Language Identification
“Software”

Amal Tarek Murtada Ahmed , 20140101

Asmaa Mostafa Mohamed Ahmed , 20140096

Esraa Mohamed Abdel-Moniem , 20140078

Level 4 , Natural Language Processing

Computers and Information
Helwan University

Dr. Ensaf Hussein
Eng. Ahmed Farrag

Version (1)

April 30, 2018

Abstract

Language Identification's a problem of determining which natural language given content is in by using a data set contain many types of languages and by comparing the written language with the one in this data set and calculate the probability to get the nearest result of the exact language that text is in.

Data set

The original source of the text corpus is wortschatz leipzig corpora. Both the train and test corpus were taken from this corpora.
Trained over 10 languages namely Arabic , Dutch , Egyptian Arabic , English, French, German, Hindi, Italian, Japanese, and Spanish.

Methodology

Experiment 1

We used **N-gram model based Char** :

- Build bi-gram model of a contiguous 2-character slice of a word. We also consider spaces as a character of the bi-grams in order to mark the distinction between beginning/end of word and the inner bi-grams.
- For example, a sentence "he eats" will have following bi-grams: _h, he, e_, _e, ea, at, ts, s_

Experiment 2

We used **N-gram model based word**:

- We build bi-gram for words as following :
    1. Pre-Processing The text corpus
    2. Bi-gram extraction using collocaions
    3. Get extracted feature trainset and combine with classifier.

Experiment 3

We used **Stopwords :**

- We Consider Stop Words as Features of Language . It works as following :

1. Parse and tokenize input text.
2. Compare the tokens with all stopwords lists contained in NLTK corpus in all available languages.
3. Select the most relevant language.
4. Calculate the relevancy level of the selected language.

Results

By using **NaiveBaise** Classifier the Accuracy of Each experiment :

Experiment 1 (N-gram based Char) conducted **67%** accuracy on test set.

Experiment 2 (N-gram based Word) conducted **92%** accuracy on test set.

Experiment 3 (StopWords) conducted **50%** - **67%** accuracy for short text but for long one **85%** - **100%** accuracy.

Discussion

- Stop-words approach is robust and doesn't require any training but fails when comes to short-texts.
- N-gram based char's went wrong because of similarity between chars in most of languages.
- N-gram based word embedding successfully identified short-texts into the actual language.

Conclusion

Hence, for long texts, stop words approach could be the best fit but for short texts, a pre-trained model based on word n-grams could be used for better results.