Building a User-Generated Content North-African Arabizi Treebank: Tackling Hell

Djamé Seddah¹  Farah Essaidi¹  Amal Fethi¹
Matthieu Futral¹  Benjamin Muller¹,²  Pedro Javier Ortiz Suárez¹,²
Benoît Sagot¹  Abhishek Srivastava¹
¹Inria, Paris, France
²Sorbonne Université, Paris, France
firstname.lastname@inria.fr

Abstract

We introduce the first treebank for a romanized user-generated content variety of Algerian, a North-African Arabic dialect known for its frequent usage of code-switching. Made of 1500 sentences, fully annotated in morpho-syntax and Universal Dependency syntax, with full translation at both the word and the sentence levels, this treebank is made freely available. It is supplemented with 50k unlabeled sentences collected from Common Crawl and web-crawled data using intensive data-mining techniques. Preliminary experiments demonstrate its usefulness for POS tagging and dependency parsing. We believe that what we present in this paper is useful beyond the low-resource language community. This is the first time that enough unlabeled and annotated data is provided for an emerging user-generated content dialectal language with rich morphology and code switching, making it an challenging testbed for most recent NLP approaches.

1 Introduction

Until the rise of fully unsupervised techniques that would free our field from its addiction to annotated data, the question of building useful data sets for under-resourced languages at a reasonable cost is still crucial. Whether the lack of labeled data originates from being a minority language status, its almost oral-only nature or simply its programmed political disappearance, geopolitical events are a factor highlighting a language deficiency in terms of natural language processing resources that can have an important societal impact. Whether the lack of labeled data originates from being a minority language status, its almost oral-only nature or simply its programmed political disappearance, geopolitical events are a factor highlighting a language deficiency in terms of natural language processing resources that can have an important societal impact. Events such as the Haïti crisis in 2010 (Munro, 2010) and the current Algerian revolts (Nossiter, 2019)¹ are massively reflected on social media, yet often in languages or dialects that are poorly resourced, namely Haitian Creole and Algerian dialectal Arabic in these cases. No readily available parsing and machine translations systems are available for such languages. Taking as an example the Arabic dialects spoken in North-Africa, mostly from Morocco to Tunisia, sometimes called Maghribi, sometimes Darija, these idioms notoriously contain various degrees of code-switching with languages of former colonial powers such as French, Spanish, and, to a much lesser extent, Italian, depending on the area of usage (Habash, 2010; Cotterell et al., 2014; Saadane and Habash, 2015). They share Modern Standard Arabic (MSA) as their matrix language (Myers-Scotton, 1993), and of course present a rich morphology. In conjunction with the resource scarcity issue, the code-switching variability displayed by these languages challenges most standard NLP pipelines, if not all. What makes these dialects especially interesting is their widespread use in user-generated content found on social media platforms, where they are generally written using a romanized version of the Arabic script, called Arabizi, which is neither standardized nor formalized. The absence of standardization for this script adds another layer of variation in addition to well-known user generated content idiosyncrasies, making the processing of this kind of text an even more challenging task.

In this work, we present a new data set of about 1500 sentences randomly sampled from the romanized Algerian dialectal Arabic corpus of Cotterell et al. (2014) and from a small corpus of lyrics coming from Algerian dialectal Arabic Hip-Hop and Raï music genre that had the advantage of having already available translations and of being representative of Algerian vernacular urban youth language. We manually annotated this data set with morpho-syntactic information (parts-of-speech and morphological features), together with glosses and code-switching labels at the word

¹https://www.nytimes.com/2019/03/01/world/africa/algeria-protests-bouteflika.html
level, as well as sentence-level translations. Furthermore, we added an additional manual annotation layer following the Universal Dependencies annotation scheme (Nivre et al., 2018), making of this corpus, to the best of our knowledge, the first user-generated content treebank in romanized dialectal Arabic. This treebank contains 36% of French tokens, making it a valuable resource to measure and study the impact of code-switching on NLP tools. We supplement this annotated corpus with about 50k unlabeled sentences extracted from both Common Crawl and additional web crawled data, making of this data set an important milestone in North-African dialectal Arabic NLP. This corpus is made freely available under a Creative Commons license.

2 The Language

As stated by Habash (2010), Arabic languages are often classified into three categories: (i) Classical Arabic, as found in the Qur’an and related canonical texts, (ii) Modern Standard Arabic, the official language of the vast majority of Arabic speaking countries and (iii) Dialectal Arabic, whose instances exhibit so much variations that they are not mutually understandable across geographically distant regions. As space is missing for an exhaustive description of Arabic language variations, we refer the reader to Habash (2010), Samih (2017) and especially to Saadane and Habash (2015) for a thorough account of Algerian dialectal Arabic, which is the focus of this work. In short, the key properties of North-African dialectal Arabic are:

- It is a Semitic language, non codified, mostly spoken;
- It has a rich-inflexion system, which qualifies this dialect as a morphologically-rich language (Tsarfaty et al., 2010), even though Saadane and Habash (2015) write that many properties present in Classical Arabic are absent from this dialect (e.g. it has simplified nominal and verbal case systems);
- It displays a high degree of variability at all levels: spelling and transliteration conventions, phonology, morphology, lexicon;
- It exhibits a high degree of code-switching; due to historical reasons and cultural influence of French in the media circles, the Algerian dialect, as well as Tunisian and Moroccan, is known for its heavy use of French words.

| Gloss | Attested forms | Lang |
|-------|---------------|------|
| why   | wa3lach w3alh 3alach 3lache | NArabizi |
| all   | ekal kal kolach kolli kol | NArabizi |
| many  | beaucoup boucoup bcp | French |

Table 1: Examples of lexical variation in NArabizi

As stated above, this dialect is mostly spoken and has even been dubbed with disdain as a Creole language by the higher levels of the Algerian political hierarchy. Still, its usage is ubiquitous in the society and, by extension, in social media user-generated content. Interestingly, the lack of Arabic support in input devices led to the rise of a romanized written form of this dialect, which makes use of alphanumeric letters as additional graphemes to represent phonemes that the Latin script does not naturally cover. Not limited to North-African dialectal Arabic, this non-standard “transliteration” concurrently emerged all over the Arabic-speaking world, and is often called Arabizi. Whether or not written in Arabizi, the inter-dialectal divergences between all Arabic dialects remain.

The following list highlights some of the main properties of Arabizi compared to MSA written in the Arabic script.

- Unlike in MSA written in the Arabic script, where short vowels are marked using optional diacritics, all vowels are explicitly written;
- Digits are used to cope with Arabic phonemes that have no counterpart in the Latin script; for instance, the digit “3” is often used to denote the ayin consonant, because it is graphically similar to its rendition in Arabic script;
- No norms exist, resulting in a high degree of variability between people writing in Arabizi.

From now on, we will call NArabizi the Algerian dialect of Arabic when written in Arabizi, thereby simultaneously referring to the language variety and to the script itself. Table 1 presents several examples of lexical variation within NArabizi. Interestingly, this variability also affects the code-switched vocabulary, which is mostly French in the case of NArabizi. A typical example of NArabizi that also exhibits code-switching with non-standard French spelling can be seen in Example 1.

(1)

Source: salem 3alikoum inchalah le pondium et les midailes d’or

<https://www.lesoirdalgerie.com/articles/2010/02/17/article.php?sid=95823&cid=2>
Norm.: Assalamu alaykum inshallah le podium et les médailles d’or
Trans.: Peace be on you God willing [we will get] the podium and the gold medals

3 Corpus

As other North-African Arabic dialects, \textit{NArabizi} is a resource-poor language, with, to the best of our knowledge, only one available corpus developed by Cotterell et al. (2014) for language identification purposes.

3.1 Data Collection

Cotterell et al. (2014)’s corpus was collected in 2012 from an Algerian newspaper’s web forums and covers a wide range of topics (from discussion about football events to politics). We collected the 9973 raw sentences from its GitHub repository\(^4\) and sampled about 1300 sentences. In addition, because they were available with translations in French and English, we included lyrics from a few dozen recent popular songs of various genres (Raï, hip-hop, etc.), leading to an additional set of 200 sentences. These 1500 sentences form the core of our \textit{NArabizi} treebank annotation project.

In order to make our corpus usable by modern, resource-hungry natural language processing techniques, we also used data-driven language identification models to extract \textit{NArabizi} samples among the whole collection of the Common-Crawl-based OSCAR corpora (Ortiz Suárez et al., 2019) as well as 2 millions sentences of additional crawled web-data, resulting in 50k \textit{NArabizi} sentences of high quality, to date the largest corpus of this language. This makes this collection a valuable test bed for low-resource NLP research.

3.2 Annotation Layers

Our \textit{NArabizi} treebank contains 5 annotations layers: (i) tokenization, (ii) morphology, (iii) code-switching identification, (iv) syntax and (v) translation.

Tokenization Following Seddah et al. (2012) and their work on the French Social Media Bank, we decided to apply a light tokenization process where we manually tokenized only the obvious cases of wrongly detached punctuations and “missing whitespaces” (\textit{i.e.} cases where two words are contracted into one token).\(^5\)

Morphological Analysis This layer consists of two sets of part-of-speech tags, one following the Universal POS tagset (Petrov et al., 2011) and the other the FTB-cc tagset extended to deal with user-generated content (Seddah et al., 2012). In cases of word contractions, we followed their guidelines and used multiple POS as in \textit{cetait} (‘itwas’)\,/\textit{PRON+VERB/CLS+V}. In addition, we added several morphological features following the Universal Dependency annotation scheme (Nivre et al., 2018), namely gender, number, tense and verbal mood. Note that instead of adding lemmas, we included French glosses for two reasons: firstly for practical reasons, as they helped manual corrections done by non-native speakers of \textit{NArabizi}, and secondly because of the non-formalized nature of this language, which makes lemmatization very hard, almost akin to etymological research as in the case of \textit{garjouma/the throat} which can either originate from French \\textit{gorge} or be of Amazigh root.

Code-Switching identification Unlike other works in user-generated content for minority languages (Lynn and Scannell, 2019), we do not distinguish between inter- and intra-sentential code-switching and consider word-level code-mixing as lexical borrowing. We annotate code-switching at the word level with information about the source language, regardless of the canonical-ness of spelling.

Syntactic Annotations Here again we follow the Universal Dependencies 2.2 annotation scheme (Nivre et al., 2018). When facing sequences of French words with regular French syntax, we followed the UD French guidelines; otherwise, we followed the UD Arabic guidelines, following the Prague Arabic Dependency UD Treebank.

Translation Layer Our final layer is made up for sentence-level translations in French. It shall be noted that the validation of these translations often led to massive rewording, as the annotators came from different regions of Algeria and could diverge in their interpretations of a given sentence.

\(^{4}\)\url{https://github.com/ryancotterell/arabic_dialect_annotation}

\(^{5}\)We corrected in average one tokenization error (less frequently two) per sentence on the web forum parts. We noticed a high degree of variance. Some users displayed this behavior much more than others. This led some of our annotators to believe it resulted from an ill-functioning input device.
A sample of 200 sentences was blindly translated (without access to the morpho-syntactic analysis) in order to favor further research on the fluency of machine translation for this dialect.

All annotations layers are displayed in Figure 1.

4 Extending Our Data Set With Noisy Unlabeled Data

The need for more data has never been more striking as they are needed for important tasks such as handling lexical sparseness issues via word embeddings, lexicon acquisition, domain adaptation via self-training, or fine-tuning pre-trained language models, its modern incarnation. The trouble with NArabizi is that it is a spoken language whose presence can be mostly found in informal texts such as social media. More importantly, the Arabizi transliteration process is also used by other Arabic dialects, making the data collection a needle in a haystack search task. We therefore present in this section the process we used to mine an additional set of 50k NArabizi sentences from two large corpora, one based on search query-based web-crawling and the other from a cleaned version of the CommonCrawl corpora, developed by Ortiz Suárez et al. (2019).

4.1 First method: SVM-based classifier

Using keywords-based web scrapping tools, we collected a raw corpus of 4 million sentences, called CrawlWeb, that in fine contained a mixture of French, English, Spanish, MSA and Arabizi texts. Since we are only interested in NArabizi, we designed a classifier to extract proper sentences from that raw corpus. The corpus we used as gold standard is made of 9k sentences of attested NArabizi from our original corpus and 18k of French and English tweets. Using language identification (Lui and Baldwin, 2012), we convert each sentence from the gold-standard corpus to a feature vector containing language-identification scores and use it as input to a SVM classifier with a classical 80/10/10 split. With a precision and recall score of 94%, we filtered out 173k code-mixed sentences out of the CrawlWeb corpus. Preliminary experiments showed promising initial results, but further analysis pointed out a high level of noise in this initial set, both in terms of erroneous language identification and on the amount of remnant ASCII artifacts that could not easily be removed without impacting the valid NArabizi sentences.

4.2 Second method: Neural-based classification

The objectives of this method are twofold: (i) selecting data from CommonCrawl using a neural classifier and (ii) using this data set to intersect the data collected with the previous method. The idea is to ensure the quality of the final resulting unlabeled corpus.

Given the large number of noisy data in CommonCrawl, a “noise” class is added to the language classification model and is built according to several heuristics. That “noisy” class corpus is made of 40k sentences randomly selected among the result of the application of these rules to a short, 10M-sentence sample of CommonCrawl. We then trained a classifier using Fasttext (Joulin et al., 2016) on 102 languages, 40k sentences each, extracted from the CommonCrawl-based, language-classified OSCAR corpus, to which we added the 9k sentences of the NArabizi original corpus and

6These heuristics are presented in the Appendix for reproducibility.
the “noise” class. The final dataset is composed of 4,090,432 sentences and is split into 80% train, 10% development and 10% test sets. The classifier consists in a linear classifier (here logistic regression) fed with the average of the n-gram embeddings. n-grams are useful in this case as they enables the model to capture specific sequences of NaArabizi characters such as 1ах, 11аn, 3а, 9а, etc. We choose to embed 2- to 5-grams. These parameters lead to precision and recall scores of 97% on the NaArabizi test set.

After an intensive post-processing step (cf. Appendix A.2), this process results in a dataset of 13,667 sentences extracted from half the CommonCrawl corpus. To evaluate the quality of the resulting data set, we randomly picked 3 times 100 sentences, and genuine NaArabizi sentences were manually identified, which allowed us to assess the accuracy of our corpus as reaching 97%. Table 2 presents the results of the evaluation of the two classification methods performed on both the development and test sets of the original NaArabizi corpus. Results show that the fastText classifier and its n-gram features is more precise than its non-neural counterpart and its language-id feature vectors.

4.3 Corpus intersection

When applied to the CrawlWeb corpus, the Fasttext model extracted 44,797 unique Arabizi sentences while the SVM model extracted 83,295 unique Arabizi sentences. The intersection of both extractions amounts to 39,003 Arabizi sentences (with a 99% precision). This means that 44,292 sentences were classified as Arabizi by the SVM model and not by Fasttext. Among them, by random sampling, it can be stated approximately that 55% are indeed NaArabizi. Mistakes are misclassified sentences (Spanish and English sentences, for instance) or sentences with only “noise” (such as symbols). 5,794 sentences were classified as NaArabizi by the Fasttext model and not SVM. Among them, by random sampling, it can be stated that approximately 60% are indeed Arabizi. Errors are long sentences with only figures and numbers or sentences with many symbols (e.g. “{ O3 }” or “!!! !!!!”).

In order to ensure that the collected corpus contains as little non-NArabizi data as possible, we only release the intersection of the data we classified, to which we add the original NaArabizi corpus (Cotterell et al., 2014) (after having removed the annotated data we extracted from it). Table 3 provides quantitative information about our corpora.

5 Pre-annotation Tool Development via Noisy Transliteration of an Arabic UD Treebank

In order to speed up the annotation process of our data, we decided to create a pre-annotation morphosyntactic and syntactic annotator trained on quasi-synthetic data obtained by “transliterating” a pre-existing Arabic (MSA) treebank, the Prague Arabic Dependency Treebank (PADT), into the NaArabizi Latin script, together with data from the French GSD UD treebank. Both are taken from the UD treebank collection (Nivre et al., 2018).

Before it can be used as training data, the PADT needs to first be transformed into a form similar to NaArabizi. Since the PADT corpus is a collection of MSA sentences with no diacritics, it is impossible to directly “transliterate” into NaArabizi. We first diacritized it, in order to add short-vowel information, and then “translitterated” it into an Arabizi-like corpus. We describe this process in this Section. The results of the pseudo-NArabizi parser trained on the “translitterated” corpus are then presented in Section 6.2.

Random diacritics As vowels are always written in Arabizi, the PADT corpus needs to be diacritized before transliteration. Using an equiprobable distribution, diacritics were added randomly, and the text then transliterated using the probabil-

| Dataset            | #Sentences | #Tokens |
|--------------------|------------|---------|
| Original source    | 9,372      | 203k    |
| Manually Annotated | 1,434      | 22k     |
| Unlabeled NaArabizi| 46,941     | 1,02M   |

Table 3: Corpus statistics

| F1-Score Arabizi |
|------------------|
| Fasttext         | 0.97      |
| SVM Classifier   | 0.94      |

Table 2: F1-scores of both language classification models on the Arabizi class.

---

7Due to computing power limitation, we were not able to run our selection on the whole CommonCrawl.

8Note that the precision and recall are slightly different in both methods, but the rounding at the second decimal made them equal.
ity distributions we describe below. The BLEU score (Papineni et al., 2002) of this method on the small parallel corpus provides a baseline of 0.31.

**Proper diacritization** Using the Farasa software (Abdelali et al., 2016), PADT sentences are diacritized with 81% precision rate, then tokens aligned with corresponding diacritized words. The text is then transliterated the same way as before. The BLEU score of this version is 0.60. An example showing how this system visibly improves the transliteration can be seen in the “Prop. Diac.” output in Example 2.

(2)

Source: berlin tarfoudhou 7oussoul charika amrikia 3ala ro5sat tasni3i dabbabat ”léopard” al almania  
Trans.: Berlin refuses to authorize an American firm to produce the ”Leopard” German tank.

Random diac.: brouliyani trfidh 7iswla chiroukou amiyirikyoi 3alia rou5soui tasaniy3i dhabouaboi lousyiwibiaridha alalmaanouyou  
Proper diac.: birlin tarfoudhou 7ousolou charikatin 2amiriqiatin 3alaa rou5sati tasni3i dabbatin ” lyuberid ” el2almeniati

| System                     | BLEU score |
|----------------------------|------------|
| Random diacritization      | 0.31       |
| Proper diacritization      | 0.60       |

Table 4: BLEU score of both transliteration systems.

**Transliteration** Once diacritized, the corpus can be properly transliterated. Arabic letters are either consonant sounds or long vowels, each one may have several different transliterations in NArabizi, depending on the writer’s age, accent, education and first learned Western language. For example, the letter ﺛ10 can be transliterated as “t” or “th”. A probability must be assigned for each possibility, and to make it as close as possible to what is produced by NArabizi speakers, a small parallel corpus of PADT sentences and their transliteration by ten NArabizi speakers was assembled, and then each letter aligned with all its possible matches to get probability distributions.

6 Usability

In this section we describe preliminary experiments on part-of-speech tagging and statistical dependency parsing that show promising results while highlighting the expected difficulty of processing a low-resource language with a high level of code-switching and multiple sources of variability.

6.1 POS Tagging

The baseline POS tagger we used is alVWTagger,11 a feature-based statistical POS tagger, which ranked 3rd at the 2017 CoNLL multilingual parsing shared task (Zeman et al., 2017). It is briefly described in (de La Clergerie et al., 2017). In short, it is a left-to-right tagger that relies on a set of carefully manually designed features, including features extracted from an external lexicon, when available, and a linear model trained using the Vowpal Wabbit framework.12 In our case, we simply created an "external" lexicon by extracting the content of the training set. It contributes to improving the POS accuracy because it provides the tagger with (ambiguous, partial) additional information about words in the right context of the current word.13

|                        | Dev All | OOV | Test All | OOV |
|------------------------|---------|-----|----------|-----|
|                        |         |     |          |     |
| OOV %                  | 32.28   | 32.75 |
| UPOS (a)               | 78.74   | 55.85 |
| MFEATS (b)             | 88.10   | 70.04 |
| (a)+(b)                | 72.66   | 40.94 |
|                        | 73.87   | 43.50 |

Table 5: POS tagging results.

6.2 Early Parsing experiments

As stated earlier in this paper, NArabizi contains a high-level of code-switching with French and is closely related to MSA. We described in Section 5 how we built a mixed treebank based on the

10Note that we performed a set of baseline experiments with UDPipe 2.0 (Straka and Straková, 2017) as well on a previous version of this data set. It reached only 73.7 of UPOS on the test set.
11https://github.com/VowpalWabbit/vowpal_wabbit/wiki
12Without this endogenous lexicon extraction step, the tagger performed slightly worse, although the difference is small.
French GSD UD treebank and our Arabizi version of the Prague Arabic Dependency Treebank. We trained the UDPipe parser (Straka and Straková, 2017) on various treebanks obtained by combining different proportions of the French GSD and our PADT-based pseudo-Arabizi treebank. We ran these parsers with already annotated gold parts-of-speech. The best scores were obtained with a model trained on a mix 30% of pseudo-Arabizi and 70% of French, which we call the MIX treebank, totaling 5,955 training sentences. We split this treebank into training, development and test sets, called MIX_train/dev/test, following a 80/10/10 split. We used a very small manually annotated NArabizi development dataset of 200 NArabizi sentences, called Arabizi_dev, to evaluate our parser. As shown in Table 6 (line “Mix”), despite good results on MIX’s development and training sets, MIX_dev and MIX_test respectively (see Table 6), this first parser did not perform very well when evaluated on Arabizi_dev. This performance level proved insufficient to speed up the annotation task. We therefore manually annotated 300 more NArabizi sentences (Arabizi_train300), to be used as additional training data. When added to MIX_train, parsing performance did improve, yet not to a sufficient extent, especially in terms of Labeled Attachment Score (LAS). It turned out that training UDPipe on these 300 manually annotated NArabizi sentences only (Arabizi_train300) produced better scores, resulting in a parser that we did use as a pre-annotation tool in a constant bootstrap process to speed up the annotation of the remaining sentences.

Table 6: Results of UDPipe (trained 100 epochs) on the preliminary test set.

| Training corpus  | Dev LAS | Test LAS | Test UAS | Test LAS | Test UAS |
|------------------|---------|----------|----------|----------|----------|
| MIX              | 87.67   | 89.42    | 87.69    | 89.44    | 39.28    | 51.52    |
| MIX+Arabizi_300  | 87.42   | 89.20    | 87.44    | 89.22    | 55.54    | 65.36    |
| Arabizi_300      | 39.11   | 49.62    | 39.14    | 49.65    | 63.03    | 71.21    |

7 Discussion

How interleaved are French and NArabizi? As stated before, NArabizi takes its root in Classical Arabic and in multiple sources of integration of French, MSA and Berber, the Amazigh language. As the NArabizi treebank contains more than 36% of French words, it is of interest to use recent methods of visualization to see how interleaved it is with some of its source languages. To this end, we extract words embeddings using fastText (Joulin et al., 2016) from a corpus made of the “translitterated” PADT described in Section 5, the French UD GSD and NArabizi original corpus (Cotterell et al., 2014). Two-dimensional representations of the resulting embeddings space for 300 selected words are shown in Figure 2 for embeddings of size 50 and 100.

We notice that the overall shapes of both representations are very similar, apart from a non significant x-axis reversal. On the first components, increasing the embedding size does not provide more information.

We also see that French and transliterated Arabic words are clearly separated into two clusters of low standard deviation, while NArabizi words are very spread out. Some fall within the French cluster, they correspond to French words present in this Algerian dialect. Others are in the middle of the Arabic cluster, these are the purely Arabic words of the dialect. Between the two, there are Amazigh words (rak, mech), arabized French words (iomobile < French automobile), Arabic words whose Berber pronunciation has resulted in an unexpected NArabizi rendering (nta instead of expected en-ta ‘you’, mchit instead of expected machayt ‘to go-2SING’).

What is the Impact of Code-Switching in POS-tagging performance? Given the large degree of interleaving between French and NArabizi, it is interesting to assess the impact of the French vocabulary on the performance of a POS-tagger trained on French data only. For these experiments, we use the StanfordNLP neural tagger (Qi et al., 2019), which ranked 1st in POS tagging at the 2018 UD shared task, trained on the UD French...
ParTUT treebank, using French fastText vectors (Mikolov et al., 2018). In order to perform a meaningful evaluation, we split the NArabizi training set into 4 buckets of approximately 25% of it size in tokens, with a increasing proportion of identified NArabizi tokens. Results in Table 7 show a clear drop of performance between the sentences that contain more code-switching (59.55% of UPOS accuracy) and those with none (16.84%). This suggests that low-resource languages with a high-level of code-switching such as NArabizi can benefit from NLP models trained on the secondary language. The level of performance to expect from these cross-language approaches is yet to be determined.

| % of NArabizi per sent. | <60 | 60-78 | 78-100 | 100 |
|-------------------------|-----|-------|--------|-----|
| bucket set size (sent.) | 322 | 286   | 283    | 276 |
| StanfordNLP (French)    | 59.55 | 35.93 | 25.41  | 16.84 |

Table 7: POS tagging Performance with regard to code-mix proportion trained on UD French Partut treebank

8 Treebanking Costs

Following Martínez Alonso et al. (2016), we provide here the cost figures of this annotation campaign. We do not include the salaries of the permanent staff, nor do we include the overhead. These figures are meant as an indication of the effort needed to create an annotated data set from scratch. It shall be noted that even though the inter-annotator agreement gave us early indications on the difficulty of the tasks, it also acted as a metric in terms of language variability among annotators. None of them come from the same part of North-Africa and none of them has the same familiarity with the topics discussed in the web-forums we annotated. We had to constantly re-annotate sentences and update the guidelines every time new idiosyncrasies were encountered and most importantly accepted as such by the annotators. Compared to what was reported in (Martínez Alonso et al., 2016), the figures are here much higher (about 5 times higher), because unlike their work on French treebanks, we could not use preexisting guidelines for this language and because we could not keep the same team all along the project, so that new members had to be trained almost from scratch or to work on totally different layers.

| Phase | 1st | 2nd | 3rd | 4th | 5th | p.m | Costs (k€) |
|-------|-----|-----|-----|-----|-----|-----|------------|
| Annotators | 8   | 2   | 2   | 3   | 15  |     | 45         |
| Jr Researcher | 2   | 5   | 7   |     |     |     | 21         |
| Confirmed     | 6   | 6   |     |     |     |     | 21         |
| total         | 8   | 4   | 7   | 3   | 6   | 28  | 87         |

Table 8: Treebanking costs. The annotation phases are (i) Morphology/tokenization, (ii) Translation, (iii) Pre-annotation Syntax, (iv) Correction, (v) Final Syntax. P.M stands for person.month

9 Related Work

Research on Arabic dialects is quite extensive. Space is lacking to describe it exhaustively. In relation to our work regarding North-African dialect, we refer to the work of (Samih, 2017) who along his PhD covered a large range of topics regarding the dialect spoken specifically in Morocco and generally regarding language identification (Samih et al., 2016) in code-switching scenario for various Arabic dialects (Attia et al., 2019).

Unlike NArabizi dialects, the resource situation for Arabic dialects in canonical written form can hardly be qualified as scarce given the amount of resources produced by the Linguistic Data Consortium regarding these languages, see (Diab et al., 2013) for details on those corpora. These data have been extensively covered in various NLP aspects by the former members of the Columbia Arabic NLP team, among which Mona Diab, Nizar Habash, and Owen Rambow, in their respective subsequent lines of works. Many small to medium scale linguistics resources, such as morphological lexicons or bilingual dictionaries have been produced (Shoufan and Alameri, 2015). Recently, in addition to the release of a small-range parallel corpus for some Arabic dialects (Bouamor et al., 2014), a larger corpus collection was released, covering 25 city dialects in the travel domain (Bouamor et al., 2018).

Regarding the specific NLP modeling challenges of processing Arabic-based languages, as part of the morphologically-rich languages, recent advances in joint models have been addressed by Zalmout and Habash (2019) that recently efficiently adapted a neural architecture to perform joint word segmentation, lemmatization, morphological analysis and POS tagging on an Arabic dialect. Recent works on cross-language learning using the whole massively multilingual pre-trained language models artillery have started to emerge
(Srivastava et al., 2019). If successful, such models could help to alleviate the resource scarcity issue that plagues low-resources languages in the more-than-ever data hungry modern NLP.

10 Conclusion

We introduced the first treebank for an Arabic dialect spoken in North-Africa and written in romanized form, NArabizi. More over, being made of user-generated content, this treebank covers a large variety of language variation among native speakers and displays a high level of code-switching. Annotated with 4 standard morphosyntactic layers, two of them following the Universal Dependency annotation scheme, and provided with translation to French as well as glosses and word language identification, we believe that this corpus will be useful for the community at large, both for linguistic purposes and as training data for resource-scarce NLP in a high-variability scenario. In addition to the annotated data, we provide around 1 million tokens (over 46k sentences) of unlabeled NArabizi content, resulting in the largest dataset available for this dialect. Our corpora are freely available under the CC-BY-SA license and the NArabizi treebank is also released as part of the Universal Dependencies project.

Acknowledgments

The work was partially funded by the French Research Agency projects ParSiTi (ANR-16-CE33-0021), SoSweet (ANR15-CE38-0011-01) and by the French Ministry of Industry and Ministry of Foreign Affairs via the PHC Maimonide France-Israel cooperation programme, as well as by the Sagot’s chair in the PRAIRIE institute funded by the French national agency ANR as part of the “Investissements d’avenir” programme under the reference ANR-19-P3IA-0001.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for arabic. In HLT-NAACL Demos.

Mohammed Attia, Younes Samii, Ali Elkahky, Hamdy Mubarak, Ahmed Abdelali, and Kareem Darwish. 2019. POS tagging for improving code-switching identification in Arabic. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 18–29, Florence, Italy. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2015. Arabic diacritization with recurrent neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2281–2285, Lisbonne, Portugal. Association for Computational Linguistics.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multialectal parallel corpus of Arabic. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14), pages 1240–1245, Reykjavik, Iceland. European Language Resources Association (ELRA).

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic dialect corpus and lexicon. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Ryan Cotterell, Adithya Renduchintala, Naomi Saphra, and Chris Callison-Burch. 2014. An Algerian Arabic-French code-switched corpus. In Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme, page 34.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual. Technical Report CCLS-13-02, Center for Computational Learning Systems, Columbia University.

Nizar Habash. 2010. Introduction to Arabic Natural Language Processing. Morgan and Claypool.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hèrve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651.
Éric de La Clergerie, Benoît Sagot, and Djamé Seddah. 2017. The ParisNLP entry at the ConLL UD shared task 2017: A tale of a Parsing- Tragedy. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 243–252, Vancouver, Canada. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea, pages 25–30. The Association for Computer Linguistics.

Teresa Lynn and Kevin Scannell. 2019. Code-switching in irish tweets: A preliminary analysis. In Proceedings of the Celtic Language Technology Workshop, pages 32–40.

Héctor Martínez Alonso, Djamé Seddah, and Benoît Sagot. 2016. From Noisy Questions to Minecraft Texts: Annotation Challenges in Extreme Syntax Scenarios. In The 2nd Workshop on Noisy User-generated Text (W-NUT).

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

Robert Munro. 2010. Crowdsourced translation for emergency response in haiti: the global collaboration of local knowledge. In AMTA Workshop on Collaborative Crowdsourcing for Translation, pages 1–4.

Carol Myers-Scotton. 1993. Common and uncommon ground: Social and structural factors in codeswitching. Language in Society, 22(4):475–503.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyeh, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Virgincia Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aloischa Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Çebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çoltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantzaz Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Drogonova, Puneet Dwivedi, Marhaba El, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richard Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarina Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Göktürmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillame, Céline Guillot-Barbance, Nizar Habash, Jan Hajčí, Jan Hajčí jr., Linh Hà Mỵ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florin Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelinek, Anders Johannsen, Fredrik Jorgensen, Hüner Kaşkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phurong Lê Hông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lysashevskaya, Teresa Lynn, Vivien Macketzkan, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cátălinu Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalovskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisepp, Pinkey Nainwani, Juan Ignacio Navarro Horniacek,
Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Robert Olúòkun, Adédayọ̀ọstling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitter, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Lauma Pretkalniņa, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibusirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Simková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Yuta Takehashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tzarfaty, Francis Tyers, Sumire Uematsu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. *Universal dependencies 2.2*. LIN-DAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Adam Nossiter. 2019. *Algeria protests grow against president bouteflika, ailing and out of sight*. In *New York Times* (March 01, 2019).

Pedro Javier Ortíz Suárez, Benoît Sagot, and Lan-
Appendix

A.1 Class-noise for our language classifier

Given the large number of noisy data in Common Crawl, a class noise is added to the classification model we presented section 4.2 and is built according to the following empirical rules:

- If the word “url” appears more than two times, the sentence is added to class noise.
- If the sentence has more than four “[” , the sentence is added to class noise.
- The same rule as above works for “]”, “ ”, “{” or “}” symbols.
- If the word “http” appears more than two times, the sentence is added to class noise.
- If more than two “@” character, the sentence is added to class noise in order to capture sentences with email address or tweets with only mentioned people.
- If the phrase ”WARC-Refers-To” appears, the sentence is added to class noise.

A.2 Post-processing steps

- Get unique sentences (about 20K sentences).
- The model is likely to classify as Arabizi sentences which contain any letter repeated a lot of times in a row (e.g. “iiiiiiii..”, “uuuuuuu..”, “ffffff...”). These sentences are deleted from the dataset.
- Due to the n-gram embeddings, ”lah” is considered as a marker of Arabizi, so a lot of sentences containing ”blah” are classified as Arabizi. If this phrase appears more than 5 times, the sentence is deleted.
- Figures and numbers are widespread in Arabizi (particularly ”3” and ”9”), so the model classifies too many sentences which contains only numbers. Therefore, sentences which have more ”number of figures” characters than 80% of the number of characters (excluded figures) are deleted.