Texas State University

Machine Learning

Identifying MBTI Using Natural Language Processing

Garrett Gridley & Jean-Michel Mailloux-Huberdeau

Outline

- Introduction
- Literature Survey
- Methodology
- Results
- Conclusions and Future Work

Introduction

- MBTI: Myers-Briggs Type Indicator
  - A self taken questionnaire with the goal of defining personality type among 16 different options
- Assumes that people all have specific preferences that guide interests, needs, values, and motivations
- Test is historically inaccurate for its poor validity, poor reliability, and the fact that the measuring categories are not independent

Literature Survey

- *Ma, Anthony, and Liu, Gus. "Neural Networks in Predicting Myers Brigg Personality Type From Writing Style."*
  - Dataset consisted of sentences from books of famous authors, MBTI were pulled from https://www.mbtidatabase.com
  - Used unsupervised clustering with Singular-Value Decomposition, Bag of words Feed-Forward NN, and a RNN with LSTM
  - RNN with LSTM gave best results
  - Different because of the nature of the datasets

Literature Survey

- *Pandey, Animesh. "Idea of a new Personality-Type based Recommendation Engine"*
  - Dataset was self reported indicator as well as preferences for things like books, video games, music, and movies
  - No natural language processing
  - Used K-means clustering and just looked at relations

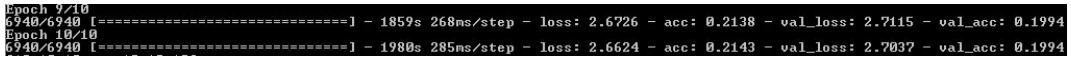
Methodology

- We first had to preprocess our data
  - We started by removing all links, ‘|||’ seperators, and URLs from the data as well as made everything lowercase
  - We then set max features of our vocabulary vector to 1000 words (padding those who didn’t have 1000 words with 0s)
  - Finally we performed One Hot encoding of the labels

Methodolgy

- RNN with LSTM
  - We used Keras to construct a RNN with LSTM
    - The training/testing features being the vocab vector, while the training/testing labels were the one hot vector
  - We tried 3 different activation functions: tanh, sigmoid, and relu
  - We also tried varying our vocabulary vector length, % of data used for training, and # of epochs to train for

Results & Future



- Accuracy after 10 epochs is ~20%
- Train multi-class for each of the four MBTI classifications to scale penalties for misclassification
- Run for many more epochs, historically many epochs have been needed to train this data
- Implement bidirectionality

Thank You!

Any Questions?