Sentiment Analysis of YouTube Comments

M/s. Khushi Shah

MCA Department,

PES, Modern College of Engineering, Pune-05

Prof. Yogeshchandra Puranik

Assistant Professor,

MCA Department,

PES, Modern College of Engineering, Pune-05

Abstract- NLP and machine learning research now has more options thanks to the growing amount of textual data. Because of the uneven quality of the data available, sentiment analysis of YouTube comments has received less attention than it deserves. Our objective is to find patterns, identify seasonal shifts, and predict future trends by using machine learning techniques to analyze attitudes expressed in comments on popular issues. This will provide insight into how public opinion is influenced by real-world occurrences. Our results validate a strong relationship between sentiment patterns and pertinent actual occurrences.

The purpose of this study is to help academics find excellent research papers for sentiment analysis. We performed our research on a carefully selected corpus of 1,500 citation sentences that were first normalized to remove noise. Then, we developed a classification system using six machine learning algorithms: Random Forest (RF), Naive-Bayes (NB), the Support Vector Machine (SVM), Logistic Regression (LR), the Decision Tree (DT), and K-Nearest Neighbor (KNN). The system's effectiveness was evaluated through the use of measures like the Accuracy Score and F-score.

Keywords: classification; machine learning; citations; sentiment analysis

I.INTRODUCTION

This study focuses on analyzing public comments posted on YouTube videos to understand users' attitudes towards different aspects of the content they engage with. Sentiment analysis will be utilized to efficiently capture the overall sentiment conveyed in a large volume of text data, aiding in the interpretation of user opinions. With YouTube boasting a massive user base of over 1 billion unique viewers and a monthly video consumption of 6 billion hours contributing significantly to web and internet traffic. The platform offers various social mechanisms, including voting, rating, sharing, and commenting, enabling users to express their opinions effectively.

YouTube not only serves as a medium for video consumption but also fosters user interactions through features like subscriptions and comments, forming a vibrant social network. This social aspect sets YouTube apart from traditional content providers.

In this research, sentiment analysis is performed on a dataset comprising 1500 annotated comments from YouTube. These comments are categorized based on predefined rules to indicate their sentiment polarity. A system has been developed utilizing six different machine learning algorithms, namely NB, SVM, LR, DT, KNN, and RF. The accuracy of these algorithms is assessed using metrics such as F-Score and Accuracy score. To enhance system performance, various feature selection techniques such as lemmatization, n-gramming, tokenization, and the elimination of stop words and punctuation are employed.

II.LITERATURE SURVEY

Numerous researchers have conducted sentiment analysis on social networks such as Twitter and YouTube. They have examined comments, tweets, and other metadata collected from user profiles or public events to gain valuable insights into the usage of these platforms by the general population. A study closely aligned with our work was carried out by Siersdorfer et al., who analyzed over numerous comments from various YouTube videos to explore connections between comments topic categories. By developing prediction models using earlier rated comments, their findings revealed promising leads in comment rating prediction for in their natural state comments.

[1]Pang, Lee, focused on same of reviews collected from IMDb. They investigated whether sentiment analysis could be considered a specialized form of topic-based text classification.However, they noted challenges in accurately classifying sentiment due to the simultaneous presence of positive and negative expressions within reviews.

[2]Another study on YouTube comments was conducted by Shree and Brolin, who proposed an unsupervised lexicon-based approach for detecting sentiment polarity. Although their approach utilized a list of terms expressing user sentiment, they found limitations in recalling negative sentiment compared to positive, attributed to the wide linguistic variation in expressing frustration and dissatisfaction.

[3]Additional research has explored sentiment analysis on social networks like Twitter, highlighting the relationship between individual moods and various social, political, cultural, and economic events. Kowcika et al. developed a system for sentiment analysis of tweets related to smartphone wars, incorporating efficient scoring mechanisms for predicting user demographics and sentiment.

[4]Moreover, Balog et al. presented for sentiment analysis of tweets regarding smartphone wars, utilizing an efficient rating system for predicting user demographics. Kouloumpis et al. focused features capturing informal language in microblogging.

[5]Furthermore, Mishne et al. conducted sentiment analysis of web text using blog posts, while Riboni investigated website classification methods, demonstrating the effectiveness of dimensionality reduction and structured weighing techniques. Frank et al. proposed an appropriate correction method for attribute priors, showing significant improvements in classification accuracy.

[6]Lastly, Maynard et al. explored a multimodal approach to sentiment analysis on social media, aiming to assist archivists in selecting material for preservation based on semantic categories. Their rule-based textual approach accounted for challenges inherent in social media such as noisy text and sarcasm.

III.METHODOLOGY

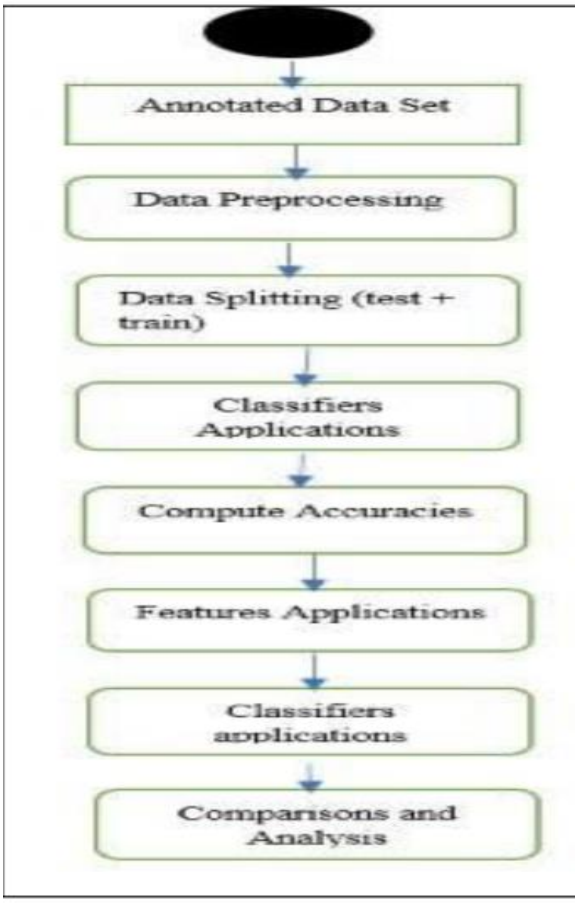
The methodology section describes the procedural steps depicted in Figure 1. Initially, we utilized an annotated dataset and implemented the process using the Python- library, Scikit-Learn. Renowned for its seamless integration with Python and intuitive interface, Scikit-Learn facilitated smooth interaction. The data was stored and read, followed by a preprocessing phase to cleanse and prepare it for machine learning algorithms. Since machine learning algorithms necessitate numeric input, the text data underwent conversion into a suitable format using the "count-vectorizer" module from Scikit-Learn, resulting in a matrix of token counts.

Subsequently, the data was divided, with 60% randomly allocated for training the classifier and 40% for testing its accuracy. The experiments unfolded in two phases. Initially, N-grams (1:3) features were applied to the data, and accuracies were computed. Then, to improve accuracy scores, additional features such as stop words and punctuation removal, lemmatization, etc., were incorporated alongside N-grams to reduce noise and data complexity. Each experiment underwent several iterations to calculate average results, totaling six experiments. Following accuracy computation in each phase, the best-performing feature and classifier were selected based on specific scenario requirements.

IV.EVALUATION METRICS.

The evaluation section holds significant importance in gauging the effectiveness and quality of a research endeavor. Here, we present an overview of the measurements employed to assess the model we developed. The evaluation of the model's performance primarily revolves around determining the accuracy of its classification outcomes. This accuracy is measured using various units.

We calculated the Macro-F Score as well as the Micro-F Score throughout our evaluation phase. False negatives (FN) are categorized as type-2 errors, whereas false positives (FP) are type-1 errors.
The F-score, a widely-utilized metric, reflects the harmonic mean between precision and recall, providing insights into the overall performance of the system.



Refer Fig. 1.

V.DATA PREPROCESSING

A total of 1500 citation sentences were meticulously compiled into the corpus used for sentiment analysis categorization, and each statement was tagged as positive, negative, or neutral based on predetermined rules. Forty percent of these sentences were assigned for testing, while sixty percent were chosen at random to train the classifier. To maximize the system's accuracy, a comprehensive cleaning process was applied to the dataset.
A. Feature Selection: A number of machine learning features, such as lemmatization, n-grams, stop words, and term-document frequency, were investigated in the development of our sentiment analysis system. These characteristics were crucial in determining the accuracy of the classifier; evaluation findings will be shown later.

B. Lemmatization: To resolve ambiguity resulting from homographic words and inflectional changes, lemmatization—a procedure of standardizing inflected word forms—was utilized. For instance, the terms "Talking," "Talks," and "Talked" are inflected variations of the word "Talk." Lemmatization was preferred over stemming even though it is similar to it in that it can handle larger datasets more efficiently.

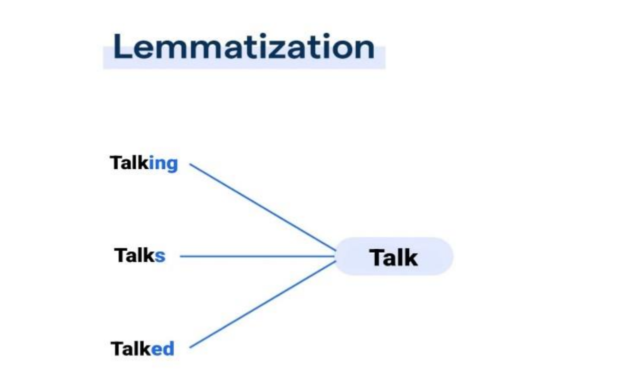


Fig. 2. Lemmatization

C. Stop Words and Punctuation: The English language contains numerous stop words, which are devoid of meaningful information and only serve to increase data size without enhancing classification accuracy. To streamline the data and improve classification efficiency, we implemented stop words removal, aligning with research advocating for this technique to reduce data dimensions.

VI.ALGORITHMS USED

This study conducted sentiment analysis using different machine learning algorithms, which are all briefly covered below:

a) Naïve Bayes: Naïve Bayes is renowned for its simplicity and effectiveness in classification tasks. Operating on Bayes' theorem, this classifier relies on probability concepts for classification. Its advantage lies in requiring relatively less training data. Pre-processing yields word-category pairings for the training set by removing numbers, foreign words, tags in HTML, and unique symbols.

b) SVM: It is a supervised algorithm known for achieving significant improvements across various tasks, including sentiment analysis. It excels in handling complex data, making it particularly suitable for sentiment analysis tasks.

c) Decision Tree: Decision trees are commonly utilized in text classification due to their interpretable classification rules. The decision-making process involves randomly selecting data from the dataset. While decision trees offer understandable prediction rules and build quickly, they may suffer from overfitting and have limitations in handling numeric attributes and missing values.

d) Random Forest: Random forest classifiers are highly regarded for their efficiency and discriminative classification capabilities. They are particularly noteworthy for their performance compared to other classifiers, making them an attractive choice for sentiment analysis tasks.

e) KNN: It is a straightforward and resource-efficient classifier, earning the moniker "lazy learner" due to its training phase, which involves storing all training examples as classifiers. However, KNN needs large amount of memory to store training information.. Based on the majority class among the K nearest neighbors, this algorithm classifies unseen data points.

Each of these classifiers offers distinct advantages and considerations, making them valuable options for sentiment analysis tasks depending on the certain specifications and features of the dataset.

VII.IMPLEMENTATION

The research showcases a standard sentiment analysis pipeline using NLP techniques and ML algorithms. Here's a step-by-step breakdown of the process:

Data Collection: Initially, a dataset comprising 1500 sentences, including positive, negative, and neutral comments, was collected from Kaggle.

Data Preprocessing: Essential libraries such as NumPy, pandas, Matplotlib, Seaborn, and NLTK were imported. The dataset was cleaned by stemming, breaking into words, reducing text to lowercase, and eliminating non-alphabetic letters., while excluding English stopwords. This process aimed to standardize the text data for analysis.

Feature Extraction: CountVectorizer from Python library was used to convert generating numerical feature vectors from text data

. It converted the text into a matrix of token counts, wherein a word from the corpus was identified by each column and a document (review) by each row. The max_features parameter specified the maximum number of features to extract.

Splitting Data: Using train_test_split from scikit-learn, the dataset was split into training and testing sets. For assessing the model's performance on untested data, this step was essential.

Model Training: Six different classification models were trained on the dataset using various machine learning algorithms. 80% of the data was used as training data, while 20% was reserved for testing. The algorithms included Gaussian NB, SVM with a linear kernel, Logistic Regression, KNN, DT, and RF. Each classifier was prepared on the training data using its respective algorithm. After training, the performance of each classifier was evaluated by generating a confusion matrix and calculating the accuracy score using the test data. This comprehensive approach allowed for a thorough comparison of the models' performance, providing insights into the most suitable algorithm for the task.

Visualization: Word cloud visualization was employed to illustrate positive sentiments. Comments with a polarity score of 1 were filtered and concatenated to form a single string. The WordCloud library was then used to create a visual representation of the most frequent words in positive comments, with stopwords removed.

VIII.RESULTS

In this work, we used a variety of machine learning classifiers, including Naïve Bayes, SVM, Decision Trees, Logistic Regression, KNN, and Random Forest, to create a sentiment analysis framework specifically for YouTube comments. Using a dataset that was divided into training and testing sets, SVM and Naïve Bayes demonstrated the greatest results. Our research revealed that the addition of uni-, bi-, and tri-gram features greatly increased the accuracy of the classifier, particularly when lemmatization was used to reduce the number of data dimensions. Using six classifiers with enlarged assessment metrics resulted in significant gains; Naïve Bayes was able to boost micro-F by 11% with bi-gram and tri-gram features and by 9% with uni-gram features. Overall, our improvements produced a maximum micro-F of 87% and macro-F of 49%, demonstrating a noteworthy advancement in our study.

	NB		SVM		LR		DT		KNN		RF	
features	Macro scores %	Micro scores %	Macro scores %	Micro scores %	Macro scores %	Micro scores %	Macro scores %	Micro scores %	Macro scores %	Micro scores %	Macro scores %	Micro scores %
A1	36	87	37	88	49	87	49	85	33	87	44	88
A2	49	83	48	86	46	88	48	85	34	87	45	88
B1	34	87	31	87	46	88	49	86	32	87	44	88
B2	46	79	47	87	46	87	48	85	34	87	46	88
C1	36	87	31	87	44	88	49	86	32	87	42	88
C2	45	77	47	87	46	87	48	85	34	87	46	88

table I- f- scores

Features	NB%	SVM%	LR%	DT%	KNN%	RF%
A1	87	87	87	87	87	88
A2	84	88	84	87	86	88
B1	87	88	88	87	87	88
B2	79	87	85	87	86	88
C1	87	88	88	87	87	88
C2	77	88	86	87	87	88

table-II- Accuracy scores

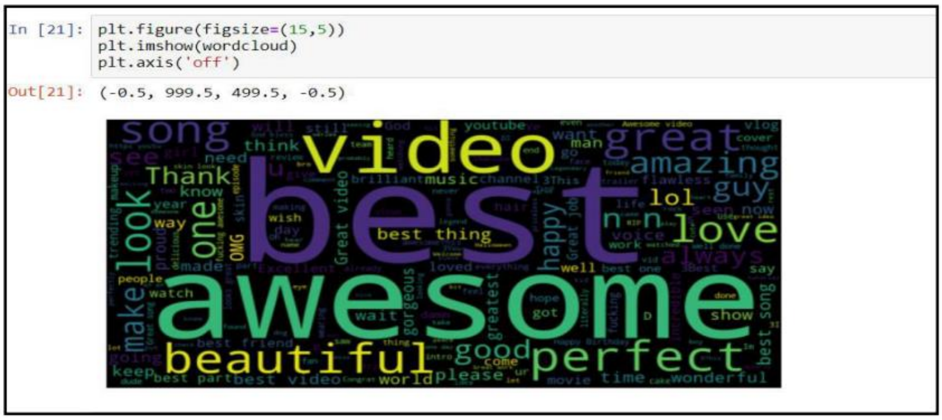


Figure-II Results of visualizations implemented on dataset.

WordCloud is a visualization technique often employed to represent textual data visually. Using this technique, words are arranged in a cloud-like structure, with the size of each word indicating how frequently or how important it is in the text.

For example, in the above dataset, “Best” appear for maximum times. Hence it reflects the highest frequency of the same followed by “Awesome”. It offers a straightforward way to identify key terms within a body of text.

As for color schemes, WordCloud allows users to customize the colors used in the visualization. This customization enables users to select colors that align with their preferences or project requirements, enhancing the visual appeal and readability of the WordCloud while ensuring it fits seamlessly with the overall design aesthetics.

IX.CONCLUSION

In this work, we used a variety of machine learning classifiers, including Naïve Bayes, SVM, Decision Trees, Logistic Regression, KNN, and Random Forest, to create a sentiment analysis framework specifically for YouTube comments. Using a dataset that was divided into training and testing sets, SVM and Naïve Bayes demonstrated the greatest results. Our research revealed that the addition of uni-, bi-, and tri-gram features greatly increased the accuracy of the classifier, particularly when lemmatization was used to reduce the number of data dimensions. Using six classifiers with enlarged assessment metrics resulted in significant gains; Naïve Bayes was able to boost micro-F by 11% with bi-gram and tri-gram features and by 9% with uni-gram features. Overall, our improvements produced a maximum micro-F of 87% and macro-F of 49%, demonstrating a noteworthy advancement in our study.

X.FUTURE ENHANCEMENTS

The focus of this research was primarily on comparing various models for identifying sentimental analysis within a localized English dataset. While supervised learning models demonstrated superior performance compared to unsupervised learning models, there's merit in exploring additional unsupervised techniques, given the slight advantage observed with the K-Means clustering model for certain feature types. Furthermore, future work could involve the amalgamation of different feature types for model training and testing.

Recognizing the limitation posed by the insufficient data volume, expanding the dataset becomes imperative for attaining more robust results. One potential approach is to employ a semi-supervised classification method to facilitate dataset annotation and model training. As highlighted in the limitations section, annotating data proves more challenging than data collection itself. Thus, by gathering a substantial amount of comments, such as 6000, and leveraging the existing 1000 annotated comments to train a classifier, we can then apply it to unannotated comments for labeling. Subsequently, each correctly labeled comment can be integrated into the annotated corpus. Through iterative iterations of this process until no further confidently labeled comments are obtained, we can streamline the creation of an annotated corpus, thereby easing the burden of manual annotation.

XI.REFERENCES

[1] Athar, A. (2014). Sentiment analysis of scientific citations (No. UCAMCL-TR-856). University of Cambridge, Computer Laboratory.

[2] Athar, A., Teufel, S. (2012, July). Detection of implicit citations for sentiment detection. In Proceedings of the Workshop on Detecting Structure in Scholarly Discourse (ppp. 18-26).

[3]Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... &Vanderplas J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research.

[4] Poria S, Cambria E, Gelbukh A, Bisio F, Hussain A (2015) Sentiment data flow analysis by means of dynamic linguistic patterns

[5] Turney PD, Mohammad SM (2014) Experiments with three approaches to recognizing lexical entailment.

[6] Parvathy G, Bindhu JS (2016) A probabilistic generative model for mining cybercriminal network from online social media: a review.

[7] Qazvinian, V., &Radev, D. R. (2010, July). Identifying non-explicit citing sentences for citation-based summarization. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 555-564). Association for Computational Linguistics.

[8]. Socher R (2016) deep learning for sentiment analysis—invited talk. In: Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis.

[9] Sobhani P, Mohammad S, Kiritchenko S (2016) Detecting stance in tweets and analyzing its interaction with sentiment. In: Proceedings of the 5th joint conference on lexical and computational semantics.

[10] The Twitter Rules [Online]. Available: https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy

[11] Facebook Comment Policy [Online]. Available: https://www.facebook.com/help/

[12] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," arXiv Prepr. arXiv1703.04009, 2017.