CSCE 5290: Natural Language Processing
Project Proposal

Project Title:

Topic Modeling and Sentiment Analysis for movie reviews

Team Members:

Gayatri Annapurna Vadlani

Harsha Preetham Pukkalla

Harshitha Ambilpur

Rakshith Dyavari Shetty

GITHUB Link:

https://github.com/Gayatri345/CSCCE5290_ProjectNLP.git

Topic Modeling and Sentiment Analysis for movie review

Motivation:

The main goal of this project is to know the opinion by using reviews from a movie reviewing website and also to identify popular topics discussed from the reviews.

Movie reviews can be used as a data source for mining public opinion on a movie. In websites like Imdb we will have number of reviews provided for each movie. By analyzing these reviews and using text mining techniques such as topic modeling and sentiment analysis we can know about a movie. We apply topic modeling to infer the different topics of discussion and sentiment analysis is applied to determine overall feelings.

Significance:

Sentiment analysis, also refers as opinion mining, is a Natural Language Processing task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive, neutral or negative. It is a useful analysis since we could possibly determine the overall opinion about a selling object, or predict stock markets for a given company like, if most people think positive about it, possibly its stock markets will increase, and so on.

In this project we choose to try to classify reviews from IMDB into “positive” or “negative” sentiment by building a models based on probabilities and comparing models. IMDb is an online database of information related to films, television programs, home videos, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critical reviews. From these reviews we can extract overall opinions of the people on a movie.

Objectives:

In this project, we focus on applying the NLP techniques which have been learned in this course. The Python language and NLP libraries appropriate to our goals will be used in our Analysis. First, we will collect our dataset and do the analysis to have a better idea about what we are doing and the efforts needed to do before model building, such as cleaning and other preprocessing steps. After cleaning the dataset and preprocessing, we will start to build models to achieve the best results in each type of model. We aim to create at least 2 models so we can compare the results and choose the best. After choosing the best model we will try to maximize the performance of the model using usual NLP and machine learning methods. After reaching the highest performance we will use that to classify input data. We will use Transfer learning model and build a sequential model and compare both the performances. Finally, we check the custom review output prediction by feeding into the model.

Features:

1.Data

Dataset Information: https://ai.stanford.edu/~amaas/data/sentiment/

To gather the data many options are possible. Additionally, to find a way of getting a corpus of reviews, we need to take of having a balanced data set, meaning we should have an equal number of positive and negative reviews, but it needs also to be large enough. Indeed, more the data we have, more we can train our classifier and more the accuracy will be. After many researches, we choose to use dataset of 50,000 reviews from Kaggle. It is composed of two columns 'review' and 'sentiment'. Review column has the raw form of reviews and Sentiment column corresponding to label class taking a binary value, 0 if the review is negative, 1 if the review is positive .

2. Topic Modelling:

Topic modeling a method to perform unsupervised classification of collected documents and those similar to clustering on numeric data, which finds natural group of items even when we're not sure what we're looking for. In text mining, sources often have collections of various documents, such as blog posts/news articles, that we would like to divide into different groups to understand them separately.

In this project we want perform Topic Modelling on the reviews document, and extract Top_n topics that are been discussed. For this we are planning to use and compare two models, 1. BERT 2. LDA for topic modelling.

We achieve topic modeling using BERT (Bidirectional Encoder Representations from Transformers) which main purpose is to extract embeddings based on the context of the word. BERT base model uses 12 layers of transformer encoders. We use clustering and TF-IDF to create topics and words occurring in topic.

We will try to apply Topic Modeling for different combination of algorithms (TF-IDF, LDA and Bert) with different dimension reductions (PCA, TSNE, UMAP). In our analysis we expect BERT to give better results than other models.

Fig.1 Flowchart of BERT

3.Sentiment Analysis:

It is a natural language processing technique used to determine whether data is positive, negative or neutral. This will give the information whether the users are having good or bad opinion on a movie.

In this project we are planning to implement Sentiment Analysis by using SVM and LSTM models, and also use different 'loss' and 'optimizers', compare both the outputs and use the best fitting model for the final output.

We want to predict whether a review is positive ('1') or negative ('0'). We want to provide metrics of measurement for accuracy, precision, recall and generate confusion matrix for the 'positive' and 'negative' features available. For visualization we want to plot the graphs between test and train accuracies of the models.

These comparisons will help to understand which model is performing better and use the best working model.

We assume that SVM performs better as it is a transfer learning model, but we would like to fine tune our LSTM model to make it perform better than SVM and use it for final output.

Support vector machine (SVM) is a learning technique that performs well on sentiment classification.

For SVM we are planning to use transfer learning model.

LSTM is a type of RNN network that can grasp **long term dependence**.

The LSTM layers are as follows:

1. Embedding Layer: that converts our word tokens (integers) into embedding of specific size

2. LSTM Layer: defined by hidden state dims and number of layers
3. Fully Connected Layer: that maps output of LSTM layer to a desired output size
4. Sigmoid Activation Layer: that turns all output values in a value between 0 and 1
5. Output: Sigmoid output from the last timestep is considered as the final output of this network

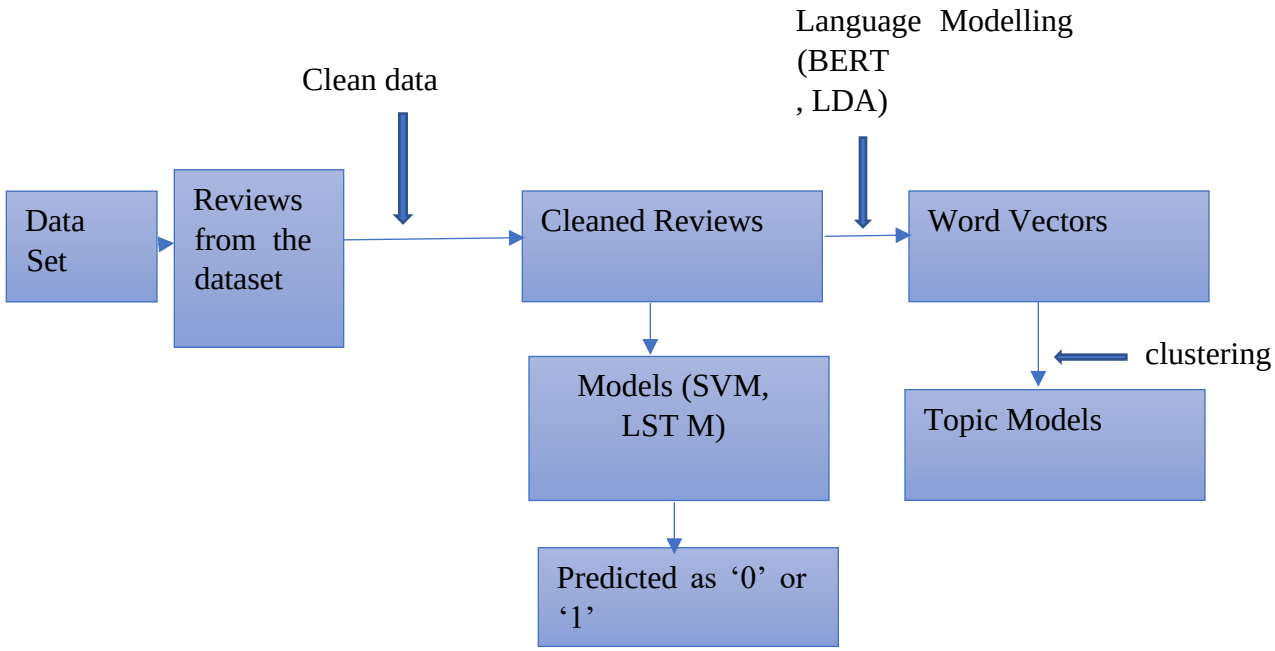


Fig.2. Flowchart

Work Plan:

SPRINTS	Module	Due
SPRINT0:	**Work Plan.**	**Nov 2**
SPRINT1:	**Data Analysis, Preprocessing**	**Nov 6-7**
SPRINT2:	**Build Algorithms for Sentiment Analysis, Train and Tune**	**Nov 13-14**
SPRINT3:	**Build Algorithms for Topic Modelling, Train and Tune**	**Nov 13-14**

SPRINT4:	**Test Models performance, test on custom reviews.**	**Nov 20-21**
SPRINT5:	**Fine Tuning models**	**Nov 20-21**
SPRINT6:	**Final Delivery**	**Nov 27-28**

Increment 1:

Dataset:

Analysis and Implementation:

We are using **IMDB dataset** from Kaggle. This dataset has 'review and 'sentiment' column. We have raw form of reviews and 'positive' for positive revie and 'negative' for 'negative' review in sentiment column.

This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. It provides a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well. Raw text and already processed bag of words formats are provided.

We decided to use Kaggle version of this dataset in csv format with total of 50,000 reviews.

Dataset Links:

https://ai.stanford.edu/~amaas/data/sentiment/

https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Implementation:

1. Reading the dataset

```
[10]: df = pd.read_csv('movie_data.csv')

      df.head(10)
```

[10]:

	review	sentiment
0	In 1974, the teenager Martha Moxley (Maggie Gr...	1
1	OK... so... I really like Kris Kristofferson a...	0
2	***SPOILER*** Do not read this, if you think a...	0
3	hi for all the people who have seen this wonde...	1
4	I recently bought the DVD, forgetting just how...	0
5	Leave it to Braik to put on a good show. Final...	1
6	Nathan Detroit (Frank Sinatra) is the manager ...	1
7	To understand "Crash Course" in the right cont...	1
8	I've been impressed with Chavez's stance again...	1
9	This movie is directed by Renny Harlin the fin...	1

We observe that the reviews need to be cleaned. So, we implemented python code using ‘re’ to clean the data.

After cleaning the reviews.

```
]: df.head()
```

]:

	review	sentiment
0	in 1974 the teenager martha moxley maggie grac...	1
1	ok so i really like kris kristofferson and his...	0
2	spoiler do not read this if you think about w...	0
3	hi for all the people who have seen this wonde...	1
4	i recently bought the dvd forgetting just how ...	0

Observing positive and negative reviews.

```
        t', 'Available']
```

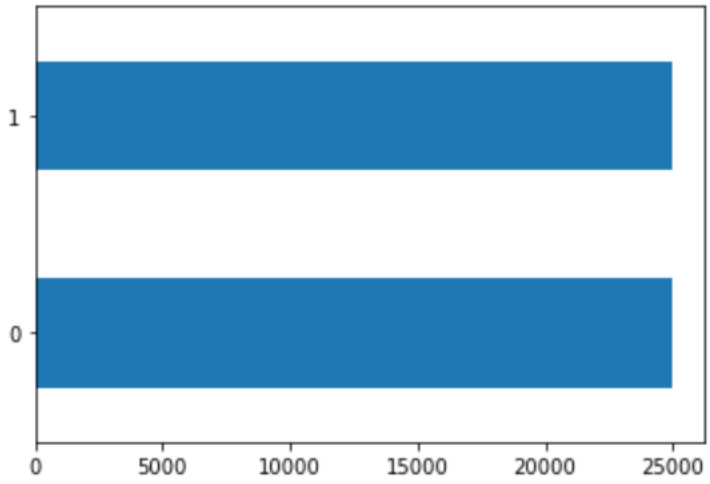
```
[106]: df.sentiment.value_counts()
```

```
[106]: 0    25000
       1    25000
       Name: sentiment, dtype: int64
```

Bar graph to analyze dataset features visually

```
]: df.sentiment.value_counts().sort_values().plot(kind = 'barh')
```

```
]: <AxesSubplot:>
```

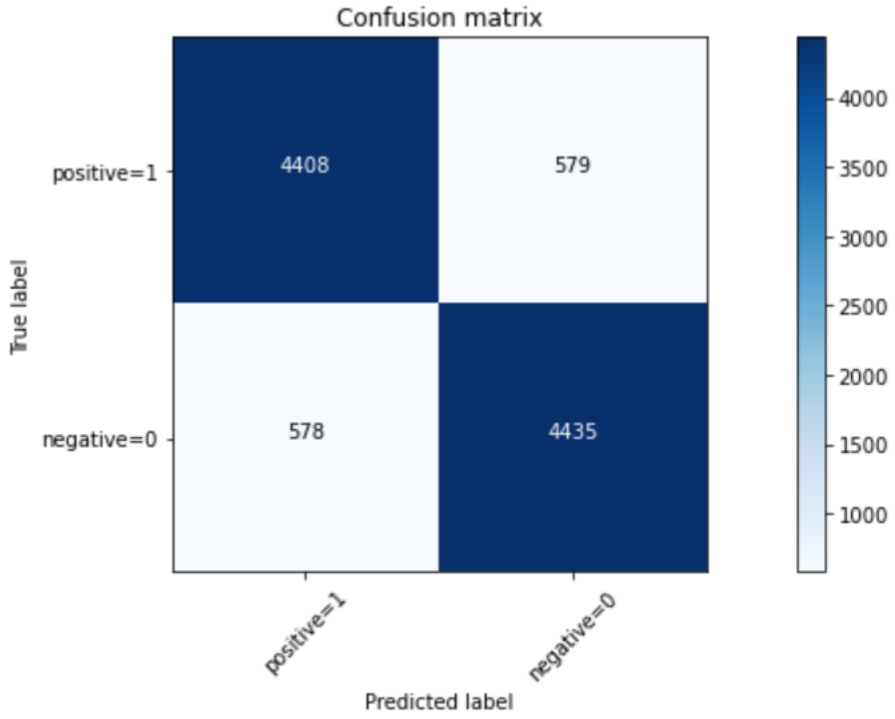


2. **Sentiment Analysis:**
   We used LinearSVC transfer learning model for our data and predicted the accuracy, we also generated confusion matrix, precision, recall and F1 scores for this model.
   This model gave us an accuracy of nearly 88%.
   We want to also implement a naïve bayes approach algorithm to do the comparison. Assuming Naïve Bayes model will have very less accuracy generated.
   We also want to experiment with LSTM and fine tuning them to achieve accuracy as good as transfer model SVM.
   Below image is the classification report for LinearSVC

```
]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.88	0.88	0.88	5013
1	0.88	0.88	0.88	4987
accuracy			0.88	10000
macro avg	0.88	0.88	0.88	10000
weighted avg	0.88	0.88	0.88	10000

Confusion matrix generated:



3. Topic Modelling:
   After going through several topic modelling techniques, we have decided to use BERT for our project. For our initial model we generated a list from all the reviews to extract the topics. The model we used for our primary analysis is BERTopic model for our initial understanding. We generated topic frequencies, which indeed generated 772 topics for our dataset (from -1 to 770).

   Below is the screenshot for the topic frequencies and Intertopic Distance map:

```
model.get_topic_freq()
```

	Topic	Count
0	-1	24017
1	0	675
2	1	635
3	2	383
4	3	327
...	...	...
756	769	10
755	765	10
754	764	10
752	766	10

Each of the topic here is having near by words, for example Topic 1 has words like worst, waste, horrible, terrible, awful. We want to implement LDA model

to check how the topics are categorized and use the best model for Topic Modelling. If time permits we want to experiment with Knn classifier to generate topics.
We assume that the results from BERT will be more accurate than LDA model. We plan to experiment with all the above mentioned models and generate the graphs for visualization of topics and compare all the models.

- ✓ **Project Management:**
  - ➢ **Implementation status report**
    - *Work completed:*

    • *Description:* The steps indicated in the table that are colored green have been completed. The orange task is what we are working on present. We have finalized the dataset and done preprocessing of the dataset. Also built a simple LinearSVC model for Sentiment Analysis and BERT model for Topic Modelling.

    •*Responsibility* (Task, Person)

    Background and references: Everyone
    Data Set: Gayatri
    Sentiment Analysis: Gayatri and Rakshith
    Topic Modelling: Harsha and Harshitha
    Writing and editing: Everone
    Video: Mahima

    • *Contributions* (members/percentage) :

    Harsha: 30%
    Harshita: 20%
    Gayatri: 30%
    Rakshith:20%

    - *Work to be completed*

    • *Description:*

    Building RNN LSTM model for sentiment Analysis. Comparing the results, fine tuning it and generating confusion matrix, predictions, classification report and compare with the SVM model. Fine tuning the model to achieve accuracy as SVM for sentiment Analysis part.

    For Topic Modelling build an LDA model and compare the results with BERT Model. After that we will use our models on custom reviews of different kinds (like complicated review which are difficult for a model to classify whether positive or negative) and check for the predicted outputs on working model .

References:

1. https://towardsdatascience.com/topic-modeling-with-bert-779f7db187e6
2. https://towardsdatascience.com/unsupervised-sentiment-analysis-a38bf1906483
3. http://ceur-ws.org/Vol-2563/aics_41.pdf
4. https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/
5. https://ai.stanford.edu/~amaas/data/sentiment/