3rabizi Language Identification

Andrea Ceres and Vivek Sharma

Abstract

Language identification of a Romanized texting language does not have as robust an implementation as that for official written languages. This project utilizes supervised Machine Learning (Naive Bayes and Support Vector Machine classifiers) in order to identify Arabizi, the Latin-script language used by Arabic speakers throughout social media. Features containing character n-grams (n=3 to 6) and word n-grams (n=1 to 3) are compared. Modest results reflect unresolved difficulties with acquiring enough data in Arabizi and with properly accounting for the imbalanced dataset.

1. Introduction

Language identification is a preliminary step prior to machine translation. Libraries, such as *langdetect* and *langid*, claim detection among nearly one hundred languages listed on the Library of Congress ISO 639 Code Tables. One deficit in current language identification approaches is the limitation of detection to conventionally written languages. In juxtaposition, since the advent of the smartphone, Romanized languages have played a significant role in social media communications. The Arabic texting language, Arabizi, refers to the transcription of dialectical Arabic using a Latin-script keyboard, such as QWERTY. This substitution of English characters for Arabic script is not readily identified by the common language detection libraries.

Language detection of Arabizi poses some challenges. Being a Romanization of spoken Arabic dialects, Arabizi is unlike Modern Standard Arabic (MSA) as Arabizi has regional vocabulary sense usage and does not have a sanctioned orthography. Some spelling conventions have arisen with uneven adoption. One such convention is the substitution of certain numerical digits for Arabic letter sounds that have no equivalent in the English alphabet. For instance, the phrase *9aba7 2l5air* ("*good morning"*) includes the following substitutions:

9 for ص

7 for ح

2 for ء

5 for خ

Furthermore, acquisition of texts containing only Arabizi is not a straightforward process, and availability of public Arabizi corpora is very limited.

2. Related Work

Much research has been done in the field of language detection. There are numerous academic papers on language identification of high-resource languages; however, research is more limited on Romanized languages such as Arabizi. The following subsections discuss approaches that have been taken by researchers in language detection of Arabizi.

2.1. Arabic Detection and Conversion to Arabic [1]

This research has two aspects, namely, Arabizi detection and Arabizi-to-Arabic transliteration. With regard to detection, the researchers used word-, character-, and sequence-level features to identify Arabizi. Training and test sets were constructed using word-level classification from tweets containing English, Arabizi, and a combination of both. Accuracy was reported as 98.5%. Noted drawbacks include the inability to distinguish Arabized English words, Arabizi words that happened to be English too, and simple Arabizi words surrounded by English words (and vice versa).

2.2. Romanized Berber and Romanized Arabic Automatic Language Identification Using Machine Learning [2]

According to the authors, Romanized Berber (RB) and Romanized Arabic (RA) are under-resourced, and detection of Arabizi is challenging because it has non-standard spelling, no fixed grammar, and regional vocabulary sense usage. The paper considers the detection of RA and RB as a task of classification and uses supervised Machine Learning to solve it. The researchers created character-based n-grams (2-5), sorted them, and considered 300 common n-grams (Canvar's method). Accuracy of character-based 5-gram was highest at 98.75%. Meanwhile, word-based n-gram had degraded performance as the value of n increased.

2.3. Language Identification Using Classifier Ensembles [3]

This research was not specific to Arabizi language detection but language detection of languages in general. The features used were character-level n-grams (1-6) and word-level unigram and bigram. An ensemble of several SVM-based classifiers were used to improve accuracy. The ensemble was combined using techniques such as Plurality Voting, Mean Probability Rule, Median Probability Rule, Product Rule, Highest Confidence, and Borda Count. The accuracy achieved was 95.54%.

2.4. LILI: A Simple Language Independent Approach for Language Identification [4]

This approach uses character-level n-grams (1-5) and word-level unigram. The paper introduces a powerful framework for Linguistic Code Switch detection using Conditional Random Fields (CRFs).

2.5. Automatic Detection of Arabicized Berber and Arabic Varieties [5]

The data for this study was collected from the social media domain: forums, blogs, Facebook, and Newsgroups. Supervised Machine Learning was utilized via Support Vector Machines (SVM), along with Canvar's text classification and with prediction by partial matching (PPM) using character- and word-based n-grams. This study achieved an F-score of 92.94%.

3. Experiment

3.1. Data Extraction and Datasets

As discussed above, there has been limited research on Arabizi language identification. The data that is publicly available is also limited. One corpus taken from a research study [6] contained two files: the original social media data in Arabizi and its translation into English.

A corpus from another study [7] was also used. This included two files from the social media domain such as blogs and forums: one from Egypt and the other from Lebanon. Each line in the files was labeled as Arabizi or English. For this project, the Lebanon file was chosen to be held out as unseen data from training.

One source used for English social media texts was the NLTK Twitter Samples corpus.

Additionally, tweets containing "*arab politics*" were extracted from Twitter using Twitter APIs. The topic was chosen to reflect a common general subject matter from the aforementioned Arabizi corpora. This English corpus was held out in the experiment as unseen data from training.

The following figure shows the process of our approach.

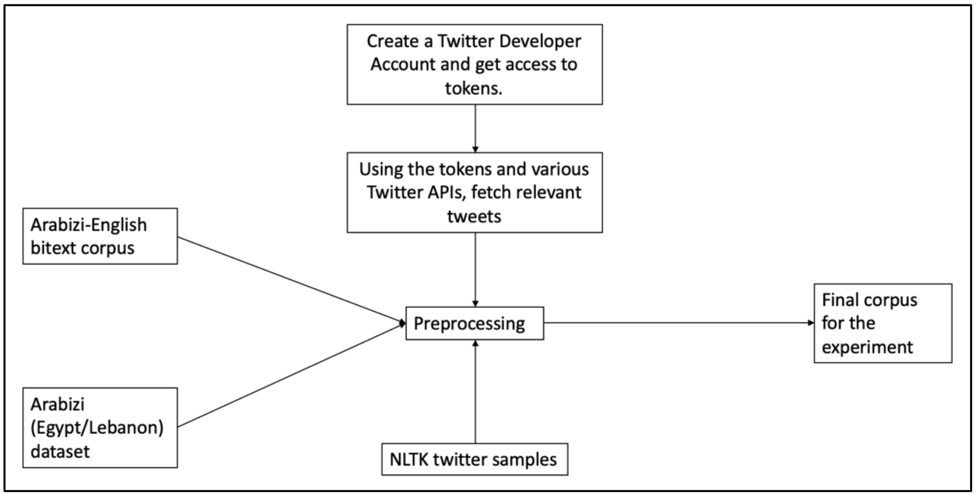
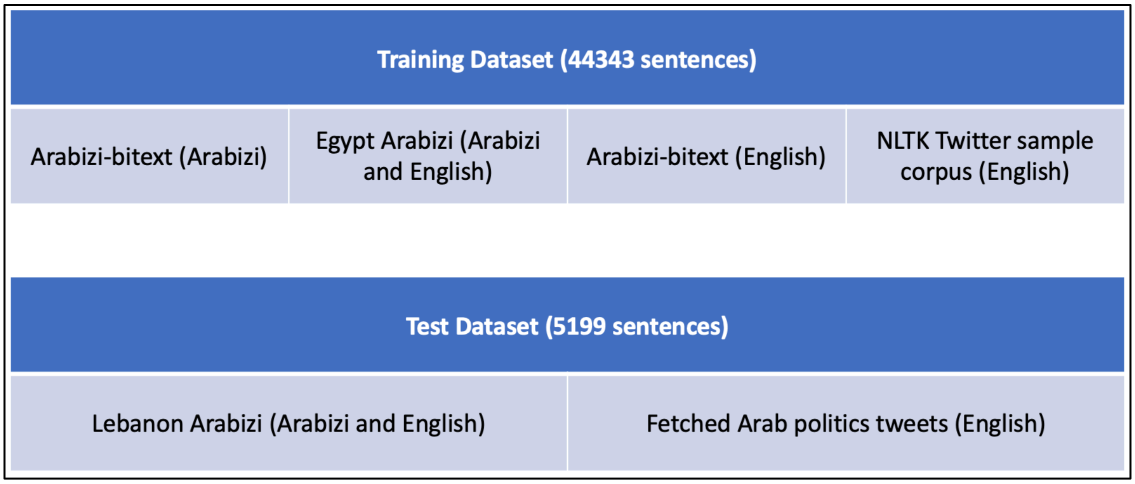

Create a Twitter Developer Account and get access to tokens.
Using the tokens and various Twitter APIs, fetch relevant tweets
Arabizi-English bitext corpus
Preprocessing
Final corpus for the experiment
Arabizi (Egypt/Lebanon) dataset
NLTK twitter samples

Figure 1 shows the process for final corpus creation.

The corpus was divided into training set and testing set. The division of data is as shown below.



Training Dataset (44343 sentences)

Arabizi-bitext (Arabizi)

Egypt Arabizi (Arabizi and English)

Arabizi-bitext (English)

NLTK Twitter sample corpus (English)

Test Dataset (5199 sentences)

Lebanon Arabizi (Arabizi and English)

Fetched Arab politics tweets (English)

Figure 2 shows division of data into training and testing set.

3.2. Features

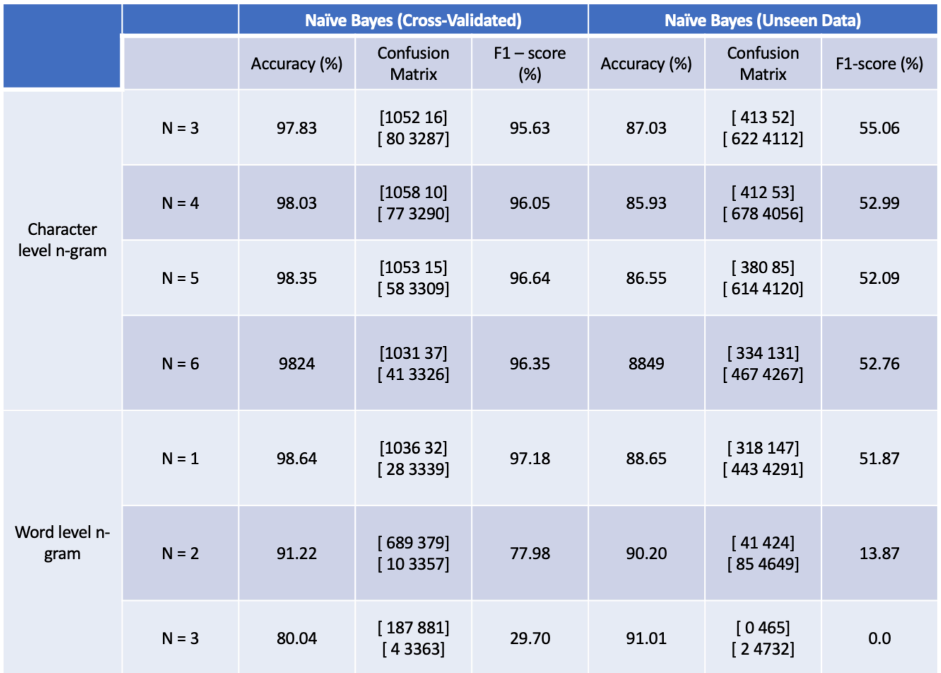
During the literature review, observations were made regarding feature selection. This project aimed to emulate some of these features, and compare the results. Nearly every research study on language identification contained two prime features: character-based and word-based n-grams. The value of n in both of the n-grams worked very differently. In character level n-grams, as the value of n increases until a certain point (generally 5 or 6), the accuracy of the model increases. However as the value of n increases in word-level n-grams, the accuracy drops drastically.

For this project, character-level n-grams (n=3 to 6) and word-level n-gram (n=1 to 3) were chosen as features to be compared.

4. Results

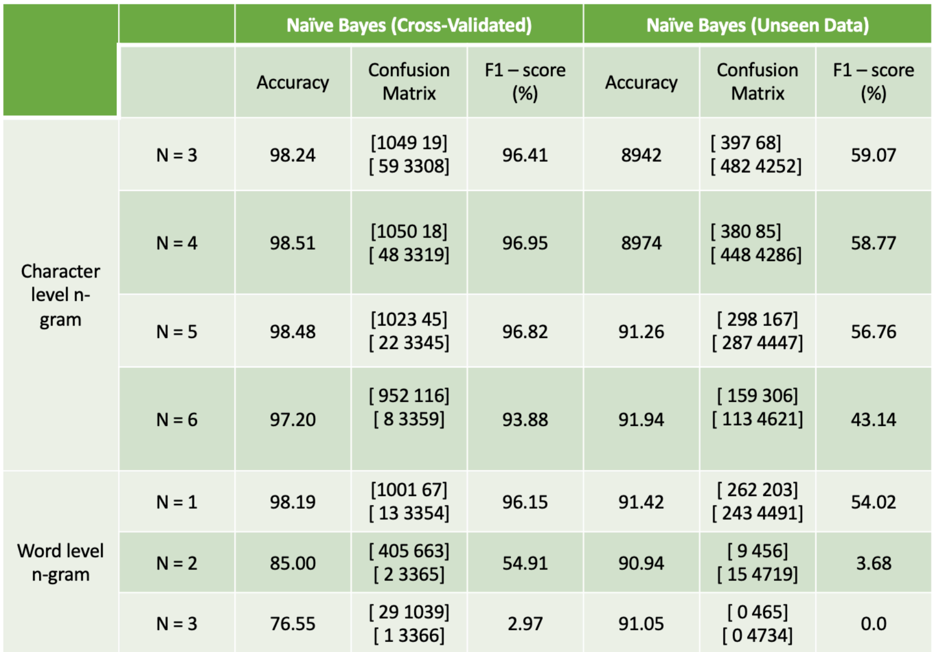
For the experiment, our model was trained using Naïve Bayes and Support Vector Machine classifiers. The training data was split into a 90/10 dataset for cross-validation. After training on the training data, the model was run again with the held out, unseen test data. The training and test datasets are both from the social media domain, but not from the same corpus.

The vectorization was performed using Count vectorization and TF-IDF vectorization. A look at the confusion matrix shows that the Arabizi language n-grams were predicted as English a lot of the time. Because of the focus on identifying Arabizi, accuracy is not the best parameter to judge this model. An F1-score was computed for each run of the model.



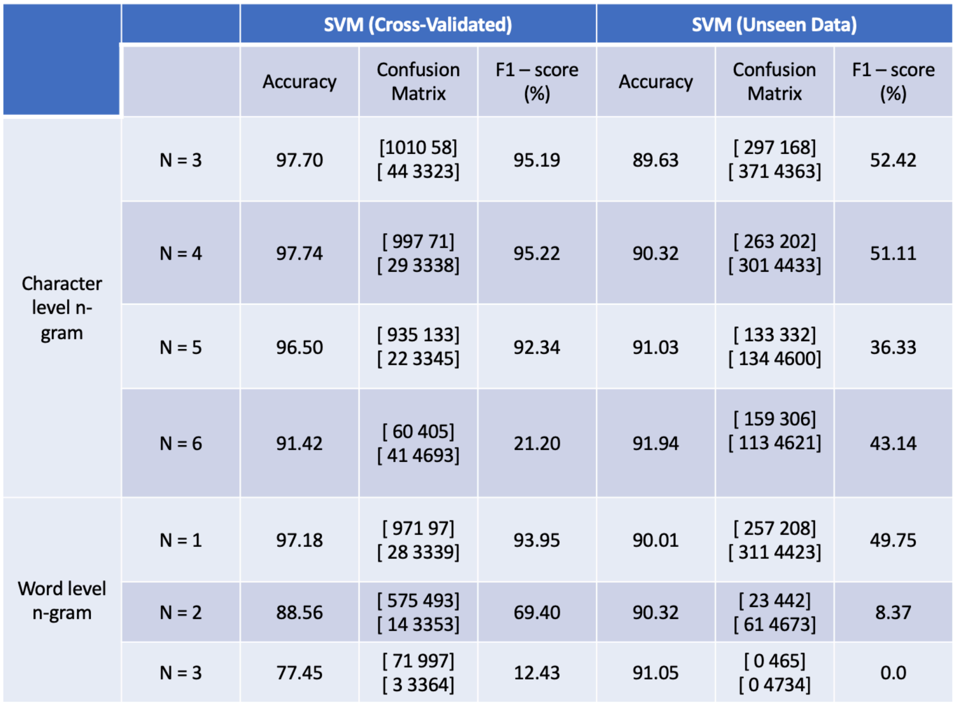
		Naïve Bayes (Cross-Validated)			Naïve Bayes (Unseen Data)		
		Accuracy (%)	Confusion Matrix	F1 – score (%)	Accuracy (%)	Confusion Matrix	F1-score (%)
Character level n-gram	N = 3	97.83	[1052 16] [80 3287]	95.63	87.03	[413 52] [622 4112]	55.06
	N = 4	98.03	[1058 10] [77 3290]	96.05	85.93	[412 53] [678 4056]	52.99
	N = 5	98.35	[1053 15] [58 3309]	96.64	86.55	[380 85] [614 4120]	52.09
	N = 6	9824	[1031 37] [41 3326]	96.35	8849	[334 131] [467 4267]	52.76
Word level n-gram	N = 1	98.64	[1036 32] [28 3339]	97.18	88.65	[318 147] [443 4291]	51.87
	N = 2	91.22	[689 379] [10 3357]	77.98	90.20	[41 424] [85 4649]	13.87
	N = 3	80.04	[187 881] [4 3363]	29.70	91.01	[0 465] [2 4732]	0.0

Figure 3 shows results of Naïve Bayes classification using Count vectorization



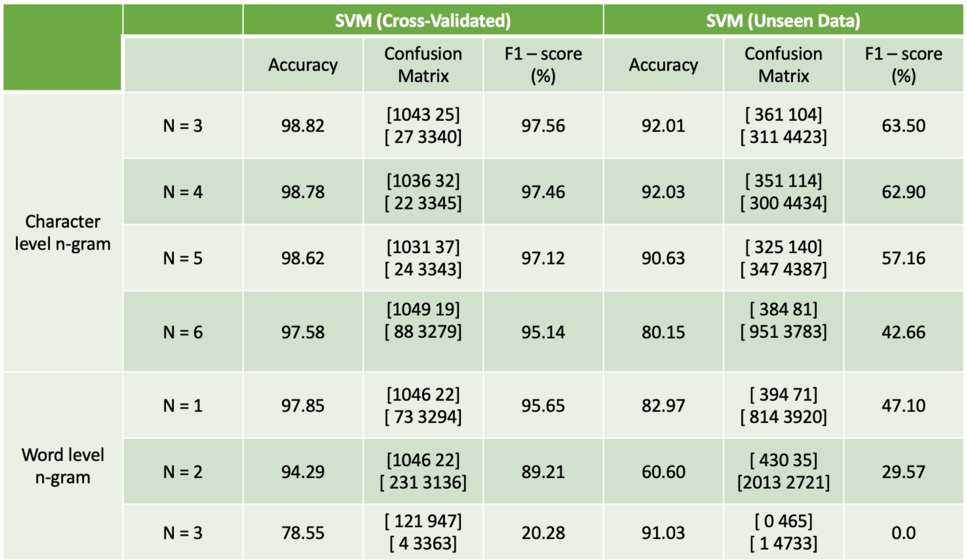
		Naïve Bayes (Cross-Validated)			Naïve Bayes (Unseen Data)		
		Accuracy	Confusion Matrix	F1 – score (%)	Accuracy	Confusion Matrix	F1 – score (%)
Character level n-gram	N = 3	98.24	[1049 19] [59 3308]	96.41	8942	[397 68] [482 4252]	59.07
	N = 4	98.51	[1050 18] [48 3319]	96.95	8974	[380 85] [448 4286]	58.77
	N = 5	98.48	[1023 45] [22 3345]	96.82	91.26	[298 167] [287 4447]	56.76
	N = 6	97.20	[952 116] [8 3359]	93.88	91.94	[159 306] [113 4621]	43.14
Word level n-gram	N = 1	98.19	[1001 67] [13 3354]	96.15	91.42	[262 203] [243 4491]	54.02
	N = 2	85.00	[405 663] [2 3365]	54.91	90.94	[9 456] [15 4719]	3.68
	N = 3	76.55	[29 1039] [1 3366]	2.97	91.05	[0 465] [0 4734]	0.0

Figure 4 shows results of Naïve Bayes classification using TF-IDF vectorization



		SVM (Cross-Validated)			SVM (Unseen Data)		
		Accuracy	Confusion Matrix	F1 – score (%)	Accuracy	Confusion Matrix	F1 – score (%)
Character level n-gram	N = 3	97.70	[1010 58] [44 3323]	95.19	89.63	[297 168] [371 4363]	52.42
	N = 4	97.74	[997 71] [29 3338]	95.22	90.32	[263 202] [301 4433]	51.11
	N = 5	96.50	[935 133] [22 3345]	92.34	91.03	[133 332] [134 4600]	36.33
	N = 6	91.42	[60 405] [41 4693]	21.20	91.94	[159 306] [113 4621]	43.14
Word level n-gram	N = 1	97.18	[971 97] [28 3339]	93.95	90.01	[257 208] [311 4423]	49.75
	N = 2	88.56	[575 493] [14 3353]	69.40	90.32	[23 442] [61 4673]	8.37
	N = 3	77.45	[71 997] [3 3364]	12.43	91.05	[0 465] [0 4734]	0.0

Figure 5 shows results of SVM classification using Count vectorization



		SVM (Cross-Validated)			SVM (Unseen Data)		
		Accuracy	Confusion Matrix	F1 – score (%)	Accuracy	Confusion Matrix	F1 – score (%)
Character level n-gram	N = 3	98.82	[1043 25] [27 3340]	97.56	92.01	[361 104] [311 4423]	63.50
	N = 4	98.78	[1036 32] [22 3345]	97.46	92.03	[351 114] [300 4434]	62.90
	N = 5	98.62	[1031 37] [24 3343]	97.12	90.63	[325 140] [347 4387]	57.16
	N = 6	97.58	[1049 19] [88 3279]	95.14	80.15	[384 81] [951 3783]	42.66
Word level n-gram	N = 1	97.85	[1046 22] [73 3294]	95.65	82.97	[394 71] [814 3920]	47.10
	N = 2	94.29	[1046 22] [231 3136]	89.21	60.60	[430 35] [2013 2721]	29.57
	N = 3	78.55	[121 947] [4 3363]	20.28	91.03	[0 465] [1 4733]	0.0

Figure 6 shows results of SVM classification using TF-IDF vectorization

A notable observation in line with literature review is that the increase in value of n in character-level n-grams improves the accuracy, whereas the increase of value of n in word-level n-grams degrades the accuracy drastically. Character trigrams consistently resulted in the highest F1-score, and TF-IDF vectorization outperformed Count vectorization.

The *sklearn.metric* formulas used for accuracy and F1-score are given below.

$$\boldsymbol{Accuracy}\left(\boldsymbol{y},\, \hat{\boldsymbol{y}}\right) = \frac{1}{\boldsymbol{n}} \sum_{\boldsymbol{i}=0}^{\boldsymbol{n_{samples}}-1} 1\, (\hat{\boldsymbol{y}}_{\boldsymbol{i}} = \boldsymbol{y}_{\boldsymbol{i}})$$

$$where\ \hat{y}_i\ is\ the\ predicted\ value\ of\ i^{th}\ and\ y_i\ is\ the\ corresponding\ true\ value$$

$$\boldsymbol{F}1 = \frac{2*(\boldsymbol{Precision}*\boldsymbol{Recall})}{(\boldsymbol{Precision} + \boldsymbol{Recall})}$$

The F1-score ranges from ~50-60% in both implementations. A low score was anticipated in part due to the fact that the test set was sourced differently than the training set.

5. Discussion

The primary use for an implementation of Arabizi language identification is as a step prior to machine translation. Such a set-up may also be used to develop a chat bot system that automatically provides support in Arabic based on user input. Furthermore, as with the difficulty in acquiring Arabizi text for this project, an existing Arabizi detection tool may assist researchers aiming to pull Arabic language tweets from Twitter (or text from other social media) that include those texts that do not use Arabic script. Such an API pull may more accurately reflect the content created by Arabic-speaking users of social media.

There are several drawbacks to the project described in this report. Both accuracy and F1 scores do not surpass those from the referenced academic papers. Such results may be improved with more training data and more experimentation with features. In particular, time and care in acquiring more Arabizi texts are needed to achieve this. Additionally, once sufficient Arabizi data is acquired, the Arabizi and English labeled datasets should be balanced prior to training the model.

Due to this already limited Arabizi corpus, Arabic dialects are lumped together in the experiment, even though transcription of texts with the same semantic meaning vary considerably among the different Arabic dialects. Separate corpora for each of the main dialects may be used in the future to train for a more refined language dialect identification.

Lastly, code switching (such as between Arabic and English or between Arabic and French) was out-of-scope for this experiment, even though code switching has a significant presence in social media text. Further literature review into effective approaches to multilingual sentences is needed.

6. References

[1] K. Darwish, “Arabizi Detection and Conversion to Arabic,” arXiv preprint arXiv:1306.6755, 2013.
[2] W. Adouane, N. Semmar, and R. Johansson, “Romanized Berber and Romanized Arabic Automatic Language Identification Using Machine Learning,” in Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3), pp. 53–61, 2016.
[3] S. Malmasi and M. Dras, “Language Identification Using Classifier Ensembles,” in Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects, pp. 35–43, 2015.
[4] M. Al-Badrashiny and M. Diab, “Lili: A Simple Language Independent Approach for Language Identification,” in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1211–1219, 2016.
[5] W. Adouane, N. Semmar, R. Johansson, and V. Bobicev, “Automatic Detection of Arabicized Berber and Arabic Varieties,” in Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3), pp. 63–72, 2016.
[6] https://ilps.science.uva.nl/resources/arabizi
[7] https://project-rbz.kmi.open.ac.uk