Language Dectector report

Zuyao Li

Feb 18 2016

1 Method

1.1 Training Algorithm

I trained Naive Bayes classifier to detect language of a given text.

$$P(lang|text) = \frac{P(text|lang)P(lang)}{P(text)} \tag{1}$$

$$P(text|lang) = \prod_{ngram} P(ngram|lang) \tag{2}$$

$$p(ngram|lang) = \frac{count(ngram, lang)}{\sum_{ngram'} count(ngram', lang)} \tag{3}$$

where ngram means n gram of characters generated from text, P(text) is a constant, and I assum ed prior P(lang) is equal for all languages.

1.2 Inference Algorithm

$$lang^* = argmax_{lang} P(lang|text) \tag{4}$$

where $lang^*$ is the best class that maximize the conditional probability P(lang—text).

1.3 Tricks to avoid underflow

since $\prod_{ngram} P(ngram|lang)$ can be close to zero, I applied log on both sides of equation 2.

$$log(P(text|lang)) = \sum_{ngram} log(P(ngram|lang)) \tag{5}$$

Then the objective function is to optimise $log(P(lang|text))$

1.4 Tricks to avoid sparsity of ngram features

To avoid sparsity of features, I set a minimum threshold to be 10 for count(ngram,lang). For the unseen ngram in the test data set, I will assign it a pseudo count 1 to avoid it to be zero.

2 Dataset

A small dataset for the task. It contains 10,000 lines of text per language split into train/test data. These are sampled from Europarl and NTCIR CLIR collections.

3 Features

The reason to use character level ngram is because that it can avoid sparsity of the feature space compared to word level ngram.

4 Evaluation

I calculate accuracy of each language class for my classifier. It is defined as:

$$accuracy = \frac{count(correct)}{count(total)} \tag{6}$$

5 Result

I tried different size of gram, such as unigram, bigram and trigram. Tables below are the accuracy of different language class and total accuracy with diffrent ngram size.

Language	ngram size	accuracy
de	1	0.987
en	1	0.969
es	1	0.968
fr	1	0.969
it	1	0.949
ja	1	1.0
ko	1	0.987
zh-CN	1	1.0
total	1	0.978625
de	2	0.997
en	2	0.992
es	2	0.991
fr	2	0.995
it	2	0.986
ja	2	0.999
ko	2	0.977
zh-CN	2	0.991
total	2	0.991
de	3	0.997
en	3	0.992
es	3	0.991
fr	3	0.995
it	3	0.99
ja	3	0.962
ko	3	0.944
zh-CN	3	0.991
total	3	0.97825

6 Conclusion

From the experiments, I found that when set gram size to 2, it gave me the best overall accuracy 0.991.

7 Future work

More training data to be applied, make a good assumption about the prior P(lang). Backoff higher order ngram feature to lower one when it is sparse.