

0.1 Introduction

This assignment was about text classification. We had to classify News titles by their topic; our classes were “science”, “health”, “business” and “entertainment”. Our raw features were the title itself and the publisher. The dataset was already divided in 10000 training samples, 1000 validation samples and 1000 test samples. Our objective is to create a feature extraction process and to build one or more classifiers.

0.2 Feature Extraction

Since we are dealing with text classification we chose to employ the Bag of Words representation to encode the titles. In order to do that we had to build a fixed size dictionary. In Figure 1 we can see how the dictionary size affects the performance of the models we intend to build. Our goal with Figure 1 was not to find the

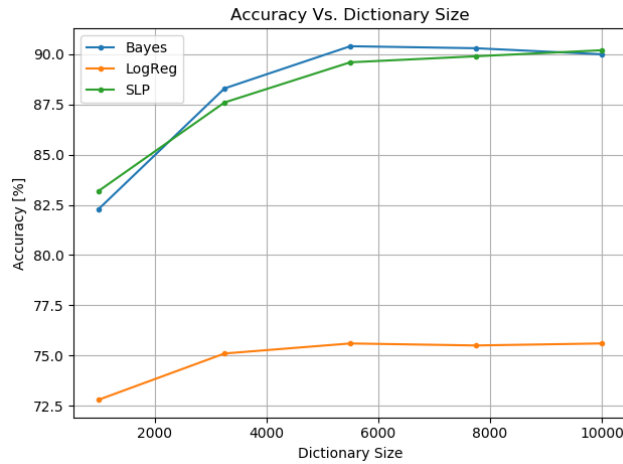


Figure 1: Test Accuracy vs. Dictionary Size

optimal dictionary size but to see if a bigger dictionary implied a better performance and this seems to be the case. Another decision we had to make was choosing if the publisher was a useful feature to keep. As we can



Figure 2: Publisher Class Distribution

see from Figure 2, the 19 different publishers mostly belong to a couple of classes. Based on that information we decide to keep the publisher information and we concatenated it to the title BoW. The simplest solution we found to encode the publishers was just to number them from 0 to 18. We also took into consideration normalization techniques, both in terms of words normalization (remove common words from the dictionary and stemming) and BoW normalization. We found out that using only one word normalization technique worsened the performance of all the models but using them both made them perform better. That said we used

word normalization techniques for all the test we performed. Each model reacted differently to different BoW normalizations so we will discuss them separately.

0.3 Classifiers

We decided to build a total of 4 classifiers: Multinomial Bayesian Classifier, Multinomial Logistic Regression, Single Layer Perceptron and Multi-Layer Perceptron. The first two choices were biased upon the results we obtained in the Sentiment Analysis assignment. The latter two were chosen because of their versatility and previous assignments results. To be completely honest we would have liked to use SVMs since they proved to be the best in text recognition but we cannot since we are dealing with a multi-class problem, we would need a sort of multinomial SVM for which we do not have the theoretical background nor any code ready to be used. We settled on a dictionary size of 8000 words, as it is very close to the maximum possible size given the normalizations performed on the corpus.

0.3.1 Multinomial Naive Bayesian Classifier

We chose the *Multinomial Bayesian Classifier* as our baseline for testing since it proved to be simple but effective. Its simplicity meant that we could run multiple tests without worrying about the computational time needed to train it. The results of this classifier caught us off guard and proved to be challenging to beat. The Multinomial Bayesian Classifier reacted poorly upon BoW Normalization. L1 and L2 nearly cut the test accuracy in half and MaxAbs lowered it by a couple of percentage points. With those results we chose not to use any normalization to let this classifier perform at its best. The training accuracy was 94.39%, the validation one was 89.2% and the test one was 91.1%.

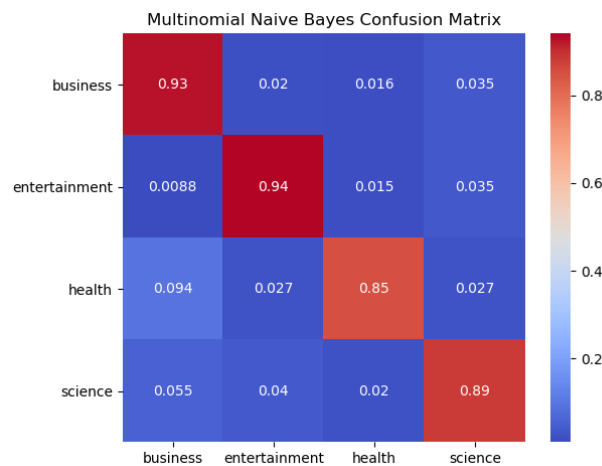


Figure 3: Bayesian Classifier Confusion Matrix (Test Set)

0.3.2 Multinomial Logistic Regression

0.3.3 Single Layer Perceptron

0.3.4 Multi-Layer Perceptron

0.4 Results Analysis

0.5 Conclusions