Evaluation of Off-the-Shelf Language Identification Tools on Bulgarian Social Media Posts

Silvia Gargova, Irina Temnikova, Ivo Dzhumerov, Hristiana Nikolaeva
Big Data for Smart Society Institute (GATE), Sofia, Bulgaria
svgargova@gmail.com, irina.temnikova@gate-ai.eu,
i.dzhumerov@gmail.com, hnikolaeva@gmail.com

Abstract

Automatic Language Identification (LI) is a widely addressed task, but not all users (for example linguists) have the means or interest to develop their own tool or to train the existing ones with their own data. There are several off-the-shelf LI tools, but for some languages, it is unclear which tool is the best for specific types of text. This article presents a comparison of the performance of several off-the-shelf language identification tools on Bulgarian social media data. The LI tools are tested on a multilingual Twitter dataset (composed of 2966 tweets) and an existing Bulgarian Twitter dataset on the topic of fake content detection of 3350 tweets. The article presents the manual annotation procedure of the first dataset, a discussion of the decisions of the two annotators, and the results from testing the 7 off-the-shelf LI tools on both datasets. Our findings show that the tool, which is the easiest for users with no programming skills, achieves the highest F1-Score on Bulgarian social media data, while other tools have very useful functionalities for Bulgarian social media texts.

Keywords: language identification, social media, evaluation, off-the-shelf tools, Bulgarian.

1 Introduction

Automatic Language Identification (LI) is a well-addressed task, with many existing approaches, tools, and evaluation initiatives (Jauhiainen et al., 2019; Garg et al., 2014). LI solves the problem of those users, who need to detect the language of a large number of texts, and thus cannot perform this task manually, as it will take them a large amount of time and manual efforts. Such users (for example linguists), do not have the knowledge, skills, or interest to develop their own LI tool or to train existing tools with their own data, and thus prefer using an existing off-the-shelf LI tool. As a first step, they are naturally interested to know which is the best tool for the specific language (e.g. Bulgarian, Romanian, Hindi) and type(s) of text of their interest (e.g. *news articles* or *social media posts*). However, there is no sufficient information about which off-the-shelf LI tools are the best for all specific language/type-of-text combinations. For this reason, we are sharing our findings of the best off-the-shelf LI tools and their functionalities for the specific language and type of text of our interest. By doing this we aim to assist other users or researchers, who need to use such tools for their language identification tasks.

Our **language of interest is Bulgarian**, and the **the type of text - social media posts**, and in this article we are reporting the results of comparing several off-the-shelf LI tools on Bulgarian social media data.

Our work is motivated by the wish to solve the issue of filtering out any non-Bulgarian tweets from social media corpora. Following our task to collect and pre-process Bulgarian social media datasets for detecting fake content, our first observation was that despite using the Twitter API for collecting only posts in Bulgarian, our dataset contained many tweets (see Table 1 for precise numbers) in languages similar to Bulgarian or written in Cyrillic alphabet (for example Macedonian, Serbian, Russian, Kazakh, etc.). We have observed a similar issue when using other dataset collecting methods, such as Facebook's CrowdTangle. Determining the best LI tool for filtering out non-Bulgarian posts was thus a must.

To be able to identify the most appropriate LI tool and motivate our choices, we have to first understand and describe the characteristics of the language (Bulgarian) and type of text (social media posts) of our interest.

The Bulgarian language is part of the South Slavic languages' group within the Indo-European language family. In lexical, phonetic and grammati-

cal terms, Bulgarian has both Slavic and non-Slavic features. It is officially written in Cyrillic alphabet, but in social media and Internet forums people often use several variants of Latin transcription. Bulgarian is the official language of the Republic of Bulgaria. It has a literary form, used in all spheres of public life, and a number of local dialects, some of which are similar to the languages of North Macedonia and Serbia.

Social media texts (including those in Bulgarian) are known for being different from standard texts by being much shorter (e.g. a tweet can contain a maximum of 280 characters), frequently containing orthographic errors, Internet slang, non-dictionary words, emoticons, hashtags, unfinished sentences, and broken or non-standard syntax, and thus being challenging for many Natural Language Processing applications (Farzindar and Inkpen, 2017). In addition to that, social media posts may sometimes contain words and phrases, written in different languages – a phenomenon, known as *code-switching* (Androutsopoulos, 2013).

A LI tool, which would be perfect for recognizing Bulgarian social media posts, thus, should:

1. Have the highest possible performance (e.g. an over 98% F1-score);

2. Be able to recognize Bulgarian texts, written both in Cyrillic alphabet and in the various Latin transcriptions (typical for the Bulgarian Internet slang);

3. Be able to handle the above described social media posts' characteristics, including the cases when the post is written in two or more languages.

In order to discover the most appropriate LI tool for correctly identifying the Bulgarian language posts in social media data, we have determined the most frequently used Off-the-Shelf LI tools (OSLI), by examining publications and consulting other researchers. We have then tested them on two datasets - a multilingual (mostly Bulgarian) dataset, collected from Twitter on the topic of Covid-19 with 2979 tweets, manually annotated for language(s), and a Bulgarian language dataset (Shaar et al., 2021), used for fake content detection initiatives, consisting of 3350 tweets.

The article provides the results of the human annotation and of testing the tools, as well as shows which tools achieve the highest F1-scores on the two datasets, and which have the most useful functionalities for social media posts.

The rest of the article lists the relevant Related work (Section 2), a description of the datasets that we used for testing the tools (Section 3), our Methodology (including human annotation and the tested tools - in Section 4), the annotation and testing Results and some Discussion (Section 5), and finally, the Conclusions (Section 6).

2 Related Work

Automatic Language Identification (LI) is a widely addressed task, but it still has some issues which are hard to resolve. Among them (Jauhiainen et al., 2019) are:

- Distinguishing between similar languages or dialects;
- Short and noisy texts;
- Documents, written in more than one language;
- Languages with different orthographies.

All these issues apply to Bulgarian social media posts.

There have been a number of previous works which include Bulgarian among other languages in their LI tasks or datasets, for example (Zampieri et al., 2015; Jauhiainen et al., 2017; Malmasi, 2017; Bergsma et al., 2012; Baldwin and Lui, 2010; Thoma, 2018). Most of them, however, use datasets compiled from types of texts, which are different from social media (e.g. Wikipedia, news articles, Europarl, and the Universal Declaration of Human Rights). Also, most of these works do not apply existing off-the-shelf LI tools to detect Bulgarian, but rather implement their own methods.

The closest works to ours are those of (Abainia et al., 2016), (Bergsma et al., 2012), (Bankov et al., 2017), and (Lui and Baldwin, 2014). Among them, however, there is no work which compares the performance of different off-the-shelf LI (OSLI) tools on Bulgarian social media posts and publishes the results.

Specifically, (Abainia et al., 2016) are similar to us as they use short forum texts, including such written in Bulgarian, but no testing of OSLI tools is performed. (Bergsma et al., 2012) compare LI methods implemented by them with three off-the-shelf tools (TextCat, Google CLD and langID.py)

on a multilingual Twitter dataset containing also Bulgarian. Their methods outperform the OSLI tools, but there are no results reported separately for Bulgarian. (Bankov et al., 2017) also observes that Twitter's accuracy for Bulgarian language identification is not satisfactory, however, the author does not test any OSLI tools on Bulgarian tweets.

Finally, there are publications on testing various off-the-shelf LI tools on specific languages, but not on Bulgarian. For example, (Lui and Baldwin, 2014) compared 8 OSLI tools on manually annotated tweets in English, Chinese, and Japanese.

While several OSLI tools include Bulgarian, according to our knowledge, there is no other published comparison of off-the-shelf LI tools for this language, especially for social media texts.

3 Data Used

We have used two datasets - a randomly selected subset of our own Twitter dataset, and the Bulgarian language dataset, made available for the CLEF2021 CheckThat! Lab, Task 1 (check-worthiness). From now on we refer to this dataset as ***CLEF2021 dataset***) [1] (Shaar et al., 2021). The large original version of our dataset contains 52810 tweets, from which we extracted 3124 tweets, which were annotated for their language by human annotators. We have removed some non immediately noticeable duplicates and did some additional cleaning (based on our annotators feedback), and obtained 2966 final human-annotated tweets, on which we tested the LI tools. Respectively, the CLEF2021 CheckThat! Lab, Task 1 dataset for Bulgarian originally contained 3350 entries (tweets).

We have decided to compare the results of the same off-the-shelf LI tools on the subset of our dataset with those on the CLEF2021 Bulgarian dataset, as they both had comparable number of tweets and are on the same topic (Covid-19).

Before testing the LI tools on the CLEF2021 CheckThat! Lab, Task 1 Bulgarian dataset, we have merged the Bulgarian versions of its *train* and *dev* datasets into one to have more data. After a quick analysis of the merged CLEF2021 dataset, we noticed two issues: unusually long entries (consisting in many tweets concatenated in one row) and a few tweets in other languages. We separated the long rows into single posts and removed the tweets that are not in Bulgarian, which resulted in 3373 final tweets.

Languages	Num. of tweets
Bulgarian	2491
Macedonian	248
Russian	214
English	43
Mongolian	38
Uzbek	22

Table 1: Number of tweets in the most frequent languages in our 3124 tweets Covid-19 dataset.

Stats	Covid-19	CLEF2021
Num. tweets	2966	3373
Total words	47628	66502
Mean tweet length	16.06	19.72
Shortest tweet	1	5
Longest tweet	54	108

Table 2: Statistics of the two datasets used for testing the tools.

The tweets in our original large dataset[2] were collected via the Twitter API for the period May 2020 - March 2021. The keywords used were "ваксина" ("vaccine", Sg.) and "ваксини" ("vaccines", Pl.) in Bulgarian language and using the Cyrillic alphabet. From this large dataset we have selected a smaller random subset from different time intervals. Each tweet from this final dataset (from now on referred to as ***Covid-19 dataset***) was manually annotated by two annotators for its language. The annotation methodology is described in detail in Section 4.1.

The first thing that we noticed during manual annotation, was that our dataset contained posts in other languages besides Bulgarian. Table 1 shows the most frequent languages in our dataset for the languages, in which there are more than 10 tweets. The length of the posts in our dataset was quite varied - we had one-word tweets and much longer tweets (differently from the CLEF2021 dataset, which contained only tweets long enough to be considered fact-checkable *claims*). Many of the posts were written in more than one language. There were also 2 posts that contained only emoji.

Table 2 shows the statistics of both datasets. "Covid-19" stands for our Covid-19 dataset, "Num. tweets" indicates the total number of tweets per

[1] https://gitlab.com/checkthat_lab/clef2021-checkthat-lab/-/tree/master/task1/data/subtask-1a–bulgarian. Last accessed on April 27, 2022.

[2] This dataset cannot be shared due to specific access restrictions.

dataset, "Total words" - the total number of words in each dataset, "Mean tweet length" is the mean length of the tweet in words, while "Shortest tweet" and "Longest tweet" were the tweets containing the lowest and the highest number of words.

As it can be seen in Table 2, the CLEF2021 dataset contains longer tweets than ours. This can be of advantage to the LI tools.

4 Methodology

In order to test the existing off-the-shelf LI tools, we have performed manual language annotation of the 3124-tweets-subset of our dataset (the one containing duplicates), which is described in Subsection 4.1, and selected a number of freely available and functioning LI tools (described in Subsection 4.3). The methodology, which we followed for testing the tools on both datasets is described in Subsection 4.2.

4.1 Annotation methodology

The aim of the manual annotation of our subset tweets dataset was to focus on **distinguishing specifically Bulgarian**, rather than correctly annotating all the languages of all the tweets in our dataset. This was motivated first by our aim to find the best LI tool for Bulgarian, but also by the knowledge of languages of our annotators.

We had two annotators, who are professional linguists, native in and specializing in Bulgarian language. They used Google Spreadsheets as an annotation tool, due to its simplicity. The spreadsheet had three columns, containing the tweet ID, the text of the tweet and several language-related categories in a fall-down menu to choose from. Appendix A shows a screenshot of the spreadsheet containing mock examples of annotated tweets.

The annotators were asked to only decide if the tweets are written in Bulgarian (**bg**) OR in Another language (**another**), without distinguishing exactly in which other language. As we are planning in future work to examine the performance of some of the tools in distinguishing the different languages present in multilingual tweets, we also asked the annotators to comment on which tweets are multilingual and whether they contain Bulgarian language or not. As there were some unclear cases, we provided an additional category "Unknown". The annotation categories are shown in Table 3.

We have considered multilingual also those tweets, which contained hashtags, written in another language (e.g. in English). However, we have asked the annotators to ignore the keyword "Covid-19" (and its versions, e.g. Covid), written in English, as they were too frequent due to the topic of our dataset.

The two annotators received initial training, worked separately, and did several rounds of the annotation process until the annotations and the guidelines were finalized.

As after this process there were still cases in which the annotators disagreed, to facilitate the comparison with the LI tools, we have assigned a third **hyper-annotator**. The hyper-annotator reviewed the cases of disagreement of the two annotators and decided on a final annotation category for each tweet. The hyper-annotator was also a a linguist, specialist in Bulgarian language. In order to take the correct final decisions, the hyper-annotator was allowed to have a look at the original tweet in Twitter and check information about the user who posted it, including his/her location and other tweets.

As testing the tools' performance in identifying multiple languages within the same tweet is beyond the scope of this article, the tweets, annotated as "bg-multilingual" and "bg" categories were merged into **"bg"** and "another-multilingual" and "another" were merged into the category **"another"**. We have also removed the tweets, left annotated as "unknown" by the hyperannotator. This gave us a final number of 2966 manually annotated tweets.

See Section 5.1 for a discussion of the manual annotation results.

4.2 Testing Methodology

Our aim was to test only freely accessible LI tools (not paid ones).

During testing, we wanted to check the performance of the LI tools with the tweets as they are (we call these tweets **"raw"**) and with tweets, from which several Twitter-specific elements were removed (we call these tweets **"cleaned"**) and whether there was any change or improvement in performance if the data was cleaned in advance. Our hypothesis was that Twitter-specific elements (e.g. hashtags, URLs, and mentions) would hinder the performance of LI tools.

For this purpose we performed two experiments - one with *raw data* and another with *cleaned data*. For the first experiment we used our dataset as it is (we only deleted duplicates). For the second

Annot. category	Explanation
bg	You are sure that the tweet is written entirely in Bulgarian language.
another	You are sure that the tweet is not in Bulgarian, regardless of whether you know what other language it is written in.
unknown	You are not sure if the tweet is in Bulgarian or in another language, but you have at least a minimal suspicion that it may be written in Bulgarian.
another-multilingual	The tweet is bilingual or multilingual, but you are sure that none of the languages is Bulgarian.
bg-multilingual	You are sure that the tweet is written in Bulgarian + another language.

Table 3: Annotation categories with their explanations.

experiment we removed URLs, emojis, hashtags (both the # sign and the entire word) and mentions, then we checked again and deleted newly appeared duplicates, and only then performed the testing experiment. We repeated the same process with the dataset from CLEF 2021.

In our manually annotated dataset we have two annotated categories - "bg" (Bulgarian) and "another". To calculate the accuracy, we first transformed our annotations into binary values. If the label is "bg" we assign 1, if the label is "another" we assign 0. Then we converted also the LI tools results into binary values. If the label is "Bulgarian" we assign 1, otherwise we assign 0. If the tool can detect more than one language we use/take only the first predicted label, or the label with the highest confidence score (usually the first one). Finally we use the binary values to make the calculation.

Unfortunately **spaCy** left some tweets without language labels. To compute its accuracy, we removed these tweets from both datasets.

In addition to the accuracy score, we also calculated precision, recall and F1-score. We obtained these scores for both datasets and their raw and cleaned versions.

4.3 Tested Language Identification Tools

All the tools that we tested support Bulgarian language and some of the other languages, written in Cyrillic alphabet, such as Russian, Macedonian, Ukrainian, etc. We have chosen these specific LI tools, because they are free (not paid), well known, and because some of them (e.g. Google Sheets's DETECTLANGUAGE function) are easy to use. While we are targeting readers, who are not interested to train these tools on their own data, we are providing enough technical details also for more technicallly-oriented users.

Without a doubt, one of the most famous and widely used tools is the **Google Translate API**. We found 2 libraries – **TextBlob** and **googletrans**. **TextBlob**[3] has a language detection function which uses Google Translate API, but currently they recommend to use instead the official API. The library **googletrans**[4] also implements Google Translate API. It uses the Google Translate Ajax API to make calls. The authors warn that this is an unofficial library and the maximum character limit on a single text is 15k. Also they cannot guarantee that the library will work properly at all times and recommend the use of the official API for more stability. When we first tested googletrans, it assigned language labels to part of the tweets. In the following tests it annotated all tweets with the tag "English". Due to the above mentioned limitations and the paid access we decided not to test Google Translate API further. However we tested another application from Google, which is free and has a language detection function (the Google Sheets DETECTLANGUAGE[5]). We refer to it from now on as **Google Sheets**.

fastText[6] (Joulin et al., 2016a,b) is developed by Facebook AI Research. It is a library for text classification and representation, which transforms text into continuous vectors that can be later used on any language-related task. fastText recognizes 176 languages and has been trained on data from Wikipedia, Tatoeba and SETimes. There are two models – a full version which is faster and more accurate, and a compressed version. The new line breaks were an issue for this tool and we had to

[3]https://textblob.readthedocs.io/en/dev/index.html. Last accessed on April 11, 2022.

[4]https://pypi.org/project/googletrans/. Last accessed on April 11, 2022.

[5]https://support.google.com/docs/answer/3093278?hl=en. Last accessed on March 10, 2022.

[6]https://fasttext.cc/docs/en/language-identification.html. Last accessed on April 11, 2022.

remove them in order to use it.

The next tool is **CLD3**[7]. It is a neural network model for language identification which uses character n-grams and calculates the fraction of times each of them appears. CLD3 supports 107 languages. We discovered that this is the only tool (among all of those that we tested), that has the very useful functionality for social media texts to recognize Bulgarian language written in Latin alphabet.

langdetect[8] is a direct port of Google's language-detection library from Java to Python. It supports 55 languages (including Bulgarian and other languages, written in Cyrillic alphabet). The original tool was trained on data from Wikipedia and tested on data from Google News or other news sites. The library *language-detection* uses Naive Bayes for classification. langdetect is fast and has good accuracy. This is the only tool that gave us an error when annotating a tweet which contains only emojis. The output is a list of the top languages that the model has predicted, along with their probabilities. When the probability of the prediction is less than 0.90, it usually adds more labels.

LangID[9] (Lui and Baldwin, 2012) is a fast language detection tool. It comes pre-trained on 97 languages and is not sensitive to domain-specific features (e.g HTML/XML markup). The model consists of a single .py file with minimal dependencies and can be deployed as a web service. The training data was collected from 5 different sources – JRC-Acquis, ClueWeb 09, Wikipedia, Reuters RCV2 and Debian i18n. Please, note that its confidence score is not normalised by default.

Another language detection tool is **polyglot**[10]. It depends on the pycld2 library which in turn depends on the cld2 library for detecting languages. This tool is suitable for mixed text messages. If the tweet contains phrases from different languages, the detector can find the most probable languages used in the text along with the confidence level. When there is not enough text to make a decision (e.g. a tweet containing only one word), the detector is forced to switch to the best effort strategy. Sometimes even using the best effort strategy, the detection is not reliable and an "Unknown Language" exception is thrown. In cases where the text contains characters that could belong to more than one language, this can be problematic. Polyglot can identify the languages supported by cld2 (up to 165). One of the problems with this tool was that our dataset contained some amount of short tweets and it wasn't very confident in its predictions.

The last tool we tested is **spaCy**[11]. It is a library for advanced Natural Language Processing. spaCy comes with pre-trained pipelines for over 60 languages, uses state-of-the-art speed and neural network models and a lot of features for language processing. It's open-source and easy to deploy. SpaCy has 2 modules with language detection capabilities: *spaCy-langdetect* and *spaCy-cld*. We used spaCy-cld for our research. This tool provides the most probable languages (up to 3) for the text. When the tweets are multilingual, these one to three hypotheses sometimes correspond to the various languages, present in the tweet, however we haven't tested its accuracy in predicting multiple languages within the same tweet. spaCy-cld also uses pycld2 and cld2. As both spaCy and polyglot use the same library, the results they gave were very similar. During our tests, we observed something interesting: the tool has left some tweets not tagged.

5 Results and Discussion

5.1 Results from the Manual Annotation

The two annotators disagreed on 31 out of 3124 tweets, which equals to 99.2% agreement between the annotators. We have additionally obtained a Cohen kappa value of 0.9691 for the Inter-Annotator Agreement (IAA) between the two annotators. The review done by the hyper-annotator has shown that both annotators did a few mistakes (probably from getting tired). Other specific cases in which they disagreed included:

- Very short tweets, composed of words, that exist in several languages (e.g. in Bulgarian and Russian: "настроение..." or "Логично и.....технологично.", "Лондон"). Translation in English: "mood...", "Logically and.....technologically.", "London".
- Cases due to the lack of extensive knowledge of the annotators in terms of Bulgarian dialects or other close languages (we cannot

[7] https://github.com/google/cld3. Accessed on April 11, 2022.

[8] https://pypi.org/project/langdetect/. Last accessed on April 11, 2022.

[9] https://github.com/saffsd/langid.py. Last accessed on April 10, 2022.

[10] https://polyglot.readthedocs.io/en/latest/index.html. Last accessed on April 11, 2022.

[11] https://spacy.io/. Last accessed on April 11, 2022.

	Raw data				Clean data			
Tools	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**
fastText	**0.96**	**0.99**	**0.97**	**0.98**	**0.97**	**0.99**	**0.98**	**0.98**
CLD3	0.93	0.96	0.96	0.96	0.94	0.97	0.97	0.97
langdetect	0.93	0.96	0.96	0.96	0.93	0.96	0.96	0.96
LangID	0.90	0.98	0.90	0.94	0.91	0.97	0.91	0.94
polyglot	0.90	**0.99**	0.89	0.94	0.91	**0.99**	0.90	0.94
Google Sheets	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**1.00**	**0.99**
spaCy*	0.89	0.99	0.87	0.93	0.90	0.99	0.88	0.93

Table 4: Results of the tests performed on our dataset.

	Raw data				Clean data			
Tools	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**
fastText	**0.98**	**1.00**	**0.98**	**0.99**	**0.99**	**1.00**	**0.99**	**0.99**
CLD3	**0.98**	**1.00**	**0.98**	**0.99**	**0.98**	**1.00**	0.98	**0.99**
langdetect	**0.98**	**1.00**	**0.98**	**0.99**	**0.98**	**1.00**	0.98	**0.99**
LangID	0.93	**1.00**	0.93	0.96	0.93	**1.00**	0.93	0.96
polyglot	0.91	**1.00**	0.91	0.96	0.93	**1.00**	0.93	0.96
Google Sheets	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
spaCy*	0.91	1.00	0.91	0.95	0.94	1.00	0.94	0.97

Table 5: Results of the tests performed on CLEF 2021 dataset.

share examples due to Twitter's data sharing restrictions).

- Tweets in English, but transliterated in Cyrillic letters, e.g. "Толд йа соу"("Told ya so").

See the following Section 5.2 on how the LI tools dealt with such short and ambiguous tweets.

5.2 Results from Testing the LI Tools

The final results can be seen in Tables 4 and 5, where "F1" stands for F1-Score. Undoubtedly, the best performing language identification tool is Google Sheets, which was a surprise for us. The second best performing tool is fastText. However, it is difficult to make a ranking because each tool has its advantages and disadvantages.

One of the first problems that we noticed while executing the code is that fastText gives an error if the text of the tweet is not in one line. We had to remove all the new line symbols before using the tool. The other tools had no problem with that. The next tool that gave us an error was langdetect. We had to remove tweets that only contained emojis or replace the emojis with some text so that the tool can annotate the data. The other tools did not give emoji-caused errors during code execution, but some of them did not annotate such tweets (spaCy), some labeled them as "unknown" or "undefined" (polyglot and Google Sheets), and some labeled them as if they were normal text (fastText, CLD3 and langID). Therefore, we removed from our dataset 2 tweets that contained only emojis.
Another problem that we encountered is that spaCy did not assign language labels to some of the tweets. We tried to understand why this was happening, but we couldn't. For our dataset, the tool did not annotate 96 posts (raw data). The length of these tweets varied between 1 and 29 words (average word length - 6.49), most of the unannotated tweets were 6 words long. The number of cleaned unannotated tweets increased to 200, their length was 1-26 words (with an average length of 7.36). Again, most of the unannotated tweets were 1 word long. We checked if all the 1-word-long tweets were not annotated, but it turned out that some of 1-word-long tweets were annotated. For CLEF2021 datasets, the unannotated tweets were fewer - 32 (raw data) and 114 (clean data). Again, we observe an increase in the number of tweets not annotated by spaCy after cleaning the data. We hypothesize that this might be due to the fact that during "cleaning" whole words (hashtags and mentions) were removed. In the raw data, the length of the tweets varied between 5 and 26 words (average word length - 12.69), with most of the unannotated tweets being

9 words long. The length of the cleaned unannotated tweets in CLEF2021 was 4-42 words (with an average length of 11.61). The highest number of unannotated tweets had a length of 7 words (for raw data) and 12 words (for cleaned data). We looked at the text of the unannotated tweets of the raw datasets, but we could not find the reason (for example, they may have contained only hashtags or code-switching), but the texts were very diverse. As spaCy does not label all the data, its results are separated from the results of the other tools in Tables 4 and 5.

All tools, in addition to language, also provide data on accuracy or a confidence score. However, only 2 of the tools output more than one language label - langdetect and spaCy. It is not described in detail on what principle they put these labels, but we noticed that they usually put several labels if they have detected more than one language in the text or arrange the languages according to accuracy. In our dataset we had only one post in Bulgarian–Latin, which was labeled correctly by CLD3. CLD3 is also the tool that provides the most detailed output.

We tested with which languages the tools most often confuse tweets, written in Bulgarian. When making mistakes, the tools most frequently tag Bulgarian tweets as Macedonian (mk) (see Table 6 for the most common mistakes of the tools when tagging Bulgarian tweets). Some of the tools tag Bulgarian (bg) tweets as Russian (ru) or Serbian (sr). These errors may be due to the amount of data in these languages in the datasets used to train them. We assume that when training fastText, the largest amount of data was in Russian. Respectively, the largest amount of data for polyglot and spaCy was probably in Serbian.

Regarding the very short tweets, which the human annotators struggled with (see the end of Section 5.1), surprisingly, the LI tools correctly recognized the language, even if they had access only to the text of the tweet. As the investigation of the hyper-annotator showed that most of these tweets were written in Russian, our hypothesis was that the tools have been pre-trained on much larger amounts of Russian texts. Further investigation of this issue is necessary.

In terms of speed, all the tools did quite fast in labelling all datasets. FastText, CLD3 and polyglot annotated the tweets in less than 5 seconds, and langID annotated data in about 10 seconds. The rest of the tools were slower, but the annotation time remains less than 1 minute. It takes spaCy about 40 seconds to annotate the data, and langdetect about 30.

	Covid-19		CLEF 2021	
Tools	**Raw**	**Clean**	**Raw**	**Clean**
fastText	ru	ru	ru	ru
CLD3	mk	mk	sr	mk
langdetect	mk	mk	mk	mk
LangID	mk	mk	mk	mk
polyglot	sr	sr	sr	sr
Google Sheets	mk	mk	mk	sr
spaCy	sr	sr	sr	sr

Table 6: The most common mistakes of the LI tools when providing language labels to tweets, written in Bulgarian.

6 Conclusions

In this article we have presented the results from comparing 7 well-known off-the-shelf Language Identification (LI) tools on identifying Bulgarian language posts in two Twitter datasets, composed of around 3000 tweets each. We provided a presentation of each tool along with its useful functionalities and eventual shortcomings. We are confident that this information will be of use to any researchers, who would like to know the performance of off-the-shelf LI tools on Bulgarian social media posts, without training them.

Our results show that the tool which has the highest scores is the **DETECTLANGUAGE()** functionality of Google Sheets. The second best is **fastText**. We have found out that CLD3 has also the functionality to recognize Bulgarian, written with Latin letters, which is useful for social media and Internet forums texts. Testing its performance for this task has still to be done. We have also discovered that **polyglot** and (partially) **spaCy** can be used to guess multiple languages, present within the same text, but their performance in executing this task needs to be properly tested too.

We haven't discovered any LI tool, which simultaneously has a high Accuracy/F1-Score, can recognize Bulgarian written with Latin letters, and recognizes the languages in multi-lingual posts. This presents an opportunity for creating such a tool.

As future work, we plan to evaluate in more detail the above mentioned functionalities of polyglot, spaCy, and CLD3, and also to implement our own

LI tool.

7 Acknowledgements

The work, presented in this article has been supported by the project GATE (funded by Operational Programme Science and Education for Smart Growth under Grant Agreement No. BG05M2OP001-1.003-0002- C01) and is part of the research project TRACES[12], which has indirectly received funding from the European Union's Horizon 2020 research and innovation action programme, via the AI4Media Open Call 1, issued and executed under the AI4Media project (Grant Agreement no. 951911).

References

Kheireddine Abainia, Siham Ouamour, and Halim Sayoud. 2016. Effective language identification of forum texts based on statistical approaches. *Information Processing & Management*, 52(4):491–512.

Jannis Androutsopoulos. 2013. 27. Code-switching in computer-mediated communication. *Pragmatics of computer-mediated communication*, page 667.

Timothy Baldwin and Marco Lui. 2010. Multilingual language identification: ALTW 2010 shared task data. In *Proceedings of the Australasian Language Technology Association Workshop 2010*, pages 4–7.

Boris Bankov et al. 2017. Extracting top trends from Twitter discussions in Bulgarian. *Izvestia Journal of the Union of Scientists-Varna. Economic Sciences Series*, (2):254–259.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific Twitter collections. In *Proceedings of the second workshop on language in social media*, pages 65–74.

Atefeh Farzindar and Diana Inkpen. 2017. Natural language processing for social media. *Synthesis Lectures on Human Language Technologies*, 10(2):1–195.

Archana Garg, Vishal Gupta, and Manish Jindal. 2014. A survey of language identification techniques and applications. *Journal of Emerging Technologies in Web Intelligence*, 6(4):388–400.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

[12]https://traces.gate-ai.eu/

Tommi Sakari Jauhiainen, Bo Krister Johan Linden, Heidi Annika Jauhiainen, et al. 2017. Evaluation of language identification methods using 285 languages. In *21st Nordic Conference of Computational Linguistics Proceedings of the Conference*. Linköping University Electronic Press.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of Twitter messages. In *Proceedings of the 5th workshop on language analysis for social media (LASM)*, pages 17–25.

Shervin Malmasi. 2017. Open-set language identification. *arXiv preprint arXiv:1707.04817*.

Shaden Shaar, Maram Hasanain, Bayan Hamdan, Zien Sheikh Ali, Fatima Haouari, Mucahid Kutlu Alex Nikolov, Firoj Alam Yavuz Selim Kartal, Giovanni Da San Martino, Alberto Barrón-Cedeño, Rubén Míguez, Tamer Elsayed, and Preslav Nakov. 2021. Overview of the CLEF-2021 CheckThat! lab task 1 on check-worthiness estimation in tweets and political debates. In *Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum*, CLEF '2021, Bucharest, Romania (online).

Martin Thoma. 2018. The WiLI benchmark dataset for written language identification. *arXiv preprint arXiv:1801.07779*.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL shared task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, pages 1–9.

Appendix A Tool used for manual language annotation

As described in the article, we have used Google Spreadsheets for manually annotating the languages of social media posts. Figure 1 shows the annotation spreadsheet with fall-down menu, containing the annotation categories. The examples of tweets are mock ones, due to Twitter's restrictions on sharing their data.

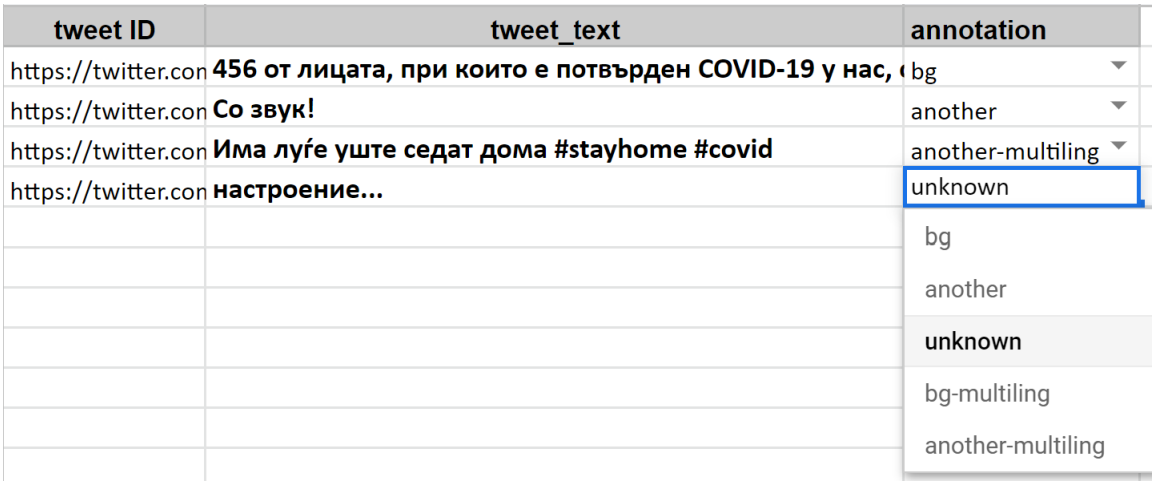

Figure 1: Annotation spreadsheet with fall-down menu.