Mitigating YouTube Recommendation Polarity using BERT and K-Means Clustering

Zoheb Ahmad
Dept. of Software Engineering
Delhi Technological University
New Delhi, India
zoheb.ahmad100@gmail.com

Danish Quasim
Dept. of Computer Engineering
Delhi Technological University
New Delhi, India
danishquasim31@gmail.com

Shreeya Garg
Dept. of Computer Engineering and Bioscience
IIIT Delhi
New Delhi, India
shreeya18415@iiitd.ac.in

Vinod Kumar
Dept. of Computer Engineering
Delhi Technological University
New Delhi, India
vinod_k@dtu.ac.in

Abstract— **YouTube's recommendation system is famous for its success in maintaining high retention rates. The cause of its success is its ability to learn and predict an individual user's preferences appropriately. An unintended consequence, however, is that users get stuck in what is known as their own "echo chambers" when dealing with and feeding users back their preferences. These echo chambers can cause increasing perspective bias within users, making it difficult for users to understand differing opinions. This work aims to prepare a model that counteracts YouTube's recommendation system by forcefully exposing users to content from varying viewpoints. The SSKA pipeline (Suno Sabki, Karo Apni) is a complementary deep learning model that involves Natural Language Processing (NLP) and K-Means clustering. It utilizes modern software libraries such as the YouTube API (Application Programming Interface) for data collection and was trained and tested on a varied set of users. The results prove that the model is successful in decreasing the bias recommendation by exposing users to the content of varying opinions and helping them break away from their echo chambers. The proposed methodology of explicitly exposing users to the content of varying opinions can positively impact local societies and the global community.**

Keywords— Bidirectional Encoder Representations from Transformers (BERT), K-Means Clustering, Encoder, Google Universal Encoder (GUE), Polarity Co-efficient, Natural Language Processing (NLP)

I. INTRODUCTION

Ever since the origin of the user-centric internet, every big player has been trying to take a crack at learning user patterns and recommending them relevant content. This is done to increase user retention on their platforms. In the black-and-white days of the early 2000s, corporations such as Google, Instagram, and many more procured their learning and recommendation architectures [1]. These rudimentary architectures employed different types of language processing and pattern coupling to suggest content to users. However, even being purely text-based, these systems were often limited and were engulfed with various types of problems (such as biases or noise) [1]. The system would often suffer by giving weight to irrelevant patterns, and users would often be recommended content not particularly of interest. These problems would only be apparent for a platform as complex and intricate as YouTube.

In 2015, YouTube updated its learning and prediction architecture by incorporating deep learning [2]. In their early white paper, YouTube's team recognized three significant problems scale, freshness, and noise [2]. Scale refers to the sheer massiveness of the YouTube data set. The obstacle is that YouTube not only houses over a billion videos but has been growing at an increasing rate since its formation. Freshness refers to the need to stay true to modern trends. Users should always be provided with content that is not only relevant to their request but is closest to what is true at the time of searching. Finally, noise refers to the irrelevant pieces of data that can potentially guide the model toward drawing out contradictory or even untrue trends, patterns, and predictions. The overall YouTube recommendation and ranking of videos constituted two neural networks, candidate generation, and ranking [2]. Candidate generation takes past activity from individuals' history as input and produces a small subset from the relatively large collection. Candidate generation uses only collaborative filtering of data. The ranking algorithm finalizes the subset of videos by assigning a score to each video according to the desired objective function using a very large set of features. According to the YouTube white paper [2], millions of classes have been used to train the model. After ranking, the video with the highest score is presented first and accordingly. The amalgamation of the two methods results in a recommendation list from a very large collection of videos for everyone.

As understood, social media outlets utilize (amongst other methods) the recommendation system to increase user retention time. While modern recommendation systems are tasked with providing users with content relevant to the user, an unintended consequence of these systems is the reinforcement of "echo chambers" and widespread bias-based misinformation. The psychological presupposition [3], [4], [5] is that the above-mentioned phenomenon increases populace-based opinion polarity. Furthermore, people do not shy away from voicing their opinion on social media [6] if they truly believe in something. Effectively, content created by people is presented specifically to like-minded users, creating "echo chambers." The qualitative surveys that were conducted as a part of this research articulated said polarity. This entire phenomenon is no less a threat given the detrimental impact internet-based services have shown to have on youngsters, for example, with respect to their cognitive growth [7]. The aim is to utilize modern machine learning techniques to mitigate such biases and expose the public to diverse opinions. The speculation is that supplemented as a feature on social media platforms (YouTube for the sake of this paper), an explicit enumeration of content from differing views on a given topic can help reintroduce the said exposure. Furthermore, the hypothesis is that such exposure can bring down polarity levels amongst different users.

The first obstacle was to successfully extract meaning out of text (wherever it is being extracted from). Universally, this objective has been named "natural language processing" (NLP) and branches out to various use cases (including "sentiment analysis"). Around the 1960s, notable progress started [8] happening when researchers opened the gates to artificial intelligence-based methodologies by studying the more rule-based paradigm of natural languages [8]. Such methodologies include a rudimentary frequency count of words (coined the "Bag-of-words'" technique) and the Term Frequency-Inverse Document Frequency (TF-IDF) technique that works to reduce redundancy by ignoring common placeholder words such as "a," "the," "was," etc. However, it took the reintroduction of probabilistic and statistical methods (seen as the backbone of modern machine learning today) around the late 1980s to early 1990s [9] that really propelled the industry-level applicability of natural language processing research. It allowed researchers to incorporate the concept of error mitigation for the sake of simulating real "learning." This transition allowed researchers to dive into complex, statistically driven machine learning [9]. Roughly around early 2010 [10], researchers capitalized on the then-modern machines and started utilizing neural networks on NLP-based problems. True groundbreaking work in NLP occurred with the proposition of a technique that vectorizes words based on definition-based similarity (now commonly implemented using the Word2Vec library) [10]. The model essentially cracked the problem of understanding the context of words by introducing the continuous bag-of-words model and the continuous skip-gram model. At first, researchers jumped on the Recurrent Neural Network (RNN) bandwagon and tried to make progress. However, a major obstacle with RNNs was that they suffered long-term memory degradation. This was especially undesirable in the case of NLP.

A lasting obstacle has been opinion extraction. Online content is an immense pool of unending opinions. Much work has been done in trying to explicitly extract context-contingent words and deriving the closest approximation of intended opinion [11]. However, a revolution in general NLP occurred when researchers introduced the concept of "transformers." These architectural components work by using positional encoding, attention, and, more particularly, self-attention. While early neural networks using transformers made great progress, the most famous in the industry today are the Generative Pre-trained Transformer (GPT) [12], BERT [13], and the Text-to-Text Transfer Transformer (T5) [14] models, as shown in Fig 1.

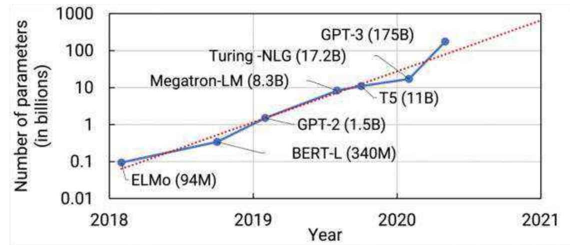

Fig. 1. Comparison of NLP-Models

Finally, the other relevant machine learning problem type is the unsupervised clustering objective. This problem type requires relative clustering of input data based purely on said data. The dimensionality of the clustering plane depends on the number of features. Researchers were quick in procuring a strong clustering model when they developed the K-Means clustering architecture [15]. This architecture, like most other architectures, takes in as many input features as needed and formulates a Cartesian coordinate for each entity. The so-called" clusters" form when entities (that share certain similar drivers) end up having coordinates closer to each other than entities that have genuinely different drivers. However, it also takes a constant "k" value that determines exactly how many clusters the given input data is to be divided into. Proposing the SSKA model; an amalgamation of the NLP BERT and the K- Means clustering architectures, the model utilizes state-of-the-art machine learning algorithms to classify videos relatively based on the user's choice, the topic at hand, and the public opinion. The model was then implemented as the brains of an entire Google Chrome extension that would detect when a user is currently watching a controversial video. It then prescribes an evenly divided set of random videos for different labels where the general idea behind the labelling is an indexing of different opinions and a potentially singular "neutral" opinion. As mentioned, the hypothesis is such that such an explicit exposure of videos with differing opinions will bring down opinion polarity levels amongst the public.

II. ARCHITECTURE

Two subject matters were selected to procure the architecture, as shown in Fig 2, and ensure the model gets trained on recent development. Since the SSKA recommendation system works entirely on YouTube, the first order of business was to gather relevant videos for each topic. This was made possible by utilizing the official YouTube API.

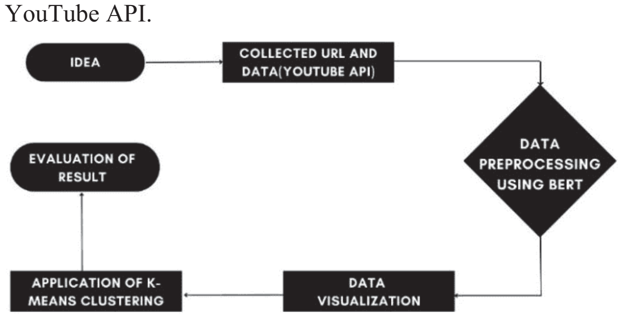

Fig. 2. Architecture of the SSKA Architecture

Utilizing the YouTube API returned 575 video uniform resource locators (URL) for Topic 1 and 519 video URLs for Topic B. To output, the model mapped each video to one of three labels (this applies separately to each topic). Three videos were manually added, each discernibly representing an opinion (for each topic). The number of videos for Topic 1 was 578 and for Topic 2 was 522. The need for segregating exactly one video for each label will be explained later. These URLs were then saved in a list based on the topic they were on. Using the lists and going back to the YouTube API, all relevant text items were extracted from each video and stored (with respect to their videos) in their respective lists. At this point, the Topic 1 list constituted a size of 578 x 4, and the Topic 2 list constituted a size of 522 x 4, where the columns represent the title of the video, an identification value, the tags, the description, and the top five comments (based on likes).

The columns (other than the identification value) for each topic were then separately (based on topic) passed through the Google Universal Encoder. Primarily based on the BERT NLP algorithm, the GUE makes use of various modern techniques to vectorize textual input. As explained above, BERT makes use of positional encoding and self-attention to maximize context-based definition derivation of each word passed into it. In return, the GUE returns a complete enumeration of the words for each video. The dimensions of this enumerated list were 578 x 512 for Topic 1 and 522 x 462 for Topic B, where each row represented a video, and the corresponding columns made up the textual sentiment embedding. The number of columns (the size of the second dimension) is completely procured by the GUE's BERT model. The textual sentiment embedding is, in fact, the final features list and can be used in the next step of the machine learning architecture. However, as it can clearly be seen, the size of the features needs some space-based optimization. This is because, while modern hardware can easily handle the features list currently available, not only can the model work faster with a smaller feature size, but optimizing the size of the features list can help make it easier for the model to focus on only the most relevant parts of each feature. Reducing the feature size should help the model home in on the true drivers behind opinion-relevant sentiment rather than be distracted by the irrelevant "noise" (as it is called) surrounding said drivers. To mitigate all potential noise, the model utilizes Principal Component Analysis (PCA) on the enumerated data. The final dimensions of the features list of Topic 1 and Topic 2 were 578 x 288 and 522 x 260, respectively. Now, the encoder was ready to hand over the input data to the K-Means clustering model. While most of the difficult work with NLP was taken care of by the GUE, one of the best ways to show relative clusters was by simply utilizing K-Means. Fig 3 shows the K-Means clustering visualization of videos from the Topic 1 data frame for all the labels (each being color coded). It is important to note that the axis doesn't hold any contextual meaning relevant to the videos, their textual encoding, or the knowledge extraction from the textual encodings. Since the clustering was relatively successful in having little to no outliers, it wasn't necessary to procure and utilize a modified algorithm inspired by [16]. As mentioned, the architectural implementation was done on each topic separately. K-Means clustering simply recognizes trends between videos and allots similar videos closer to each other.

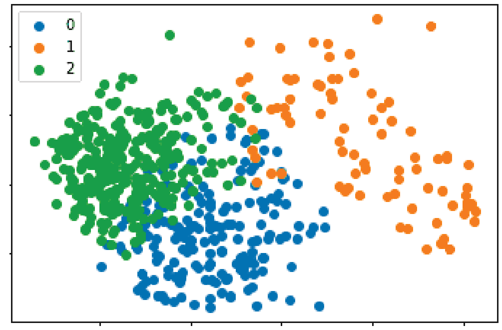

Fig. 3. Clusters formed on videos after applying SSKA from Topic 1

However, the K-Means clustering algorithm has no understanding of what the clusters mean. Thus, even a clear visual output without context meant there was no way to objectively discern which color represents which opinion. In the beginning of the architecture, when three videos were manually appended to the data frame of the topic (a video representing the extreme of an opinion), it came in handy to solve this lack of translation. Since the opinion name of the three videos was known, the derivation process can be reversed and the color output from the clustering algorithm can be renamed to reflect the opinion. This means, to return the actual opinion label for an output, the color labels returned by the K-Means algorithm must be renamed to the opinion they represent based on what the opinion of their respective "extreme" video holds. These extreme videos can be thought of as analogous to the "support vectors" found in the Support Vector Machine algorithm. This marks the completion of the machine-learning model. At this point, videos remotely relevant to the subject may be appended and the algorithm is expected to be able to map the video voicing a similar opinion or sentiment. This is because the clustering model works entirely based on the possibility of opinion derivation from the encoded textual data provided.

III. Implementation

Python data management and analysis libraries were imported and utilized. A script was then run to extract videos with respect to each of the two topics chosen. The videos were then stored in their respective data frames. An "extreme" video that could be arguably used to "represent" one of the views for each opinion is also appended to its respective data frame. Furthermore, a separate set of lists (one for each topic) was created that stores the label to the video believed to be representing it. With the complete list of YouTube video URLs, the YouTube API was used to extract textual data points for each video. The data points were stored related to their respective videos by appending columns to the two data frames. At this point, the data frames simply allocate each video with the text that makes up their title, their tags, their description, and their comments (top five based on likes). After cleaning up the data frames (removing null values and ensuring appropriate dimensionality), the model separately fed said data frames into the Google Universal Encoder. The GUE can take as input any form of textual data. While early BERT implementations used sentence-based embeddings, the model has advanced to the ability to take even a list of varying lengths of text (words, sentences, and even paragraphs). The BERT model ran multiple epochs that then started embedding the words of each textual data as one matrix (per video). The matrices are then passed back, and the original data frame is readjusted to accommodate the change in dimension. Now, for each topic, the videos in their respective data frames are paired with the sentiment-based textual embedding that has, within it, patterns matching videos with similar sentiments. This embedding is crucial (and is literally the very backbone of) the entire process of clustering the videos. However, as explained in the architecture, it was necessary to optimize the embeddings by extracting only the driving components and removing irrelevant components. To accomplish this, the Principal Components Analysis method from the "sklearn" module was imported, each data frame was passed through the PCA, and an optimization was observed (effectively decreasing the size of the second dimension for each data frame). Finally, from the "sklearn.cluster" module, the K-Means clustering model was imported. Since three opinions for each topic was decided upon (two completely opposite views/labels and one as the "neutral" view/label), the "n clusters" argument was set to three and the data frame fitting process was executed. One last time, the data frames were readjusted to have, instead of textual embeddings paired with each video, a final label representing where the video falls. However, the integer values being displayed in the labels' column don't hold any contextual meaning. To provide context, the list made in the beginning of the process, which stored an "extreme" video, labelled according to the opinion it represented, is required. Utilizing this list, it is possible to name the data frame each video is in with the label each video has been allotted (in the earlier list).

IV. RESULTS AND DISCUSSION

To finally test the effectiveness of the model, it needed to be exposed to the public. After procuring the final clustering, the model was integrated with the front end and brought test subjects to use YouTube with and without the model. The front end would simply recommend a set of random videos that each belong to one of the labels. As explained, while YouTube brings recommendations strictly based on the patterns of personal preference for the user, the recommendation system lists videos belonging to each of the different labels prescribed. The test subjects were allotted to either one of the topics. Before using the extension, the users were tasked with watching videos YouTube was recommending them on the topic they were allotted for a constant period. Tabs were kept on how many videos of each opinion YouTube recommended for the test subjects. Based on the number of videos suggested, the polarity coefficient was calculated between the different videos.

Videos of one opinion were labelled "A," the opposing opinion labelled "B," and the neutral opinion labelled "N." Also, the number of videos under label A were denoted as vA, videos under label B denoted as vB, and videos under label N denoted as vN. The coefficient of polarization is calculated for each user by the given equation in (1).

$$|vA - vB| / (vA + vB + vN)| \qquad (1)$$

Graphing the count of recommended videos (based on opinion) and the derived coefficient of polarity for each test subject results in the relations seen in Table I.

TABLE I. RESULTS WITHOUT EXTENSION

ID	Age	Gender	Topic	Videos of Opinion A (vA)	Videos of Opinion B (vB)	Videos of a Neutral Opinion (vN)	Polarity Coefficient
1	49	Male	Article 370	11	0	1	91.67
2	49	Female	Article 370	3	13	0	62.5
3	16	Female	US Elections	2	13	1	68.75
4	54	Male	Article 370	6	1	0	71.43
5	25	Female	US Elections	2	9	1	58.33
6	18	Male	Article 370	4	9	3	31.25
7	51	Male	Article 370	2	8	6	37.5
8	42	Male	Article 370	6	3	1	30
9	36	Female	Article 370	10	4	3	35.29
10	55	Male	US Elections	13	2	10	44
11	20	Male	Article 370	5	0	4	55.56
12	25	Male	Article 370	10	4	7	28.57
13	21	Female	Article 370	1	8	10	36.84
14	38	Male	US Elections	1	9	3	61.54
15	19	Female	US Elections	6	4	9	10.53
16	30	Male	Article 370	3	0	3	50
17	27	Male	Article 370	9	2	5	43.75
18	46	Female	US Elections	7	3	5	26.67
19	47	Female	US Elections	11	3	1	53.33
20	35	Male	Article 370	8	2	3	46.15
21	30	Female	Article 370	2	9	6	41.18
22	28	Male	Article 370	8	5	3	18.75
23	52	Male	Article 370	1	7	1	66.67
24	36	Female	Article 370	3	9	2	42.86
25	24	Male	Article 370	9	5	8	18.18
26	49	Female	Article 370	10	2	4	50
27	38	Female	US Elections	4	12	6	36.36

TABLE II. RESULTS WITHOUT EXTENSION

ID	Age	Gender	Topic	Videos of Opinion A (vA)	Videos of Opinion B (vB)	Videos of a Neutral Opinion (vN)	Polarity Coefficient
1	49	Male	Article 370	7	4	3	21.43
2	49	Female	Article 370	5	1	6	33.33
3	16	Female	US Elections	2	8	2	50
4	54	Male	Article 370	7	3	0	40
5	25	Female	US Elections	5	4	7	6.25
6	18	Male	Article 370	4	6	2	16.67
7	51	Male	Article 370	6	3	5	21.43
8	42	Male	Article 370	4	4	6	0
9	36	Female	Article 370	4	9	7	25
10	55	Male	US Elections	5	7	7	10.53
11	20	Male	Article 370	6	5	3	7.14
12	25	Male	Article 370	8	3	10	23.81
13	21	Female	Article 370	5	4	9	5.56
14	38	Male	US Elections	6	4	7	11.76
15	19	Female	US Elections	6	5	7	5.56
16	30	Male	Article 370	4	7	3	21.43
17	27	Male	Article 370	5	6	8	5.26
18	46	Female	US Elections	3	4	5	8.33
19	47	Female	US Elections	8	5	7	15
20	35	Male	Article 370	4	6	6	12.5
21	30	Female	Article 370	5	7	8	10
22	28	Male	Article 370	4	4	4	0
23	52	Male	Article 370	6	3	7	18.75
24	36	Female	Article 370	4	5	4	7.69
25	24	Male	Article 370	3	5	4	16.67
26	49	Female	Article 370	7	6	4	5.88
27	38	Female	US Elections	6	8	14	7.14

TABLE III. ABSOLUTE AND PERCENTAGE REDUCTION IN POLARITY AFTER USING THE EXTENSION

ID	Age	Gender	Topic	Reduction in Polarity after Using Extension	Percentage Reduction in Polarity After Using the Extension
1	49	Male	Article 370	70.24	76.62
2	49	Female	Article 370	29.17	46.67
3	16	Female	US Elections	18.75	27.27
4	54	Male	Article 370	31.43	44
5	25	Female	US Elections	52.08	89.29
6	18	Male	Article 370	14.58	46.66
7	51	Male	Article 370	16.07	42.85
8	42	Male	Article 370	30	100
9	36	Female	Article 370	10.29	29.16
10	55	Male	US Elections	33.47	76.07
11	20	Male	Article 370	48.42	87.15
12	25	Male	Article 370	4.76	16.66
13	21	Female	Article 370	31.28	84.91
14	38	Male	US Elections	49.78	80.89
15	19	Female	US Elections	4.97	47.2
16	30	Male	Article 370	28.57	57.14
17	27	Male	Article 370	38.49	87.98
18	46	Female	US Elections	18.34	68.77
19	47	Female	US Elections	38.33	71.87
20	35	Male	Article 370	33.65	72.91
21	30	Female	Article 370	31.18	75.72
22	28	Male	Article 370	18.75	100
23	52	Male	Article 370	47.92	71.88
24	36	Female	Article 370	35.17	82.06
25	24	Male	Article 370	1.51	8.31
26	49	Female	Article 370	44.12	88.24
27	38	Female	US Elections	29.22	80.36

After some time, a qualitative survey of all the test subjects was taken and their opinions were mapped to their allotted topic. Then, the users were tasked with watching videos on the same subject, but now with the SSKA extension activated. The test subjects were only allowed to watch videos recommended to them from the front end of the extension. Again, the count of recommended videos and derived coefficient values were graphed to get Table II.

The model searched randomly and gave the test subjects a set of relevant videos from each label, as expected. The video links were recorded and, after conducting the experiment for a period, a qualitative survey of the overall experience of each test subject was taken. Several test subjects that had once insisted on watching videos closer to their opinion, expressed genuine interest in content from the "other side". Table III shows the absolute and percentage-based polarity change between the recommended videos with and without the SSKA extension.

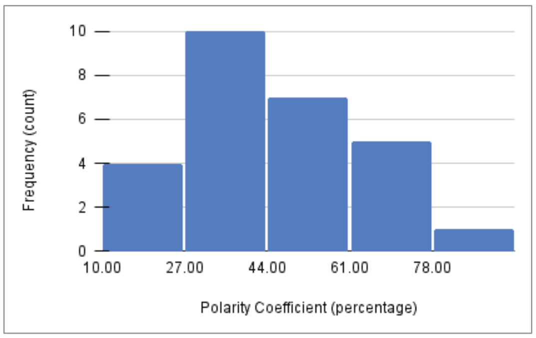

Fig. 4. Frequency of videos for different polarity ranges (without extension)

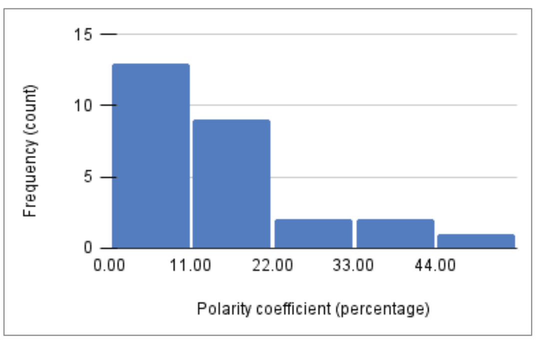

Fig. 5. Frequency of videos for different polarity ranges (with extension)

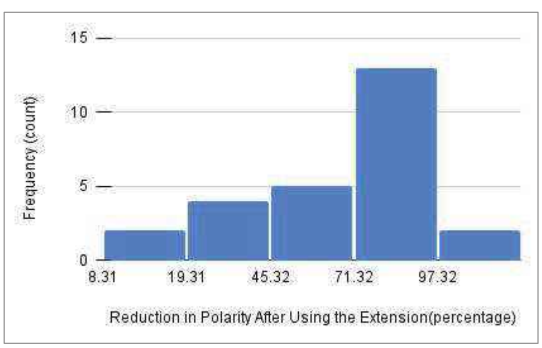

Fig. 6. No. of users experiencing percentage reduction in polarity

To drive the point, a clear visual difference between Figs 4 and 5 can be observed. Fig 4 represents the frequency of videos shown in different polarity ranges without the extension; Fig 5 shows the frequency in polarity ranges with the extension. Both histograms map the polarity range to the frequency of exposure based on the recommendation system used.

As can be seen, the frequency of exposure for more extensive polarity ranges decreased drastically, whereas the frequency of lower polarity ranges increased just as drastically. This difference has been mapped in Fig 6, where the different percentage reductions of polarity and how many users experienced it are shown.

V. Conclusion

The SSKA architecture was proposed; with the intent of extensively bias-based content exposure as seen on YouTube. The model learns and presents YouTube content based on varying opinions, effectively decreasing the observable polarity in the view of users. It searches for relevant videos to a topic and divides videos based on their different views. It then exposes users to this new set of recommendations. The results have clearly shown a reduction in the polarity of views of the subjects, thus showing great potential in helping people appropriately acknowledge views they may be biased against. For example, one test subject exclaimed how "invested," they had become "in the humanness behind the opinion opposing my own."

In contrast, another test subject admitted that "great industrial potential exists in the application." The qualitative evidence of the frequency shift, as seen in the interviews taken of the test subjects, gives hope of greater harmony amongst people, caused simply by the ability to understand where the many perspectives on a subject matter are coming from. Walking in another's shoes is an old and proven way of increasing understanding and wisdom among the average human. However, there is no doubt that much more can be accomplished with the SSKA model. The model needs to be adaptable to other social media outlets and other types of content, and it needs extensive work in its feature set.

The technologically meaningful and psychologically healthy growth as a potential of this model and its implementation in the proposed problem set gives great hope. This model should be a source for much development in the field of polarity mitigation, effectively decreasing individual bias and misinformation across the world. The technology used can find a place and grow in all types of systems (even outside of social media recommendation systems). Whether that be increasing the label columns and textual embeddings (to support the ability of representing more views and opinions) or implementing the model on different technology systems (such as native apps, cloud-based systems, or even fully fledged artificial intelligence networks).

References

[1] J. Liu, E. Pedersen, and P. Dolan, 'Personalized News Recommendation Based on Click Behavior', in 2010 International Conference on Intelligent User Interfaces, 2010.

[2] P. Covington, E. Sargin, and J. Adams, "Deep Neural Networks for YouTube recommendations," Deep Neural Networks for YouTube Recommendations. Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf.

[3] F. P. Santos, Y. Lelkes, and S. A. Levin, “Link recommendation algorithms and dynamics of polarization in online social networks,” Proceedings of the National Academy of Sciences, vol. 118, no. 50. Dec. 06, 2021. doi: 10.1073/pnas.2102141118.

[4] A. Matakos, E. Terzi, and P. Tsaparas, “Measuring and moderating opinion polarization in social networks,” Data Mining and Knowledge Discovery, vol. 31, no. 5. Springer Science and Business Media LLC, pp. 1480–1505, Jul. 15, 2017. doi: 10.1007/s10618-017-0527-9.

[5] J. Cho, S. Ahmed, M. Hilbert, B. Liu, and J. Luu, “Do Search Algorithms Endanger Democracy? An Experimental Investigation of Algorithm Effects on Political Polarization,” Journal of Broadcasting & Electronic Media, vol. 64, no. 2. Informa UK Limited, pp. 150–172, Apr. 27, 2020. doi: 10.1080/08838151.2020.1757365.

[6] G. Vashisht and Y. N. Sinha, “Sentimental Study of CAA by location-based tweets,” International Journal of Information Technology, vol. 13, no. 4, pp. 1555–1567, March. 22, 2021 doi: 10.1007/s41870-020-00604-8.

[7] G. Tripathi and M. A. Ahad, “Impact of excessive use of internet on cognitive development of youngsters,” International Journal of Information Technology, vol. 9, no. 3, pp. 281–286, July. 22, 2017. doi: 10.1007/s41870-017-0028-5.

[8] NLP - Overview. Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/nlp/overview_history.html.

[9] G. W. Blackwood, “Lattice rescoring methods for statistical machine translation: Semantic scholar,” Semantic Scholar, 01-Jan-1970. Available:https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.596707

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv.org, 07-Sep-2013. Available: https://arxiv.org/abs/1301.3781.

[11] T. K. Tran and T. T. Phan, “Mining opinion targets and opinion words from online reviews,” International Journal of Information Technology, vol. 9, no. 3, pp. 239–249, Jul. 31, 2017. doi: 10.1007/s41870-017-0032-9.

[12] A. Radford and K. Narasimhan, “Improving language understanding by generative pre-training: Semantic scholar,” undefined, 01-Jan-1970. Available: https://gwern.net/doc/www/s3-us-west-2.amazonaws.com/d73fdc5ffa8627bce44dcda2fc012da638ffb158.pdf .

[13] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Proceedings of the 2019 Conference of the North, 2019. doi: 10.18653/v1/n19-1423.

[14] C. Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” arXiv, 2019. doi: 10.48550/ARXIV.1910.10683.

[15] C. Ding and X. He, “K-Means Clustering with Distributed Dimensions,” Twenty-first international conference on Machine learning - ICML '04, 2004. doi: 10.1145/1015330.1015408.

[16] S. K. Sahu, P. Kumar, and A. P. Singh, “Modified K-NN algorithm for classification problems with improved accuracy,” International Journal of Information Technology, vol. 10, no. 1, pp. 65–70, 2017, Nov. 11, 2017. doi: 10.1007/s41870-017-0058-z.