University of Essex
Department of Mathematical Sciences

MA981: DISSERTATION

How Streaming Platforms Can Use NLP Methods on Twitter and Reddit data to Derive Feedback from Audiences and Potential Audiences, with application to the Disney+ Original Series Ms. Marvel

Your Name
PG21149082

Supervisor: Dr. Spyridon Vrontos

August 26, 2022
Colchester

Contents

1 Abstract **5**

2 Introduction and Literature Review **6**

3 Methodology **11**

3.1 Sentiment Analysis 13

3.2 Topic Modelling 15

4 Results **17**

4.1 Positive Sentiment 18

4.2 Negative Sentiment 27

5 Discussion and Conclusion **37**

A Appendix A - Python Code for Data Collection **46**

B Appendix B - Python code for sentiment analysis classification **50**

C Appendix C - Python Code for sentiment bar charts **55**

D Appendix D - Python code for word cloud **58**

E Appendix E - Python code for LDA topic modelling **59**

List of Figures

3.1 Flowchart showing methodology 12
3.2 Diagrammatic illustration of LDA topic modelling generative process.[11] . . 15

4.1 Count of Positive Tweets compared to Negative Tweets about Ms. Marvel during the month of July 2022, see code in Appendix C 17
4.2 Count of Positive comments compared to Negative comments about Ms. Marvel on two popular posts from selected subreddits, see code in Appendix C 18
4.3 50 most frequent words associated with positive tweets, see code in Appendix D 19
4.4 50 most frequent words associated with positive Reddit comments, see code in Appendix D 19
4.5 Top 10 terms for each topic in the LDA topic model for positive tweets, see code in Appendix E 21
4.6 30 most salient terms, visualised using pyLDAvis, see code in Appendix E . . 22
4.7 Intertopic distance map for LDA topic model on positive tweets, visualised using pyLDAvis, see code in Appendix E 23
4.8 Top 10 terms for each topic in the LDA topic model for positive Reddit comments, see code in Appendix E 24
4.9 30 most salient terms for LDA Topic Model on Reddit comments labelled as having positive sentiment, visualised using pyLDAvis, see code in Appendix E 25
4.10 Intertopic distance map for LDA Topic Model on Reddit comments labelled as having positive sentiment, visualised using pyLDAvis, see code in Appendix E 26
4.11 Wordcloud showing 50 most frequent terms found in data frame of tweets with negative sentiment, see code in Appendix C 27

4.12 50 most frequent words associated with negative Reddit comments about Ms. Marvel, see code in Appendix C 30
4.13 Top 10 terms for each topic in the LDA topic model for negative tweets, see code in Appendix E 31
4.14 30 most salient terms for tweets with negative sentiment according to LDA topic model, visualised with pyLDAvis, see code in Appendix E 32
4.15 Intertopic distance map for LDA Topic Model on tweets labelled as having negative sentiment, visualised using pyLDAvis, see code in Appendix E . . . 33
4.16 Top 10 terms for topics in Reddit comments labelled as having negative sentiment, see code in Appendix E 34
4.17 30 most salient terms found in LDA topic model for Reddit comments labelled as negative, see code in Appendix E. 35
4.18 Intertopic distance map for LDA Topic Model on Reddit comments labelled as having negative sentiment, visualised using pyLDAvis, see code in Appendix E 36

CHAPTER 1

Abstract

With social media acting as the main public sphere for discussion of everything from science, to politics to the arts, users often take to platforms like Reddit and Twitter to express their opinion about a variety of media content, including that which is produced by companies like Netflix, Amazon Prime and Disney+[21]. Social media platforms, therefore, provide an abundance of data for content writers, producers and marketers at these streaming services to extract audience and potential audience feedback. These companies can improve their content accordingly. This technique of retrieving feedback from social media platforms, known as 'social listening', is already being employed by Netflix[20]. However, since it is not known exactly what machine learning and data science methods are used by the company, this paper will be focused on the types of data science methods that are likely to be suitable for this kind of 'social listening'. The main aim of this paper is to therefore use the natural language processing (NLP) methods of sentiment analysis and LDA topic modelling on Twitter and Reddit data about the Disney+ series Ms. Marvel to see what can be learned in terms of audience opinion about the series. This paper shows that using sentiment analysis on social media data as a way of defining the direction of topic modelling on the text, is a good way to know what themes are associated with positive sentiment and what themes are associated with negative sentiment.

CHAPTER 2

Introduction and Literature Review

When the streaming site Disney+ launched in November 2019, the streaming site became a major competitor for streaming sites like Netflix and Amazon Prime. Disney has produced thousands of blockbusters and hit shows and announced it would stop distributing most of its content to Netflix after launching the streaming platform [29]. In addition, since Disney owns Marvel Studios and the rights to many other franchises, the company announced a wide array of original content to appeal to new subscribers. One of Disney's franchises is the popular superhero comic brand, Marvel. Disney viewed Marvel as having good 'strategic fit' for the company's product portfolio and would allow for them to 'maintain its primary brand appeal and target markets'[35].

Marvel announced the release of the series 'Ms. Marvel' on August 24th 2019, and said it would be available to watch on Disney+. This was just one of many new projects to appeal to social media users' desire for more diversity in the franchise (this is the first Marvel series with a Muslim woman superhero)[15]. However, once the series was released on Disney+ in June 2022, multiple news reports claimed the series initially had the lowest viewership out of all of Disney+' recent releases and was panned on IMDb after the release of the first episode [5]. On the other hand, the series was also critically acclaimed on Rotten Tomatoes as well as sites like IGN [15]. Due to the mixed opinions about the show as well as the fact that it is a stark example of Disney and Marvel producing a show in order to appeal to the fanbase and audiences on social media, this study will analyse social media data about the series 'Ms. Marvel'. We hope to perform a sentiment analysis on Twitter and Reddit data

about the show in attempt to: 1.) Classify tweets and Reddit posts as positive or negative. 2.) Perform a separate topic model on positive and negative data to see what key words and key themes are associated with positive sentiment, compared to what is associated with negative sentiment. 3.) Prove that this is a method of gaining audience feedback. In order words, the aim of this paper is to see if social media analysis methods like sentiment analysis and topic modelling are useful for producers, writers and marketers to see what audiences think and say about their content. We are applying this question to Disney+, specifically their series 'Ms Marvel'.

Perhaps one of the most relevant forms of existing literature about this topic is Cara Macdonald's paper "Listening in: Investigating Social Media Activity in the Streaming Service Industry". MacDonald writes that social listening helps to 'identify opportunities to engage with consumers' and discover 'pain points' from those who 'take to social media to complain'[20]. Word of mouth is also a driving force behind the effectiveness of discussions on social media about the products/content put out by streaming platforms. MacDonald analyses tweets to examine discussion about streaming platforms themselves. The paper showed that at the time of data collection, Netflix holds a 'pioneer advantage' in terms of Twitter mentions [20]. This makes sense since it holds the largest number of subscribers. Hulu, a streaming platform whose shares were largely acquired by Disney in 2019, still held more mentions than Disney+ at the time this data was collected [20]. Variations of mention frequency based on region are also noted in the paper. Most tweets are in English and from the USA and UK. But French tweets about Disney+ were also very prevalent. As a result, the paper suggests that Disney+ should also consider its French audience when thinking about customer and audience feedback [20]. However, Macdonald's paper only examines social media posts about the brands of the streaming platforms (Netflix, Hulu and Disney+) themselves rather than social media posts about their content. Equally, the study is not a data science study on social media and instead uses the social listening tools Awario and Talkwalker. The paper does not look at data from other platforms besides Twitter where many people discuss streaming platforms and their content, namely, Reddit.

The major streaming platform Netflix uses big data analytics from the platform itself. For example, monitoring when users stop watching a series or movie, what genres they tend to prefer, when they pause and the ratings they give to content [6]. In addition, Netflix have previously predicted the success of shows like House of Cards and Stranger Things by

noticing the preferences of users e.g. that users had a preference for watching films starring Kevin Spacey and enjoyed the British version of House of Cards [24]. One of Netflix's marketing strategies for their originals, Bird Box and Bandersnatch allowed Netflix subscribers to influence the direction of the episode [4]. This marketing strategy bears the most resemblance to the aims of this paper - letting user feedback influence the direction of a series or movie. Despite this, there is little information or literature available to provide sufficient detail about Netflix's social listening strategy and whether these streaming platforms mobilise user opinion with social media analytics at all. Awario and Talkwalker are both very powerful social listening tools with various major clients and businesses that use their analytics, but we do not know if their clients include Disney+ and Netflix [20].

Therefore, it is useful to examine literature about how other businesses besides entertainment streaming platforms use social media analytics as customer feedback. As mentioned before, Awario and Talkwalker both utilise AI and deep learning techniques, especially sentiment analysis, to provide insights to their clients [20]. Talkwalker used advanced analytics and visualisations to 'monitor the evolution of sentiment across time' and automate 'reports on a weekly and monthly basis' for Dubai TV (a TV channel available primarily in the Middle East and North Africa) to gain more insight into audience opinion and feedback [36]. They noted that some of Dubai TV's challenges before hiring Talkwalker were that sentiment analysis 'didn't look at the whole picture, especially with popular shows that had polarizing opinions among viewers' and that it was difficult to prove that social media conversations were a reflection of the TV viewers' voices. Talkwalker claimed to be responsible for the growth of social media reach by roughly 250%, saying that their use of analytics and social listening led to better decision making in Dubai TV and improvement in the overall audience experience [36]. The Singapore Management University reports that companies have also used social listening tools to generate sales, by analysing tweets and predicting whether or not a customer would be interested in a certain product based on their tweets [20].

In their discussion of Big Data and Creative Industries, Morelli and Spagnoli highlight that Big Data Analytics not only provide businesses an opportunity for better decision making, but also the opportunity to hear previously unheard voices and improve social inclusion [16]. Thus, Big Data can significantly aid creative industries like that of film and TV on streaming services. However, one obstacle faced by businesses in creative industries is a 'shortage of talent' of people with 'deep expertise in statistics and machine learning' [16]. This means

that content producers and writers may often miss out on learning from Big Data due to a lack of skill or knowledge in this particular domain. Therefore, Big Data can really support Digital Creative Industries in developing new business activities and services, as well as enabling companies to reach more users and better target their needs. While data science studies of social media opinion about films and television series', as well as art in general are limited, multiple papers have performed different data science methods on movie reviews. For example, Songpan and Moolthaisong performed an emotion classification on 700 movie reviews from MetaCritic [18]. The paper firstly analyses the reviews by looking at the most frequent words with a word cloud (to see some general themes in the reviews), and then uses TF-IDF to vectorize the text and classify the emotions of the reviews based on those results. The paper found that this method of classification had a recall of 93% for positive reviews and 55% for negative reviews [18]. In addition, researchers at Princeton University analyse the ratings of Oscar nominated movies on Twitter compared to mainstream film review sites, and answer the question of whether such data can be used to predict box office success. They found that for most of the movies tracked, the number of positive reviews on Twitter exceeded the number of negative reviews and argue that 'such a large positive bias may be due to the psychology of sharing positive and helpful image among followers'[13]. Thus, when developing general marketing strategies, sellers and distributors may want to focus on enhancing the 'already high proportion' of positive reviews on social media sites, and use virality to gain audience members [13]. They also found that Twitter users are generally more positive about films than review sites like IMDb and Rotten Tomatoes, but a high score on both social media and IMDb may be more predictive of box office success [13].

Khan et al. use bag of words feature extraction, Naïve Bayes sentiment analysis, and then word2vec (a Python package that allows for the vectorization of words) for semantic clustering and categorising the movie reviews [1]. This paper even recommends this technique for audiences and executives at companies like Netflix to see an empirical summary of what viewers think about the content. So that audiences can see if the content is worth watching and producers can see where they went wrong [1]. They write that 'the job of review mining/classification and summarization (RCS) involves two steps: the former step is related to review classification, which categories the movie reviews as negative or positive, the latter step is related to review summarization, which produces a condensed summary from the movie reviews' [1]. Although Khan et al's research argue that Naïve Bayes is the

best method for sentiment classification of textual materials (like movie reviews in their case), the paper's methodology bears the greatest similarity to the methodology of this paper. We take an approach similar to the 'RCS' approach in Khan et al's paper. However, instead of movie reviews we are using social media data, and slightly different deep learning methods.

CHAPTER 3

Methodology

10,000 tweets were collected during the month of July 2022 from Twitter's API (Application Programming Interface) about the new series 'Ms. Marvel', released on the now major streaming platform Disney+. The data collection was performed using Twitter's Python library 'tweepy' with consumer keys and access keys provided by the site [26]. This data collection method was repeated for the Reddit API for which 7198 comments from two posts about the quality of Ms. Marvel. One post was a viral Reddit post from the 'Marvel Studios' subreddit (r/MarvelStudios) about the critical acclaim, but supposedly low viewership [27]. The other was 'final discussion' about the series, following the release of the season finale [**?**]. The comments of the posts were scraped using client keys provided by Reddit as well as the Python API library PRAW [25]. This data was then saved into a dataframe file and undergo pre-processing such as inputting into a pandas data frame, cleaning the text of stop words, html links and any other irrelevant characters before tokenization. Once the data is pre-processed, the sentiment analysis will be carried out. This sentiment analysis will then serve the basis for classifying posts as positive or negative. If the sentiment score given by the sentiment analyzer is above 0, the post will go into the data frame of positive posts. Whereas if the sentiment score given by the sentiment analyzer is below 0, the post will go into a data frame of negative posts. After this, a word cloud (showing the 50 most frequent terms in the data frame) will be created from the data frame of posts with positive sentiment and then from negative sentiment. This will help us to see what words are associated with positive sentiment and what words are associated with negative sentiment. Lastly, we will

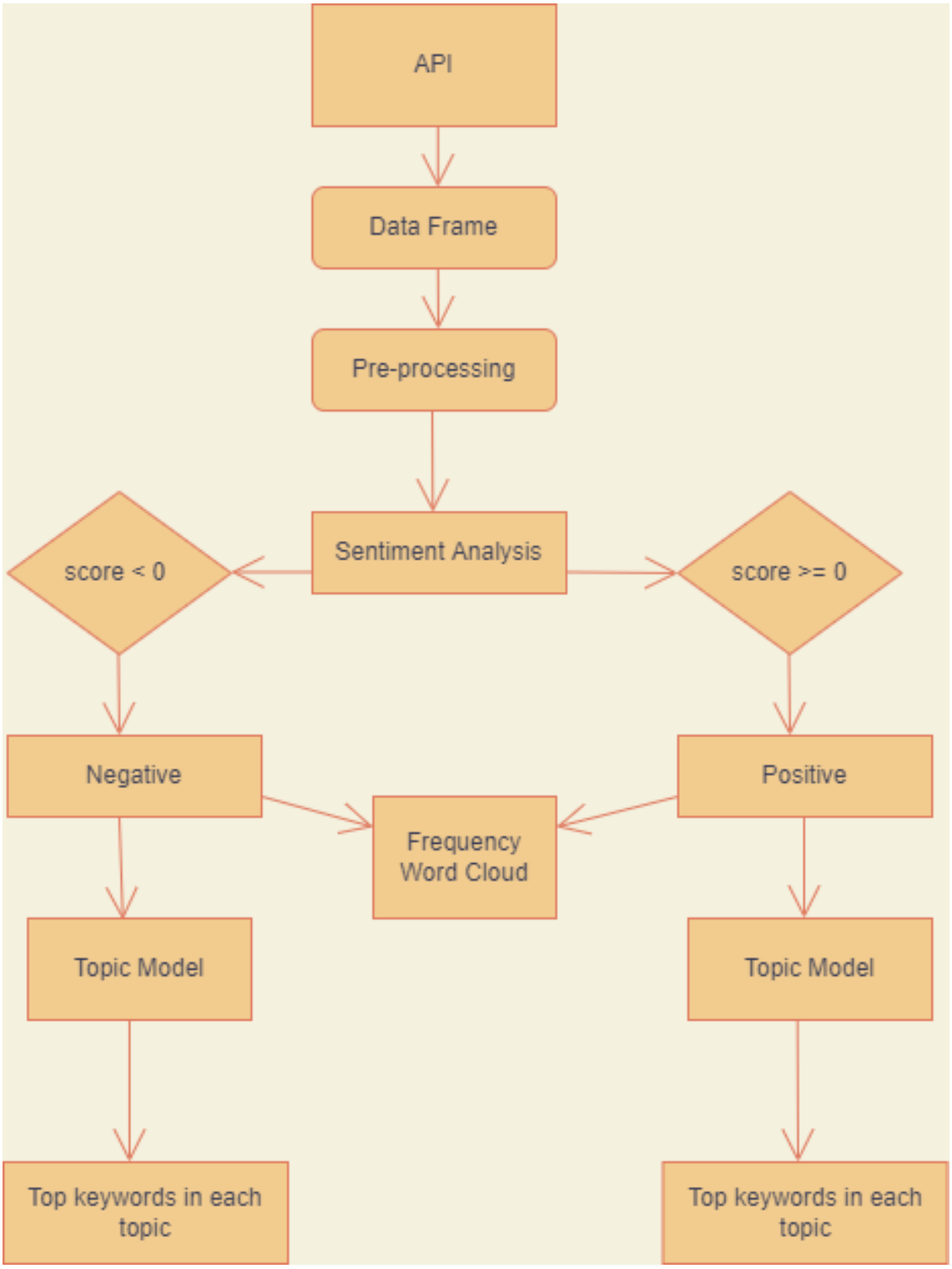

Figure 3.1: Flowchart showing methodology

perform an LDA topic modelling analysis on the positive posts and the negative posts - seeing what themes can be detected from posts that class as having positive sentiment about Ms. Marvel compared to the themes that can be detected from posts that class as having negative sentiment. Figure 3.1 visualises the methodology of this paper.

Hypothetically, such an analysis would help the producers of this series at Disney+ to gain insight about what viewers or potential viewers of a series are discussing in relation to the show. It may also allow them to speculate or learn about the opinions surrounding the show. Although they may need necessary information about context, thus an element of qualitative research may also be important. For the sake of identifying differences in the data analysis results for the two social media platforms, the Reddit and Twitter data will be analysed and summarised separately.

3.1 Sentiment Analysis

While there are various (typically machine learing, deep learning or lexicon based) methods of sentiment analysis, this paper will use a combination of lexicon based methods and deep learning because research shows that this combination is often more efficient than using single models [19]. For instance, sentiment analysis with R is often very limited due to the fact that it relies purely on lexicon-based approaches [23].

The word embeddings are derived from a text-file from Common Crawl's online archive. Common Crawl is a non-profit organisation that 'crawls' the web and gathers data from a variety of sites. The organisation makes all its data available online for researchers or businesses to use [8]. Once the word embeddings are extracted from the text file and put into a Numpy array, they are used to make sense of two positive and negative word lexicons (which are also in the form of a text file). In other words, a large text file of words considered 'positive' and a text file of words considered 'negative' are inputted into a list object. Then, the word embeddings from the Common Crawl data, for these positive/negative words are extracted and used to construct the desired input and output arrays.

The desired input (vector notations of positive and negative word embeddings) and output (target values of +1 for positive and -1 for negative sentiment) arrays are split into training and testing data. Once the input and output arrays are defined and split into training and testing data, they are used to fit the deep learning model for classification, which is

optimized with Stochastic Gradient Descent (SGD). The SGDclassifier will be employed with the Sci-Kit learn Python package. SGD classification models in Sci-Kit learn require two essential arguments, a loss function (which in this case will be 'log', denoting logistic regression) and the number of iterations.

$$E(w,b) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(w)$$

$$f(x) = w^T x + b$$

$$L(y_i, f(x_i)) = log(1 + exp(-y_i f(x_i)))$$

The formulae above represents the SGD classification technique. The first formula is for the minimization of regularized training error, which we use to find the model parameters. The vectors for the word embeddings act as the inputs ($x_1, x_2, x_3 ... x_n, y_n$) of the sentiment scoring function $f(x)$ in our second formula. L represents the loss function and R is the regularization term. The third formula shows the loss function L for the type of loss function we have chosen, which is logistic regression. The SGD classifier approximates the true gradient of $E(w,b)$ by considering one single training example at a time and implementing a first-order SGD learning routine [30]. The algorithm for this classifier iterates over the training input and updates the model parameters. The classifier model used has an estimated prediction accuracy of 93.96681749622925%, and a maximum number of 100 iterations.

We are essentially taking a logistic regression and fitting it with stochastic gradient descent. The model can predict an output value based on its inputs, but the way a sentiment score is retrieved by calculating the log probability of positive sentiment minus the log probability of negative sentiment (from the SGD classifier), and calculating its mean, when the algorithm is given a sentence or dataset of text.

$$sentiment = \mu(P(+|x)) - (P(-|x))$$

The simple formula above shows how the sentiment score per row (of the data frame) is calculated using the mean of the log probabilities that the row has positive sentiment, minus the log probability that the row has negative sentiment.

In simple terms, any form of text - in this case a Reddit post or Tweet, is passed through a function that matches the words in that text with its vector notated word embeddings and calculates an overall sentiment score using the mean of the log probabilities of positive or negative sentiment calculated with the SGD classifier.

This method of sentiment analysis was chosen since one of the key parts of this methodology is classifying posts. We must classify posts into either positive or negative, so that we can put them into separate dataframes and perform separate topic model analyses on those posts with positive sentiment compared to those with negative sentiment. While Naïve Bayes is another model used for classification tasks with natural language processing instead of logistic regression, it is considered less accurate in calculating probabilities that a sentence is in the positive class or negative class. This is because logistic regression is much robust to correlated features and is a common default because it 'generally works better on larger documents or datasets[10].

3.2 Topic Modelling

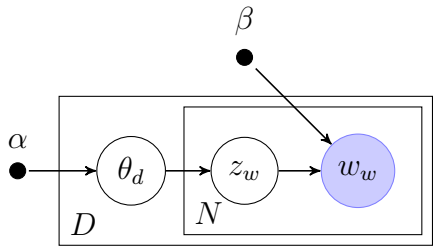

Figure 3.2: Diagrammatic illustration of LDA topic modelling generative process.[11]

In the visualisation of LDA (figure 3.2), α is the parameter that sets the topic probabilities per document. θ_d represents the topic distribution for a given document (D). N refers to the number of words. z_w is the topic for the jth word in the document. β denotes the word probabilities per topic. w_w is the output of observed word in a topic. For more information 'Latent Dirichlet Allocation'

LDA (Latent Dirichlet Allocation) topic modelling was chosen above NMF (non-negative matrix factorisation) because LDA is the most widely used and accepted method in text analytics [2]. LDA assumes that each document (in other words, each tweet or Reddit post) is a mix of topics, and each topic is a mix of words. For the LDA topic modelling, we will

begin by taking the data frame of posts labelled as 'positive' by our sentiment classifier, retokenizing the words, putting them into a corpus, creating a term document matrix (where the columns are documents and rows are words), converting the term document matrix into a topic word matrix to find the most optimized document-topic distribution and topic-word distribution [31]. The aim is to identify which topics would most likely be generated by a particular tweet or reddit post, as well as which words are most likely to generate a particular topic.

In simple terms, LDA topic modelling takes a document, the human-defined number of topics (k) and randomly assigns each word in each document to one of the k topics. The results of the first iteration are optimized by reiterating over each document and each word, computing the percentage of words in a document that were assigned to the topic, and the percentage of times the word was assigned to the topic over each document. For instance, The LDA algorithm will then transfer a word from one topic 1 to topic 2 when the probability of the words in one document, and probability of words in every document being in topic 2, is greater than those same probabilities for topic 1. LDA is supposed to keep iterating until it has given the most optimized representation of words in each topics [2]. This paper will look at the top 10 terms for each topic (words that have the highest probability $p(w_w|z_w, \beta)$ of being in that particular topic), the top 30 most salient (noticeable) terms in the topic model, the intertopic distance between topics and the coherence score of the topic model for the tweets labelled as having positive sentiment, the tweets labelled as having negative sentiment, the Reddit comments with positive sentiment and the Reddit comments with negative sentiment.

$$salience(w) = \sum_{T} P(T|w) log \frac{P(T|w)}{P(T)}$$

The formula above shows how the salience of a term in the LDA topic model is calculated, where 'w' denotes the word and 'T' denotes the topic [17]. For more information the methodology behind the coherence scores and intertopic distance of topic models, see methodology behind pyLDAvis [7].

CHAPTER 4

Results

Starting with the analysis of Twitter data, figure 4.1 (below) is a bar chart showing the number of tweets classified as having positive sentiment compared to the number of tweets classified as negative sentiment.

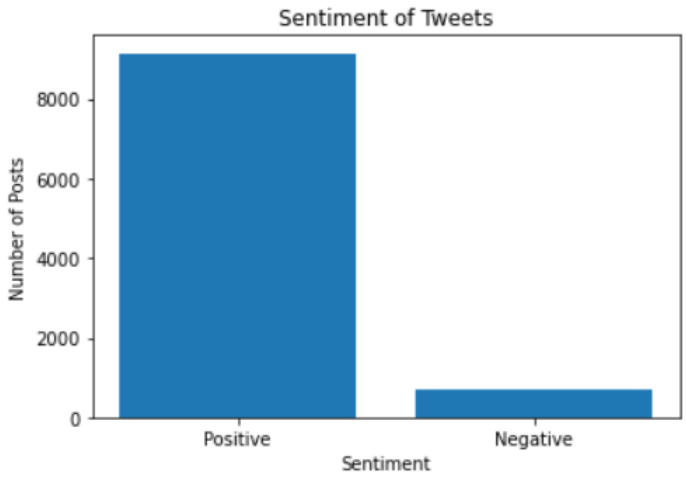

Figure 4.1: Count of Positive Tweets compared to Negative Tweets about Ms. Marvel during the month of July 2022, see code in Appendix C

We can see that according to the sentiment classifier, sentiment about the Disney+ series Ms. Marvel is overwhelmingly positive. 9143 tweets have been classified as positive. Whereas the number of tweets that were classified as negative is only 699. The remainder of the 10,000 tweets could not be classified by the sentiment analyzer. This means that 93% (of tweets that could be classified by the sentiment analyzer) are suggested to be making positive statements

in relation the show, and only 7% of tweets were suggested to be making negative tweets in relation to the show. Although Ms. Marvel is a series rather than a film, the number of positive tweets being much higher than the number of negative tweets, makes sense when remembering the fact that reviews of movies on Twitter were found to be generally much more positive than reviews given by movie critics, [13]. This may be used to suggest that opinion about the new series is generally positive, and that Twitter users are enjoying the show.

Moving onto the analysis of Reddit data. Figure 4.2 shows that sentiment was also found to be overwhelmingly geared towards positive sentiment. However, the percentage of negative comments was slightly higher. 5869 comments, or 81.5% (rounded to the nearest decimal point) of comments were found to have a positive sentiment score. Whereas 784, or 18.5% of comments were found to have a negative sentiment score. This suggests that there may be more negative statements being made in relation to Ms. Marvel on Reddit compared to Twitter, but that sentiment is still geared towards being positive.

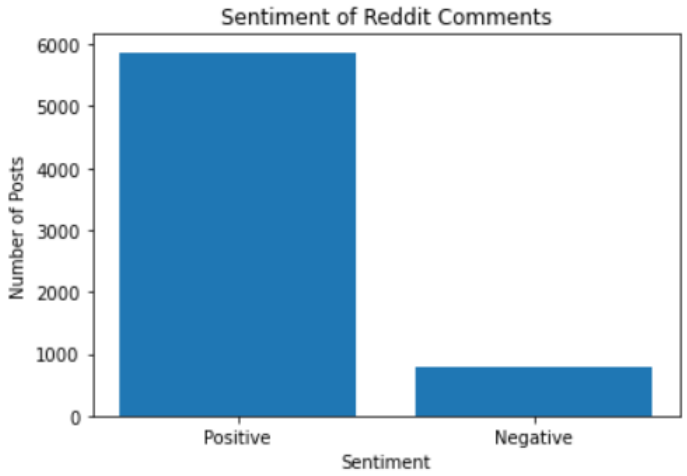

Figure 4.2: Count of Positive comments compared to Negative comments about Ms. Marvel on two popular posts from selected subreddits, see code in Appendix C

4.1 Positive Sentiment

It is important to note that the word 'marvel' had to be removed from the dataset during the cleaning and tokenization process. Although the word 'marvel' in this case refers to the superhero comic brand, the definition of the word itself meant its presence would risk

affecting the results of the sentiment analysis. Nevertheless, the overwhelming detection of positive sentiment was found despite the removal of the word 'marvel'.

Furthermore, the word cloud in figure 4.3 shows the 50 most frequent words in the data frame of tweets classified as having 'positive' sentiment. We can see terms like 'love', 'Kamala', 'show','mcu','good', 'great' and 'finale' in the word cloud. The lead actress' name 'Iman Vellani' is also visible in the word cloud of tweets with positive sentiment. The wordcloud for Reddit comments labelled as positive by the sentiment classifier is shown in figure 4.4. The words seem to be rather generic, as was also found in the word cloud for tweets labelled as having positive sentiment.

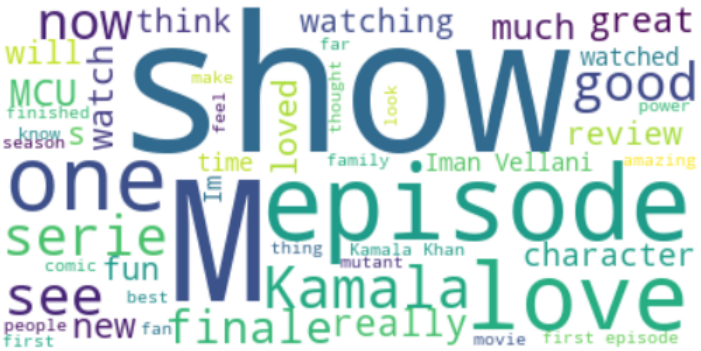

Figure 4.3: 50 most frequent words associated with positive tweets, see code in Appendix D

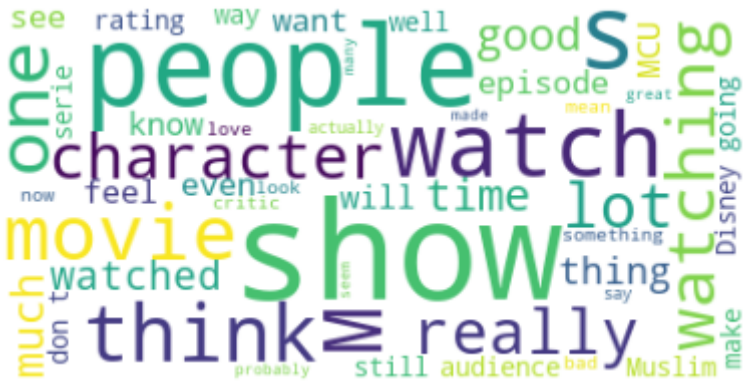

Figure 4.4: 50 most frequent words associated with positive Reddit comments, see code in Appendix D

The results of the LDA topic model for positive tweets are shown in figure 4.5. It is difficult to identify and distinguish exact themes from the topics, since many of them overlap and have the same words. Nonetheless, almost every topic apart from topic 7 and topic 4 include the word 'love' - so we can infer that many/most of the positive tweets are likely to

be talking about how much the users love the show, or aspects of it. For example, Topics 0 and 1 include words such as 'show', 'love', 'season', 'Kamala', 'episode', 'mcu' and 'great', suggesting these tweets may be users discussing their love for an episode they have just watched in relation to the main character, Kamala Khan, as well as the fact that it is from the 'MCU' (Marvel Cinematic Universe). In contrast, 30 most salient terms (see Figure 4.6) for tweets with positive sentiment found in the LDA topic model also show almost all the same terms to the word cloud, such as 'love', 'show', 'kamala' and 'episode'. But we can also see words not present in the word clouds such as 'bruno', referring to a supporting character in the series.

The topic model for tweets with positive sentiment has a chosen topic number of 10. However, the coherence score for the topic model for positive tweets is only 0.295. This means that the semantic similarity between the words in each topic is low. Thus, a different number of topics may have allowed for better results in our topic model. Especially since we are analysing a large amount of data. The LDA model for positive tweets has low inter-topic distance (as illustrated by Figure 4.7) between topic 1, 2, 3, 4, 5, 7 and 8. This means that the topics discussed in association with positive tweets about the series are broadly similar. These results make sense, given the fact that the words 'episode', 'love' and 'show' are present in every topic in the model. Only topics 9 and 10 have a high inter-topic distance from the other topics. Audiences on Twitter who are posting tweets with positive sentiment, may therefore have similar views about the Ms. Marvel series.

Moreover, the results of the LDA topic model on Reddit comments labelled as having positive sentiment is shown in figure 4.8 below:

Just like the LDA topic model results for the Reddit comments labelled as positive, there appears to be some words that occur in multiple topics. Namely words like 'show', 'people', 'episode' and 'watch'. Topics 0 and 8 seems to be about people discussing the main character in Ms. Marvel in relation to the MCU (Marvel Cinematic Universe). Topic 1 includes the word 'Muslim', suggesting that people are discussing the religious background of the main character in Ms. Marvel.

The intertopic distance for Reddit comments labelled as positive is moderate, as we can see in figure 4.11 that topics 1, 4, 2, 3, 5, 7, 6 and 10 all lie in principal component 2. Whereas only topics 8 and 9 have a high distance from the other topics. The marginal topic distribution (size) for topic 1 is especially high, meaning that many of the Reddit comments lie in topic 1 of

Figure 4.5: Top 10 terms for each topic in the LDA topic model for positive tweets, see code in Appendix E

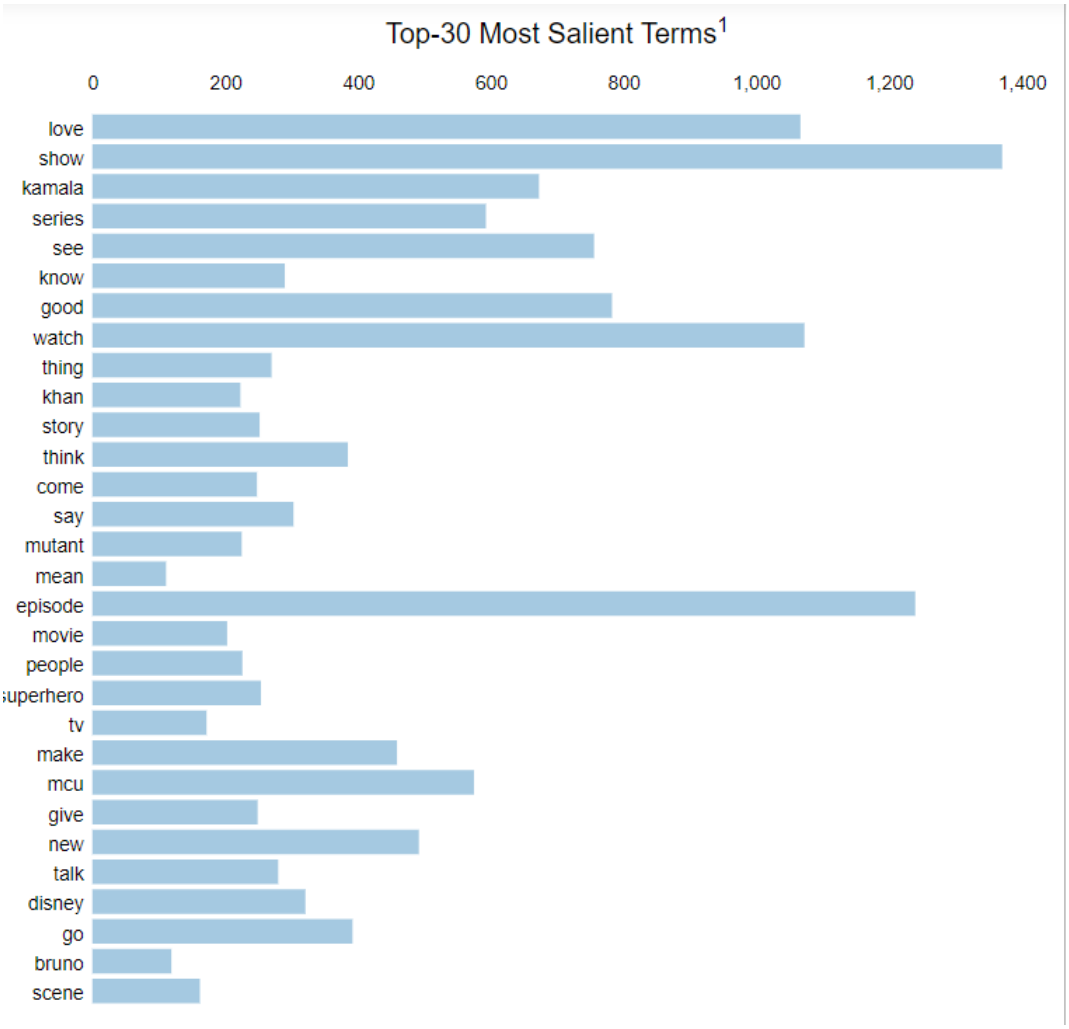

Figure 4.6: 30 most salient terms, visualised using pyLDAvis, see code in Appendix E

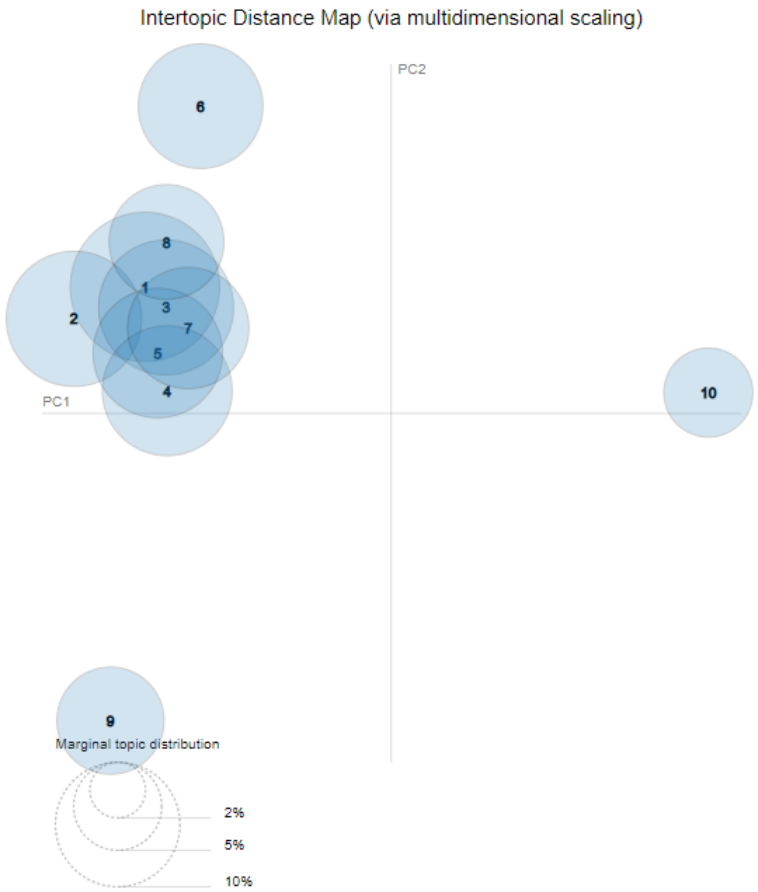

Figure 4.7: Intertopic distance map for LDA topic model on positive tweets, visualised using pyLDAvis, see code in Appendix E

Figure 4.8: Top 10 terms for each topic in the LDA topic model for positive Reddit comments, see code in Appendix E

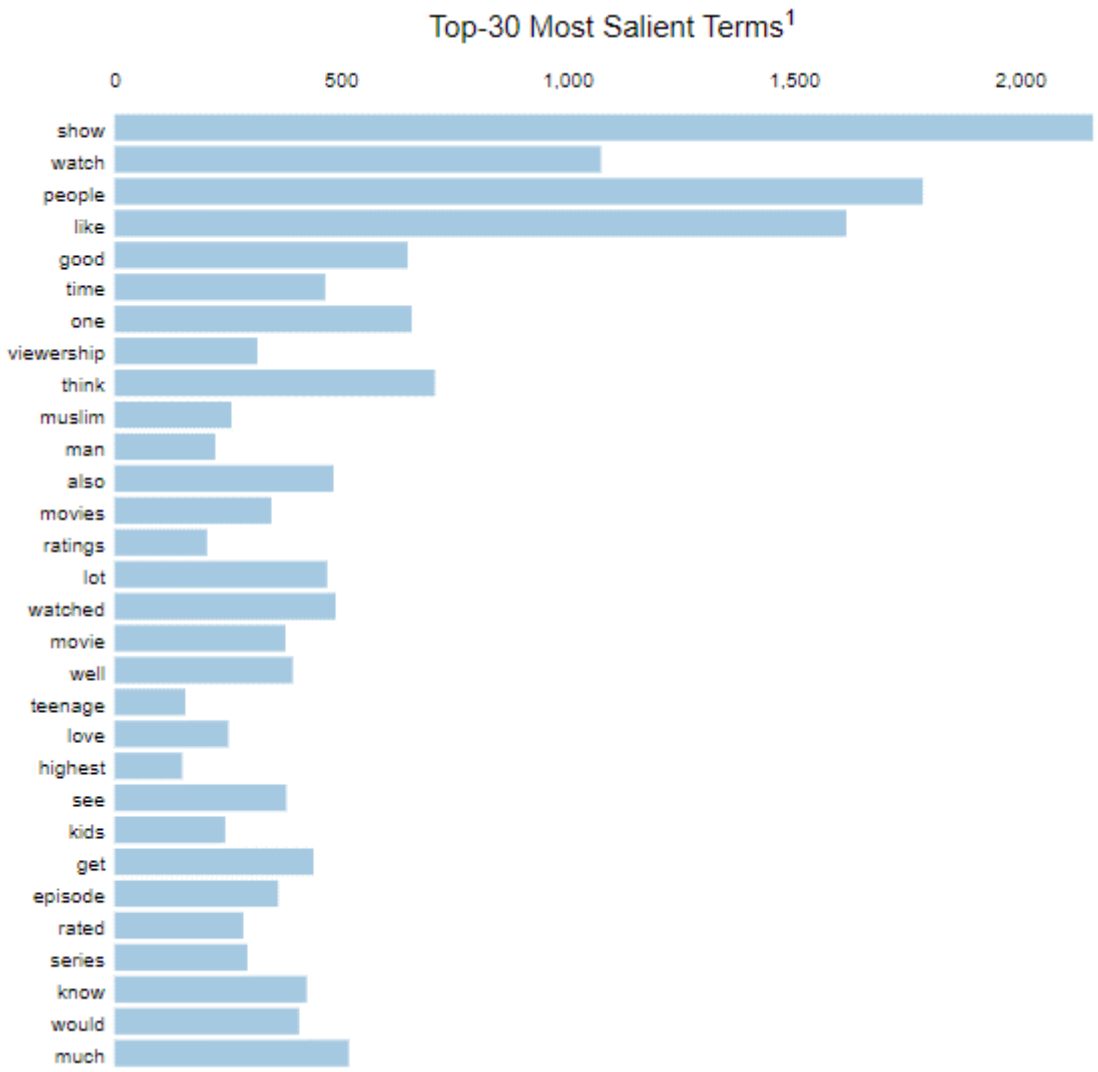

Figure 4.9: 30 most salient terms for LDA Topic Model on Reddit comments labelled as having positive sentiment, visualised using pyLDAvis, see code in Appendix E

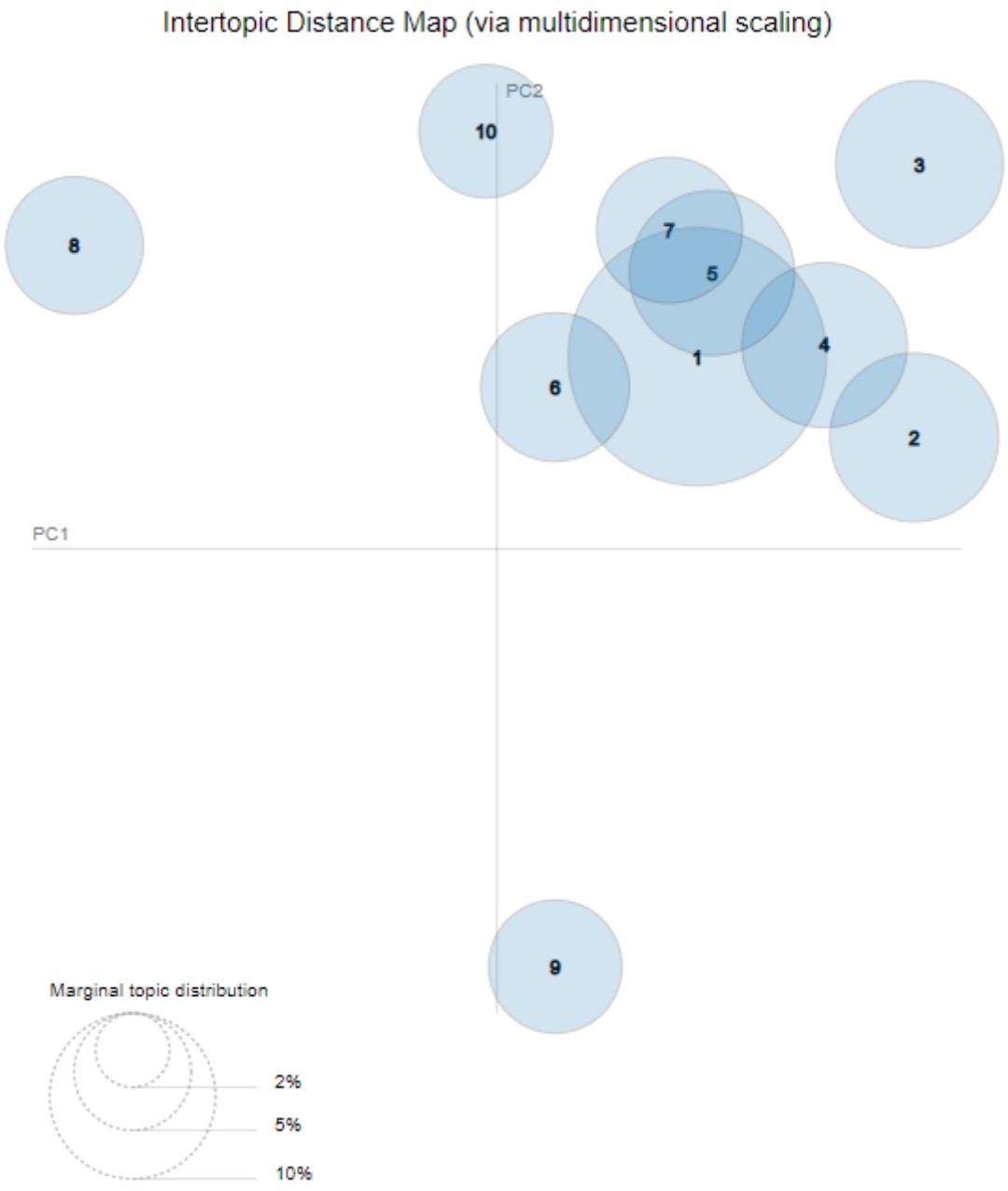

Figure 4.10: Intertopic distance map for LDA Topic Model on Reddit comments labelled as having positive sentiment, visualised using pyLDAvis, see code in Appendix E

the model. This once again suggests that when dividing the positive comments into 10 topics, the statements and terms being discussed by users are generally similar and overlap. Even so, like the topic model for Twitter data, this cannot be attributed largely to users making posts with positive sentiment are broadly making the same statements. The large amount of data but small chosen number of topics (10) likely play a significant role. The coherence score for the topic model on Reddit comments labelled as positive is even lower than it was for tweets labelled as positive - at 0.251. Showing that there is low semantic similarity between the terms in each topic - thus placing a limitation on the reliability of the topic model.

4.2 Negative Sentiment

For posts with negative sentiment. We have much less data since the sentiment classifier labelled over 90% of the tweets and Reddit comments were classified as having positive sentiment. Nevertheless, there is more variation in the results for negative posts, and the results provide more insight than for posts with positive sentiment.

Figure 4.11: Wordcloud showing 50 most frequent terms found in data frame of tweets with negative sentiment, see code in Appendix C

Figure 4.11 shows the word cloud of the 50 most frequently occurring terms in the data frame of negative tweets. 'Episode' is still a standout word, as is Kamala. This suggests that users are often discussing the main character of the show along with the content of each episode. There is also profanity, such as 'shit'. The word 'mutant' is also a frequent term in the negative tweets. Words such as 'racist', 'mcu', 'trash', 'review', 'spoiler' and 'shang chi' are also important to take note of, but from this analysis alone one cannot be sure whether users are referring to 'Ms. Marvel' itself as 'trash'.

As we can see in figure 4.12, 'people' is the most frequent word, found in both the negative and positive word clouds for Reddit data. This was not discovered as a frequently used term

in the word clouds for Twitter data. Notably, 'racist' is also the second most frequently used term for the reddit comments with negative sentiment. Other standout words include 'show' (as found in the Twitter data as well), 'rating', 'critic', 'Muslim', 'deleted' and 'bad'.

The tweets and Reddit comments that were labelled as having negative sentiment both seem to include negative adjectives in relation to the show, as well as profanity and discussion of identity like race and religion - hence the presence of 'racist' and 'muslim'.

Figure 4.13 shows the top terms in the topic model for tweets with negative sentiment. Topic 0 includes words such as 'hawkeye', 'captain' and 'red guardian' and 'okoye'. Thus it seems to be discussing Ms. Marvel in relation to other characters who have been portrayed in Marvel superhero movies. This may suggest that some users on Reddit compare the series 'ms. marvel' with other content created by Disney and Marvel studios, hence the mentions of 'shang chi' in the topic modelling results for negative tweets as well. Meanwhile, topics 1, 2, 3, 6 and 8 include the word 'mutant'. It is possible that tweets discussing a key plot point of the series (that the main character is actually a mutant) are being labelled as negative due to the semantic connotations of 'mutant' being a negative term, even though in the context of the series it is not necessarily negative. The most salient terms (in figure 4.14) for tweets with negative sentiment differ slightly to the most frequent terms in the word cloud for tweets with negative sentiment. In figure 4.14, we see notable words like 'bad', 'boring' and 'trash'. Once again, we cannot know for sure if users are using these adjectives to describe the series Ms. Marvel itself - but seeing as the posts were given a negative sentiment score by the sentiment classifier (and the analyzer uses the mean of the log probabilities of positive minus negative sentiment) for scoring, it is certainly possible.

The coherence score for the topic model on negative tweets is 0.527. This is higher than it is for the topic model on positive tweets. Thus meaning that the semantic similarity between words in each topic is high. The topic model for negative tweets therefore summarises the tweets with negative sentiment in relation to Ms. Marvel better than the topic model for positive tweets. This is likely to be a result of the much larger number of tweets classified as positive compared to negative. As shown in figure 4.15, there is much higher intertopic distance for tweets with negative sentiment compared to the intertopic distance for tweets with positive sentiment. This means that we can see more clearly what themes negative sentiment is associated with, compared to with positive sentiment.

For the topic model on Reddit comments (figure 4.16) labelled as having negative senti-

ment, we can see that 'racism' and 'racist' are present in all topics apart from topics 7 and 8. Topic 0 seems to be discussing viewership in relation to racism, which suggests users may be discussing whether one can attribute the reports of low viewership of Ms. Marvel to racism. Topic 4 includes 'obi_wan', 'Muslim' and 'racist'. Obi Wan is another show recently released by Disney+ around the same time as Ms. Marvel that suggests users may be debating whether lack of viewership can be attributed to racism or other content being released by Disney at the same time. Topic 7 includes words such as 'hate', 'bad', 'watch'. Whereas topic 8 includes 'critic' and 'review' and 'rating', suggesting that users are making negative statements in relation to the reviews of the show.

Figure 4.13 shows the 30 most salient terms found in the LDA topic model for comments labelled as negative. The bar chart goes hand in hand with the word cloud - highlighting 'racist' and 'people' as the most frequent salient terms. Words such as 'sexism' and 'boring' are also notable since they were not picked up by the word clouds but scored high for salience. Perhaps users are also attributing viewership and ratings to the extent of sexism, due to the female lead of the show.

The inter-topic distance map for the LDA topic model on negative Reddit comments (figure 4.18) shows large intertopic distance between most of the topics in the model - since we can see that the topics are distributed in every principal component of the graph - though there is slight overlap between topics 1 and 7, as well as topics 8 and 4. The topic model for negative reddit comments also has a similar coherence score to our topic model for negative tweets. The coherence score was found to be 0.532. Meaning there is only moderate semantic similarity between words in each topic. Yet again, the findings tell us that the topic model for negative sentiment is generally more reliable than the topic model for positive sentiment.

Further, the question we are asking is the following: what can the content producers of Ms. Marvel learn from this data to summarise audience feedback and make improvements?

Figure 4.12: 50 most frequent words associated with negative Reddit comments about Ms. Marvel, see code in Appendix C

Figure 4.13: Top 10 terms for each topic in the LDA topic model for negative tweets, see code in Appendix E

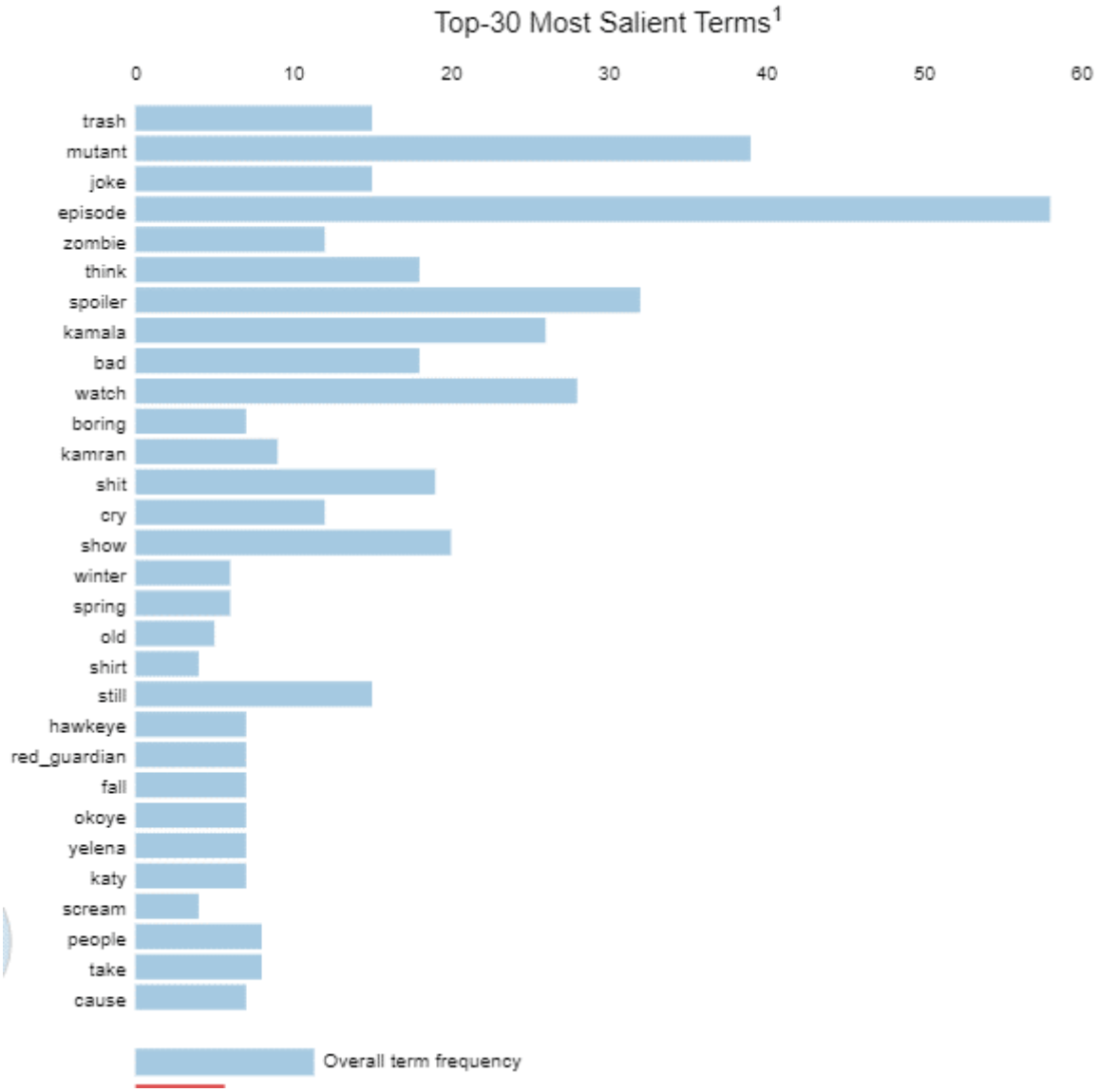

Figure 4.14: 30 most salient terms for tweets with negative sentiment according to LDA topic model, visualised with pyLDAvis, see code in Appendix E

Figure 4.15: Intertopic distance map for LDA Topic Model on tweets labelled as having negative sentiment, visualised using pyLDAvis, see code in Appendix E

Figure 4.16: Top 10 terms for topics in Reddit comments labelled as having negative sentiment, see code in Appendix E

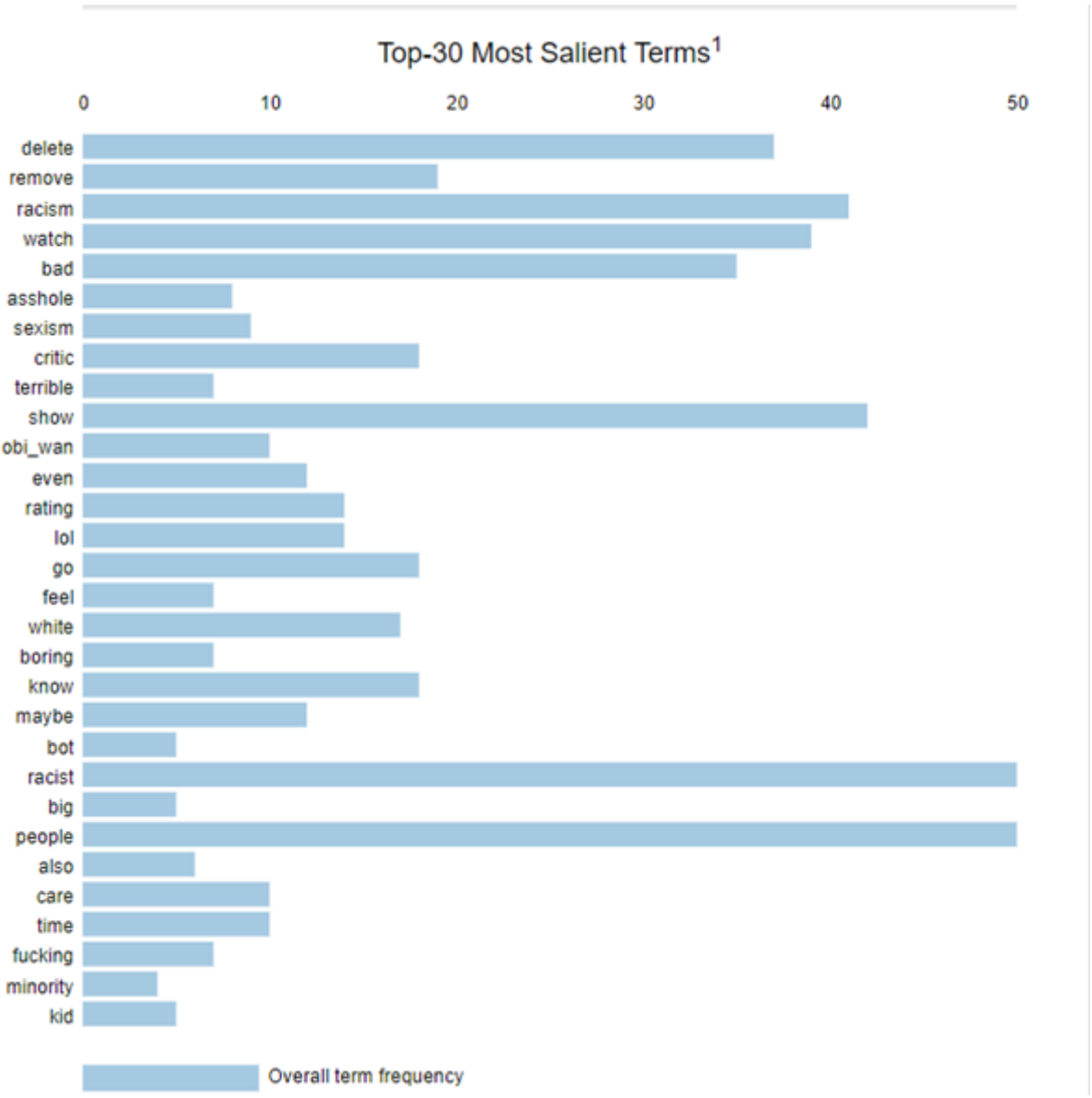

Figure 4.17: 30 most salient terms found in LDA topic model for Reddit comments labelled as negative, see code in Appendix E.

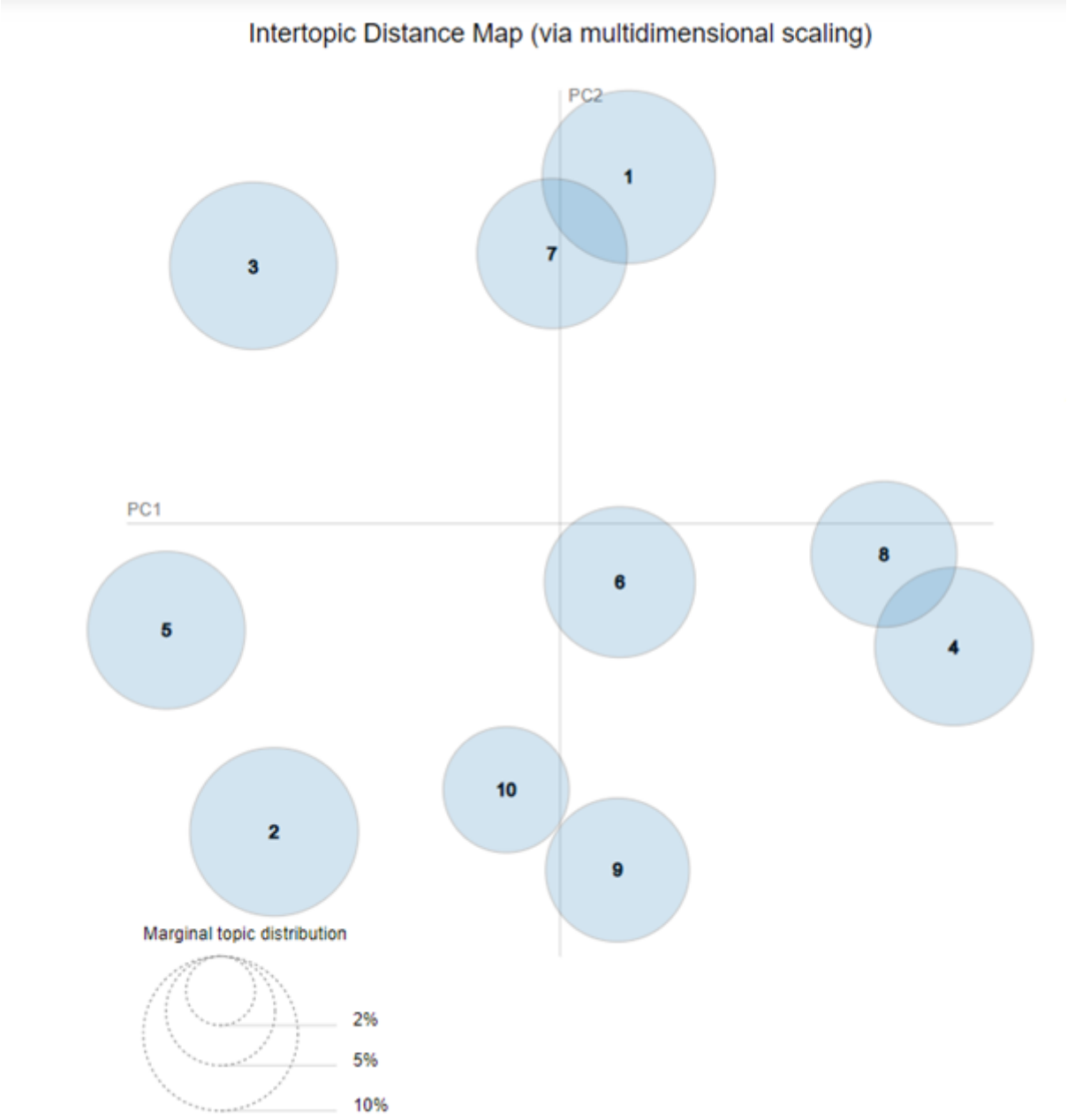

Figure 4.18: Intertopic distance map for LDA Topic Model on Reddit comments labelled as having negative sentiment, visualised using pyLDAvis, see code in Appendix E

CHAPTER 5

Discussion and Conclusion

The results of the sentiment analysis show that sentiment about the Disney+ series Ms. Marvel is overwhelmingly positive on both Reddit and Twitter. The topic models for positive sentiment give us a vague concept of what themes and topics are being discussed by audiences and potential audience members of the Disney+ series. We had better results for the results of the topic models for negative sentiment/ Due to the low intertopic distance between LDA topics for tweets and Reddit comments labelled as positive compared to negative, there is more to be learned from the results of our LDA topic model on social media posts with negative sentiment. Lack of contextualisation makes knowing exactly what users are saying about the series somewhat difficult, let alone thematic detection. However, as long as analysts have enough qualitative research and knowledge of the context of the data, they can still extract some level of audience feedback from the data, especially from the posts with negative sentiment since they tend to come in a smaller number. We have learned that racism, critical reception, the religious identity of the main character (Islam) and the plot points in the series e.g., the main character being a 'mutant' are significant points of discussion about the series [28]. Perhaps the large scale Big Data analysis can act as a tool for content writers, marketers and producers of shows like Ms. Marvel to then do a 'small data' analysis, focusing more closely (i.e., by looking at specific posts from the data) on what statements some users are actually making [9]. Equally, the results of the Big Data analysis can be used in combination with qualitative research (such as news articles and online reviews) to put the data into context and make sense of the results of the sentiment analysis and LDA topic modelling.

This is why many social scientists, data scientists and computer scientists advocate for a synergy between qualitative 'Small Data' analyses and quantitative 'Big Data' analyses [34]. This is because computers are still limited in their ability to understand text the way humans can.

An important question to ask is, to what extent can the results of this data analysis be generalised to actual viewers, audience members and subscribers to the Disney+ streaming platform? It must first be noted that neither Twitter or Reddit are representative of the general population. Twitter and Reddit users are more likely to speak English, live in America, and have a liberal or left-leaning political stance [22] [33]. Therefore, the demographic bias on these social media platforms means the results of the social media analysis are more likely to apply to Western viewers and subscribers than to non-English speaking audiences who live outside the USA. One could argue that the results may still be very applicable, since the majority of Disney+ subscribers may also from the United States. However, it is actually India that is estimated to have the highest number of subscribers to Disney+ (there are approximately 45,000,000 subscribers in India according to FlixPatrol), but are underrepresented compared to Americans on Twitter and Reddit [14]. The sentiment analysis and topic modelling techniques used in this paper do not allow us to analyse non-English text including posts in Hindi or another Indian language. Equally, there is also the issue of whether users discussing Ms. Marvel on these platforms are actual viewers or the show, they some may in fact be discussing it on Reddit or Twitter in anticipation rather than reaction to the series. Thus, the results of this data analysis can only be considered 'audience feedback' to an extent, it can also be viewed as feedback from 'potential audiences'.

Biases of the data sample should not be mitigated either. APIs from social media platforms do not always provide all the relevant data that is available on their platforms, due to the fact that they must protect user privacy and adhere to certain ethical considerations. For example, the Twitter API does not provide access to tweets from protected accounts (users who choose to keep their content private and only available to their followers). In addition, the rate limit restriction imposed by the Twitter API guidelines means that only roughly 1000 requests per hour can be made [12]. So, although tweets were collected throughout the month of July 2022 (during which new episodes of Ms. Marvel were being released on a weekly basis), there are likely to be some days of the month where the algorithm could not pick up any tweets due to the rate limit, which is easy to exceed. There is also an equal bias of the loudest voices being

the most heard, but not always the most important. Some Reddit and Twitter users, as well as subscribers to streaming platforms like Disney+, do not post as often as others. They instead 'read their newsfeeds and Twitter streams with great interest on a daily basis' but 'barely post anything to the stream themselves' [34]. In terms of the data from Reddit, although the rate limit is more easily bypassed, even less is known about the restrictions of the Reddit API and whether there is some invisible data that researchers cannot access [34]. The fact that comments were extracted from only two (albeit, very popular) Reddit posts about Ms. Marvel is also a significant contributor to sample bias. When users comment on a post, they are commenting in relation to the title of that particular post. Therefore, one of the Reddit posts was discussing reports of the supposedly low viewership of Ms. Marvel [27], meaning that users are probably more likely to be comparing Ms. Marvel to other content created by Disney+, and more likely to be discussing the most polarising aspects of the show. This may be why 'racism' was a significant topic in the Reddit posts with negative sentiment. Overall, sample bias is a significant limitation of any social media analysis like the one employed in this paper

Latent Dirchlet Allocation is a widely used and accepted approach to topic modelling and detecting themes in text. Even so, this paper found low intertopic distance (high overlap between topics when performing topic modelling on data with positive sentiment). This reinforces Vayansky and Kumar's critique of the LDA method for topic modelling. They argue that LDA is often ineffective and imprecise for large datasets and due to its assumption that topics are independent of one another, it does not account for correlation between topics - something that is often found in social media data [2]. LDA also does not consider the order of words within documents, assuming that documents and words within documents are independent This is inherent to the function of Dirichlet probability methods and thus is impossible to overcome. Therefore, although the topic models were somewhat helpful for speculating about the themes in posts with negative sentiment, it was far more limited for posts with positive sentiment and it may be preferred to use another topic modelling method better suited for social media data. Correlated topic models, NMF (non-negative matrix factorization), and the Pachinko Allocation model may be more suitable for the nature of data used for the methodology in this paper.

Although one criticism of sentiment analysis is that it fails to consider the context of words. For example, 'love' may be used in the sentence 'I didn't love the new show', but the

algorithm only picks out the word 'love' and considers it a reflection of positive sentiment. However, the expectation with the sentiment analysis technique used in this paper is that, because of the amount of vocabulary in the pre-trained word embeddings, the algorithm will still pick up on the relevent terms and that sentiment scores will average out to accurately classify posts as positive or negative [32]. Be that as it may, data scientists at ConceptNet discovered a dangerous AI bias in the word embeddings provided by Common Crawl in their GloVe dataset (the word embeddings used for the sentiment analysis in this paper). While AI bias is a difficult thing to avoid, a concerning finding was that the sentiment analyzer displayed a tendency to give different ethnicities, nationalities and races differing sentiment scores. The sentence 'Let's go get Italian food' had a sentiment score of 2.0429. Whereas the sentence 'Let's go get Mexican food' had a sentiment score of only 0.388 [32]. Both are neutral statements yet one has a remarkably higher sentiment score than the other. Even more concerning, the sentiment analyzer using Common Crawl's data was found to give the word 'Muslim' a negative sentiment score of -2.030210306934939, and 'Christian' a positive sentiment score [32]. As Narayanan and Caliskan's machine learning test found, 'text corpora contain recoverable and accurate imprints of our historic biases' even those as 'problematic as toward race or gender' [3]. Even if the bias isn't enormous, or is negated based on actual sentence structure, this sets a dangerous precedent for how the sentiment analyzer may classify some sentences on Twitter and Reddit compared to others.

A possible improvement to the methodology could be utilising the sentiment analysis not just for topic modelling but also in other valuable ways. For instance, researchers sometimes analyze the way social media sentiment changes over time [37]. This method would be particularly useful for analyzing the Twitter data, because it is easy to collect the date in which the tweets are created. This way, streaming companies can also see how sentiment changes after a new episode of a particular series is released, or the difference before and after a new movie is released. This would then help to understand what kind of reactions users are having to certain plot developments and so on. Alternatively, if more data could be provided by streaming platforms, it would also be useful to to establish some correlation between sentiment and viewership numbers of content on Netflix or Disney+, which would also help to know if the results of the sentiment analysis and topic model are representative of audiences.

This paper raises debates about what writers, producers, marketers and more roles at

streaming platforms can do with social media to learn about their audience and perhaps cater to their audiences more. Such a method can and is being used by other types of companies - as noted in the literature review. Another example of where the methodology in this paper can be applied is with music. For instance, music distributors can have a look at sentiment scores and perform topic models to see what songs from a particular album are associated with positive or negative sentiment. On the other hand, there are other considerations to remember. Violations of user privacy and surveillance may be possible if unethical methods of social media data mining or unapproved uses of APIs are used [34]. Although this data analysis only looks at publicly posted data that has been approved by Reddit and Twitter APIs - avoiding scraping any data about the users themselves - they still may sometimes be unaware that their data can be used in such a way. There is also the question of art - should 'data-driven' film and television series make the content of a show or movie solely dependent on its social media reactions and marketing value to define their direction? Could this kind of data science put pressure on creative industries to over-accommodate technocracy? This paper cannot answer such questions, but it shows that the future of screenwriting and understanding movie/TV audiences in creative industries in general (let alone streaming platforms) could well be very influenced by the pace of developments in AI and Big Data.

Bibliography

[1] M. Irfan Uddin Atif Khan, Muhammed Adnan Gul. Summarizing online movie reviews: A machine learning approach to big data analytics. *Scientific Programming*, 2020.

[2] Sathish P. Kumar Ayke Vayansky. A review of topic modelling methods. *Information systems*, 2020.

[3] Arvind Narayanan Aylin Caliskan, Joanna J Bryson. Semantics derived automatically from language corpora contain human-like biases. *Science, vol. 356, no. 6334*, 2017.

[4] Sophie Barber. What can 'black mirror: Bandersnatch' tell us about the media landscape? Available at https://theclickhub.com/what-can-black-mirror-bandersnatch-tell-us-about-the-media-landscape/ (2019/01/18).

[5] Adam Bentz. Ms. marvel review bombed on imdb after disney+ release. Available at https://screenrant.com/ms-marvel-show-review-bomb-release-details/ (08/06/2022).

[6] Zach Bulygo. How netflix uses analytics to select movies, create content, and make multimillion dollar decisions. Available at https://neilpatel.com/blog/how-netflix-uses-analytics/ (22/08/2022).

[7] Kenneth E. Shirley Carson Sievert. Ldavis: A method for visualizing and interpreting topics. *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, 2014.

[8] CommonCrawl. The data. Available at https://commoncrawl.org/the-data/get-started/ (2021/06/28).

[9] danah boyd. Critical questions for big data. *Information, Communication and Society*, 2012.

[10] James H. Martin Daniel Jurafsky. *Speech and Language Processing*. Stanford University, 3rd edition edition, 2021.

[11] Michael I. Jordan David M. Blei, Andrew Y. Ng. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[12] Twitter Developers. Developer platform. Available at https://developer.twitter.com/en/docs/twitter-api/rate-limits#v2-limits/ (2022/06/08).

[13] Mung Chiang Felix Wong, Soumya Sen. Why watching movie tweets won't tell the whole story? *Association for Computing Machinery*, 2012.

[14] Flixpatrol. Disney+ subscribers. Available at https://flixpatrol.com/streaming-service/disney/subscribers/ (2022/06/30).

[15] Emma Fraser. Ms. marvel: Series premiere review. Available at https://www.ign.com/articles/ms-marvel-premiere-review (08/06/2022).

[16] Francesca Spagnoli Giovanna Morelli. Creative industries and big data: A business model for service innovation. *Lecture Notes in Business Information Processing*, 2017.

[17] Jeffrey Heer Jason Chuang, Christopher D. Manning. Termite: Visualization techniques for assessing textual topic models. *Stanford University Computer Science Department*, 2014.

[18] Wararat Songpan Kamoltep Moolthaisong. Emotion analysis and classification of movie reviews using data mining. *IEEE*, 2020.

[19] Ana Gjorgjevikj Kostadin Mischev. Evaluation of sentiment analysis in finance: From lexicons to transformers. 2020.

[20] Kara MacDonald. Listening in: Investigating social media activity in the streaming services industry. 2020.

[21] Sidneyeve Matrix. The netflix effect: Teens, binge watching, and on-demand digital media trends. *Jeunesse Young People Texts Cultures*, 2014.

[22] Amy Mitchell Michael Barthell. Reddit news users more likely to be male, young and digital in their news preferences. *PEW Research*, 2016.

[23] Maurizio Naldi. A review of sentiment computation methods with r packages. *Computation and Language*, 2019.

[24] Greg Petraetis. How netflix built a house of cards with big data. Available at https://www.idginsiderpro.com/article/3207670/how-netflix-built-a-house-of-cards-with-big-data.html/ (2017/07/27).

[25] ReadtheDocs. Praw: The python reddit api wrapper. Available at https://praw.readthedocs.io/en/stable/ (2022/06/28).

[26] ReadtheDocs. Tweepy documentation. Available at https://docs.tweepy.org/en/stable/ (2022/06/28).

[27] Reddit. -ms. marvel: Highest rated vs. lowest viewership. can someone explain why please. Available at https://www.reddit.com/r/marvelstudios/comments/vo6sjo/ms_marvel_highest_rated_vs_lowest_viewership_can/ (2022/06/08).

[28] Julian Roman. Ms. marvel season one finale recap review: An uneven narrative gets multiculturalism credit. Available at https://movieweb.com/ms-marvel-season-one-finale-review-disney-plus/ (2022/07/14).

[29] Drew Ryan. Available at https://www.whats-on-netflix.com/leaving-soon/when-will-the-remaining-disney-shows-movies-leave-netflix- (2022/02/14).

[30] SciKitLearn. 1.5. stochastic gradient descent. Available at https://scikit-learn.org/stable/modules/sgd.html#sgd-mathematical-formulation/ (2022/06/28).

[31] Naha Seth. Part 2: Topic modeling and latent dirichlet allocation (lda) using gensim and sklearn. Available at https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-#:~:text=LDA/ (2021/06/28).

[32] Robyn Speer. How to make a racist ai without really trying. Available at http://blog.conceptnet.io/posts/2017/how-to-make-a-racist-ai-without-really-trying/ (2017/07/13).

[33] Adam Hughes Stefan Wojcik. Sizing up twitter users. *PEW Research*, 2019.

[34] Anja Bechmann Stine Lomborg. Using apis for data collection on social media. *The Information Society*, 2012.

[35] Jordan Sturgill. Beyond the castle: An analysis of the strategic implications of disney+. *EAST TENNESSEE STATE UNIVERSITY STATE UNIVERSITY*, 5:4–33, 2019.

[36] Talkwalker. How dubai tv channels use talkwalker's social intelligence to make data-driven decisions. 2022.

[37] Mike Thelwall. Sentiment analysis and time series with twitter. *Twitter and Society*, 2014.

Appendix A - Python Code for Data Collection

```
# import relevant python packages
import tweepy
import csv
import html
# define consumer keys and API keys by assigning them to a variable
consumer_key=""
consumer_secret=""

access_token=""
access_token_secret=""

# authenticate twitter handle
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

# create variable for inputing search term
sw = input('Enter␣Search␣Term:',)

# assign search word inputed by user to search_words
search_words = sw + "-filter:retweets"
```

```
# create variable for user to input how many tweets
tweet_num = input('How many tweets do you want?',)
# convert user input to integer
tweet_num = int(tweet_num)

# define tweets variable based on Twitter request
# Include and define all relevant variables
tweets = tweepy.Cursor(api.search_tweets, truncated = False,
q=search_words, lang="en",
wait_on_rate_limit = True,
tweet_mode = "extended").items(tweet_num)

# open and create a file to append the data to
csvFile = open('MsMarvel_tweets.csv', 'a')
csvWriter = csv.writer(csvFile)
    # use the csv file
    # loop through the tweets variable and add contents to the CSV file
for tweet in tweets:
    text = tweet.full_text.strip()
    #convert the text to ascii
    text_ascii = text.encode('ascii','ignore').decode()
    #split the text on whitespace and newlines into a list of words
    text_list = text_ascii.split()
    #iterate over the words, removing @ mentions or URLs
    (word.startswith('@') or
    text_list_washed = [word for word in text_list
    if not (word.startswith('@')
    or word.startswith('http'))]
    #join the list back into a string
    text_washed = ' '.join(text_list_washed)
    #decoding html escaped characters
    text_washed = html.unescape(text_washed)
    #write text to the csv file
```

```
    csvWriter.writerow([tweet.created_at, text_washed])
    print(tweet.created_at, text_washed)
csvFile.close()

# get reddit data
import praw
import pandas as pd
from praw.models import MoreComments
reddit = praw.Reddit(client_id="Lpun3-eV6GrcRgiJlO5ADw",
client_secret="mS7MTrynaCa9QlshKh6yP1u7C63ACg", user_agent="MSc-research")
subreddit = reddit.subreddit('msmarvelshow')
for submission in subreddit.hot(limit=5):
    print(submission.title)
    print("Submission ID = ", submission.id, '\n')
# get comments from "Episode 6: The Finale [Discussion Post]"
Post1 = reddit.submission(id='vxv1ft')
lst_comments = []
Post1.comments.replace_more(limit=None)
try:
    for comments in Post1.comments.list():
        lst_comments.append(comments.body)
except NotFound:
    pass
# create submission and comment object
# Get data about ms marvel discussion on r/marvelstudios subreddit
Post1 = reddit.submission(id='vo6sjo')
lst_comments = []
Post1.comments.replace_more(limit=None)
try:
    for comments in Post1.comments.list():
        lst_comments.append(comments.body)
except NotFound:
    pass
```

```
import pandas as pd
pd.options.mode.chained_assignment = None  # default='warn'
msmarvel = pd.read_csv('msmarvel_tweets.csv');
view = msmarvel[:2]
print(view)#view text

# get rid of line skips
msmarvel = msmarvel.iloc[::2]
# remove NaN's
print(msmarvel.head())
```

APPENDIX B

Appendix B - Python code for sentiment analysis classification

```
# import relevant packages
import numpy as np # for working with arrays and matrices etc.
import pandas as pd # for working with data frames
import matplotlib # for creating visualisations
import seaborn # uses matplotlib to create graphs
import re # regular expressions for parsing text
import statsmodels.formula.api
from sklearn.linear_model import SGDClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

def extract_wordlex(filename):
    # create word lexicons
    wordlex = []
    with open(filename, encoding='latin-1') as infile:
        for line in infile:
            line = line.rstrip()
            if line and not line.startswith(';'):
                wordlex.append(line) #add selected terms to word lexicon
```

```
    return wordlex

positive = extract_wordlex('positive-words.txt')
negative = extract_wordlex('negative-words.txt')

# remove words not present in word embeddings of GloVE lexicon
plus_vectors = embeddings.loc[positive].dropna()
# .loc[] searches list of positive words in embeddings array
# and gets the vector notation / embedding for each word
minus_vectors = embeddings.loc[negative].dropna()

# create arrays of desired inputs and outputs,
#+1 for positive sentiment and -1 for negative sentiment
vectors = pd.concat([plus_vectors, minus_vectors])
print(vectors)
targets = np.array([1 for entry in plus_vectors.index]
+ [-1 for entry in minus_vectors.index])
labels = list(plus_vectors.index) + list(minus_vectors.index)
print(labels)

# split input arrays (vectors) and output arrays (targets) into training and test
trainvec, test_vectors, traintarg, test_targets, train_labels, test_labels = \
    train_test_split(vectors, targets, labels, test_size=0.1, random_state=0)

# use stochastic gradient descent deep learning model,
#with a loss function of logistic regression
model = SGDClassifier(loss='log', random_state=0, max_iter=1000)
# fit the SGD classifier and train the model
#provide input array x of vectors, with target values
# / output of positive or negative sentiment scores
model.fit(trainvec, traintarg)

# prediction accuracy for words outside the embeddings
acc_score = accuracy_score(model.predict(test_vectors), test_targets) * 100
```

```
print("Estimated␣Model␣Accuracy:",acc_score,'%')

# create function to see the sentiment
def vector_sentiment(vect):
    # predict_log_proba gives the log probability for each class
    predictions = model.predict_log_proba(vect)
    return predictions[:, 1] - predictions[:, 0]
def words_sentiment(words):
    vect = embeddings.loc[words].dropna()
    # vectors defined by searching embeddings dictionary
    log_odds = vector_sentiment(vect)
    # vector sentiment is calculated by getting the log probability
    return pd.DataFrame({'sentiment': log_odds}, index=vect.index)

# get sentiment score for text
import re
TOKEN_RE = re.compile(r"\w.*?\b")

def sentiment_score(text):
    try:
        tokens = [token.casefold() for token in TOKEN_RE.findall(text)]
        sentiments = words_sentiment(tokens)
        return sentiments['sentiment'].mean()
    except KeyError:
        pass
# remove unnecessary columns from pandas dataframe
msmarvel.drop('Date', inplace=True, axis=1)
msmarvel.drop('Place', inplace=True, axis=1)
msmarvel.drop('Faulty_Text', inplace=True, axis=1)
# remove words beginning with '#'
msmarvel['Text']=msmarvel['Text'].str.replace('(\#\w+.*?)',"")
# remove word 'marvel' is this could severely alter our results
msmarvel['Text'] = msmarvel['Text'].str.replace('Marvel', '')
msmarvel.Text.to_string()
```

```
senti_scores = []
positive_tweets = []
negative_tweets = []
for row in msmarvel.Text:
    try:
        print(row)
        tweet_sent = sentiment_score(row)
        #print(sentiment_score(row))
        print(tweet_sent)
        senti_scores.append(tweet_sent)
        try:
            # here I am defining the decision boundary, is it good?
            if tweet_sent > 0:
                senti = "positive"
                print(senti)
                positive_tweets.append(row)
            if tweet_sent < 0:
                senti = "negative"
                print(senti)
                negative_tweets.append(row)
        except TypeError:
            pass

    except AttributeError:
        pass
    except ValueError:
        pass
# Analyse sentiment of Reddit comments about 'Ms Marvel'
senti_scores2 = []
positive_comments = []
negative_comments = []
for row in lst_comments:
    try:
        print(row)
```

```
        comment_sent = sentiment_score(row)
        #print(sentiment_score(row))
        print(comment_sent)
        senti_scores2.append(comment_sent)
        try:
            if comment_sent >= 0:
                senti = "positive"
                print(senti)
                positive_comments.append(row)
            if comment_sent < 0:
                senti = "negative"
                print(senti)
                negative_comments.append(row)
        except TypeError:
            pass

    except AttributeError:
        pass
    except ValueError:
        pass
```

APPENDIX C

Appendix C - Python Code for sentiment bar charts

```
# Add each list of tweets to a separate dataframe
positive_df = pd.DataFrame(positive_tweets, columns = ['Tweet'])
negative_df = pd.DataFrame(negative_tweets, columns = ['Tweet'])

print(len(positive_tweets))

positive_no = 0
for row in positive_tweets:
    positive_no = positive_no + 1
print(positive_no)

negative_no = 0
for row in negative_tweets:
    negative_no = negative_no + 1
print(negative_no)
# visualise number of positive reddit posts compared to negative Reddit posts
positive_no = 0
for row in positive_comments:
    positive_no = positive_no + 1
print(positive_no)
```

```
negative_no = 0
for row in negative_comments:
    negative_no = negative_no + 1
print(negative_no)
# number of comments overall
comments_no = 0
for row in lst_comments:
    comments_no = comments_no + 1
print(comments_no)


import matplotlib.pyplot as plt
Sentiment_Type = ['Positive','Negative']
Number_of_Posts = [9143,699]
plt.bar(Sentiment_Type,Number_of_Posts)
plt.title('Sentiment␣of␣Tweets')
plt.xlabel('Sentiment')
plt.ylabel('Number␣of␣Posts')
plt.show()

# Add each list of reddit comments to a separate dataframe
positive_comments = pd.DataFrame(positive_comments, columns = ['Comment'])
negative_comments = pd.DataFrame(negative_comments, columns = ['Comment'])

# remove word 'marvel'
positive_comments['Comment'] =
positive_comments['Comment'].str.replace('Marvel', '')
negative_comments['Comment'] =
negative_comments['Comment'].str.replace('Marvel', '')

Sentiment_Type = ['Positive','Negative']
Number_of_Posts = [5869,784]
plt.bar(Sentiment_Type,Number_of_Posts)
plt.title('Sentiment␣of␣Reddit␣Comments')
```

```
plt.xlabel('Sentiment')
plt.ylabel('Number of Posts')
plt.show()
```

APPENDIX D

Appendix D - Python code for word cloud

```
from wordcloud import WordCloud
# join posts together
positive_posts = ','.join(list(positive_df['Tweet'].values))
negative_posts = ','.join(list(negative_df['Tweet'].values))
wordcloud1 = WordCloud(background_color="white",
max_words=50, contour_width=3, contour_color='#CAFF70')
wordcloud2 = WordCloud(background_color="white",
max_words=50, contour_width=3, contour_color='#DC143C')
wordcloud1.generate(positive_posts)
wordcloud2.generate(negative_posts)
# Visualize the word cloud
wordcloud1.to_image()
wordcloud2.to_image()
```

APPENDIX E

Appendix E - Python code for LDA topic modelling

```
import gensim
from gensim.utils import simple_preprocess
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

stop_words = stopwords.words('english')
stop_words.extend(['marvel','ms','from', 're', 'use'])

def del_stop(texts):
    return [[word for word in simple_preprocess(str(doc))
             if word not in stop_words] for doc in texts]

def sentword(sentences):
    for sentence in sentences:
        # deacc=True removes punctuations
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))
# first clear the positive posts of stop words
posi_data = positive_df.Tweet.values.tolist()
posi_words = list(sentword(posi_data))
# remove stop words
```

```
posi_words = remove_stopwords(posi_words)
print(posi_words[:1][0][:30])

# Build the bigram and trigram models for positive sentiment
bigram = gensim.models.Phrases(posi_words, min_count=5, threshold=100)
# sets higher threshold fewer phrases.
trigram = gensim.models.Phrases(bigram[posi_words], threshold=100)
bigram_model = gensim.models.phrases.Phraser(bigram)
trigram_model = gensim.models.phrases.Phraser(trigram)
import spacy
def preprocess(terms, stop_words=stop_words,
allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):
    terms = [bigram_model[doc] for doc in terms]
    terms = [trigram_model[bigram_model[doc]] for doc in terms]
    out = []
    nlp = spacy.load("en_core_web_sm")
    for sent in terms:
        doc = nlp("␣".join(sent))
        out.append([token.lemma_ for token in doc
        if token.pos_ in allowed_postags])
    # remove stopwords once more after stemming
    out = [[word for word in simple_preprocess(str(doc))
    if word not in stop_words] for doc in out]
    return out
posi_words = preprocess(posi_words)

from pprint import pprint
import gensim.corpora as corpora
# Create Dictionary
id2word = corpora.Dictionary(posi_words)
# Create Corpus of posts
posi_texts = posi_words
# create Term Document Frequency matrix
corpus = [id2word.doc2bow(text) for text in posi_texts]
```

```
# View
print(corpus[:1][0][:30])
topic_no = 10
posi_lda_model = gensim.models.LdaMulticore(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=topic_no)

# Print the terms in the 10 topics
from pprint import pprint
pprint(posi_lda_model.print_topics())
doc_lda = posi_lda_model[corpus]
import matplotlib.colors as mcolors

cols = [color for name, color in mcolors.TABLEAU_COLORS.items()]
wc = WordCloud(stopwords=stop_words,
                   background_color='white',
                   width=2500,
                   height=1800,
                   max_words=15,
                   colormap='tab10',
                   color_func=lambda *args, **kwargs: cols[i],
                   prefer_horizontal=1.0)

topics = posi_lda_model.show_topics(formatted=False)

fig, axes = plt.subplots(5, 2, figsize=(10,10), sharex=True, sharey=True)

for i, ax in enumerate(axes.flatten()):
    fig.add_subplot(ax)
    topic_words = dict(topics[i][1])
    wc.generate_from_frequencies(topic_words, max_font_size=200)
    plt.gca().imshow(cloud)
    plt.gca().set_title('Topic' + str(i), fontdict=dict(size=16))
    plt.gca().axis('off')
```

```
plt.subplots_adjust(wspace=0, hspace=0)
plt.axis('off')
plt.margins(x=0, y=0)
plt.tight_layout()
plt.show()


# repeat for posts with negative sentiment
negi_data = negative_df.Tweet.values.tolist()
negi_words = list(sentences_words(negi_data))
# remove stop words
negi_words = remove_stopwords(negi_words)
print(negi_words[:1][0][:30])

# build bigram models for negative sentiment
bigram = gensim.models.Phrases(negi_words, min_count=5, threshold=100)
#sets higher threshold fewer phrases.
trigram = gensim.models.Phrases(bigram[negi_words], threshold=100)
bigram_model = gensim.models.phrases.Phraser(bigram)
trigram_model = gensim.models.phrases.Phraser(trigram)

negi_words = preprocess(negi_words)

from pprint import pprint
# Create Dictionary
neg_id2word = corpora.Dictionary(negi_words)
# Create Corpus of posts
negative_texts = negi_words
# create Term Document Frequency matrix
negativecorpus = [neg_id2word.doc2bow(text) for text in negative_texts]
# View
print(negativecorpus[:1][0][:30])
```

```
topic_no = 10
negative_lda_model = gensim.models.LdaMulticore(corpus=negativecorpus,
                                                id2word=neg_id2word,
                                                num_topics=topic_no)
# Print the terms in the 10 topics
pprint(negative_lda_model.print_topics())
doc_lda = negative_lda_model[negativecorpus]

import pyLDAvis.gensim_models as genismvis
import pickle
import pyLDAvis
import os

print(LDAvis_data_filepath)

pyLDAvis.enable_notebook()
pyLDAvis.gensim_models.prepare(posi_lda_model, corpus, id2word)
from gensim.models.coherencemodel import CoherenceModel
coherence = CoherenceModel(model=posi_lda_model, texts=posi_words,
dictionary=id2word, coherence='c_v')
coherencelda = coherence.get_coherence()
print('Coherence␣Score:␣', coherencelda)
```