N-Gram Language Identification

Joseph Burns

University of Colorado
Colorado Springs
jburns6@uccs.edu

Abstract

This document serves as a demonstration as to how to classify language through the use of N-Grams. Language can be broken down into components that can be used to properly represent a language. This model has shown to be accurate between 89.99% and 98.80% on classifying up to 235 different languages.

1 Introduction

The importance of identifying a document's language is often overlooked, even though it is used in a myriad of ways. Search engine usage for translation, machine translation, sentiment analysis, text summarization and further have a critical stake in language identification (Thoma, 2018).

Breaking down a language into distinct elements is a highly effective way to build a languages "profile" to refer to when determining a texts language. Once the language has been broken down into distinct elements, there is a need to gain a frequency of use of the elements in each language, as many languages may have very similar elements. By using the most frequently used elements associated with a given language, the language's profile is built, which can then be used to test against other profiles, unknown and known, to determine their relationship.

Determining what those distinct elements are is the first question. A common method, and the one discussed in this paper, is the use of N-Grams.

2 N-Grams

N-Grams are all contiguous combinations of characters of a specific string, with spaces added before the beginning and after the end of a string (Cavnar et al., 1994). The "N" in N-Gram represents the length of the contiguous slice of the string.

For example, if the string was PROFILE:

- 5-Gram: _PROF, PROFI, ROFIL, OFILE, FILE_
- 4-Gram: _PRO, PROF, ROFI, OFIL, FILE, ILE_
- Tri-Gram: _PR, PRO, ROF, OFI, FIL, ILE, LE_
- Bi-Gram: _P, PR, RO, OF, FI, IL, LE, E_
- Uni-Gram: _, P, R, O, F, I, L, E, _

Seeing as we now have a method for breaking down a string/language into its distinct elements, how to organize them in a meaningful manner as to properly represent any given language becomes the next priority in the proposed method.

3 Generating Language Profile from N-Gram Frequencies

A common theme to all written language is that sentences are formed up of a combination of words, and words generated from a combination of characters of the language's alphabet. Not all languages have the same alphabet or words but the composition is the same.

Using this knowledge, our method parses a line of the given text from the WiLI Benchmark Dataset into the words it is composed of after removing punctuation and numbers, then splits those into the all N-Gram combinations, where N is all integers from 1 to 5. While getting this list of N-Grams, we keep track of their number of occurrences in the line. For the training of the model, we repeated the process for all lines of the same language, gaining a list of N-Grams for the language. The list is sorted from most frequent/common N-Grams to least.

With this list, we chose to take the 300 most frequent N-Grams, as a proper representation of the

line. 300 were chosen because we did not want to narrow the language by choosing too few, as well as not choosing too many, potentially causing overlap into different languages. These 300 of the most frequent N-Grams are used as the language profile, a representation of the languages most common letters, as well as their most common prefixes and suffixes.

To recognize what language a text is written in, we must then determine the most accurate way to compare an unknown text's profile, with each of the trained language profiles.

4 Out-Of-Place Measure

The Out-of-Place measure is a method of comparing said profiles, adopted from Cavnar et al.. The measure takes each N-Gram from the unknown texts' profile and locates its placement in the language profile it is tested against. Based on its' position in the unknown profile, a distance measure for that N-Gram is obtained by getting the difference of positions. If an N-Gram is not in the tested languages profile, a max distance value is given (in our case, 300), and all of the N-Gram distance values are summed to gain an overall distance measure for the language the text was compared against. This allows us to guess the unknown language based on the lowest distance value. In Figure 1, an example of the process is given from Cavnar et al. (1994).

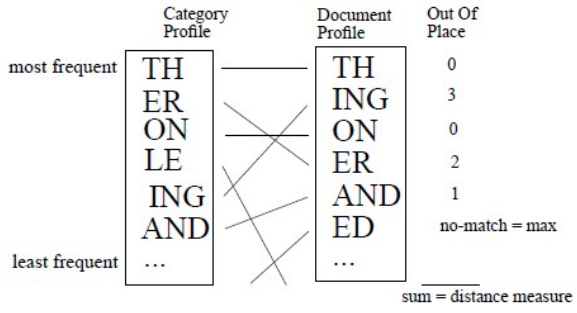

Figure 1: Example of the Out-Of-Place Measure

5 Results

As stated earlier, the data set used for training the language profiles, as well as to test against those profiles is the WiLi- 2018 - Wikipedia Language Identification data set (Thoma, 2018). Using this data set, we are given 500 paragraphs (1 per line) for 235 different languages to train the language profiles, and another 500 paragraphs for each language to test against.

Starting with only 2 languages, English and Spanish, we obtained an average accuracy of 98.8%, getting 500 out of 500 English paragraphs correct, and 488 out of 500 Spanish. The whole test running in under 30 seconds.

Testing on an increased amount of languages (English, Spanish, French, Portuguese, German, Dutch, Assamese, and Thai) increased the time the testing ran to just under 2 minutes, but still obtained a respectable accuracy of 97.48%.

Given the good results above, we increased the testing to include all 235 languages provided in the data set. The run time was increased to almost 10 hours, but the average accuracy was still high, obtaining a 89.99% accuracy for the entire data set.

6 Difficulties

There were many difficulties surrounding our task of language classification, most of which have been fought through and passed. The biggest obstacle that we had to overcome was our initial accuracy ratings. The first few tried methods only obtained about an 8.23% accuracy. After further examination, we found that this was due to many reasons, one of which was that the distance score was not being computed properly.

The primary reason for low accuracy was found when we were still only achieving an accuracy of 23.6%. We found that we were trying to generate N-Grams from too large of a string. A sentence from any language can be composed of any number of words in any order. Instead of trying to form N-Grams for each word, we were forming N-Grams for the entire paragraph. Which would be okay if the provided training set was larger, as we were only capturing 1 specific combination of words. By changing the N-Grams to word length + 2, allows the training set to be smaller because there are less combinations of characters to form words than there are words to form sentences or paragraphs, thus capturing the language more appropriately.

There was also a reason for choosing only the top 300 N-Grams for the language profile: time efficiency. The N-Gram method can and will produce well over the 300 we limited it to, but after around the 300 mark or so, the N-Grams become more specific to the topic of the document rather than being just the language-specific N-Grams we

are wanting to classify a language under (Cavnar et al., 1994).

7 Conclusion

Using this N-Gram method to identify what language a text is written in is very useful as the results show, at least for fewer languages. This may be due to some languages that share many common components, such as some of the more common Latin based languages. This algorithm may not be able to detect the minor differences between such languages, thus causing more inaccurate results. A larger testing paragraph may help cancel this shortcoming, as well as a larger data set, rather than just Wikipedia entries.

Acknowledgments

This algorithm and method of testing is an implementation of the method and algorithm put forth by Cavnar et al. on a different data set. Further assistance can be accredited to Tom Conley, a lecturer in the Computer Science department at the University of Colorado Colorado Springs, who helped walk through logic flaws and help with language processing understanding.

References

William B Cavnar, John M Trenkle, et al. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175. Citeseer.

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