INLS 690

Yunwei Cao

Vector Voyager (Yunwei)

03/19/2023

**Technical Document for HW2**

The task of this assignment is a binary classification of text data. I explored two models for this: logistic regression and transformer. This is because based on the blog articles I have read, these two models perform well in text classification scenarios.

For logistic regression, I explored three aspects. First, I tried two feature extraction methods: Count Vectors & TF-IDF Vectors. For TF-IDF Vector, I chose three levels of input tokens: Word Level, N-gram, and Character Level. Based on the performance of the validation dataset, the most basic Count Vectors had the highest accuracy. This was somewhat frustrating but within my expectations.

Second, after deciding to use Count Vectors, I used the chi-square test for feature selection provided in class. To improve accuracy, I set different numbers of features and C1 parameter values in logistic regression. According to my blind 5-7 attempts, I found that when the maximum number of features was set to 15000 and C1 was set to 0.1, it performed best on the test dataset. Another interesting observation is that when I set the maximum number of features to 17000 and C1 to 0.3, the accuracy performance on the validation dataset was the same as that of the (15000, 0.1) combination at 88.63%, but their performance on the test dataset was different and I haven’t figured out the reason...

Finally, since all of these methods have corresponding Python packages or pre-written functions, I directly called them when writing code. while at the same time ensuring that I truly understood their underlying mathematical principles, I spent a considerable amount of time reading source code implementing Count Vectors, TF-IDF Vector, and logistic regression so that I could implement them from scratch.

By using logistic regression, I achieved the highest accuracy rate of 88.9%. Then I decided to try the transformer model. Since I was unfamiliar with it, I found two ready-made repos on hugging faces using the defaulted modes: One is distilbert-base-uncased-finetuned-sst-2-english with revisions (af0f99b) and the other is the general prototype Bert-base-uncased. For the first one, I haven’t even trained the model with my own train dataset but their original one (a collection of English books sorted by Stanford), and it achieved an accuracy rate of 90.3% on the test dataset, which shows its robustness. For the second model, I trained with the custom dataset on google Collab and it gets the highest accuracy of 92.5%. It is worth mentioning that I just use the first 3000 reviews from the training dataset since it would be too time-consuming to use all of them(more than 24 hours). I believe with more feedings of data and a fine-tuned model, the accuracy of the result could get higher.