Similar Language Detection

Naveen B(140050034), Chanukya (140050043), Uday K(140050048),
Raghuveer U(140050050)
Guide: Ganesh Ramakrishnan.

IIT Bombay

02/04/2017

Outline

- Overview
- Project Statement
- Approach
- Stage 1
- Stage 2
- Research Papers
- Data Sets

Overview

- Discriminating between disparate languages has been a trivial problem, but discriminating similar languages and dialects is not trivial and in fact is far from solved.
- This project deals with problem of discriminating between similar languages like Argentine Spanish, Peninsular Spanish.

Project Statement

- Given an input sentence the aim is to predict which of the closely related languages the corresponding input belongs to.
- The training data consists of sentences labeled with 13 languages which are Bulgarian (bg), Macedonian (mk), Bosnian (bs), Croatian (hr), Serbian (sr), Czech (cz), Slovak (sk), Spanish (esar), Peninsular Spanish (eses), Brazilian Portuguese (ptbr), European Portuguese (ptpt), Indonesian (id), Malay (my).

Approach

- We are implementing a hierarchical method which first divides the data into groups and then labels the sentence.
- We initially use the word frequencies to classify the languages into groups that are very similar with themselves but the degree of similarity between the groups is less. For this classification we use a cosine similarity classifier based on word frequencies.
- We then use Multi-class SVMs, by deriving dual problem(Stage 2).

Stage 1

- From the training data, for each language we first parse through the sentences and find the frequencies of all the words appearing in all the sentences.
- Among all the words and their corresponding frequencies generated for each language, we choose the top 1000 words based on their frequency values.
- Define x^l for each language l as vector $\in R^{1000}$, x^l = 1000-tuple of frequency of the chosen words.

Stage 1

- Now given an input statement s we define $x^l(s)$ for language l as follows
- $x^l(s) \in R^{1000}$ first intialised to all zeros
- $x^l_i(s) = 1$ if the i th word of the l th language is present in the input sentence.

Stage 1

- Label for statement s is label(s) = $argmax_{l \in 1 \ldots 13}$(dotproduct of $x^l(s)$, x^l).
- The intuition is clear.

Stage 1 Results

- The confusion Matrix generated after running the code on training data(18000 samples for each language) is as follows:
-

$$C = \begin{array}{c|cccccccccccc} - & bg & bs & cz & esar & eses & hr & id & mk & my & ptbr & ptbt & sk & sr \\ bg & 17379 & 0 & 0 & 0 & 0 & 0 & 0 & 620 & 0 & 0 & 0 & 0 & 1 \\ bs & 0 & 1232 & 34 & 0 & 48 & 288 & 0 & 0 & 0 & 51 & 22 & 25 & 16299 \\ cz & 0 & 396 & 12486 & 0 & 199 & 128 & 0 & 0 & 1 & 1115 & 411 & 1843 & 1421 \\ esar & 1 & 0 & 0 & 0 & 17999 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ eses & 2 & 0 & 0 & 0 & 17998 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ hr & 0 & 1872 & 57 & 0 & 73 & 1146 & 0 & 0 & 0 & 92 & 31 & 35 & 14694 \\ id & 2 & 0 & 1 & 0 & 29 & 0 & 14879 & 0 & 3077 & 7 & 4 & 0 & 1 \\ mk & 36 & 1 & 1 & 0 & 0 & 0 & 0 & 17961 & 0 & 0 & 0 & 0 & 0 \\ my & 0 & 0 & 9 & 0 & 25 & 0 & 2227 & 0 & 15715 & 1 & 1 & 2 & 2 \\ ptbr & 1 & 0 & 0 & 0 & 9819 & 0 & 0 & 0 & 0 & 7557 & 622 & 0 & 1 \\ ptbt & 1 & 0 & 0 & 0 & 10879 & 0 & 0 & 0 & 0 & 5822 & 1297 & 0 & 1 \\ sk & 0 & 75 & 151 & 0 & 62 & 49 & 0 & 0 & 0 & 943 & 349 & 15769 & 602 \\ sr & 0 & 214 & 5 & 0 & 59 & 38 & 0 & 0 & 0 & 38 & 12 & 29 & 17605 \end{array} \quad (1)$$

- This will be used for dividing the languages into groups.

Stage 2

- The feature vector for the SVM is taken from the word n-grams and character n-grams.
- Word n-grams for sentence $w_1 \ldots w_k$ are $w_1 \ldots w_n$, $w_2 \ldots w_{n+1}$, $\ldots w_{k-n+1} \ldots w_k$ where $w_1, w_2 \ldots w_k$ are words in the sentence.
- Character n-grams for sentence $c_1 \ldots c_k$ are $c_1 \ldots c_n$, $c_2 \ldots c_{n+1}$, $\ldots c_{k-n+1} \ldots c_k$ where $c_1, c_2 \ldots c_k$ are characters in the sentence.
- We intend to limit ourselves to use word n-grams for n $\in 1, 2$ and character n-grams for n $\in 3, 4, 5, 6$.

Feature Pruning

- We then use term frequencyinverse document frequency(tf-idf) to select a subset of features in order to avoid overfitting for the final step which is implementing ensemble of SVM models.

Research Papers

- The following link contains the research paper related to this project.
- `http://cs229.stanford.edu/proj2015/335_report.pdf`

Data Sets

- We will be working on the data set that was provided in the competition. There are 18000 training samples for each language and 2000 validation samples(development samples).
- `https://github.com/Simdiva/DSL-Task/tree/master/data/DSLCC-v2.0`