SHORT TEXT ANALYSIS FOR SENTIMENT EVALUATION

JiHo Lee, Ji Hyun Kim & HyeongJin Kim *
Department of Computer Science
University of Virginia
Virginia, VA 22903, USA
{jiholee,mqa4qu,jmp8nz}@virginia.edu

ABSTRACT

Traditional language models perform well with long texts but often struggle with sentiment analysis in short text datasets, lacking the ability to synthesize reviews and detect trends in emotional changes over time. In response, this paper evaluates various machine learning algorithms and BERT using a Twitter dataset to determine which model most effectively analyses short texts. Our findings indicate that traditional machine learning models outperform BERT in terms of prediction accuracy on short texts. Utilizing the top three best-performing machine learning models, we further tested its performance with real-time dataset collected from YouTube comment section, which showed slightly reduced but still reliable accuracy. To incorporate a more comprehensive interpretation on the classified sentiments, we selected the best-performing model, Naïve Bayes, and added features that can analyze sentiments on subtopics and across different time-frame. These enhancements aim to provide a more comprehensive and user-friendly sentiment analysis tool.

1 INTRODUCTION

With the exponential growth in internet usage and a corresponding increase in dependence, a vast amount of information is continuously being updated on various platforms, including reviews, news and research articles, and personal logs on social media. This outpouring information highlights the importance of understanding public opinion on various topics through sentiment analysis and promptly reflecting their feedback. Such analysis is now vital not only for the business realm but also for the broader research community, providing insights on the changing trends. For sentiment analysis, language models are considered as a primary tool due to their effectiveness in automation and accuracy. Therefore, we aim to understand how language models perform sentiment analysis and enhance comprehension of the characteristics and performance differences among various models. In this work, we extend beyond simple emotion classification to tracking real-time emotion trend over time.

2 RELATED WORK

Sentimental analysis within the field of NLP has gained wide acceptance. There has been an increasing demand for sentiment analysis from sources such as reviews and news to understand the trends and facilitate decision-making in businesses and researchWankhade et al. (2022); Nandwani & Verma (2021). The advent of large-scale language models such as GPT-3 and BERT has underscored the significance of transfer learning and pre-training in sentiment analysis. Sentiment analysis using language models primarily trains

*These authors contributed equally.

and analyzes data consisting of vast amounts of short texts such as Twitter Azzouza et al. (2020); AlBadani et al. (2022). Ahmad et al. (2019) have analyzed tweets across various domains, examining the sentiments expressed in the tweets. Arun et al. (2017) extracted opinions from a Twitter dataset and attempted to go beyond sentiment classification using R-studio to understand the overall flow of public opinion.Kiritchenko et al. (2014) investigated sentiment detection from short and informal texts to address the challenges ranging from casual writing styles such as emojis, slang, and acronyms.

Lastly, the exploration of deep sentiment analysis models continues, with an emphasis on modeling sentiment changes in time-series data. Despite the emergence of advanced models like Large Language Models (LLMs), training them to reflect the characteristics of informal and time-varying data remains challenging and high costSun et al. (2023). Therefore, Zhang et al. (2023) emphasizes that training via machine learning (ML) can be more effective, depending on the type and amount of data, as well as the capacity of device resources. Given the dynamic nature of literature trends, staying updated through academic databases and relevant conferences is advisedZhang et al. (2023); Zhou et al. (2021); Ahmad et al. (2019).

3 PROBLEM SETUP

Sentiment analysis remains a formidable challenge for Language Models (LMs), particularly in processing and interpreting the vast, varied data generated by online platforms. As we set the scope of our project, three significant issues emerged:

Efficiency and Scale of Models Traditional Large Language Models (LLMs) excel at analyzing sentiment from extensive texts due to their comprehensive training on large datasets. However, their efficiency reduces when tasked with real-time analysis of numerous short texts, such as reviews or comments. For a better result, traditional LLMs require additional training or fine-tuning which can be time-consuming. In contrast, small models that are designed specifically for short text analysis can be trained with less data yet achieve better results.

Complexity of Online Language Online platforms like Twitter are a source of numerous unstructured data characterized by neologisms, abbreviations, and emojis. The challenge lies not only in processing data with such characteristics, but also in accurately interpreting its sentiment. Relying on generic linguistic data for training may not capture the nuances of modern linguistic nuances, resulting in less efficient classification. Models trained in languages commonly used in online communities could be more accurate in their sentiment classification.

Nuanced Sentiment Classification The traditional binary classification of sentiment into positive and negative categories are not sufficient to capture the complex spectrum of the actual public opinion. Often, the sentiment expressed in user reviews and comments is not straightforwardly positive or negative but reflects a diverse range of views and emotions. A more nuanced approach is necessary to synthesize these opinions accurately and provide a comprehensive analysis of overall sentiment trends.

4 METHOD

The overarching framework of our project involves several key stages: 1) identifying and pre-processing raw data, 2) developing an optimal model using machine learning algorithms for dataset classification, 3) building a model based on BERT for dataset classification, and 4) evaluating and comparing the performance and results derived from analyzing short, unstructured texts 5) adding features and visualizations for better user interface and comprehensive understanding of sentiment analysis.

4.1 Machine Learning Algorithms

Choosing Classifier Model We actively experimented with multiple classifier models that has the potential to accurately classify short, unstructured texts utilizing both the twitter dataset and the YouTube comment section. Based on the characteristics of each classifier algorithms, we selected mainly 5 Machine Learning algorithms to explore with our short text dataset to find the one that offers the best accuracy and comprehension on the nuance of short text sentiments.

- **Naïve Bayes**: This model is particularly effective for short text analysis due to its assumption on missing features by calculating the probability of a text belonging to a certain category (like positive or negative sentiment) based on the features it contains. This approach is robust in working with concise and sparse data like short texts.
- **Linear SVC**: This model suitable for text classification with robust performance in distinguishing between different sentiments. Karthikeyan NG & Cole (2019)
- **Logistic Regression**: This model is simple and efficient, and we expect the model to serve as a baseline for sentiment analysis due to its simplicity and interpretability. P. Sujan Reddy (2021)
- **Random Forest**: This algorithm tends to have better performance with dataset that is unbalanced and missing. As it is an ensemble approach, including decisions from each decision trees, it can reduce the risk of overfitting and generalize the missing data, which is ideal for our short text sentiment analysis as it requires a good quality of generalization to classify short texts. Breiman (2001)
- **Gradient Boosting**: This algorithm effectively captures patterns from complex datasets. It improves from weak learners, focuses on instances that were not able to be classified. This characteristics can bring accurate classification on ambiguous data that requires nuanced interpretation in our short sentiment analysis texts. However, it requires attention in avoiding overfitting, by experimenting with and tuning various hyper-parameters.

4.2 BERT Classifier

BERT is also powerful for sentiment analysis, and we use BERT and RoBERT to analyze Twitter and YouTube comments to perform sentiment analysis. We expect to be able to analyze more effectively due to pre-trained language skills when prior knowledge of a specific topic is required. We also expect to improve the performance of the model by fine-tuning it with these datasets to reflect the relatively short nature of Twitter and YouTube comments. In Hsu et al. (2021), they reported that ML outperforms BERT when embedding YouTube, and our study confirms this after experimenting with a newer version of BERT and fine-tuning.

5 Experiment setup

Dataset For the experiment, we utilized two datasets: Twitter and YouTube comments. Both datasets consist of short, unstructured text, but they differ in their characteristics. The Twitter dataset was employed for training and testing both machine learning algorithms and BERT, to evaluate their effectiveness in sentiment analysis of short texts. To enhance testing, we collected comments from YouTube on a specific topic for analysis by the model trained on the Twitter dataset. This real-time dataset shares the informal characteristics of the Twitter dataset but provides more contemporary examples.

5.1 SENTIMENT140 DATASET WITH 1.6 MILLION TWEET

KazAnova (2009) To initially train and test the model's classifier, we used datasets from Kaggle that are extracted from Twitter. The text data from Twitter is suitable for our study as the length of each tweet is limited to 280 characters and includes hashtags and emojis, meaning it is short and unstructured. This dataset includes 1,600,000 individual tweets collected in 2009, which are annotated based on the sentiment of each tweet. They are labeled into two different sentiments: Negative and Positive. This dataset also includes several indices such as IDs, dates, flags, usernames, text, and sentiment labels.

5.2 REAL-TIME YOUTUBE COMMENT DATASET

For real-time data, we directly accessed YouTube comments using API resources instead of relying on pre-collected datasets. The YouTube comments, characterized by their short length and informal language including emojis, provide a type of short text similar to what we aimed to analyze with the Twitter dataset. The raw dataset collected via the YouTube API includes *author ID*, the *date* and *time* comments were published or updated, the number of likes, and the content of the comments. We employed the `Search.list` and `commentThreads().list` methods from YouTube API v3 to extract 1,080 comments from 40 trending videos related to the specific keyword *'Impact of AI'*. This real-time data not only serves as unseen data to further test our models' effectiveness in analyzing various forms of short texts but also overcomes the limitations associated with using outdated Twitter data collected in 2009.

6 SENTIMENT ANALYSIS ON TWITTER DATASET

6.1 PRE-PROCESSING DATA

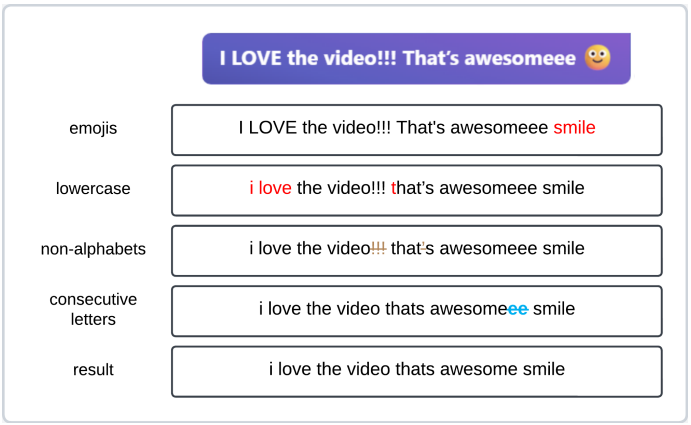

Figure 1: Example of Pre-processing Stage

Twitter Dataset For our preliminary studies, we utilized the twitter dataset both for training and testing since it already had labeled dataset and was easier to pre-process. After importing the text data, we went through some basic processing steps as Figure 1 shows, including lower-casing all words for unity, deleting unnecessary parts such as consecutive letters that are repeated at least three times, non-alphabet letters, and so on. In addition, indicators such as *urls* and *user id* and *names* were first replaced with the same term so that it can be deleted afterwards if needed. Furthermore, the emojis used in tweets were replaced with English

words that represents them, working as a valuable signal for sentiment analysis. Although we considered using nltk library to exclude stopwords, after a few times of exploration, we found out that stopwords might work as noise data for longer text, however, in short texts parsing out the stop words critically influenced the models to correctly classify the sentiment of short texts. Therefore, stop words were kept for analysis. This standardized pre-processing step ensured our models to optimally analyze and learn from the dataset.

YouTube Dataset We follow the format of the ML algorithm training and testing with Twitter dataset, which classifies sentiment based on text input. Thus, we retain only the content of the comments for processing. Since comments on YouTube usually lack the emotional context of the information related to the video, we extracted comments from videos on specific keywords that are more emotionally expressive or controversial. We assumed that this would reveal the overall clear emotional trends of the topic and provide insights. Since these texts were collected in real-time without emotional labels, so we verified the accuracy through human evaluation after testing on the trained ML models.

Common Processing For both Twitter dataset and YouTube dataset, we divided the full dataset into 70% of Training data and 30% of Testing data. For twitter dataset, due to the computational limit on running BERT on our devices, we partially used the dataset, in total of 250,000 individual tweets. The dataset was well balanced with 50% of each sentiment, 0 and 1. Then we ran a TF-IDF Vectorizer that finds unique and important terms. This helps models to not only focus on terms that simply appear often in the document, but also the ones that has more importance yet appeared less often. This considers that words that are frequent in one document but happens rare in others has the possibility of a keyword. Therefore, by using n_gram_range parameter as (1,2), TF-IDF Vectorizer was able to consider words of both single (1-grams) and pairs of consecutive words (2-grams) both as separate terms and features to find its importance compared to the other document terms.

6.2 Preliminary Results

Our preliminary studies primarily utilized a Twitter dataset to train and test the ML algorithms and BERT mentioned earlier, aiming to identify the most effective one. Initially, we evaluated five machine learning models to determine the top three performers. Subsequently, these top models were compared with BERT to assess their relative effectiveness.

6.2.1 Evaluating Machine Learning Models on Twitter dataset

Table 1: Results of 5 ML Models Classifying Twitter Dataset

	Naïve Bayes		**Linear SVC**		**Logistic Regression**		**Random Forest**		**Gradient Boosting**	
	Neg	**Pos**	**Neg**	**Pos**	**Neg**	**Pos**	**Neg**	**Pos**	**Neg**	**Pos**
accuracy	78		80		81		70		73	
precision	79	78	80	79	81	80	70	71	71	74
f1 score	78	79	80	80	80	81	71	70	74	72
recall	78	79	79	81	80	81	72	68	76	70

Initially, we assessed machine learning models on the Twitter dataset to discern each algorithm's strengths and weaknesses in analyzing sentiment of short texts. Utilizing the scikit-learn library, we experimented with Naïve Bayes, Linear SVC, Logistic Regression, Random Forest, and Gradient Boosting.

As shown in Table 1, Logistic Regression outperformed the other 5 Machine Learning Models with 81% accuracy, also with a great balance in precision and recall scores. Linear SVC also had a similarly high accuracy of 80%, along with balanced scores for precision, recall, and f1 scores. Possible reasons why Logistic Regression could have aced in classifying sentiments of twitter texts could be due to its inherent characteristics. The model is a very straightforward algorithm that is especially useful for binary classification, it is possible that this model suited best for such sentiment analysis tasks. In addition, tweets are typically short texts, and the feature vectors representing them would often be sparse. Therefore, Logistic Regression would have handled these spaces well with generalization. Lastly, our twitter dataset included emojis that are replaced with words that indicate such emotions. For instance, if someone texted ':(', our data processing part replaced them with 'sad'. This clear and distinct sentiment signals would have been a strong predictive indicator that works best with simple decision boundaries.

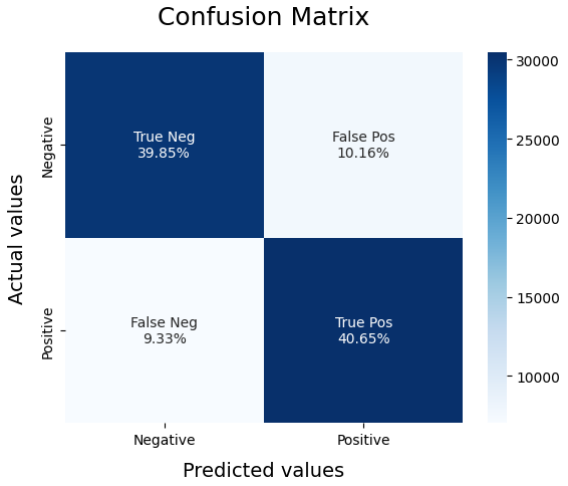

Figure 2: Confusion Matrix of Logistic Regression on Twitter Dataset

As Figure 2 indicates, Logistic Regression Model demonstrates a fairly balanced approach between identifying positive and negative instances. However, there were 7,620 cases of False Positives (10.16%) where model classified negative texts as positive and about 7,000 cases of False Negatives (9.33%) where the model classified positive texts as negative. Although 620 cases might seem minor within the context of 75,000 test cases, this difference underscores the model's tendency to misclassify negative instances as positive. Depending on the topics, such as sentiment towards a political figure or reactions to a newly launched advertisement, this error could be significant. In these scenarios, the predicted sentiment being more positive than the actual public reaction could lead to misleading conclusions about public opinion.

High Performing Models As a result of our initial testing, the top three models identified from the five machine learning algorithms were Naïve Bayes, Linear SVC, and Logistic Regression. Although Logistic Regression demonstrated the highest accuracy, we decided to use all three top models for subsequent comparisons and testing with real-time data. This decision was based on the relatively small differences in accuracy among these models and the potential for significant changes in model performance when applied to unseen, more recent datasets.

6.2.2 Evaluating BERT on Twitter Dataset

To compare the ML models with DL model, we explore well known large language model, BERT. Specifically, we used the pre-trained BERT multilingual base model which consists of 12 encoding layers. We followed the pre-processing steps mentioned in Section6.1 to train the Twitter dataset on BERT. After pre-processing, the data was transformed into BERT embeddings form. We fine-tuned the pre-trained model for our sentiment analysis taskfollowing the same process with ML model training. As a result, BERT shows 66% accuracy.

6.2.3 Model Comparison

Table 2: Model Comparison between ML and BERT on Twitter Dataset

Model	**ML algorithms**			**DL**
	Naïve Bayes	**Linear SVC**	**Logistic Regression**	**BERT**
Accuracy (%)	78	80	81	66

When it comes to comparison between the ML models and BERT, despite BERT's advanced capabilities in capturing contextual nuances in larger texts, its performance on our short-text Twitter dataset resulted lower accuracy than three ML algorithms. As shown in Table2, Logistic Regression shows the best accuracy among ML algorithms and DL algorithm. This lower performance compared to traditional machine learning models can be attributed to BERT's complexity and the nature of short texts, which provide less contextual information for BERT to leverage effectively. Machine Learning (ML) models tend to be simpler and more efficient when dealing with short texts. Deep Learning (DL) models such as transformer-based generally perform better at capturing context and dependencies between words. DL models may not provide benefit in short texts. Therefore, ML model can be a cost-effective solution since it is trained relatively quickly with less powerful hardware.

6.3 Sentiment Analysis on Real-Time Data

To validate the effectiveness of machine learning models trained on Twitter datasets in classifying sentimental analysis in short texts, we tested the models using different dataset. We selected the top three machine learning (ML) algorithms among the five that were used to evaluate the Twitter dataset, to assess their performance on a real-time YouTube comments dataset.

Table 3: Model Comparison between ML Models on Real-time YouTube Comments Dataset

Test Dataset	**ML models**		
	Naïve Bayes	**Linear SVC**	**LR**
YouTube comments	78	74	77
Twitter Dataset	78	80	81

Model Evaluation Since the real-time test data lacks labels for verifying the accuracy, we performed a human evaluation to calculate it. Currently, we have reviewed 250 data points, with additional evaluations ongoing. Among the five ML algorithms tested, Naïve Bayes and Logistic Regression showed the highest accuracy on a test dataset similar to the Twitter dataset in Table3, achieving 78%, and 77% accuracy respectively. While Linear SVC shows the highest accuracy on the Twitter dataset, Naïve Bayes showed the highest

performance on the YouTube dataset. Naïve Bayes utilizes prior probabilities for the data, which enhances its reliability when the data is sparse or has a high distribution. Thus, this implies that Naïve Bayes may perform more reliably on unseen data that has not been previously trained.

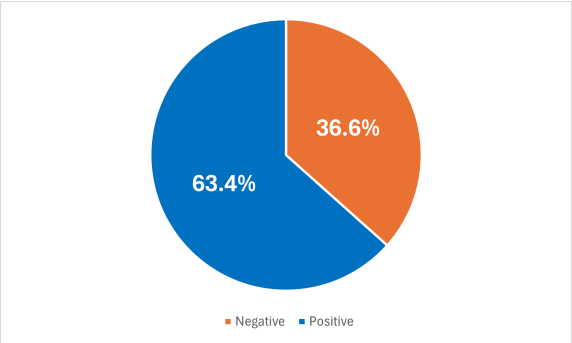

Figure 3: Sentiment on 'Impact of AI' Classified with Naïve Bayes

Sentiment Analysis Since the Naïve Bayes model exhibited the highest accuracy with unseen YouTube data, we used it to analyze the sentiment of comments on the topic of 'Impact of AI'. Figure 3 displays these results, revealing that 63.4% of the comments expressed a positive sentiment towards the potential impact of AI or the content of the YouTube video. Conversely, only 36.6% of the comments conveyed a negative sentiment, which is roughly half the number of positive responses.

7 Additional Features

After evaluating the best-performing model for short text sentiment analysis, we identified key limitations in its outcomes. Notably, the model lacks the capability to differentiate sentiments across subtopics. Users often search for broad topics that encompass multiple related subtopics, each potentially having distinct sentiments. This limitation highlights the need for a more specific approach in sentiment analysis to effectively capture and interpret the different emotions within the main topic. For instance, within the broad topic of **"Impact of AI"**, sentiments can vary significantly across different aspects. While the term **"AI"** might elicit positive reactions, the associated **"risks"** it poses to **"jobs"** could provoke feelings of worry or concern. Secondly, the sentiment analysis that our model provided cannot capture shifts in sentiment over time. Comments may be posted years after the video's release, reflecting changes in public opinion. To address this limitation and enhance multidimensional sentiment analysis, we have expanded our model to include temporal analysis and visualization features. Specifically, we utilize the Naïve Bayes model, which demonstrated superior performance in analyzing YouTube comments, to conduct sentiment analysis both on subtopics and over time. The outcomes are then visualized to provide a clearer representation of sentiment changes.

7.1 Sentiment Analysis on Subtopic

To overcome the limitation of overlooking the sentiment of specific subtopics and enabling a multidimensional sentiment analysis, we incorporated a process that filters only the comments that contains the keyword of the subtopic suggested by the user.

Keyword Extraction To select a sufficient number of comments related to a subtopic, we first collected transcripts from multiple videos of the same topic using the YouTube Transcript API. After merging multiple

transcripts into one, we calculated the relevance score of the subtopic and selected the top five keywords. To calculate the relevance score, we tried two methods; the first was to use the Spacy module. This module analyzes the grammatical structure of the given text to extract keywords that relates to the description of subtopics. In this process, we limited our keyword extraction only to nouns. The second method is to use the BERT tokenizer, which identifies semantic relationships between words through token embeddings that take into account the text as a whole, and then identifies the words that are most contextually similar to a particular keyword. It provides a more direct and specific similarity measure for a particular keyword. Out of these two methods, we chose the second method because it presented a more keywords that were highly relevant to our subtopic. We analyzed the recruitment keywords to get top five keywords ('risk' : 'potential', 'fear', 'most', 'question', 'could' / 'jobs' : 'roles', 'salaries', 'hiring', 'productivity', 'industries') as shown in the Figure 4.

```
subtopic 'risk': [('potential', 0.6495169997215271), ('fear', 0.5935340523719788), ('most', 0.528360903263092),
('question', 0.5163553357124329), ('could', 0.48472192883491516)]

['risk', 'potential', 'fear', 'most', 'question', 'could']


Top related words for subtopic 'jobs': [('roles', 0.7962190508842468), ('salaries', 0.7298945188522339),
('hiring', 0.6955376863479614), ('productivity', 0.6455509066581726), ('industries', 0.5423421263694763)]

['jobs', 'roles', 'salaries', 'hiring', 'productivity', 'industries']
```

Figure 4: Example of relevance scores and a distribution

Filtering Comments To calculate relevance scores for all comments, we used the word2vec module. We calculated relevance scores for the top 5 keywords and subtopics that we had previously obtained, giving a weight of 1.5 for subtopics. By separating the comments into words and comparing them to the keywords, we were able to get vector values for the relevance score of each comment. By plotting the distribution of each comment according to its relevance score and extracting the top 25% of comments, we were able to extract comments that were highly relevant to the subtopic and increase the accuracy of the sentiment analysis. As you can see in Figure 5, we can get relevance scores between keywords and comments and get a distribution for getting top 25% of comments.

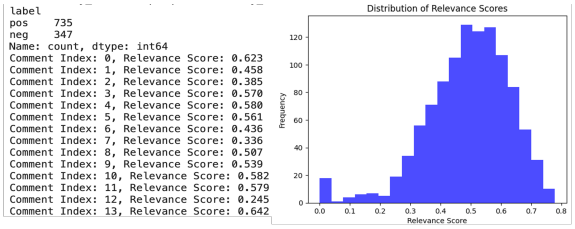

Figure 5: Example of relevance scores and a distribution

Sentiment Analysis With Filtered Comments We retested these extracted comments using the model we chose in Section 6, and the results of the subtopic task are shown in Figure 6. For both subtopics 'risk' and 'jobs', the percentage of positives is higher than in the traditional sentiment analysis, which can be inferred that people are more in agreement with the benefits of AI than the threats, and that they expect AI to have a positive impact, such as creating more jobs and helping them in their work, rather than worrying that it could lead to fewer jobs.

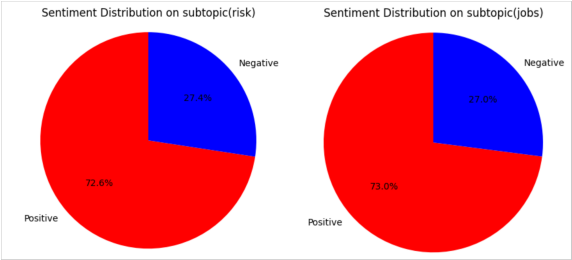

Figure 6: Sentiment Analysis on “risk” and “jobs”

7.2 Sentiment analysis through timeline

The videos curated by topic were uploaded at different times, and the comments alongside the videos were posted at different times. If sentiment analysis is conducted by integrating them without distinction, it is not possible to analyze changes in sentiment analysis over time, such as when new facts are discovered and perceptions change for the better or worse, or when sentiment changes at a specific time in the past or present. To overcome this, we imported comments from various videos, sorted them in chronological order, and analyzed them by year. We extracted comments from the seven videos used in section 6, listed them in chronological order, and categorized them into three timelines: 2024, 2023, and before 2022. Then, we conducted sentiment analysis to check the positive and negative ratios of the topics and listed them according to the timeline. The results of the sentiment analysis can be seen in Figure 7. The analysis shows that the percentage of positive sentiment toward AI increases over time. Sorting comments by time and categorizing them into specific time periods has the advantage of following sentiment changes, looking at events that occurred at inflection points to help identify the reasons for sentiment changes, and focusing on people’s recent sentiments.

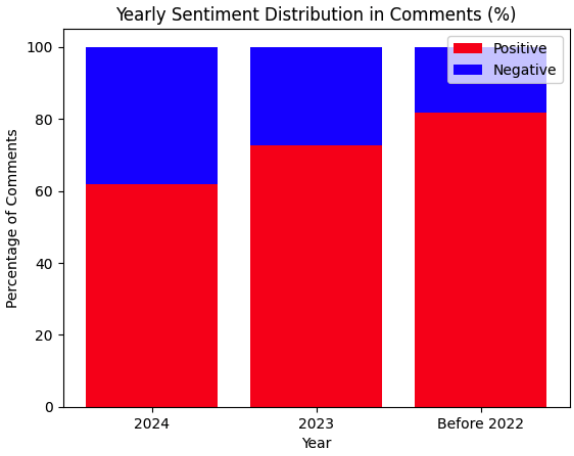

Figure 7: Sentiment Analysis on ”risk” and ”jobs”

7.3 VISUALIZATION

For a comprehensive understanding and better interpretation, the visualization of sentiment on 1) main topic, 2) subtopics, 3) changes over time, and 4) wordcloud is presented as the final result for our model. Each word cloud present keywords that have the biggest influence on each sentiment, positive and negative. This allows users to explore and interpret the reasons of each sentiment.

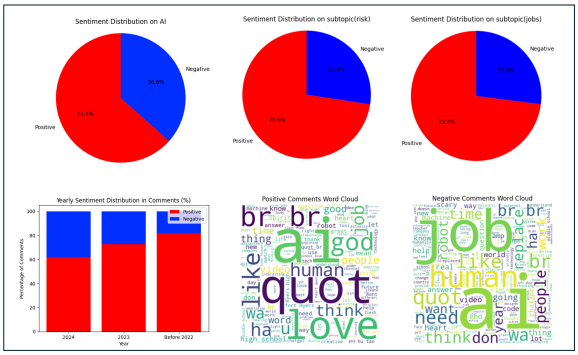

Figure 8: Example of visualization

Figure 8 is an example of the final visualization on the topic we investigated. The initial pie chart provides users with a general overview of sentiment regarding the main topic. The next two pie charts detail sentiments of subtopics that particularly interest users, derived from the main topic. A bar chart then illustrates changes in sentiment over time regarding the main topic, while a word cloud highlights specific terms strongly associated with each sentiment across the topic.

Interpretation Example Using our example of the impact of AI, the visualizations can be interpreted as follows: In general, the public holds a positive view of AI's impact. The subtopic pie charts reveal that comments related to jobs, such as salary and wages, are predominantly positive. Additionally, comments related to risks are also positive, suggesting a perception that AI poses no significant threats. However, the word cloud shows that negative comments about AI's impact frequently mention jobs. This indicates that while those with positive sentiments discuss various factors including jobs, humanity, and affection, those with negative views focus mainly on jobs, exerting a significant influence on the word cloud. This nuanced sentiment representation, particularly prominent in the word cloud, is not fully captured by the subtopic pie charts due to the overall fewer mentions. We believe that such interpretations can be possible with the visualization that the model provides, and can help communicate the details of sentiment analysis on a multidimensional perspective.

8 DISCUSSION

8.1 MORE SOPHISTICATED LABELS FOR EMOTION

When analyzing sentiment, we used the Twitter dataset to analyze both positive and negative sentiment. Recent research has provided labels for analyzing not just two emotions, but also detailed emotions such as joy, compassion, and hope for positive emotions and anger, frustration, and depression for negative emotions. If these labels are used, it is necessary to analyze whether machine learning-based models perform well or

LLM-based models perform well. If it performs well, it will be very helpful in understanding the complexity of people's emotions.

8.2 Labeled datasets for analyzing comments

We initially tried to collect live tweets by keyword via the Twitter API, but we recently learned that monetized access or legal restrictions on crawling or scraping are in place. Therefore, we switched to collecting test data from YouTube, which shares similar characteristics such as short text, emoji usage, and keywords. However, the YouTube comment test data consists of unlabeled data with no prior sentiment analysis; therefore, it is impossible to quantitatively assess the accuracy of text classification. To mitigate this issue, we manually classified about 250 comments to estimate the accuracy of our model and qualitatively analyzed the data for our study, but we will need a larger training set of labeled comments to obtain the overall model accuracy in the future.

8.3 Preprocessing comments

When analyzing sentiments in our research, We preprocessed the data using stopwords. Although this method is simple, it ignores a large number of comments, which makes it difficult to analyze sentiment on videos with few comments. In recent social media and videos, people express their emotions in a variety of ways. They use a variety of emoticons, new neologisms and abbreviations, anecdotes of related events, and even photos. These different ways of self-expression will only become more varied and frequent over time. We collected comments that excluded these, but recent research has focused on understanding different modes of expression. By leveraging these findings, we can expect to be able to analyze sentiment more accurately in the future.

8.4 A matrix by leveraging sentiment on other topics

While conducting research, we realized that there are subtopics that have various conflicting sentiments. In videos comparing home appliances or election-related videos, similar but opposite subjects appeared, which greatly affects the sentiment analysis of the overall video. However, subjects A and B are highly related in the transcript, so it is difficult to distinguish them at this stage. To solve this problem, we tried to conduct sentiment analysis on subtopics, but it is difficult to classify opposite subtopics that are too closely related. However, we think it would be helpful to have a metric that can help classify these opposite sentiments. For example, assuming that there are two candidates A and B in an election, a negative sentiment towards candidate B can be considered as a positive sentiment towards candidate A. If such a metric can be created, it will be possible to analyze sentiment towards various entities.

9 CONCLUSION

We compared sentiment analysis models for short sentences to study how to investigate people's opinions on a specific topic through social media. We compared three ML models and one deep learning model trained on Twitter dataset and found that Logistic regression achieved the highest accuracy. To apply this in real-time, we selected videos on a specific topic and analyzed the comments. We found out that the Naive Bayes model achieved the highest accuracy by a small margin. Using this model, we built additional features such as sentiment analysis for subtopic, sentiment analysis for time change, and result visualization. This made it easy for users to understand multi dimensional sentiment analysis. In the future, we propose emotional analysis, preprocessing of various expression methods, and sentiment analysis metrics for more advanced sentiment analysis.

REFERENCES

Shakeel Ahmad, Muhammad Zubair Asghar, Fahad M Alotaibi, and Irfanullah Awan. Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. *Human-centric Computing and Information Sciences*, 9:1–23, 2019.

Barakat AlBadani, Ronghua Shi, and Jian Dong. A novel machine learning approach for sentiment analysis on twitter incorporating the universal language model fine-tuning and svm. *Applied System Innovation*, 5 (1):13, 2022.

K Arun, A Srinagesh, and M Ramesh. Twitter sentiment analysis on demonetization tweets in india using r language. *International Journal of Computer Engineering In Research Trends*, 4(6):252–258, 2017.

Noureddine Azzouza, Karima Akli-Astouati, and Roliana Ibrahim. Twitterbert: Framework for twitter sentiment analysis based on pre-trained language model representations. In *Emerging Trends in Intelligent Computing and Informatics: Data Science, Intelligent Information Systems and Smart Computing 4*, pp. 428–437. Springer, 2020.

Leo Breiman. Random forests. 2001.

Ching-Wen Hsu, Hsuan Liu, and Jheng-Long Wu. A pretrained youtuber embeddings for improving sentiment classification of youtube comments. *International Journal of Computational Linguistics and Chinese Language Processing*, 26(2), 2021.

Arun Padmanabhan Karthikeyan NG and Matt R. Cole. *Mobile Artificial Intelligence Projects : Develop Seven Projects on Your Smartphone Using Artificial Intelligence and Deep Learning Techniques*. Packt Publishing, 2019.

Μαριο Μιχαηλιδη KazAnova. Sentiment140 dataset with 1.6 million tweets. 2009. URL `https://www.kaggle.com/datasets/kazanova/sentiment140/data`.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

Pansy Nandwani and Rupali Verma. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11(1):81, 2021.

C.Srikar Reddy Dr. Subhani Shaik P. Sujan Reddy, D. Renu Sri. Sentimental analysis using logistic regression. 2021. URL `https://www.kaggle.com/datasets/kazanova/sentiment140/data`.

Xiaofei Sun, Xiaoya Li, Shengyu Zhang, Shuhe Wang, Fei Wu, Jiwei Li, Tianwei Zhang, and Guoyin Wang. Sentiment analysis through llm negotiations. *arXiv preprint arXiv:2311.01876*, 2023.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780, 2022.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. Sentiment analysis in the era of large language models: A reality check. *arXiv preprint arXiv:2305.15005*, 2023.

Yuxiang Zhou, Lejian Liao, Yang Gao, Rui Wang, and Heyan Huang. Topicbert: A topic-enhanced neural language model fine-tuned for sentiment classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Github Repo The code used for this paper is provided in a repository at `https://github.com/kmjhyn/NLP-Short-Text-Sentiment-Analysis`.