Project Proposal - IMDB Movie Sentiment

Prepared for: Annie Lee, PHD, MMATH
Prepared by: Jairo Melo, Ing. MSC
March 27, 2019
Proposal number: ML1010-IMDB001

1. Problem Definition

Sentiment analysis is a well-known method in the world of natural language processing. The objective is to determine the polarity of of the reviews. The inputs the Movie reviews collected by IMDB, and the output is simple whether the review is positive or negative. This analysis will help us to predict whether the

2. Dataset

This is a dataset for binary sentiment classification. The IMDB dataset 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.
http://ai.stanford.edu/~amaas/data/sentiment/

The labeled data set consists of 50,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary, meaning the IMDB rating < 5 results in a sentiment score of 0, and rating ≥7 have a sentiment score of 1. No individual movie has more than 30 reviews. The 25,000 review labeled training set does not include any of the same movies as the 25,000 review test set. In addition, there are another 50,000 IMDB reviews provided without any rating labels.

3. Data Cleaning and Data Exploration

1. Remove HTML characters
2. Remove all punctuation from words.
3. Fetch all words from the movie review corpus
4. Remove all words that are not purely comprised of alphabetical characters.
5. Remove all words that are known stop words.
6. Remove all words that have a length <= 1 character.
7. Defining a vocabulary of preferred words.
8. Create a frequency distribution of all words

4. Feature Engineering

We will make use of Flair NLP Library through the below techniques to maximize the number of features:

1. Tokenization
2. Adding Tags to Tokens
3. Adding Labels or tagging sentences
4. Tagging pre-trained classification models
5. Use the classic word and character Embeddings to explore and compare with advance techniques.
6. Finally, we will use BERT and ELMO Embeddings to create contextualized word embeddings so we can feed these embeddings to our model.

We will visualize the embeddings by using principal component analysis (PCA) by leveraging TensorBoard.

5. Project First Milestone

For the first milestone, I will proceed with the previous steps: Data preparation, exploration, and feature engineering.
Then after loading the corpus we will implement two algorithms: Sequence Labeling Model and Text Classification Model.
Cross Validation: We run the Trained model using our test data; then using a confusion matrix we will analyze the results, evaluating the Precision and Recall:

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

TRUE POSITIVE : measures the proportion of actual positives that are correctly identified.
TRUE NEGATIVE : measures the proportion of actual positives that are not correctly identified.
FALSE POSITIVE : measures the proportion of actual negatives that are correctly identified.
FALSE NEGATIVE : measures the proportion of actual negatives that are not correctly identified.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

In case of an unbalance between Precision and Recall, we consider using F1 Score if there is an uneven class distribution (large number of Actual Negatives).

We finally evaluate and compare between the two models as to how many predictions are matching and how many are not by levering the confusion matrix we obtained. We will plot a validation curve in order to observe the model performance in terms of model complexity.

Acknowledge

https://github.com/google/eng-edu/tree/master/ml/guides/text_classification
https://github.com/zalandoresearch/flair/tree/master/resources/docs