Language Identification for Social Media: Short Messages and Transliteration

Pedro Miguel Dias Cardoso
Synthesio
8 rue Villédo
Paris, France
pedro@synthesio.com

Anindya Roy
Synthesio
8 rue Villédo
Paris, France
aroy@synthesio.com

ABSTRACT

Conversations on social media and microblogging websites such as Twitter typically consist of short and noisy texts. Due to the presence of slang, misspellings, and special elements such as hashtags, user mentions and URLs, such texts present a challenging case for the task of language identification. Furthermore, the extensive use of transliteration for languages such as Arabic and Russian that do not use Latin script raises yet another problem.

This work studies the performance of language identification algorithms applied to *tweets*, i.e. short messages on Twitter. It uses a previously trained *general purpose* language identification model to semi-automatically label a large corpus of tweets - in order to train a tweet-specific language identification model. It gives special attention to text written in transliterated Arabic and Russian.

Keywords

language identification; document classification; microblogging; transliteration

1. INTRODUCTION

Social media is becoming an important source of information for companies, from public relations to marketing and sales. We can detect new events [1], predict political outcomes [11], predict sporting event outcomes [10] or classify and visualize data for market and customer studies. As the initial step of any Natural Language Processing (NLP) pipeline, being sure that the input language is the correct one is very important.

Language identification was already considered a solved problem by [8], even for data from the World Wide Web. However, in practice, the performance remains poor for short and noisy messages such as those seen on social media. One example of this problem is described in [9] where the authors study language identification on forums with multilingual users. In these forums, users mix multiple languages, sometimes in the same sentence. In our experiments, this is often seen in retweeted messages[1]. Another problem is the use of slang words and special tokens as seen in this tweet example, with the token definitions in table 1:

```
@carusobello Audi e-tron quattro dominate
#LeMans24 http://t.co/8sk6x4so
```

Verbatim	Token type
@carusobell	user reference, no language
Audi e-tron quattro	car model, partially Italian
dominate	text in english
#LeMans24	Hashtag
http://t.co/8sk6x4so	URL

Table 1: Token definitions for an example tweet.

The second problem of social media is the use of non-official and non-formal language transliteration. Such cases may be interpreted as a new language to classify. However, this is complicated by the fact that they do not follow well-defined conventions and there is no single way to write the same word transliterated. Each group of users - be it based on geography or on demographics such as age and gender - will have its own way of writing in transliterated mode. An example can be seen here for the Arabic language:

```
wktashaft fel a5er eny bakol la7ma naya
w roz msh mstwewy -.-
```

In english : *I found out at the end that I was eating raw meat and badly cooked rice* We choose Arabic and Russian as languages to identify in transliterated text, as they are two widely used languages on the internet. Russian, that exists with an official transliteration table, and Arabic, that grew organically by users.

Related work is briefly presented in section 2. In section 3, we briefly describe transliterated Arabic and Russian. In section 4, we present our training and evaluation corpora. Model creation is described in section 5 and detailed evaluation results in section 6. Principal conclusions are drawn in section 7.

2. RELATED WORK

Cavnar [2] was one of the first to present a work for language identification as a document classification problem, using a linear classifier and character n-gram as features. More recently, [4] used a multinomial Naive Bayes classifier with n-gram feature selection over multiple domains. Following the authors, we denote this as the `langid.py` model in this work. Naive Bayes is a simple and lightweight algorithm that presents good results when preceded by an efficient feature selection step [7]. In this case, feature selection

[1] A retweet is when a user forwards a message, adding some times new text at the start or end

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
DOI: http://dx.doi.org/10.1145/2872518.2890560 .

is done by choosing the unigram, bigram and trigrams that give the highest Information Gain with respect to languages but low with respect to specific domains. Implementation details of this algorithm are provided in [5]. Detailed comparisons of this algorithm against other algorithms using a language-labeled Twitter message database are provided in [6]. We shall refer to this later.

3. TRANSLITERATION

The term *transliteration* denotes the conversion of a text from one script to another. This is usually a one-to-one mapping and fully reversible.

In the internet, the growth of transliteration use came from the hardware limitations. The first personal computers and mobile phones did not allow writing in local script forcing users to find ways to communicate. This was, in most cases, an organic evolution, not always using the defined standards for transliteration. Nowadays, even as hardware evolved, and local scripts got supported, some social media user still use transcriptions, as sometimes it is seen as easier for short and un-informal messages.

3.1 Arabic Transliteration

In [3] the author presents a comprehensive list of existing transliterations used for the Arabic language. Our focus, being on social media, is on the so called Arabic Chat Alphabet, also called Arabesh. All other forms of more formal transcriptions, such as SATTS and Buckwalter[2] are widely ignored by Internet users.

Across the Middle East and North Africa, classical Arabic[3] is spoken and understood, despite regional differences and semantical variations. Locally, official languages are merged with traditional regional languages, or those from colonial times, building local variations also known as dialects. Sometimes these differences are such that speakers can not communicate across countries in the region.

Arabesh follows these regional variations, and therefore has multiple variations. Although done for practical purposes, grouping a few countries together in terms of dialect can be inaccurate and may lead to the loss of semantic and phonetic differences. Still, we present a possible geographical and cultural split over 3 main areas, namely:

1. Levant (Syria, Jordan, Palestine, and Lebanon), Egypt and Sudan,
2. Iraq and the gulf countries,
3. North Africa

While some countries may use the same English letters to transliterate a letter, the commonalities of transliteration approaches are still contingent on:

1. Semantic differences
2. Phonetic differences
3. The integration of numbers to replace the Arabic sounds not available within the English language.

An example of multiple possible transliterations is seen in the letter Ayn that can be transcribed by ai, ay or 3. The last one is due to the graphical proximity of 3 to the original symbol.[4]

3.2 Russian Transliteration

Russian transliteration to Latin script is less used these days. It is used mostly by Russian speakers when they do not have access to Cyrillic keyboard, for example, when they are abroad.

Russian does have an official transliteration table, but in our evaluations we noticed visible differences across age and regional groups.

4. CORPORA

The corpora used to train the general purpose `langid.py` system described in [4][5] consist mostly of long and well-written documents, including webpages, Reuters news stories, and manual translations of software packages maintained by the Debian project. For the purpose of language identification specifically for social media, we created new in-house corpora primarily using data collected by our sourcing section during the last six years and stored in Synthesio's data centres. Two types of corpora were created: 1) semi-automatically labeled corpora for training and initial evaluation, and 2) hand-labeled corpora for final evaluation.

4.1 Semi-automatically labelled corpora

Four categories of social media were defined, plus a fifth category using Wikipedia. They are as follows:

- **Wikipedia**: articles from Wikipedia. These present a formal writing style.
- **professional**: posts from newspapers, company, institution and government pages, magazines, news agencies, job listings and official releases. These are typically well-written.
- **personal**: posts from non-professional sources. These include blogs, forums, consumer opinions, customer surveys, hosted communities. Typically less well-written.
- **micro-blogging**: posts coming from micro-blogging sources, such as Twitter and Seina-Weibo. Typically, short and noisy. (ref. Section 1)
- **comments**: These are comments on sharing sites, like Facebook, and Youtube. These are not the messages themselves, but the comments linked to them. They are typically short and present the same level of difficulty as micro-blogging.

In contrast to [5], we included a micro-blogging category. Raw messages in the micro-blogging category were cleaned by removing special tokens such as user mentions, emails, URLs and hashtags. The cleaned messages were grouped into a sixth category, micro-blogging *clean*.

4.1.1 Language labelling

Language labels for messages in the in-house corpora were set as follows: For professional and personal, the language selected manually by the sourcing team was used as an initial guess which was confirmed using `langid.py`. For each source (newspaper, company web page, magazine, etc), if the score provided by `langid.py` for the manually selected language was less than 0.8, the source was rejected. For microblogging and comments, we used `langid.py` to estimate the language, rejecting all messages with the `langid.py` score less than 0.9. For Wikipedia, the language is already known.

4.1.2 Transliterated languages

To collect text in transliterated languages, we used search queries based on common words of the language. We restricted to specific countries, for improved precision. Some of the words we used for Arabic are as follows:

[2] http://www.qamus.org/transliteration.htm

[3] Classical Arabic refers to the one used in writing for official and formal reasons (i.e. news, books, formal communication)

[4] https://en.wikipedia.org/wiki/Ayin

```
ana, inta, inti, inte, howa, houe, hiya,
hiye, ni7na, humma, houma, hinne,
hiyya, kint, kinte, konti, kanou, kanoo,
eih, shou, shoo, chniya, chnawwa,
chkoun, 3lach, 3lah, fi, 3ala
```

These were restricted to Morocco, Algeria, Tunisia, Iraq, Egypt, Libya, Sudan, Saudi Arabia, United Arab Emirates, Jordan, Kuwait, Lebanon, Oman, Qatar, Syria, Yemen and Palestine. The search was done on an indexed databased collected by Synthesio over multiple years.

These corpora were divided into two parts uniformly across all 6 categories, in the ratio 4:1. The first (bigger) part is termed as SynthesioTrain and the second (smaller) part as SynthesioEval1. Table 2 shows the total number of messages and their average size in terms of number of tokens per category. As expected, the average size of the micro-blogging clean category is smaller. Languages, when possible, are equally represented in each category, including transliterated languages when these exist, i.e. for micro-blogging, comments, and personal.

Class	SynthesioTrain		SynthesioEval1	
	#messages	size	#messages	size
Wikipedia	1230K	2792.9	208K	2284.8
professional	529K	1091.0	99K	1098.0
personal	483K	642.1	92K	652.1
micro-blogging	1072K	146.2	123K	220.8
micro-blog clean	1072K	103.6	123K	160.7
comments	141K	247.5	28K	278.6

Table 2: Total number of messages and average size per message for 6 categories of SynthesioTrain and SynthesioEval1 corpora.

4.2 Hand-labelled corpora

We collected a separate corpus composed exclusively of tweets that were entirely *hand-labelled*, which we denote as SynthesioEval2 corpus. Table 3 presents the total messages in this corpus, for a total of 32 languages. We do not have transliterated examples in this corpus. As with the micro-blogging categories in the SynthesioTrain and SynthesioEval1, the SynthesioEval2 is divided into micro-blogging and micro-blogging clean corpora, where the latter was created by removing the special tokens from the former.

Class	#messages	size
micro-blogging	14620	98.8
micro-blogging clean	14620	75.6

Table 3: Total number of messages and average size of message for SynthesioEval2 corpus.

4.3 TwitUser corpus

Additionally, we also consider the TwitUser corpus presented in [6] for comparison. This corpus contains 14,178 language-labeled Twitter messages across 65 languages, constructed using a mostly-automated method.

5. MODEL CREATION

The SynthesioTrain corpus was used for training, closely following the approach presented in [5]. The final goal was to create a new model that will be able to identify multiple languages plus two transliterated language: Arabic and Russian. For comparison of results, a second model was created using SynthesioTrain, but removing transliterated languages. For comparison with our new models trained on our in-house corpus, we also use the original `langid.py` model defined in [5][6].

Our new model identifies 80 languages (78 languages + 2 transliterated) with a total of 14,179 n-gram features. The models developed for the `langid.py` package contain a total of 7480 n-gram features for a total of 97 languages. We can see that the use of smaller documents will force the use of more features, therefore being able to classify smaller messages, as we will see below.

6. RESULTS

In this work, we want to understand if the addition of non-structured languages, such as transliterated languages, will affect the identification of other languages, and what accuracy we can obtain for the added languages. Also, we will try to understand the improvement brought by the use of an adapted training corpus, in this case, one containing micro-bogging messages. Finally, we will evaluate the influence of user language priors on the results.

For the prior calculation, we calculate the probability of each user to write in a particular language. The calculation is done with a simple "add-one" smoothing on all tweets for that user that we can find in our database. This might not be the full list of tweets by the same user. We also take into consideration the profile language definition, where the language of the profile was considered to count as 20 written messages in that language. This strategy was optimised experimentally.

6.1 SynthesioEval1

First, we compare our new models using the five categories of the SynthesioEval1 corpus, excluding Wikipedia. Table 4 presents the results for the models in terms of percent accuracy. Column **w/o translate** denotes the case where transliterated data was excluded for *both* training and evaluation. Column **w/ translit** considers the case where transliterated data from Arabic and Russian was used (along with non-transliterated data) for both training and evaluation. It is observed that addition of transliterated data degrades performance, but only slightly. Expectedly, the micro-blogging category is the one with the worst results. Micro-blogging clean shows slightly better results. Longer messages such as personal and professional are the ones with best results, close to what is expected for a normal language identification system.

Corpus	w/o translit	w/ translit
personal	95.486	95.148
professional	96.811	96.491
micro-blogging	80.073	79.931
micro-blogging clean	81.190	81.021
comments	92.675	92.228

Table 4: Classification accuracy with and without transliterated languages using SynthesioTrain corpus for training and SynthesioEval1 corpus for evaluation.

The performance specifically for Arabic and Russian is further detailed in table 5. It shows different cases with and without transliteration. Two models are considered, 1) **w/o translit** trained without transliterated text, and 2) **w/ translit** trained using both transliterated and normally written text. For evaluation, we consider 3 cases: 1) text written in the language's normal script (i.e. not transliterated), denoted as Arabic or Russian, 2) text entirely transliterated, denoted as Arabic trans and Russian trans, and 3) text which is a mix of normal script and transliterated, denoted as Arabic mix and Russian mix.

It shows that when having the two languages, the results are worse for both. This is so due to many messages being written in original and new scripts. An example of this is when a message is a comment to a previous one, the first being written in the normal alphabet and the second part in Latin alphabet. When we do not differentiate between the two ways of writing, the results improve. We can see it on the rows Arabic mix and Russian mix. Still, we do not reach the same accuracy as when not considering the transliterated languages.

model	language	pers	prof	micro	comm
w/o translit	Arabic	99.2	99.9	98.4	99.8
w/ translit	Arabic	95.8	99.7	94.4	97.8
w/ translit	Arabic trans	91.9	94.5	90.8	90.9
w/ translit	Arabic mix	97.5	97.5	94.0	97.1
w/o translit	Russian	98.5	98.1	99.2	98.0
w/ translit	Russian	96.5	97.2	93.8	94.5
w/ translit	Russian trans	87.5	87.5	78.9	88.4
w/ translit	Russian mix	96.1	95.7	99.1	96.1

Table 5: Detailed classification accuracy for transliterated languages using SynthesioEval1 corpus.

6.2 SynthesioEval2

In Table 6, we evaluate the original `langid.py` model from [5] and compare its performance with our model trained using SynthesioTrain corpus. For evaluation, the hand-labelled SynthesioEval2 corpus was used. Unlike SynthesioEval1 corpus, where data was semi-automatically labeled using the `langid.py` model, SynthesioEval2 is composed of tweets *entirely* labelled by hand. We show results for both micro-blogging and micro-blogging clean categories from this corpus.

Category	Model	Accuracy
micro-blogging	`langid.py`	66.738
	our model	86.698
micro-blog clean	`langid.py`	66.764
	our model	87.839

Table 6: Classification accuracy using SynthesioEval2 corpus, with original langid.py model and our new model.

We can see that the original `langid.py` model is significantly outperformed by the new model, which benefited from short and noisy messages in the training data. The improved performance is most probably due to the bigger quantity of features used. This is probably coming from the feature selection, where more features are selected. Whereas in a big document we would not select many features, with short messages we are forcing the selection algorithm to select more.

Again, cleaned messages show a slight improvement in classification.

6.3 TwitUser

We compared our model, with the original `langid.py` model, using the TwitUser corpus. For our model, we cleaned the tweets from URL, hashtag, user mentions and emails, as well as punctuations. For the original model we left the tweet as it was, because it showed higher accuracy in the corresponding paper[6]. Our model achieved an accuracy of 87.6 compared to 84.2 by `langid.py`. The result shows that our model, trained with short messages as part of the training corpus, presents better results. It also presents the highest accuracy among all other language classification methods presented in [6].

7. CONCLUSIONS

We presented results for identification of non-official languages, written in transliterated format with Latin script. Their results were not at the same level as other languages that have a standard way of writing, but still they reached a reasonably good level. This was possible without a big impact for other languages. The biggest impact is on the official languages themselves, ex: formal Arabic vs transliterated Arabic, the main reason being that in the corpus, many messages contain parts in Arabic characters and other parts in transliterated.

Further, we show that by including micro-blogging messages in the training corpus, we are able to substantially improve language identification performance on short and noisy messages.

8. REFERENCES

[1] F. Atefeh and W. Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 2013.

[2] W. B. Cavnar, J. M. Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.

[3] D. R. Lawson. An evaluation of arabic transliteration methods. *School of Information and Library Science, North Carolina, USA*, pages 1–55, 2008.

[4] M. Lui and T. Baldwin. Cross-domain feature selection for language identification. In *In Proceedings of 5th International Joint Conference on Natural Language Processing*. Citeseer, 2011.

[5] M. Lui and T. Baldwin. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30. Association for Computational Linguistics, 2012.

[6] M. Lui and T. Baldwin. Accurate language identification of twitter messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM) @ EACL 2014*, pages 17–25. Association for Computational Linguistics, 2014.

[7] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[8] P. McNamee. Language identification: a solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges*, 20(3):94–101, 2005.

[9] D.-P. Nguyen and A. S. Dogruoz. Word level language identification in online multilingual communication. Association for Computational Linguistics, 2013.

[10] S. Sinha, C. Dyer, K. Gimpel, and N. A. Smith. Predicting the nfl using twitter. In *Proceedings of the ECML/PKDD Workshop on Machine and Data Mining for Sports Analytics*, 2013.

[11] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.