

San José State University
Department of Applied Data Science

DATA 225: Database Systems for Analytics
Instructor: Simon Shim

Lab-2 Report
ChatGPT Tweets Analysis
Group 4

Group Members:

Tiffany Li
Pranavi Avula
Keerthana Raskatla
Jinthy Swetha Mamillapalli
Deva Kumar Gajulamandyam

Sl. No.	Table of Contents
1.	Problem Statement
2.	Solution Requirement
3.	Limitations
4.	Conceptual Database Design
5.	De-Normalization
6.	Document Structure
7.	Functional Analysis
8.	MongoDB Queries <ul style="list-style-type: none">● MQL Queries● Access Privileges
9.	Visualization in MongoDB Atlas
10.	NoSQL Performance Measurement

ChatGPT Tweets Analysis

I. PROBLEM STATEMENT

Since its debut, ChatGPT has spread significantly throughout every corner of the globe and across a variety of businesses, particularly the technology industry. A lot of businesses, from little startups to multibillion-dollar conglomerates, are expanding their personnel in an effort to create a competitive strategy that would mimic the functioning of this product. The general public responded unevenly to this trend, with some seeing it as a threat to many sectors of the workforce since it replaces people and results in significant global unemployment and others seeing it as a significant step forward in the development of Artificial General Intelligence.

In order to evaluate and generate a full report on ChatGPT and how to estimate its public appeal, firms must create strategies to compete with it. It is an intimidating task to complete all of this quickly while minimizing revenue loss. One efficient and unbiased way to do this is to use sentiment analysis of tweets about ChatGPT that were published by members of the general public. This ChatGPT Tweets analysis is an analytical application system that we have developed. To make it simple for users to identify and follow information that interests them, hashtags help aggregate Tweets and conversations around related topics. Therefore, all the profiles and public posts that include a particular hashtag can be found by clicking on it or searching for it. It is necessary to learn more about individuals who use the hashtag #ChatGPT in their tweets. To do this, we must evaluate the data, save the important details, and generate analytical conclusions as soon as possible. Based on the findings, businesses may make wise decisions, target their products more precisely, and possibly minimize the amount of time needed to bring them to market.

II. SOLUTION REQUIREMENT

- This system is built for companies to help them monitor twitter, one of the largest and most active social media platforms, for competitor analysis.
- The business would be able analyze audience preferences, prospective clients, modern trends and threats.
- The system would enable them to learn the number of tweets associated with a specific hashtag.
- In addition, the number of tweets from a user regarding a specific topic, the time zone, country, and languages that are most engaged in a specific tweet subject, and

the most well-liked individuals associated with the topic of interest (those with the highest like/retweet counts).

- The system would also give the company the ability to examine how its rivals are faring, what their plans are, and what the company's counterstrategies would be.
- This would also extend out into the modern era of marketing, impacting audiences and boosting revenue for companies.
- The business can separate nation- and language-specific relationships and tactics because it uses language and location as filters.

III. LIMITATIONS

- This application does not have access to user's personal information such as mobile number, email, location/geographical coordinates.
- Users have no ability to update or remove their old tweets using the application.
- The supplied dataset is limited to tweets from the 22nd to the 24th of January, but further data can be added as and when needed.
- The user's credentials are not currently being saved. Consequently, there is no method to recover a lost account.

IV. CONCEPTUAL DATABASE DESIGN

We have a dataset containing data and metadata of tweets posted in various languages with wide variety of hashtags concentrated on a single topic ChatGPT. Based on the initial analysis and requirements, we have defined a non-relational document based schema for this application using MongoDB.

Our chatgpt database has single Collection. It has several fields, arrays and embedded documents. The fields of our **user_tweets** collection inside **chatgpt** database are defined as follows:

1. **Username** (STRING) - This field represents the unique user name of each user. It is in string format.
2. **UserURL** (STRING) - This is a string field containing the URL for the user's twitter profile.
3. **Tweets** (ARRAY) - Tweets field is an embedded document containing array of objects. Each object contains data and metadata associated to a single tweet and the array contains details of all the tweets posted by the user. Each field within the object of tweet array is described below.

- a) **Datetime** (DATETIME) – Timestamp when the tweet was posted on twitter. MongoDB internally stores this in ISODate format.
- b) **tweet_id** (LONG INT) – This is a unique identifier for each tweet auto generated by twitter. It is of type long int.
- c) **Text** (STRING) - This field contains the content of the tweet and supports multiple languages.
- d) **Permalink** (STRING) - This field contains the URL of the tweet.
- e) **Outlinks** (ARRAY) – This contains list of all direct links attached in the tweet. Each URL is stored as a string.
- f) **Countlinks** (ARRAY) – This field is also an array of strings and is same as outlinks but has shortened links for click analytics.
- g) **ReplyCount** (INTEGER) – This integer field shows the number of direct replies for a given tweet
- h) **RetweetCount** (INTEGER) – Represents the number of retweets for a given tweet.
- i) **LikeCount** (INTEGER) – This field contains the count of likes for corresponding tweet.
- j) **QuoteCount** (INTEGER) – This field represents number of times a tweet has been quoted by other users.
- k) **ConversationId** (LONG INT) – It stores the tweet_id of parent tweet if the given tweet is a reply to another tweet.
- l) **Language** (STRING)– This field contains 2 letter ISO 639-2 language code, based on the language used in that particular tweet.
- m) **Source** (STRING) – This field refers to the platform or application used for posting that particular tweet. For example, Android, iPhone, LinkedIn etc..
- n) **Media** (STRING) – This field describes any media content that is attached to a tweet such as Image, Video or GIFs in the form of string
- o) **QuotedTweet** (STRING) – It contains the URL of parent tweet in string format if the given tweet is quoted from another tweet.
- p) **MentionedUsers** (ARRAY) – This field contains usernames of users mentioned in the tweet as an array.
- q) **Hashtags** (ARRAY) – It consists of array of all the hashtags used in the given tweet.
- r) **hashtagCount** (INTEGER) – This field represents the number of hashtags used in the given tweet.

V. DE-NORMALIZATION

The original dataset, csv file was actually in denormalized form. Unlike in SQL, here we don't have to normalize our database in NoSQL for efficient querying. NoSQL does not have join concept. Rather, it uses documents and embedded documents for faster querying and stores data in JSON like format and does not enforce any constraints. However, we have slightly modified the structure of schema to organize the data properly and load it in database collection. We preprocessed the dataset using python to comply with our approach and used a single collection for our application.

VI. DOCUMENT STRUCTURE

According to the Conceptual Design, we have a single Collection in our database and it comprises of several Embedded documents inside along with nested arrays. Below is the pictorial representation of our Document Structure.

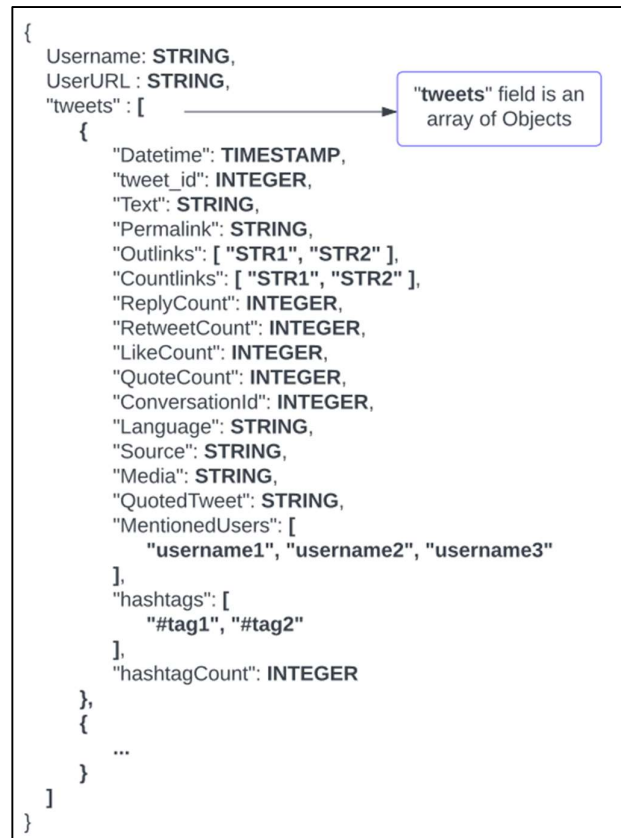


Fig.1 – Document Structure

VII. FUNCTIONAL ANALYSIS

The database is designed for a company that wants to develop strategies based on tweets. The database has the following functional components:

DATA:

Data is the information that needs to be changed and processed in order to have any real meaning. To hold the names and attributes of the data elements that are being used in this, a database dictionary is required. A database maintains metadata, which is information that describes the data. This facilitates the management and storage of data inside a database.

The dataset which is being referred and used for analysis is the ChatGPT Twitter Dataset is an open Data from Kaggle. We have used this dataset to create our own database in cloud using MongoDB. This database is capable of autoscaling based on the traffic and has cross region compatibility. It has 3 shards by defaults – 1 primary and 2 secondary in case of a disaster.

When a twitter user tweets about something along with ‘ChatGPT’, it will be recorded in our system. We record the Date and time, Tweet Id assigned to the tweet, the text and user name. We even record any external links attached in the tweet, the replies received by the tweet, number of times it was retweeted, number of likes received by the tweet, number of times the tweet has been quoted by other users in a conversation, The language in which the tweet was written and from which kind of device it has been made (be it an iPhone or laptop or android), the links to all the quotes and original tweet. We also record any mentions of other people made by the user in that tweet along with the hashtags and their counts.

This would give us the essential data required to build our Entities.

Using this data, we build a database model, which would help the company access the following information

We give the company:

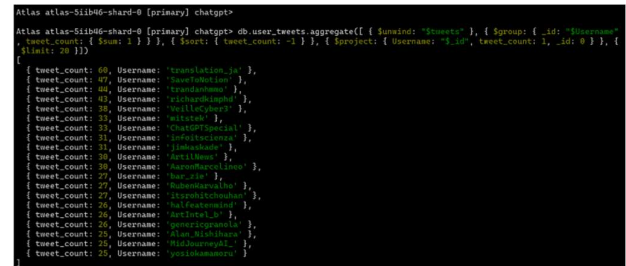
- user engagement rate, which entails how much influence a user has over others in a discussion. (This would help the company find potential influencers to advertise their product).
- Retweet link, mentioning how many users have attached a retweet (external) link to their tweet. (To get information about other systems or competitors).
- The top ten people who have tweeted the most in a discussion.
- How many times is person (usually a celebrity) has been mentioned (usually to know which person’s endorsement can help or hurt their product the most).
- Users who are most active on twitter regarding the ChatGPT topic.
- The tweet which has got most attention.
- Hangtags used prominently along with ChatGPT.
- The most famous tweet. (Can be used for analysis about the product’s potential customer)

VIII.MONGO DB QUERIES

A. MQL Queries

[1] Query to get users(top 20) with most tweets

```
db.user_tweets.aggregate([
  { $unwind: "$tweets" },
  { $group: { _id: "$Username", tweet_count: { $sum: 1 } } },
  { $sort: { tweet_count: -1 } },
  { $project: { Username: "$_id", tweet_count: 1, _id: 0 } },
  { $limit: 20 }
])
```

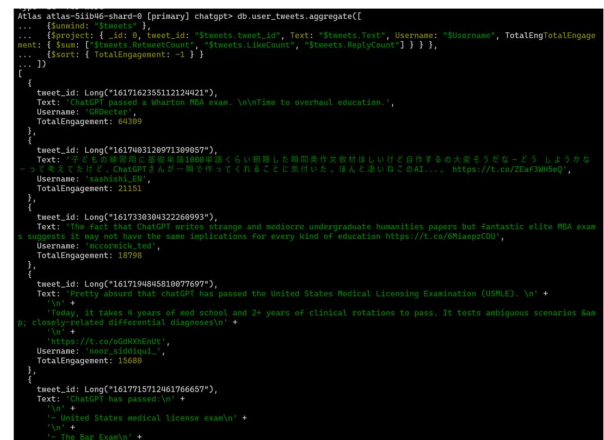


The screenshot shows the output of the MongoDB query in the Atlas interface. It lists 20 users with their usernames and tweet counts, sorted in descending order. The first user is 'translating_ja' with 69 tweets, followed by 'tactobian' with 60 tweets, and so on, down to 'psikamamto' with 25 tweets.

Fig.2 – Query 1 Output

[2] Query to get most viral tweets (wrt retweets)

```
db.user_tweets.aggregate([
  { $unwind: "$tweets" },
  { $project: { _id: 0, tweet_id: "$tweets.tweet_id", Text:
"$tweets.Text", Username: "$Username",
TotalEngagement: { $sum: ["$tweets.RetweetCount",
"$tweets.LikeCount", "$tweets.ReplyCount" ] } } },
  { $sort: { TotalEngagement: -1 } }
])
```



The screenshot shows the output of the MongoDB query in the Atlas interface. It lists 20 tweets with their tweet IDs, text, usernames, and total engagement (sum of retweets, likes, and replies). The first tweet is from 'tactobian' with a tweet ID of '1617162355112124421' and a total engagement of 60369. The second tweet is from 'tactobian' with a tweet ID of '1617483126971399657' and a total engagement of 21151.

Fig.3 – Query 2 Output

[3] Query to fetch top 20 most used hashtags used in tweets

```
db.user_tweets.aggregate([
  { $unwind: "$tweets" },
```

```
{ $match: { "tweets.hashtagCount": { $gt: 0 } } },
{ $project: { hashtags: "$tweets.hashtags" } }, {
$unwind: "$hashtags" },
{ $group: { _id: "$hashtags", count: { $sum: 1 } } },
{ $sort: { count: -1 } },
{ $limit: 20 }
}
```

```
Atlas atlas-5iib46-shard-0 [primary] chatgpt> db.user_tweets.aggregate([ { $unwind: "$tweets" }, { $match: {
"tweets.hashtagCount": { $gt: 0 } } }, { $project: { hashtags: "$tweets.hashtags" } }, { $unwind: "$hashtags"
}, { $group: { _id: "$hashtags", count: { $sum: 1 } } }, { $sort: { count: -1 } }, { $limit: 20 } ] ]
{
  "_id": "#chatgpt", "count": 9462 },
  { "_id": "#ai", "count": 2543 },
  { "_id": "#openai", "count": 1011 },
  { "_id": "#artificialintelligence", "count": 723 },
  { "_id": "#microsoft", "count": 597 },
  { "_id": "#chatgpt", "count": 311 },
  { "_id": "#technology", "count": 303 },
  { "_id": "#ia", "count": 283 },
  { "_id": "#google", "count": 277 },
  { "_id": "#machinelearning", "count": 258 },
  { "_id": "#tech", "count": 237 },
  { "_id": "#chatgpt", "count": 217 },
  { "_id": "#chatgpt3", "count": 205 },
  { "_id": "#cybersecurity", "count": 149 },
  { "_id": "#chatbot", "count": 148 },
  { "_id": "#openaiengine", "count": 136 },
  { "_id": "#crypto", "count": 135 },
  { "_id": "#gpt3", "count": 131 },
  { "_id": "#education", "count": 125 },
  { "_id": "#deeplearning", "count": 122 }
}
```

Fig.4 – Query 3 Output

[4] Query to find tweets that have at least 50 retweets and have at least one user mentioned

```
db.user_tweets.find(
{"tweets.RetweetCount": {"$gt": 50},
"tweets.MentionedUsers": {"$ne": [ ]}},
{"Username": 1, _id: 0, "tweets.tweet_id": 1,
"tweets.RetweetCount": 1, "tweets.MentionedUsers":
1}).sort({"tweets.RetweetCount" : -1 })
```

```
Atlas atlas-5iib46-shard-0 [primary] chatgpt> db.user_tweets.find({"tweets.RetweetCount": {"$gt": 50}, "tweets.MentionedUsers": {"$ne": [ ]}}, {"Username": 1, _id: 0, "tweets.tweet_id": 1, "tweets.RetweetCount": 1, "tweets.MentionedUsers": 1}).sort({"tweets.RetweetCount" : -1 })
{
  Username: 'Techie_Kid097s',
  tweets: [
    {
      tweet_id: Long('1617403892152819713'),
      RetweetCount: 170,
      MentionedUsers: [ 'hey_waliet' ]
    }
  ],
  Username: 'RSDT',
  tweets: [
    {
      tweet_id: Long('1617349810910265344'),
      RetweetCount: 110,
      MentionedUsers: [ 'juanandres_ga', 'alperowitch' ]
    }
  ],
  Username: 'kelseyhighower',
  tweets: [
    {
      tweet_id: Long('1617283297394208976'),
      RetweetCount: 9,
      MentionedUsers: [ 'Grady_Boach' ]
    }
  ],
  Username: 'JenColamester',
  tweets: [
    {
      tweet_id: Long('1617516164019638274'),
      RetweetCount: 70,

```

Fig.5 – Query 4 Output

[5] Query to fetch top platforms(android, iphone,etc) used for chatgpt discussions/tweets with #chatgpt

```
db.user_tweets.aggregate([
{ $unwind: "$tweets" },
{ $unwind: "$tweets.hashtags" },
{ $match: { "tweets.hashtags": "#chatgpt" } },
{ $group: { _id: "$tweets.Source", count: { $sum: 1 } } },
{ $sort: { count: -1 } },
{ $limit: 20 },
])
```

```
{ $project: { _id: 0, Source: "$_id", count: 1 } }
])
```

```
Atlas atlas-5iib46-shard-0 [primary] chatgpt> db.user_tweets.aggregate([ { $unwind: "$tweets" }, { $unwind: "$
tweets.hashtags" }, { $match: { "tweets.hashtags": "#chatgpt" } }, { $group: { _id: "$tweets.Source", count:
{ $sum: 1 } } }, { $sort: { count: -1 } }, { $limit: 20 }, { $project: { _id: 0, Source: "$_id", count: 1 } }
])
{
  count: 3990, Source: 'Twitter Web App' },
  { count: 1998, Source: 'Twitter for iPhone' },
  { count: 1737, Source: 'Twitter for Android' },
  { count: 181, Source: 'LinkedIn' },
  { count: 179, Source: 'Buffer' },
  { count: 172, Source: 'TweetDeck' },
  { count: 153, Source: 'Twitter for iPad' },
  { count: 131, Source: 'Hootsuite Inc.' },
  { count: 46, Source: 'trancebar' },
  { count: 44, Source: 'IFTTT' },
  { count: 31, Source: 'dLive.it' },
  { count: 30, Source: 'Sprout Social' },
  { count: 29, Source: 'Jaxpack.com' },
  { count: 26, Source: 'HubSpot' },
  { count: 24, Source: 'Artful AI' },
  { count: 23, Source: 'TweetDeck Web App' },
  { count: 22, Source: 'BlogSocial APP' },
}
```

Fig.6 – Query 5 Output

[6] Query to find out number of tweets based on language used

```
db.user_tweets.aggregate([
{ $unwind: "$tweets" },
{ $group: { _id: "$tweets.Language", TweetCount:
{ $sum: 1 } } },
{ $project: { _id: 0, Language: "$_id", TweetCount: 1 } },
{ $sort: { TweetCount: -1 } }
])
```

```
Atlas atlas-5iib46-shard-0 [primary] chatgpt> db.user_tweets.aggregate([
... { $unwind: "$tweets" },
... { $group: { _id: "$tweets.Language", TweetCount: { $sum: 1 } } },
... { $project: { _id: 0, Language: "$_id", TweetCount: 1 } },
... { $sort: { TweetCount: -1 } }
... ])
{
  TweetCount: 32076, Language: 'en' },
  { TweetCount: 5046, Language: 'ja' },
  { TweetCount: 3315, Language: 'es' },
  { TweetCount: 2492, Language: 'fr' },
  { TweetCount: 1207, Language: 'de' },
  { TweetCount: 1175, Language: 'pt' },
  { TweetCount: 443, Language: 'it' },
  { TweetCount: 436, Language: 'tr' },
  { TweetCount: 423, Language: 'und' },
  { TweetCount: 395, Language: 'gme' },
  { TweetCount: 392, Language: 'ar' },
  { TweetCount: 319, Language: 'nl' },
  { TweetCount: 251, Language: 'in' },
  { TweetCount: 193, Language: 'th' },
  { TweetCount: 187, Language: 'fa' },
  { TweetCount: 181, Language: 'ru' },
  { TweetCount: 149, Language: 'zh' },
  { TweetCount: 141, Language: 'ko' },
  { TweetCount: 113, Language: 'iw' },
  { TweetCount: 113, Language: 'ca' }
}
```

Fig.7 – Query 6 Output

[7] Query to fetch top users with highest average like count

```
db.user_tweets.aggregate([
{ "$unwind": "$tweets" },
{ "$group": { "_id": "$Username", "avg_likes_per_tweet":
{ "$avg": "$tweets.LikeCount" } } },
{ $sort: { avg_likes_per_tweet: -1 } }
])
```

```

Atlas atlas-51816b-sha0 [primary] chatgpt> db.user_tweets.aggregate([
...   { '$sumind': '$tweets' },
...   { '$group': { '_id': '$username', 'avg_likes_per_tweet': { '$avg': '$tweetsLikeCount' } } },
...   { '$sort': { 'avg_likes_per_tweet': -1 } }
... ])
[
  { '_id': '@GDMceter', 'avg_likes_per_tweet': 20861 },
  { '_id': '@Watchdogsw', 'avg_likes_per_tweet': 18836.5 },
  { '_id': '@Veski1', 'avg_likes_per_tweet': 9125 },
  { '_id': '@mccormick_ted', 'avg_likes_per_tweet': 8468.5 },
  { '_id': '@sashibh123', 'avg_likes_per_tweet': 6666666666.6667 },
  { '_id': '@disclosetv', 'avg_likes_per_tweet': 5911 },
  { '_id': '@hasantox', 'avg_likes_per_tweet': 5682 },
  { '_id': '@Vittitoack', 'avg_likes_per_tweet': 5513 },
  { '_id': '@ChristopherM1', 'avg_likes_per_tweet': 4833 },
  { '_id': '@vmoeham', 'avg_likes_per_tweet': 4643 },
  { '_id': '@Pampiani', 'avg_likes_per_tweet': 4492 },
  { '_id': '@Grindmarse', 'avg_likes_per_tweet': 4413 },
  { '_id': '@stare_teez', 'avg_likes_per_tweet': 3915 },
  { '_id': '@Fitfounder', 'avg_likes_per_tweet': 3732 },
  { '_id': '@MikeScully', 'avg_likes_per_tweet': 3529 },
  { '_id': '@michelpark', 'avg_likes_per_tweet': 3509 },
  { '_id': '@moor_siddiqui', 'avg_likes_per_tweet': 3322.5 },
  { '_id': '@vunderground', 'avg_likes_per_tweet': 2776 },
  { '_id': '@patrickbetdavid', 'avg_likes_per_tweet': 2184 },
  { '_id': '@Firechip_dev', 'avg_likes_per_tweet': 1936 }
]

```

Fig.8 – Query 7 Output

[8] Query to check most popular web links(Outlinks) posted in the tweet based on count of occurrence

```
db.user_tweets.aggregate([
  { $unwind: "$tweets" },
  { $unwind: "$tweets.Outlinks" },
  { $group: { _id: "$tweets.Outlinks", "count": { $sum: 1 } } },
  { $project: { Outlink: "$_id", count: 1, _id: 0 } },
  { $sort: { count: -1 } },
  { $limit: 10 }
])
```

```

Atlas atlas-811b6-share@primary chatgpt> db user.te tweets.aggregate(
  {
    $unwind: "$source",
    $match: { $or: [ {source: "news.bbc.com"},
    { $group: { _id: "news.bbc.com", $count: { $sum: 1 } } },
    { $group: { _id: "news.bbc.com", $count: 1, _id: 0 } } } ],
    $sort: { $count: -1 } },
    { $limit: 10 }
  },
  {
    }
  }
}

{
  count: 196,
  Outlink: "https://www.ft.com/content/7229b6b8-1a2d-49f6-9e2f-f8c7933a97e",
}

{
  count: 182,
  Outlink: "https://twitter.com/news_sdd6ql/status/1617194045818077897",
}

{
  count: 112,
  Outlink: "https://twitter.com/90Wector/status/1617162355112124621",
}

{
  count: 98,
  Outlink: "https://twitter.com/gubector/status/1617162355112124621",
}

{
  count: 87, Outlink: "http://rukeynews.co.uk",
}

{
  count: 77,
  Outlink: "https://fortune.com/2023/01/21/chatgpt-passed-wharton-mba-exam-one-professor-is-sounding-alarm-artificial-intelligence/",
}

{
  count: 75,
  Outlink: "https://www.bloomberg.co.jp/news/articles/2023-01-21/NOZ790ZK2P18",
}

{
  count: 71,
  Outlink: "https://www.mediaviv.org/content/18.1101/2022.12.19.22268b2d",
}

```

Fig.9 – Query 8 Output

[9] Query to fetch all the tweets that contain a specific hashtag

```
db.user_tweets.aggregate([
  {$unwind: "$tweets"},
  {$match: {"tweets.hashtags": "#chatsonic"}},
  {$project: {_id: 0, Username: 1, "tweets.Datetime": 1,
    "tweets.Text": 1, "tweets.hashtags": 1}}
])
```

```
Atlas atlas-s1ibug-shard-0 [primary] chatgpt> db.user.tweets.aggregate([
...   { $unwind: '$tweets' },
...   { $match: { 'tweets.hashtags': '#chatsonic' } },
...   { $project: { '_id': 0, 'Username': 1, 'tweets.Timestamp': 1, 'tweets.Text': 1, 'tweets.Hashtags': 1 }}
... ])
{
  Username: 'PoliticsWatchin',
  tweets: [
    {
      Timestamp: ISODate('2023-01-27T15:59:06.000Z'),
      Text: 'Colleges and Universities are now scrambling in panic by the introduction of ChatGPT &amp; unusu
&#x2D;ly how they respond to students using it to write their assignments as part of continuous assessment. An int
&#x2D;resting experience #artificialintelligence #chatgpt #chatsonic',
      hashtags: [ 'artificialintelligence', 'chatgpt', 'chatsonic' ]
    }
  ],
  Username: 'R_CodigoRojo',
  tweets: [
    {
      Timestamp: ISODate('2023-01-22T20:33:28.000Z'),
      Text: '&#xA0;El hashtager&#xA0;hay otro mucho mejor ya que ChatGPT se llama Chatsonic',
      hashtags: [ 'chatgpt', 'chatsonic' ]
    }
  ],
  Username: '@BizzBuzzNews',
  tweets: [
    {
      Timestamp: ISODate('2023-01-23T05:52:12.000Z'),
      Text: '&#xA0;&#xA0;Deep Chatsonic are some alternatives of ChatGPT that you can try #ChatGPT #ChatSonic #DeepWR
&#xA0;ite #DeepWR&#xA0; amplify generate stream https://a.co/bw0z1dVw',
      hashtags: [
        'chatgpt',
        'chatsonic',
        'deepwr',
        'amplify',
        'generate',
        'stream',
        'https'
      ]
    }
  ],
  Username: 'noc_3370',
```

Fig.10 – Query 9 Output

[10] Query is for counting the number of likes that were associated with each hashtag in all the tweets.

```
db.user_tweets.aggregate([
  { $unwind: "$tweets" },
  { $unwind: "$tweets.hashtags" },
  { $group: { _id: "$tweets.hashtags", TotalLikes: { $sum:
"$tweets.LikeCount" } } },
  { $project: { _id: 0, hashtag: "$_id", TotalLikes: 1 } },
  { $sort: { TotalLikes: -1 } }
])
```

```

Atlas atlas-slabus-shard-0 [primary] chatopt> db.user tweets.aggregate([{$sumind: '$tweetid'},{$sumind: '$userid'},{$group: {'_id': '$tweetid', 'TotalLikes': {$sum: '$tweets.LineCount'}}},{$project: {'_id': 0, 'hashtag': '$_id', 'TotalLikes': 1}}, {$sort: {'TotalLikes': -1}}])
{
  {
    TotalLikes: 36089, hashtag: '#chatopt' },
    {
    TotalLikes: 6316, hashtag: '#ai' },
    {
    TotalLikes: 3337, hashtag: '#chatgpt' },
    {
    TotalLikes: 3153, hashtag: '#microsoft' },
    {
    TotalLikes: 2613, hashtag: '#openai' },
    {
    TotalLikes: 1386, hashtag: '#it' },
    {
    TotalLikes: 1304, hashtag: '#aijourney' },
    {
    TotalLikes: 1189, hashtag: '#中国人工智能 2' },
    {
    TotalLikes: 1183, hashtag: '#artificialintelligence' },
    {
    TotalLikes: 1104, hashtag: '#chatgpt_1' },
    {
    TotalLikes: 92, hashtag: '#tech' },
    {
    TotalLikes: 799, hashtag: '#edchat' },
    {
    TotalLikes: 715, hashtag: '#teitchbook' },
    {
    TotalLikes: 702, hashtag: '#establishfusion' },
    {
    TotalLikes: 683, hashtag: '#achievethelearning' },
    {
    TotalLikes: 640, hashtag: '#ai' },
    {
    TotalLikes: 619, hashtag: '#edcoach' },
    {
    TotalLikes: 616, hashtag: '#edcoaches' },
    {
    TotalLikes: 606, hashtag: '#googleedu' },
    {
    TotalLikes: 596, hashtag: '#edcelebrates' }
  }
}

```

Fig.11 – Query 10 Output

[11] Query to get the tweet count based on the language

```
db.user_tweets.aggregate([
  { $unwind: "$tweets" },
  { $group: { _id: { $dateToString: { format: "%Y-%m-%d", date: "$tweets.Datetime" } }, count: { $sum: 1 } } },
  { $sort: { "_id": 1 } }
])
```

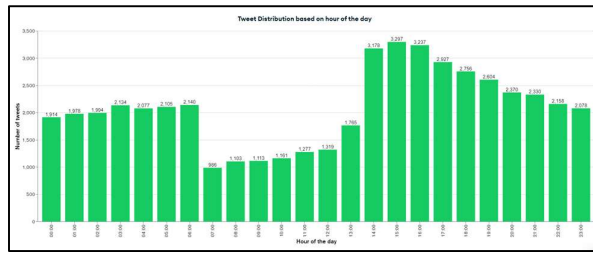



Fig. 17 – Tweet Distribution based on hour of day

- Donut Chart representing most used platforms for tweeting.

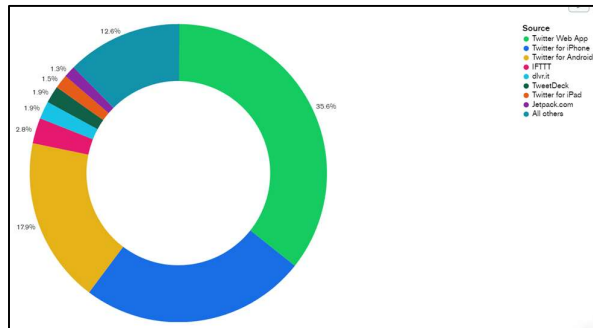


Fig. 18 – Most frequently used platforms for tweeting

- Visual Representation of most used languages in tweets.

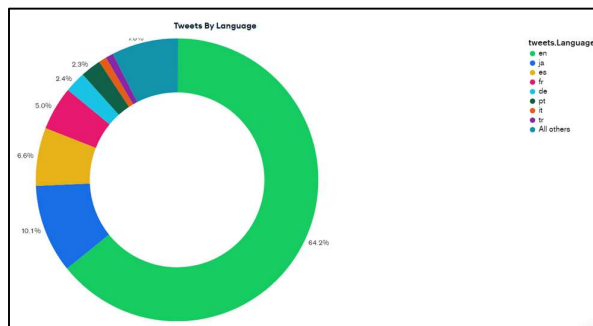


Fig. 19 – Donut Chart of most used language

- Stacked Bar Chart indicating most viral tweet with respect to likes, retweets, replies and quotes.

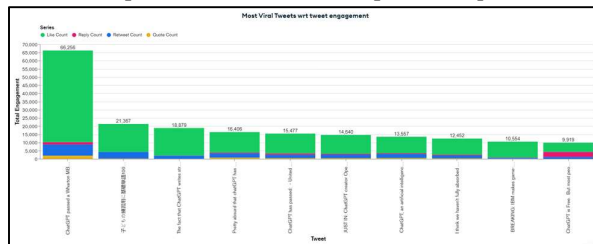


Fig. 20 – Most Viral Tweets

- Visualization showing average and maximum likes per tweet for each user.

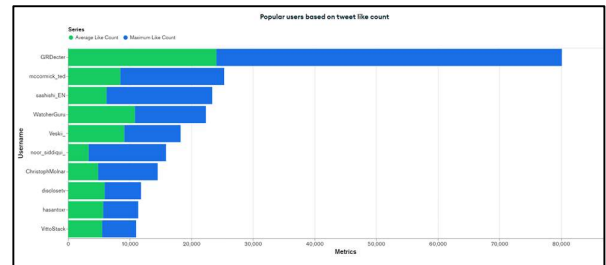


Fig. 21 – Average and Max like count per tweet for each user

X. NOSQL PERFORMANCE MEASUREMENT

The performance is one of the main aspect to keep in mind while designing a database application. The main advantage of using NoSQL database is its performance in comparison to SQL type databases which take considerable time for large datasets to perform complex joins. In NoSQL, we store data in json format and hence can query faster even on large scale data. This provides better latency which proves to be crucial in minimizing loss for businesses and improve revenue.

MongoDB provides several options to evaluate performance of querying from Compass or through CLI. We have used explain() method with “executionStats” argument to evaluate query statistics and database engine execution. This provides detailed reports of the query execution plan, stages, pipelines, server info etc.

Below is a query that fetches likes associated to each hashtag. When executed in both MySQL and MongoDB, the latter gave faster response as it uses document based storage whereas MySQL takes more time to perform the complex join operations required for this query.

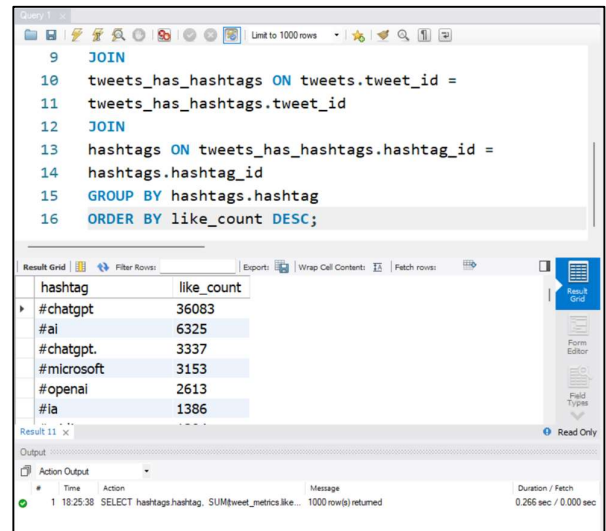


Fig. 22 – Query Execution in MySQL (266 ms)

```

Atlas atlas-sib06-shard-0 [primary] chatgpt> db.user_tweets.aggregate([ { $unwind: '$tweets' }, { $unwind: '$
tweets.hashtags' }, { $group: { _id: '$tweets.hashtags', TotalLikes: { $sum: '$tweets.LikeCount' } } }, { $pro
ject: { _id: 0, hashtag: '$_id', TotalLikes: 1 } }, { $sort: { TotalLikes: -1 } } ]).explain('executionStats')
{
  explainVersion: '1',
  stages: [
    {
      '$cursor': {
        queryPlanner: {
          namespace: '661dcd807d29c352efc88ed_chatgpt.user_tweets',
          indexFilterSet: false,
          parsedQuery: {},
          queryHash: '6AFC06F',
          planCacheKey: '6AFC06F',
          maxIndexedOrSolutionsReached: false,
          maxIndexedAndSolutionsReached: false,
          maxScansToExplodeReached: false,
          winningPlan: {
            stage: 'PROJECTION_SIMPLE',
            transformBy: { tweets: 1, _id: 0 },
            inputStage: { stage: 'COLLSCAN', direction: 'forward' }
          },
          rejectedPlans: []
        },
        executionStats: {
          executionSuccess: true,
          nReturned: 38433,
          executionTimeMillis: 152,
          totalKeysExamined: 0,
          totalDocsExamined: 38433,
          executionStages: [
            {
              stage: 'PROJECTION_SIMPLE',
              nReturned: 38433,
              executionTimeMillisEstimate: 6,
              works: 38435,
              advanced: 38433,
              needTime: 1,
              needYield: 0,
            }
          ]
        }
      }
    ]
  }
}

```

Fig. 23 – Query Execution in MongoDB (152 ms)

In the above output, *executionTimeMillis* represents the total time taken to execute the query in milli seconds. It also provides estimated time taken during each stage of execution.

To improve the efficiency further we can create index on specific fields to reduce the number of documents to examine. Another advantage of NoSQL is that it natively supports Horizontal scaling and cross region replication. Overall, NoSQL outperforms SQL based storage systems as the data increases or unstructured data gets introduced thereby reducing latency.