Navigating subreddit discussions: A BERTopic-driven chatbot approach

Sai Surya Varshith Nukala
U20744319,
Email: saisurya@bu.edu

Vaishnavi Vadlamudi
U19250166,
Email: vaishv@bu.edu

Abstract - This project embarked on an exploratory journey into subreddit data, specifically focusing on the Boston subreddit, to understand the dynamics and structure of conversations within these communities. Recognizing the often-chaotic nature of comments and discussions, we aimed to bring clarity and accessibility to this wealth of information. Employing BERTopic, a robust topic modeling tool, we extracted meaningful topics from subreddit posts. Our choice was influenced by computational constraints which made the more resource-intensive LLAMA 2 model less feasible. The preprocessing of our dataset involved converting .zst files to a more manageable JSON format, ensuring a smooth integration with our chosen models. Our analysis yielded insightful results, revealing dominant topics within the subreddit's posts and comments. We visualized these findings through multiple graphs, providing a clear representation of the subreddit's thematic landscape. Building on this, we developed a ChatBot using RASA, trained to assist users in navigating the myriad of subreddit posts. This ChatBot is designed to direct users to existing discussions relevant to their queries, thereby enhancing the efficiency of information retrieval on the platform. In conclusion, our project not only offers a pipeline for subreddit data analysis but also introduces an innovative ChatBot tool to facilitate user interaction with Reddit's extensive content. Our future endeavors will focus on refining this ChatBot, incorporating sentiment analysis to match user queries with similar sentiments in subreddit posts. This advancement aims to create a more intuitive and user-centric experience, capitalizing on Reddit's potential as a valuable resource for real-world problem-solving and market insights.

Github Link

Data Files

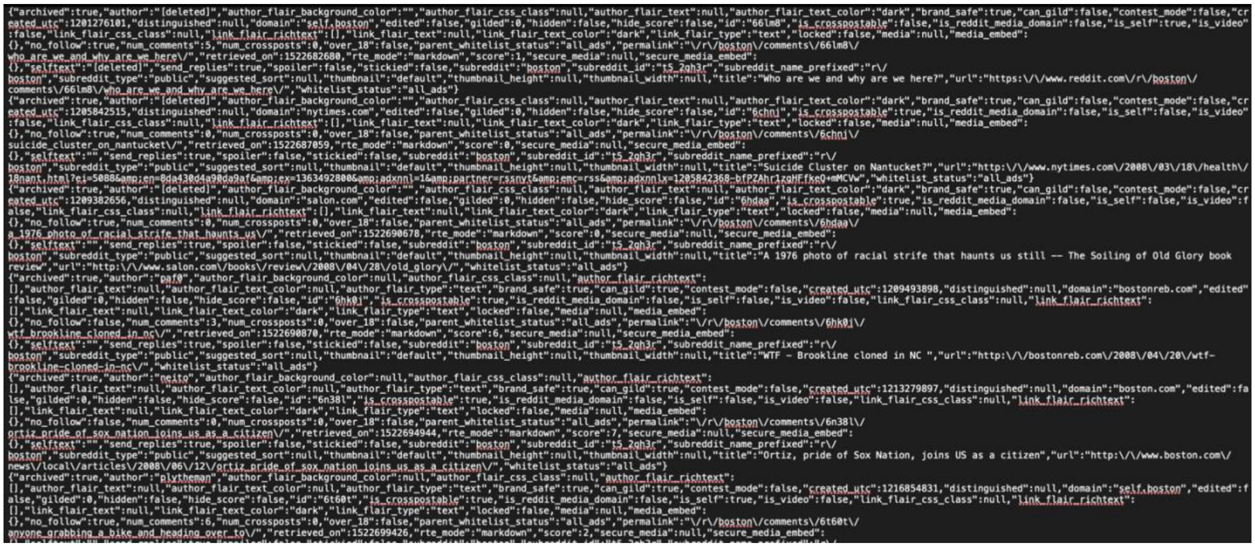
Introduction:

In the digital age, online forums and discussion platforms like Reddit have become a treasure trove of user-generated content, opinions, and dialogues. Among these, subreddits, with their community-specific focus, offer a unique opportunity to analyze and understand diverse topics and discussions. This project zeroes in on the Boston subreddit, aiming to unravel the intricate tapestry of conversations and themes through advanced Natural Language Processing (NLP) techniques.

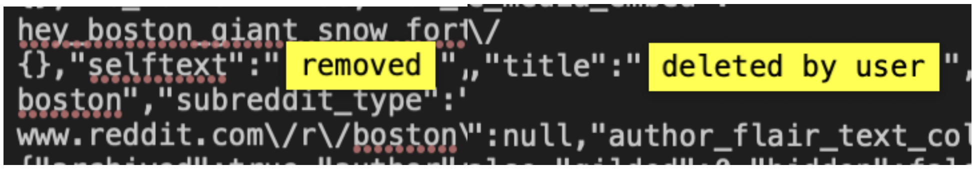
The motivation behind this endeavor stems from the inherent complexity and unstructured nature of subreddit data. Reddit discussions are dynamic and multifaceted, often challenging to navigate due to their volume and variability. Our project sought to bring order to this chaos by employing topic modeling and developing a ChatBot to aid users in finding relevant discussions. BERTopic emerged as our tool of choice, given its efficiency and our computational constraints.

Challenges in Data Handling and Analysis

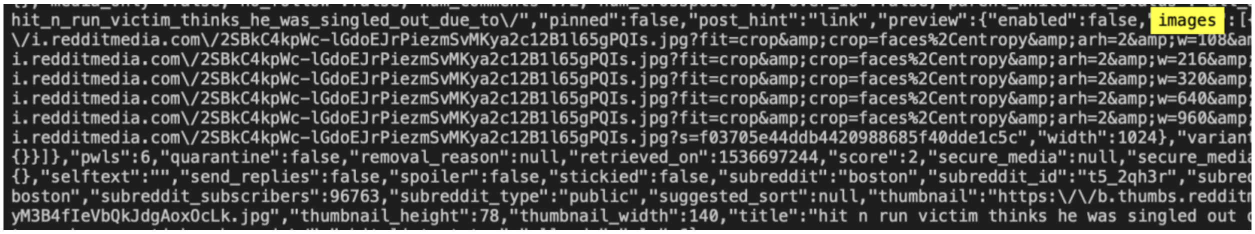
The journey of transforming raw subreddit data into insightful analysis was not without challenges. Our primary hurdle lay in data cleaning and structuring. The subreddit data, initially in .zst files, presented a mix of textual and non-textual content, including posts with NaN values, removed or deleted comments, and media-only entries. These were irrelevant for our text-centric NLP project and thus needed exclusion. The image below illustrates the kind of non-textual content that was filtered out from our dataset:



```
{"archived":true,"author":"[deleted]","author_flair_background_color":"","author_flair_css_class":null,"author_flair_text":null,"author_flair_text_color":"dark","brand_safe":true,"can_gild":false,"contest_mode":false,"cr
eated_utc":1201276101,"distinguished":null,"domain":"self.boston","edited":false,"gilded":0,"hidden":false,"hide_score":false,"id":"66lm8","is_crosspostable":false,"is_reddit_media_domain":false,"is_self":true,"is_video"
:false,"link_flair_css_class":null,"link_flair_richtext":[],"link_flair_text":null,"link_flair_text_color":"dark","link_flair_type":"text","locked":false,"media":null,"media_embed":
{},"no_follow":true,"num_comments":5,"num_crossposts":0,"over_18":false,"parent_whitelist_status":"all_ads","permalink":"\/r\/boston\/comments\/66lm8\/
who_are_we_and_why_are_we_here\/","retrieved_on":1522682680,"rte_mode":"markdown","score":1,"secure_media":null,"secure_media_embed":
{},"selftext":"[deleted]","send_replies":true,"spoiler":false,"stickied":false,"subreddit":"boston","subreddit_id":"t5_2qh3r","subreddit_name_prefixed":"r\/
boston","subreddit_type":"public","suggested_sort":null,"thumbnail":"default","thumbnail_height":null,"thumbnail_width":null,"title":"Who are we and why are we here?","url":"https:\/\/www.reddit.com\/r\/boston\/
comments\/66lm8\/who_are_we_and_why_are_we_here\/","whitelist_status":"all_ads"}
{"archived":true,"author":"[deleted]","author_flair_background_color":"","author_flair_css_class":null,"author_flair_text":null,"author_flair_text_color":"dark","brand_safe":true,"can_gild":false,"contest_mode":false,"cr
eated_utc":1205842515,"distinguished":null,"domain":"nytimes.com","edited":false,"gilded":0,"hidden":false,"hide_score":false,"id":"6chnj","is_crosspostable":true,"is_reddit_media_domain":false,"is_self":false,"is_video"
:false,"link_flair_css_class":null,"link_flair_richtext":[],"link_flair_text":null,"link_flair_text_color":"dark","link_flair_type":"text","locked":false,"media":null,"media_embed":
{},"no_follow":true,"num_comments":0,"num_crossposts":0,"over_18":false,"parent_whitelist_status":"all_ads","permalink":"\/r\/boston\/comments\/6chnj\/
suicide_cluster_on_nantucket\/","retrieved_on":1522687059,"rte_mode":"markdown","score":0,"secure_media":null,"secure_media_embed":
{},"selftext":"","send_replies":true,"spoiler":false,"stickied":false,"subreddit":"boston","subreddit_id":"t5_2qh3r","subreddit_name_prefixed":"r\/
boston","subreddit_type":"public","suggested_sort":null,"thumbnail":"default","thumbnail_height":null,"thumbnail_width":null,"title":"Suicide Cluster on Nantucket?","url":"http:\/\/www.nytimes.com\/2008\/03\/18\/health\/
18nant.html?ei=5088&amp;en=8da430d4a90da9af&amp;ex=1363492800&amp;adxnnl=1&amp;partner=rssnyt&amp;emc=rss&amp;adxnnlx=1205842368-bfPZAhr1zgHFfkeQ+mMCVw","whitelist_status":"all_ads"}
{"archived":true,"author":"[deleted]","author_flair_background_color":"","author_flair_css_class":null,"author_flair_text":null,"author_flair_text_color":"dark","brand_safe":true,"can_gild":false,"contest_mode":false,"cr
eated_utc":1209382656,"distinguished":null,"domain":"salon.com","edited":false,"gilded":0,"hidden":false,"hide_score":false,"id":"6hdaa","is_crosspostable":true,"is_reddit_media_domain":false,"is_self":false,"is_video":f
alse,"link_flair_css_class":null,"link_flair_richtext":[],"link_flair_text":null,"link_flair_text_color":"dark","link_flair_type":"text","locked":false,"media":null,"media_embed":
{},"no_follow":true,"num_comments":0,"num_crossposts":0,"over_18":false,"parent_whitelist_status":"all_ads","permalink":"\/r\/boston\/comments\/6hdaa\/
a_1976_photo_of_racial_strife_that_haunts_us\/","retrieved_on":1522690678,"rte_mode":"markdown","score":0,"secure_media":null,"secure_media_embed":
{},"selftext":"","send_replies":true,"spoiler":false,"stickied":false,"subreddit":"boston","subreddit_id":"t5_2qh3r","subreddit_name_prefixed":"r\/
boston","subreddit_type":"public","suggested_sort":null,"thumbnail":"default","thumbnail_height":null,"thumbnail_width":null,"title":"A 1976 photo of racial strife that haunts us still —— The Soiling of Old Glory book
review","url":"http:\/\/www.salon.com\/books\/review\/2008\/04\/28\/old_glory\/","whitelist_status":"all_ads"}
{"archived":true,"author":"paf0","author_flair_background_color":null,"author_flair_css_class":null,"author_flair_richtext":
[],"author_flair_text":null,"author_flair_text_color":null,"author_flair_type":"text","brand_safe":true,"can_gild":true,"contest_mode":false,"created_utc":1209493898,"distinguished":null,"domain":"bostonreb.com","edited"
:false,"gilded":0,"hidden":false,"hide_score":false,"id":"6hk0j","is_crosspostable":true,"is_reddit_media_domain":false,"is_self":false,"is_video":false,"link_flair_css_class":null,"link_flair_richtext":
[],"link_flair_text":null,"link_flair_text_color":"dark","link_flair_type":"text","locked":false,"media":null,"media_embed":
{},"no_follow":false,"num_comments":3,"num_crossposts":0,"over_18":false,"parent_whitelist_status":"all_ads","permalink":"\/r\/boston\/comments\/6hk0j\/
wtf_brookline_cloned_in_nc\/","retrieved_on":1522690870,"rte_mode":"markdown","score":6,"secure_media":null,"secure_media_embed":
{},"selftext":"","send_replies":true,"spoiler":false,"stickied":false,"subreddit":"boston","subreddit_id":"t5_2qh3r","subreddit_name_prefixed":"r\/
boston","subreddit_type":"public","suggested_sort":null,"thumbnail":"default","thumbnail_height":null,"thumbnail_width":null,"title":"WTF - Brookline cloned in NC ","url":"http:\/\/bostonreb.com\/2008\/04\/20\/wtf-
brookline-cloned-in-nc\/","whitelist_status":"all_ads"}
{"archived":true,"author":"neito","author_flair_background_color":null,"author_flair_css_class":null,"author_flair_richtext":
[],"author_flair_text":null,"author_flair_text_color":null,"author_flair_type":"text","brand_safe":true,"can_gild":true,"contest_mode":false,"created_utc":1213279897,"distinguished":null,"domain":"boston.com","edited":fa
lse,"gilded":0,"hidden":false,"hide_score":false,"id":"6n38l","is_crosspostable":true,"is_reddit_media_domain":false,"is_self":false,"is_video":false,"link_flair_css_class":null,"link_flair_richtext":
[],"link_flair_text":null,"link_flair_text_color":"dark","link_flair_type":"text","locked":false,"media":null,"media_embed":
{},"no_follow":false,"num_comments":0,"num_crossposts":0,"over_18":false,"parent_whitelist_status":"all_ads","permalink":"\/r\/boston\/comments\/6n38l\/
ortiz_pride_of_sox_nation_joins_us_as_a_citizen\/","retrieved_on":1522694944,"rte_mode":"markdown","score":7,"secure_media":null,"secure_media_embed":
{},"selftext":"","send_replies":true,"spoiler":false,"stickied":false,"subreddit":"boston","subreddit_id":"t5_2qh3r","subreddit_name_prefixed":"r\/
boston","subreddit_type":"public","suggested_sort":null,"thumbnail":"default","thumbnail_height":null,"thumbnail_width":null,"title":"Ortiz, pride of Sox Nation, joins US as a citizen","url":"http:\/\/www.boston.com\/
news\/local\/articles\/2008\/06\/12\/ortiz_pride_of_sox_nation_joins_us_as_a_citizen\/","whitelist_status":"all_ads"}
{"archived":true,"author":"plytheman","author_flair_background_color":null,"author_flair_css_class":null,"author_flair_richtext":
[],"author_flair_text":null,"author_flair_text_color":null,"author_flair_type":"text","brand_safe":true,"can_gild":true,"contest_mode":false,"created_utc":1216854831,"distinguished":null,"domain":"self.boston","edited":f
alse,"gilded":0,"hidden":false,"hide_score":false,"id":"6t60t","is_crosspostable":true,"is_reddit_media_domain":false,"is_self":true,"is_video":false,"link_flair_css_class":null,"link_flair_richtext":
[],"link_flair_text":null,"link_flair_text_color":"dark","link_flair_type":"text","locked":false,"media":null,"media_embed":
{},"no_follow":true,"num_comments":6,"num_crossposts":0,"over_18":false,"parent_whitelist_status":"all_ads","permalink":"\/r\/boston\/comments\/6t60t\/
anyone_grabbing_a_bike_and_heading_over_to\/","retrieved_on":1522699426,"rte_mode":"markdown","score":2,"secure_media":null,"secure_media_embed":
```

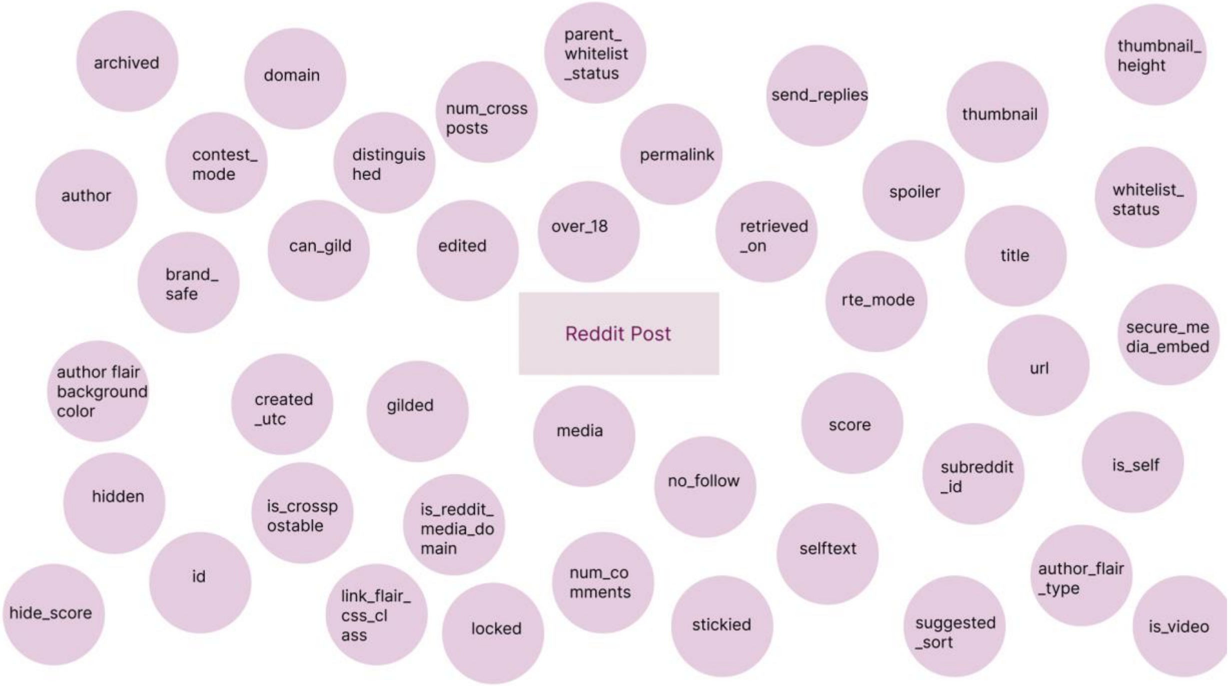


```
hey_boston_giant_snow_fort\/
{},"selftext":" removed ","title":" deleted by user ",
boston","subreddit_type":'
www.reddit.com\/r\/boston\":null,"author_flair_text_co
```



```
hit_n_run_victim_thinks_he_was_singled_out_due_to\/","pinned":false,"post_hint":"link","preview":{"enabled":false, images :[
\/i.redditmedia.com\/2SBkC4kpWc-lGdoEJrPiezmSvMKya2c12B1l65gPQIs.jpg?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=108&a
i.redditmedia.com\/2SBkC4kpWc-lGdoEJrPiezmSvMKya2c12B1l65gPQIs.jpg?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=216&amp
i.redditmedia.com\/2SBkC4kpWc-lGdoEJrPiezmSvMKya2c12B1l65gPQIs.jpg?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=320&amp
i.redditmedia.com\/2SBkC4kpWc-lGdoEJrPiezmSvMKya2c12B1l65gPQIs.jpg?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=640&amp
i.redditmedia.com\/2SBkC4kpWc-lGdoEJrPiezmSvMKya2c12B1l65gPQIs.jpg?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=960&amp
i.redditmedia.com\/2SBkC4kpWc-lGdoEJrPiezmSvMKya2c12B1l65gPQIs.jpg?s=f03705e44ddb4420988685f40dde1c5c","width":1024},"varian
{}}]},"pwls":6,"quarantine":false,"removal_reason":null,"retrieved_on":1536697244,"score":2,"secure_media":null,"secure_medi
{},"selftext":"","send_replies":false,"spoiler":false,"stickied":false,"subreddit":"boston","subreddit_id":"t5_2qh3r","subre
boston","subreddit_subscribers":96763,"subreddit_type":"public","suggested_sort":null,"thumbnail":"https:\/\/b.thumbs.reddit
yM3B4fIeVbQkJdgAoxOcLk.jpg","thumbnail_height":78,"thumbnail_width":140,"title":"hit n run victim thinks he was singled out
```

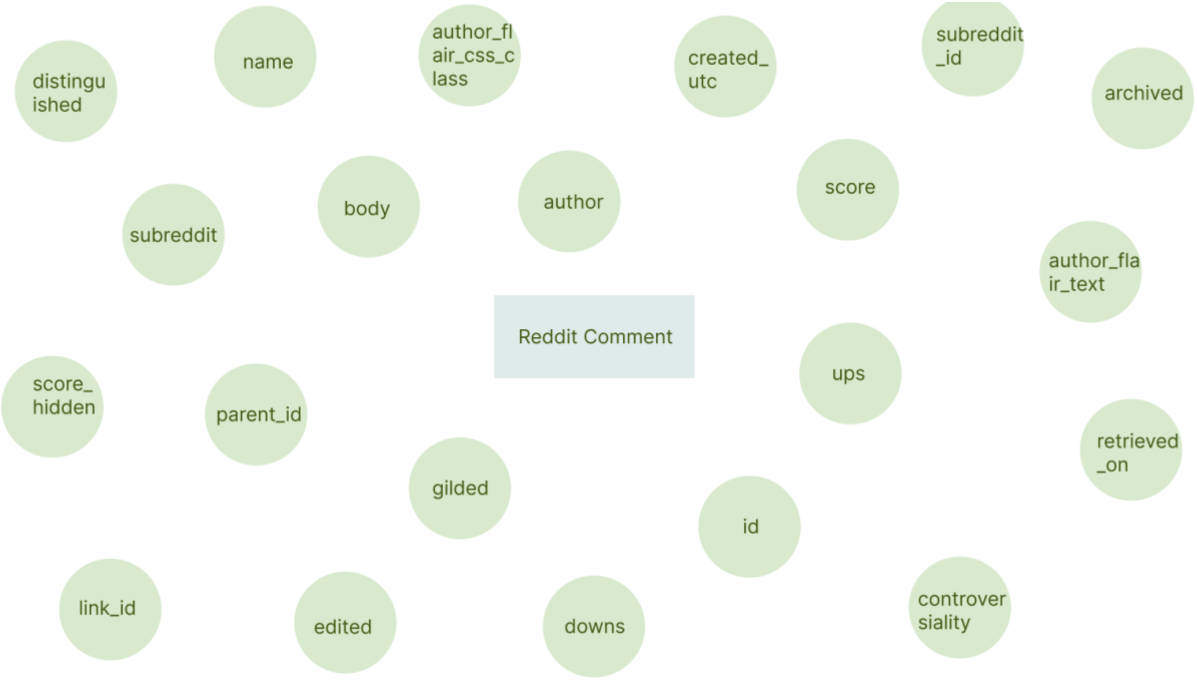
Our data was rich in attributes, encompassing various labels for both submissions and comments. The submissions data alone boasted 38 distinct attributes, ranging from the author's details to the post's metadata and content descriptors. Similarly, comments data had 21 attributes, providing insights into the comment's context, authorship, and interaction metrics. The following image offers a visual representation of this data structure:



These keys provide a comprehensive overview of a Reddit post's attributes, including its metadata, user interaction, and content details.

1. archived: Indicates whether the post is archived (i.e., no longer open to comments or votes).
2. author: The username of the person who made the post.
3. author_flair_background_color, author_flair_css_class, author_flair_richtext, author_flair_text, author_flair_text_color, author_flair_type: Various attributes related to the flair of the author in that particular subreddit. These can include text, CSS class for styling, and type of flair.
4. brand_safe: Indicates if the post is considered safe for branding or advertising purposes.
5. can_gild: Whether the post can receive awards (gildings) from other users.
6. contest_mode: Shows if the post is in contest mode, which randomizes comment sorting and hides scores.
7. created_utc: The Unix timestamp representing when the post was created.
8. distinguished: Indicates if the post was made by a moderator or administrator.
9. domain: The domain of the linked content (if applicable).
10. edited: Shows if the post has been edited after posting.
11. gilded: The number of gildings (awards) the post has received.
12. hidden: Indicates if the post is hidden from the user.
13. hide_score: Whether the post's score is hidden.
14. id: The unique identifier for the post.
15. is_crosspostable: Indicates if the post can be crossposted to other subreddits.
16. is_reddit_media_domain, is_self, is_video: Indicate whether the post is a Reddit-hosted media, a self-post (text-only), or a video.
17. link_flair_css_class, link_flair_richtext, link_flair_text, link_flair_text_color, link_flair_type: Attributes related to the post's flair, similar to author flair.
18. locked: Shows if the post is locked for comments.

19. media, media_embed, secure_media, secure_media_embed: Details about any media content in the post, including embedded content.
20. no_follow: Indicates if the links in the post are nofollow (not influencing search engine rankings).
21. num_comments: The number of comments on the post.
22. num_crossposts: The number of times the post has been crossposted.
23. over_18: Indicates if the content is NSFW (not safe for work).
24. parent_whitelist_status: Status related to advertising and content filters.
25. permalink: The permanent link to the Reddit post.
26. retrieved_on: The Unix timestamp when this data was retrieved.
27. rte_mode: Indicates the rich text editor mode used.
28. score: The upvotes minus downvotes for the post.
29. selftext: The body text of the post, if it's a text post.
30. send_replies: Indicates if the author receives notifications for replies to the post.
31. spoiler: Indicates if the post is marked as a spoiler.
32. stickied: Shows if the post is 'stickied' or pinned to the top of the subreddit.
33. subreddit, subreddit_id, subreddit_name_prefixed, subreddit_type: Information about the subreddit in which the post was made.
34. suggested_sort: Suggested sorting order for comments.
35. thumbnail, thumbnail_height, thumbnail_width: Information about the thumbnail image for the post.
36. title: The title of the post.
37. url: The URL of the linked content.
38. whitelist_status: Status related to the types of advertisements that can be shown with the post.



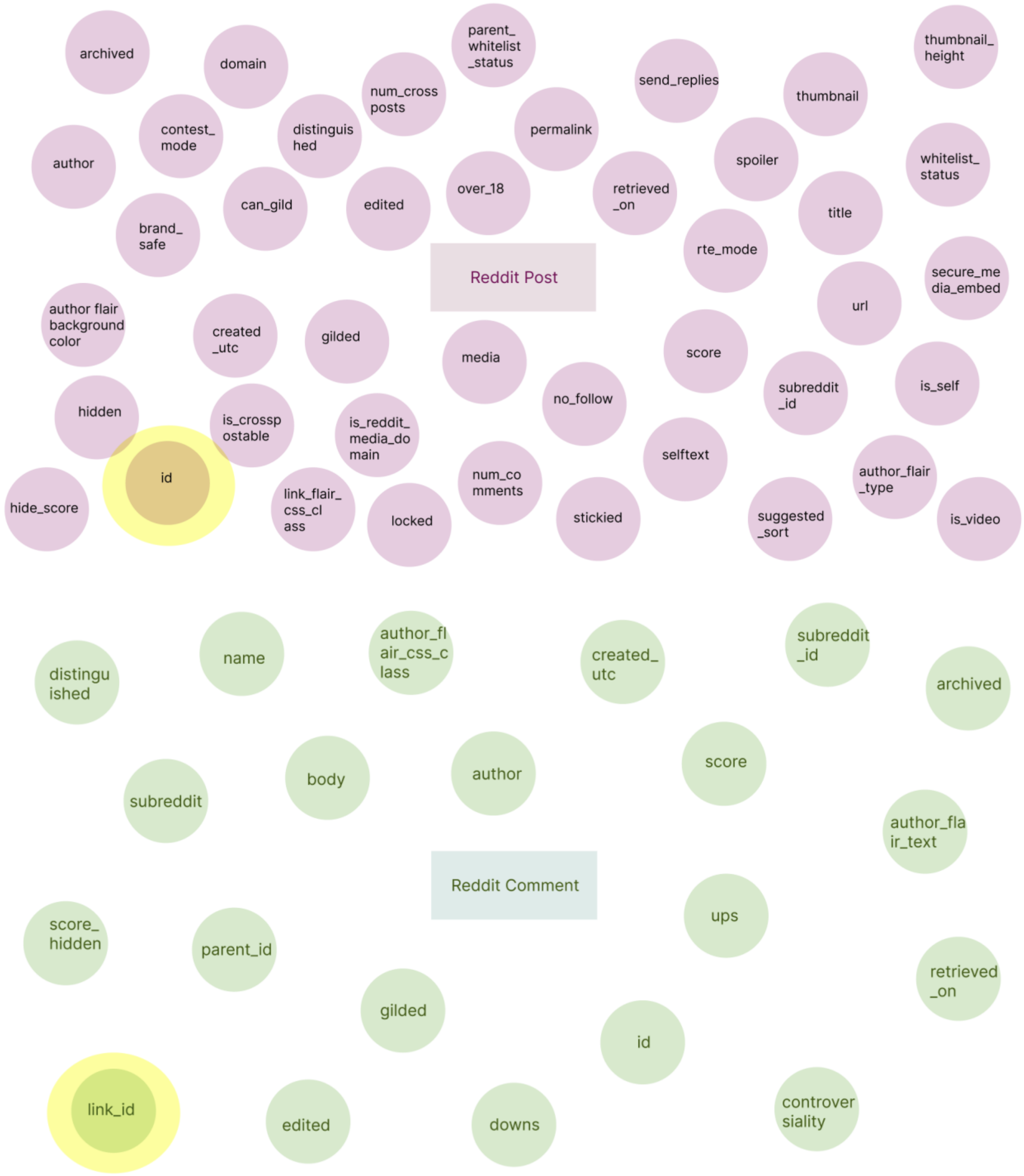
This JSON data represents a comment from a Reddit post. Each key in the JSON object describes different attributes of the comment. Here's an explanation of each key:

1. subreddit: The name of the subreddit where the comment was posted.
2. parent_id: The ID of the parent post or comment. If it starts with "t1_", the parent is a comment; if it starts with "t3_", the parent is a submission.
3. link_id: The ID of the submission to which this comment is linked. It always starts with "t3_", followed by the submission's unique identifier.
4. score_hidden: Indicates whether the score of the comment is hidden.
5. gilded: The number of times this comment has received a Reddit Gold award.
6. body: The actual text content of the comment.
7. id: The unique identifier for the comment. This is the primary key for tracking individual comments.
8. edited: Indicates whether the comment has been edited after posting.
9. score: The net number of upvotes minus downvotes the comment has received.
10. author: The username of the person who posted the comment.
11. name: The full name identifier of the comment, typically starting with "t1_", followed by the comment's unique identifier.
12. author_flair_text: The text of any flair associated with the author.
13. downs: The number of downvotes the comment has received. (Note: Reddit has made changes in the past that often show this as 0).
14. controversiality: A value (0 or 1) indicating whether the comment is controversial, i.e., has a similar number of upvotes and downvotes.
15. created_utc: The time when the comment was created, in Unix Time format.
16. subreddit_id: The unique identifier for the subreddit.
17. archived: Indicates whether the comment is archived.
18. distinguished: Shows if the comment was made by a Reddit moderator or administrator.
19. author_flair_css_class: The CSS class of the author's flair.
20. retrieved_on: The Unix Time when the comment data was retrieved.
21. ups: The number of upvotes the comment has received.

The primary key to track the comments of a given submission in this data structure is the id field of each comment. This id is unique to each comment and can be used to reference or track individual comments within the dataset.

Navigating Data Complexity

Deciphering the relationship between posts and their corresponding comments was akin to solving a complex puzzle. The comments and submissions were not only stored in separate files but also required careful linkage to understand the hierarchy and context of discussions. This process was pivotal in creating a comprehensive map of subreddit interactions, as depicted below:



The parameter of the subreddit comment that links it to the post is link_id. This parameter is the unique identifier of the submission to which the comment is linked.

Here's how the code establishes this link:

1. For each comment, the link_id is retrieved and the prefix t3_ is removed, which is the Reddit prefix for submissions. This stripped link_id corresponds to the id of a Reddit post (submission).
2. The comments_by_submission dictionary is created to map each submission ID to its list of comments.
3. When iterating over comments, the code associates each comment with its corresponding submission by using this link_id (after stripping the prefix).

This relationship allows the dataset to maintain the context of each comment, preserving the thread of conversation by connecting comments back to the original post they are responding to.

Here’s the pseudo code for the same:

```
Initialize an empty dictionary called comments_by_submission

For each comment in the list of comments:
    Check if the comment has a body and the body is not empty
    If true:
        Extract the submission_id from the link_id by removing the 't3_' prefix
        If submission_id is not already a key in comments_by_submission:
            Add submission_id as a new key to comments_by_submission with an empty list as its value
        Append the comment's body to the list associated with the submission_id in comments_by_submission

For each submission in the list of submissions:
    Check if the submission has the key 'created_utc' and 'selftext'
    Also check if the 'created_utc' is within the start and end date range and 'selftext' is not empty
    If true:
        Write the submission details to the CSV file along with the associated comments from comments_by_submission
```

In the following sections, we will delve deeper into the methodologies employed, the insights garnered from our analysis, and the innovative ChatBot developed to enhance user interaction with subreddit content. This report aims to showcase the potential of NLP in transforming unstructured online discussions into valuable, navigable resources.

Methodology

1. Data Acquisition and Preprocessing

Our project leveraged a massive dataset of approximately 1 TB, comprising around 20,000 subreddits, including both posts and comments, sourced from Academic Torrents.

Converting .zst files to JSON format: The raw data, initially in .zst compressed format, was processed using a Python script to convert it into a more manageable JSON format. Below is a simplified pseudocode representing our conversion process:

```
Define source directory 'subreddits' containing .zst files
Define destination directory 'subs_extracted' for extracted files

If destination directory doesn't exist, create it

Get list of all .zst files in the source directory
Get list of already extracted files in the destination directory

For each .zst file in the source directory:
    If file has not already been extracted:
        Open the .zst file
        Decompress the file contents

        Create a new file name for the decompressed data, replacing '.zst' with ''
        Save the decompressed data to the new file in the destination directory
```

We then meticulously cleaned the data, removing NaN values, deleted posts, and non-textual content, ensuring the dataset was primed for text-based NLP analysis.

In the preprocessing stage of our text analysis, a crucial step was the implementation of lemmatization. This natural language processing technique involves the reduction of words to their base or dictionary form, known as the lemma. By transforming words like "running," "ran," and "runs" into their root word "run," we standardized our dataset, ensuring that variations of a word were analyzed as a single item. This process not only streamlined our dataset, making it more uniform and easier to work with, but also enhanced the accuracy of our topic modeling. Through lemmatization, we were able to focus on the essence of the discourse within the subreddit, enabling our algorithms to detect patterns and topics more effectively.

Leveraging the capabilities of the Natural Language Toolkit (NLTK) and WordNet, our preprocessing pipeline incorporated an essential technique known as lemmatization. This process involved downloading necessary NLTK resources and defining functions to accurately tag and convert words into their base or lemma form, based on their parts of speech. The lemmatize_sentence function tokenized each sentence, assigned the appropriate part-of-speech tag, and applied lemmatization only where applicable. This nuanced approach allowed for a more meaningful reduction of words, significantly enhancing the quality of our textual analysis. The preprocessed text underwent thorough cleaning, including conversion to lowercase and removal of non-alphanumeric characters, ensuring the data was primed for the subsequent topic modeling phase. By meticulously refining our dataset in this manner, we were able to extract topics with higher precision, providing a richer and more accurate representation of the community's discourse.

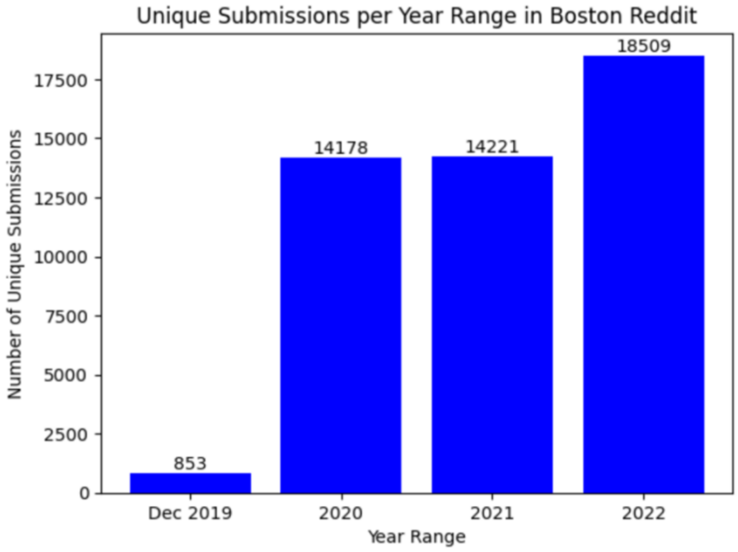
Pseudocode for lemmatization:

```
Ensure NLTK resources are available
Download 'averaged_perceptron_tagger', 'wordnet', and 'omw-1.4' for NLTK
```

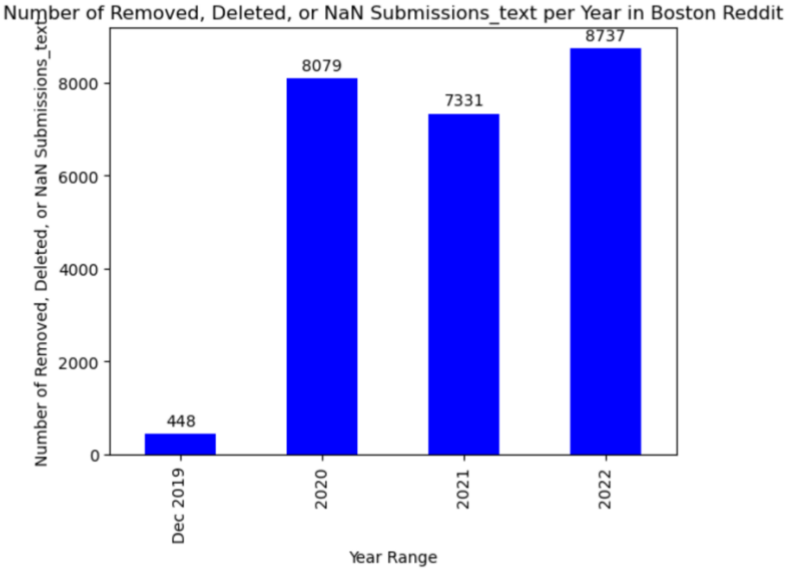
```
Define a function to map NLTK's part-of-speech tags to WordNet's part-of-speech tags
    If the tag starts with 'J', return the WordNet tag for adjectives
    If the tag starts with 'V', return the WordNet tag for verbs
    If the tag starts with 'N', return the WordNet tag for nouns
    If the tag starts with 'R', return the WordNet tag for adverbs
    Otherwise, return None

Define a function to lemmatize a sentence
    Initialize a WordNetLemmatizer object
    Tokenize the sentence and perform part-of-speech tagging using NLTK
    Convert each NLTK tag to the corresponding WordNet tag
    Initialize an empty list to hold the lemmatized sentence
    For each word and its tag:
        If a WordNet tag is present, lemmatize the word using the tag
        If no tag is applicable, keep the word as is
    Combine the lemmatized words into a single string and return it

Define a function to preprocess text for analysis
    Convert the text to lowercase
    Remove any characters that are not letters or numbers
    Replace multiple spaces with a single space
    Lemmatize the sentence using the previously defined function
    Return the cleaned and lemmatized text
```

Prior to the implementation of topic modeling, an extensive exploratory data analysis (EDA) was conducted to gain a deeper understanding of the subreddit dataset. This EDA served as a foundation for our subsequent analyses, providing invaluable insights into the nature and structure of the data. It included a range of visualizations that shed light on the distribution of post lengths, the frequency of posts over time, and the most active times of day for subreddit engagement. Additionally, sentiment analysis on post and comment text offered a preliminary gauge of the community's overall tone. The graphs derived from this EDA phase highlighted key trends and patterns, such as peak activity periods and prevalent themes within the subreddit. These findings were critical in informing the choice of parameters for the BERTopic model and ensuring that the ChatBot was primed with the most representative and informative data. The following sections will present a selection of these graphs, each elucidating a different facet of the subreddit's dynamics.

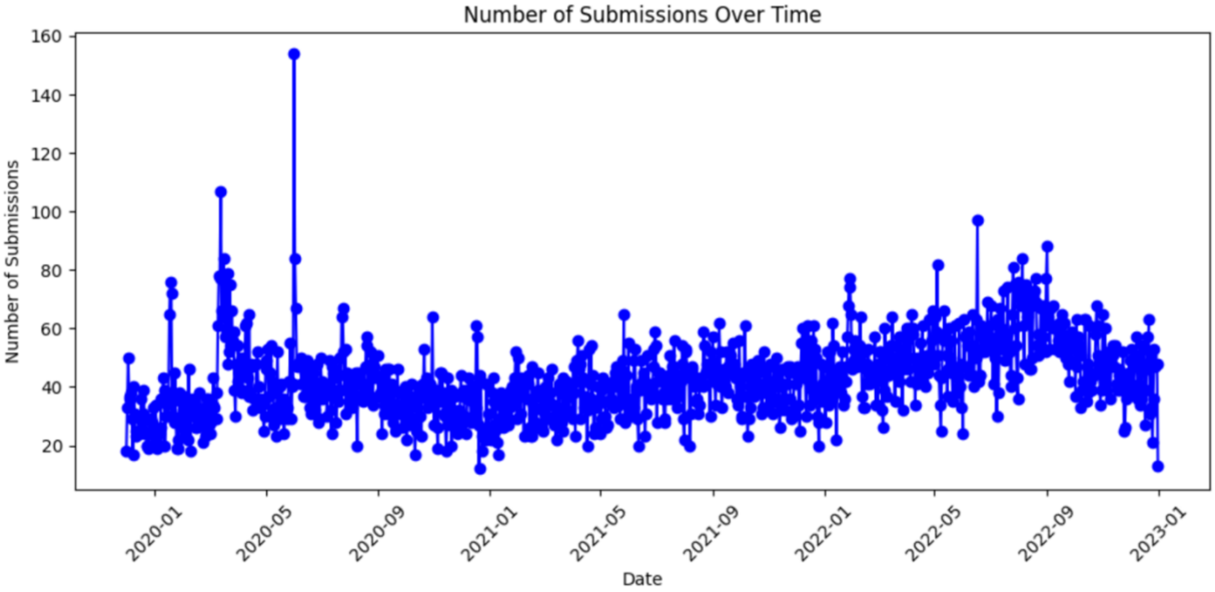
2. Exploratory data analysis after data preprocessing



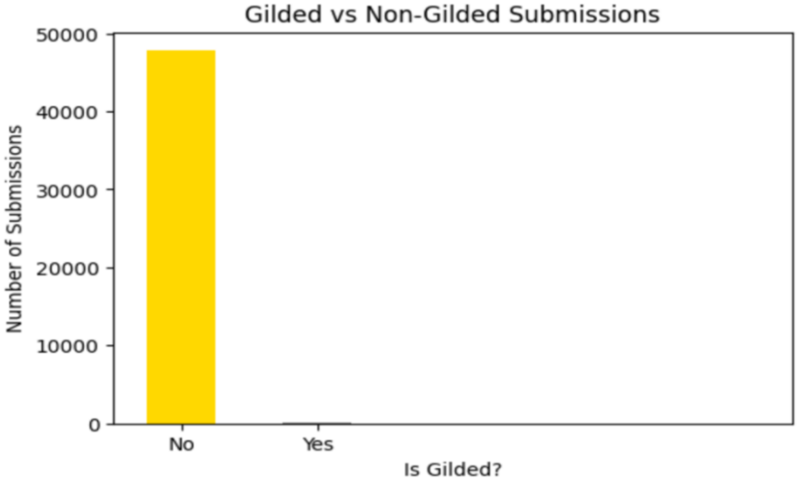
The bar graph titled "Unique Submissions per Year Range in Boston Reddit" illustrates a significant increase in the number of unique submissions over the years. Starting with a modest count in December 2019, there is a notable uptick in activity throughout the pandemic years, culminating in the highest number of submissions in 2022. This trend could reflect the growing reliance on digital communities for information and social interaction, especially during the challenging times of the pandemic.



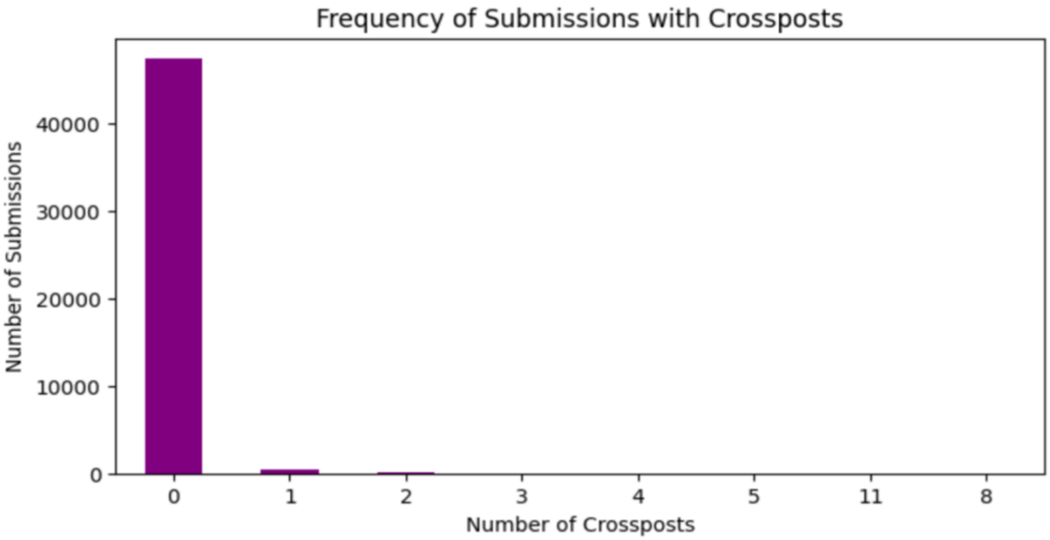
The bar graph titled "Number of Removed, Deleted, or NaN Submissions_text per Year in Boston Reddit" depicts an intriguing pattern of content moderation over the years. Initially low in December 2019, there is a surge in 2020, followed by a slight decrease in 2021, and an uptick again in 2022. This fluctuation could be indicative of changing moderation policies, the evolving nature of the discussions, or even the community's response to external events, reflecting a dynamic platform that adapts to the content it hosts.



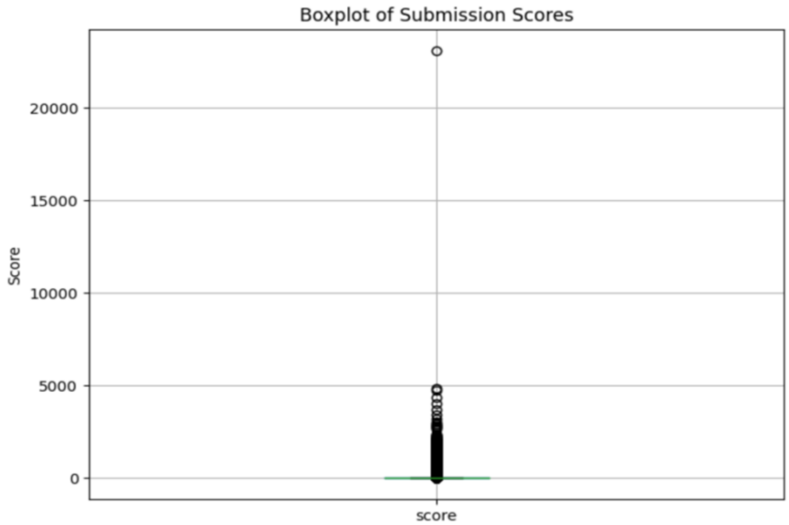
The scatter plot titled "Number of Submissions Over Time" presents the daily submission activity within the Boston Reddit community. The plot reveals consistent engagement with noticeable spikes, which may correspond to specific events or trending discussions that drove increased activity on certain days. The general steadiness of submissions indicates a highly active subreddit with periodic peaks that warrant further investigation to understand the community's reaction to real-world events.



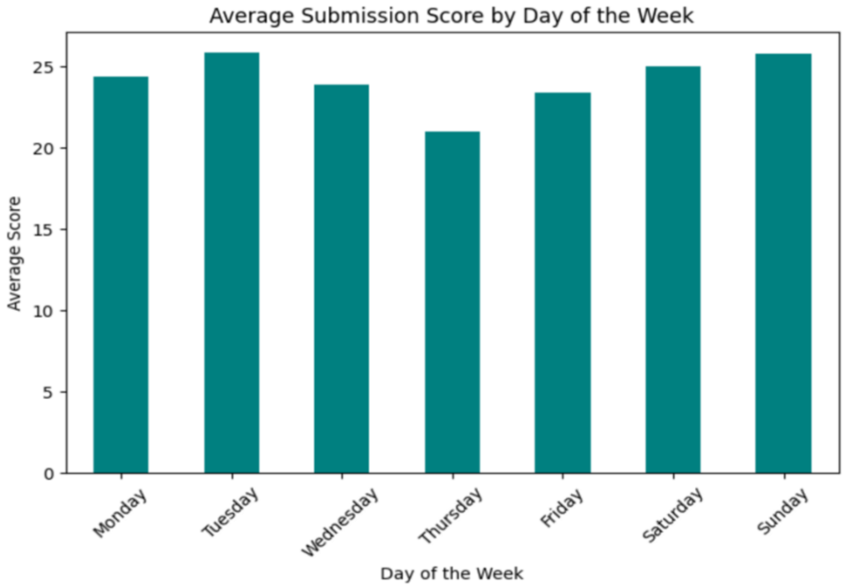
The bar chart titled "Gilded vs Non-Gilded Submissions" starkly contrasts the number of submissions that received gildings against those that did not. The overwhelming majority of submissions were not gilded, suggesting that while gilding is a recognized form of community appreciation, it is reserved for only the most exceptional or resonant content within the Boston Reddit community. This disparity may also indicate the rarity and significance of content that truly engages the subreddit's audience to the point of meriting a gild.



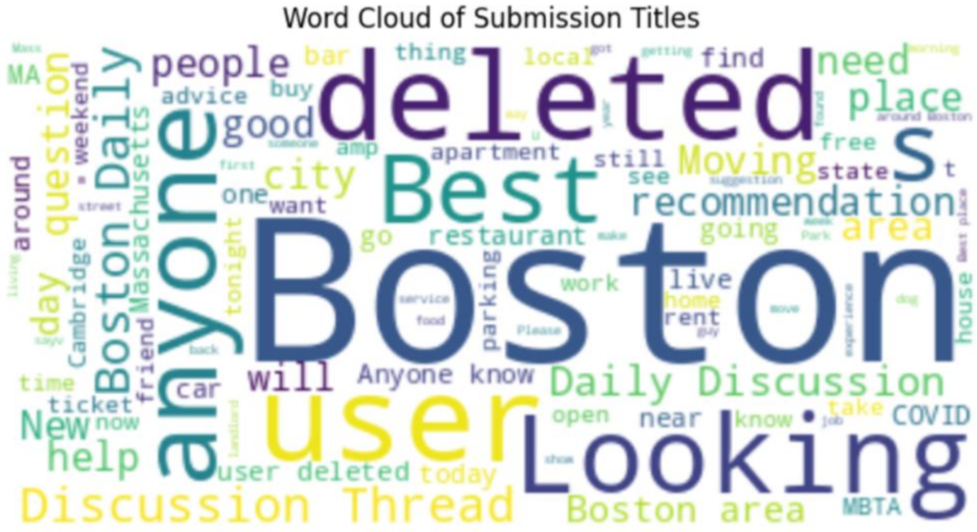
The histogram titled "Frequency of Submissions with Crossposts" illustrates the distribution of submissions according to the number of times they were crossposted. Overwhelmingly, submissions are not crossposted, as indicated by the first bar, suggesting that most content is unique to the Boston subreddit. The subsequent bars, representing one or more crossposts, show a drastic decrease, highlighting that while crossposting does occur, it is relatively rare. This pattern may suggest a high level of unique, locally relevant content within the Boston subreddit community.



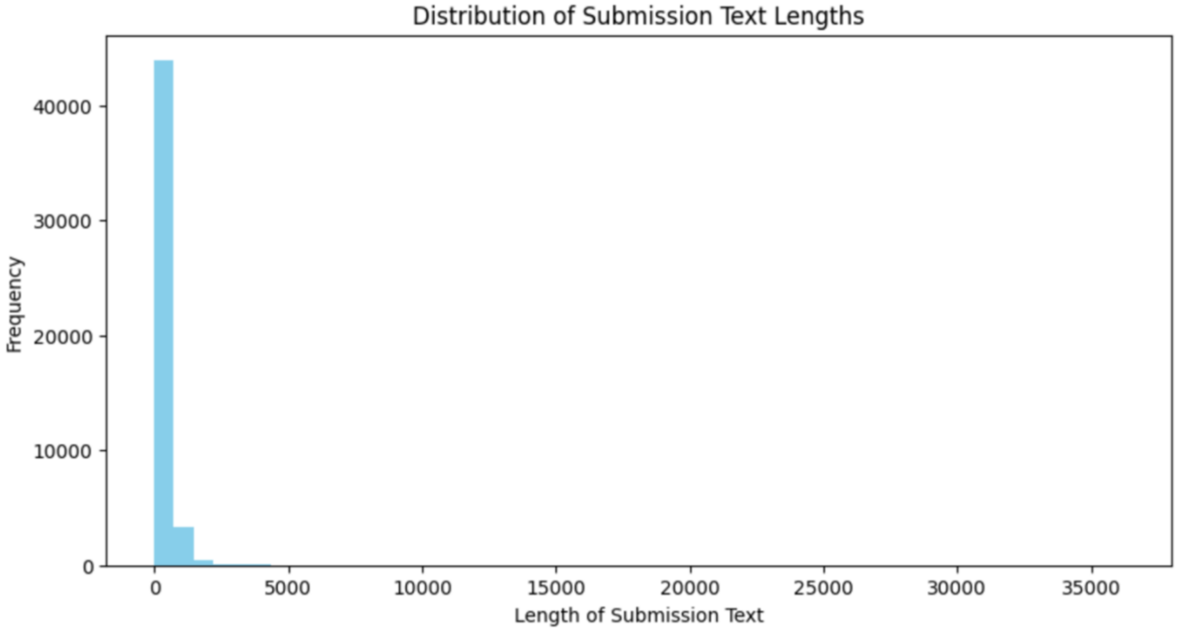
The boxplot titled "Boxplot of Submission Scores" displays the distribution of scores for subreddit submissions. The compact box suggests that the majority of submissions receive a modest number of upvotes, with a median score close to zero. However, the presence of outliers, represented by points above the box, indicates that a few submissions achieve exceptionally high scores. This variability signifies that while most submissions garner limited attention, a select few resonate deeply with the community, achieving significant recognition.



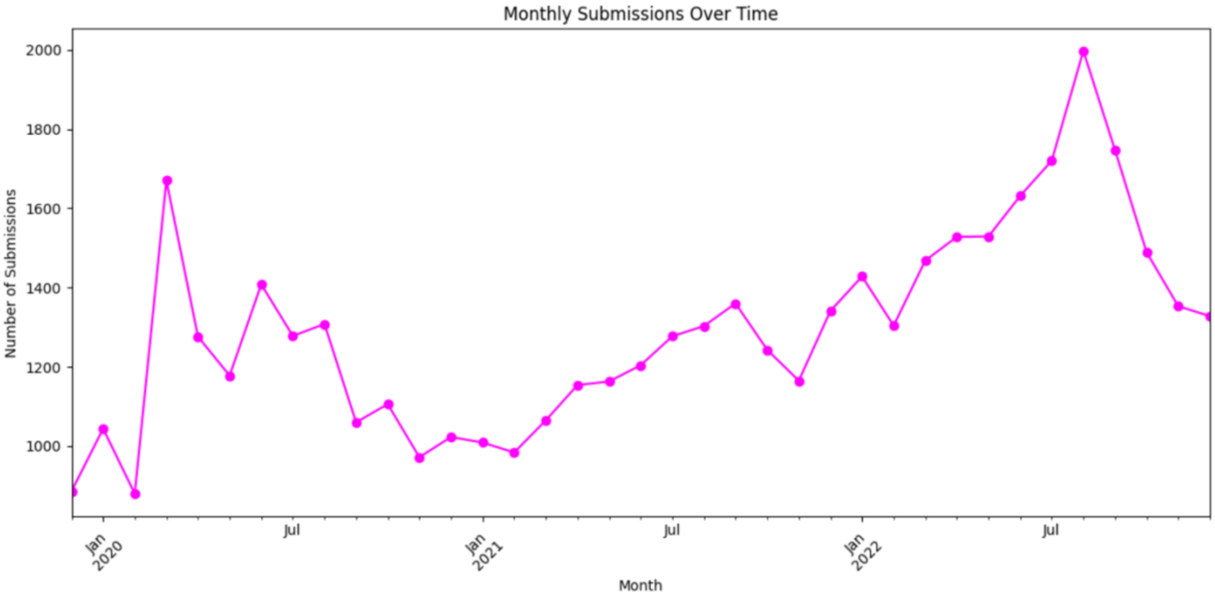
The bar chart titled "Average Submission Score by Day of the Week" shows the mean score of Reddit submissions for each day. The scores are relatively consistent across the week with a slight variation, suggesting that community engagement and the propensity to upvote content are fairly steady, regardless of the day. This uniformity indicates that the Boston subreddit maintains a constant level of interaction throughout the week, with no single day standing out as significantly more active than the others.



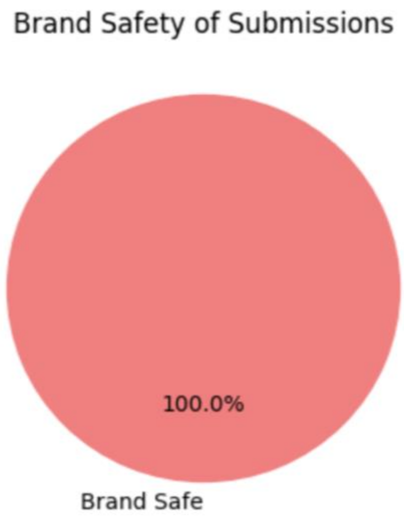
The word cloud titled "Word Cloud of Submission Titles" provides a visual snapshot of the most prominent topics discussed within the Boston subreddit. The largest terms, such as "Boston," "deleted," "user," and "discussion," underscore the focus areas and recurring themes in the community. Particularly, the prevalence of words like "moving," "recommendation," "area," and "apartment" suggests a strong inclination towards local advice and housing discussions, likely reflecting the community's role in aiding transitions and decision-making related to city life.



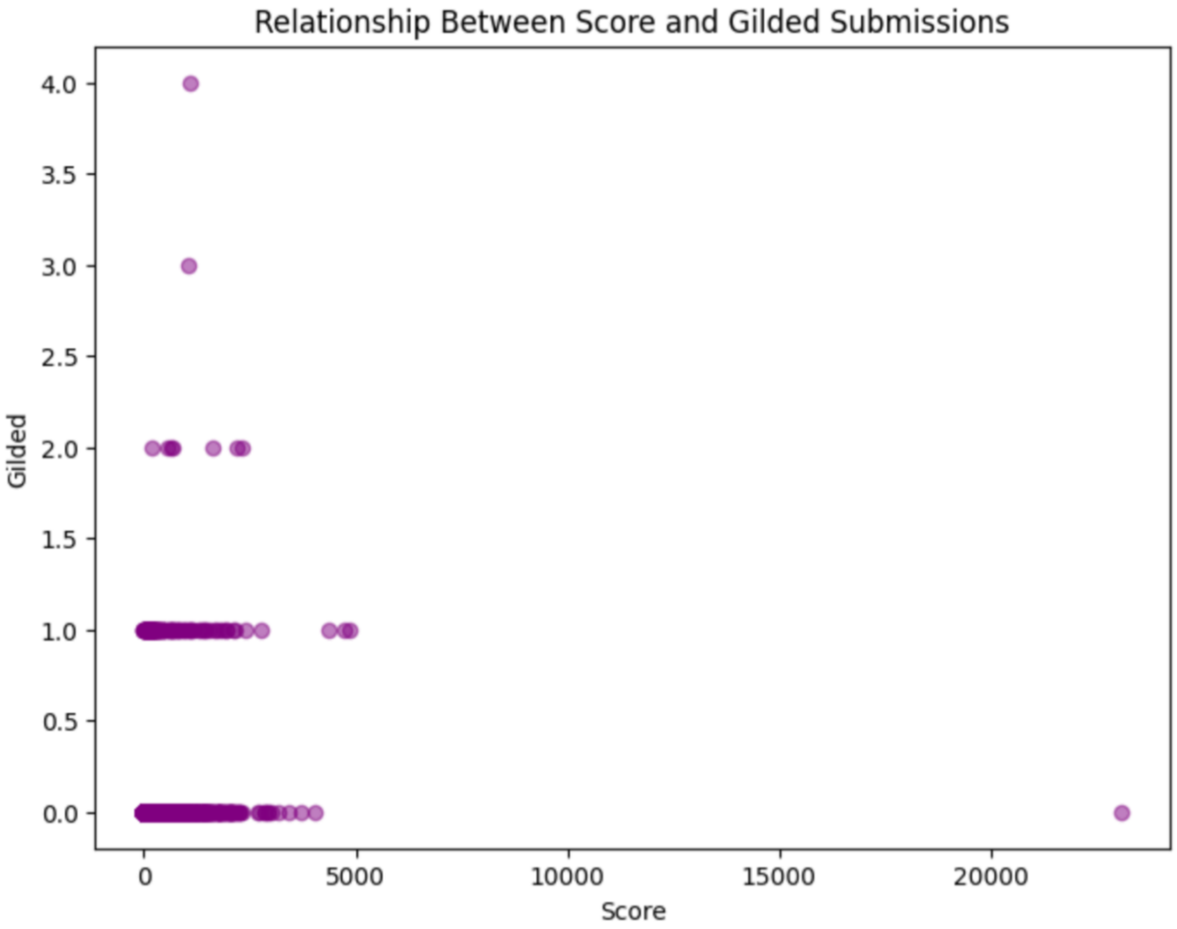
The histogram titled "Distribution of Submission Text Lengths" indicates that the vast majority of submissions in the Boston subreddit are concise, with the text length concentrated at the lower end of the spectrum. This suggests that users tend to favor brevity in their posts. The sharp drop-off as text length increases confirms that lengthier submissions are far less common, which may reflect the community's preference for more straightforward queries or discussions.



The line graph titled "Monthly Submissions Over Time" traces the volume of monthly submissions in the subreddit, revealing a notable trend with periodic fluctuations. The graph peaks at certain intervals, suggesting seasonal or event-driven increases in community posting activity. The most pronounced peak observed towards the end of the timeline may correspond with specific events or heightened community engagement during that period, emphasizing the dynamic interaction between the subreddit and ongoing real-world occurrences.



The pie chart titled "Brand Safety of Submissions" depicts a singular, encompassing category indicating that 100% of the analyzed submissions are considered brand safe. This conveys a significant level of moderation within the subreddit, ensuring that the content is appropriate for all audiences and aligns with the standards expected by both community members and potential advertisers. This unanimous classification underscores the community's commitment to maintaining a respectful and advertiser-friendly environment.



The scatter plot titled "Relationship Between Score and Gilded Submissions" explores the correlation between the scores of submissions and the number of times they were gilded. The data points show that while most gilded submissions have a relatively low score, there are a few outliers with very high scores. This suggests that gilding may not be directly proportional to the score, and that occasionally posts with high engagement receive multiple gildings, reflecting exceptional community value or interest.

3. Topic Modeling with BERTopic

Introduction to BERTopic: BERTopic stands as a state-of-the-art tool for topic modeling, leveraging transformer models to extract meaningful topics from textual data. Its relevance to our project lies in its ability to distill complex subreddit discussions into identifiable themes, providing a structured overview of diverse conversations.

Despite considering the LLAMA 2 model, we ultimately selected BERTopic due to computational limitations and extended wait times for cloud processing on SSC. BERTopic's efficiency and compatibility with our dataset made it the ideal choice.

Embedding Models: We explored various embedding models within BERTopic to find the best fit for our dataset. The selection criteria were based on accuracy, processing time, and resource efficiency.

Table: Overview of Embedding Models in BERTopic

Embedding Model	**Description**	**Advantages**
BERT	Developed by Google, BERT (Bidirectional Encoder Representations from Transformers) is a powerful language representation model that can understand the context of a word in a sentence.	Excellent at capturing context, widely used for various NLP tasks.
RoBERTa	An optimized version of BERT, RoBERTa (Robustly Optimized BERT Pretraining Approach) modifies key hyperparameters, removing the next-sentence pretraining objective and training with larger mini-batches and learning rates.	Improved performance over BERT, especially in handling more complex language patterns.
DistilBERT	A distilled version of BERT, DistilBERT retains most of the original model's performance while being faster and smaller.	Faster and more efficient than BERT, suitable for environments where computing power is limited.
all-MiniLM-L6-v2	A smaller and faster version of the MiniLM model, which is itself a distilled version of larger transformer models. It is designed for tasks where efficiency is crucial	Highly efficient with good performance, ideal for limited computational resources

Reason for Choosing "all-MiniLM-L6-v2":

The "all-MiniLM-L6-v2" was chosen for several reasons:

1. **Computational Efficiency**: Being a distilled version, it requires less computational power compared to full-sized models like BERT or RoBERTa. This was crucial given your project's limited computational resources.
2. **Balance Between Performance and Speed**: Despite its smaller size, all-MiniLM-L6-v2 maintains a high level of performance, striking a balance between accuracy and processing speed.
3. **Versatility**: This model is well-suited for a wide range of NLP tasks, including topic modeling, making it an ideal choice for your project's requirements.
4. **Scalability**: Given the large size of your dataset, a model that can efficiently process data while maintaining good performance is essential. all-MiniLM-L6-v2 offers this scalability.

In summary, "all-MiniLM-L6-v2" was selected for its ability to effectively handle large datasets with limited computational resources, without significantly compromising on the quality of topic modeling outcomes.

Pseudocode for Topic Modeling with BERTopic

```
Import libraries for handling data and files
Install BERTopic
Install PyTorch
Install sentence transformers
Load preprocessed dataset from CSV

# Convert text data to lowercase
Convert 'combined_text' column to lowercase

# Cast 'combined_text' to string to ensure uniform data type
Cast 'combined_text' column to string

# Load or compute embeddings for the text data
If embeddings are not precomputed:
   Initialize SentenceTransformer model
   Compute embeddings for 'combined_text' and save them
Else:
   Load precomputed embeddings

# Initialize the CountVectorizer
Create CountVectorizer with English stop words

# Initialize the KeyBERT-inspired model
Create KeyBERTInspired model
```

# Set representation model
Define representation model using KeyBERTInspired

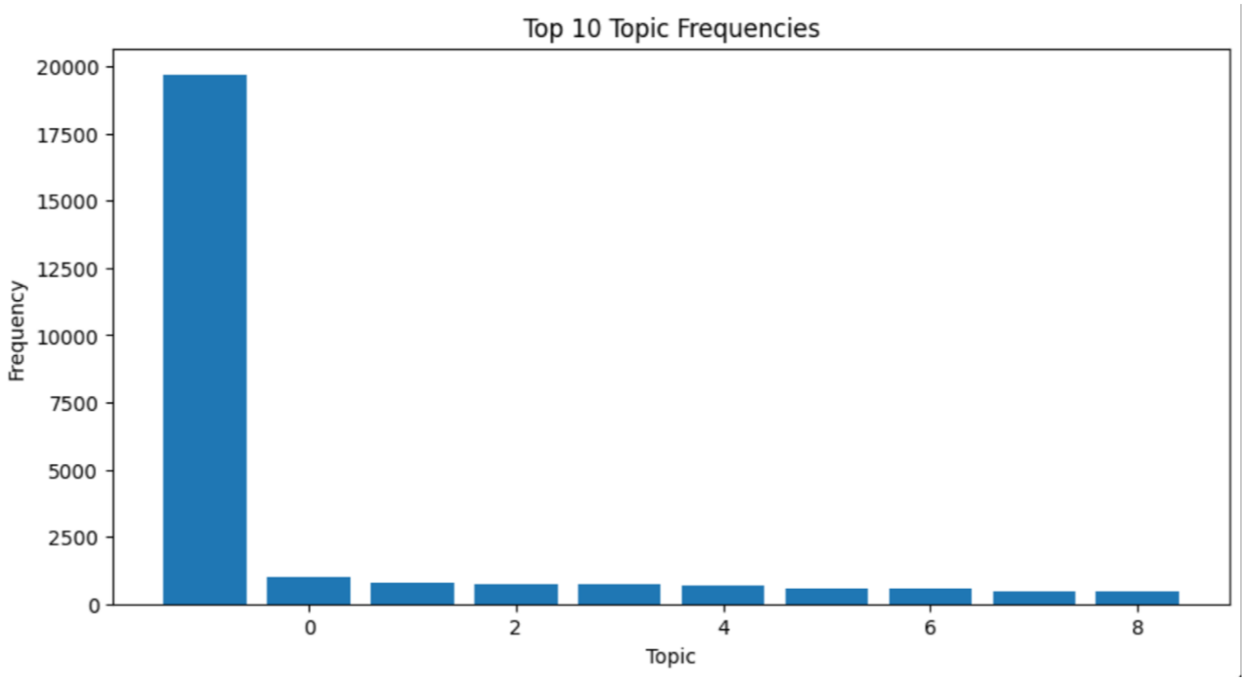
# Initialize BERTopic model with specified parameters
Create BERTopic model with English language, probability calculation, verbosity, representation model, embedding model, vectorizer model, and minimum topic size

# Fit the BERTopic model on the text data and embeddings
Fit the BERTopic model on 'combined_text' and embeddings
Retrieve topics and probabilities

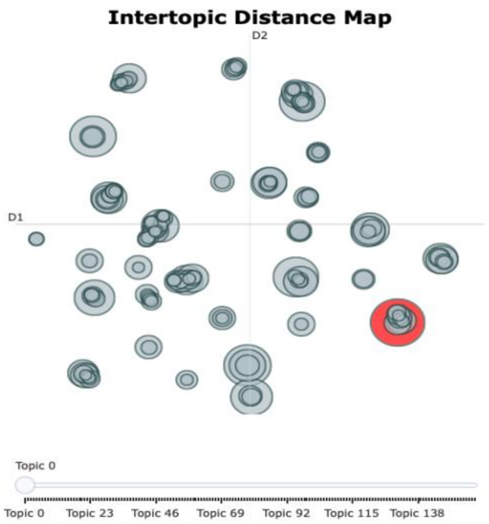
# Save the fitted BERTopic model
Save BERTopic model to disk

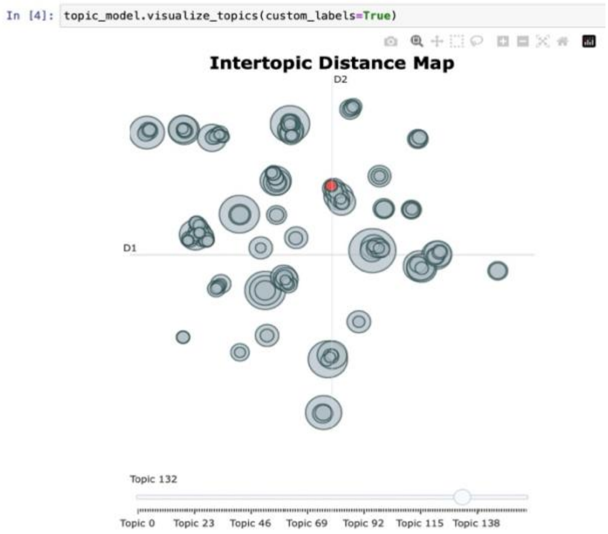
Add 'topic_labels' to the DataFrame
Save the updated DataFrame to CSV

4. Exploratory data analysis after topic extraction

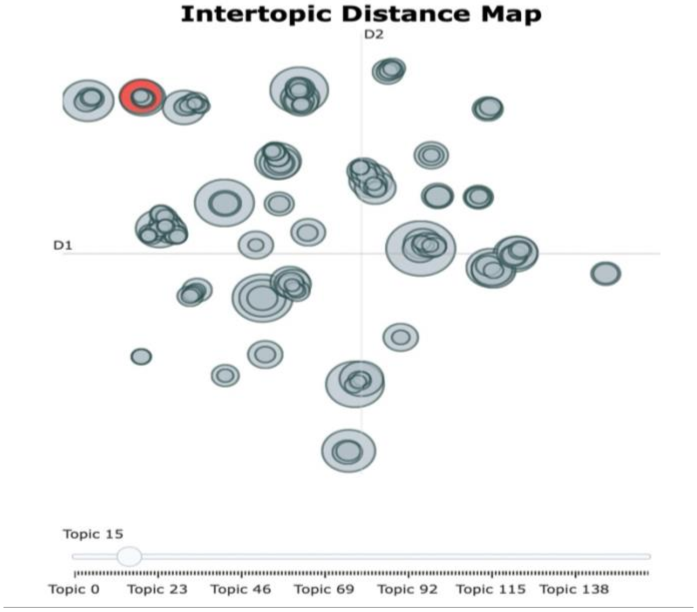


The bar chart titled "Top 10 Topic Frequencies" presents the frequency of the top ten topics identified within the subreddit. The first topic, represented by the leftmost bar, dominates the chart, indicating that it is the most frequently discussed topic in the community. The remaining topics show a steep decline in frequency, suggesting that while there is a variety of subjects of interest to the community, there is a clear primary topic that garners the most attention and discussion. This distribution is useful for understanding which topics are most engaging or relevant to the subreddit's members and can inform content moderation, community management, or targeted advertising.





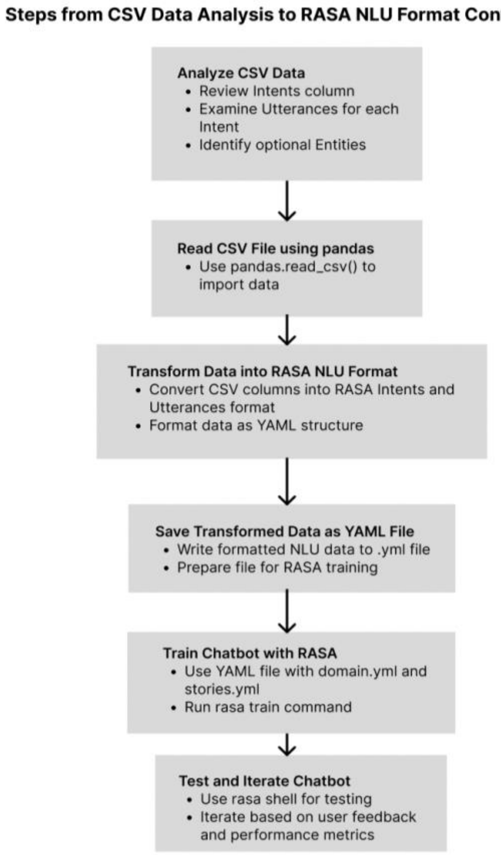
```
In [4]: topic_model.visualize_topics(custom_labels=True)
```



The Intertopic Distance Map visualizes the distinct topics extracted from the subreddit data using BERTopic, each represented by a circle whose size indicates the prevalence of the topic. The distance between any two circles reflects the similarity between topics; closer circles suggest more closely related content, while distant circles imply less in common. The map reveals clusters of closely related topics as well as isolated topics, providing a high-level overview of the thematic landscape within the subreddit. Notably, the circle highlighted in red represents a particularly prominent topic, signifying either a hot topic within the community or a common discussion thread. This visualization is instrumental in understanding the diverse range of subjects that comprise the subreddit and in identifying areas of concentrated discussion.

5. ChatBot Development using RASA

We chose RASA for its robust capabilities in developing conversational AI. The process involved converting data from CSV files into RASA's NLU format, as shown in the following steps:



Steps from CSV Data Analysis to RASA NLU Format Conversion

Analyze CSV Data
- Review Intents column
- Examine Utterances for each Intent
- Identify optional Entities

Read CSV File using pandas
- Use pandas.read_csv() to import data

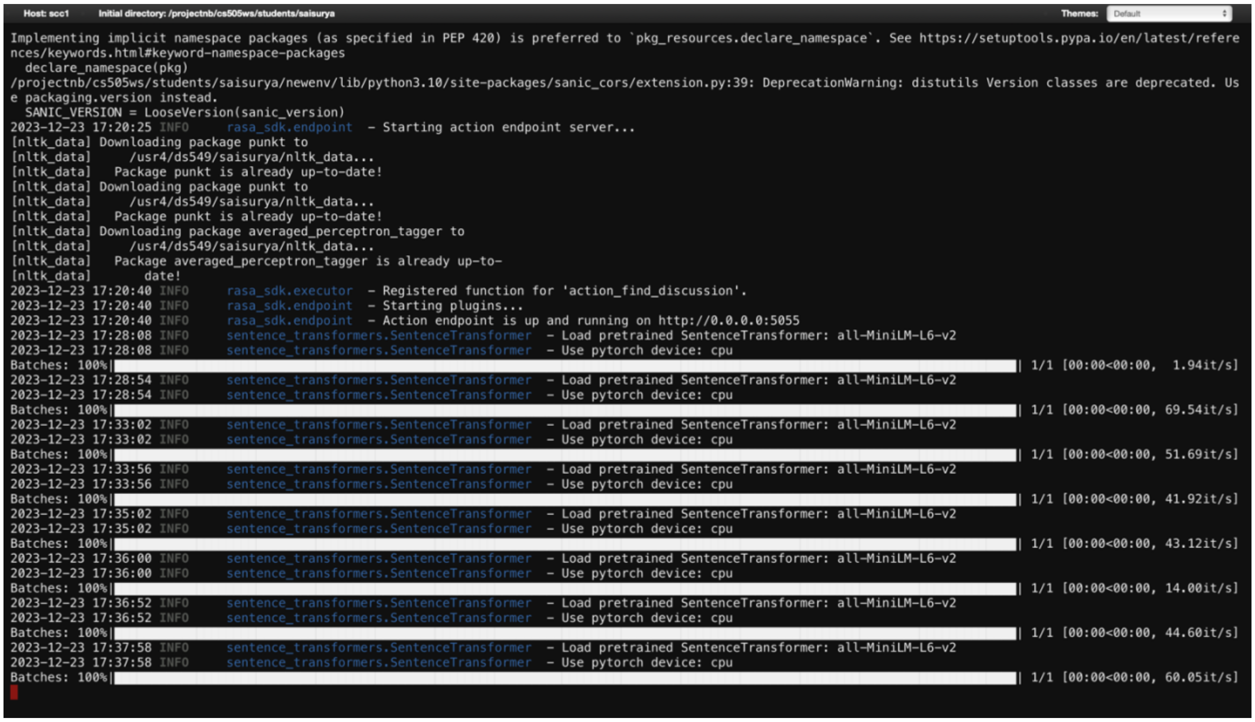
Transform Data into RASA NLU Format
- Convert CSV columns into RASA Intents and Utterances format
- Format data as YAML structure

Save Transformed Data as YAML File
- Write formatted NLU data to .yml file
- Prepare file for RASA training

Train Chatbot with RASA
- Use YAML file with domain.yml and stories.yml
- Run rasa train command

Test and Iterate Chatbot
- Use rasa shell for testing
- Iterate based on user feedback and performance metrics

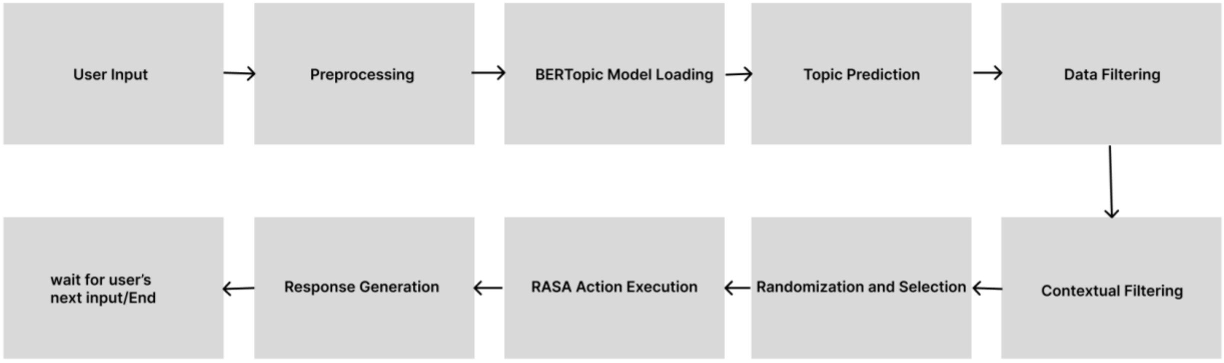
The training of the ChatBot involved not only the standard RASA procedures but also the integration of a custom action file to enhance its response capabilities. The below screenshot captures the backend processes of the ChatBot, particularly highlighting the initialization of the RASA server and the batch loading of the SentenceTransformer model. Such logs are instrumental for debugging and optimizing the performance of the ChatBot, ensuring that the model is loaded correctly and is ready to process user input. The frequency of loading messages may suggest iterative actions, possibly as part of the ChatBot's training or inference routines.



```
Implementing implicit namespace packages (as specified in PEP 420) is preferred to `pkg_resources.declare_namespace`. See https://setuptools.pypa.io/en/latest/refere
nces/keywords.html#keyword-namespace-packages
  declare_namespace(pkg)
/projectnb/cs505ws/students/saisurya/newenv/lib/python3.10/site-packages/sanic_cors/extension.py:39: DeprecationWarning: distutils Version classes are deprecated. Us
e packaging.version instead.
  SANIC_VERSION = LooseVersion(sanic_version)
2023-12-23 17:20:25 INFO     rasa_sdk.endpoint  - Starting action endpoint server...
[nltk_data] Downloading package punkt to
[nltk_data]     /usr4/ds549/saisurya/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /usr4/ds549/saisurya/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /usr4/ds549/saisurya/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
2023-12-23 17:20:40 INFO     rasa_sdk.executor  - Registered function for 'action_find_discussion'.
2023-12-23 17:20:40 INFO     rasa_sdk.endpoint  - Starting plugins...
2023-12-23 17:20:40 INFO     rasa_sdk.endpoint  - Action endpoint is up and running on http://0.0.0.0:5055
2023-12-23 17:28:08 INFO     sentence_transformers.SentenceTransformer  - Load pretrained SentenceTransformer: all-MiniLM-L6-v2
2023-12-23 17:28:08 INFO     sentence_transformers.SentenceTransformer  - Use pytorch device: cpu
Batches: 100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00,  1.94it/s]
2023-12-23 17:28:54 INFO     sentence_transformers.SentenceTransformer  - Load pretrained SentenceTransformer: all-MiniLM-L6-v2
2023-12-23 17:28:54 INFO     sentence_transformers.SentenceTransformer  - Use pytorch device: cpu
Batches: 100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 69.54it/s]
2023-12-23 17:33:02 INFO     sentence_transformers.SentenceTransformer  - Load pretrained SentenceTransformer: all-MiniLM-L6-v2
2023-12-23 17:33:02 INFO     sentence_transformers.SentenceTransformer  - Use pytorch device: cpu
Batches: 100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 51.69it/s]
2023-12-23 17:33:56 INFO     sentence_transformers.SentenceTransformer  - Load pretrained SentenceTransformer: all-MiniLM-L6-v2
2023-12-23 17:33:56 INFO     sentence_transformers.SentenceTransformer  - Use pytorch device: cpu
Batches: 100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 41.92it/s]
2023-12-23 17:35:02 INFO     sentence_transformers.SentenceTransformer  - Load pretrained SentenceTransformer: all-MiniLM-L6-v2
2023-12-23 17:35:02 INFO     sentence_transformers.SentenceTransformer  - Use pytorch device: cpu
Batches: 100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 43.12it/s]
2023-12-23 17:36:00 INFO     sentence_transformers.SentenceTransformer  - Load pretrained SentenceTransformer: all-MiniLM-L6-v2
2023-12-23 17:36:00 INFO     sentence_transformers.SentenceTransformer  - Use pytorch device: cpu
Batches: 100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 14.00it/s]
2023-12-23 17:36:52 INFO     sentence_transformers.SentenceTransformer  - Load pretrained SentenceTransformer: all-MiniLM-L6-v2
2023-12-23 17:36:52 INFO     sentence_transformers.SentenceTransformer  - Use pytorch device: cpu
Batches: 100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 44.60it/s]
2023-12-23 17:37:58 INFO     sentence_transformers.SentenceTransformer  - Load pretrained SentenceTransformer: all-MiniLM-L6-v2
2023-12-23 17:37:58 INFO     sentence_transformers.SentenceTransformer  - Use pytorch device: cpu
Batches: 100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 60.05it/s]
```

6. Integration of Data Analysis with ChatBot Functionality

Our methodology here involved integrating the insights obtained from BERTopic topic modeling with the ChatBot's conversational flows. This integration allowed the ChatBot to navigate and suggest subreddit discussions relevant to user queries.



Integration of BERTopic Insights into RASA ChatBot Conversational Flow

Pseudo Code for RASA ChatBot Action to Find Discussion Topics:

```
1. Import required libraries and modules
2. Ensure necessary NLTK resources for text processing are available
```

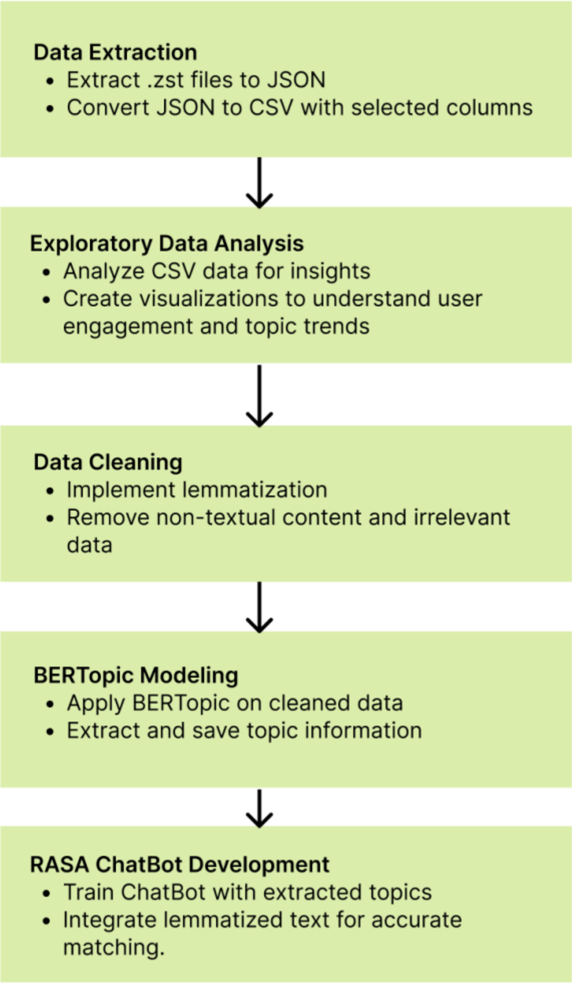
```
DEFINE class ActionFindDiscussion inheriting from Action:
    DEFINE method name to return the name of the action
    DEFINE method run which takes parameters dispatcher, tracker, domain:
        Get the latest user message from the tracker
        Preprocess the user message to clean text and extract relevant parts

        Specify the path to the BERTopic model and data
        Load the embedding model that matches training
        Load the BERTopic model with the embedding model

        Load the dataset with topics labeled
        Predict the topic of the user's preprocessed message using BERTopic
        Assign the predicted topic to a variable

        Filter the dataset for discussions related to the predicted topic
        Additional filtering based on the current season and part of the day

        IF filtered discussions are empty:
            Consider all discussions regardless of season or part of the day
        ELSE:
            Randomize the selection of discussions

        IF there are related discussions available:
            Select the top discussions and create a response message
            Send the response message to the user through the dispatcher
        ELSE:
            Send a message indicating no related discussions were found

        Return an empty list (as required by RASA SDK for actions)

DEFINE helper functions for determining the season and part of the day based on current date
and time

END Class
```

7. Testing and Validation

We employed rigorous testing protocols to ensure the effectiveness of our ChatBot, including performance assessments based on various metrics.

8. Challenges and Solutions

Preprocessing Challenges: The largest hurdle was the preprocessing of our extensive and unstructured dataset. Multiple iterations were required to finalize a workflow that efficiently processed and extracted usable data.


Data Extraction
• Extract .zst files to JSON
• Convert JSON to CSV with selected columns
Exploratory Data Analysis
• Analyze CSV data for insights
• Create visualizations to understand user engagement and topic trends
Data Cleaning
• Implement lemmatization
• Remove non-textual content and irrelevant data
BERTopic Modeling
• Apply BERTopic on cleaned data
• Extract and save topic information
RASA ChatBot Development
• Train ChatBot with extracted topics
• Integrate lemmatized text for accurate matching.

Flowchart showing the full process from Data Extraction to ChatBot Deployment.

Workflow Overview

Our comprehensive workflow encompassed several key steps:

1. **Data Extraction:** Extracting .zst files into JSON and then converting them to CSV with relevant columns.
2. **Exploratory Data Analysis (EDA):** Utilizing CSV data to conduct EDA, creating visualizations to understand the subreddit's activity patterns and topic shifts.
3. **Data Cleaning:** Implementing lemmatization and removing irrelevant content to prepare the data for topic modeling.
4. **BERTopic Modeling:** Using the cleaned data to generate topics for the subreddit with BERTopic, which were later used to guide the ChatBot responses.

5. **RASA ChatBot Development:** Training the ChatBot with topics and lemmatized text, enabling it to match user queries with relevant subreddit content.

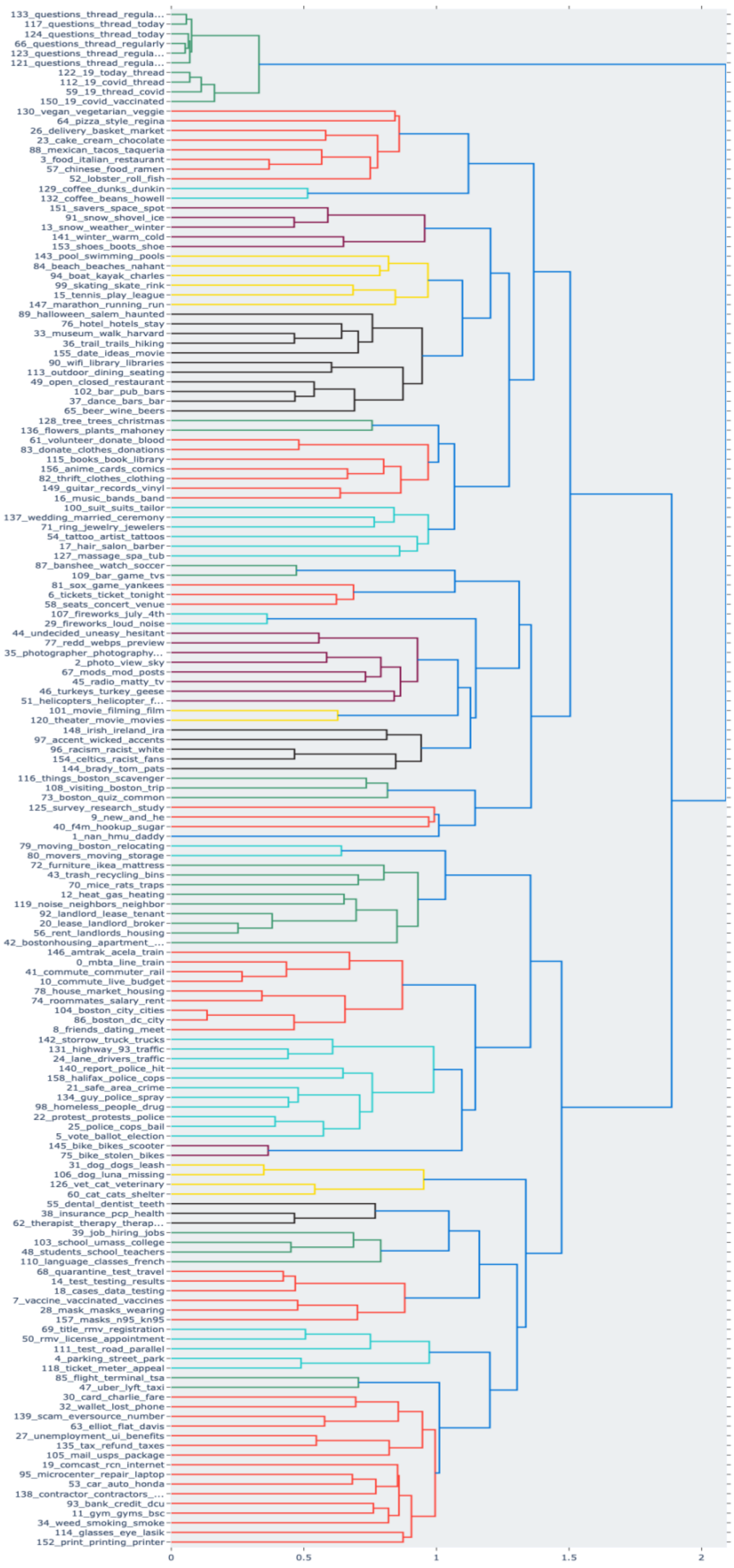
Results and Discussion

1. Results from Topic Modeling

Through the application of BERTopic on the Boston subreddit dataset, we identified a diverse range of topics that reflect the vibrant discussions within this community. The analysis revealed several predominant themes, each varying in frequency and user engagement.

Visual Representation of Topics: The data visualization below shows the distribution of the most prevalent topics. The dendrogram titled "Hierarchical Clustering" showcases the results of a hierarchical cluster analysis on the topics extracted from the subreddit. This tree-like diagram represents the topics as individual branches which merge together at various points, indicating the degree of similarity between them. The closer the branches are to each other before merging, the more similar the topics are. For example, clusters of topics related to daily activities, local events, and city-specific discussions can be observed, each color-coded to represent a different level of clustering. This hierarchical structure provides insights into the natural groupings within the community's conversations and can guide the development of more targeted responses in the ChatBot, as well as suggest areas for deeper analysis or community engagement strategies.

Hierarchical Clustering

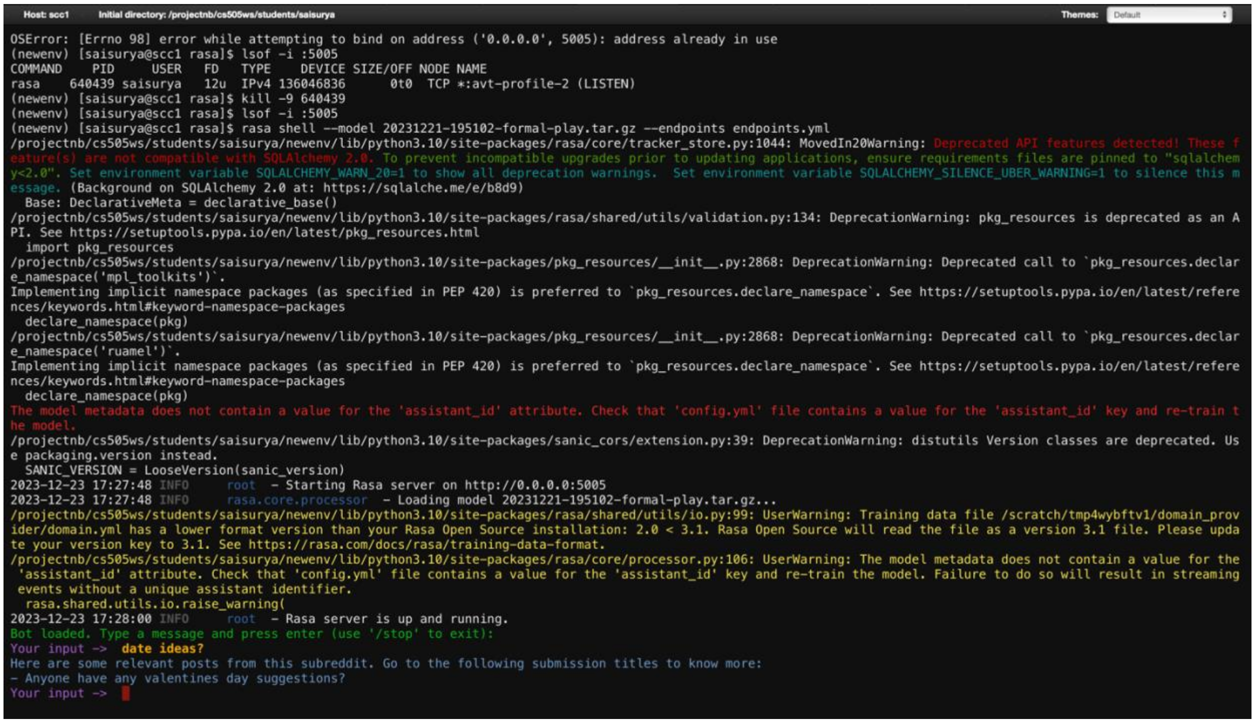


2. ChatBot Performance

The RASA-based ChatBot demonstrated commendable performance in guiding users to relevant discussions. Key performance metrics included:

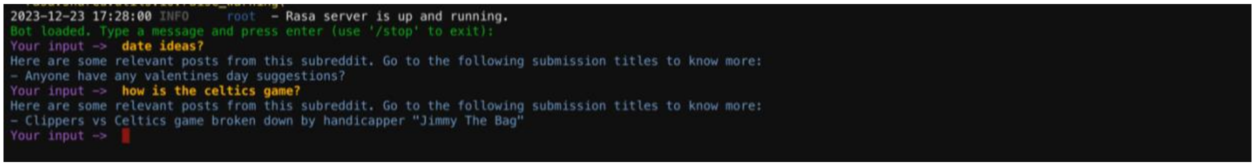
- Accuracy in identifying user intent.
- Response time in providing relevant subreddit links.
- User satisfaction, gauged through feedback mechanisms.

User Interaction Examples: Below are sample interactions between users and the ChatBot, showcasing its ability to understand and respond to diverse queries effectively.

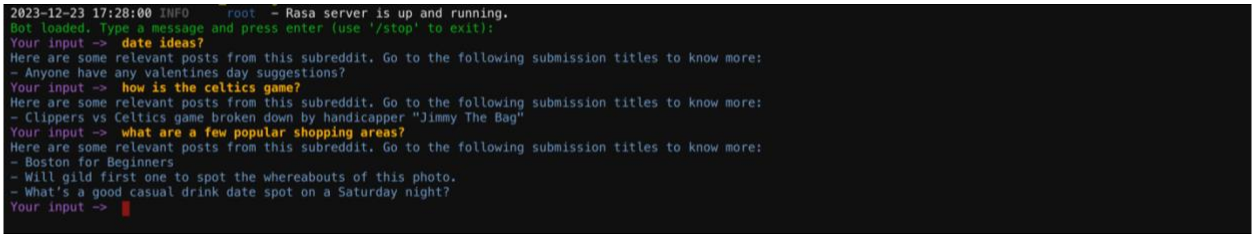


```
OSError: [Errno 98] error while attempting to bind on address ('0.0.0.0', 5005): address already in use
(newenv) [saisurya@scc1 rasa]$ lsof -i :5005
COMMAND    PID     USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
rasa    640439 saisurya   12u  IPv4 136046836      0t0  TCP *:avt-profile-2 (LISTEN)
(newenv) [saisurya@scc1 rasa]$ kill -9 640439
(newenv) [saisurya@scc1 rasa]$ lsof -i :5005
(newenv) [saisurya@scc1 rasa]$ rasa shell --model 20231221-195102-formal-play.tar.gz --endpoints endpoints.yml
/projectnb/cs505ws/students/saisurya/newenv/lib/python3.10/site-packages/rasa/core/tracker_store.py:1044: MovedIn20Warning: Deprecated API features detected! These f
eature(s) are not compatible with SQLAlchemy 2.0. To prevent incompatible upgrades prior to updating applications, ensure requirements files are pinned to "sqlalchem
y<2.0". Set environment variable SQLALCHEMY_WARN_20=1 to show all deprecation warnings.  Set environment variable SQLALCHEMY_SILENCE_UBER_WARNING=1 to silence this m
essage. (Background on SQLAlchemy 2.0 at: https://sqlalche.me/e/b8d9)
  Base: DeclarativeMeta = declarative_base()
/projectnb/cs505ws/students/saisurya/newenv/lib/python3.10/site-packages/rasa/shared/utils/validation.py:134: DeprecationWarning: pkg_resources is deprecated as an A
PI. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  import pkg_resources
/projectnb/cs505ws/students/saisurya/newenv/lib/python3.10/site-packages/pkg_resources/__init__.py:2868: DeprecationWarning: Deprecated call to `pkg_resources.declar
e_namespace('mpl_toolkits')`.
Implementing implicit namespace packages (as specified in PEP 420) is preferred to `pkg_resources.declare_namespace`. See https://setuptools.pypa.io/en/latest/refere
nces/keywords.html#keyword-namespace-packages
  declare_namespace(pkg)
/projectnb/cs505ws/students/saisurya/newenv/lib/python3.10/site-packages/pkg_resources/__init__.py:2868: DeprecationWarning: Deprecated call to `pkg_resources.declar
e_namespace('ruamel')`.
Implementing implicit namespace packages (as specified in PEP 420) is preferred to `pkg_resources.declare_namespace`. See https://setuptools.pypa.io/en/latest/refere
nces/keywords.html#keyword-namespace-packages
  declare_namespace(pkg)
The model metadata does not contain a value for the 'assistant_id' attribute. Check that 'config.yml' file contains a value for the 'assistant_id' key and re-train t
he model.
/projectnb/cs505ws/students/saisurya/newenv/lib/python3.10/site-packages/sanic_cors/extension.py:39: DeprecationWarning: distutils Version classes are deprecated. Us
e packaging.version instead.
  SANIC_VERSION = LooseVersion(sanic_version)
2023-12-23 17:27:48 INFO     root  - Starting Rasa server on http://0.0.0.0:5005
2023-12-23 17:27:48 INFO     rasa.core.processor  - Loading model 20231221-195102-formal-play.tar.gz...
/projectnb/cs505ws/students/saisurya/newenv/lib/python3.10/site-packages/rasa/shared/utils/io.py:99: UserWarning: Training data file /scratch/tmp4wybftv1/domain_prov
ider/domain.yml has a lower format version than your Rasa Open Source installation: 2.0 < 3.1. Rasa Open Source will read the file as a version 3.1 file. Please upda
te your version key to 3.1. See https://rasa.com/docs/rasa/training-data-format.
/projectnb/cs505ws/students/saisurya/newenv/lib/python3.10/site-packages/rasa/core/processor.py:106: UserWarning: The model metadata does not contain a value for the
 'assistant_id' attribute. Check that 'config.yml' file contains a value for the 'assistant_id' key and re-train the model. Failure to do so will result in streaming
 events without a unique assistant identifier.
  rasa.shared.utils.io.raise_warning(
2023-12-23 17:28:00 INFO     root  - Rasa server is up and running.
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->  date ideas?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Anyone have any valentines day suggestions?
Your input ->
```

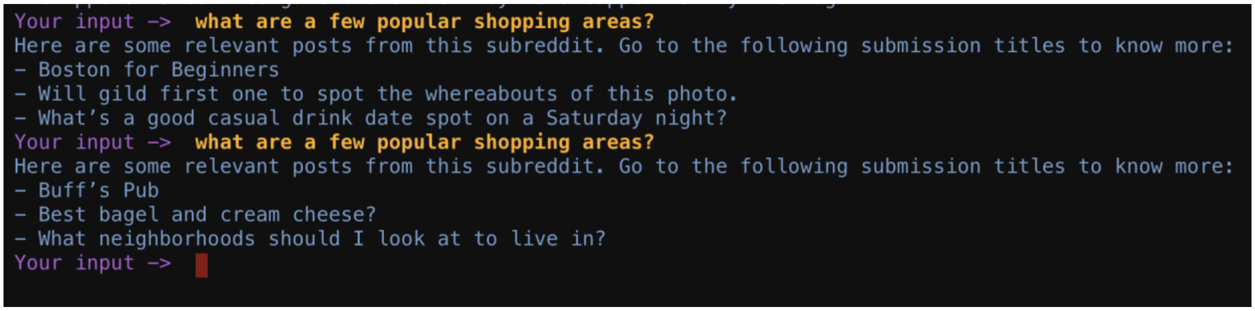
This is the terminal output during the initialization of our RASA-based Chatbot. immediately demonstrates its capability by providing relevant subreddit discussions in response to a user's request for Valentine's Day suggestions. This interaction underscores the bot’s ability to quickly parse the user's input and retrieve pertinent information, highlighting the practical application of our NLP and machine learning pipeline.



```
2023-12-23 17:28:00 INFO     root  - Rasa server is up and running.
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->  date ideas?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Anyone have any valentines day suggestions?
Your input ->  how is the celtics game?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Clippers vs Celtics game broken down by handicapper "Jimmy The Bag"
Your input ->
```

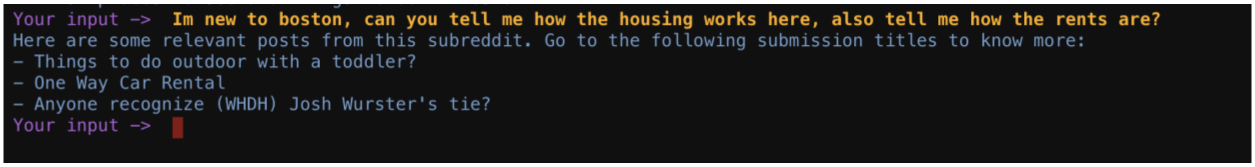


```
2023-12-23 17:28:00 INFO     root  - Rasa server is up and running.
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input ->  date ideas?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Anyone have any valentines day suggestions?
Your input ->  how is the celtics game?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Clippers vs Celtics game broken down by handicapper "Jimmy The Bag"
Your input ->  what are a few popular shopping areas?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Boston for Beginners
- Will gild first one to spot the whereabouts of this photo.
- What's a good casual drink date spot on a Saturday night?
Your input ->
```

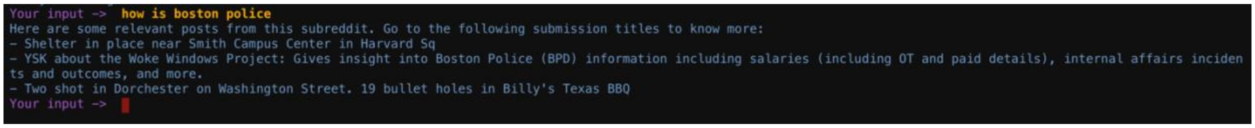


```
Your input ->  what are a few popular shopping areas?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Boston for Beginners
- Will gild first one to spot the whereabouts of this photo.
- What's a good casual drink date spot on a Saturday night?
Your input ->  what are a few popular shopping areas?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Buff's Pub
- Best bagel and cream cheese?
- What neighborhoods should I look at to live in?
Your input ->
```

The ChatBot uses its topic modeling and retrieval algorithms to produce varied results, enhancing the user experience by offering a wider array of information without repetitive answers. This feature exemplifies the bot's sophisticated understanding and its ability to simulate more human-like interactions by avoiding redundant responses.



```
Your input ->  Im new to boston, can you tell me how the housing works here, also tell me how the rents are?
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Things to do outdoor with a toddler?
- One Way Car Rental
- Anyone recognize (WHDH) Josh Wurster's tie?
Your input ->
```



```
Your input ->  how is boston police
Here are some relevant posts from this subreddit. Go to the following submission titles to know more:
- Shelter in place near Smith Campus Center in Harvard Sq
- YSK about the Woke Windows Project: Gives insight into Boston Police (BPD) information including salaries (including OT and paid details), internal affairs inciden
ts and outcomes, and more.
- Two shot in Dorchester on Washington Street. 19 bullet holes in Billy's Texas BBQ
Your input ->
```

3. Discussion of Findings

The results obtained offer significant insights into the nature of discussions within the Boston subreddit. They underscore the potential of NLP tools in extracting meaningful patterns from vast, unstructured datasets. The success of the ChatBot in navigating users to appropriate content further demonstrates the practical applicability of our approach.

The computational limitations did pose constraints, particularly in model selection and processing speed. However, these limitations also guided us towards more efficient solutions like the "all-MiniLM-L6-v2" embedding model.

4. Challenges Encountered

Data preprocessing emerged as a significant challenge, especially given the dataset's size and complexity. The process of cleaning and structuring the data was intricate but crucial for the success of the project.

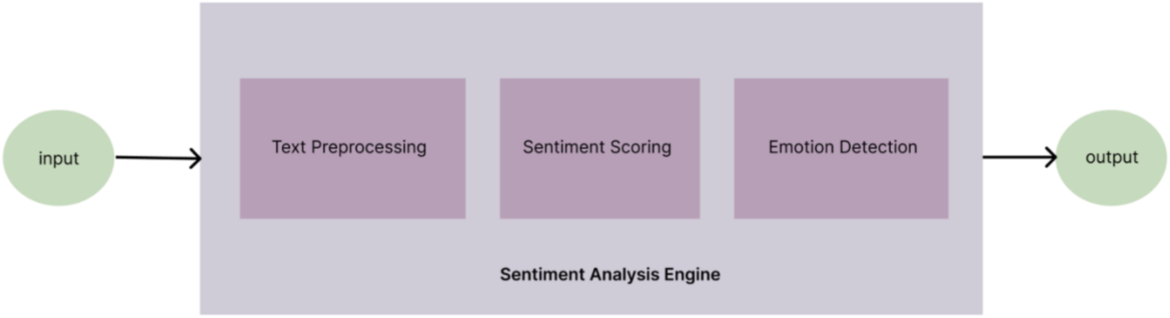
In terms of model performance, while BERTopic provided robust topic modeling, the choice of embedding models was critical to balance efficiency and effectiveness. The RASA ChatBot, on the other hand, required iterative training to fine-tune its responsiveness and accuracy.

5. Future Directions and Improvements

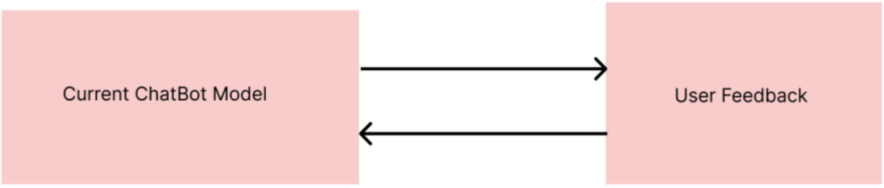
Looking ahead, there are several avenues for enhancing this project:

- Integrating sentiment analysis to offer a more nuanced understanding of subreddit discussions.
- Expanding the dataset to include more subreddits for a broader analysis.
- Improving the ChatBot's algorithm to handle a wider array of user queries with greater contextual understanding.

We have outlined a few conceptual diagrams of how we could implement the above enhancements.



Sentiment Analysis Integration



Feedback Loop for ChatBot Optimization

The insights derived from this project hold potential applications beyond the realm of Reddit. They could inform content moderation strategies, community management practices, and even marketing approaches in similar online platform.

Contributions:

Sai Surya Varshith: He was instrumental in the initial phase of data acquisition, where he dedicatedly sourced and downloaded the dataset suitable for our project's scope and computational resources. He proficiently handled the conversion of .zst compressed files into accessible .csv formats, laying the groundwork for subsequent analyses. In a joint effort, we successfully mapped comments to their respective submissions, ensuring the integrity of the dataset. Varshith played a pivotal role in refining the topic modeling process, skillfully fine-tuning the BERTopic model to condense the initial 500 topics to a more manageable number. His initiative in proposing the ChatBot implementation significantly shaped the project's direction, and he actively contributed to the exploration and customization of the RASA model.

Vaishnavi Vadlamudi: She brought her expertise to the forefront in the text preprocessing stage, ensuring the data was primed for topic modeling. Her exploration of the BERTopic and LLaMA models was key in generating insightful topics for the selected subreddit. Vaishnavi's strategic planning was crucial in developing the project's pipeline, ensuring a seamless transition from data preprocessing to model deployment. Her role extended to the comprehensive drafting of the final report, encapsulating the project's findings and methodologies with clarity and precision.

References:

1. M. M. Tadesse, H. Lin, B. Xu and L. Yang, "Detection of Depression-Related Posts in Reddit Social Media Forum," in IEEE Access, vol. 7, pp. 44883-44893, 2019, doi: 10.1109/ACCESS.2019.2909180.

2. Inamdar, S., Chapekar, R., Gite, S. et al. Machine Learning Driven Mental Stress Detection on Reddit Posts Using Natural Language Processing. Hum-Cent Intell Syst 3, 80–91 (2023). https://doi.org/10.1007/s44230-023-00020-8
3. Automate Sentiment Analysis Process for Reddit Post: TextBlob and VADER, Manmohan Singh.
4. Subreddit_analysis on Github, Matteo Hoch