An Assessment of Language Identification Methods on Tweets and Wikipedia Articles

Pedro V. Gonçalves
CDTec
UFPel – Brazil
pvgoncalves@inf.ufpel.edu.br

Larissa A. de Freitas
CDTec
UFPel – Brazil
larissa@inf.ufpel.edu.br

Abstract

Language identification is the task of determining the language which a given text is written. This task is important for Natural Language Processing and Information Retrieval activities. Two popular approaches for language identification are the N-grams and stopwords models. In this paper, these two models were tested on different types of documents such as short, irregular texts (tweets) and long, regular texts (Wikipedia articles).

1 Introduction

The language identification task is a very important step in many Natural Language Processing (NLP) pipelines and Information Retrieval (IR) systems. Text processing techniques developed in NLP and IR generally pre-suppose that the language of the input text is known, and many techniques assume that all documents are in the same language (Jauhiainen et al., 2019). To identify the language of a text two approaches are commonly used: the N-grams (typically trigrams) (Zubiaga et al., 2014; Grag et al., 2014) and the stopwords models (Truica et al., 2015).

The first approach consists of tokenizing the text into its N-grams, then checking the language(s), one or more, for which each N-gram is particularly common, adding to the chances of each of those languages of being the one to be identified.

The second approach consists of tokenizing the text into what seems to be words, then, for each known stopword found among the tokens, add to the chances of the related language(s) of being the one to be identified.

This paper aims to compare the results obtained with each of those methods for two distinct types of linguistic content, which are encyclopedia articles collected from Wikipedia and short messages (tweets) collected from Twitter.

For such comparison, both models have been implemented in C++ as a single command-line tool.

2 The Models

Both language identification models covered by this work are based on language statistics. That essentially means they work by comparing information about the frequencies of some linguistic elements in a given text with previously known information, in turn, extracted from multiple corpora whose language is known.

2.1 N-grams Model

The N-grams model uses the fact that for each written language, there are some contiguous sequences of words which, particularly for that language, occur much more frequently than others. Based on that, one can extract a list of the most frequent sequences of N-grams from a text and compare it to previously known lists. Those sequences are called N-grams where, usually, $2 \leq N \leq 4$ and words mean, actually, graphemes, which are the smallest units of a writing system.

N-grams are hardly exclusive to one single language. As a consequence of that, the N-grams model tends lower confidence scores and smaller differences among confidence scores for each identified language. On the other hand, since any piece of writing contains at least a few N-grams and even a mis-

spelled word may also contain some recognizable ones, this model can work even for short and irregular inputs, such as tweets.

2.2 Stopwords Model

The stopwords model, on the other hand, uses the fact that for each language, there are some words use to occur very frequently only for that language, while being virtually absent in others. Those words are usually called stopwords in the context of NLP and consist, in most cases, of things such as articles, personal pronouns, prepositions, conjunctions and, auxiliary verbs. Even among languages with high degrees of similarity and shared vocabulary, like those spoken in Scandinavia or the Iberian Peninsula, for example, the spelling of such words still differs.

As opposed to the N-grams model, the one based on stopwords gets unsuitable and inaccurate as inputs get small since smaller pieces of text mean fewer words and, consequently, less stopwords. Also, given that language identification usually occurs early in NLP pipelines, before complex things such as spell checking, and that misspelled words tend not to match the ones extracted from the corpora, this model gets unsuitable as inputs get less regular.

Stopwords models show higher confidence scores and tends to overcome or at least catch up with the accuracy of other approaches for bigger and well-written inputs.

2.3 Comparing the models

The N-grams approach is a more universal one, keeping good results with all types of input, while the stopwords one loses its ability to deliver correct results as inputs get smaller. The stopwords model, however, is simpler and, therefore, faster, given that it involves fewer comparisons than the N-grams model.

3 Datasets

Datasets[1] used for the tests were automatically collected and built using publicly available Python modules. For the Wikipedia articles, the Python API[2] works in such a way that one has to set the desired language before downloading the articles. Tweets, on the other hand, can not be fetched by language, so they were collected from users known to speak the desired language and then classified using a popular Python module for language identification[3]. Tweets were stored and made available as lines of normalized Unicode text in one file for each language, while Wikipedia articles were stored as separate files named after their respective languages.

In total, 130505 tweets and 26570 Wikipedia articles have been collected to the two datasets, with counts by language as listed in **Tables 1** and **2**.

Table 1: Results for the N-grams and stopwords Models on Tweets, by language.

			N-grams Model		**Stopwords Model**	
Language	**ISO 639-2B code**	**Count**	**Acc.**	**Avg. Confidence**	**Acc.**	**Avg. Confidence**
TOTAL	-	-	90%	67%	87%	79%
English	eng	52532	91%	64%	92%	81%
Portuguese	por	71849	88%	68%	84%	79%
Spanish	spa	6124	92%	69%	86%	76%

4 Results

Accuracy (Acc.) and Average Confidence (Avg. Confidence) metrics have been calculated by language, for each dataset and model (**Tables 1** and **2**). Those values have been generated by running the imple-

[1] Datasets are available at https://github.com/pedrovernetti/nlp-datasets

[2] Wikipedia Python module: https://pypi.org/project/wikipedia/

[3] Polyglot Python module: https://pypi.org/project/polyglot/

Table 2: Results for the N-grams and stopwords Models on Wikipedia Articles, by language.

			N-grams Model		Stopwords Model	
Language	**ISO 639-2B code**	**Count**	**Acc.**	**Avg. Confidence**	**Acc.**	**Avg. Confidence**
TOTAL	-	-	97%	69%	95%	90%
Arabic	ara	1574	100%	86%	97%	92%
Danish	dan	1555	98%	69%	94%	89%
Dutch	dut	1566	98%	70%	99%	96%
English	eng	1604	97%	61%	100%	94%
Finnish	fin	1551	98%	64%	92%	83%
French	fre	1528	98%	67%	100%	91%
German	ger	1574	99%	70%	99%	93%
Irish	gle	1607	95%	74%	91%	93%
Greek	gre	1578	99%	91%	99%	94%
Hungarian	hun	1533	96%	60%	87%	83%
Italian	ita	1548	98%	66%	96%	88%
Portuguese	por	1598	91%	65%	93%	87%
Romanian	rum	1571	94%	66%	98%	94%
Spanish	spa	1547	99%	68%	99%	94%
Swedish	swe	1533	98%	71%	99%	89%
Turkish	tur	1529	95%	65%	80%	64%
Vietnamese	vie	1574	97%	63%	100%	97%

mented tool[4].

Given each language identification process results in a list of language-confidence pairs, what was taken as the final result was the language for which the confidence value was the highest. The measured Acc. stands here for how many times, out of the total run tests, the final results matched the actual language.

The best results obtained in this task are from the 92% Acc. reported for irregular texts (tweets), or the 100% Acc. reported for regular text (Wikipedia articles) from the N-gram and stopwords models.

There are still many aspects of this topic for one to focus on, such as the effects of the number of supported languages on the confidence and accuracy of the results. Experimenting with some common problems involving similarities between closely related languages and their possible workarounds could deliver new results to be analyzed. Assessments on the performance and time efficiency of each model could also be of interest.

References

Zubiaga, Arkaitz and San Vicente, Iñaki and Gamallo, Pablo and Pichel, José R. and Alegria, Iñaki and Aranberri, Nora and Ezeiza, Aitzol and Fresno, Víctor. 2014. Overview of TweetLID: Tweet Language Identification at SEPLN 2014. *Twitter Language Identification Workshop at SEPLN 2014*, 1–11.

Garg, Archana and Gupta, Vishal and Jindal, Manish. 2014. A Survey of Language Identification Techniques and Applications. *Journal of the Emerging Technologies in Web Intelligence*, 6(4):388–400.

Jauhiainen, Tommi and Lui, Marco and Zampieri, Marcos and Baldwin, Timothy and Lindén, Krister. 2019. Automatic Language Identification in Texts: A Survey. *Journal of the Artificial Intelligence Research*, 65:675–782.

Truica, Ciprian-Octavian and Velcin, Julien and Boicea, Alexandru. 2015. Automatic Language Identification for Romance Languages Using Stop Words and Diacritics. *17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 243–246.

[4]Codes are available at https://github.com/pedrovernetti/omnglot