Language Identification

Bulat Nasrulin

Innopolis University, Russia

Introduction

Problem formulation:

1. Build a system to do language identification
2. Start with simple features, like character bigrams

Dataset

To evaluate language identification quality we use Textcat dataset, that contains twit ids. One of the problem that we faced while mining these twits that a part of them are is not available anymore. Precisely, we successfully retrieve only 70 % of the original data.

Size of training and testing dataset is equal to 1400 sentences each having one of the language class: german, english, spanish, dutch, french.

We remove while processing all words with user tags or hyperlinks.

Baseline solution

As a starting point we use simple character-bigram scheme.

A simple feature set is is feed to train Naive Bayes classification. Note that even this simple approach leads to average 90 % accuracy 1,2.

Extensions

We test the model score with different features and tune different machine learning models.

Character-grams

The experiments at the Figure 3 show that model with character-ngram with n =3 gives the best f-score function.

Ngram combination

We increase the accuracy of the model by using a bigger feature vector formed by combination of ngrams, i.e. n = 1, 2, 3 ... etc.. Indeed, the total accuracy increased comparing to the model that uses for example only 3grams (Figure 4).

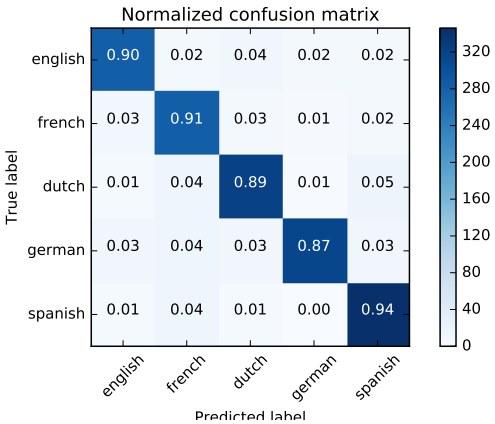


Fig. 1. Character bigram model confusion matrix

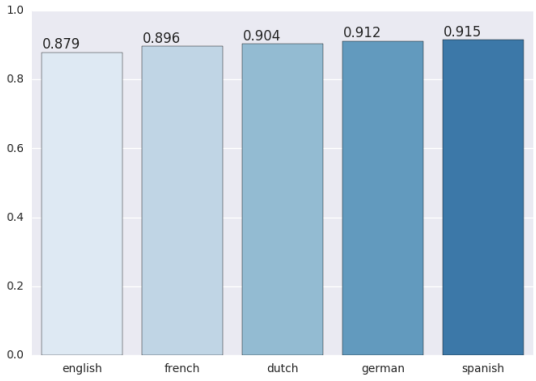
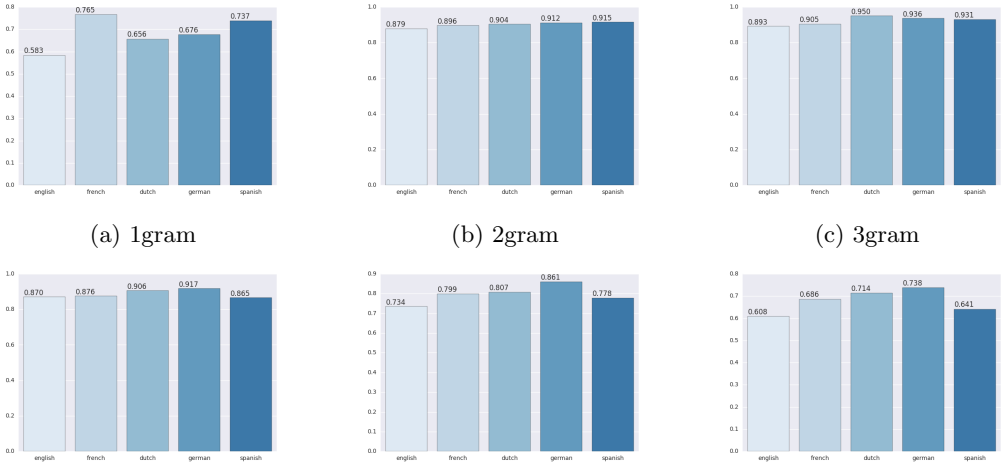


Fig. 2. Character bigram f-score



(a) 1gram (b) 2gram (c) 3gram

(d) 4gram (e) 5gram (f) 6gram

Fig. 3. F1 score for different character gram models from 1 to 6

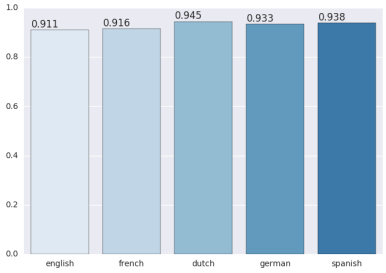
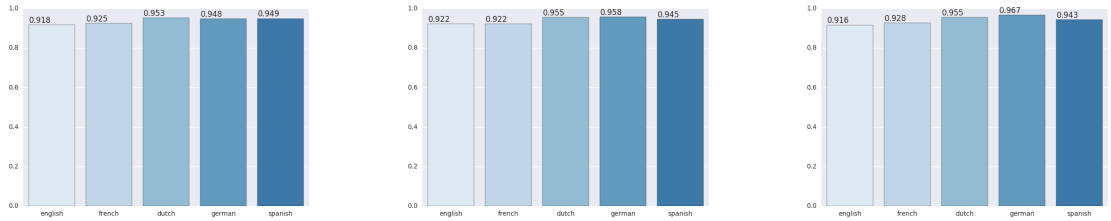


Fig. 4. Character bigram f-score



(a) 3gram (b) 4gram (c) 5gram

Fig. 5. F1 score of ngram models with additional stopwords features

Additional words as features

We can increase the accuracy further if we use some words as additional features. For example, each of this languages has specific and concrete set of stopwords, that are well known and used. To save more space and we take all stopwords sets from nltk and store only the number of intersect with each of these sets. For example, if the sentence contains more german stopwords, it has higher probability that specific sentence is german.

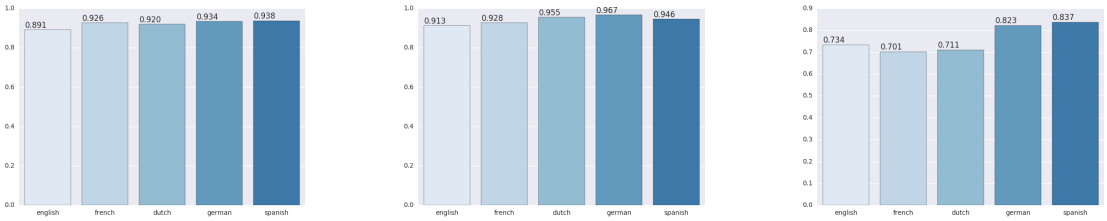
Again, we vary the number of n in ngrams to get the best possible feature set. The best model has 1-5 character gram model and stopwords intersection ration as a feature vector. (Figure 5)

Feature selection

Most Informative Features:
e‘ = 1 french : german = 50.8 : 1.0
ein = 1 german : spanis = 45.7 : 1.0
aar = 1 dutch : german = 44.5 : 1.0
oor = 1 dutch : french = 42.3 : 1.0
german_stop = 2 german : spanis = 40.4 : 1.0
van = 1 dutch : german = 39.3 : 1.0
y = 2 englis : dutch = 38.2 : 1.0
een = 1 dutch : spanis = 37.3 : 1.0
th = 2 englis : french = 35.8 : 1.0
our = 1 french : spanis = 35.4 : 1.0
aa = 1 dutch : french = 33.9 : 1.0
french_stop = 3 spanis : german = 33.2 : 1.0
uf = 1 german : englis = 32.8 : 1.0
eer = 1 dutch : german = 32.7 : 1.0
ik = 1 dutch : french = 32.6 : 1.0

Using the most informative we prune the rest of the features and train Random Forest model.

The experiments show that naive bayes model is showing best result for language identification (Figure 6).



(a) Random Forest (b) Naive Bayes (c) Decision Tree

Fig. 6. F1 score of ngram models with additional stopwords features