Fall 2024

EDS 6397 – Natural Language Processing

Assignment #2 – Sentiment Analysis Using Naïve Bayes Text Classification

Use of artificial intelligence assistant such as ChatGPT in developing the code for this assignment is allowed and encouraged. However, the report needs to be completely written by you. Use of grammar correction tools is disallowed.

Assignment Overview

The objective of this assignment is to familiarize you with a probabilistic text classification method named Naïve Bayes classifier. You will train a Naïve Bayes classifier to categorize some short movie reviews into two classes; positive and negative.

Assignment Description

Make sure you follow the instructions below very carefully.

Input Data Description

You need to download the data from Kaggle at the following web address:
https://www.kaggle.com/datasets/kingabzpro/movie-reviews-nlp

Analysis Required

Your objective is to write a code in Python (in a Jupyter notebook) to do the following:

- Read the movie reviews
- Clean up the input (**see Data Cleanup section below**)
- Perform regular expression to remove punctuation and symbols (**See preprocessing section below**)
- Randomly select 80 percent of reviews for training and 20 percent for testing.
    - Make sure to fix a random state when you use scikit-learn to form your training and test sets. This way the same random split of training and testing data happen in all your runs.
- Train a Naïve Bayes classifier that categorizes the reviews into two classes; positive and negative considering the following:
    - Calculate prior probabilities for each class
    - Calculate likelihood values of each word given each class
    - Calculate the posterior probability using logarithmic representation of probabilities
    - Use Laplace Smoothing (add 1)
    - Remove unknown words
- Perform test on 20% of the reviews that you set aside for testing
- Calculate confusion matrix using scikit-learn and report precision, recall, and F-1 score.

Data Cleanup

The filename is moviereviews.tsv and it can be read Pandas using **read_table** function. The first column is "label" and second column is the "review" itself. This file contains 2000 brief movie reviews. There are a few missing reviews or Null values that you need to remove.

You also need to convert the labels to numerical values (0 for negative, 1 for positive)

Preprocessing

You need to create a preprocessing function that takes care of removing symbols, punctuations, extra spaces, etc. This function will look like this:

```
preprocess_text(text, lemmatize_words=True, remove_stop_words=True, handle_logical_negation=True)
```

The three Boolean input arguments will help you run a few scenarios:

- Test if lemmatization helps improve the accuracy.
- Test if removing stop words helps improve the accuracy.
- Test if logical negation improves the accuracy metrics. Refer to lecture 5 for how to do this. Note that for logical negation handling to work, you need to lemmatize the input text (Spacy does this automatically when you create a doc object). So can't have lemmatize_words=Flase but handle_logical_negation=True.

Therefore, you need to run the entire training and testing 4 times:

1- Without lemmatization, removal of stop words, or handling logical negation
2- With lemmatization
3- With lemmatization and removal of stop words
4- With lemmatization, removal of stop words, and handling logical negation

Your need to submit the accuracy assessment results for all 4 scenarios above.

Tip: try to write your code with functions. Then you can dedicate one cell to calling appropriate pre-processing, training, testing, and accuracy assessment functions for each scenario.

Questions

Dedicate a section of your report to directly answering the following questions:

1- In your Naïve Bayes Classifier, how are misspelled words treated?
2- Does Naïve Bayes classifier show vulnerability towards typos or misspelled words?
3- Is there any recommendation you have to fix spelling errors?
4- Do you think it is beneficial to fix spelling errors?
5- Do you think if we had formed n-grams and used those probabilities instead of Bayes classifier, we could get better results?

What to submit:

1- Your Python Jupyter Notebook should include plenty of comments and explanatory cells to ensure that your code is easy to read and understand. The Notebook should also contain the results of each cell executed by you. The notebook should be named **[Last_name]_HW2.ipynb**
2- A report (1-2 pages) **in PDF format** that briefly discusses the assignment and the results of 4 scenarios. Make one table that lists precision, recall, and f-1 score for each class and for each scenario. You can also compare overall accuracy values of the 4 scenarios. The last section of the report should clearly answer the 5 questions asked above. The name of the file should be **[Last_name]_HW2.pdf**

Late submissions will be penalized at a rate of 1 point per hour. We round up the time to the nearest hour.

Due Date: September 19, 2024 by 11:59 PM (submit through Teams)