Data Genre Detection Between Fairytale and Mystery

Will Moore
?Affiliation?
jwm18575@uga.edu

YuLong
?Affiliation?
yw98883@uga.edu

Cheng Chen
?Affiliation?
Cheng.C@uga.edu

Abstract

(Collectively) We should talk about our project, methods, challenges, results.

1 Background

(Collectively) Here should have one/two paragraph on genre detection.

1.1 Gutenberg Digital Library

(Collectively) We need to write about 300 words about the Gutenberg datasets.

1.2 Genre Classification

(Collectively) We need to write about 300 words about the importance of genre classification.

Naive Bayes - Yulong Yulong should write about 300 words to describe the background of the NB approach with references.

Logistic Regression – Will Moore Will should write about 300 words to describe the background of the LG approach with references.

Neural Network – Cheng Chen Erick should write about 300 words to describe the background of the LSTM approach with references. Due to the high accuracy of prediction of artificial neural networks (ANN), we also implemented long short term memory (LSTM) algorithm to classify the those two genres.

1.3 New Hypothesis

(Collectively) Here we need to introduce our new hypothesis and the length is one paragraph.

2 Methodology

2.1 Data Pre-processing

(Collectively) We will talk about how did we pre-process our datasets. One paragraph about the bag-of-words, and another paragraph should mention word embedding method.

Data used in the data set was first selected by experts in the literary fields from the collective Project Gutenberg data set of the top one hundred authors. Data was selected on a book by book basis with criteria of the book being either of the mystery or fairytale genre by the expert. A data set of eighty four books was selected by the literary experts, of which contained fifteen fairytales and sixty nine mysteries by various authors. All books were in text format and processed using the Natural

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

Language Tool Kit, NLTK, library (reference for package source here). Each book contained a header and a footer that contained extraneous information for our experiment and were removed using python's built in input and output methods. After removal of headers and footers, books were then processed to remove punctuation, stop words, numeric words, and words with a length of one or less. All capitalization was removed from each book as well. Books were also tokenized into an array of strings, where each string represented one individual word in the book.

After the process of cleaning the books and tokenization, books were organized into a single array of documents using a bag of words approach. Utilizing the Scikit Learn method CountVectorizer, a sparse matrix containing the word counts of each unique word for each document was formed(reference for package source here). CountVectorizer was passed the parameters specifying to only take words occurring in at least five documents and at most seventy percent of documents. A minimum occurrence of five was chosen in order to eliminate words that occurred too often to be telling about the book genre they represent. A maximum of seventy percent occurrence was chosen in order to prevent any universally identifying words to influence the models used. The result of the bag of words was a ten thousand parameter array of features for logistic regression, and the results for LSTM was an one thousand paramter array of features.

Another pre-processing approach used instead of the bag of words models was Linear Semantic Analysis, LSA. CONTINUE WITH PROCESS OF LSA HERE....

2.2 Naive Bayes - Yulong

Here you should describe the details of your approach with equations.

2.3 Logistic Regression – Will Moore

Here you should describe the details of your approach with equations.

The processed data set and resulting bag of words model was run through a logistic regression model using SciKit Learn's logistic regression package . The model utilized a simple seventy percent training set and thirty percent testing set created using the SciKit Learn train-test-split method. Ground truth for the model was the established genre for the book as determined by the literary experts. The model was then fit to two different version of the bag of words. The first version used the unaltered bag. The second version ran the bag of words through the Term Frequency - Inverse Document Frequency,TF-IDF, algorithm in order to gauge the change in accuracy when looking at the importance of the word in the book to the entire corpus versus just the count of the word. The General form of the TF-IDF algorithm used was: (TF-IDF Equation Here) The logistic regression model was then optimized using the liblinear solver from Scikit Learn. Models then predicted genres using the testing set and were graded for accuracy. The equation for liblinear is as follows: (liblinear Equation here)

2.4 Neural Network – Cheng Chen

Here you should describe the details of your approach with equations.

3 Results

3.1 Accuracy

(Collectively) we should make a table or figure to compare the accuracy of our different approaches.

4 Conclusion & Future Work

(Collectively) we should conclude our work and talk about if we accomplish our hypo. Mention our findings and results again in one sentence.

References

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.