COMP30027 Project 2: Language Identification

Kara La'Brooy
the_unasked_upturn

Abstract

Language classification has never been more important than in this online era. Twitter has an Alexa rank of 17 [April 2017] – it is one of the top social media websites in the world - and we will mainly use its data in the form of tweets in developing a language classification system.

1 Introduction

Text Classification is the classification of documents in to predefined classes. Language Identification is the task of determining the natural language of a source. It is subdomain of text categorisation and can be solved by statistical methods. This paper follows the development of a language identification classifier focussing on - but not limited to - Twitter posts.

Humans perform manual language identification extremely well, despite not necessarily being proficient in the given language or if the amount of text is small. Upon observation, at a word, or even character level, our educated options are reduced dramatically, making it. For example, Japanese, Chinese, Korean characters do not occur in any European language, and similarly Latin and Greek diacritics and accents found in the majority of European languages, do not occur in Japanese, Chinese or Korean text. Similar to a human deciding on the correct foreign language of a text, an automated language identification of makes use of character n-grams frequencies rather than more linguistically refined or semantic methods.

A person fluent in a particular language would have a deeper understanding of the phonetics, phonology, morphology and syntax of that language. It is these differences in languages that make it possible for them to be classified. And it is translation that motivates the need to classify languages.

2 Naïve Bayes'

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Bayes' rule gives us a model of the world described by the unknown *h*, and we observe the data *D*. The probability model P(*D*|*h*) explains that given *h*, it will tell us what data to expect. We must also have prior belief to what P(*h*) is. The Bayesian analysis answers the question: Given the data, what do we know about *h*?[1]

D represents the twitter data and *h* represents what class they belong to – each data instance belongs to one class.

Naïve Bayes' is a classification model based on probability theory. Naive Bayes' uses Bayes' theorem with two simplifying assumptions: the independence between the attributes and position independence of the attributes.'[2]

$$\text{posterior } P(c_j|x_1, x_2, \ldots, x_n) = \frac{likelihood * prior}{evidence}$$

3 Approach

To complete the Language Identification task, we have been supplied with a training set, a development set, and a training set.

During the training phase, a language library is created for each of the 20 predefined languages and objects are also created to help with the Naïve Bayes calculations later. The

library acts as a dictionary of n-grams, to be scanned and a similarity to the text to be measured.

During the testing phase, each text instance is compared against these libraries to determine the most probable language.

The accuracy of the classifier is improved with more training data. More correct data input into each of the language libraries during the training renders it more complete and a more detailed and reliable reference point for a given language.

3.1 Variable Tweaking

n-gram level

Character level n-grams are preferred over word level n-grams. Word level n-grams are too restrictive and are only valid for longer texts. Character level n-grams are robust to spelling errors – most of the n-grams generated by an incorrectly spelt word will still be the same as the original word, especially in the case of a transposition error. However, in word based n-grams this would create an entirely different, useless and incorrect word for the language library. The dimensionality is also increased[1] exponentially for word based n-grams. There are 171,476 words in current use in the English Language contrasted with only 26 letters in the English alphabet.

Value of n

The value of n for the character level n-grams is a trade-off between low n with high frequency and less information and high n with lower frequency and more information.

Threshold for Unknown language tag

The Naive Bayes classification method tends to output probabilities close to the edges 0 and 1. The threshold was increased until the accuracy increase was negligible – threshold = .95

Stripping characters

Stripping characters from the text before creating n-grams resulted in a small increase in classifier performance. Prior to this the top n-gram for "en" - english – was " ", an extremely uninformative n-gram for classifying a language. These whitespace dominant n-grams are best removed and replaced, motivating the text stripping method.

3.2 Accuracy

The Naaive Bayes classification model, fed by character-level n-grams performed best with n = 5, threshold = 0.95. With these parameters an accuracy of ~88.37% was achieved with training the system with the provided training.json file and testing the system with the provided test.json file.

The following graph represents the change in accuracy of the model with varying n

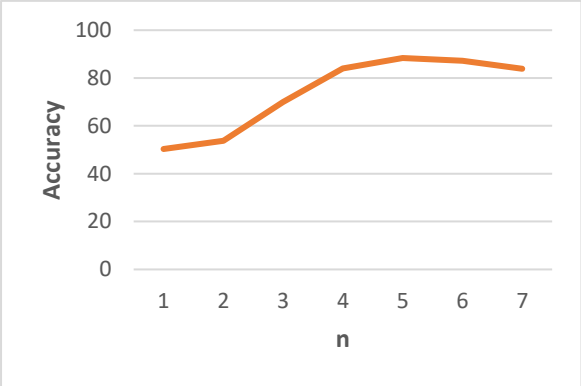

Figure 3.1.1

As seen from Figure 3.1.1 the accuracy of the model peaks at n = 5.

4 Conclusion

Despite the nature of language appearing extremely complex, a human is an efficient at language categorization. Likewise, an automatic system to classify language is extremely efficient when training over a decent dataset and breaking the seemingly daunting task into smaller probabilistic ones.

The simple legacy methods of Naïve Bayes trained by an n-gram corpus achieves an accuracy of 88.37% over mainly Twitter based text sources.

References

1. K. F. Mohamed. 2015. Using Naive Bayes and N-Gram for Document Classification