Language Discrimination

Karel van Wayenburg

March 2021

1 Introduction

Languages are diverse, often far more diverse than most people realise. Currently about 7.100 distinct living languages are thought to exist [1]. These languages use a variety of different symbols to convey meaning. However, knowing only the script a language is written in gives far from enough information to correctly identify a language, since many scripts are shared across multiple languages. Since natural languages are constantly changing, rule-based systems for determining what language a text is will only remain effective for a limited timespan. Because of this, a machine learning model that can automatically identify languages is much more useful and can easily be applied to new language data to adapt to changes in natural languages. This is the type of model we are trying to create.

2 Background Theory

2.1 UTF-8

UTF-8 is an encoding that can encode any of all the 143,859 [3] Unicode characters, or more specifically, it encodes their code points. These code points are the numbers assigned to specific characters by the Unicode consortium[**bron**]. The code point of a unicode character is often denoted as U+nnnn, where nnnn is the hexadecimal number associated with the charcter. The UTF-8 encoding does this by encoding all valid code points into a sequence of 4 or fewer bytes. The first few bits of a byte in this encoding are used to specify the relation of that byte to the bytes surrounding it, so 4 1-byte characters can be distinguished from for 1 4-byte character etc. The remaining bits are then used to denote the actual code point the character.

2.2 Byte n-grams

A model such as a word n-gram is one that might come to mind for a classification such as this one. It is useful for classifications such as sentiment classification so why not here? There are several reasons.

The major problem of word n-grams in this case is that it assumes that words are easily separable, like is the case in English, where they are separated by space characters. This is far from the truth. Many languages do not have this convention of using spaces which might make it harder to separate words. Another problem is the amount of word forms. In agglutinative languages such as Finnish and Japanese, many small meaning segments can be added to words to change their meaning resulting in a very sparse frequency data. Finally another problem is the size of the model due to the vocabulary size of other languages. Word n-grams for a single language already have a space complexity of $O(v^n)$, and taking a such an n-gram over l languages would give a complexity of $O((lv)^n)$ where v is the average vocabulary size. Such a size would lead to very sparse data, which would make it not very generalisable.

An alternative way to obtain information about character (co)-occurence in a document is using byte n-grams. These models work similarly to word n-grams, but instead of modelling the the frequency of words, they model the frequency of bytes near each other. This means

It might seem like a good idea to take the intermediate approach and create a character n-gram for all characters that occur next to each other, but rarely any literature mentions such models, possibly due to the fact that unicode contains, at the time of writing, 143,859 characters , which would result in large, sparse n-grams. This requires further research however.

2.3 WiLI dataset

The WiLI dataset is a dataset of 1000 paragraphs in 235 different languages. This dataset is specifically made for monolingual written natural language identification.

3 Methods

We have opted for a logistic regression model that uses uni-, bi- and trigram counts as features. We also plan to engineer our own features based on Unicode blocks. We trained this model on the training set of the WiLI-2018[2] dataset.

4 Results Discussion

4.1 Results

For our first results, we used a small subset of 7500 paragraphgs of the WiLI dataset, on which we applied a logistic regression, using only the byte unigram counts as a feature. On a devset of 2500 paragraph our converged model had a macro-averaged F1-score of 0.82.

4.2 Discussion

The WiLI dataset is a monolingual one. Monolingual datasets are useful, but they may not generalise to all language segments, such as wikipedia articles or online conversations, where fragments of different languages might be used alongside the "main" language of the text. This makes the model trained on this data less generalisable.

References

[1] *Ethnologue*. URL: https://www.ethnologue.com/. (accessed: 07.03.2021).

[2] Martin Thoma. "The WiLI benchmark dataset for written language identification". In: *CoRR* abs/1801.07779 (2018). arXiv: 1801.07779. URL: http://arxiv.org/abs/1801.07779.

[3] *Unicode Statistics*. URL: https://www.unicode.org/versions/stats/chart_charbyyear.html. (accessed: 07.03.2021).