CIS 511 NLP PROJECT REPORT - Group 5

Text Sentiment Analysis for Data in The Wild

Abstract

This paper presents an experimental approach to determine the overall sentiment of Twitter tweets. We propose several models and techniques, including TF-IDF, Bag of Words, and multiple classifiers to determine which of the approaches can achieve the highest accuracy. We discuss the results of our experiments and compare them to a baseline, ultimately finding that an SVM classifier using TF-IDF yielded the best accuracy rating.

Introduction

Sentiment analysis is a largely researched problem in natural language processing, but it is one that has many possible methods and applications. In the age of social media, great importance is given to people's opinions online. As there are so many posts and opinions posted daily, it becomes necessary to develop a machine learning approach to determine the overall sentiment of such posts. Increasingly important over time are reviews that people leave, and how people feel about things on Twitter. Using data like this, research can be done into societal trends and overall sentiments.

Keeping this in mind, our project seeks to add to the research done on this topic. By using training data from a set of movie reviews, and testing our methods on a test set of tweets, we seek to investigate how effective common sentiment analysis techniques can be across platforms with completely different types of posts. By doing this, we can build towards finding the optimal method for global sentiment analysis detection. Our approach for achieving this is by comparing the accuracies of several methods. We compare the performance of a Bag of Words approach, TF-IDF with a Decision Tree classifier, TF-IDF with a Support Vector Machine classifier, and an Automated Sequence Model with the goal of determining which approach yields the highest accuracy. This paper highlights our experimental methodologies and results and presents our conclusions.

Related Work

Our project contributes to the existing body of research in sentiment analysis by building upon the existing research into understanding people's feelings on Twitter. Starting with the work of Kouloumpis et al. (2011), who looked at how specific words and hashtags can help figure out what people are feeling. Subsequent research then moved on with Saif et al. (2012) who added another layer by looking at the deeper meaning of words. In 2013, Ghiassi et al. (2013) worked on making a special list of words just for Twitter, using techniques like n-grams and number-based analysis. Then, Neethu et al. (2013) addressed the tricky parts of Twitter's language, like slang and typos, with a new way of analyzing tweets. Lastly, we utilized an approach of comparative analysis of various models to investigate the ability of common sentiment analysis techniques on a wide range of text formats.

Data Description

To analyze people's opinions, we worked with two types of data, Movie-related opinions in one set, and real-time thoughts from twitter in the other.

- **Training Data:**
  This dataset of 2000 different movie reviews came from various platforms like websites and magazines gathered by a hugging face dataset, offering a wide range of opinions about movies along with labels.
- **Testing Data:**
  The tweet data we gathered from twitter contained people's thoughts on current news, movies, or trending subjects. These were brief and casual messages where different people shared their opinions and reactions.
- **Challenge:**
  We found it hard to identify the feelings from tweets using what we knew from movie reviews because of the different words and styles, use of special characters, and jargon.

Methodology

In this project, we explored four methodologies and conducted a comparative analysis of our assessments:
Implemented a classifier using the bag-of-words approach.
The bag-of-words model serves as a feature extraction method.
Dataset Reading and Initialization: Data reading of train and test datasets. Created 2 dictionaries for positive word counts and negative counts and mapped ‘0’ as negative and ‘1’ as positive for the respective sentiments.
BOW creation (Feature extraction, cleaning, and training): Iterated through each row in the training data, the algorithm tokenizes the text into words, removes stopwords and punctuations, and updates word counts based on sentiment (positive or negative).
Data Prediction (testing): Predicted labels for twitter data based on BOW from movie data. The prediction is calculated by checking the count of positive or negative words a sentence has and its frequencies.
Accuracy calculation: Comparing predicted twitter data with the original data one.

TF-IDF technique with a Decision Tree classifier and SVM classifier.
In sentiment analysis, TF-IDF is crucial for feature extraction as it helps to capture important words. It highlights words which are frequent and distinctive across the entire dataset.
Data reading, cleaning: Removing the stopwords and punctuations.
TF and IDF calculation: TF is the number of times a word appears divided by count of the most frequent word in the document. IDF is calculated as a log scale of total number of documents divided by documents with a word. Then taking a product as TF-IDF. These are used as feature vectors.
Data training (Decision Tree and SVM classifier): These feature vectors are passed as train data to decision tree classifier and SVM.
Data prediction and accuracy calculation: Predicting the test data and accuracy calculation using the accuracy scores.

Automated sequence model.

Here, we implement a Transformer based Sentiment Classifier. A pre-trained Bert transformer classifier is finetuned with our training data. The input sequence (in our case, text sentences) is first represented as embeddings. These embeddings capture the semantic meaning of words or tokens in the input sequence. Positional embeddings are added to provide information about the position of each token in the sequence. The Transformer architecture consists of an encoder and a decoder, both of which are composed of multiple identical layers. Both the encoder and decoder layers use layer normalization and residual connections to stabilize and speed up training. The

output of the decoder is transformed into probabilities using a linear layer followed by a softmax activation. This step is crucial for sequence generation tasks. The model is trained using labeled data with a suitable loss function (e.g., cross-entropy loss). Optimization techniques like Adam are commonly used to update the model's parameters. During inference, the model can generate sequences by autoregressive predicting one token at a time, using its own predictions as input for subsequent steps.
Few of the parameters chosen for optimization are as below:
No. of Epochs: 3, BatchSize: 8, Cross Entropy Loss as evaluating criterion, Optimizer: Adam, Learning Rate: 1e-5

Experimental Discussion

We experimented with multiple approaches to solving the sentiment analysis problem. For each experiment, we divided our data into training and testing subsets using our multiple datasets. The training data set was composed of movie reviews, and the testing dataset was composed of Twitter tweets.

Our **Bag of Words model's accuracy was 55.3%** and then compared to a random guess baseline. Using the random guess baseline offered basic validation; however, in the future, comparing it to more sophisticated baselines, such as stratified random or simple heuristic-based methods, could provide deeper insights.

For the Decision Tree and Support Vector Machine classifier, we used TF-IDF as the features as before and split the features into training and testing sets. Then we trained the TF-IDF vectors with the **Decision tree classifier which achieved an accuracy of 67.82%**. For the SVM classifier, experimenting with multiple kernel types, the highest performing kernel with the metric that we measured was found to be the poly kernel, at **70.81% accuracy**. The SVM model was compared to a baseline of random guess and a dummy classifier with a strategy that included the most frequent as well as the random accuracy, the high accuracy suggests that it learned meaningful patterns from the data.

The final analysis of the three models — Decision Tree, Bag of Words, and Support Vector Machine (SVM) — reveals significant insights into the performance and applicability of each approach in the context of sentiment analysis on social media data. This comparative analysis establishes the SVM's superiority in this context and contributes to a deeper understanding of the functioning of different sentiment analysis models.
Lastly, as an additional experiment we used an automated sequence model which uses the concept of neural networks to check the accuracy. The accuracy for this model is **92.7%(train data) and 89.7% (train + validation)**. This model outperforms the BOW model, TF-IDF with Decision tree classifier and TF-IDF with SVM classifier.

Contribution

All members: Basic flow of the project, discussion on methodology, dataset research, and finalization, model classifier finalization, report, and presentation

- Ryan Ellis - Data Visualization, Code Integration, Related Works, References
- Omkar Deshpande - Support Vector Machine, TF-IDF SVM
- Kiran Davuluri - Automated Sequence Model
- Pratiksha Gaikwad - Data Preprocessing-IDF decision tree, BOW words
- Siddhi Baravkar- Data Preprocessing, BOW words, TF-IDF decision tree

Conclusion

In conclusion, our project on text sentiment analysis for Twitter data has demonstrated the complexities and challenges inherent in this domain. Through rigorous experimentation and comparative analysis, we determined that the **automated sequence model** had the best performance, with **92.7% (train data) and 89.7% (train + validation)**. This is expected as it is a neural network based approach. As for our classifiers, the **Support Vector Machine (SVM) classifier using the TF-IDF** approach provided the most accurate results, with an accuracy rating of **70.81%**. This outcome underscores the effectiveness of SVM in handling high-dimensional data and its robustness in text classification tasks. However, it's important to acknowledge that sentiment analysis, especially on a diverse and dynamic platform like Twitter, is an evolving field with room for improvement. Our experiments revealed limitations in the adaptability of models trained on one type of data (movie reviews) when applied to another (Twitter tweets), highlighting the need for more versatile and context-aware approaches.

Future work could explore the integration of more advanced natural language processing techniques, such as deep learning models, and the use of larger and more varied datasets to enhance the generalizability and accuracy of sentiment analysis models. Additionally, refining feature extraction methods and exploring the impact of contextual factors, like tweet timing and user demographics, could provide deeper insights into the nuances of online sentiment.

References

1. Kouloumpis, E., Wilson, T., & Moore, J. D. (2011). Twitter Sentiment Analysis: The Good The Bad and The OMG!. Proceedings of the International AAAI Conference on Web and Social Media
2. Saif, H., He, Y., & Alani, H. (2012). Semantic Sentiment Analysis of Twitter
3. Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter Brand Sentiment Analysis: A Hybrid System Using N-gram Analysis and Dynamic Artificial Neural Network. Expert Systems with Applications
4. Neethu, M. S., & Rajasree, R. (2013). Sentiment Analysis in Twitter Using Machine Learning Techniques. Proceedings of the Fourth International Conference on Computing, Communications and Networking Technologies
5. https://huggingface.co/datasets/imdb/viewer/plain_text/test
6. https://huggingface.co/datasets/tweet_eval/viewer/sentiment
7. https://huggingface.co/blog/sentiment-analysis-python
8. Sklearn Library: Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
9. NLTK Library: Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.
10. Wordcloud Library: https://pypi.org/project/wordcloud/
11. Seaborn Library: Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021, https://doi.org/10.21105/joss.03021.