MOVIE GENRE PREDICTION USING SENTIMENT ANALYSIS

Mayank Rao	01FB16ECS199
Prajna Nagaraj	01FB16ECS259
Shreyas Nitin Pujari	01FB16ECS371
UE16CS333	NLP Final Project

ABOUT THE PROJECT

Project idea:

Using reviews data set of movies, the aim of this project is to predict which genre the movie will belong to along with the sentiment of the review.

UNIQUENESS AND ANALYSIS

Why you think your project is somewhat uncommon ?

Many Movie review based analysis, focus on trying to identify if the review given about a movie is positive or negative.

That is, to try and identify if the movie was liked or disliked by the person who gave the review.

This project focuses on, not only identifying if the review was negative or positive, but also predict which category it belongs to. (Adventure, Mystery, Action etc.)

DATASET SOURCE AND PREPROCESSING DONE

1. Data set source:

   □ TMDb data set

   □ IMDb (Inbuilt IMDB data set in python)

2. For RNN and LSTM , one hot encoding is used, and then average of review lengths are taken. Short reviews are zero padded and long ones are cut off.

3. The preprocessing steps performed are:

□ Reformatting the the genre column of the data set, into a readable, easy to extract and process, JSON format

□ Removed the stop words

□ Removed punctuations

□ Word tokenization, sentence tokenization

QUANTITY OF WORK - HIGH LEVEL BLOCK DIAGRAM OF OUR IMPLEMENTATION

The high level model of the experiment

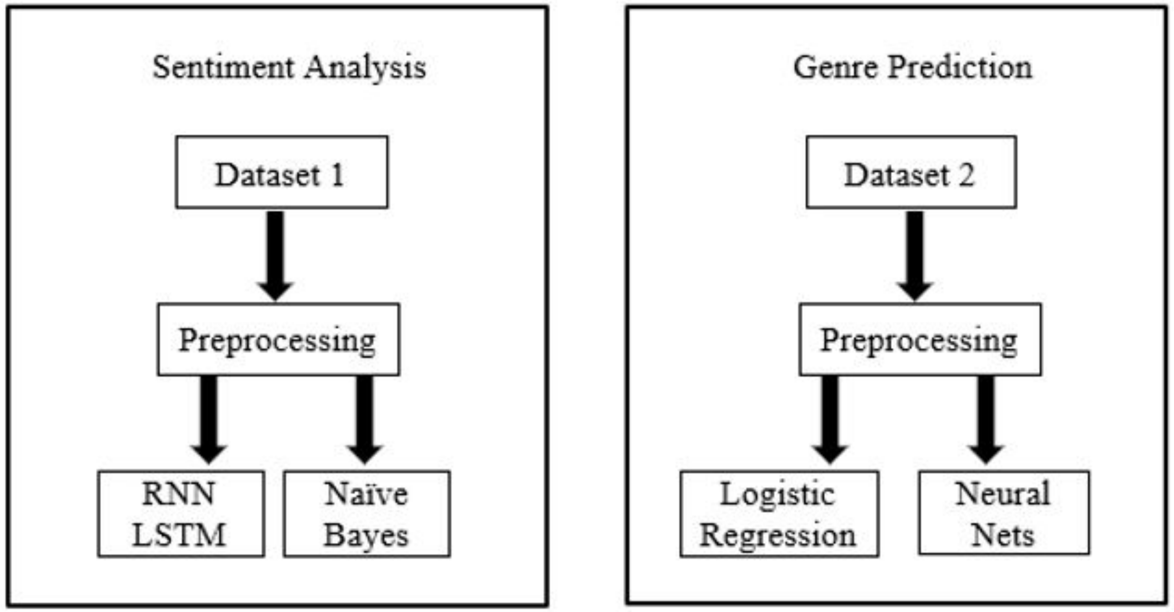

Figure:
Sentiment Classification using RNN and LSTM
Genre prediction using Logistic regression and Neural Nets

QUANTITY OF WORK – THE MAIN CODE MODULES (WHAT THEY DO)

Serial no	Code module description	Status (% complete)	What it does ?
1	Natural Language Toolkit for Python Basic processing.	100%	Stop word removal, word and sentence tokenization for pre-processing
2	Sklearn - Regression, classification and clustering algorithms support	100%	Gaussian Naive Bayes classifier.
3	Numpy - Matrix support library	100%	Matrix and vector support and shaping for Neural Networks.
4	Keras - Deep Neural Networks	100%	Tokenizer,masking, LSTM, sequence, drop out and embedding.
5	seaborn and matplotlib - Python tools for visualizing the dataset	100%	The two tools allows to visualize the dataset in a way that we prefer.
6	json, tqdm, pandas and numpy	100%	These are the basic helper modules that were used in the preprocessing of the dataset

QUALITY OF WORK – MILESTONES THAT ARE DONE AND WORKING

Serial no	Milestone description	Status (% complete)	Comments
1	Pre-processing, padding and vectorizing	5%	Involves averaging reviews as well as test-train split.
2	Vector shaping and embedding layers.	10%	Issues with deciding shape of the vectors for training.
3	RNN and LSTM trained and tested with accuracy of 87%	30%	RNN models overfit very easily.
4	Reducing overfitting and re-training model. New accuracy 86.4%	40%	Completed part 1.
5	Running Naive Bayes classifier on the data.	50%	Completed comparison with RNN
6	Applying the logistic regression model for genre classification with an accuracy of 43%	75%	
7	Running a feed forward neural network for genre classification on the data with a training and testing accuracy of 62%	100%	The feed forward network needs a little more improvement in terms of the activation units used

OUR TOP THREE LEARNING IN THIS PROJECT

1. Learning # 1 : RNN with LSTM : How to work with these types of neural networks and the advantages of RNN with LSTM as compared to other methods.
2. Learning # 2 : Naive Bayes and how it actually works. Including when attributes are not independent as required, and the loss in accuracy observed as a result.
3. Learning # 3 : Preprocessing of the data set plays a major role in deciding the accuracy of the algorithms used in prediction.
4. Learning # 4 : Fixing overfitting issues in RNN and how LSTM helps prevent vanishing and exploding gradient problem.

TOP CHALLENGES UNRESOLVED SO FAR

1. Issue #1: Only two kinds of sentiments possible. Positive and Negative. These need to be extended to more emotions.
2. Issue #2 : Most matrices for reviews are one hot encoded. They are mostly sparse matrices and therefore have a lot of zeros.
3. Issue #3 : Some reviews are extremely long - over a 1000 words, whereas some are only around one sentence. We need to therefore decide the length of the reviews to be chosen, but no proper method exists to decide properly.

OUR GOING FORWARD PLAN (IF ANY)

1. Step 1 : Use Hidden Markov Models to detect similar reviews and automatically tag them in respective genres.
2. Step 2 : Give more sentiments than positive and negative. For example, disgust, anger ,happiness etc. as emotions.
3. Step 3: Switch to multinomial Naive Bayes to classify multiple emotions as mentioned above.