horizontal line

Sprint 4 Results

Sprint End: 11.17.2019

Sprint Captain: Rachel Li

Sprint Co-captain: Hongying Liu

**─**

# Sprint Goal for the Past Week

## Optimize Neural Nets and the Train Function:

* Modify the train function for better performance: **on track**

1. Add better stopping method in the train function based on validation loss to avoid overfitting
2. Read more papers and try to modify the baseline model

* Generate additional features based on the n-gram that we created before and add them into the model **on track**
* Add more layers into the neural nets **on track**

## Feature Engineering:

* Continue adding more features to train neural network models. This week we will focus on POS tags, and Automated Feature Engineering. **on track**

# Results

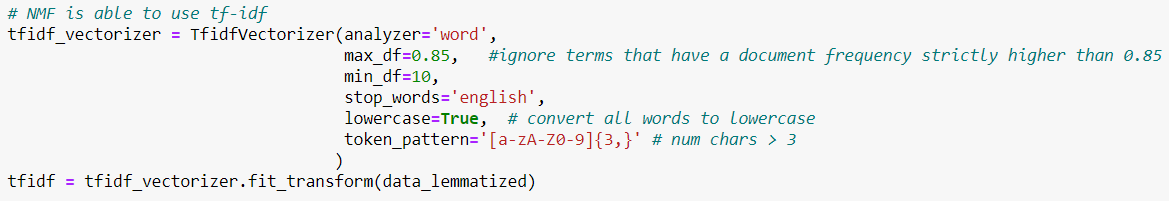
1. Transform the output labels of gold standard dataset

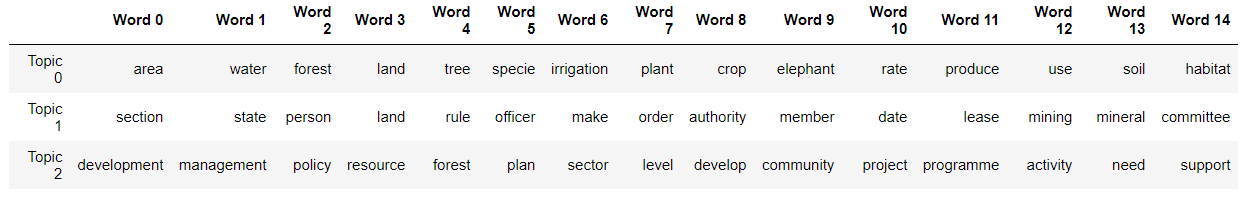
* Since there is only a tiny proportion of data belonging to “negative” class, we transformed the original output labels from 3 classes into 2 classes to avoid missing “negative” class in predictive results. The transformation is shown as below:
  + Neutral & negative —> non-positive
  + Positive —> positive
* The F1 score using the updated datasets with 2 classes turned out to be 0.818, which is close to the score generated using the dataset with 3 classes.

1. Feature Engineering - Topic modeling

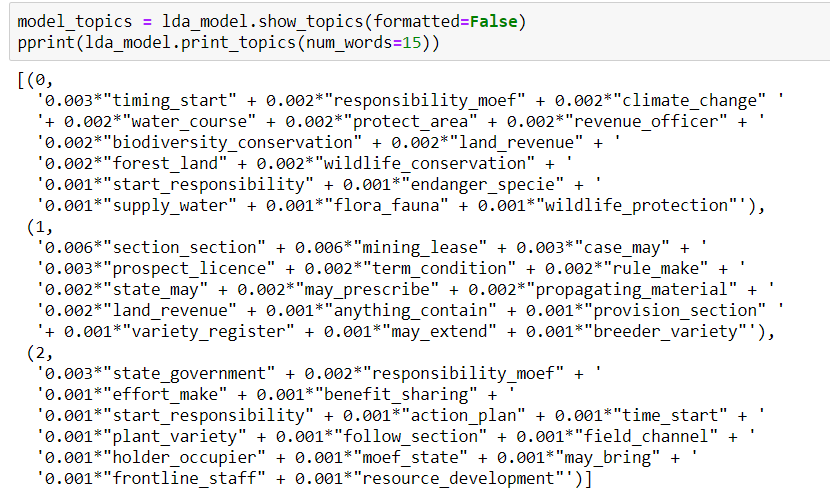
We have tried both unigram and bigram methods to help understand the topics:

* unigram: We have used tfidf to filter tokens and obtain the frequency of each singer word(the tfidf vectorizer for LDA is shown as follows), and then removed top frequent words (with a threshold of frequency at 0.85) and words that show up less than 10 times. With the remaining words we used grid search and obtained the optimal model with 3 topics.





* bigram: We utilized the nltk package to generate bigram tokens and used gensim package to get topic models. The optimal resulting model turns out to include 3 topics as well. The key words in each topic are shown as follows. In order to better understand the topics, we also found out the dominant sentences under each topic which are shown in the github file.



1. RoBERTa embedding and Neural Network Model

* Modified the neural network model for binary classification (non-positive or positive) instead of the original classification with 3 classes
* Added ROC AUC score for model evaluation
* Amended the early stopping method which stops training when the evaluation loss has not been improved for 10 epochs
* Current evaluation scores:
  + F1 score: 0.509
  + Accuracy score: 0.58
  + ROC AUC score: 0.525

1. BabbleLabble implementation

We have figured out the rules of writing certain babbleLabble explanations, but got stuck on how to transfer our csv data to pkl file, as Babble only takes in the pkl format data. We tried many ways but still not sure how to achieve this transformation. The pkl file in the babble tutorial contains specific data structures and need further explanations and help on this.