Movie Plot Classifier

Setup

To run this application, please follow the instructions in the project’s README file.

Description

This project provides a browser-based user interface that allows the user to search for a movie by title, and have that movie classified into a genre. When a search result is returned, the movie’s plot will be displayed, along with an image. Beneath the image, the movie will be classified into a genre. Additionally, genre classification probabilities will be displayed.

Architecture

The project is JavaScript based. The application’s front end is built in Vue. The application’s backend is a simple express server. The library used to perform text classification is called Natural Node. Specifically, the [Bayesian Classifier](http://naturalnode.github.io/natural/bayesian_classifier.html) is used.

To train the classification model, a [dataset](https://www.kaggle.com/datasets/hijest/genre-classification-dataset-imdb?resource=download) was found on Kaggle that matches movie plots to genres. This dataset had the following structure: ID ::: TITLE ::: GENRE ::: DESCRIPTION. To leverage the dataset, parsing was built into the training.js file.

Within training.js, a classifier is instantiated. After training data is parsed and prepped in the proper format, *documents* are added to the classifier in the form of plot:genre pairs. After all of the samples have been added to the classifier, a call to the classifier’s train method is made. With the given dataset, training took a few hours.

Usage

After the classifier trains, training.js persists it in a file called training.json. This file can then be used in the api layer, within index.js, to perform classification on new data samples.

When the user runs the program locally, the UI that they use in the browser contains a search bar. When the user searches by a movie title, the api layer utilizes the [OMDB](https://www.omdbapi.com/) API to retrieve the movie’s full plot in text and an image to display. The api layer then inputs the movie plot text into the classifier and classifies the movie into a genre. The UI will also display class probabilities produced by the classifier.

Training

Classification into a genre is largely dependent on training data used to train the classifier. This project currently contains 3 different datasets and 3 different classifiers. By default, classifier.json is set at line 11 of index.js. To try the other classifiers, line 11 of index.js can be modified.

*Classifier.json*

This classifier was created using the [dataset](https://www.kaggle.com/datasets/hijest/genre-classification-dataset-imdb?resource=download) mentioned above, which maps movie plots to genres. It works with the training file called training.js. Training this model is performed with node training.js.

*Classifier-s.json*

This classifier was created using a [dataset](https://www.kaggle.com/code/lykin22/movies-genre-classification-nlp/data?select=kaggle_movie_train.csv) that maps snippets of movie scripts to movie genres. It works with the training file called training-s-.js. Training this model is performed with node training-s.js.

*Classifier-p-s.json*

This classifier was created by combining the first two datasets. It works with the training file called training-p-s.json. Training this model is performed with node training-p-s.js.

Results

A test script was created in test/testing.js. The script is designed to have a classifier classify movies from IMDB’s Top 250 titles into genres.

*Classifier.json*

Results of testing for this classifier are found in results.txt. The main observation with this classifier is that 154/226 titles have been classified as war movies. Here it would seem that the classifier is overtrained on the war genre.

*Classifier-s.json*

Results of testing for this classifier are found in results-s.txt. The main observation with this classifier is that 212/226 titles have been classified as romance movies. Here it would seem that the classifier is overtrained on the romance genre – and with results that are much worse than classifier.json.

*Classifier-p-s.json*

Results of testing for this classifier are found in results-p-s.txt. Here war has been reduced to 147/226 titles.

Improving the Results

The most immediately noticeable problem with the classifier is that it tends to think that every movie is a movie about war. Granted, many movies are about war – and many movies in the training dataset are about war – but many movies are not. To correct for this, custom tuning was added to the api layer of the project. Within tune/util.js, a simple method called lessWar has been built. lessWar takes a movie plot, genre classification, and set of classification probabilities as inputs. If a movie has been classified as war, then lessWar will check to see if “war”, “army”, or “soldier” are actually in the plot description. If one of these words are in the plot description, then the movie will remain classified as war; if not, then the second most likely genre from the movie’s classification set will be assigned to the movie and returned as its genre. This technique brought the count of war movies done to 31/226. Results are shown in results-tuned.txt.