Arabizi Language Models for Sentiment Analysis

Souhir Gahbiche and **Guillaume Gadek**
Airbus, Advanced Information Processing, France

Gaétan Baert and **Alexandre Pauchet**
Litis, INSA Rouen Normandie
alexandre.pauchet@insa-rouen.fr

Abstract

Arabizi is a written form of spoken Arabic, relying on Latin characters and digits. It is informal and does not follow any conventional rules, raising many NLP challenges. In particular, Arabizi has recently emerged as the Arabic language in online social networks, becoming of great interest for opinion mining and sentiment analysis. Unfortunately, only few Arabizi resources exist and state-of-the-art language models such as BERT do not consider Arabizi.

In this work, we construct and release two datasets: (i) LAD, a corpus of 7.7M tweets written in Arabizi and (ii) SALAD, a subset of LAD, manually annotated for sentiment analysis. Then, a BERT architecture is pre-trained on LAD, in order to create and distribute an Arabizi language model called BAERT. We show that a language model (BAERT) pre-trained on a large corpus (LAD) in the same language (Arabizi) as that of the fine-tuning dataset (SALAD), outperforms a state-of-the-art multi-lingual pretrained model (multilingual BERT) on a sentiment analysis task.

1 Introduction

Nowadays, Online Social Networks (OSNs) are highly popular means of communication. With pictures, videos, urls, mentions and of course comments, people share ideas, opinions and sentiments on Twitter[1], Facebook[2], LinkedIn[3] and many other service providers. OSN posts are of great interest for marketing and reputation management, enabling companies to collect feedback about brands and products. Automatic systems facilitating these large-scale tasks, including sentiment classifiers, have progressively reached promising accuracy (Abbasi et al., 2008; Catal and Nangir, 2017; Giatsoglou et al., 2017).

In parallel, the NLP field has experienced several scientific developments until the recent language models. Efficient representation of texts with vectors is difficult and this challenge has seen excellent work: bag-of-words with tf*idf (Robertson et al., 1995), Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Joulin et al., 2016), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2018) for instance. Each of these scientific steps has successively outperformed the previous ones on a variety of NLP tasks including sentiment analysis, with a focus on English texts. Moreover, some approaches such as BERT propose pre-trained language models, including a multi-lingual version.

The particularity of texts post on OSN is that they frequently mix made-up terms and inventive spelling with more classic syntax. Twitter, for instance, contains invented words, original spelling and typing errors (Maynard and Funk, 2011), but also includes special entities such as hashtags and user mentions that often appear as labels without any syntactic role (Gadek et al., 2017). Even state-of-the-art architectures such as BERT usually suffer from the particularities of OSN languages, as illustrated on a task of hate detection in tweets (Gertner et al., 2019).

Moreover, OSN posts are also the opportunity for code-switching, i.e. alternating between two or more alphabets and/or languages in conversation, in particular for many Arabic speakers. As they often are familiar with Latin keyboards and use them frequently, writing Arabic using Latin alphabet is easier and faster than switching between alphabets. This typing method, called Arabizi, enables to express

[1] https://twitter.com
[2] https://www.facebook.com
[3] https://www.linkedin.com

Proceedings of the 28th International Conference on Computational Linguistics, pages 592–603
Barcelona, Spain (Online), December 8-13, 2020

both Modern Standard Arabic (MSA) or Arabic dialects. Arabizi is a non-standard romanization of Arabic script that is widely adopted for communication over the Internet or for sending messages (Bies et al., 2014). It has emerged as the Arabic language of informal communications online. Arabizi appears in several challenges for computational linguistics: 1) Arabizi is a written form of non-written Arabic dialects; 2) there exists no broadly accepted rules or incitement to correctly write Arabizi: inventive spelling is therefore a standard. This language diversity rises significant challenges to NLP: Arabizi is not "official", does not follow any rules to be written and is often mixed with other foreign languages. Unfortunately, only very few Arabizi resources are publicly available and existing state-of-the-art language models such as BERT, even in its multilingual version, are not optimised to process Arabizi posts on OSN.

This article contributes to the study of user-generated contents in Arabizi on OSN, by releasing a 7.7M tweet corpus, called LAD (Large Arabizi Dataset). A subset of these tweets, named SALAD (Sentiment Annotations from LAD), has been manually annotated according to their sentiment. We illustrate the interest of such resources on a sentiment analysis classification task, with an Arabizi-trained BERT-based language model (BAERT). We aim at enabling researchers to exploit LAD on Arabizi NLP tasks and SALAD to evaluate the performance of sentiment analysers on Arabizi tweets; the size of SALAD may be insufficient to train greedy machine learning models.

The remaining of this article is organised as follows. Section 2 presents the characteristics of Arabizi and introduces some works on sentiment analysis for Arabizi. Section 3 introduces our collected corpus of Twitter data named LAD, as well as the pre-processing and annotation tasks that lead to SALAD (Sentiment Annotations from LAD). Section 4 presents some experiments comparing a multilingual BERT model with our Arabizi BAERT model, pre-trained on LAD and tested on SALAD; the results are presented in Section 5. Finally, Section 6 concludes this article.

2 Arabizi, Sentiment Analysis and Language Modeling

This section introduces the linguistic phenomenon of Arabizi and covers the technical aspects of text classification models and their adequacy to Arabizi.

2.1 Arabizi as a challenge fo NLP

Arabic language is spoken in twenty-two countries, by about 447 million of speakers. It is the fourth most used language on the Web[4]. Modern Standard Arabic (MSA), the official written form of Arabic, is generally not spoken as a mother tongue, but is mostly used in administrative documents, classrooms, movie subtitles and official news.

Spoken Arabic exists as dialects that differ according to countries or even regions. Dialect is the every-day-life language of the Arab world: at home, at the grocer's, with friends and even at school out of classrooms. Arabic dialects differ one from another, depending on the historical events of each country (protectorate, colonisation, sovereignty, ...) and on the geographic location. Country dialects are influenced by the former occupants and by the neighbouring countries. Many classifications of dialects have been done. (Cotterell and Callison-Burch, 2014) for instance has classified dialects into five groups: Maghrebi (spoken in the whole North Africa), Egyptian (spoken in Egypt, but understood universally), Levantine (spoken primarily in the Levant, Syria and Palestine), Iraqi (spoken in Iraq) and Gulf (spoken primarily in Saudi Arabi, UAE, Kuwait and Qatar). Until the early 2000s, the Arabic dialects were not written languages, but new technologies (internet, email, SMS, blogs, ...) have prompted Arabic speakers to write in their own dialect to facilitate communication. Since most people have Latin keyboards, Arabic in mostly written in Latin characters, leading to the emergence of Arabizi. Arabizi is the contraction of *arabi* (Arabic) and *inglizi* (English); a second etymology could be the union of *arabi* and *easy* (Gonzalez-Quijano, 2014). Nowadays, Arabizi is the new written form of Arabic all over Internet. It is a rewritten form of Arabic using a combination of Latin characters and numbers (LC+N). These numbers represent the letters whose sound does not exist in the Latin alphabet. For example, the number 7 often represents

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

the sound *ha* (ح letter), the number 3 matches the sound *ain* (ع letter).

With the intrinsic diversity of OSN, Arabizi reveals a few difficulties to adapt classic NLP solutions. Arabizi is an informal language: no rule officially defines transliterations and several possibilities appear. A single sound may have different representations in various dialects, such as the letter *qaf* that can be represented by *q, 9, k, g, ...* As a result, words do not have any official spelling convention. For instance, (Cotterell et al., 2014) has identified 69 different ways to write انشَاء الّله (Insh'Allah, God willing). The main differences arise in the pronunciation of vowels and different representations of letters depending on countries. For example, the number 9 represents the letter *sad* (ص) in the Middle East countries, and *qaf* (ق) in the Maghreb. These differences appear even within the same country (Allehaiby, 2013). Finally, Arabizi is mainly used on OSN and therefore exhibits the same inventive usage than in other languages (typos, inventive spellings, emojis, ...), maybe even more as the written representation of dialects. For instance, a letter can be repeated to express feelings, such as *mbrooook!* (which approximately means *congratulatiooooons!*. As most NLP systems are token-based, matching the correct word requires intensive character permutations, repetitions and deletions.

Moreover, Arabizi usually appears tied with another language, such as English or French, depending on the second language spoken in the country. For instance, *"What's your name ba2a?"* (*ba2a* means *then*) contains only one word in Arabizi. However, most language models are pre-trained on sentences written in a single language. Even the multilingual models are pre-trained with a mix of languages at a higher level than the sentence. A language model for Arabizi should be tolerant to the presence of other languages in the same sentence, i.e. with several syntactic rules and semantics for a same meaning.

2.2 Arabizi and Sentiment Analysis

Sentiment Analysis (SA) is one of the most interesting NLP tasks with many applications in various areas. Brands, companies and services can exploit SA to automatically collect user opinions, in particular on OSN. As Arabizi is widely used online, SA on Arabizi messages seems an evidence.

Before any sentiment process, Arabizi has to be recognised. (Darwish, 2014) identifies Arabizi at the word-level: a CRF classifier handles trigrams of characters, and has been trained on a binary classification task (English versus Arabizi). However, language identification is more commonly realised at sentence-level or document-level (Tobaili, 2016).

A frequent intuition to process Arabizi texts is to convert them into MSA thanks to expert rules (Eskander et al., 2014), or machine learning (Bies et al., 2014). For instance, (Duwairi et al., 2016) transcript Arabizi Tweets to Arabic with a rule-based converter, and then annotate the resultant tweets according to a sentiment (positive, negative or neutral) thanks to crowdsourcing. In (Guellil et al., 2018), the approach consists to automatically classify sentiments of Algerian Arabizi after transliterating to MSA. (Tobaili et al., 2019) has created a sentiment lexicon for Arabizi. In a very recent work, (Fourati et al., 2020) has released a 3,000-comments sentiment analysis dataset named TUNIZI[5], which has been collected from social networks, preprocessed and manually annotated by Tunisian native speakers.

All these works have constituted corpora that could be exploited for SA on Arabizi, particularly on SMS [6]. Unfortunately, these datasets are not publicly available and/or not sufficiently large: 101,292 messages in (Bies et al., 2014), 10,000 triples arabizi-MSA-English in the NIST campaign. Even if they enable relevant specific classification models, they are too small to generalise on similar tasks with

[5] https://github.com/chaymafourati/TUNIZI-Sentiment-Analysis-Tunisian-Arabizi-Dataset/blob/master/TUNIZI-Dataset.txt

[6] E.g. NIST campaign: https://www.nist.gov/itl/iad/mig/openmt-challenge-2015

slightly different scopes. Concerning Arabic, a few resources exist, such as the annotated Arabic Sentiment Tweets Dataset gathered from Twitter and consisting of about 10,000 tweets (Nabil et al., 2015). These tweets are a mixture of MSA and dialect, but written in Arabic alphabet.

2.3 Language Models

Independently of Arabizi, scientific advancements have reached impressive results on various NLP tasks, up to the recent emergence of *deep* language models. Recurrent neural network models (e.g. LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Chung et al., 2014)) exploit internal memory mechanisms, enabling to forget useless aspects and to focus on important ones. The main limitation of these models for NLP is that semantically related words are often separated within a sentence and therefore memory cells can fail to connect the corresponding words. The Transformer architecture (Vaswani et al., 2017) tackles this problem. The key idea is to model the relationships between entities in a sequence regardless of their position, using an attention mechanism. Currently, a large majority of the state-of-the art methods in NLP are based on this architecture (e.g. ELMo (Peters et al., 2018) or BERT (Devlin et al., 2018)).

The Encoder of the Transformer architecture can be pre-trained on unsupervised tasks such as Masked Language Modeling (MLM) and/or Next Sentence Prediction (NSP) (Devlin et al., 2018). Then, a second step of fine-tuning is used to reach state-of-the-art performance on a wide variety of supervised tasks, from sentiment analysis to translation (Lample and Conneau, 2019). The pre-training step is generally performed on a large corpus of generic English texts, or large corpora from multiple languages (Pires et al., 2019). Recent works demonstrate that better results during the fine-tuning step can be obtained on a specific language if the model is pre-trained on a corpus from the same language (Martin et al., 2020).

2.4 Summary and discussion

Arabizi has emerged as the Arabic language in OSN, becoming of great interest for SA. The difficulty to process Arabizi with NLP techniques comes from the mix of languages, its inventive usage, the lack of rules and the variations between local dialects. As a result, Arabizi is a challenge for language modelling. To the best of our knowledge, currently there exist no Arabizi language model on the most recent architectures such as BERT, nor are there available and sufficiently large datasets to learn from scratch such an Arabizi model. We therefore propose to tackle these two concerns by (1) constituting a linguistic resource to train a BERT-like Arabizi language model, and (2) comparing the performance of our own model against a state-of-the-art text classifier on a sentiment analysis task. A corpus such as TUNIZI (Fourati et al., 2020) is insufficient to pre-train a language model but can be exploited for fine-tuning.

In this article, we consider that Arabizi represents only the dialects written in LC+N. We mainly focus on Egyptian dialect, as it is the most spoken Arabic dialect with more than 78 million speakers worldwide, to constitute an Arabizi resource to learn an Arabizi language model.

Although BERT pre-trained models are well documented and re-training is also frequent, for instance on tweets (Gertner et al., 2019; Müller et al., 2020), the literature remains unclear whether pre-trained multilingual resources can process dialects and assimilated languages. BERT exists in various versions applied on different languages. In particular, BERT-base-multilingual-uncased (BMU) is a language model learned on 102 languages, including English, French and Arabic languages. As Arabizi is a mix of these languages, we believe that this model should be the best available model to serve as baseline for our experiments, as well as a good candidate architecture to develop an Arabizi language model.

3 steps are necessary to solve a NLP task with a Transformer model: First, a vocabulary (token list) needs to be defined using for instance the Byte Pair Encoding algorithm (Sennrich et al., 2016). This step can be omitted by using an already existing vocabulary such as the BMU vocabulary, if this vocabulary can correctly represent the task domain. Secondly, the language model is pre-trained on generic unsupervised tasks like MLM or NSP (Devlin et al., 2018). These tasks can be started from scratch, continued from already pre-trained weights, or skipped by using any existing pre-trained weights such as the BMU weights. Finally, the language model is fine-tuned on the desired NLP. Concerning Arabizi, 4 options are therefore possible: 1) to build a new vocabulary, to pre-train from scratch the model and then to fine-tune it; this model is called **BAERT-arz-scratch**; 2) to use the BMU vocabulary, to pre-train from scratch the model and then to fine-tune it; this model is called **BAERT-bmu-scratch**; 3) to re-train from the

BMU model (therefore using the BMU vocabulary) and then fine-tune it; this model is called **BAERT-bmu-retrain**; 4) to fine-tune directly the **BMU** model, without any pre-training; this corresponds to the baseline BMU. BAERT stands for Bidirectional Arabizi Encoder Representations from Transformer. In this research, we hypothesise that options 1) and 3) should outperform options 2) and 4), depending of the capacity of a more generic vocabulary to represents the domain of the fine-tuning task.

3 LAD and SALAD: Arabizi resources

To create an open resource in Arabizi and to learn an Arabizi language model, we have collected Arabizi tweets using Egyptian keywords as seeds. The corpus is called **LAD** (Large Arabizi Dataset). Then, we manually annotated a subset of LAD according to their sentiment. This subset is named **SALAD** (Sentiment Annotations from LAD). LAD and SALAD are downloadable on demand[7].

3.1 Data collection and pre-processing (LAD)

A tweet collector named Twint[8] was exploited to construct the Arabizi dataset. A set of 48 common words in Egyptian, such as "*zehe2t, a7la, la2a, 3ayz*"[9], served as keyword seeds to collect tweets published between 2015 and 2019. These keywords were incrementally chosen so as to build a general Egyptian corpus in Arabizi: they are quite generic in meaning (words such as "but", " hello", "then", but also "sweet", "lost", 'I would"), while covering a wide range of word classes. All these words match the definition of Arabizi, i.e. they are written using LC+N. False positive inducing keywords (returning non-Arabizi tweets) were removed and we ensured that the tweets have a meaning in Arabic.

First attempts to create the keyword list resulted in data containing (i) Tweets written only in Arabic alphabet, when a username contains one of the keywords while some of posted texts are in Arabic, and (ii) Urdizi[10] tweets. Tweets in Urdu, as well as texts fully written in Arabic characters are excluded, whereas mix texts are included. Since tweets contain a lot of noisy data such as URLs, spams, images and so on, the collected data was filtered to remove a large part of this noise.

To protect privacy, collected data have been anonymized and all @mentions have been replaced by `NONAME`, after ensuring that it is not present in any tweet. Further on, the token `NONAME` is processed in order to have no impact on the semantics of the messages.

LAD (Large Arabizi Dataset) contains 7.7 million tweets.

3.2 Annotation according to sentiment (SALAD)

We randomly extracted and manually annotated 1,700 tweets from LAD, in order to create SALAD (Sentiment Annotations from LAD).

Classic sentiment annotation is focused on Positive versus Negative, sometimes also including Neutral class. As many social media posts from LAD are difficult to understand without context, we decided to use five classes: `Positive`, `Negative`, `Neutral`, `Conflict` and `ConflictTxtvsEm`. `Neutral` tweets do not express any feelings. `Conflict` class corresponds to tweets that contain both a negative and a positive sentiment. For example, the sentence *Msh 3arfa adhak wala a3ayat* which means in English "*I don't know if I laugh or I cry*" is annotated as Conflict. `ConflictTxtvsEm` represents tweets containing emojis in conflict with the emotion expressed by the text. This class can sometimes be interpreted as irony or sarcasm. For example, "*dlw2ty any z3lan* ☺" means "*Now I'm angry* ☺".

Table 1 provides information about the sentiment classes of SALAD: SALAD contains 50.5% of `Positive` tweets, 24.8% `Negative` tweets, 11.9% of `Neutral` tweets without any sentiment, 5% of tweets containing a `Conflict` of sentiment and 7.6% of tweets tagged as `ConflictTxtvsEm`. The proportions are imbalanced, which makes any supervised learning task more difficult. In comparison, SA datasets do not exhibit any particular distribution: for example, SemEval 2015 task 10 (Rosenthal et

[7] `http://saphirs.projets.litislab.fr/`

[8] `https://github.com/twintproject/twint`

[9] The list of all the keywords is available on the dataset description.

[10] Comes from Urdu-izi: some of the seed keywords are also used in Urdu. Urdu is among the official national languages of Pakistan and is transliterated similarly as Arabizi in social media.

Class	Tweets	Length	Emojis	Mentions	Hashtags
Positive	859	50.6	0.692	0.064	0.008
Negative	422	64.6	0.244	0.014	0.004
Neutral	203	44.4	0.167	0.030	0.010
Conflict	86	72.7	0.418	0.034	0.012
ConflictTxtvsEm	130	45.5	1	0.084	0.007

Table 1: Class distribution on SALAD. Length, emojis, mentions and hashtags are averaged per tweet.

al., 2015) contains 3 classes (Positive, Negative and Neutral) with an imbalanced distribution, whereas IberEval (Fersini et al., 2018) has only 2 classes (mysoginist or not) with a balanced distribution.

The average length of the tweets in SALAD is 54 characters (52 for LAD). On average, a tweet in SALAD contains 0.53 emojis (1.03 for LAD), less than 0.01 hashtag (0.06 for LAD) and 0.04 mentions (0.48 for LAD). The difference seems to be important for emojis and mentions, but mentions are anonymized and therefore do not provide much information. Over the different classes, emojis are the most variant, from 0.16 to 1. Actually, this is not surprising, as `ConflictTxtVsEm` requires an emoji, whereas `Neutral` does not express any sentiment and logically exhibits a weaker proportion of emojis. A more interesting aspect concerns the difference between `Negative` and `Positive`, where emojis seems to be more often used to express positive sentiments than negative ones on this dataset.

4 Experiments: Arabizi language models and sentiment classifiers

LAD is exploited as a pre-training dataset to construct several Arabizi language models which are then fine-tuned and compared on two different Arabizi SA tasks.

4.1 Language models: architecture

As detailed in Section 2.4, the multilingual version of BERT (Devlin et al., 2018), BERT-base-multilingual-uncased (BMU), serves as baseline for our experiments. The Huggingface implementation of BERT[11] is used since it obtains the exact same results as the original version (Wolf et al., 2019).

Three other language models are tested against BMU, all of them grounded on the BMU architecture. The first proposed model exploits the BMU vocabulary, i.e. the token list, as well as the BMU weights by retraining BMU on LAD (BAERT-bmu-retrain). The second model only uses the token list of BMU but is pre-trained from scratch on LAD (BAERT-bmu-scratch). Finally, the third model learns from scratch both an Arabizi-specific token list and the weights (BAERT-arz-scratch).

4.2 Language models: pre-training on LAD

BAERT-bmu-retrain, BAERT-bmu-scratch and BAERT-arz-scratch are pre-trained on LAD, varying according to vocabulary (token list) and to initial weights.

Token list construction

BAERT-bmu-retrain and BAERT-bmu-scratch exploit the BMU token vocabulary to tokenize any text as input of the models. Possible modifications of the vocabulary consist in adding specific tokens, as BMU contains some unused token slots. We therefore only add a NONAME token to encode the @mentions.

On the contrary, BAERT-arz-scratch has his own Arabizi vocabulary. We create a specific token list, called Arz-vocabulary, thanks to Byte Pair Encoding (Sennrich et al., 2016) applied on LAD and more specifically the SentencePiece implementation [12]. Arz-vocabulary encodes Egyptian Arabizi.

Mask language modeling as pre-training task

BERT-like language models can be pre-trained thanks to two different tasks: MLM and/or NSP. NSP consists in learning if two segments of text come from the same text or not. However, several experiments have highlighted better or similar results without NSP (Liu et al., 2019; Joshi et al., 2019; Lan et al., 2019). We thus decide to exclude NSP from our pre-training step and to only focus on MLM.

[11] `https://huggingface.co/transformers/index.html`
[12] `https://github.com/google/sentencepiece`

MLM usually obtains good performance as pre-training step for Transformer model, in order to solve NLP tasks, in particular with long sequences of text (Devlin et al., 2018). As tweets are short texts, often consisting of a single sentence, we concatenate tweets up to the model size (512 tokens): it enables to accelerate the training (padding is replaced by data and more data is included in each batch).

Classic BERT parameters for pre-training are used (Devlin et al., 2018): a learning rate of 10^{-4}, a L2 weight decay of 0.01. For computational reasons, a batch size of 64 replace the usual 256 one. After some epochs, the learning rate is divided by 10, as the number of epochs depends on the model convergence. we observed that BAERT-bmu-retrain and BAERT-bmu-scratch converges much faster than BAERT-arz-scratch. We also adopt the dynamic masking of RoBERTa (Liu et al., 2019), as it has demonstrated some improvements compared to the static masking of BERT.

1% of the training set (~77,800 tweets) serves as validation set to check the convergence of the models.

4.3 Fine-tuning for sentiment analysis on Arabizi

The four language models are fine-tuned to be evaluated on two Arabizi sentiment-oriented datasets: SALAD and TUNIZI dataset (Fourati et al., 2020). TUNIZI contains 22 of the 100 most frequent words in common with LAD.

On SALAD, 3 different classifiers are proposed: a first classifier using the initial 5 classes, a second one with only 4 classes by merging the two conflict classes and a last one with only 3 classes (positive, negative and other). We decide to test multiple classifiers on SALAD in order to manage the imbalance characteristic of the dataset. The TUNIZI dataset is annotated into 2 classes, positive and negative.

As recommended in (Devlin et al., 2018), the following parameters are used during the fine-tuning step: a learning rate of 3.10^{-5} with a linear scheduler and a warm-up of 1 epoch, a L2 weight decay of 0.01 and a batch size of 32. The test protocol consists in a 10 stratified shuffle split of each dataset, with 90% of the datasets in the training set and 10% in the test set for each split. Every model is fine-tuned on each split during 5 epochs, slightly more than recommended in (Devlin et al., 2018) to ensure complete process. F1-score averages and standard deviations are then computed concerning the best of each training session (models are tested every half epoch), for each model on each task.

5 Results and Discussion

We first compare BMU-vocabulary with Arz-vocabulary, then the results of the fine-tuning experiments on the two Arabizi SA tasks are provided.

5.1 Comparison between BMU and Arz vocabularies

Some metrics can help to highlight the importance of a vocabulary compared to another one. Actually, changing the token list in a Transformer model breaks any pre-training, except for minor changes such as adding a specific token.

The first metric proposed consists in counting the number of common tokens between two vocabularies. The BMU vocabulary contains 105,879 tokens. Arz vocabulary contains 32,002 tokens, which is a standard value for a single language model (e.g. the English BERT uncased vocabulary contains almost 30,500 tokens). On top of that, the Byte Pair Encoding algorithm used to build the Arz-vocabulary sorts the tokens by occurrence: Figure 1 enables to explore if the common tokens are also the most frequent ones or not. Between Arz and BMU, this is not the case (among the 5,000 most frequent tokens, less than 50% are common tokens). On the 32,002 tokens from Arz-vocabulary, 9,267 tokens are common with the BMU vocabulary: a coverage rate of 28.9% (over the Arz-vocabulary). In other words, more than 70% of the Arz-vocabulary tokens are not in the BMU vocabulary.

We then propose a second metric which consists in counting the number of tokens needed to tokenize a dataset and comparing it to the mean number of tokens per word. Results are presented on Table 2. The difference between both vocabularies is quite important on LAD and on the TUNIZI dataset: almost 18% more tokens are necessary for both datasets with the BMU vocabulary than with the Arz-vocabulary, while the number of tokens in the vocabulary is divided by almost 3.3.

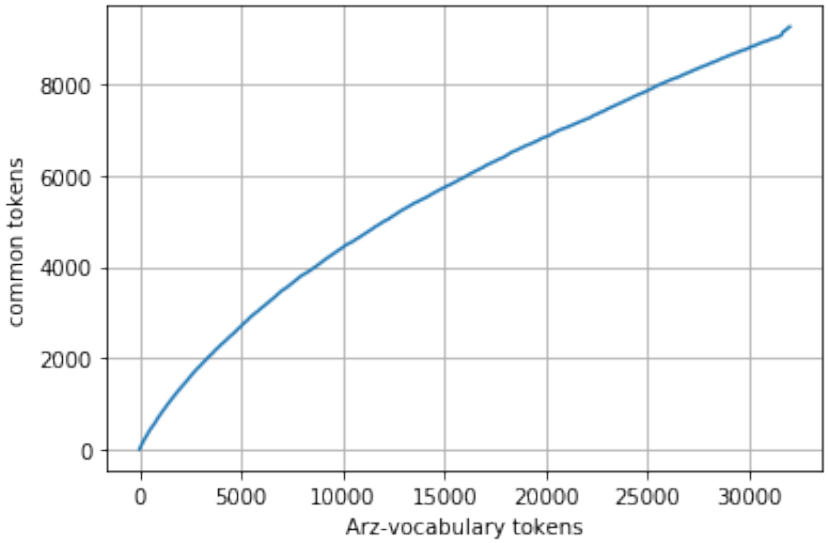

Figure 1: Common tokens between Arz-vocabulary and BMU vocabulary. The Arz-vocabulary tokens are sorted by decreasing occurrence.

Finally, the number of `[UNK]` tokens over all the tokens can be computed: `[UNK]` tokens correspond to characters that are not present in the vocabulary. Results are presented on Table 3. More `[UNK]` tokens appear on LAD when using the BMU vocabulary compared to using Arz-Vocabulary. In other words, much missing information appear with BMU on LAD. On the other hand, the contrary can be observed on the TUNIZI dataset: the BMU vocabulary can represent more information than the Arz-vocabulary, but the proportion of `[UNK]` tokens for Arz-vocabulary is not high, as it is the similar to the proportion on LAD.

Dataset	Characters	BMU vocabulary	Arz-vocabulary
LAD	5.25	1.97	1.67
TUNIZI	5.84	2.19	1.86

Table 2: Tokens per word depending of the vocabulary used. The *Characters* column corresponds to the average number of characters per word in the dataset.

Dataset	BMU `[UNK]` frequence	Arz-vocabulary `[UNK]` frequence
LAD	3.1%	0.6%
TUNIZI	0.01%	0.6%

Table 3: Frequence of `[UNK]` token over all tokens needed to represent the corpora.

5.2 Results of fine-tuning

Table 4 presents the results of classification protocols as described in Section 4.3.

The best model on SALAD is BAERT-arz-scratch, and BAERT-bmu-retrain on the TUNIZI dataset. This can be explained by the nature of both datasets: SALAD contains mostly Egyptian Arabizi which is often mixed with English, and TUNIZI contains Tunisian Arabizi which is mostly mixed with French. The BMU model is pretrained on 102 languages, including French and English. BAERT-bmu-retrain has already seen French sentences which can help for TUNIZI dataset; on the contrary, BAERT-arz-scratch only encountered Arabizi mixed with English. However, both models obtain close results on this task: the difference between them is lower than their Standard Deviation.

Comparing BAERT-bmu-scratch and BAERT-bmu-retrain corresponds to an ablation study. The lower performance of BAERT-arz-scratch on TUNIZI is probably due to the token list, as there is less difference

between BAERT-bmu-scratch and BAERT-bmu-retrain than between BAERT-bmu-scratch and BAERT-arz-scratch. Logically, both vocabularies are not fitting the dataset perfectly, but the BMU token list is more generic than the Arz-vocabulary (it can handle better french words, for example, and generates less `[UNK]` tokens on TUNIZI, as shown on Table 3).

On the 5 classes task, an analysis of the confusion matrix shows that the main challenges concern the `Conflict` class and the `ConflictTxtvsEm` class. Only BAERT-arz-scratch obtains a decent result on the last one (average F1 of the class of 70% for BAERT-arz-scratch, 45% for BAERT-bmu-retrain) and all the models failed to handle the `Conflict` class properly (average F1 is no more than 5% for all the models). This is probably due to the data imbalance (the `Conflict` class represents only 5% of the total dataset) and the difficulty of the task, and explains the large improvements when we merge these classes together or with the `Neutral` class.

Model \Dataset	**SALAD 5 classes**	σ	**4 classes**	σ	**3 classes**	σ	**TUNIZI**	σ
BMU	41.6	4.0	51.4	3.7	58.9	3.0	78.8	2.4
BAERT-bmu-retrain	52.3	2.0	63.6	1.9	68.2	2.7	**83.8**	1.1
BAERT-bmu-scratch	52.0	3.9	62.5	2.8	67.4	2.8	83.6	1.6
BAERT-arz-scratch	**59.9**	2.8	**71.2**	2.4	**74.4**	3.3	82.7	2.6

Table 4: F1-Score averages and Standard deviation (in percentage) of the protocol described in Section 4.3. Best results for each task appear in bold.

To summarize: 1) exploiting the BMU pre-trained weights is almost ineffective concerning performance (even if the convergence is 2x faster, as we needed 200k batches of 64 for BAERT-bmu-retrain, and 400k for BAERT-bmu-scratch) and 2) adapting the vocabulary to Arabizi improves the performance.

6 Conclusion and future work

Arabizi is a challenge for computational linguistic: there exist only few data resources, no official grammar nor orthography. Moreover, the language is constantly evolving through time and space, as each country has its specific dialect. Language models such as ELMo (Peters et al., 2018) or BERT (Devlin et al., 2018) get state-of-the-art results on most of the NLP tasks. Their pre-trained models, in English or multilingual versions, obtain excellent performance on a lot of use-cases, but do not perform correctly on Arabizi, due to the lack of any large corpus in such language to carry out a pre-training step.

Our contribution in this article is twofold: Firstly, we provided LAD, a Large Arabizi Dataset constituted of tweets. From this dataset, we made SALAD, a smaller manually annotated Arabizi dataset for Sentiment Analysis, taking into account sentiment conflicts in the text or between the text and the emojis. Secondly, we trained BAERT, a BERT-like language model specific to Arabizi and obtained good results on Tunizi and Egyptizi-like SA task, showing that a pre-training on a local specific Arabizi dialect can help to get better results on this specific dialect but also on other ones. A pre-training from scratch with a new vocabulary seems preferable for a new language, even for a language that is a mix of different ones (Arabizi is a mix of English, French and Arabic using Latin alphabet).

In the future, we plan to exploit LAD to train a model on different NLP tasks such as translation with English, or transliteration with MSA. Also, the metrics on the efficiency of the tokenizer seem an interesting lead to follow to evaluate the interest between pre-training from scratch a Transformer model or exploiting pre-trained weights from an existing model. In particular, the frequence of unkown tokens seems to be promising, and need to be further investigated, with other Arabizi corpora from different local dialects for example. Pre-training a Transformer model from scratch is computationally expensive comparing to start with already pre-trained weights, and finding metrics that can help to make this choice without any excessive computational cost could be really useful.

Acknowledgements

Part of this work was performed using the computing resources of CRIANN (Normandy, France) and within the SAPhIRS project[13].

[13] `http://saphirs.projets.litislab.fr/`

References

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12.

Wid H. Allehaiby. 2013. Arabizi: An analysis of the Romanization of the Arabic Script from a Sociolinguistic Perspective. *Arab World English Journal (AWEJ)*, 4(3):52–62.

Ann Bies, Zhiyi Song, Mohamed Maamouri, Stephen Grimes, Haejoong Lee, Jonathan Wright, Stephanie Strassel, Nizar Habash, Ramy Eskander, and Owen Rambow. 2014. Transliteration of arabizi into arabic orthography: Developing a parallel annotated arabizi-arabic script sms/chat corpus. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 93–103.

Cagatay Catal and Mehmet Nangir. 2017. A sentiment classification model based on multiple classifiers. *Applied Soft Computing*, 50:135–141.

J. Chung, C. Gulcehre, K.-H. Cho, and Y. Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Ryan Cotterell and Chris Callison-Burch. 2014. A multi-dialect, multi-genre corpus of informal written Arabic. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Ryan Cotterell, Adithya Renduchintala, Naomi Saphra, and Chris Callison-Burch. 2014. An algerian arabic-french code-switched corpus. In *LREC Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools*.

Kareem Darwish. 2014. Arabizi Detection and Conversion to Arabic. *ANLP 2014*, page 217.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

R. M. Duwairi, M. Alfaqeh, M. Wardat, and A. Alrabadi. 2016. Sentiment analysis for Arabizi text. In *2016 7th International Conference on Information and Communication Systems (ICICS)*, pages 127–132.

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. *EMNLP 2014*, page 1.

Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. In *IberEval@SEPLN*.

Chayma Fourati, Abir Messaoudi, and Hatem Haddad. 2020. TUNIZI: a Tunisian Arabizi sentiment analysis Dataset. *CoRR*, abs/2004.14303.

Guillaume Gadek, Josefin Betsholtz, Alexandre Pauchet, Stephan Brunessaux, Nicolas Malandain, and Laurent Vercouter. 2017. Extracting contextonyms from twitter for stance detection. In *Proceedings of ICAART17*, pages 132–141.

Abigail S Gertner, John Henderson, Elizabeth Merkhofer, Amy Marsh, Ben Wellner, and Guido Zarrella. 2019. Mitre at semeval-2019 task 5: Transfer learning for multilingual hate speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 453–459.

Maria Giatsoglou, Manolis G Vozalis, Konstantinos Diamantaras, Athena Vakali, George Sarigiannidis, and Konstantinos Ch Chatzisavvas. 2017. Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications*, 69:214–224.

Yves Gonzalez-Quijano. 2014. Technology literacies of the new media: Phrasing the world in the "arab easy" (r)evolution. In Leila Hudson, Adel Iskandar, and Mimi Kirk, editors, *Media Evolution on the Eve of the Arab Spring*, pages 159–166. Palgrave Macmillan, New York.

Imane Guellil, Ahsan Adeel, Faical Azouaou, Fodil Benali, Ala-eddine Hachani, and Amir Hussain. 2018. Arabizi sentiment analysis based on transliteration and automatic corpus annotation. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 335–341, Brussels, Belgium, October. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv:1607.01759 [cs]*, August. arXiv: 1607.01759.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July. Association for Computational Linguistics.

Diana Maynard and Adam Funk. 2011. Automatic detection of political opinions in tweets. In *The semantic web: ESWC 2011 workshops*, pages 88–99. Springer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, September. arXiv: 1301.3781.

Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. ASTD: Arabic Sentiment Tweets Dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519, Lisbon, Portugal, sep. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv:1802.05365 [cs]*, February. arXiv: 1802.05365.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4996–5001. Association for Computational Linguistics.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, and others. 1995. Okapi at TREC-3. *Nist Special Publication Sp*, 109:109.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 451–463, Denver, Colorado, June. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Taha Tobaili, Miriam Fernandez, Harith Alani, Sanaa Sharafeddine, Hazem Hajj, and Goran Glavaš. 2019. SenZi: A sentiment analysis lexicon for the latinised Arabic (Arabizi). In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1203–1211, Varna, Bulgaria, September. INCOMA Ltd.

Taha Tobaili. 2016. Arabizi Identification in Twitter Data. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 51–57, Berlin, Germany, aug. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.