University of Glasgow

COMPSCI5078
Web Science

Topic Modeling

Coursework Report

Submitted by:

Ahnaf Ismat Tasin

GUID 2739690T

Table Of Contents

1. Introduction

Topic modeling is a widely used technique in the field of natural language processing that has gained popularity due to its ability to extract meaningful insights from large text datasets. With the explosion of social media platforms, particularly Twitter, there has been an exponential increase in the volume of text data generated. As a result, topic modeling has become an increasingly important tool for analyzing and summarizing Twitter data. The ability to extract relevant topics from Twitter data can help gain insights into customer sentiment, monitor brand reputation, and identify emerging trends.

In this project, I aim to explore the application of topic modeling techniques on a dataset of tweets collected from Twitter and assess the effectiveness of various topic modeling algorithms in extracting meaningful insights from the data.

---

2. Single Tweets

2.1 Overview

We are provided with a tweets.csv file containing a set of tweets. Each row corresponds to a single tweet. A tweet possesses the following attributes:

- *id*: The Tweet ID, which can be utilized to reference the tweet on the Twitter website.
- username: The user who posted the tweet.
- *text*: The actual content of the tweet.
- *qScore*: A subjective quality score assigned to the tweet.

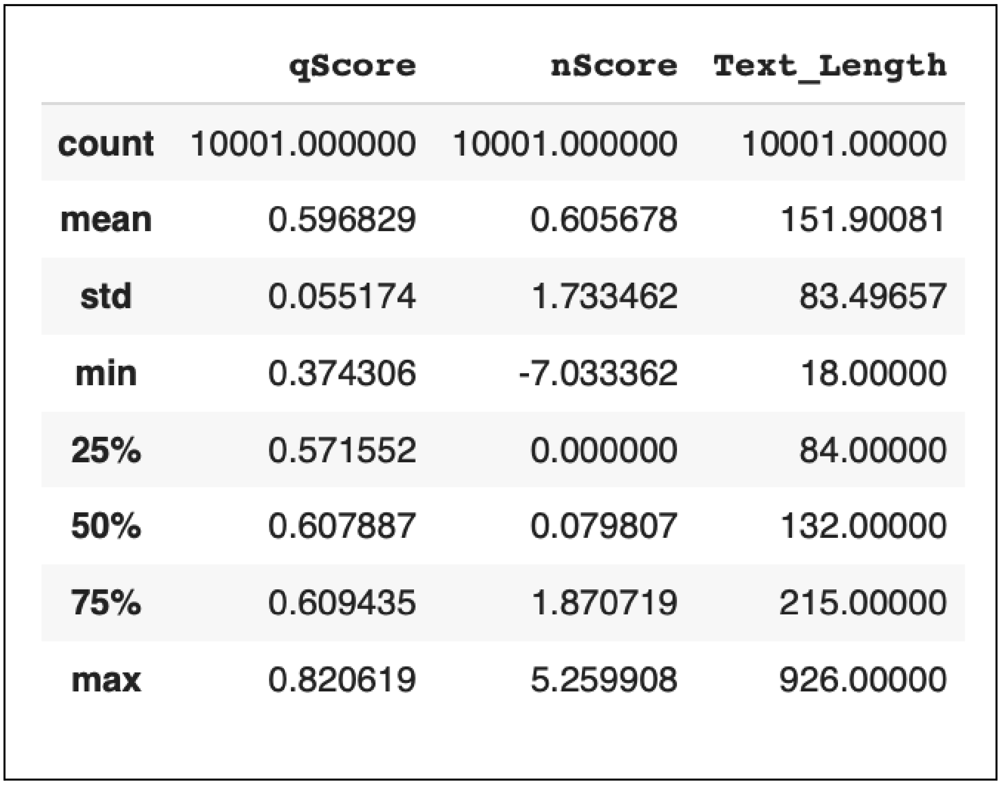
- *nScore*: An additional subjective metric evaluating newsworthiness based on the user profile and tweet content.
- We have a selection of 10,001 tweets to choose from.

2.2 Notable libraries used for the task

After initially using spaCy for pre-processing, I had to resort to using Gensim because after pre-processing the text, I discovered that spaCy does not have built-in support for bag-of-words vectorization.

2.3 Statistics

- I added a new column *Text_Length* to the DataFrame that contains the length of tweets.
- To get exploratory statistics from the Pandas DataFrame in Python, I used the *describe()* method.



	qScore	nScore	Text_Length
count	10001.000000	10001.000000	10001.00000
mean	0.596829	0.605678	151.90081
std	0.055174	1.733462	83.49657
min	0.374306	-7.033362	18.00000
25%	0.571552	0.000000	84.00000
50%	0.607887	0.079807	132.00000
75%	0.609435	1.870719	215.00000
max	0.820619	5.259908	926.00000

Statistics of tweet texts for single tweets

The *Text_Length* column is the collection of the tweet length statistics. For this attribute,

- The **mean** is approximately 151.9 which means every tweet gives an average of 151.9 characters of data to extract topics from.
- The standard deviation, **std**, is approximately 83.6 characters long and it indicates how much the length of each tweet varies.
- The smallest tweet length, **min**, was 18 characters long
- The **median** tweet length is 132 characters long
- The maximum tweet length, **max**, is approximately 926 characters long.

The *qScore* is the quality score of the tweet. For this attribute,

- The average quality score, **mean**, is approximately 0.597.
- The standard deviation, **std**, is approximately 0.055 which means the quality score doesn't vary as much from the mean
- The minimum quality score, **min**, is approximately 0.37
- The median quality score, **median**, is approximately 0.61
- The maximum quality score, **max**, is approximately 0.82

The *nScore* is the newsworthiness score of the user. For this attribute,

- The average newsworthiness score, **mean**, is approximately 0.606.
- The standard deviation, **std**, is approximately 1.73.
- The minimum newsworthiness score, **min**, was approximately -7.03
- The median newsworthiness score, **median**, was approximately 0.0798
- The maximum newsworthiness score, **max**, is approximately 5.26

2.4 Generic outline of topic modeling

Topic modeling is the process of using a topic model to discover the hidden topics represented by a large collection of tweets.

1. **Data collection**: For this coursework, we were provided with two CSV files containing tweet data.

2. **Data pre-processing**: Perform data cleaning to remove noise and irrelevant data from the tweets. This includes removing URLs, mentions, hashtags, and special characters, as well as converting text to lowercase and removing stop words.

3. **Tokenization**: Tokenize the tweets into separate words or phrases, which is a crucial preprocessing step that enables text data to be easily analyzed by various algorithms. The tokenization process involves several stages:
    a. First, remove any URLs in the text using regular expression substitution.
    b. Split the text into individual tokens based on whitespace characters and remove non-ASCII characters (employing the non-ASCII character removal function mentioned earlier).
    c. Filter out user mentions by eliminating any token that begins with the "@" symbol.
    d. Remove stop words (common words like "the", "a", "and", etc.) from the remaining tokens using nltk.stopwords.
    e. Remove any tokens containing numeric digits (utilizing the has num function). While this may eliminate some context, it results in a better overall outcome.
    f. Further clean each remaining token by dividing it into parts based on a regular expression and removing any empty parts, those beginning with "https" or "amp", or those present in the stopword list.
    g. Lastly, return a list of clean tokens.

4. **Lemmatization**: Implement lemmatization or stemming to decrease the word count and group words with similar meanings. Lemmatization involves reducing words to their base or core form, also known as the lemma. This process is used to standardize words with the same meaning but different inflections, such as "walking" and "walked," to their shared root form, "walk." The necessary WordNet data is first downloaded by the code from *nltk*. Utilizing a nested list comprehension, each token in every text from the list of tokenized texts, referred

to as "tokenized texts," is lemmatized. Consequently, a new list of tokenized texts is generated, with each token now paired with its corresponding base form or lemma.

5. **Create a document-term matrix**: Transform the preprocessed and tokenized tweets into a document-term matrix, where each row signifies a tweet, and each column represents a word or phrase from the corpus. The Dictionary class creates a mapping between words and their numerical identifiers based on a list of documents, with each document depicted as a list of tokens. Using the resulting dictionary, a document (expressed as a list of tokens) can be converted into a bag-of-words vector, where each vector element corresponds to the count of a specific word in the document. To filter out rare and common terms that may not be relevant to the investigation, the dictionary is scrutinized using the "filter extremes" method. The "no below" parameter determines the minimum number of documents in which a word must appear to be included in the dictionary. Words that occur in fewer than two texts in this context will be removed from the dictionary. The final dictionary will contain only terms that appear in at least two documents in the corpus, and each word will be assigned a unique integer ID. This dictionary can then represent the corpus of documents as a bag-of-words vector, allowing for further analysis using the LDA model we will train next.

6. **Topic modeling**: Apply a topic modeling algorithm, such as Latent Dirichlet Allocation (LDA), to the document-term matrix to extract the underlying topics. The key idea behind LDA is to represent documents as probability distributions over latent topics, and topics as probability distributions over words. The LDA algorithm aims to maximize the likelihood of observing the given collection of documents based on the estimated topic-word and document-topic distributions. The "num topics" parameter allows for adjusting the number of topics in the model.

7. **Evaluation metrics**: Evaluate the results using metrics such as perplexity, coherence, and topic distribution to determine the optimal number of topics and assess the quality of the topics extracted.

8. **Visualize**: Visualize the topics and their relationships using tools such as word clouds. Word clouds are a popular visualization tool used in topic modeling to display the most frequent words or phrases in a collection of documents. A word cloud typically consists of a collection of words or phrases that are arranged in a visually appealing way, with the size of each word indicating its frequency in the text data.

9. **Interpret results**: Interpret the topics and their relationships to gain insights into the underlying themes and patterns in the tweet data.

2.5 Pre-processing of tweet texts

It is important to note that topic modeling on the given tweets was challenging due to the noisy and informal nature of the text data. Therefore, careful preprocessing and parameter tuning was critical for obtaining accurate and meaningful results.

Initially, *nScore* was used to drop items from the initial data frame. I used the 25th percentile value of 0.0 as the threshold. Items below this threshold were dropped considering that their newsworthiness is negative. This step will ensure that a good chunk of spam is discarded from our dataset.

Then, the spaCy library was used for pre-processing the data. The *preprocess_text()* function in spaCy is a utility function that applies a series of text preprocessing steps to a given input text. The function takes a string of text as input and returns a processed version of the text.

Here are some of the text preprocessing steps that the *preprocess_text()* function applies:

- Lowercasing the text: The function converts all the characters in the text to lowercase.
- Removing whitespaces: The function removes any leading or trailing whitespaces from the text.
- Removing digits: The function removes any digits (0-9) from the text.
- Removing punctuation: The function removes any punctuation marks from the text.
- Removing stop words: The function removes any stop words (commonly occurring words like "the", "and", "is", etc.) from the text.
- Lemmatization: The function applies lemmatization to the text, which reduces words to their base or dictionary form. For example, the word "running" would be reduced to "run". At this step, I also used a command to only show nouns, adjectives, verbs, and adverbs.

To fine-tune the results based on my task, I modified the function to do the following additional tasks to help me reduce noise in the text data and make it easier to extract meaningful topics.

- Replacing the '\n' character with spaces.
- Removing spaces: The function removes any spaces from the text.
- Identifying and keeping only the words that correspond to nouns, adjectives, verbs, or adverbs. I got rid of any other words because they weren't important enough to be a topic by themselves.

2.6 Metrics studied

Kullback-Leibler divergence (KLD):

It is a measure of the dissimilarity between two probability distributions. In the context of topic modeling, KLD can be used to compare the topic distribution of a model to a reference distribution, such as a human-generated topic distribution or a topic distribution from a previous study.

To use KLD for topic modeling, one can compare the topic distribution of different models for different numbers of topics. The KLD is non-negative and is equal to zero if and only if two topics are identical.

KLD can be a useful metric for topic modeling, as it can provide a quantitative measure of how well the model captures the reference distribution. However, it should be noted that KLD alone may not be enough to determine the optimal number of topics. It is often used in conjunction with other metrics, such as perplexity and coherence, as part of a more comprehensive evaluation of topic models.

Perplexity:
Perplexity measures how well a topic model predicts a held-out test dataset. Lower perplexity values indicate better predictive performance. However, perplexity alone may not be enough to determine the optimal number of topics.
Coherence:
Coherence measures the semantic similarity between the top words in a topic. Higher coherence values indicate that the top words in a topic are more semantically related. This metric can help to identify topics that are more interpretable and coherent.

2.7 How the number of topics was chosen

1. First, a list of numbers called *num_topics_list* was defined and an empty list called *combine_list* was defined.

2. Next, each value of *num_topics* was looped through in the num_topics_range. Within the loop, an LDA model was created using the Gensim library, with the number of topics specified by *num_topics*.

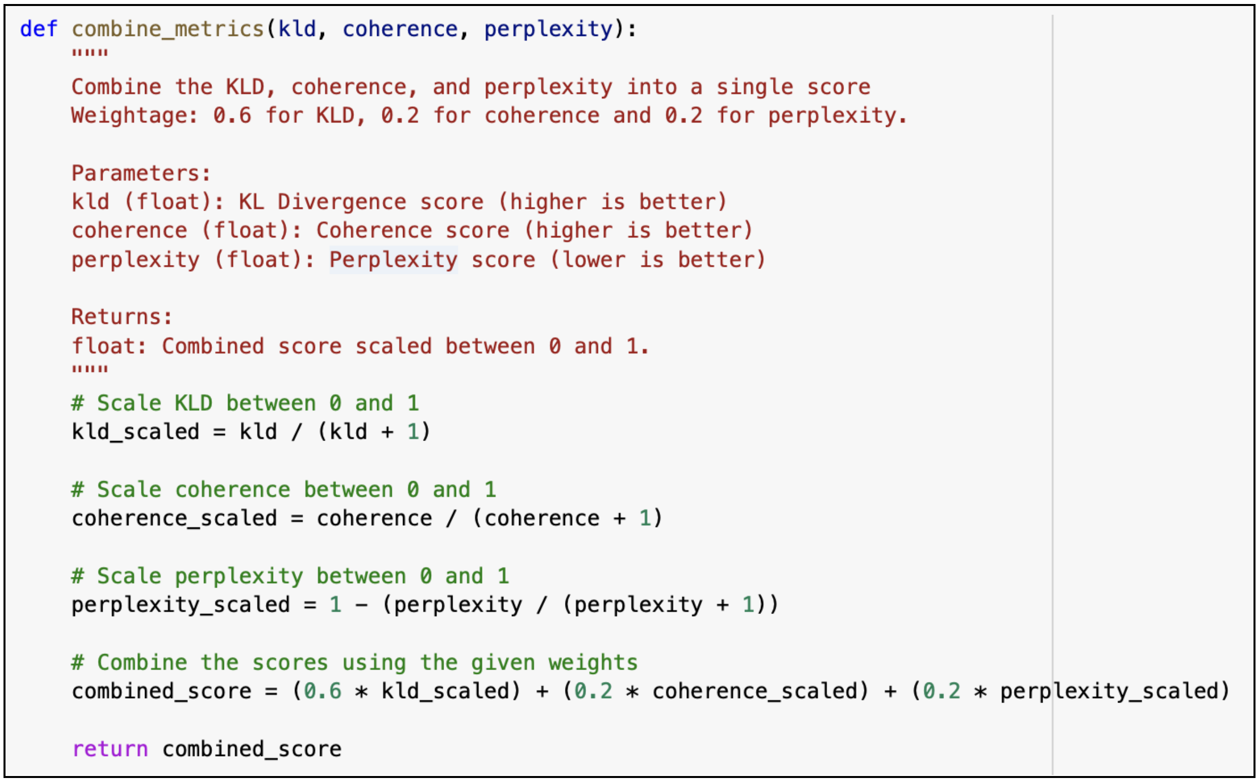
3. After training the model, the coherence score of the model was calculated using the *CoherenceModel* function from the Gensim library. The coherence score

measures how well the topics in the model are separated and how semantically coherent the words are within each topic.

4. The Kullback-Leibler Divergence (KLD) score, perplexity score, coherence score, and combined score for the current number of topics being tested were finally printed. The combined score is calculated by a custom function called *combine_metrics*, which takes in the KLD score, coherence score, and perplexity score as inputs and returns a single combined score.

In the general process of topic modeling, determining the optimal number of topics is a crucial step that directly impacts the quality and interpretability of the extracted topics. This decision is primarily influenced by the specific requirements of the task and, to a certain extent, relies on subjective judgment. To address this point, and to undertake a more creative approach, I wrote a custom function that employs a mathematical approach to systematically aid me in selecting the number of topics. By incorporating variables and their corresponding weights, the function enables a more objective, data-driven, and context-sensitive determination of the appropriate number of topics, thus enhancing the reliability and effectiveness of the topic modeling process.

My custom Python function, named *combine_metrics*, integrates the values of Kullback-Leibler Divergence (KLD), coherence, and perplexity into a unified output by assigning specific weights to each metric (0.6 for KLD, 0.2 for coherence, and 0.2 for perplexity). The rationale behind these weight allocations is grounded in the distinctive characteristics of each metric and their relative significance in evaluating topic models. By balancing these three metrics with their respective weights, the combine_metrics function enables a comprehensive and well-rounded evaluation of topic models, facilitating the selection of an optimal model that exhibits both interpretability and generalization capabilities.



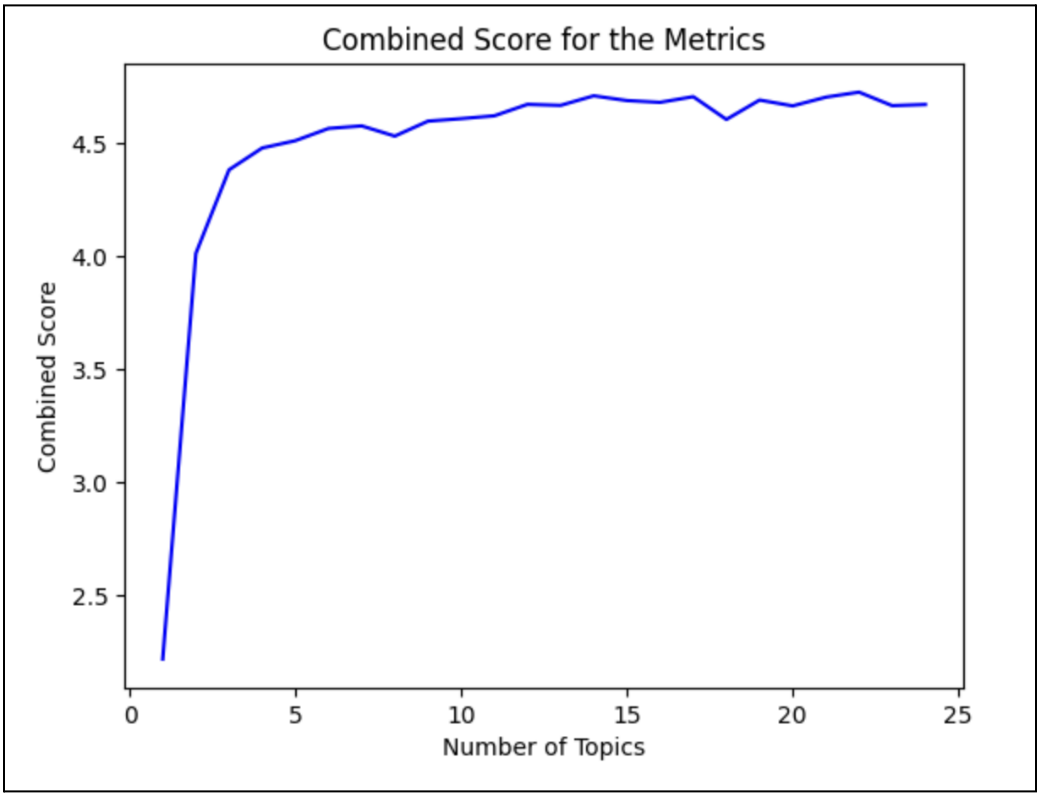
```python
def combine_metrics(kld, coherence, perplexity):
    """
    Combine the KLD, coherence, and perplexity into a single score
    Weightage: 0.6 for KLD, 0.2 for coherence and 0.2 for perplexity.

    Parameters:
    kld (float): KL Divergence score (higher is better)
    coherence (float): Coherence score (higher is better)
    perplexity (float): Perplexity score (lower is better)

    Returns:
    float: Combined score scaled between 0 and 1.
    """
    # Scale KLD between 0 and 1
    kld_scaled = kld / (kld + 1)

    # Scale coherence between 0 and 1
    coherence_scaled = coherence / (coherence + 1)

    # Scale perplexity between 0 and 1
    perplexity_scaled = 1 - (perplexity / (perplexity + 1))

    # Combine the scores using the given weights
    combined_score = (0.6 * kld_scaled) + (0.2 * coherence_scaled) + (0.2 * perplexity_scaled)

    return combined_score
```

The *combine_metric* function

KLD, which quantifies the dissimilarity between two probability distributions, is assigned the highest weight (0.6) because it effectively captures the distinctiveness of the topics, ensuring that the generated topics are non-overlapping and interpretable. Coherence, which measures the semantic consistency within each topic, is assigned a weight of 0.2, as it contributes to the overall interpretability and relevance of the discovered topics. Lastly, perplexity, a measure of how well the topic model generalizes to unseen data, is also assigned a weight of 0.2, since it indicates the model's predictive performance.

This function uses a scaling formula that maps the input values to the range [0,1]. For KLD and coherence, the formula divides the value by itself plus 1, so that higher values get mapped to values closer to 1. For perplexity, the formula inverts the scale by dividing the value by itself plus 1 and subtracting it from 1.

Finally, the code appends the current number of topics and combined score to the *num_topics_list* and *combine_list* respectively to generate the following line chart.



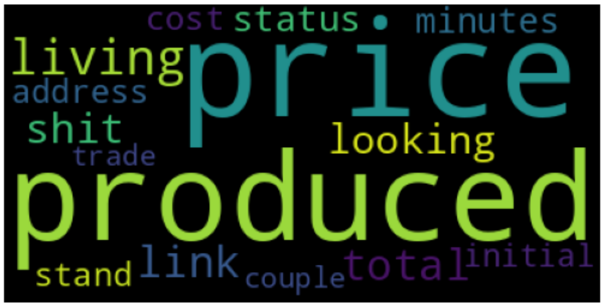
The combined score graph for Single Tweets

It is essential to acknowledge that these models depend on stochastic processes, resulting in diverse outcomes. Upon examination, it is evident that the cumulative score experiences no discernible enhancement when the number of topics exceeds 15. Consequently, to preserve the model's parsimony and minimize overfitting potential, it has been ascertained that the optimal number of topics should be set at 15.

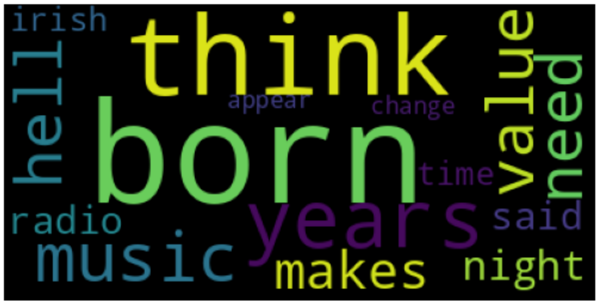
2.8 Word Clouds

Word Cloud is a visualization technique that displays the most frequent words or terms in a corpus of text. The words are arranged in a way that their size represents their frequency. We obtain the most prominent words from the inferred topics and generate word clouds using the word cloud Python package. A slight conversion is needed to transform the Gensim topic information output into a dictionary format that the word cloud library can accept. The word clouds for each of the fifteen topics are given below:

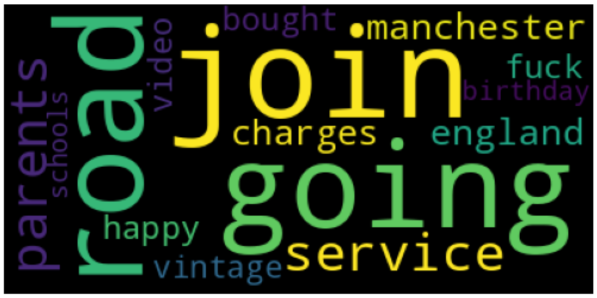
Topic #1



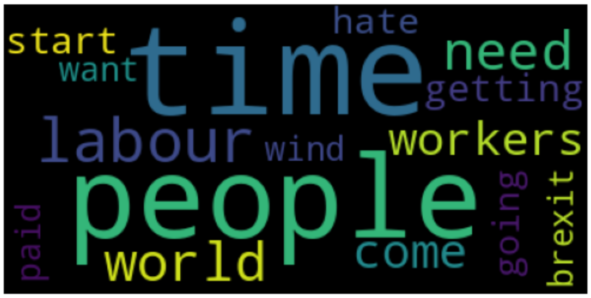
Topic #2



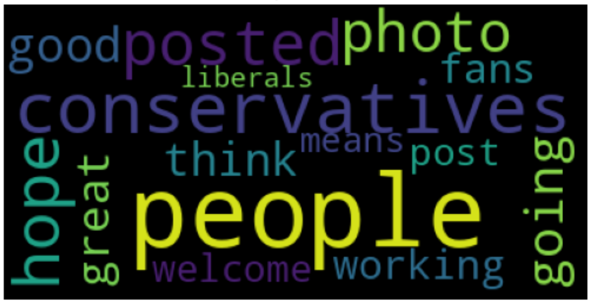
Topic #3



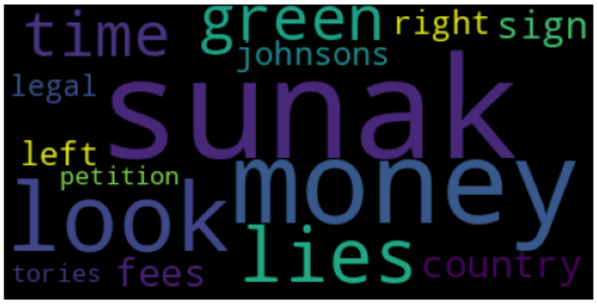
Topic #4



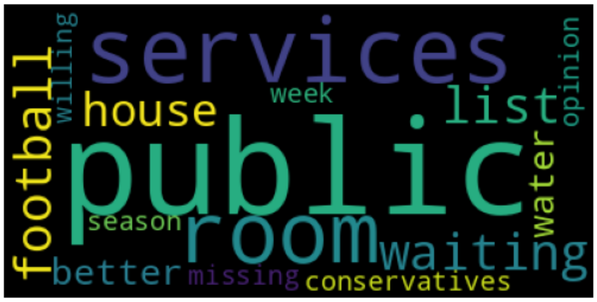
Topic #5



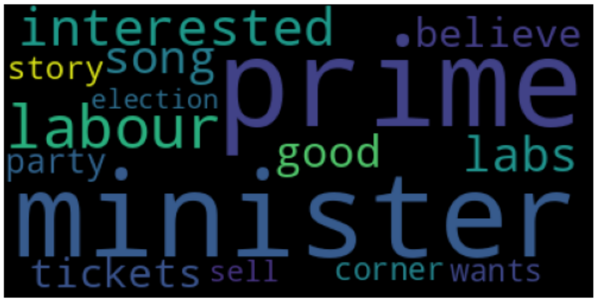
Topic #6

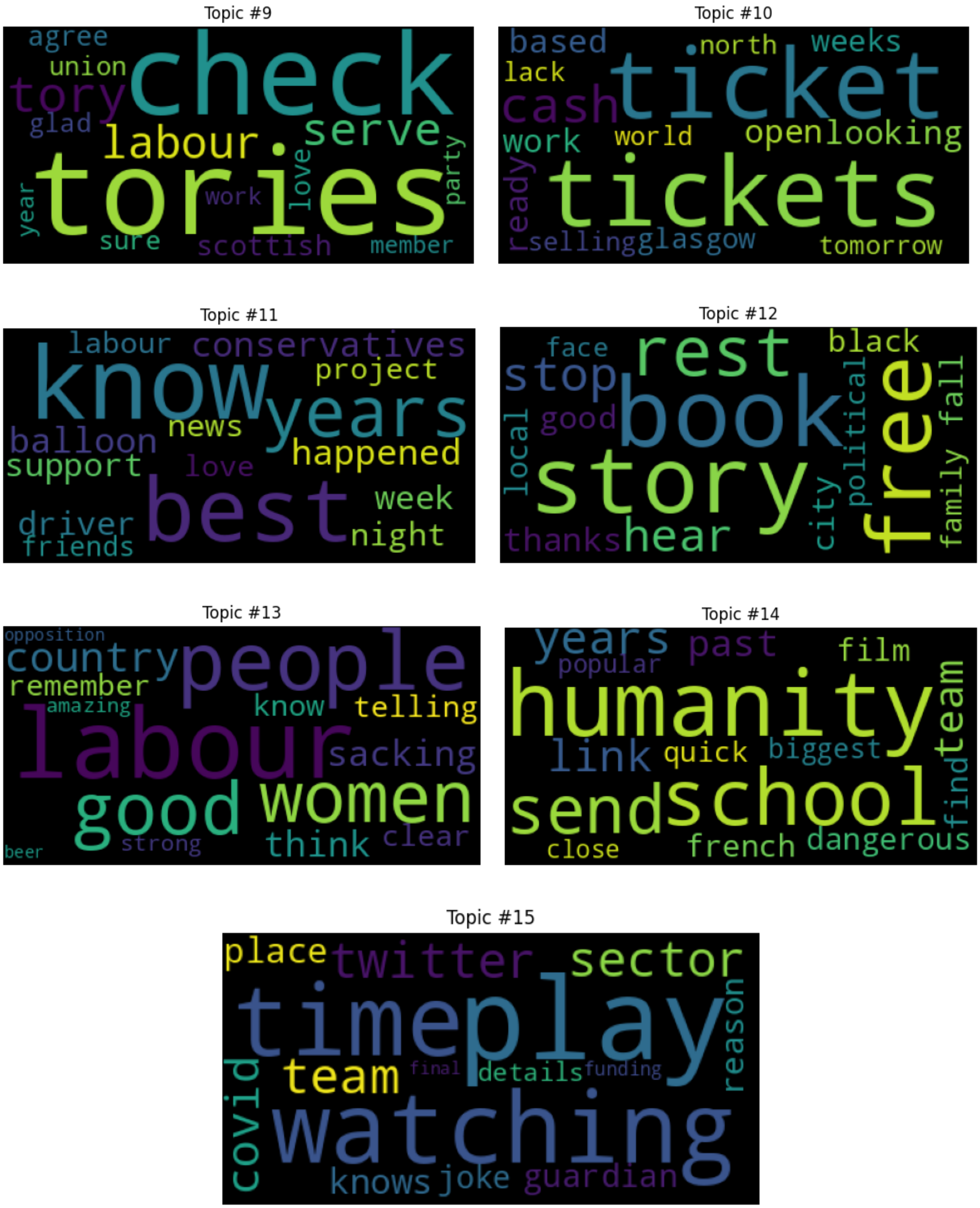


Topic #7



Topic #8





Topic #9

Topic #10

Topic #11

Topic #12

Topic #13

Topic #14

Topic #15

Word Clouds for single-tweet topics

Topic #	Words (ordered by frequency)
1	price, produced, living, link, total, shit, looking, status, minutes, stand, address, cost, initial, couple, trade
2	born, think, years, music, value, hell, need, makes, night, said, radio, irish, time, appear, change
3	join, going, road, service, parents, england, manchester, charges, fuck, bought, happy, video, vintage, birthday, schools
4	time, people, labour, world, need, come, workers, getting, start, going, hate, wind, brexit, want, paid
5	people, conservatives, hope, posted, photo, going, good, great, think, working, fans, welcome, post, means, liberals
6	sunak, money, look, lies, green, time, country, sign, fees, johnsons, right, left, legal, petition, tories
7	public, services, room, waiting, football, list, house, better, water, missing, opinion, willing, conservatives, season, week
8	prime, minister, labour, interested, labs, song, good, tickets, believe, party, story, wants, corner, sell, election
9	tories, check, tory, serve, labour, scottish, agree, love, sure, union, glad, work, year, member, party
10	ticket, tickets, cash, looking, based, work, ready, open, weeks, glasgow, world, lack, tomorrow, selling, north
11	know, best, years, conservatives, balloon, happened, support, week, driver, night, news, project, love, labour, friends

12	book, story, free, rest, stop, hear, thanks, good, city, black, fall, local, political, family, face
13	labour, people, good, women, country, sacking, think, telling, clear, remember, know, strong, amazing, opposition, beer
14	humanity, school, send, years, link, team, past, dangerous, find, film, french, quick, biggest, popular, close
15	play, watching, time, twitter, sector, team, covid, place, knows, guardian, reason, joke, details, funding, final

List of top words for each topic, ordered by frequency

2.9 Textual Interpretation:

The list of words is sorted in descending order based on their frequency, indicating that the words appearing at the top of the list have a higher occurrence rate and are considered more significant. Here are the topics that, in my subjective opinion, are being discussed.

Topic 1:

The challenges of living in a society where the cost of living is high, affordable housing is scarce, and conscious choices must be made to avoid the production of harmful products. *(Economics)*

Topic 2:

The impact of music and radio over the years, how it makes us think and feel, and the changes that have appeared in the industry, with a particular focus on Irish artists. *(Music)*

Topic 3:

About the experience of traveling to new places, the joys, and challenges of road travel, as well as the discovery of vintage or interesting locations along the way, with discussions of cost, convenience, safety, personal growth, and memories that such journeys can bring. *(Travel)*

Topic 4:

The challenges of the labour party and fair compensation for workers, particularly in light of Brexit, with a focus on the need for people to come together and start advocating for change while navigating the hurdles of hate and economic forces like wind power. *(Politics)*

Topic 5:

About a social media post, possibly a photo, that sparked debate and discussion among people with different political affiliations, with conservatives expressing hope while liberals weighed in with their thoughts, and fans and supporters of the post welcomed the good news. *(Politics)*

Topic 6:

About a petition calling for legal action against the Johnson government and Sunak over alleged lies and unethical conduct related to money and green policies, with both the right and left weighing in on the issue and discussing the impact on the country's future, including fees and taxes, as well as the role of the Tories in the situation. *(Politics)*

Topic 7:

About the state of public services, such as waiting rooms and water supply, with comparisons made to the ups and downs of a football season, as well as the opinions and willingness of the conservatives to address these issues. *(Community)*

Topic 8:

About the upcoming election and the tactics being used by the different parties, with a focus on the prime minister and their story, as well as the Labour party's interests and beliefs, the possibility of selling tickets or using songs to rally support. *(Politics)*

Topic 9:

About the relationship between the Scottish Labour party and the Tories, some members express agreement and appreciation for the work being done, while others remain skeptical and want to check that the union is being served, but all express love and gladness to be a part of the party for another year. *(Politics)*

Topic 10:

About the selling of tickets for an event, possibly based in Glasgow, with some people looking for cash sales while others are ready to sell online, and the lack of interest or preparedness from some buyers or sellers, and some upcoming events. *(Entertainment)*

Topic 11:

About recent political news and events, with a focus on the support and love for Labour or the Conservatives among friends and acquaintances, the happenings of a project or initiative over several years. *(Politics)*

Topic 12:

About a book or story, possibly with political or local themes, that is being freely distributed and discussed throughout the city, with people taking time to stop and hear each other's opinions and thanks being expressed for the good work being done. *(Community)*

Topic 13:

Regarding the state of the Labour party and the opposition, some people think that sacking certain individuals or being more clear about their stance could be good for the country and women in particular, while others remember strong and amazing individuals who have played a role in the party's history. *(Politics)*

Topic 14:

About the dangerous and popular French school of filmmaking, with a focus on the link between past and present teams, as well as the biggest and most influential films from the past few years. *(Films)*

Topic 15:

About a final or important play, possibly in the context of COVID-19 and the effect it has had on the sector, with people watching and discussing the performance on Twitter and elsewhere. *(Covid-19)*

2.10 Are the topics good or bad?

Yes, the suggested topics are suitable as they cover a diverse range of relevant subjects. It's not surprising that politics in the United Kingdom is the most frequently discussed topic given the time period during which the tweets were collected. Besides politics, the topics cover various areas such as economics, music, movies, travel, and entertainment, which could lead to significant discussions.

3. Grouped Tweets

3.1 Overview

For the provided dataset of grouped tweets, single-pass clustering was employed to aggregate tweets exhibiting 'similarity' based on a distance metric such as cosine similarity. The data was supplied in a file named groupedTweets.csv, where each row represents a tweet with the following attributes:

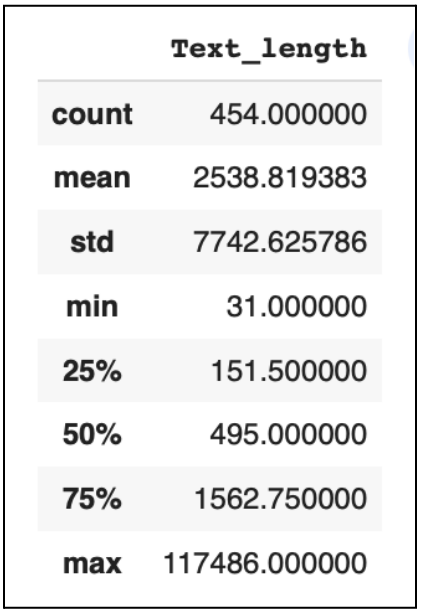
- *group*: The assigned cluster or group from the single-pass clustering process.
- *tweetID*: A unique identifier for the tweet that enables referencing it on the Twitter platform.
- *username*: The username of the individual who posted the tweet.
- *text*: The actual content of the tweet.
- *qScore*: A subjective assessment of the tweet's quality.
- nScore: An additional subjective metric evaluating the tweet's newsworthiness, derived from the user's profile and the content of the tweet.

3.2 Statistics

The statistics for *qScore* and *nScore* are the same as for the single tweets dataset. This data set had 9968 tweets.

As with Single Tweets, I used *nScore* to drop items from the data frame. I used the value of 0.0 as the threshold. I dropped items below this threshold considering that their newsworthiness is negative. This step will ensure that a good chunk of spam is discarded from our dataset.

Now, all tweet text in the same group number was concatenated together by group number. Next, following the same process as for the Single Tweets, a new Text_length column was made.



	Text_length
count	454.000000
mean	2538.819383
std	7742.625786
min	31.000000
25%	151.500000
50%	495.000000
75%	1562.750000
max	117486.000000

Statistics of tweet texts for grouped tweets

When grouped by the group number assigned, there are 454 distinct groups. The *Text_Length* column is the collection of the tweet length statistics. For this attribute,

- The **mean** cluster text was 2438.82 characters long
- The standard deviation, **std**, for text lengths of clusters was very high: 7742.63.
- The smallest tweet length, **min**, was 31 characters long
- The **median** group text length is 495 characters long
- The maximum text length, **max**, is approximately 117486 characters long.

3.3 Pre-processing of grouped tweet texts

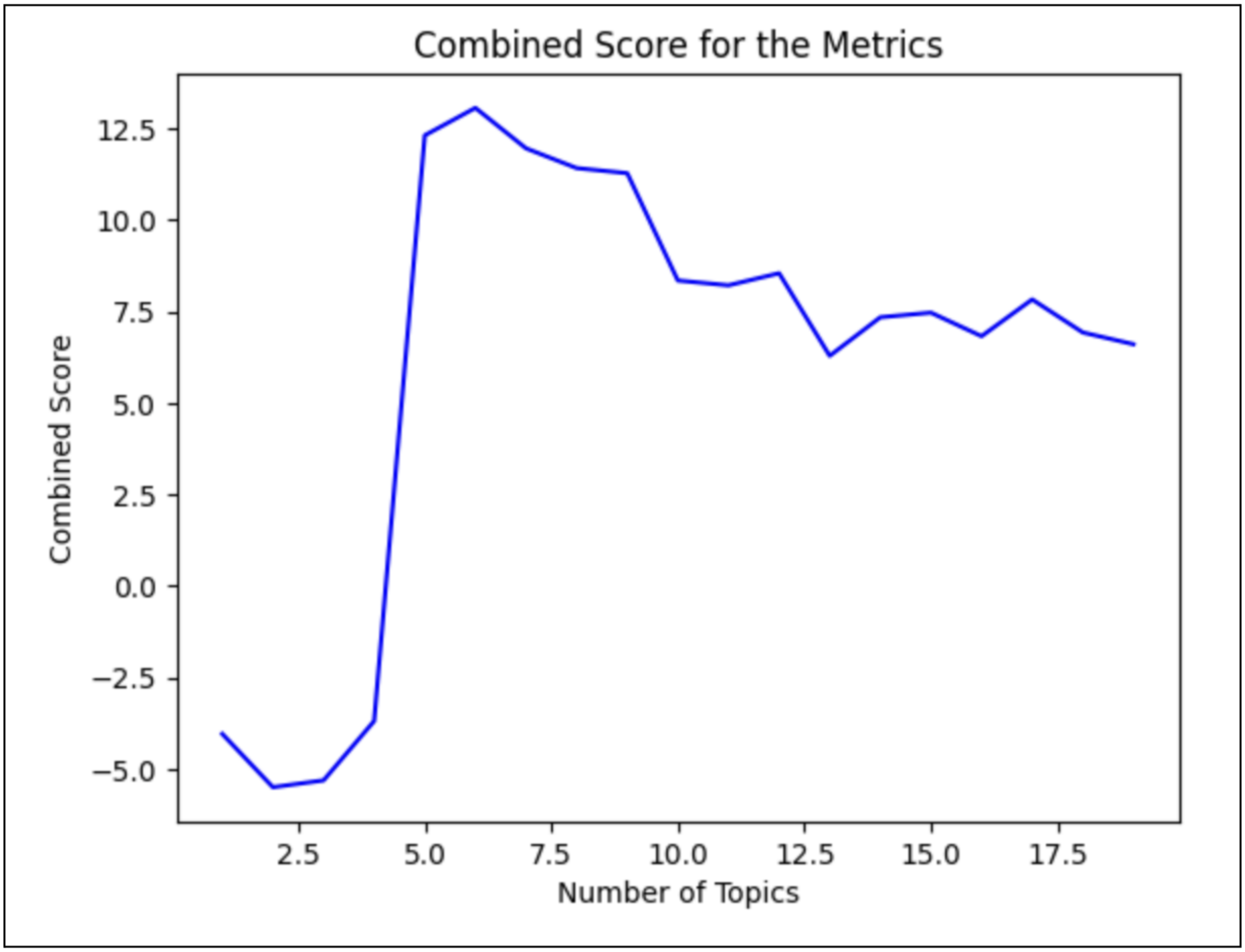
Most of the pre-processing remains the same for this clustered tweets approach. The only additional step is that all tweets in a group have their texts appended together.

3.4 Metrics studied

The metrics studied for this section are identical to the previous section.

3.5 How the number of topics was chosen

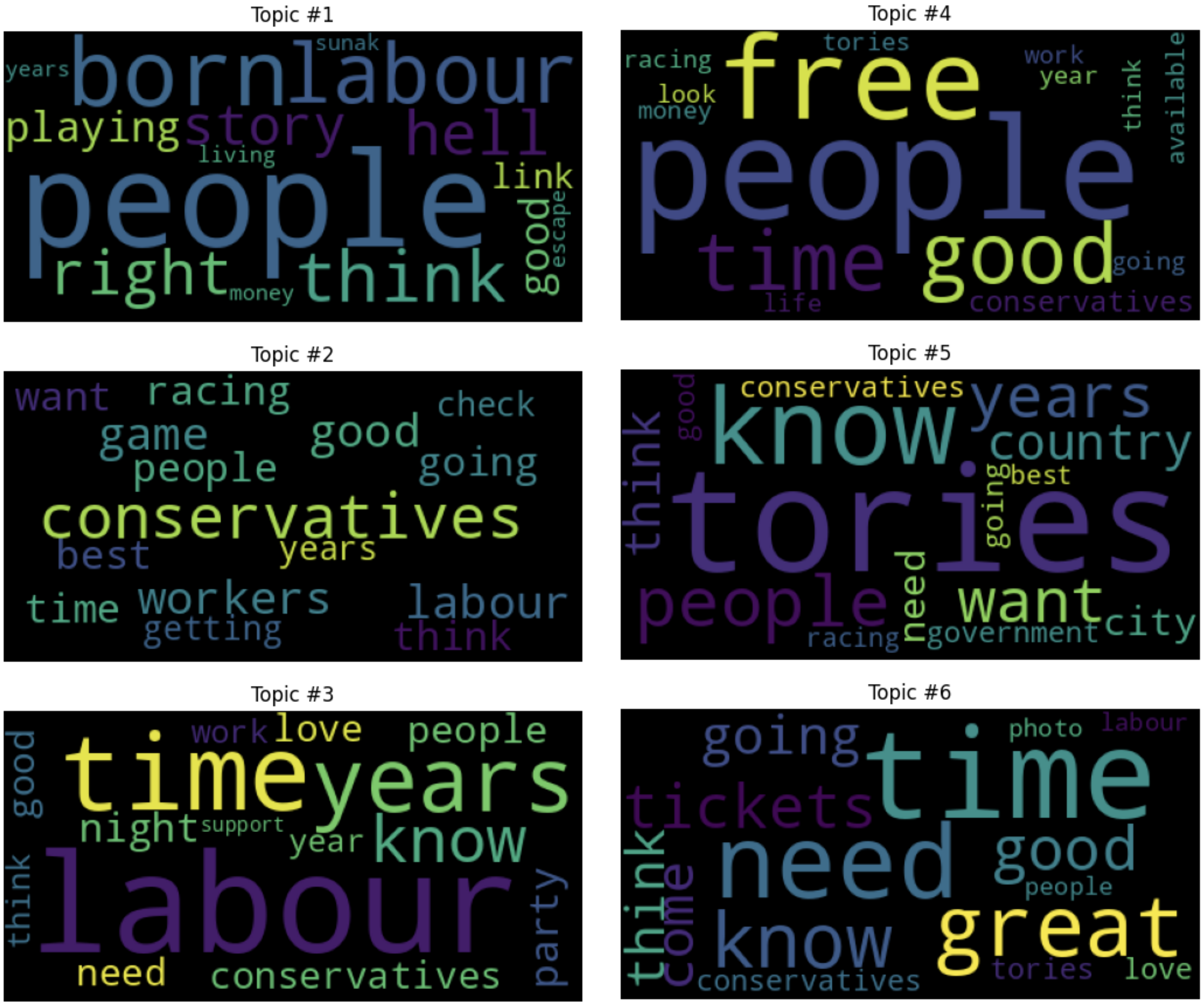
The methodology studied for this section is identical to the previous section.



The combined score graph for Single Tweets

According to the graph, the highest point of the combined score occurs when the number of topics is between 6 and 7. Therefore, I have determined that the optimal number of topics is 6.

3.6 Word Clouds



Word Clouds for grouped tweets

Topic #	Words (ordered by frequency)
1	people, born, labour, think, right, hell, story, playing, good, link, living, money, sunak, escape, years
2	conservatives, good, labour, game, workers, racing, people, going, think, time, best, want, check, years, getting
3	labour, time, years, know, night, party, people, conservatives, love, good, need, think, year, work, support
4	people, free, good, time, conservatives, think, work, year, going, racing, life, look, money, tories, available
5	tories, know, people, want, years, country, think, city, need, government, good, going, conservatives, racing, best
6	time, need, great, know, tickets, good, going, think, come, conservatives, love, tories, photo, labour, people

List of top words for each topic, ordered by frequency

3.7 Textual Interpretation:

Topic 1:

About the intersection of politics and personal finance, with a focus on the experiences and stories of people who were born and raised during Labour rule, and discussions of the link between money and happiness and the role of figures such as Sunak in shaping economic policies over the years. *(Politics)*

Topic 2:

About the state of politics and the economy, with a focus on the workers and ordinary people, as well as the rivalry between the Conservatives and Labour parties and the game of politics, checking facts and statistics, and discussing the best ways of getting ahead. *(Politics)*

Topic 3:

About the evolution and challenges of the Labour party, reflection on personal experiences and support for the party, and discussions of hard work and support that will be needed to move forward and achieve good outcomes during these challenging times. *(Politics)*

Topic 4:

About the intersection of money, work, and leisure, with a focus on the availability of free time and the best ways of using it, including activities such as horse-racing, while also considering the political context and the views of Conservatives and Tory groups. *(Politics, Life, Sports)*

Topic 5:

About the state of politics and society, feelings towards the government, the Tories, and the Conservatives, as well as discourse about horse-racing. *(Politics, Sports)*

Topic 6:

About the upcoming political events and the perspectives and needs of the people, including the Conservatives and the Labour party, with people expressing their love and support for the different sides, while also considering the need to unify and think strategically to achieve great outcomes. *(Politics)*

3.8 Are the topics good or bad?

Indeed, the topics are commendable since they encompass a wide array of discussions about diverse subject matters, albeit all of them can be categorized within the broader sphere of UK politics and sports, especially horse-racing.

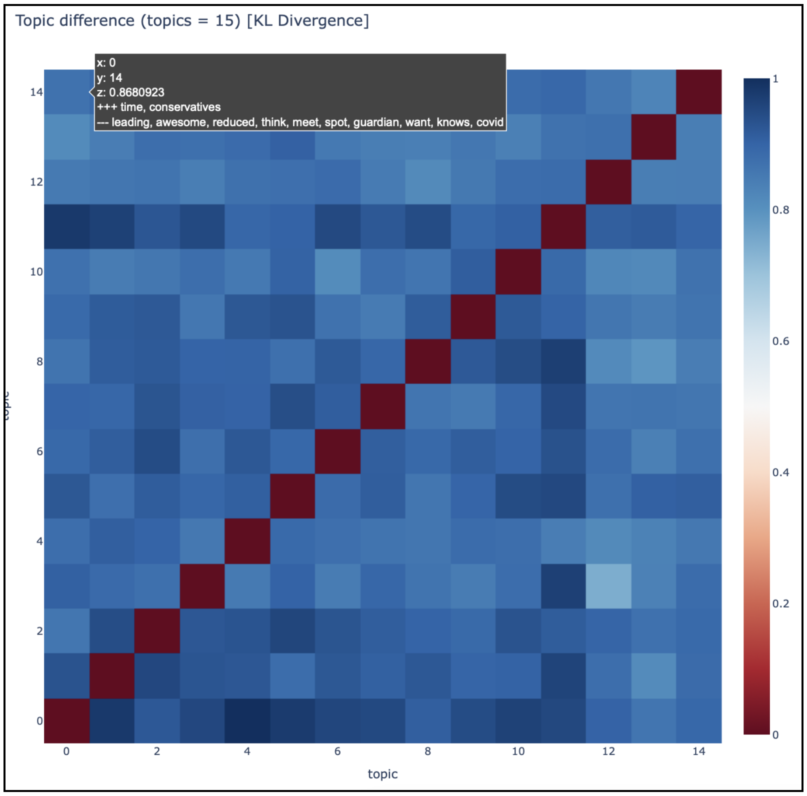
4. Discussion

4.1 Overall Comparison

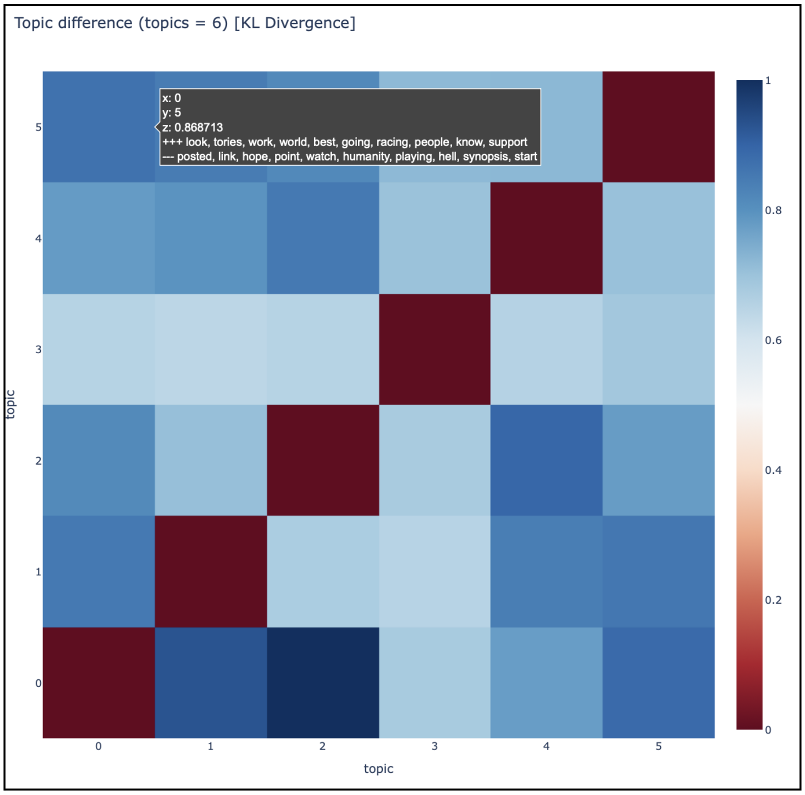
In this study, we compare the quality of topics obtained from two different approaches discussed in Chapters 2 and 3. To compare the two approaches, KL Divergence (KLD) matrices were used, which are presented in the figures below.

KLD is a widely recognized metric for measuring the similarity between two probability distributions, which makes it a suitable choice for evaluating the quality of topic models. The analysis shows that KLD values for individual tweets were superior to those for clustered tweets. The average KL divergences were calculated to be 0.83 and 0.64 for individual and clustered tweets, respectively, indicating that the single-tweet approach outperforms the clustered-tweet approach.

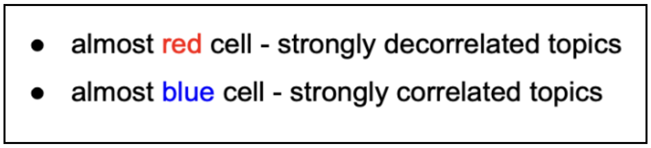
The single-pass clustering approach yielded inferior results compared to the regular approach, primarily because it grouped together tweets that were not compatible with each other. This led to competing themes within the same document, resulting in a loss of information and a less accurate model overall.



Single Tweets Topic Modeling



Grouped Tweets Topic Modeling



- almost red cell - strongly decorrelated topics
- almost blue cell - strongly correlated topics

- +++ make, world, well - words from the intersection of topics = present in both topics;
- --- money, day, still - words from the symmetric difference of topics = present in one topic but not the other.

It is worth noting that coherence values were found to be dependent on the sizes of the documents, which emphasizes the importance of selecting an appropriate approach for topic modeling on Twitter data.

4.2 In-Depth Comparison with Examples

The tweets analyzed in this study were collected in January 2023 and were primarily confined to the UK. The experiments revealed that the most prevalent topic among Twitter users was politics (**@BorisJohnson I wasn’t in anyway a Boris Johnson fan**), with several topics dedicated to political parties (**@CathySt35873400 Because Pierre didn't cut them a cheque for 585 million. Oh, and they hate conservatives**) and current events, like the economy (**@tomhfh Look at the result of her policies on the UK economy**) and human rights (**We're all still fighting for these rights, and transgender people will be fighting with you**).

There were also a few topics on sports (**Hey got 3x for Manchester United VS Crystal Palace…. Anyone interested**), music, films, and entertainment (**@ameuf_chat Are you watching Derry Girls?**) but these topics received less attention due to their ordinary nature, while the more nuanced and unique topics were overshadowed by the vast number of tweets on mainstream subjects. In some cases, tweets like these that were considered ‘spam’ content were made to be filtered out to improve the quality of data.

While the topic modeling techniques were successful in capturing broad themes, additional filtering and curation efforts are required for finer topic detection. Nevertheless, we believe that the quality of topics identified in Sections 2.9 and 3.7 is adequate for initial estimates and provides useful insights for further analysis.

5. Further Discussion

5.1 Issues Encountered

The nature of tweets posed several challenges for the project of topic modeling. These tweets made it difficult for algorithms to effectively identify and extract significant subjects from them. Some of these issues are

1. **Limited Character Count:**

   Twitter restricts tweets to a maximum of 280 characters. This limit can be a hindrance to forming coherent and comprehensive topics since users may not be able to convey their ideas fully. Consequently, tweets may be fragmented and not provide enough context for forming complete topics.

   For example, a tweet such as "The new healthcare policy is a disaster #healthcare #policy #disaster" may not provide enough information to form a comprehensive topic on healthcare policy.

2. **Use of Slang and Abbreviations:**

   Twitter users often use slang and abbreviations to save on character count or express their opinions succinctly. However, this can create difficulties in topic formation as not all users may understand the terms used. The use of such language may result in a lack of clarity and context.

   For instance, a tweet such as "IDK why POTUS said that SMH #politics #POTUS #SMH" may be challenging to form a topic from since it includes slang and abbreviations that not all readers may comprehend.

3. **Lack of Context:**

   Twitter users often assume that their followers understand the context of their tweets. However, this may not always be the case, leading to difficulties in forming topics. A tweet may be ambiguous, and it may be challenging to deduce its intended meaning without additional context.

   For example, a tweet such as "I can't believe he did that #outrageous #shocked" may not provide enough information to form a complete topic, as it lacks context.

5.2 Future Work

To overcome these obstacles, we may implement the following tactics to enhance the quality of our subject modeling results:

1. **Refined Preprocessing:** We could consider using named entity recognition techniques to extract relevant entities and improve topic modeling accuracy. Another important step is to use domain-specific

knowledge and lexicons to identify topic-specific terms and eliminate noise further.

2. **Extend Textual Data:** To augment our short text data, we may use additional textual data sources, such as user profiles, user interactions, hashtags, and metadata linked with tweets, which may give extra context and information to enhance our topic modeling outcomes.

3. **Consider Context:** Context is crucial in brief text analysis. Using word embeddings, a sort of deep learning method that may assist capture the semantic meaning of words and phrases depending on their context, we can include context into our research.

4. **Ensemble Approaches:** Several models can be combined to overcome the sparse and noisy character of a brief text. For instance, we may develop a more accurate topic model by combining topic modeling methods such as Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF).

Even though the brief text might provide obstacles for topic modeling, by utilizing the aforementioned tactics, we can increase the accuracy of our results and gain relevant insights from our Twitter data.

~ End ~