Machine Learning - CSCI 5622

HW 6 - Feature Engineering

Santhanakrishnan Ramani

Wednesday 16th November, 2016

Analysis

Kaggle Username: santhanakrishnanr

The table below represents the various feature combinations I tried and their respective training and validation accuracy by doing a 5-Fold cross validation.

Action Performed	Training Accuracy	Validation Accuracy
Just ran the given code	0.8735	0.6596
Removed stopped words	0.9191	0.6627
Added n_grams of range (1,3)	0.9902	0.6741
Removed n_grams and changed CountVectorizer to TfidfVectorizer	0.8584	0.6821
Pre-processed by removing punctuations	0.8704	0.6820
Performed stemming on the data	0.8432	0.6837
Added the page column to the existing data	0.8609	0.7322
Added the trope column to too	0.882	0.76
Added the trope column by adding a space before all capital letters	0.8760	0.756
Added a new column called num_present	0.8757	0.7583

I initially started by just running the given code in order to set a baseline, when submitted to Kaggle I got an accuracy of 0.62. Then I tried the basics in any text classification like removing stopwords, making all the text to lowercase, adding ngrams. But the training accuracy increased a lot, so I thought it was over fitting and removed n_grams. Then I converted the CountVectorizer which I was using all along to TfidfVectorizer, that gave a bit increase in my validation accuracy. Then I added a preprocessor function argument with a function handle which removes any punctuations and stemmed the text, this in turn increased my validation accuracy.

As of now I was just using the sentence column from the given data, then I tried appending the page and trope column to the data. This gave a quite an increase in my validation accuracy suggesting me that they are of some importance. Then I thought of an idea to split the words in the trope column by adding a space before every capital letter in the given text. This didn't improve my validation accuracy but I got a good increase in my test accuracy when I submitted to Kaggle. As a last feature I added a new column called num_present which basically is a binary field based on whether there is number in the sentence text, which too gave me good increase in my Kaggle score.

After trying all the possible combinations, the best combination that gave me the highest test kaggle score of 0.71 was stopwords + lowercase + removing punctuation + stemming + adding 'page' and 'trope' column and 'num_present' column, which was also supported by the good training and validation accuracy achieved using this combination.

Used 5 late Days.