**COMP 472 Project Two Report, Naive Bayes Classiﬁer**

Thanh Tung Nguyen 1

1 40042891 nguyenthanhtung2605@gmail.com

1. Introduction
   1. Purpose

This purpose of this report is to compare to explore different types of machine learning (ML) models to determine the most likely language of a tweet written in one of the languages given (Basque *(eu)*, Catalan *(ca)*, Galician *(gl)*, Spanish *(es)*, English *(en)* and Portuguese *(pt)*). The method used to solve the question is Naive Bayes classification implementing different hyperparameters which are vocabulary, size of n-grams and smoothing value. It is also required to develop a more sophisticated model (BYOM) that could outperform its predecessors.

* 1. Technical Details

Part one of this project has already been submitted and demoed. It involved using Naive Bayes classification to determine which language the given tweets are written in. Since Naive Bayes classifier is an implementation of machine learning, it needs training data with the linguistic features being labelled and fed to the ML models. It is, therefore provided with a dataset containing two files - a training set and a test set of tweets and each tweet of both sets already has its language identified. The test set is labelled in order to measure the models’ performances.

Before feeding in the models, the tweets are to be processed and the processing should be based on vocabulary types – the accepting sets of characters, size of n-grams – the combination in sequence of n number of characters and a smoothing value - a real in the interval [0 . . . 1]. These hyper parameters should result in numerous ways tweets are being parsed, converted to n-grams, and fed to the models. The combination of vocabulary and size of n-grams also produces different vocabulary sizes, for example, the 26 letters accepting regex [a-z, A-Z] and the character bigrams generate a vocabulary size of 2,704 (*(26 \* 2) ^ 2*). Vocabulary sizes, smoothing values and other attributes are factors that varies the probabilities of tweets being predicted falling into the six languages. The models predict each tweet to be the language with highest probability.

The BYOM is expected to produce better results than the given ones. Therefore, two new hyperparameters are introduced. The first is options to remove irrelevant linguistic features [1, 2] from tweets. They are hashtags (i.e.: #iloveyou), user tags (i.e.: #@tungnguyen) and urls (i.e.: <https://www.xyz.com/>). The second is not to use n-grams of letters but tokenize tweets into words instead and compute the vocabulary size accordingly.

1. Analysis of the Initial Dataset

#### The training dataset is a collection of 18.3k tweets approximately. Each of the tweets was labelled with its corresponding language code. The table below gives a good summary of the given dataset.

**Table 1.** The numbers of tweets by language and dataset. Distribution of the languages in Training Dataset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Language** | **Training Set** | | **Initial Test Set** | | **Demo Test Set** | |
|  | **(#)** | **(%)** | **(#)** |  | **(#)** | **(#)** |
| **ca** | 1,493 | 8% | 1,391 | 20% | 75 | 1% |
| **en** | 971 | 5% | 483 | 7% | 516 | 7% |
| **es** | 12,855 | 70% | 4,589 | 65% | 3,973 | 56% |
| **eu** | 374 | 2% |  | 0% | 380 | 5% |
| **gl** | 456 | 2% | 506 | 7% | 1 | 0% |
| **pt** | 2,169 | 12% | 96 | 1% | 2,055 | 29% |
| **Total** | **18,318** | **100%** | **7,065** | 100% | **7,000** | 100% |

As can be seen in the training set, the tweets are unevenly distributed to the languages. This unbalance can be understood that some languages are far more popular than others. These popular languages should cover a majority of their vocabularies. Whereas, the less popular ones might miss a large number of words available in their vocabularies (these missing words should be given a smoothing value). This could result in a low probability when checking the given n-gram words in each tweet against the available vocabularies of the less-popular languages. Consequently, the language prediction would favor the languages with greater popularity and overlook the real linguistic features. In other words, this dataset could prone to overfitting to some extent and might need a more balanced distribution of observations per class.

1. Motivation and Description of BYOM
   1. Motivation:

As can be noticed that the models being asked in the question do not consider noisy tokens [1, 2] such as user mentions, hashtags, URLs, etc. For instance, in the tweet ‘Es muy aburrido trabajar desde casa @the\_lone\_ranger #workingfromhome #bored #quarantine #self\_isolation #panedemic’, the noisy irrelevant Twitter artifacts would make the models confused between Spanish and English.

Tweets are also being processed not very much in a nature way when using n-grams of letters instead of proper words. For Latin-based languages, words should best differentiate one to another rather than letters. This would make the classification more frequency-based than linguistic feature-based. In some cases of using unigram or bigram of letters of [a-z] or [a-z][A-Z] regex, the models should often classify tweets to as Spanish (es) – the language with greatest frequency.

To address these two issues, two new hyperparameters are introduced in the BYOM. One is to remove the noisy tokens and the other one is to process tweets by words.

* 1. Description:

The BYOM would best perform by using the two new hyperparameters. However, it should still do better than the given models by implementing one.

* + 1. *Removal of Twitter Artifacts:*

This additional hyperparameter allows the models to have options to removes 3 types of artifacts - urls, hashtag and @user tags like a filter. A single artifact, combination of two or all together filter can be applied on top of the given hyper hyperparameters to produce better classification results. It should take a flag letter of *-f* and value ranging from 0 to 7 – with 7 is to filter all.

* + 1. *Use of Words instead of N-grams of Letters:*

This hyperparameter should replace the size of size of n-grams. The other hyperparameters remain unchanged. It takes a flag letter of *-n* and value of 4.

1. Analysis of the Results
   1. Results of Demo Test Set

7,065 tweets of the demo test set were annotated automatically using 4 different sets of hyperparameters as described below:

*'-V 0 -n 2 -d 0.01: vocabulary of 26 letters of [a-z] regex, character bigrams, smoothing value of 0.01.*

*'-V 0 -n 2 -d 0.5: vocabulary of letters accepted by the isalpha() method, character bigrams, smoothing value of 0.5.*

*'-V 0 -n 3 -d 0.5: vocabulary of letters accepted by the isalpha() method, character trigrams, smoothing value of 0.5.*

*'BYOM: -V 2 -n 4 -f 7 -d 0.5: vocabulary of letters accepted by the isalpha() method, using words instead of n-grams and filtering all twitter artifacts.*

Each of the sets above is considered a ML model and their performances are compared as shown in the Figure 1 below.

**Figure 1- Performance of Models with Demo Test Set**

Figure 1 reveals that -V 0 -n 2 -d 0.01 and -V 0 -n 2 -d 0.5 undoubtably the worst performers having same results. This meets the expectation mentioned in BYOM’s motivation. The issue is that these two models both use bigram of letters which is very simple and overlooks important linguistic features. It makes these two models favor the language with greatest frequency. The fact that the trigram performs better the bigram ones also supports this claim because the more grams are taken into account, the more intrinsic characteristics of language are gathered and fed to the models. This explains why the BYOM clearly has the best results, as it considers tweets by words/tokens which contain the most linguistic features. These features are the key criteria to best distinguish one language among different ones.

To analyze in detail, the confusion matrix of the demo test set is provided in the appendices as Figure 5. As shown in the matrix, -V 0 -n 2 -d 0.01 and -V 0 -n 2 -d 0.5 models predict all the tweets as Spanish (es) – the most popular language with the highest frequency in the training set. The relatively high values of accuracy and F1 measures was due to the majority of tweets in the test set were originally in Spanish. They were all lucky matches. If most of the tweets were collected from the other languages. The performance would significantly reduce. With more linguistic features being considered, trigram model could correctly identify more languages than just Spanish. The BYOM is doing best with the most properly matched tweets. The per-class precision, accuracy and f1-measures shown in Figure 6 in the appendices also show that BYOM is the best performer whereas the other three models only do well with Spanish.

* 1. Results of Initial Test Set

Like the demo test set, same sets of hyperparameters/models are applied to this initial test set to annotate 7k of tweets. It is expected to generate the similar outcomes.

**Figure 2 - Performance of Models with Initial Test Set**

As expected, -V 0 -n 2 -d 0.01 and -V 0 -n 2 -d 0.5 are still the worst performers with same results shown in Figure 2 above and Figure 7 - Per-class Precision, Recall and F1 Measure in the appendices. These models still go for only Spanish. Yet again, the -V 0 -n 3 -d 0.5 model performs slightly better and BMYO undoubtedly outperforms the rest.

To see the contribution of each new hyperparameter to the BYOM’s performance, the Twitter Artifact Filter is isolated by switching the value from 7 (filtering all) to 0 (no filtering) to see the metrics shifting.

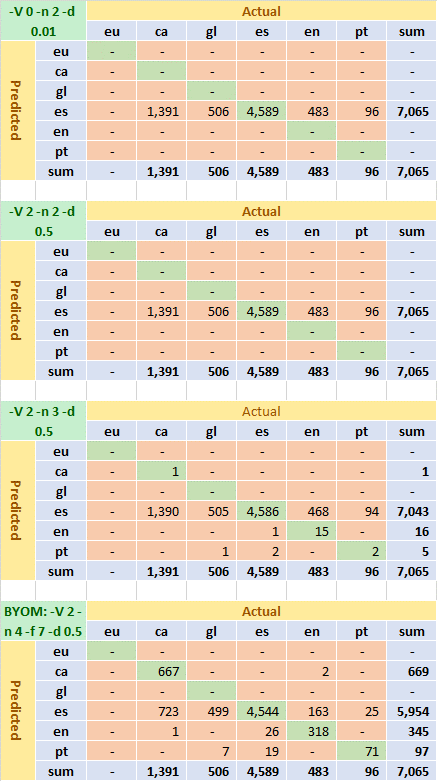
**Figure 3 - Performance of BYOM with and without filter**

The chart (Figure 3) indicates that the metrics has been very slightly shifted unfavorably when adding filter to the BYOM. In fact, it is not in line with the expectation mentioned in the motivation. This could be explained that the filter does not have much impact to the model and the Twitter artifact might have been written in the given languages.

* 1. Conclusion

It is clear that the BYOM implementing tokenized words has outperformed the assigned models, since it captures the most linguistic features instead of overlooking them by using n-grams of letters. To simply put, the selection of feature extraction methods is key to produce desired results. The unbalanced distribution of observations could cause ML models overfitted and result in inaccurate classification when using Naïve Bayes Classification. In addition, it is important to filter out noisy factors while processing data input, but it might not necessarily improve the performance if those factors are trivial as shown in the experiment above.

1. Appendices - Tables or Graphs
   1. Confusion matrix of Demo Test Set



**Figure 5- Confusion matrix of Demo Test Set**

* 1. Demo Test Set: Per-Class Precision, Recall and F1 Measure

**Figure 6 - Demo Test Set: Per-Class Precision, Recall and F1 Measure**

* 1. Initial Test Set: Per-Class Precision, Recall and F1 Measure

**Figure 7 - Initial Test Set: Per-Class Precision, Recall and F1 Measure**

References

1. Arkaitz Zubiaga, Inaki San Vicente and others: Overview of TweetLID: Tweet Language Identification at SEPLN, 3–5 (2014).
2. Jennifer Williams and Charlie K. Dagli: Twitter Language Identification of Similar Languages and Dialects Without Ground Truth, pg. 81 (2017).