Sentiment Analysis on Movie Reviews



Team Name: Movie busters (Group 9)

Team Members:

Pranay Gudur(prg7@psu.edu)

Rahul Kejriwal([rxk88@psu.edu](mailto:rxk88@psu.edu))

Zhixuan Yang (zuy73@psu.edu)

Chaoyi (David) Zhu(cxz94@psu.edu)

# Project Summary

We are given a dataset that is from rotten tomatoes, which is a website that provides reviews on various movies and tv series. Our job as data scientists is to conduct a sentiment analysis using machine learning techniques to classify the movie reviews and summarize the overall feedback on each movie title and observe how people responded. Responses can be negative, somewhat negative, neutral, somewhat positive, positive. Obstacles like sentence negation, sarcasm, terseness, language ambiguity, and many others make this task very challenging.

# Project Description

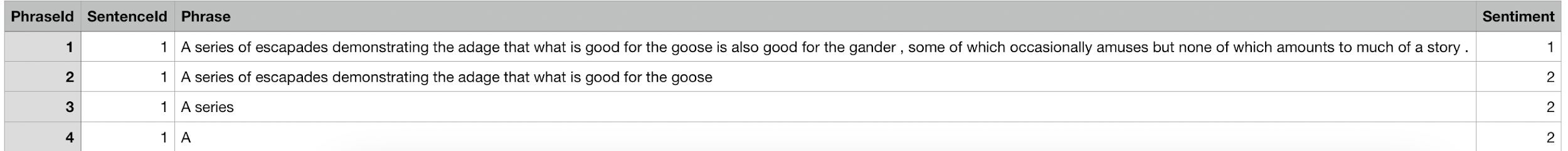
In this project, we are aiming to predict the sentiment levels of movie reviews, so this problem is a sentiment analysis problem. Sentiment analysis is the computational study of opinions, sentiments, and emotions expressed in the text (Indurkhya & Damerau, 2010) and it is a subarea of natural language processing (NLP) and data science. In order to implement the sentiment analysis, we will train a model to classify the semantic level of a movie review that was collected from The Rotten Tomatoes. The sentiment levels labeled on a scale of five values: negative, somewhat negative, neutral, somewhat positive, positive. Our goal is that the model can help us automatically label the sentiment level of a movie review. For example, a movie review writes, “This movie is a masterpiece!” and then our model will classify this review as a positive review because this statement shows that the audience has a very positive attitude toward this film.

Sentiment analysis is an interesting and important problem, as it provides other useful information regarding the quality of some certain things (e.g., a movie). While some review sites do ask the commenter to give a rating themselves, others don’t. And even for those who do, each user has a different understanding of the scale (for example, 4 out of 5 may be seen as ‘almost perfect’ by a picky user, but seen as ‘disappointing’ by another, more lenient user). Thus, it’s important to analyze the sentiment level through comments, texts, and so on. Tech companies (Google, Facebook, etc.) are interested in these solutions because they want to know what users think about their products to improve the quality of products. Additionally, business analytics companies are interested in these solutions as well because their core businesses are tied to customers’ sentiments.

The dataset is comprised of tab-separated files with phrases from The Rotten Tomatoes website. (*Sentiment Analysis on Movie Reviews*, 2020) The train/test split has been preserved for the purposes of benchmarking, but the sentences have been shuffled from their original order. Each sentence has been parsed into many phrases by the Stanford parser. Each phrase has a PhraseId. Each sentence has a SentenceId. Phrases that are repeated (such as short/common words) are only included once in the data. Therefore, each sample has four features: PhraseId, SentenceId, Phrase, and Sentiment. The sentiment labels are: 0 - negative, 1 - somewhat negative, 2 - neutral, 3 - somewhat positive, 4 - positive. This dataset has divided into train.tsv, and test.tsv, and their descriptions are following:

* train.tsv contains the phrases and their associated sentiment labels. We have additionally provided a SentenceId so that you can track which phrases belong to a single sentence.
* test.tsv contains just phrases. You must assign a sentiment label to each phrase.

Sample data look like:



To evaluate the performance of a method, we will calculate the accuracy of our classifications because the evaluation metric for this competition is classification accuracy.

In order to tackle this type of problems, we have to convert the words into vectors first because most of the data for sentiment analysis is textual, but the training model requires features. A common technique to solve this problem is word2vec (Mikolov & et al., 2013). In *Efficient Estimation of Word Representations in Vector Space*, they introduced a way that trains the words into vectors using Continuous Bag-of-Words model and Skip-Gram models through a neural network. However, feature selection and feature extraction are important steps in any natural language processing problem, and word2vec can help us solve this problem.

Medhat, Hassan, and Korashy in *Sentiment analysis algorithms and applications: A survey* (Medhat & Hassan & Korashy, 2014) tackles a comprehensive overview of sentiment analysis, and introduced some common algorithms for sentiment analysis, such as Naive Bayes algorithm, decision tree algorithm, SVM, etc. Meanwhile, it also introduced some data processing techniques like term presence and frequency, opinion words and phrases, etc. This paper primarily focuses on the probabilistic classifiers, and it introduced a method of Naive Bayes classifier on sentiment analysis problem, in which the classifier calculated the posterior probability of a class based on the distribution of the words with the bag of words techniques. Thus, we can use probabilistic classifiers and bag of words techniques on this project. This article helped us to get a basic understanding of the different algorithms and some common techniques to tackle sentiment analysis problems.

In *Semi-supervised recursive autoencoders for predicting sentiment distributions*, Socher and his colleague (Socher & et al., 2011) introduce a method of sentiment analysis that involves deep learning. This method starts from a single vector of a single word to the vectors of the whole paragraph, and we get the analysis result gradually. The advantages of this method are (1) it is able to train both on unlabeled domain data and on supervised sentiment data and does not require any language-specific sentiment lexica or parsers (2) it exploits hierarchical structure and uses compositional semantics to understand the sentiment. This article gives us an idea and method to solve the problem of sentiment analysis, which we can use in future experiments.

Overall, these three papers helped us to know a couple of techniques do people often use sentiment analysis problems, and we can apply these techniques in future experiments.

The key technical challenges of this problem are (1) feature extraction and feature selection (2) picking an ideal machine learning model (3) relatively large dataset. Since sentiment analysis is an NLP problem, feature extraction and feature selection are potential technical challenges, and word2vec could be the solution for these challenges. For picking an ideal machine learning model, we think the only way is testing different classifiers on this problem, and then we will run this experiment many times until we get the best model. The train data has 65,534 rows, which is a relatively large dataset compared to our previous experiences. So, we will CoLab to run this experiment, and we don’t have to worry about the size problem anymore.

For data cleansing, we were considering dropping the phrase IDs and sentence IDs from our training and testing datasets since they’re not directly involved in the classification. After that, we will vectorize these words using the word2vec technique to perform the next steps. One of our approaches would be to use TextBlob as a library and calculate the polarity of each sentence and add it as one of our attributes. Polarity can also be described as positivity or negativity if a sentence; therefore, the more the positive words, the higher the polarity, and the more the negative words, the lower the polarity. We believe this feature would have a strong correlation with the classes we predict. Polarity is only one of the features that we would be adding. We could be looking for the number of punctuation - let’s say a positive review could have more exclamation points, upper case words, so on and so forth. We will also be exploring different classification models like - decision tree classifier, random forest classifier, support vector machine algorithm, and maybe gradient boosting tree algorithms to see which works better. We think these approaches would be successful because our approaches are very similar to the way mentioned in our main references. For example, we will use the word2vec technique to convert words into vectors and use the algorithms that mentioned in *Sentiment analysis algorithms and applications: A survey* to train our model. That is why we believe these approaches would be successful.

Our team will meet twice a week to discuss the project and share our individual approaches on how we can solve this problem. We will examine our individual strengths and weaknesses and divide the work accordingly. Everyone as a team member will contribute equally and will be treated with respect. Initially, we will sit as a group and analyze the problem and figure out the most significant and effective solution that will provide us with the best outcome.

**Team Member Responsibilities**

|  |  |
| --- | --- |
| **Team Member** | **Responsibilities** |
| Pranay Gudur | Project Summary, Plan of team work,Team name, Project Title, Technical approaches |
| Rahul Kejriwal | Project Description, Cover Photo, Literature Review, Technical approaches |
| Zhixuan Yang | Literature Review, Technical approaches, Project problem, Project summary |
| Chaoyi (David) | Literature Review, Technical approaches, Project problem, Problem analysis |

**Group Activities**

|  |  |  |
| --- | --- | --- |
| **Date** | **Activity** | **Attendance** |
| 02/11/20 12:05 - 12:35 PM | Discussed project ideas and statistical and ML approaches | All Members |
| 02/13/20 12:05 - 13:00 PM | Discussed Project report structured and divided the work amongst team members | All Members |

Reference

Indurkhya, N., & Damerau, F. J. (2010). Handbook of natural language processing. Boca Raton: Chapman & Hall/CRC.

Medhat, W., Hassan, A., & Korashy, H. (2014, May 27). Sentiment analysis algorithms and applications: A survey. Retrieved February 16, 2020, from <https://www.sciencedirect.com/science/article/pii/S2090447914000550>

Mikolov, Tomas, Chen, Kai, Corrado, Greg, … Jeffrey. (2013, September 7). *Efficient Estimation of Word Representations in Vector Space*. Retrieved February 16, 2020, from https://arxiv.org/abs/1301.3781

Sentiment Analysis on Movie Reviews. (2020, February 16.). Retrieved February 16, 2020, from <https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data>

Socher, R., Pennington, J., Pennington, J., Huang, E. H., Huang, E. H., Ng, A. Y., … Macquarie University. (2011, July 1). *Semi-supervised recursive autoencoders for predicting sentiment distributions*. Retrieved February 16, 2020, from https://dl.acm.org/doi/10.5555/2145432.2145450