

Final Report for Syslab
General Online Opinion Detector
Ethan Cheung, William Coryell, Myung Lee
5/27/2025
Yilmaz - Period 3

Table of Contents

Abstract

I. Introduction

..... 1

II. Background

..... 3

III. Applications

..... 3

IV. Methods

..... 5

V. Results

.....7

VI. Limitations

.....8

VII. Conclusion

..... 8

VIII. Future Work and Recommendations9

References

Appendix

Abstract

The internet offers many diverse perspectives, which can be difficult to navigate when indulging in the use of social media. We aimed to solve this problem by developing a system to gather this information and analyze its overall sentiment towards specific subjects in order to create a holistic view of the opinions present in the media. Using the Newspaper and Article APIs, we made a web scraper to systematically extract the text from posts and articles of popular social media websites surrounding a user-inputted topic. Afterwards, the text would be analyzed for sentiment using a DeBERTa_v3 model trained on a downstream task (Aspect-based Sentiment Analysis). This reached a fair degree of success, as our short-text and long-text models achieved ~85% and ~80% accuracy on their datasets respectively, which contained roughly 1000 entries. Our project has successfully been applied towards echo chamber detection, allowing users to formulate their own opinion without external bias from a singular flow of viewpoints.

I. Introduction

With the variety of opinions on the internet, it can be extremely tedious and headache inducing to discover and compile all the information into a general idea. However, understanding the differences in viewpoints can facilitate opportunities for learning as well as personal growth and thoughtful opinion formation. Given how overwhelming websites like X and other news outputs can be, we sought to develop a solution to this prevalent issue. This problem can easily be realized by going on any of these websites, such as reddit, and typing in a keyword or topic. These topics can generally have hundreds of relevant posts with all sorts of different thoughts, overwhelming users instantly. We found this difficulty to be especially interesting and pressing, as the attention economy continues to be entirely consumed by social media. Billions of people globally spend hours every day on popular social media apps and websites, which can be considered detrimental to humanity's progression. Many large corporations profit off of this, and continue to popularize their forms of media to gain more share in the global attention

economy. Thus, methods of simplifying media have not developed. We sought to resolve this issue by using aspect based sentiment analysis (ABSA) with NLP. Although many models and language processing tools exist to resolve the internet's abundance of information, we innovated by developing a system to search and compile text from various sources and apply ABSA to be more user friendly. This allows people without proficient technological expertise to still gain value from these models, which contributes to a growing diversity of online opinions.

II. Background

Many solutions exist to perform sentiment analysis, as it remains a prolific subject in NLP. A prominent one is to use attention mechanisms to weigh word importance with RNNs to determine sentiment (Kardakis et al., 2021). However, we found that this had severe limitations, as dropout rates were not enabled. This restricts the model's ability to generalize to unseen data, as dropout rates are used to deactivate specific neurons in the neural network, which often helps the model avoid overfitting. Another solution we discovered utilized gradient boosted Support Vector Machines to analyze the sentiment of user reviews on popular social media platforms, including X and Reddit (Khalid et al., 2020). However, we found that the applications were especially constrained, as it trained on datasets with only tf-idf (term frequency measurement) of uni-, bi-, and tri-gram as features. This differs heavily from the common datasets used for sentiment analysis, so the model had severe limitations. We also discovered a solution that performed sentiment analysis with deep CNNs and a sequential algorithm (Feng et al., 2018). This was done by extracting aspects of words and speech vectors, and then applying a sequential algorithm to obtain a sentiment annotation of a sentence. However, CNNs are notably infamous for struggling with lesser hardware, so the model had limited performance with computing power.

To perform sentiment analysis, we used datasets found on the Huggingface Library, which is generally preferred for NLP. The datasets include pieces of text, ranging from a sentence to entire paragraphs. Labels are either negative, neutral, or positive, and correspond to a given term/word in the input text, as shown in *Fig 1*.




text	category	polarity
string · lengths	string · classes	string · classes
		
23 411	361 values	74 values
\$25 prix fixe sounds like a good deal, but dinner was still \$100 for two, the portions wer...	price, miscellaneous, food	neutral, negative, negative
\$45 a head with tip for bad gnocchi, mediocre wine (as recommended by a condescending waiter,...	food, miscellaneous	negative, neutral
' When I called the waitress on it, she said that they simply couldn't serve tap water.	staff, food	negative, neutral
' When I reasoned with them that my dining partner already received her hot entree and was...	food, staff	neutral, negative
(Note: Price variations were listed nowhere--and I had asked for a to-go drink, deciding--after...	price, food	negative, neutral
(So long that it prevented me from increasing their sales by getting more possible drinks) If...	price, food, staff	neutral, neutral, negative

Fig 1. A snapshot of the dataset used for Aspect Based Sentiment Analysis.

III. Applications:

Our solution can be used whenever someone feels overwhelmed by social media, and whoever wants to stop their doom scrolling habits. Many Gen Z are addicted to social media and short form content, which negatively impacts their health and social skills. Although short form video content may be more ubiquitously consumed in recent years, Meta and X remain highly popular platforms for the sharing of ideas. Individuals who are tired of spending hours scrolling through different posts by the endless egotistical narcissistic personas that inundate these sites can simply resort to using GOOD, and summarize all of it in a TL;DR. Brainless consumption of media can really be a shame, and we believe that it's humanity's collective responsibility to thwart it. Our solution can also be used to mitigate echo chambers and get a more comprehensive perspective on a specific topic. A lot of people are in an echo chamber—specifically through social media—which is an environment where a person only encounters beliefs and opinions that reinforce their own. GOOD can help break these echo chambers by getting diverse viewpoints from across the internet and presenting them in a balanced, digestible summary.

IV. Methods:

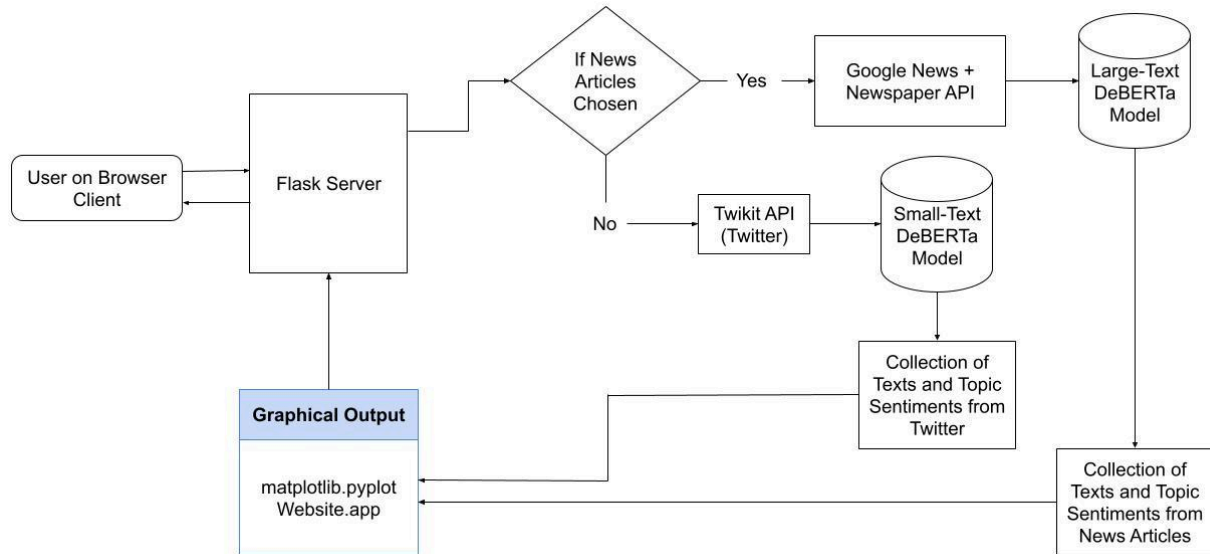


Fig 2. Systems Architecture Diagram

For the input, the user puts a topic and a source into the website. For the topic, it can be any topic that has articles or Twitter/X posts about it, which decides which model is used as shown in *Fig 2*. For the source, there is a choice between news articles and X posts.

The output displayed to the user is a graph with a sentiment calculated to each article/X post fetched by the website in relation to the topic chosen. The sentiments are on a number line that goes from -1 to 1, where -1 is the most negative sentiment and 1 is the most positive sentiment.

To get this graph, we split up our articles/posts into smaller segments and we run our ABSA model on each of the segments for the articles/posts in relation to the topic chosen. This can be seen in the pseudocode of our program (*Fig 3.*). Once we have the sentiment for each segment, the project averages the sentiment for each segment and returns it to the website to be added to the graph.

For each article/X post

Split the article/X post into segments with roughly the same length

Set sum to 0

For each segment

Run the model on the segment with respect to the topic

Add the sentiment to the sum

Divide the sum by the amount of segments

Send the overall average sentiment of the article to be graphed by the website

Fig 3. Pseudocode of program

For the small text model, we used Yelp reviews of restaurants to train and test, with a 90% 10% split respectively. For the large text model, we used both Yelp reviews of restaurants and political tweets, specifically about the Russian-Ukraine war and NATO. For both of these datasets, we once again used a 90% 10% split for training and testing. We used both Yelp reviews and a political dataset, because we needed the dataset size of the Yelp reviews in order to get enough training and testing material, but we wanted to influence the model to be better at analyzing political text, as that would be likely the majority of topics chosen in our website.

We used the Microsoft/DeBERTa-V3-Large and Microsoft/DeBERTa-V3-Base models. For the large model, there are 1024 input nodes, 24 layers, and 1024 output nodes. For the base model, there are 768 input nodes, 12 layers, and 768 output nodes.

We tried creating a single model for both large and small text, but it was always much less accurate for the large text, which caused the articles to be much less accurate. To remedy this, we split our models into small and large text, in order to have more accurate results for both twitter posts and articles.

We would like to add more varied social media sources for data, including YouTube, TikTok, and Instagram, because more sources would contribute to a nuanced and greater understanding of a subject. We would use a speech to text model to process the text, then do a sentiment analysis on it. Additionally, we would like to reduce time limitations on our ABSA model, improve our article fetching times, and increase the amount of articles used, to enhance the user experience.

If we had a chance to redo our project we would have likely tried to combine twitter post and articles together to get a better view of both sides, but in the current way we have our website operatable, we have to split between Twitter posts and articles, because running the two models at the same time is too intensive for our computers.

We use the following libraries: PyABSA, Flask, GoogleNews, newspaper, pandas, and twikit.

V. Results:

Our program consists of two models capable of handling short and long amounts of text, reaching ~85% and ~80% accuracy on their datasets respectively, which contained roughly 1000 and 10000 entries. Finally, a cumulative output is displayed on our website (*Fig 4.*), allowing users to view and read a list of summaries of the information compiled and analyzed for overall

sentiment. A graph of the sentiments is displayed for the user to assist in visualization, demonstrated in *Fig 5*.

[home](#)[github](#)[about](#)

General Online Opinion Detector 🔍

brought to you by GMI

a DeBERTa solution to polarization and opinion

Find Varied Opinions on Topics Across the Internet

Enter Topic:

Select Source:

News/Long Text Model ▾

Submit





Fig 4. Appearance of website



Fig 5. Graphical output of website

VI. Limitations:

Despite our successes with GOOD, the website does face several limitations. Processing time and computing power were two very big challenges that hindered the usability of the website. Processing time made the site sometimes take too long to load due to the computational demands of the underlying models, as DeBERTa was a very complicated process with many layers. Limited computing power also restricted the volume and speed at which data can be analyzed, often resulting in only a subset of articles being processed at any given time. Our laptops did not have a GPU and could not run the code as effectively as we wanted. Additionally, ABSA requires more detailed data, which can be difficult to obtain consistently. We trained on Yelp and Twitter posts, but if we could find a dataset compiled of more intricate and variable sources, it would have improved our models incredibly.

VII. Conclusion:

In conclusion, our product allows for readers to understand diverse perspectives by condensing data from multiple sources into a readable format. It allows for people to learn through analyzing different viewpoints and encourages the formation of unique opinions.

VIII. Future Work:

For future work, we aim to expand the range of social media sources from which data is collected, incorporating platforms like YouTube, TikTok, and Instagram to provide a more

comprehensive and representative analysis. By using a speech-to-text tool, we would extract the text from videos relating to a topic and do a sentiment analysis on it. Reducing time-related limitations is also a key goal; this includes optimizing the speed of our ABSA model and accelerating the process of fetching articles and posts. Additionally, we plan to increase the number of articles analyzed to improve the breadth and depth of insights. Diversifying the underlying model will further enhance its adaptability and accuracy across different types of content and sources, ultimately making the platform more robust and efficient.

References:

Kardakis, S., et al. "Examining Attention Mechanisms in Deep Learning Models for Sentiment Analysis." *Applied Sciences*, vol. 11, no. 9, 2021, p. 3883. MDPI, <https://www.mdpi.com/2076-3417/11/9/3883>.

Khalid, M., et al. "GBSVM: Sentiment Classification from Unstructured Reviews Using Ensemble Classifier." *Applied Sciences*, vol. 10, no. 8, 2020, p. 2788. MDPI, <https://www.mdpi.com/2076-3417/10/8/2788>.

Feng, J., et al. "Enhanced Sentiment Labeling and Implicit Aspect Identification by Integration of Deep Convolution Neural Network and Sequential Algorithm." *Cluster Computing*, vol. 22, 2019, pp. 5839–5857. [Springer, https://link.springer.com/article/10.1007/s10586-017-1626-5](https://link.springer.com/article/10.1007/s10586-017-1626-5).

APPENDIX:

- I. CODE: <https://github.com/williamcoryell/GOOD.git>