COMP 6721 Project 2

Vida Abdollahi

40039052

vida.abdolai@gmail.com

1. Introduction and Technical Details

1.1 Text Preprocessing

Baseline text preprocessing includes:

- Tokenization — convert sentences to words
- Removing unnecessary punctuation, tags
- Converting all letters to lowercase
- Removing numbers
- Lemmatization

Extra steps that I took for text preprocessing as requested in the assignment include:

- Removing stop words
- Removing characters which have length less than 2 or more than 9

1.2 Classification Workflow

The classification has two phases, a learning phase, and the evaluation phase. In the learning phase, classifier trains its model on a given dataset and in the evaluation phase, it tests the classifier performance. Performance is evaluated on the basis of various parameters such as accuracy, error, precision, and recall.

1.3 Naive Bayes Classifier

Naive Bayes is a family of probabilistic algorithms that take advantage of probability theory and Bayes' Theorem to predict the tag of a text (like a piece of news or a customer review). They are probabilistic, which means that they calculate the probability of each tag for a given text, and then output the tag with the highest one. The way they get these probabilities is by using Bayes' Theorem, which describes the probability of a feature, based on prior knowledge of conditions that might be related to that feature.

1.4 Feature Engineering

The first thing we need to do when creating a machine learning model is to decide what to use as features. In this case though, we don't even have numeric features. We just have text.

We need to somehow convert this text into numbers that we can do calculations on. We use **word frequencies**.

1.5 Classes

We also have four classes in this problem: story, ask_hn, show_hn, poll
Conditional probability of each of these classes have been calculated and smoothed.

1.6 Dealing with 0 frequency

The frequency-based probability might introduce zeros when multiplying the probabilities, leading to a failure in preserving the information contributed by the non-zero probabilities. Therefore, a smoothing approach, for example, the Laplace smoothing, must be adopted to counter this problem.

This solves the zero-probability problem in the dataset. I also considered the case that there might be word in our test set never seen by our train set. For this assignment, I also added a small probability to each of these words based on the Laplace smoothing.

1.7 Result of the Prediction

To prevent under-flow, instead of multiplying the probability of each word, I calculated the sum of the logarithm of all of them; summation is also much faster than the production.

1.8 Performance

Performance is evaluated on the basis of various parameters such as accuracy, error, precision, and recall.
If we want to calculate these parameters, first we need the exact definition of TP, TN, FP, FN.
Let only consider class "story" as an example:

True positives (TP): the cases for which the classifier predicted 'story' and the text was actually 'story'.
True negatives (TN): the cases for which the classifier predicted 'not story' and the text was actually 'not story'.
False positives (FP): the cases for which the classifier predicted 'story' but the text was actually 'not story'.
False negatives (FN): the cases for which the classifier predicted 'not story' but the text was actually 'story'.

So, based on these definitions we can calculate different parameters:

$$\text{Accuracy} = \frac{\#\text{correctly classified items}}{\#\text{all classified items}} \qquad \text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$f_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

2. Result for Baseline

Performance Table

	Total	Story	Show_hn	Ask_hn	Poll
Accuracy	0.9773	0.9886	0.9767	0.9978	0
Precision	0.6357	0.9989	0.7374	0.8064	0
Recall	0.7322	0.9774	0.9567	0.9946	0
F1-measure	0.6805	0.9880	0.8343	0.8906	0

All the values are calculated based on the definition described in the introduction phrase. The 'Total' column's value are the average of the four values received from each class.

Confusion Matrix

	Actual Value				
Predicted Value		**Story**	**Show_hn**	**Poll**	**Ask_hn**
	Story	123991	114	4	18
	Show_hn	1659	4691	0	11
	Poll	0	0	0	0
	Ask_hn	1202	98	2	5425

The 'Actual Values' are the real values of each class, means that in the test set they were labeled exactly as their same value.
The predicted values are the result from the classifier.
The cell in first row – first second that is highlighted in green shows all the 'story' post type that were labeled correctly.
So technically the diagonal values all represent correctly identified prediction for each class.

3. Result for Stop Words

Performance Table

	Total	Story	Show_hn	Ask_hn	Poll
Accuracy	0.9768	0.9766	0.9618	0.9948	0
Precision	0.6332	0.9990	0.7329	0.8010	0
Recall	0.7333	0.9766	0.9618	0.9948	0
F1-measure	0.6796	0.9876	0.8318	0.8874	0

Confusion Matrix

	Actual Value				
Predicted Value		**Story**	**Show_hn**	**Poll**	**Ask_hn**
	Story	123891	93	4	18
	Show_hn	1708	4716	0	10
	Poll	1	0	0	0
	Ask_hn	1252	94	2	5426

4. Result for Word Length Filter

Performance Table

	Total	Story	Show_hn	Ask_hn	Poll
Accuracy	0.97000	0.9765	0.8309	0.9429	0
Precision	0.6223	0.9910	0.7015	0.7967	0
Recall	0.6875	0.9761	0.8309	0.9429	0
F1-measure	0.6533	0.9813	0.7881	0.8991	0

Confusion Matrix

	Actual Value				
Predicted Value		**Story**	**Show_hn**	**Poll**	**Ask_hn**
	Story	123830	810	5	307
	Show_hn	1729	4074	0	4
	Poll	1	0	0	0
	Ask_hn	1292	19	1	5143

5. Result of infrequent word filtering

Unfortunately, I could not finish this part on time as the running time of the program for each round is really high. I try to finish it by the day of the presentation and will print the result on the paper.

6. Results and analysis of the smoothing experiment

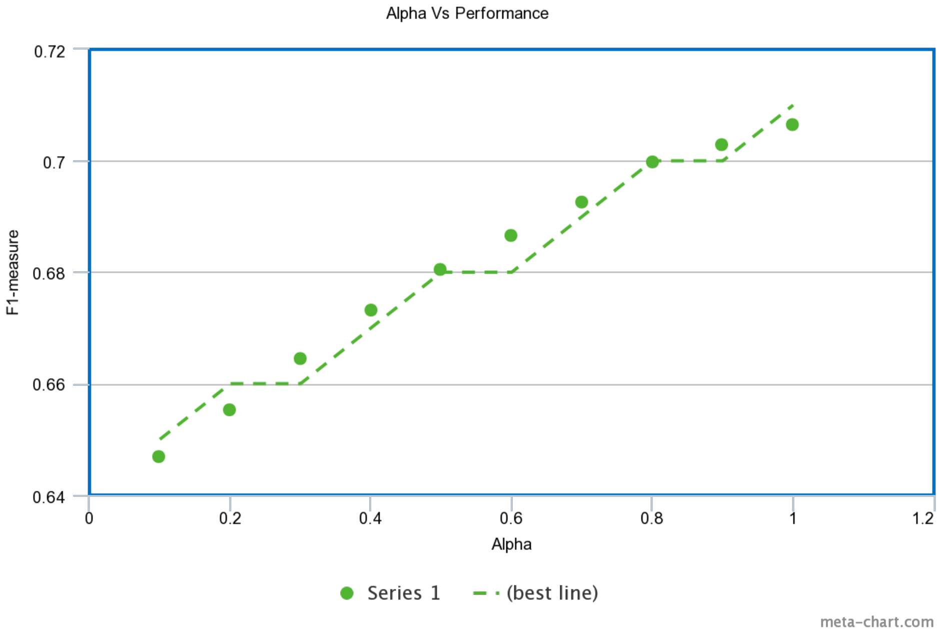

Fig. 1. Plotting F1-measure values with respect to the Alpha values

If α = 0 then we have no smoothing, it means that our model cares too much about the training data.
If α = 1 then we have Laplace smoothing.

As the alpha increases, the likelihood probabilities are moved to uniform distribution.

7. Compare and discuss the results of the 4 experiments

- Performance slightly differs among the 4 experiments, but it's almost the same. I think that's because in the pre-processing step, I already used regular expression to remove unnecessary punctuations and also some words that are in the stop words list were already ignored in the preprocessing step.
- There seem to be not enough sentences labeled with 'poll', the classifier ended up with almost 0 accuracy in this class.
- Among all the 4 classes, story has the highest accuracy and F1-measure. It shows that the train set has enough information about this category to predict new sentences with high accuracy.
- As the smoothing value increases to 1, the overall performance gets slightly better. For this part I plotted alpha to the F1-measure variable.

8. Future Development

If I wanted to continue working on this assignment, I would have considered using K- Fold Cross Validation.
It is the best way to determine optimal values hyper parameters such as Alpha in Naïve Bayes.
Once we are done with training our model, we just can't assume that it is going to work well on data that it has not seen before. In other words, we can't be sure that the model will have the desired accuracy and variance in production environment. We need some kind of assurance of the accuracy of the predictions that our model is putting out. For this, we need to validate our model.
Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

The general procedure is as follows:

- Shuffle the dataset randomly.
- Split the dataset into k groups
- For each unique group:
    - Take the group as a hold out or test data set
    - Take the remaining groups as a training data set
    - Fit a model on the training set and evaluate it on the test set
    - Retain the evaluation score and discard the model
- Summarize the skill of the model using the sample of model evaluation scores

References

1. https://medium.com/hugo-ferreiras-blog/confusion-matrix-and-other-metrics-in-machine-learning-894688cb1c0a
2. https://www.nltk.org/
3. https://en.wikipedia.org/wiki/Additive_smoothing
4. https://www.youtube.com/watch?v=HBi-P5j0Kec