French colonialism period, educated people mastered Latin alphabet which was also used, for pedagogical purpose, to transcribe Arabic texts based on some phonological criteria (Souag, 2004).

RA is mainly dialectal Arabic which uses non-standard spellings, no fixed grammar and regional vocabulary-sense usage, i.e. the meaning of words depends on the area it is spoken in. Moreover, the use of the Latin script has increased the number of possible spellings per word at both vowels and consonants levels. With consonants, the main issue is the non-existence of some Arabic sounds in the Latin alphabet. Consequently, people use different characters to express those sounds. Unfortunately, the spellings are inconsistent even inside a group of people of the same area. RB also uses different national standardized orthography where each country has created its own standard spelling which is not necessarily used by another.

There are many false friends between RA and RB. For instance, each word in the Romanized Berber sentence 'AHml sAqwl mA$y dwl kAn'[6] which means 'love is from heart and not just a word' has a false friend in MSA and all Arabic dialects, namely when written in Latin script. In MSA, the sentence means literally 'I carry I will say going countries was' which does not mean anything. Both RA and RB share the use of mix-languages depending on the country or the region they are spoken in. In North Africa, RA is mixed mostly with Berber, French or English[7] and in the Middle East, it is mixed with English and some other languages. The same is applicable for Berber where some dialects use lots of French and Maghrebi Arabic words whereas others use only Maghrebi Arabic words for historical reasons.

4 Linguistic Resources

The main challenge in automatically processing any under-resourced natural language, using supervised machine learning approaches, is the lack of human annotated data. To overcome this serious hindrance, we created linguistic resources, namely corpora, for both RA and RB which are commonly used in social media. They are also used for commenting on events/news published on news agencies websites. In its standard form, RB is also used in media. We compiled a list of websites and platforms (micro-blogs, forums, blogs and online newspapers from all over the Arab world to ensure that many Arabic dialects are included) where RA and RB are used. Both manually and using a script, we collected content roughly published between 2013 and 2016. We collected 20,000 documents (144,535 words)[8] for RA and 7,000 documents for RB (31,274 words) from North Africa including various dialects[9] as well. Data collection and cleaning took us two months. We made sure to include various word spellings for both languages.

The included documents are short, between 2 and 236 words, basically product reviews, comments and opinions on quite varied topics. In terms of data source distribution, for RA, the majority of the content is comments collected from popular TV-show YouTube channels (9,800 documents, 49% of the data), content of blogs and forums (3,600 documents, 18% of the data), news websites (2,800 documents, 14 % of the data), the rest comes from Twitter (2,400 documents, 12% of the data) and Facebook (1,000 documents, 5% of the data). For RB, most content comes from Berber websites promoting Berber

[6]We use Buckwalter Arabic transliteration scheme. For the complete chart see: `http://www.qamus.org/transliteration.htm`.

[7]Based on the data used in this paper.

[8]By document, we mean a piece of text containing between 1 to 5 sentences; approximately 2 - 236 words. It is hard to precisely say how many sentences there are in each document because users use punctuation inconsistently in micro-blogs.

[9]Berber has 13 distinguished varieties. Here, we include only the six most popular dialects, namely Kabyle, Tachelhit, Tarifit, Tachawit, Tachenwit and Tamzabit.

culture and language (4,900 documents, 70%), YouTube (910 documents, 13%), news websites (700 documents, 10%) and Facebook (490 documents, 7%). With the help of two Arabic native speakers (Algerian and Lebanese) who are familiar with other Arabic dialects, we cleaned the collected data and manually checked that all the documents are written in RA. The same for RB, the platforms from which we collected data are 100% Berber and a Berber native speaker (Algerian) checked the data. For Romanized Arabic, it is hard for an Arabic speaker not to recognize Arabic and the task is easy (is a text written in Arabic or not) compared to classifying Arabic dialects (finding which Arabic variety a text is written in). The same is applicable for RB. Therefore, we consider the inter-annotator agreement (IAA) to be satisfactory. We are expending the RA and RB corpora and planning to use human annotators to compute the IAA using Cohen's kappa coefficient[10].

RA and RB use lots of mix-languages[11]. Consequently, we allowed mix-language documents[12] given that they contain clearly Arabic/Berber words in Latin script and a native speaker can understand/produce the same (sounds very natural for a native speaker). A preliminary study of the collected corpus showed that Berber (only for data collected from North Africa), French and English are the most commonly used languages with RA. Berber uses lots of French words and many Arabic words for some dialects like Tamzabit and Tachawit. It is also important to mention that in the entire Romanized Arabic corpus, only four (4) documents (0.02%) were actually written in Modern Standard Arabic (MSA) and the rest of documents were written in different Arabic dialects[13]. This indicates clearly that RA is commonly used to write dialectal Arabic. In terms of the dialectal distribution of the collected data, we noticed that most of the content in RA comes from North Africa (Maghrebi and Egyptian Arabic) and less from Levantine Arabic (mainly from Lebanon) and even less in Gulf and Kuwaiti/Iraqi Arabic.

Our corpora contain a mixture of languages (Arabic, Berber, English and French words all in Latin script). Also some German, Italian, Spanish and Swedish content is found, but not that frequent compared to English and French. This has motivated our choice to build a system which is able to distinguish between all these co-existing languages. In addition, we thought it would be good to add Maltese and Romanized Persian languages. The decision of adding Maltese language is based on the fact that Maltese is the only Semitic language written in Latin script in its standard form. This means that it has lots of common vocabulary with Arabic, namely Tunisian dialect[14]. We would like to add the Cypriot Arabic[15] variety written in Latin (not the variety using the Greek script), but unfortunately we could not collect enough data. We hardly collected 53 documents (287 words). We also added Romanized Persian (RP) language since Persian is one of the few non-Semitic languages that uses the Arabic script in its standard form. It has many false friends with Arabic, i.e. sharing the same word forms (spelling) but having different meanings. This causes an automatic language identifier to get confused easily when dealing with short texts. In addition, we would like to add Romanized Pashto[16] to the collection, but as with Cypriot Arabic we found it hard to collect enough data and find a native speaker to check it.

In addition to the data collected for RA and RB, we have collected, from social media platforms and news websites, 1,000 documents (6,000 - 10,000 words) for each of the mentioned languages (English (EN), French (FR), Maltese (ML), Romanized Persian (RP)) with the help of a native speaker of each language. From the entire data set, we removed 500 documents (around 6,000 words) for each language to be used in training and evaluating our system. We used the rest of the data to compile lexicons, for each language, by extracting the unique vocabulary using a script. We also used external lexicon for RB.

[10] A standard metric used to evaluate the quality of a set of annotations in classification tasks.

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

[12] Documents containing vocabulary of different languages. In our case, Arabic written in Latin script plus Berber, English, French, German, Spanish and Swedish words.

[13] Including Algerian, Egyptian, Gulf, Kuwaiti/Iraqi, Levantine, Moroccan and Tunisian Arabic.

[14] Being familiar with north African Arabic dialects, we have noticed that Maltese is much closer to Tunisian Arabic.

[15] An Arabic dialect spoken in Cyprus by the Maronite community and which is too close to Levantine Arabic for historical reasons and when written in Latin script, it is easily confused with Romanized Arabic.

[16] Pashto, an Easter Iranian language belonging to Indo-European family, is an official language of Pakistan. It has its own script but when written in Latin script, it has many false friends with Romanized Arabic.

We manually cleaned the word lists and kept only the clearly vocabulary in one of the corresponding mentioned languages (this took us almost two months). We were left with clean lexicons of more than 46,000 unique words for RA, 35,100 for RB and 2,700 for RP. Still RA and RB lexicons contain various spellings for the same word. In the absence of a reference orthography, we allowed all possible spellings (as found in the data) and introduce some normalization rules, namely lower-casing of all characters and the reduction of all the repeated adjacent characters to a maximum of two. For instance, all the words *'kbir', 'kbiiir' and 'kbiiiiiir'* refer to the same Arabic word 'big' with different emphasis. We should have reduced all the repeated characters to one occurrence as the doubling does not add much meaning to the word. This would be aggressive for EN, FR and ML which allow two consecutive repeated characters. For RB, we simply included all the possible spellings for each word as found in our corpus. The normalized lexicons contain 42,000 unique words for RA and 35,100 for RB. We added extra lexicons for both EN and FR (containing 14,000 and 8,400 unique words respectively). The same for ML, we used an extra list including 4,516,286 words. The added extra lexicons include different morphological inflections of the same word.

5 Methods and Experiments

Various methods have been applied to Automatic Language Identification since early 1960's. In this paper, we use two techniques of supervised machine learning, namely Cavnar's method and Support Vector Machines (SVM). As features, we experiment with both character-based and word based n-grams of different lengths. We use the term frequency-inverse document frequency[17] (TF-IFD) scheme to weight the importance of the features. Both methods require training data which we pre-processed to filter unimportant tokens such as punctuation, emoticons, etc. We also want to build an automatic language identifier which learns linguistic information rather than learning topical and country specific words. Therefore, we remove all Named Entities (NE) such as names of people, organizations and locations using a large NE database which includes both RA and RB NEs we compiled for an ongoing project. For experiments, we use a balanced dataset of 500 documents (between 4,506 - 117,000 words) for each language (total of 3,000 documents or 640,207 words) divided into 1,800 documents or 420,300 words (300 documents for each language) for training and the remaining 1,200 documents or 219,907 words for evaluation. As mentioned before, a document is an entire user's comment which may contain between 2 to 5 sentences depending on the social media platform.

5.1 Cavnar's Method

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 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 reasons which which are 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 no longer informative, i.e. do not really represent a given language or 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[18]. At the end, the language with the minimum distance from the source text will be the identified language.

We implement the Cavnar's classifier as described above. We experimented with different text and n-gram lengths. We found that bigrams outperform the rest of the character-based n-grams. Also increasing the text length increases the accuracy of the Cavnar's classifier. Table 1 shows the performance of the Cavnar's classifier per language for maximum text length of 140 characters (the maximum length of a Tweet) using character-based bigrams as features. The text length limitation to 140 characters means that we consider only the 140 first characters of each document. The purpose of doing this is to build a

[17] A statistical measure used to filter stop-words and keep only important words for each document.

[18] 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.

language identifier which is able to identify RA and RB regardless of the platform length restriction.

Language	Precision (%)	Recall (%)	F-score (%)
RA	**88.73**	**94.50**	**91.53**
RB	**97.50**	**97.50**	**97.50**
EN	94.29	99.00	96.56
FR	97.01	97.50	97.26
ML	97.50	97.50	97.50
RP	96.59	85.00	90.43

Table 1: Cavnar's classification per language using character-based bigrams.

For these settings, the macro-average F-score of the classifier is 95.13%. Overall, the results show that Cavnar's method is better at detecting text written in RB (F-score of 97.50%) compared to those written in RA. It performs slightly less for RA (F-score of 91.53%). An error analysis shows that the classifier is confused between RA and RP (21 times) and between RB and RP (3 times). The confusion is mainly caused by false friends and the use of the same vocabulary. Our purpose in using the Cavnar's method is to set its classification results as our baseline.

5.2 Support Vector Machines Classifier

We use the LinearSVC classifier (SVM) as implemented in Scikit-learn package (**?**)[19] with the default parameters. We experiment with both character and word n-grams as features. In both cases, we use the binary classification setting[20] as opposed to the 6-class classification. For instance, 'is a document written in RB or something else (other language)?' as opposed to 'is a document written in RB, EN, FR, ML, RA or RP?'.

5.2.1 Experiment 1

We use text maximum length of 140 characters when using character-based n-gram and text maximum length of 15 words[21] for word-based n-grams. The classification results are shown in Table 2.

	Accuracy (%)	
Features	**Character-based**	**Word-based**
Unigram	95.33	**95.91**
Bigrams	98.41	73.41
Trigrams	98.49	41.91
4-grams	98.66	27.91
5-grams	**98.75**	21.41
1+2-grams	98.25	94.66
1+3-grams	98.57	94.58

Table 2: SVM performance using different features.

For character-based n-grams, increasing the length of the n-gram improves the classification, 5-grams outperform all the rest of the n-gram lengths (5-grams have access to more information compared to shorter n-grams), giving 98.75% accuracy. Also, combining character-based unigram with trigrams has a positive effect on the classification; the accuracy has slightly increased to 98.57% compared to using only unigram or trigrams, 95.33% and 98.49% respectively. However, increasing the length of the word-based n-gram decreases the classifier's performance. This is caused by the data sparsity where it is

[19]For more information see: http://scikit-learn.org/stable/.

[20]We also experimented with the 6-class classification setting, and we found that the results were close to the binary classification.

[21]The choice of maximum 15 words is arbitrary for the sake of illustration. Still the focus is on short texts.

unlikely for long matches to occur frequently by chance. The word-based unigram scores the best, with an accuracy of 95.91%. Table 3 shows the performance of the SVM classifier per language using the combination of character-based unigram and trigrams for text with a maximum length of 140 characters.

Language	Precision (%)	Recall (%)	F-score (%)
RA	**98.98**	**97.50**	**98.24**
RB	**99.01**	**100**	**99.50**
EN	98.51	99.00	98.75
FR	99.00	99.00	99.00
ML	99.50	99.00	99.25
RP	97.51	98.00	97.76

Table 3: SVM classification using the combination of character-based unigram and trigrams.

The macro-average F-score of the SVM is 98.75%. Overall, the classifier identifies accurately RB (F-score of 99.50%) as well as RA (F-score of 98.24%). The SVM method performs better than the Cavnar's classifier (the baseline). The top-3 classification errors of the SVM are confusions between RA and RP (3 times), RA and FR (2 times) and between RP and RA (2 times). All the confused documents are very short (less than 10 words in our case).

5.2.2 Experiment 2

In another experiment, we use the same previous experimental setup but this time we combine the word-based unigram with the entries of the compiled lexicons as features. The SVM classifier accuracy has slightly improved to 97.50% compared to using only the word unigram 95.91%. This indicates that combining the word unigram with the lexicon entries (language-specific word) has a positive effect on the classification. Still there is confusion between RP and RA caused mainly by false friends. Furthermore, we combine character-based 5-grams with the entries of the compiled lexicons using the same experimental setup. The SVM accuracy has increased to 99.02%. Table 4 summarizes the SVM performance using the combination of character-based 5-grams and the entries of the compiled lexicons as features.

Language	Precision (%)	Recall (%)	F-score (%)
RA	97.04	98.50	**97.77**
RB	100	99.50	**99.75**
EN	99.00	99.50	99.25
FR	99.01	100	99.50
ML	99.50	99.50	99.50
RP	99.49	97.00	98.23

Table 4: SVM classification using the combination of character-based 5-grams and lexicons.

The classifier's macro-average F-score is 99.00%. Using the combination of the character-based 5-grams and the entries of the compiled lexicons as features has improved the overall accuracy of the SVM 99.02% compared to 98.75% using only character-based 5-grams. It has also positive effect on each language except for RA where the F-score has slightly decreased to 97.77% compared to 98.24%.

5.2.3 Experiment 3

To be able to compare the word-based n-grams and character-based n-grams, we rerun the same experiment using text full length. Still, the character-based 5-grams outperform the word-based n-grams, F-score of 99.77% and 97.89% respectively.

There are a few misclassifications between different languages as shown in Table 5. The few errors are caused by false friends between close/similar languages such as RA and RP and also in case of mix-languages, for instance between RA and FR where the former uses lots of words from the latter. An error

analysis of the sample shows that most errors occurred in very short documents (less than 10 words in our case).

Correct languages	Misclassified languages					
		RA	***RB***	***EN***	***FR***	***ML***	***RP***
	RA	196	0	0	2	0	2
	RB	0	199	0	0	0	1
	EN	0	0	198	0	1	1
	FR	0	0	0	200	0	0
	ML	0	0	1	0	199	0
	RP	5	0	1	0	0	194

Table 5: The confusion matrix of the system for the same settings as in Table 4.

6 Conclusion

We have described the linguistic resources built and used to train and evaluate our Romanized Arabic (RA) and Romanized Berber (RB) Automatic Language Identification (ALI) tool. We used supervised machine learning techniques, using various features. The focus is on short documents (social media domain) maximum text length of 140 characters or 15 words approximately, and the language identification is done at the document level. We assume that if the system works well for short documents, it should work better for longer ones since it will have access to more information. We found that using character based 5-grams perform reasonably well in detecting both RA and RB and slightly better than word-based unigram. Combining both character-based 5-grams and word-based unigram with the compiled lexicons has improved the SVM overall performance. In all cases, the SVM classifier outperformed our baseline (Cavnar's classifier). Our main purpose in this paper is to apply some ALI standard methods to Romanized Berber (RB) and Romanized Arabic (RA) rather than proposing new methods. Our motivation is that the existing ALI methods have not been not applied to neither RA or RB.

In this paper, we used a small sample of the data for training and testing. The limited text length allowed in social medial platforms, using very short documents (2-250 tokens), can be seen as distinguishing between the included languages at a sentence level especially that punctuation is mostly ignored. As a future work, we are planning to test our system on large dataset. We want to identify each RA and RB varieties. We want also to transliterate the compiled RA lexicon into the Arabic script, both dialectal Arabic and Modern Standard Arabic (MSA) equivalents. We believe that this will help in adapting the existing Arabic Natural Language Processing tools. The collected corpora are valuable for the Automated Identification of RA and RB, but also for linguistic and sociolinguistic research, as well as further applications in both language groups. Therefore, the datasets are freely available for research from the first author.

References

Cyril Goutte, Serge Léger and Marine Carpuat. 2014. The NRC System for Discriminating Similar Languages. *In the Proceedings of the VarDial Workshop.*

David Sankoff. 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, pages 8–21.

Eirlys Davies, Abdelâli Bentahila and Jonathan Owens. 2013. Codeswitching and related issues involving Arabic. *Oxford Handbook of Arabic Linguistics, Sociolinguistics*, pages 326–348.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Machine learning in Python. Journal of Machine Learning Research*, 12, pages 2825–2830.

Fatma Zohra Chelali, Khadidja Sadeddine and Amar Djeradi. 2015. Speaker identification system using LPC-Application on Berber language. *HDSKD journal*, 1(2):29–46.

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

Kareem Darwish. 2014. Arabizi Detection and Conversion to Arabic. *In the Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP).Doha, Qatar*, pages 217–224.

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

Mohamed Al-Badrashiny, Ramy Eskander, Nizar Habash and Owen Rambow. 2014. Automatic Transliteration of Romanized Dialectal Arabic. *In the Proceedings of the Eighteenth Conference on Computational Language Learning, Baltimore, Maryland USA*, pages 30–38.

Marco Lui and Timothy Baldwin. 2012. langid. py: An Off-the-shelf Language Identification Tool. *In Proceedings of the ACL*.

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

Omar F. Zaidan and Chris Callison-Burch. 2011. The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. *In the Proceedings of the Association for Computational Linguistics (ACL)*, pages 37–41.

Peter Behnstadt and Manfred Woidich. 2013. Diactology. *In the Oxford Handbook of Arabic Linguistics*.

Ramy Eskander, Mohamed Al-Badrashiny, Nizar Habash and Owen Rambow. 2014. Foreign Words and the Automatic Processing of Arabic Social Media Text Written in Roman Script. *In the Proceedings of The First Workshop on Computational Approaches to Code Switching. Doha, Qatar*, pages 1–12.

Ramzi Halimouche, Hocine Teffahi and Leila Falek. 2014. Detecting Sentences Types in Berber Language. *International Conference on Multimedia Computing and Systems (ICMCS)*, pages 197–200.

Shervin Malmasi, Eshrag Refaee and Mark Dras. 2015. Arabic Dialect Identification using a Parallel Multidialectal Corpus. *In the Proceedings of the 14th Conference of the Pacific Association for Computational Linguistics (PACLING 2015)*, pages 209–217, Bali, Indonesia.

Wafia Adouane. 2016. *Automatic Detection of Under-resourced Languages: The case of Arabic Short Texts*. Master's thesis, University of Gothenburg.

William B. Cavnar and John M. Trenkle. 1994. N-Gram-Based Text Categorization. *In the Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, University of Nevada, Las Vegas*.