567 Computational Linguistics
Project1 Report

Jayaram Kuchibhotla
ID : 50208766
Ubit:jayaramk

Introduction:

Language Identification is task of guessing the natural language of the given text content. This requires an efficient algorithm or approach. Statistical approaches can be used to identify the patterns and frequencies, which vary from language to language .Language classifications can be done using a collection of text called 'corpus' for each language which can be identified by the algorithm. The input text is compared to each corpus and pattern matching is used to Identify the strongest correlation to a corpus.

As there are many words in a language to form a corpus, profiling algorithms are used to create a sub set of words for each language to be used in corpus .The strategy is to choose very common words .For Example, in English words such as 'of' ,'the', 'or' ,'and' can be part of such a subset .

The input text should be long for easier identification of the language as the number of words are many. If the input text is short, algorithm may classify incorrectly. Some languages do not have spaces. Hence, instead of relying on the original words in a sentence, n-grams are formed and are used by models and algorithms.

Another technique is to compare the compressibility of the text to the compressibility of texts in a set of known languages. This approach is called mutual information based distance measure.

Some of the famous language Identification tools are

- TextCat
- CLD
- Ling Pipe

The evaluation of a Language Identification tool or algorithm is bounded by

- fixed list of languages,
- length of sentences and
- Skewed proportion of training to testing instances.

The challenge faced by language identification systems is to distinguish between closely related languages. Similar languages like Serbian and Croatian or Indonesian and Malay present significant lexical and structural overlap

Naïve Bayes Model:

Naive Bayes classifiers belong to the group of simple probabilistic classifiers based on applying Bayes' theorem with naïve assumption that features are independent of each other. They are prominently used for

1. Language identification
2. Classification of documents into spam or legitimate
3. Text categorization

Implementation of Naïve Bayes model for language identification:

For this project three datasets are provided 1.Training(labelled) 2.Testing(unlabelled) and 3.Dev(labelled) – used for accuracy calculation

Training dataset has to be considered for training the model. In this dataset there are 16816 lines, each line has three parts 1.Id 2.Text 3. Label of the language .This data is converted into a Document object where the elements are String- Document Id, Array of strings – the text is converted to character N-grams and String -Language.

The N-grams represent the features and Language of the document represents the class

Naïve Bayes Equations:

Bayes Rule states that

Posterior = (Prior * Likelihood)/Evidence
which can be understood as
P (language of the document/Document given)
= P(Language) *P(Document/Language) /P(Document)
The denominator is dropped as P(Document) is constant

Let $(F_1,F_2,F_3....Fn)$ be the N-grams of the Document object which are considered as the features C_K be the Language ,then the join probability distribution is given by $P(C_k,F_1,F_2,F_3…Fn)$
The conditional Probability can be given as
$P(C_k / F_1,F_2,F_3…Fn) = P(C_k)P(F1/C_k)P(F_2/C_k)P(F_3/C_k). . . . . P(F_n/C_k)$

As $(F_1,F_2,F_3…Fn)$ are part of a single document above equation can also be written as
$P(Ck/D) = P(C_k)P(F1/C_k)P(F_2/C_k)P(F_3/C_k). . . . . P(F_n/C_k)$

P(Ck) = number of documents of a language k /total number of documents
$P(F_i/C_k)$ =
(no. tokens of the N-gram F_i + λ) /(no. of all N-grams in the C_k + λ * total no. of languages)

The above product may result in overflow and hence the equation can be converted into logarithmic form as
$P(Ck/D) = \log(P(C_k)) + \log(P(F1/C_k)) + \log(P(F_2/C_k)) + \log(P(F_3/C_k)) + \ldots + \log(P(F_n/C_k))$

As we are not interested in just the value of P(Ck/D) and we consider the maximum probability value , conversion of the above equation to logarithmic form would not impact as log() is monotonically increasing function

The parameters

1. $P(C_k)$ = the prior language probability
2. no. tokens of the N-gram F_i
3. no. of all N-grams in the C_k
4. total no. of languages

are determined in the training the phase while $P(C_k/D)$ and the values $P(F_i/C_k)$ are calculated for each token of a document of the testing dataset in the testing phase

Also, λ is added for smoothing i.e. to prevent the above probability value from becoming zero which can happen if a N-gram from testing dataset is not found in the training parameters .

The value P(Ck/D) is calculated for k ranging from 1 to 21 (for 21 languages)
If P(Ck/D) is maximum for a language then the document probably belongs to the that language

Results:

The language is predicted for each document in the testing dataset for one combination of lambda and N value.

The range of N values considered is [2, 10]
The range of lambda considered is [0.1, 1] in steps of 0.1

Tuning of parameters is done by fixing the value of N as 2 and lambda value is changed for every iteration.
From the below table it can be observed that N= 2 and lambda = 0.1 gives the maximum accuracy which is 85.27%

N= 2	
Lambda	Overall Accuracy
0.1	85.27%
0.2	85.23%

0.3	85.27%
0.4	85.18%
0.5	85.18%
0.6	85.23%
0.7	85.18%
0.8	85.08%
0.9	85.04%
1	85.04%

Hence Lambda =0.1 is fixed and N is varied in the below table. The maximum accuracy 91.50% is obtained for N=4 and lambda = 0.1

Lambda =0.1	
N	Overall Accuracy
3	89.69%
4	91.50%
5	91.26%
6	90.36%
7	88.08%
8	85.32%
9	82.23%
10	79.86%

For N =4 and lambda = 0.1 the accuracy for all 21 languages is given in the below table

S.No	Language	Accuracy
1	hun	95.00%
2	ell	97.00%
3	dan	83.00%
4	swe	95.00%
5	slo	87.00%
6	nor	78.00%
7	ita	93.00%
8	fin	95.00%
9	fre	97.00%
10	pol	95.00%
11	rum	93.00%

12	cze	86.00%
13	ind	96.00%
14	por	90.00%
15	dut	93.00%
16	tur	96.04%
17	spa	90.00%
18	eng	91.00%
19	vie	81.00%
20	ice	96.04%
21	ger	95.00%

Observations:

1. The accuracy increased from 89.69 % to 91.50% when N is increased from 3 to 4. However, it is observed that accuracy decreased with increase in N from 4 to 10 in second table above.

   Reason:

   The keywords or most frequently occurring words in a corpus of languages would be of lower size. As the N-gram size increases, it is difficult to form such keywords and find them in the training parameters. Hence, there is a decrease in accuracy.

2. It is observed that with increase in lambda value , there is a decrease in accuracy but not much change is observable in the significant part of the accuracy value. Such a decrease can be attributed to increase in the denominator value of the below mentioned expression that is multiplied with 21(number of languages) and would be greater compared to the lambda value in the numerator.

   $P(F_i/C_k)$ =

   (no. tokens of the N-gram F_i + λ) /(no. of all N-grams in the C_k + λ * total no. of languages)

3. For 21 languages and with same values N =4 and lambda = 0.1, it can be observed from the third table above that accuracy is different for different languages.

Reason:

The language identification is mainly done using the prior (the probability of the language) and the likelihood probability value for each token in the text sentence .The more the N-grams are found more would be the likelihood probability and consecutively the chance of assigning a language to that sentence. Finding N-grams of test data in the training parameters is the key. The number of N-grams that are found in training parameters (similar to finding keywords in corpus) which is indicative of a language may not be same for every language. Each language has its own patterns and N-gram frequency. Hence the difference in accuracy

References:

[1] https://lizrush.gitbooks.io/algorithms-for-webdevs-ebook/content/chapters/language-detection.html

[2] https://en.wikipedia.org/wiki/Language_identification

[3] http://stackoverflow.com/questions/29290107/detecting-language-using-stanford-nlp

[4] https://web.stanford.edu/class/cs124/lec/naivebayes.pdf