Comparing Predictions of YouTube Video Like to Dislike Ratios Using Sentiment Analysis Tools

Giacomo Marsanich
STUDENT NUMBER: 2038231

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE IN COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

Dr. Giovanni Cassani
Dr. Mirella De Sisto

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
September 2022

Preface

This thesis is original, unpublished, independent work written in partial fulfillment of the requirements for the degree of Bachelor of Science in Cognitive Science & Artificial Intelligence at Tilburg University.

By writing this thesis, I was able to bring together the knowledge I acquired during my studies and apply it to a topic I was interested in. I hope you are inspired to do the same. I want to thank my friends and family for their continued support. I also want to thank Dr. Giovanni Cassani, my supervisor, for his help and guidance in writing this thesis.

I hope you enjoy it and/or find it interesting.

Comparing Predictions of YouTube Video Like to Dislike Ratios Using Sentiment Analysis Tools

Giacomo Marsanich

In this thesis, I attempt to predict the like to dislike ratio of a given YouTube video by using the ratio of positive, neutral and negative comments related to that video. Previous research attempted to generalize the opinion of the viewers of a given video using sentiment analysis. Based on this and in addition to my main goal, I also attempt to generate a new engagement metric using the polarities of the video's comments. I collected over 300,000 comments across over 500 videos from the top 50 most searched terms. I concluded that the approach I used to predict the like to dislike ratio did not provide satisfactory results. Both a Pearson correlation test and a Spearman correlation test found no statistically significant correlation between the ratio of positive, neutral and negative comments to a video's like to dislike ratio (M = .0025, SD = .0327, p = .62), (M = -.12, SD = 0.076, p = .14).

1. Introduction

On the 10th of November 2021, YouTube removed the public dislike counter from all videos. Allegedly, this was done to avoid targeted dislike attacks towards small creators and reduce dislike bombing - where users see a heavily disliked video and dislike it purely based on the negative ratio. This resulted in a worsened user experience where judging the trustworthiness, reliability, relevancy and overall quality of a video has become quite difficult without watching at least part of the video, or scrolling to read the comments[1](Marsanich).

To judge the contents of a YouTube video, users can turn to the comments section (Thelwall, 2018), where they can read what other users think of the video in question. Thus, by using sentiment analysis on the contents of the comments section, it could be possible to determine what the users generally think of the video in question (Benkhelifa and Laallam, 2018). This would likely improve the user experience, and the resulting data could be used by brands to improve their products (Irawaty, Andreswari, and Pramesti, 2020).

Since both the comments section and the like to dislike ratio inform users about the quality of a video, it could be possible to predict the like to dislike ratio based on data collected through sentiment analysis. It could also be possible to generate a new

1 There is currently no empirical study to support this claim. However, there are opinion pieces from reputable sources that indicate the validity of this claim (see:
https://www.forbes.com/sites/petersuciu/2021/11/24/youtube-removed-dislikes-button–it-could-impact-how-to-and-crafts-videos/,
https://www.theverge.com/2021/11/17/22787080/youtube-dislikes-criticism-cofounder-jawed-karim-first-video-description-zoo,
https://www.bbc.com/news/newsbeat-59264070)

engagement metric based purely off the polarity of the comments. This information, if accurate, could improve the user experience on YouTube. Conducting sentiment analysis on social media might also better inform us about how language is used on social media, and how it has evolved over time - particularly how people express positive and negative feelings online (Ben-Ze'ev, 2004).

In this thesis, I aim to predict the like to dislike ratio by using three sentiment analyzers to classify the polarity of each comment under a given video, and compare their performance. I will use the ratio of positive, neutral and negative comments to perform this prediction. As an additional task, I will use the polarities generated by the three models to assign a score to each collected video, in order to create an additional engagement metric.

2. Related Work

The concept of sentiment analysis is rooted in studies of public opinion conducted during the 20th century (Mäntylä, Graziotin, and Kuutila, 2016). However, large scale computer based sentiment analysis was only popularized with the introduction of the Internet and its many forums and messaging boards (Mäntylä, Graziotin, and Kuutila, 2016).

The first computer based sentiment analysis paper was published in 2002. It was based on product reviews which it would classify with either "thumbs up" or "thumbs down" using an unsupervised Pointwise Mutual Information - Information Retrieval (PMI-IR) algorithm (Turney, 2002). This algorithm compares the similarity of a given sentence to a positive reference. In this case, the word "excellent" - and a negative reference, in this case, the word "poor". If the input is more similar to the positive reference, it receives a thumbs up. If the input is more similar to the negative reference, it receives a thumbs down (Turney, 2002).

While the algorithms have become more complex (Brown et al., 2020; Wolf et al., 2019), the applications of sentiment analysis have remained similar. Sentiment analysis is still mostly used to classify product reviews (Yang et al., 2020; Mukherjee and Bhattacharyya, 2012). However, as social media became more and more prevalent in society, the value of analyzing how the public feels about certain products, topics or people has made itself clear (Joyce and Deng, 2017; Martin-Domingo, Martín, and Mandsberg, 2019; Neri et al., 2012). The prediction of closing market prices based on sentiment analysis of relevant social media channels is also an area of interest for researchers (Jin, Yang, and Liu, 2020) and corporations alike. It therefore seems logical that we could apply sentiment analysis on on social media such as YouTube to learn what viewers think of a given video. Studies by Cunha, Costa, and Pacheco (2019) and Benkhelifa and Laallam (2018) show just this. Cunha, Costa, and Pacheco (2019) use a deep neural network to classify YouTube comments to generate a comment based rating to supplement the like and dislike counters. Benkhelifa and Laallam (2018) aim to classify comments under cooking videos as positive, neutral or negative. These examples show how sentiment analysis can be applied to YouTube comments. It must be said that most works on the application of sentiment analysis on YouTube comments are written by students, highlighting what could be a generational gap between researchers.

There are challenges related to using sentiment analysis on social media. According to Pozzi et al. (2017), grammar rules are not always followed on social media, which means that algorithms trained on grammatically correct text may struggle to analyze YouTube comments. Furthermore, slang, abbreviations, emojis and other variations in language may also impact how an algorithm perceives a given text. Pozzi et al. (2017)

also state that as language evolves with technology, the tools used to analyze it must also evolve. In the context of sentiment analysis this could mean that rulesets need to be kept up to date, or that models should implement some way of updating themselves to the language used on social media. Hussein (2018) believe that there are challenges in how sentiment analyzers parse certain aspects of natural languages, such as negations and domain specific language. It is also possible to identify patterns of subtopic difference (i.e. words that are specific to a particular subtopic may be difficult to accurately classify) as well as gender (i.e. whether commenters of a particular gender express more positive or more negative sentiments) and sentiment (i.e. seeking comments that are expressly more polarizing) which may lead to missing comments that are more moderate (Thelwall, 2018) and thus skewing the distribution of the data towards a more polarized direction.

3. Experimental Setup

The data used in this thesis could not come from an existing dataset, since datasets with YouTube comments such as SenTube (Uryupina et al., 2014) usually do not contain information about the video from which they were collected. Such datasets are also mainly concerned with training and testing machine learning algorithms rather than to predict data using pre trained models. Instead, the data had to be collected from the source in order to preserve the information needed for the analysis.

The data consists of two parts: the comments, and the like to dislike ratio for a given video. A limitation of this data is the like to dislike ratio, specifically the dislike count. This is due to the fact that dislikes are an estimation based on "A combination of archived data from before the official YouTube dislike API shut down, and extrapolated extension user behavior" (https://github.com/Anarios/return-youtube dislike). This is not ideal, but there is no other way of collecting dislikes other than asking the creator of the video, which is highly impractical.

Since the crux of this thesis is the prediction of dislikes based on the sentiment of comments, a baseline needed to be generated. The simplest way to do this is to generate the mean like to dislike ratio for all videos collected and comparing the prediction from each model to that baseline. The prediction was generated by taking the ratio of the sum of positive and negative comments, divided by the negative comments.

3.1 Design & Procedure

In this experiment, the independent and dependent variables were the polarity of the comments and the like to dislike ratio respectively. The experiment is rather simple: once the comments had been analyzed and labeled (negative, neutral, positive), a ratio of labels was generated (e.g. 25% negative, 30% neutral, 45% positive) and based on this ratio, the like to dislike ratio was predicted. This prediction was then compared to the actual like to dislike ratio for that given video as well as the average ratio for all collected videos. This comparison was done using statistical tests such as Pearson and Spearman correlations and mean absolute error (MAE).

The sentiment analysis stage was conducted as follows: each file in the directory was loaded and its contents were read. The contents were then passed through the googletrans Python library where they would be translated into English from their original language. The translated comments would then be passed to a multiprocessing pool to be analyzed by Vader and TextBlob concurrently. The outputs were saved to separate lists that were then saved to an additional list. Since BERT is multilingual, the

comments were passed without being translated. The output of BERT's classification was saved in a separate list. After that, dictionaries were built containing the original comment as well as the polarities for that comment as given by the three models. Each dictionary was then placed into a Pandas dataframe as a row. Next, the labels (positive, neutral and negative) were generated based on the polarity scores. They were then placed into the dataframe as columns. Finally, the dataframe was saved as a CSV file to avoid rerunning the analysis. To put it more simply, 100 JSON files were analyzed per batch, and the results of the analysis were saved in a CSV file containing the comment, the score given by each model, as well as a label for each score.

3.2 Data

There are multiple steps in the process of collecting data for this experiment. The first step was to collect YouTube search terms, which would then be used to collect links to YouTube videos. More specifically, the terms were the top 50 most searched terms globally on YouTube. The search terms were collected from https://ahrefs.com/blog/top-youtube-searches/. Each term was then passed as an argument to a function that would call an API , which would run a search and return the results - more specifically, a list of lists containing links. Once the list of links was flattened and saved, the likes and dislikes of each video needed to be collected. This was done through the ReturnYoutubeDislikes API. The results of the API call were saved in a Python dictionary containing the link and its corresponding likes and dislikes. The dictionaries were saved as a CSV file for later use. The file contained 911 links. Table 1 shows a sample of a video link with its corresponding like and dislike counter.

Table 1
Sample of like/dislike dataset with corresponding link

URL	likes	dislikes
https://www.youtube.com/watch?v=feLcMaiClOw	312,761	19,093
https://www.youtube.com/watch?v=tASwv-tkMlc	35,026	165
https://www.youtube.com/watch?v=3U2dNKBM28o	126,427	6,010
https://www.youtube.com/watch?v=1cPDfXU95Xw	23,783	18
https://www.youtube.com/watch?v=pptIU_ZLlOo	4,600	7
https://www.youtube.com/watch?v=ClQ-ymoXJZc	1,079,676	8,874
https://www.youtube.com/watch?v=8EJ3zbKTWQ8	11,371,069	1,420,624
https://www.youtube.com/watch?v=t99KH0TR-J4	1,435,738	33,092
https://www.youtube.com/watch?v=DovdIspaqmw	146,007	5,324

Lastly, using the links, video comments were collected using the YouTube API. Since many popular videos can have upwards of tens of thousands of comments, the analysis was going to take multiple consecutive days to run. Due to time constraints as well as API rate constraints, this was deemed unfeasible. Therefore only a sample of comments was collected from each video - around 600 per video. Some videos had their comments disabled, which meant that they were of no use, and some videos had fewer than 600 comments, so they had to be removed. Once that was done, we were left with 569 JSON files with 600 comments each. A total of 339,364 comments were collected and analyzed.

No preprocessing was done on the comments except removing newline characters and tab characters. Emojis were preserved, as well as other non-standard character com-

binations. Since the models in use provided their own simplified APIs and pipelines, I believed it would be best to let the models use their own preprocessing and tokenization functionalities.

The data was split into 6 batches of 100 files each in order to speed up the analysis process.

3.3 Method, Models & Libraries

The experiment was conducted in an IPython notebook (Pérez and Granger, 2007) running a Python 3.10.4 kernel (Van Rossum and Drake, 2009). Data manipulation and visualization was done through the Pandas (McKinney et al., 2010) Python library. Further statistical analyses were conducted using SciPy (Virtanen et al., 2020) and scikit-learn (Pedregosa et al., 2011). Additional data visualization for this thesis was done through Matplotlib (Hunter, 2007) and Seaborn (Waskom et al., 2017). The requests library (Chandra and Varanasi, 2015) was used to make HTTP requests to various APIs in order to fetch data.

The data was analysed using three different sentiment analyzers: VADER (Hutto and Gilbert, 2014), TextBlob (Loria, 2018) and BERT (Wolf et al., 2020). Each model is radically different from each other. Vader is a lexicon-based-purpose-built sentiment analysis tool. According to the developers, "VADER incorporates a "gold-standard" sentiment lexicon that is especially attuned to microblog-like contexts" (Hutto and Gilbert, 2014). 10 separate human experts were used to classify over 9000 tokens in order to build up the lexicon. This, combined with the fact that it was intended to be used on social media, makes VADER an ideal candidate for this analysis.

While VADER is highly specialized in social media, TextBlob is more general-purpose. It is a Python library for NLP tasks such as tokenization, part-of-speech tagging, noun-phrase extraction, and sentiment analysis (Loria, 2018). It uses a Naive-Bayes analyzer trained on movie reviews to perform sentiment analysis tasks (Loria, 2018). Due to its ease of use, it is not unreasonable to expect that many people who wish to conduct NLP tasks might turn to this model. Therefore, it makes a good candidate for this thesis.

Lastly, BERT is the most recent of the three models, being introduced in 2018. BERT is based on transformers, which means that it weighs the significance of each token in any given input text instead of simply reading the text from beginning to end - it reads the text from end to beginning as well - hence the bidirectional part of the name (Wolf et al., 2020). Due to its architecture, BERT is able to utilize the context in which a word is used to determine its intended meaning (Wolf et al., 2020). For instance, take the following texts: "I water the plants" and "I drink water". Other models might assign the same meaning to the word water regardless of context, whether it is used as a verb (e.g. "to water") or as a noun ("the water"). BERT is more likely to assign the correct meaning for the corresponding context, i.e. verb for the first text and noun for the second. This ability makes it another ideal candidate for this thesis.

Each model behaves slightly differently in how they report sentiment scores. Vader's developers suggest using the "compound score" as the polarity score for most sentiment analysis tasks. They state that "the compound score is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). This is the most useful metric if you want a single unidimensional measure of sentiment for a given sentence" (Hutto and Gilbert, 2014). The other two scores *"pos"* and *"neu"*

are used in contexts where the context or presentation of the input is more important than the input itself. They were not needed for this experiment and were thus ignored.

BERT does not return negative values by default. Instead, it returns a positive polarity score along with a label consisting of one to five stars. For example, the sentence "I love going to the beach" might get a polarity of 0.8 and four stars, meaning it is very positive; the sentence "I hate bugs" might get a polarity of 0.3 and two stars, meaning it is very negative. To make BERT behave similarly to the other two models, this property had to be altered in the software. BERT's behavior was altered like so: if the number of stars is between three and five (inclusive), the polarity is multiplied by one, so as to retain the positivity. If instead the number of stars is one or two, the polarity is multiplied by negative one. This ensures that all models have the same range of polarity. TextBlob reports subjectivity in addition to polarity. While this might be a useful metric for certain tasks (for instance classifying reviews as helpful or unhelpful based on subjectivity and polarity), it was not in this case and it was therefore discarded.

Ranges had to be set for each model in order to generate labels. Each model had to use a range specific to it. The Vader developers suggest using the following range: compound scores greater than or equal to 0.05 should be labeled as positive, compound scores smaller than or equal to -0.05 should be labeled as negative, and scores between -0.05 and 0.05 should be labeled as neutral. TextBlob and BERT do not have suggested ranges. Initially, I had set both to use the same range: values greater than or equal to 0.33 should be set to positive, values between 0 and 0.33 should be set as neutral, and values smaller than 0 should be set to negative. While this seemed to work for BERT, it produced far too many neutral values for TextBlob and thus it had to be changed for TextBlob. Though TextBlob's developers do not indicate a specific range, many online guides suggest setting values greater than 0 as positive, values smaller than 0 as negative, and values equal to 0 as neutral.

3.4 Accounting for different languages

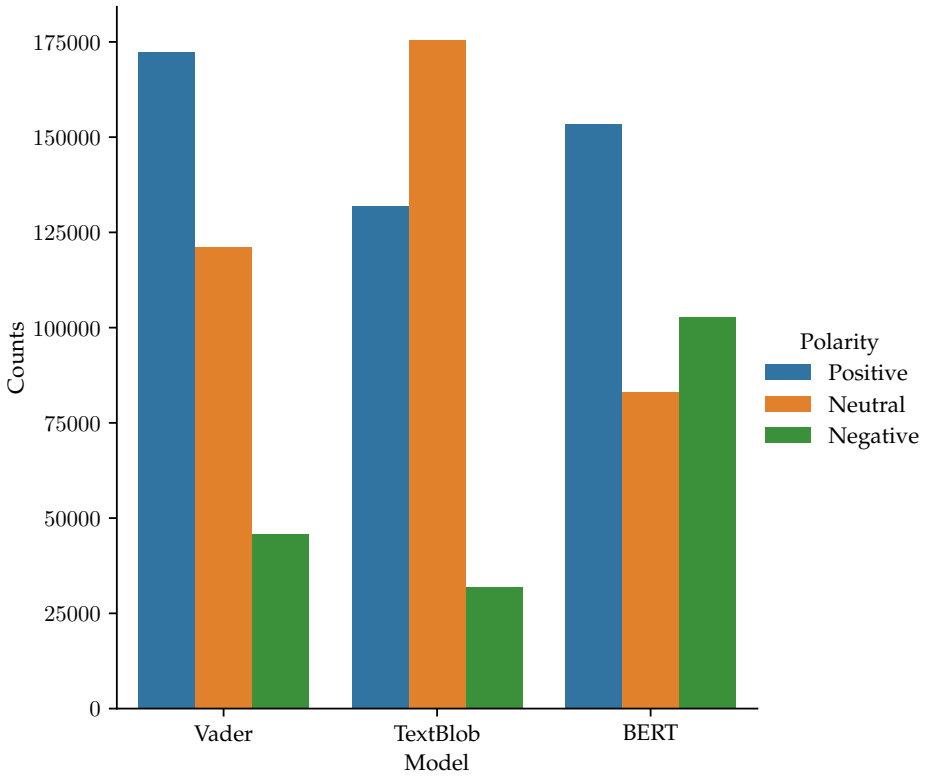
These models all perform well on English text; however, not all text on social media is in English, especially on YouTube. BERT is a multilingual model, therefore it has no issues with non-English text. We are therefore presented with a choice: find and instantiate multingual models for each individual comment, making the analysis much more complicated and time consuming than it should be, translating comments using an accurate translator and analyzing the translation, or simply discarding non-English comments. The latter can immediately be discarded as far too much data will be lost. As for the remaining options, the translation is clearly the simpler choice, and that is what was done. Each comment was passed through Google Translate using the googletrans Python library before being analyzed with VADER and TextBlob. Google Translate was at one point the most accurate machine translation service (Aiken and Ghosh, 2009). However, many years have passed and services such as DeepL Translator (DeepL) now claim to be the most accurate machine translation services on the market. The reason Google Translate was chosen was purely practical: it is an unlimited rate API. This means that it can be used as many times and as often as required, without having to wait for the quota to reset in-between uses. While DeepL may have provided more accurate translations and therefore more accurate sentiment polarities, it simply could not be used.

4. Results

4.1 Labels

For each comment, a label was generated based on the polarity score returned by each classifier. The labels were negative, neutral and positive. Each label was counted for each model and plotted in a bar graph for easier visualization (Figure 1).

Figure 1
Total counts for each polarity label for each model



TextBlob seems to classify most comments as neutral, while VADER classifies most comments as positive. BERT's predictions are more balanced. Both VADER and TextBlob classify very few comments as negative.

4.2 Polarities

The sentiment analysis step of the experiment generated polarity scores for each comment and for each model. The polarity scores are values between -1 and 1, where -1 is extremely negative and 1 is extremely positive. To gauge overall trends, all comment files were merged into a single file. The mean polarities for Vader, TextBlob and BERT were 0.26(SD = *0.45*), 0.15(SD = *0.32*) and 0.20(SD = *0.46*) respectively. Other relevant statistics can be found in table 2.

Table 2
Descriptive statistics for polarity scores per model

	Vader	TextBlob	BERT
mean	0.26	0.15	0.20
std	0.45	0.32	0.46
min	-1.00	-1.00	-0.99
25%	0.00	0.00	-0.29
50%	0.09	0.00	0.31
75%	0.64	0.35	0.52
max	1.00	1.00	0.99

It may be insightful to look at the distribution of polarities in addition to the labels, to get a better idea of how each model actually classifies the comments. Figure 2 is a Kernel Density Estimation (KDE) plot showing the distribution of polarities per model.

Figure 2
KDE plot of polarity distribution per model

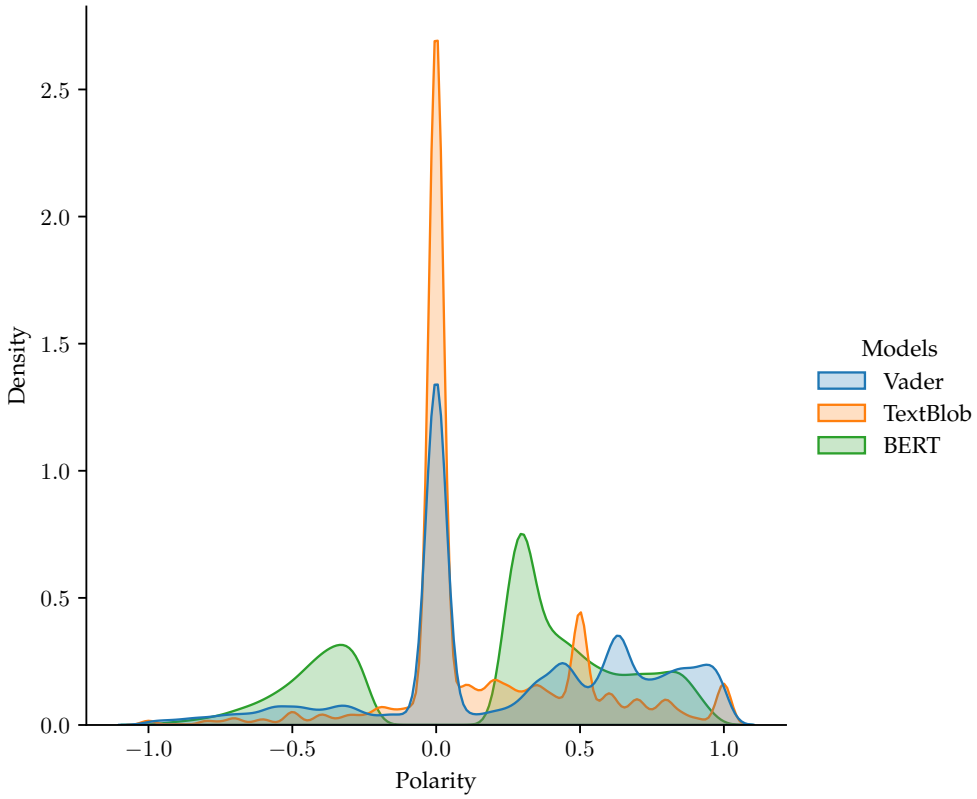
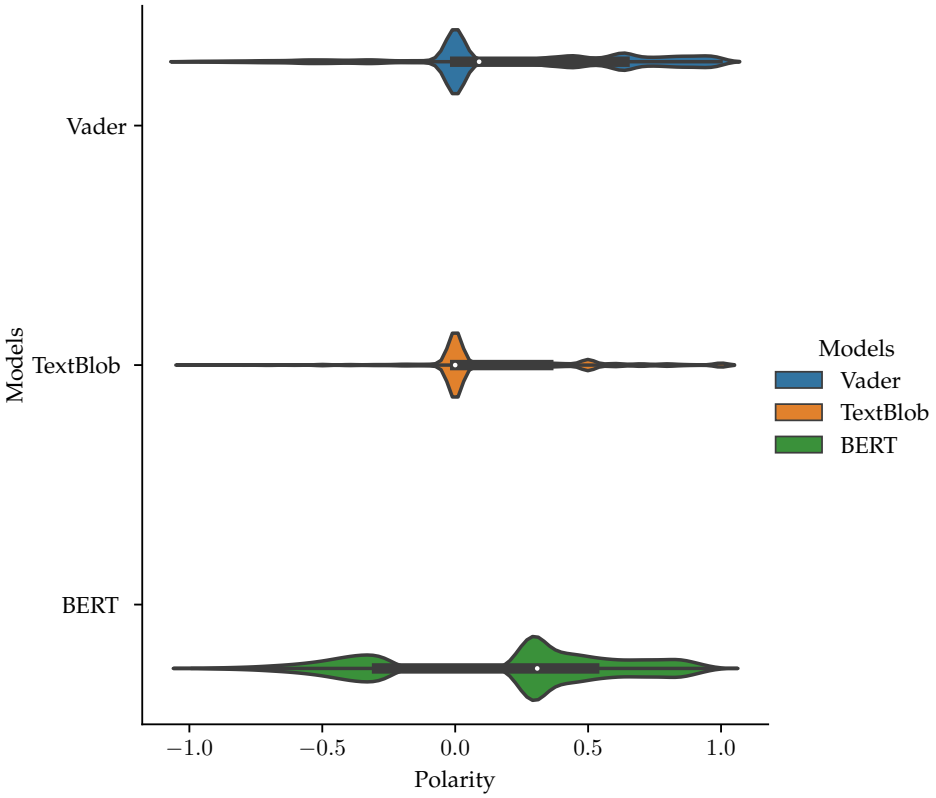


Figure 2 shows that BERT (shown in green) never assigns a polarity of 0. This provides an explanation for BERT's relatively low number of neutral comments. The

range set for BERT to classify comments as neutral was set to be between 0 (inclusive) and 0.33 (exclusive). As the the trough begins in the negative range, this is likely to be by design and not an artefact of the range that was set. As for why it was designed this way, one can only speculate. Perhaps the authors believed that the text on which BERT was trained on and intended to be used on would not be completely unpolarized. Figure 2 highlights spikes in Vader and TextBlob's distributions. In both cases these spikes are around 0, meaning that these models classified a large part of the inputs as neutral or slightly positive. Since KDE plots are essentially smoothened histograms and are thus prone to errors, it might be worth looking at a different kind of plot to get an even better idea of how the polarities are distributed.

Figure 3
Violin plot of polarity distribution per model



The violin plot in figure 3 confirms what we see in figure 2: Vader and TextBlob have a spike near zero, and BERT has a noticeable lack of polarities around zero. We also see that Vader's spike is wider than TextBlob's, indicating that it may be less centered on zero.

Figure 4 shows a boxen plot - a box plot that shows additional quantiles in the tail and head (Waskom et al., 2017). We notice that Vader has a positively skewed distribution, with most of its assigned polarities being larger than zero. TextBlob has a very long tail but few negative polarities, in line with figure 2. BERT's distribution

seems more even, with more polarities between -0.5 and 0.5 and fewer extreme positives and extreme negatives without over representing neutral sentiments.

Figure 4
Boxen plot of polarity distributions per model

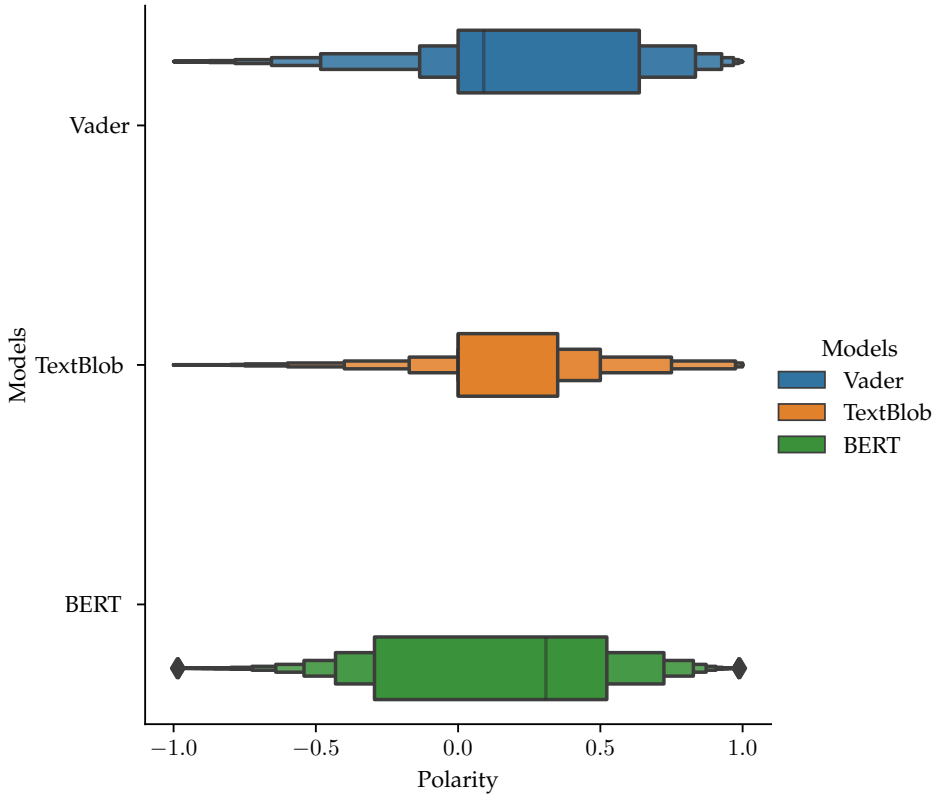


Table 3
Correlation Matrix of assigned polarity scores

	Vader polarity	TextBlob polarity	BERT polarity
Vader polarity	1.00	.52	.45
TextBlob polarity	.52	1.00	.45
BERT polarity	.45	.45	1.00

Table 3 shows a correlation matrix of the assigned polarity scores for each model. All models seem to correlate with each other, though Vader and TextBlob correlate more with each other $r(339,363) = .52$ than with BERT $r(339,363) = .45$. This makes sense considering both models' distributions are spiked in the same region.

4.3 Prediction of video specific ratios

The prediction of the like to dislike ratio was decidedly unsuccessful. All models performed poorly with Pearson's r coefficients very close to 0. Table 4 shows each r coefficient and its corresponding p value per model. BERT performed slightly less poorly. It was the only model with a positive Pearson's r coefficient. Vader had the lowest Pearson's r coefficient - although the p value suggests almost no statistical significance, being very close to 1. We can affirm that Vader was the worst performer out of the three models for Pearson correlation specifically. BERT performed the best, with its r value furthest from 0 *(r(339,363) = .037, p = .38)* and the lowest p value of the three models. Despite this, BERT's performance is still not statistically significant.

The Spearman correlation test showed different results. Vader and TextBlob both have low negative correlations to the true ratio. Both models have p values below the α value, meaning that their results are statistically significant. In contrast, BERT's correlation is much smaller than both Vader's and TextBlob's, and its p value is much larger. This tells us that BERT did not perform as well as the other models.

Table 5 shows the true ratios against the predicted ratios for each model. BERT's ratios *(M = 2.71)* are much smaller than both Vader*(M = 8.85)* and TextBlob *(M = 13.58)* . The explanation for this could be due to the distribution of polarities across BERT's classification and their impact on the computation of the ratio. The ratio of comments was a simple sum of the number positive and neutral comments, divided by the number of negative comments. Figure 2 shows that BERT almost never predicted a score of 0, thereby making its neutral predictions much less common than the others, which in turn made the numerator in the fraction smaller while keeping the same denominator. Thus, the ratio computed must necessarily be closer to 1 than the ratios computed by Vader and TextBlob.

Table 4
Correlations of predicted like/dislike ratio and true like/dislike ratio per model

Model	Pearson's r	p value	Spearman's r	p value
Vader	-.0018	.97	-.15	<.001
TextBlob	-.028	.51	-.17	<.001
BERT	.037	.38	-.03	.42

Table 6 shows the mean absolute errors for each model. The overall performance was unsatisfactory as no model came close. BERT has the highest MAE, with Vader and TextBlob following suit. This is in line with the results of the correlation that showed BERT as being less able to predict the true ratio than both Vader and TextBlob.

4.4 Prediction of mean ratios across models

To further corroborate the analysis, the mean predicted ratio per model was compared against a baseline consisting of the mean of the true ratios. Since correlations on single values are unreliable at best and outright misleading at worst, using the raw differences between the true mean ratio and the predicted mean ratios could be a good way of evaluating performance. We see once again that BERT is furthest from the true mean ratio and TextBlob is the closest. It is important to note that no model performed particularly well.

Table 5
Sample of real like/dislike ratio vs. predicted like/dislike ratio per model

ID	like/dislike ratio	Vader ratio	TextBlob ratio	BERT ratio
feLcMaiClOw	16.38	6.69	8.84	1.38
tASwv-tkMlc	212.28	12.64	7.82	2.57
3U2dNKBM28o	21.04	26.28	23.00	3.84
ClQ-ymoXJZc	121.67	3.92	4.41	1.29
8EJ3zbKTWQ8	8.00	8.15	10.44	3.13
t99KH0TR-J4	43.39	6.23	11.77	3.65
DovdIspaqmw	27.42	5.90	21.22	0.64
DQQRjFzB8gY	17.43	10.30	17.16	2.94
8DytqFTwNSc	42.55	11.00	16.14	2.43
2cyzCReoNgU	96.27	9.68	17.12	2.93

Table 6
Mean absolute error per model

Model	MAE
Vader	90.04
TextBlob	87.83
BERT	94.27

Table 7
Raw difference of true mean ratio and model predicted ratio

Model	Raw mean difference
Vader	114.70
TextBlob	109.97
BERT	120.84

4.5 Scoring

Based on the labels, each comment was assigned a score: -1 for negative, 0 for neutral and 1 for positive. The scores were then summed to generate a total score for the video. Negative total scores indicate negative opinions about a given video while positive total scores indicate neutral and positive opinions. The scoring is intended to be used as an additional engagement metric to supplement the like to dislike ratio. Refer to Table 9 for a sample of the scoring. Vader has the highest scores on average, while BERT has the lowest. This is not surprising considering the fact that Vader classifies most comments as positive, whereas TextBlob classifies most as neutral and BERT's classifications are more balanced. The scores seem to be very strongly correlated despite their raw size difference (see table 8). Much like in table 3, Vader and TextBlob correlate more with each other than with BERT.

Table 8
Pearson correlation matrix of generated scores

	Vader score	TextBlob score	BERT score
Vader score	1.00	.85	.76
TextBlob score	.85	1.00	.78
BERT score	.76	.78	1.00

Table 9
Sample of videos with corresponding generated scores

ID	Vader total	TextBlob total	BERT total
feLcMaiClOw	294.0	199.0	-101.0
tASwv-tkMlc	365.0	177.0	122.0
3U2dNKBM28o	333.0	253.0	152.0
ClQ-ymoXJZc	241.0	229.0	233.0
8EJ3zbKTWQ8	62.0	61.0	-298.0
t99KH0TR-J4	285.0	198.0	113.0
DovdIspaqmw	326.0	187.0	139.0
DQQRjFzB8gY	304.0	218.0	149.0
8DytqFTwNSc	310.0	286.0	258.0
2cyzCReoNgU	304.0	202.0	198.0

5. Discussion

5.1 Contextualizing the data

Before discussing the results, it is important to contextualize the data used. Given that YouTube is a social media platform, it is not unreasonable to assume that not every user decides to engage with every video they watch. Furthermore, if a user decides to engage with the video, it does not necessarily mean that they will always leave a positive comment and a like if they enjoyed the video or vice versa. Users are more likely to consume content passively. For example, the current second most viewed video on YouTube has over seven billion views. However, it only has 53 million total ratings (48 million likes and three million dislikes) and about four million total comments. If each comment were to be counted as an interaction or engagement with the video, it means that only 0.81% of users have actually engaged with the video. Considering that leaving a like or a dislike is less of an effort than typing a comment, it is therefore not unreasonable to expect that the opinion of every single viewer will not be reflected in the comments. This means that the comment section is a subsample of a subsample, where the sample is the total number of viewers and the first subsample is the number of ratings. It also does not seem irrational to think that users who do not feel strongly about the video they are watching will not engage with it at all, especially by commenting.

Furthermore, the dislike counter is an estimation by the ReturnYoutubeDislike extension and not an exact counter. This exacerbates the issues mentioned previously: not only are we attempting to estimate a concrete rating from a subsample of a subsample,

the rating we are trying to estimate is in itself an estimation. With this information in mind, we can proceed with the discussion of the results.

5.2 Interpreting the results

The Pearson correlation test shows that all models had very small correlations (positive and negative) to the true ratio. This does suggest that there may be a correlation between the ratio of positive and neutral comments over negative comments and the like to dislike ratio, however this correlation is so incredibly insignificant that it may as well be 0. This can be confirmed by looking at the correspnding p values (Table 4): we can see that they are all well above the α value of .05. This means that we must reject the hypothesis that there is a consistent link between the true ratio and the predicted ratio, and that we also fail to reject the null hypothesis.

In contrast, the Spearman correlation test suggests that results are statistically significant in two cases: for Vader and TextBlob, the p value is below the α value of .05 and their respective correlation coefficients *(r(339,363) = -.15, p < .001), r(339,363) = (-.17, p < .001)* are further from 0 than in the Pearson correlation, indicating a stronger (albeit still weak) correlation. BERT seems to have a very small negative correlation *(r(339,363) = -.03, p = .42)*. This is almost exactly the inverse of the r value given by the Pearson correlation *(r(339,363) = .037)*. Spearman's correlation suggests that BERT's correlation is more likely to be due to chance than both Vader and TextBlob, indicating that BERT's predictions were less in line than its counterparts'. These results tell us that we could fail to accept the null hypothesis in Vader and TextBlob's cases. However, taking into consideration the performance of all three models, we can say that there is indeed no statistically significant correlation between the predicted ratios and the true ratios *(M = -.12, SD = 0.076, p = .14)*. Furthermore, figure 2 shows that all models have abnormal distributions. Vader and TextBlob have spikes around 0, and BERT has a trough around the same region. Due to these factors, I am hesitant to make absolute statements about the performance of any of the models.

Table 6 and table 7 seem to confirm what the correlations suggested previously. Table 6 shows the mean absolute error of the ratios across all videos. Both correlations agreed (to some extent) that BERT's predictions are less correlated to the true ratios, and the errors seem to suggest that its predictions were consistently further from the true ratios than the other two models. Interestingly, the mean of TextBlob's predictions was closer to the true mean ratio *(MAE = 87.83)* than both Vader's *(MAE = 87.83)* and BERT's *(MAE = 87.83)*.

This is seen in the raw mean differences as well: TextBlob's mean prediction was closer to the true mean *(RMD = 109.97)* than Vader's *(RMD = 114.70)* and BERT's *(RMD = 120.84)*. This makes sense, considering that the Spearman correlation test showed that Vader and TextBlob did have a statistically significant correlation with the true ratios and that BERT did not.

While the scoring part of this experiment is not as important as the prediction of the ratio, it is still worth discussing. The fact that the models seem to be correlated in their classification and assignment of polarities (table 3) suggests that the scores generated should be quite similar to each other, and as mentioned in section 4.5, the scores generated by the models were indeed strongly correlated to each other despite being very different in absolute size. This suggests that the scores might indeed be a helpful engagement metric.

5.3 Model performance

There may be a number of reasons that caused the models to perform so poorly. One such reason could be related to translation inaccuracies caused by Google Translate, which in turn could affect the quality of the translated text and which would eventually affect how the models parsed and classified the text. This chain could lead to a poor understanding of the intended underlying meaning of the comment, which would then lead to an inaccurate classification and therefore a poor prediction of the like to dislike ratio. By using a different machine translation service, the results might be different.

Another reason may have to do with the granularity of the labels. For each model, a range was defined which determined whether the comment was labeled positive, neutral or negative. However, that range could have been made more granular: instead of having either positive or negative, those labels could be further divided into very positive and positive or very negative and negative. More granular labels would mean more detailed information for both the scoring system and the prediction. The scoring could be implemented by assigning 2, 1, 0, -1 and -2 points for very positive comments, positive comments, neutral comments, negative comments and very negative comments respectively. By giving more or fewer points to highly polarized comments we could better represent the two extremes. As for the prediction, more weight could be given to very positive and very negative comments, thereby better representing the subset of users that are more likely to leave a like or a dislike.

The models also did not classify comments uniformly, leading to uneven distributions. TextBlob's proportion of neutral comment classification (see Figure 1) is bizarre given the context of the data: it is possible that the Naive-Bayes classifier that it uses cannot discern any features in the input and it thus defaults to the prior, giving the comment a polarity score of 0 and therefore a neutral label. There is no way to confirm this however, as the documentation does not provide this information. Both TextBlob and Vader seem reluctant to score comments as negative, instead marking them as positive or neutral more often. At first glance a reasonable explanation could be that out of the relatively few people that choose to comment on YouTube videos, more people may choose to say what they like about the video rather than what they did not like. However, BERT's distribution is more reminiscent of the J-shaped distribution seen in online product reviews, where more polarized consumers (i.e. those with stronger positive or negative opinions) are more likely to leave reviews than those with moderate reviews (Hu, Pavlou, and Zhang, 2009). This could also be translated into the context of social media: if a viewer has no strong feelings about the video they are watching, they are less likely to engage with it, while more polarized viewers might choose to comment positively or negatively. Although the models' classifications were quite different from each other (while still being correlated), the scores they generated were much more strongly correlated to each other than their polarities (table 9).

This difference in distribution between the models may be due to a few factors. Firstly, Vader's lexicon may be out of date - the GitHub repository shows the last commit as four years ago, and given how quickly online lingo changes (Randall, 2002), Vader may have been left behind. A plausible explanation for TextBlob's strange behavior has been provided earlier in this section; however, it may also be possible that TextBlob may not be adapted to this particular context as it was trained on movie reviews (Loria, 2018). TextBlob may struggle with parsing YouTube or social media specific lingo and slang. As for BERT - considering it is the most recent and most complex model (with over 100 million parameters), it seems more likely that it would have a more accurate distribution of polarities across this dataset despite it being trained on product reviews

as well. With that in mind, it is important to reiterate that BERT's predictions were the furthest from the true values in this particular task and that any further research should investigate why that is. Perhaps it was implemented incorrectly in this experiment, or it was simply not designed for this sort of task. Either way, it seems strange that such an advanced and powerful model could perform this poorly. Perhaps it would have been a good idea to fine tune BERT on social media specific text. As for Vader and TextBlob, not much could be done to improve performance for this specific task other than cleaning the input data more thoroughly.

6. Conclusion

This thesis aimed to compare predictions by three different sentiment analyzers of the like to dislike ratio of YouTube videos based on the ratio of the polarities of the comments found under that video. It also aimed to generate an engagement metric based on the same data. The results of the prediction show that, for all models, there is a very small correlation between the true like to dislike ratio and the predicted ratio. In all cases, the Pearson correlation coefficient was extremely close to 0 and the high p values suggest that the results are not statistically significant.

However, the results seem to be somewhat contradictory: the Pearson correlation test suggests that there is no statistically significant correlation, while the Spearman correlation test suggests that there might indeed be one for two out of the three models.

The scoring of videos based on comment sentiments seems to be rather successful: all models are strongly correlated with each other suggesting that the scores might be representative of the public's opinion.

Further research could find a stronger correlation between these two data points. Such research should ideally tackle the limitations mentioned in the discussion section. It may also be worth repeating this analysis on algorithms and models trained on YouTube specific datasets in order to make use of more site specific terminology.

To sum up, I believe that while this experiment did not prove overly successful, it might provide background information for others who also wish to investigate the connection between the viewers' opinions and the metrics designed to capture them.

References

Aiken, Milam and Kaushik Ghosh. 2009. Automatic translation in multilingual business meetings. *Industrial Management Data Systems*, 109(7):916–925.

Ben-Ze'ev, Aaron. 2004. *Love online: Emotions on the Internet*. Cambridge University Press.

Benkhelifa, Randa and Fatima Zohra Laallam. 2018. *Opinion Extraction and Classification of Real-Time YouTube Cooking Recipes Comments*, volume 723. Springer International Publishing.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Chandra, Rakesh Vidya and Bala Subrahmanyam Varanasi. 2015. *Python requests essentials*. Packt Publishing Ltd.

Cunha, Alexandre Ashade Lassance, Melissa Carvalho Costa, and Marco Aurélio C. Pacheco. 2019. Sentiment analysis of youtube video comments using deep neural networks. In *Artificial Intelligence and Soft Computing*, page 561–570, Springer International Publishing.

DeepL. About deepl.

https://github.com/Anarios/return-youtube dislike. *Return YouTube Dislike*.

Hu, Nan, Paul A Pavlou, and Jie Jennifer Zhang. 2009. Why do online product reviews have a j-shaped distribution? overcoming biases in online word-of-mouth communication. *Communications of the ACM*, 52(10):144–147.

Hunter, J. D. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Hussein, Doaa Mohey El-Din Mohamed. 2018. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338.

Hutto, C. and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Irawaty, Irene, Rachmadita Andreswari, and Dita Pramesti. 2020. Development of youtube sentiment analysis application using k-nearest neighbors (nokia case study). In *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, pages 39–44.

Jin, Zhigang, Yang Yang, and Yuhong Liu. 2020. Stock closing price prediction based on sentiment analysis and lstm. *Neural Computing and Applications*, 32(13):9713–9729.

Joyce, Brandon and Jing Deng. 2017. Sentiment analysis of tweets for the 2016 us presidential election. In *2017 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pages 1–4.

Loria, Steven. 2018. textblob documentation. *Release 0.15*, 2.

Marsanich, Giacomo. Thesis proposal.

Martin-Domingo, Luis, Juan Carlos Martín, and Glen Mandsberg. 2019. Social media as a resource for sentiment analysis of airport service quality (asq). *Journal of Air Transport Management*, 78:106–115.

McKinney, Wes et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56, Austin, TX.

Mukherjee, Subhabrata and Pushpak Bhattacharyya. 2012. Feature specific sentiment analysis for product reviews. In *Computational Linguistics and Intelligent Text Processing*, pages 475–487, Springer Berlin Heidelberg, Berlin, Heidelberg.

Mäntylä, Mika, Daniel Graziotin, and Miikka Kuutila. 2016. The evolution of sentiment analysis - a review of research topics, venues, and top cited papers. *Computer Science Review*, 27.

Neri, Federico, Carlo Aliprandi, Federico Capeci, Montserrat Cuadros, and Tomas By. 2012. Sentiment analysis on social media. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 919–926.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pérez, Fernando and Brian E Granger. 2007. Ipython: a system for interactive scientific computing. *Computing in Science & Engineering*, 9(3).

Pozzi, F.A., E. Fersini, E. Messina, and B. Liu. 2017. *Challenges of Sentiment Analysis in Social Networks*. Elsevier.

Randall, Neil. 2002. Lingo online: A report on the language of the keyboard generation.
Thelwall, Mike. 2018. Social media analytics for youtube comments: potential and limitations. *International Journal of Social Research Methodology*, 21(3):303–316.
Turney, Peter D. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews.
Uryupina, Olga, Barbara Plank, Aliaksei Severyn, Agata Rotondi, and Alessandro Moschitti. 2014. SenTube: A corpus for sentiment analysis on YouTube social media. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4244–4249, European Language Resources Association (ELRA), Reykjavik, Iceland.
Van Rossum, Guido and Fred L. Drake. 2009. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.
Waskom, Michael, Olga Botvinnik, Drew O'Kane, Paul Hobson, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Cameron Pye, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Pete Bachant, Marcel Martin, Kyle Meyer, Alistair Miles, Yoav Ram, Tal Yarkoni, Mike Lee Williams, Constantine Evans, Clark Fitzgerald, Brian, Chris Fonnesbeck, Antony Lee, and Adel Qalieh. 2017. mwaskom/seaborn: v0.8.1 (september 2017).
Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. *Transformers: State-of-the-Art Natural Language Processing*. Association for Computational Linguistics.
Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.
Yang, Li, Ying Li, Jin Wang, and R. Simon Sherratt. 2020. Sentiment analysis for e-commerce product reviews in chinese based on sentiment lexicon and deep learning. *IEEE Access*, 8:23522–23530.