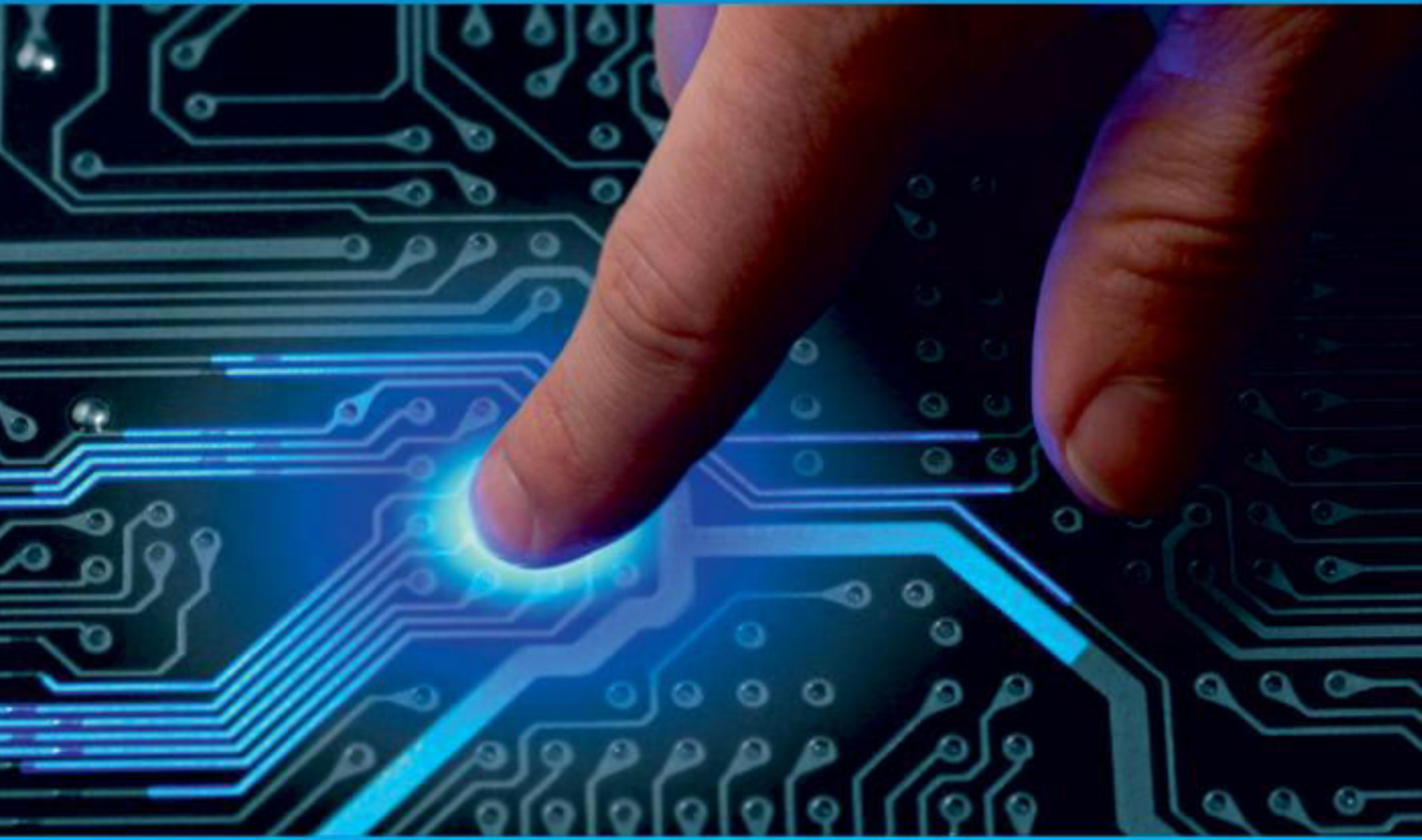
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 6, June 2021

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 6, June 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0906247 |

Knowledge Extraction from Unstructured Data using Scrapy

Mr. N. Brahma Naidu[1], **K. Bhavya Choudary**[2], **G. Swetha**[3], **G. Manvitha Bhavani**[4], **K. Naveen Kumar** [5]

Assistant Professor, Department of Computer Science and Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India[1]

UG Student, Department of Computer Science and Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India [2,3,4,5]

ABSTRACT
Frequent pattern mining, with the aim of finding information in the form of repetitive patterns, is an imperative data mining activity. In the last two decades, many powerful pattern mining algorithms have been identified, but most do not scale to the sort of data we are presenting today, the so-called "Big Data." In this case, scalable parallel algorithms hold the key to resolving the problem. This paper reviews recent developments in regular pattern mining in tandem, examining them through the prism of Big Data. The main problems to be overcome are load balancing and task partitioning. When Big Data grows without boundaries, these problems often evoke groundbreaking approaches to do so. Conquering unstructured data to identify frequent trends is the greatest obstacle as before. Semi-Structured Doc-Model and trend rankings are used to do this.

KEY WORDS: Data Mining, Doc-Model, Frequent Pattern Mining, Pattern Rank, Unstructured Data

I. INTRODUCTION

Frequent Pattern Mining is an observational method that finds frequent patterns, correlations, or causal constructs from data sets contained in different kinds of databases such as relational databases, transactional databases, and other repositories. Given a series of transactions, this process aims to find the rules that allow us to predict the occurrence of a particular item based on the occurrence of other items in the transaction. This pattern mining is often performed on Big Data that is in an unstructured format. The aim of the project is to find frequent patterns in unstructured data based on the user needs and providing ranks to the patterns.
We get data from the website and store it in the MongoDB server in this approach. Since we're receiving data from a website, we'll need an internet link to store the web information in the archive. Following the storage of content in the MongoDB registry, the data must be retrieved from the archive. The data is retrieved in this case depending on the user's interests. When a user specifies his or her search criteria for a specific word, the word is checked in the database. The count of each word in each text in each database set is then returned in decreasing order of count.

II. REVIEW OF LITERATURE

Web scraping was addressed by Bo Zhao [13] in his article. This paper describes the methods for extracting data from the network that are relevant to our project. Web scraping is a method of extracting information from the internet and saving it to a file server or archive for later retrieval or analysis. Scraping the web is often referred to as web extraction or harvesting.
In their article, Rupali Arora and Rinkle Rani Aggarwal explored document databases and compared NoSQL and SQL queries. In contrast to relational databases, which store data as normalized relational tables, text databases store data as records. These documents' data formats include JSON, BSON, and XML. Collections are where documents

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 6, June 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0906247 |

are kept. The relational counterpart of document and set records (tuple) and relational equivalent of relational equivalent of relational equivalent of relational equivalent of relational equivalent of relational equivalent of (table). In his article, Niteshwar Datt Bhardwaj compared and contrasted two NoSQL document-oriented databases. The first is MongoDB, and the second is RavenDB. We choose MongoDB over the other document-oriented database models because there are so many of them. The value of MongoDB over RavenDB is discussed in this article. Document-oriented NoSQL databases are distributed in both open source and licensed versions.

III. PROPOSED SOLUTION

We have two phases to complete in our proposed solution.

- Data Extraction
- Frequent Pattern Mining and Pattern Ranking
- Graphical Representation

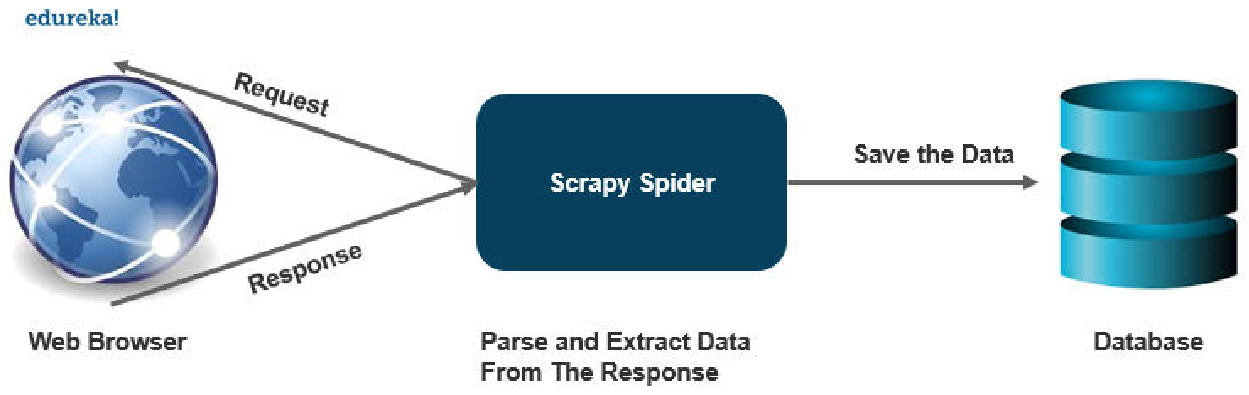


Fig 1: Flow of Proposed Solution

The data must be retrieved from the chosen websites in the first phase. As seen in Fig 1, the data is extracted from the website using HTML tag content that can be maintained using web scraping framework like Scrapy. These methods are used to collect data from the contents of HTML tags. Data must be stored in a database after being extracted from web pages using Scrapy. Here we extract the data of staff members from the website https://www.vvitguntur.com/

Data must be saved in the MongoDB registry after it has been retrieved. The data is organized into categories in the database, and the collections contain documents. The database is given a name, and it contains a number of different sets with different names. Each set includes a variety of documents organized by key-value pairs that include all of the information about each staff member. We will access data from the database based on our use of data after the data has been retrieved from the site and placed in the database. In this section, we retrieve the most common patterns found in each set. This popular pattern mining is based on a word search combined with user preferences. For example, if I try to scan the database for the word "data," it will look at every collection and every record within each collection, retrieving data in descending order depending on the word count field of the searching word.

If we use an online cluster to store the databases and tables, then we can use MongoDB charts to generate different and unique graphs like bar chart, pie chart etc. Which makes it easy for the users to understand the data. The program for searching for a particular word runs continuously until the user exits from the program. When the user starts running the program, the loop runs continuously where each time it asks the user to enter the word he wants to search for. If the user gives the word ‘exit’ as the key then the loop breaks and the program ends. Otherwise, if he types for a keyword then it searches for the word in every document in all the collections in the database. This helps the user to compare all the departments staff members based on their experience in each topic. The flow chart representation is,

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 6, June 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0906247 |

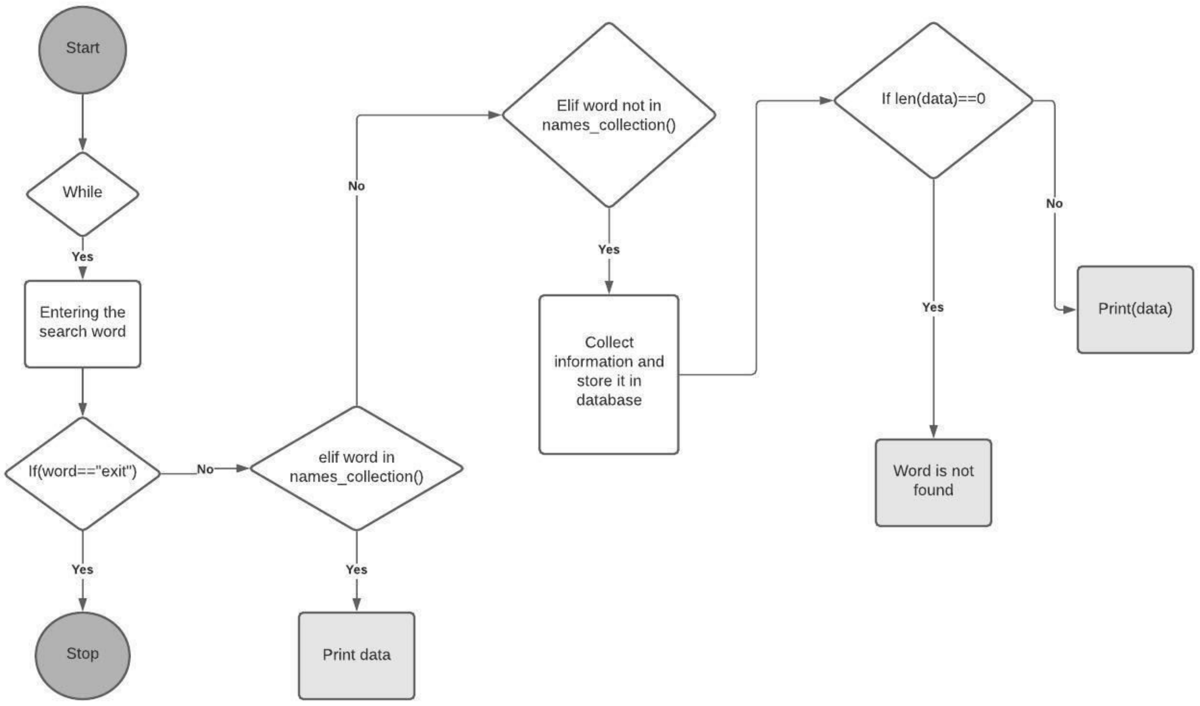
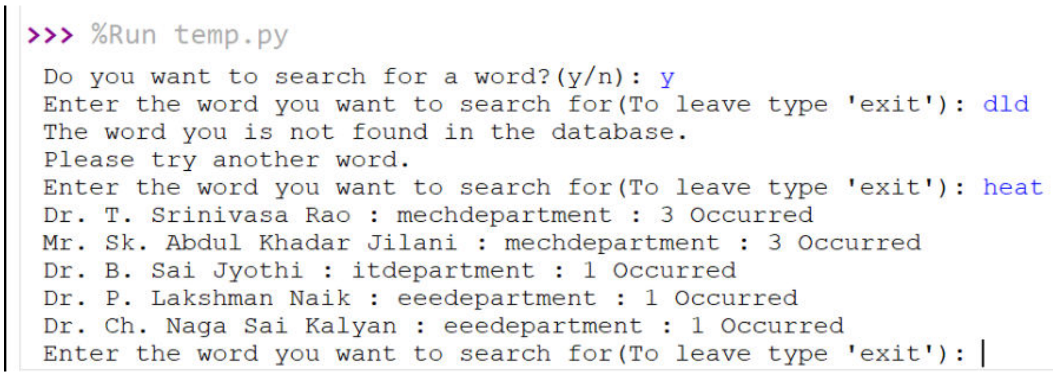


Fig 2: *Flow chart*

IV. RESULTS



```
>>> %Run temp.py
 Do you want to search for a word?(y/n): y
 Enter the word you want to search for(To leave type 'exit'): dld
 The word you is not found in the database.
 Please try another word.
 Enter the word you want to search for(To leave type 'exit'): heat
 Dr. T. Srinivasa Rao : mechdepartment : 3 Occurred
 Mr. Sk. Abdul Khadar Jilani : mechdepartment : 3 Occurred
 Dr. B. Sai Jyothi : itdepartment : 1 Occurred
 Dr. P. Lakshman Naik : eeedepartment : 1 Occurred
 Dr. Ch. Naga Sai Kalyan : eeedepartment : 1 Occurred
 Enter the word you want to search for(To leave type 'exit'):
```

Fig 3: *Output when typed "heat"*

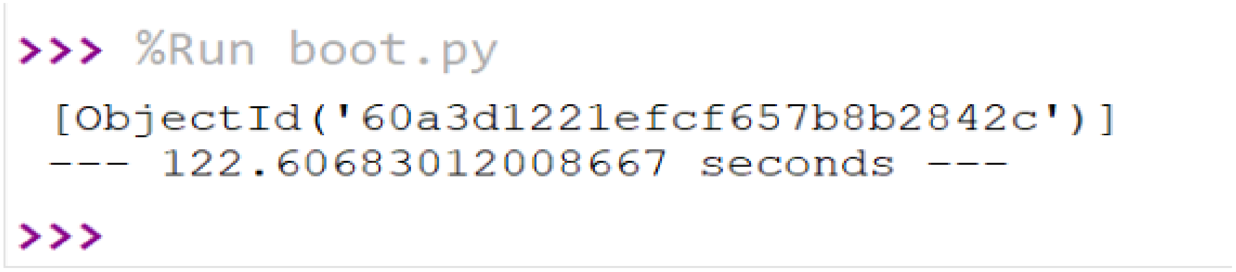
When made to run the output is as shown in Fig 3. As we are finding the frequent patterns and pattern ranking when the user is trying to search for a word the program checks all the departments database and stores the retrieved data in the words database with the user given name for the word. Where the name of the word the user has given is stored as the collection name in the words database. Each collection in this database contains three fields one is the name of the staff member; another is the department where the member is working and count of the referred word of the user. Each and every member contains a number at the starting of the line in the output where it shows the ranking of the pattern.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542 |

|| Volume 9, Issue 6, June 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0906247 |

TIME OF EXECUTION:


```
>>> %Run boot.py
 [ObjectId('60a3d1221efcf657b8b2842c')]
 --- 122.60683012008667 seconds ---

>>>
```

Fig 4: *Object Creation time using Beautiful Soup*

Only for a single object in BeautifulSoup the execution time is 122* seconds.

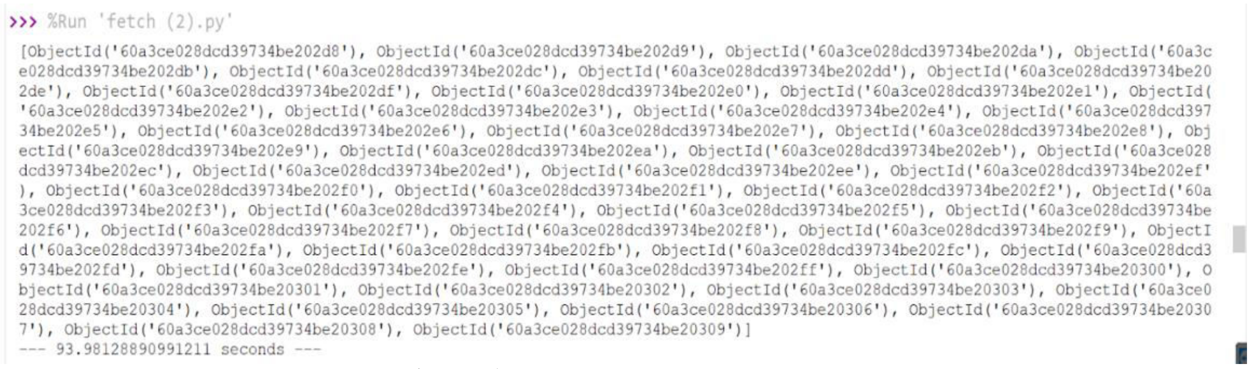

```
>>> %Run 'fetch (2).py'
 [ObjectId('60a3ce028dcd39734be202d8'), ObjectId('60a3ce028dcd39734be202d9'), ObjectId('60a3ce028dcd39734be202da'), ObjectId('60a3c
 e028dcd39734be202db'), ObjectId('60a3ce028dcd39734be202dc'), ObjectId('60a3ce028dcd39734be202dd'), ObjectId('60a3ce028dcd39734be20
 2de'), ObjectId('60a3ce028dcd39734be202df'), ObjectId('60a3ce028dcd39734be202e0'), ObjectId('60a3ce028dcd39734be202e1'), ObjectId(
 '60a3ce028dcd39734be202e2'), ObjectId('60a3ce028dcd39734be202e3'), ObjectId('60a3ce028dcd39734be202e4'), ObjectId('60a3ce028dcd397
 34be202e5'), ObjectId('60a3ce028dcd39734be202e6'), ObjectId('60a3ce028dcd39734be202e7'), ObjectId('60a3ce028dcd39734be202e8'), Obj
 ectId('60a3ce028dcd39734be202e9'), ObjectId('60a3ce028dcd39734be202ea'), ObjectId('60a3ce028dcd39734be202eb'), ObjectId('60a3ce028
 dcd39734be202ec'), ObjectId('60a3ce028dcd39734be202ed'), ObjectId('60a3ce028dcd39734be202ee'), ObjectId('60a3ce028dcd39734be202ef'
 ), ObjectId('60a3ce028dcd39734be202f0'), ObjectId('60a3ce028dcd39734be202f1'), ObjectId('60a3ce028dcd39734be202f2'), ObjectId('60a
 3ce028dcd39734be202f3'), ObjectId('60a3ce028dcd39734be202f4'), ObjectId('60a3ce028dcd39734be202f5'), ObjectId('60a3ce028dcd39734be
 202f6'), ObjectId('60a3ce028dcd39734be202f7'), ObjectId('60a3ce028dcd39734be202f8'), ObjectId('60a3ce028dcd39734be202f9'), ObjectI
 d('60a3ce028dcd39734be202fa'), ObjectId('60a3ce028dcd39734be202fb'), ObjectId('60a3ce028dcd39734be202fc'), ObjectId('60a3ce028dcd3
 9734be202fd'), ObjectId('60a3ce028dcd39734be202fe'), ObjectId('60a3ce028dcd39734be202ff'), ObjectId('60a3ce028dcd39734be20300'), O
 bjectId('60a3ce028dcd39734be20301'), ObjectId('60a3ce028dcd39734be20302'), ObjectId('60a3ce028dcd39734be20303'), ObjectId('60a3ce0
 28dcd39734be20304'), ObjectId('60a3ce028dcd39734be20305'), ObjectId('60a3ce028dcd39734be20306'), ObjectId('60a3ce028dcd39734be2030
 7'), ObjectId('60a3ce028dcd39734be20308'), ObjectId('60a3ce028dcd39734be20309')]
 --- 93.98128890991211 seconds ---
```

Fig 5: *Object Creation time using Scrapy*

But when we use Scrapy for many objects, the time taken is 93* seconds only. **Here we are trying to demonstrate the efficiency of using Scrapy.**

GRAPHS:

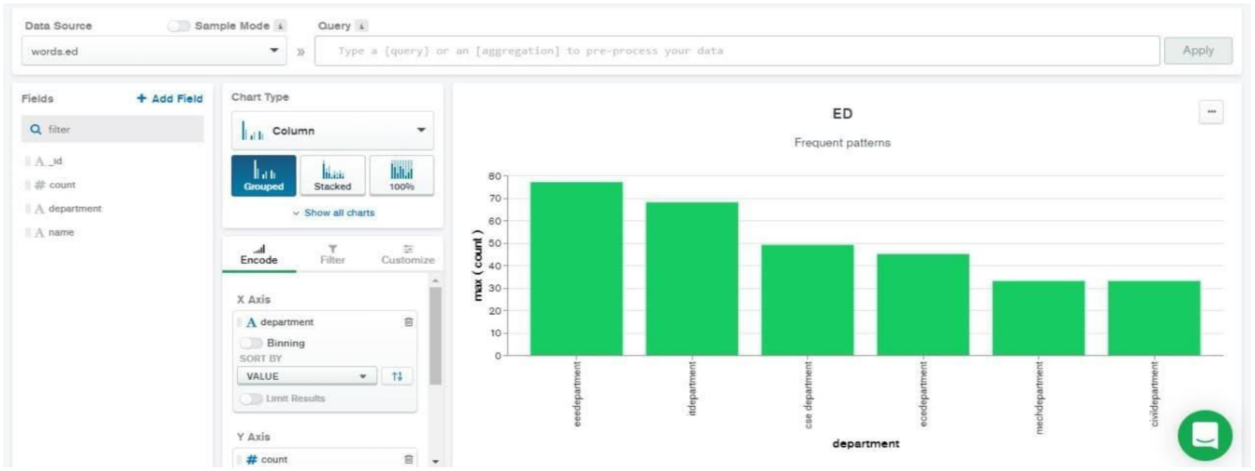
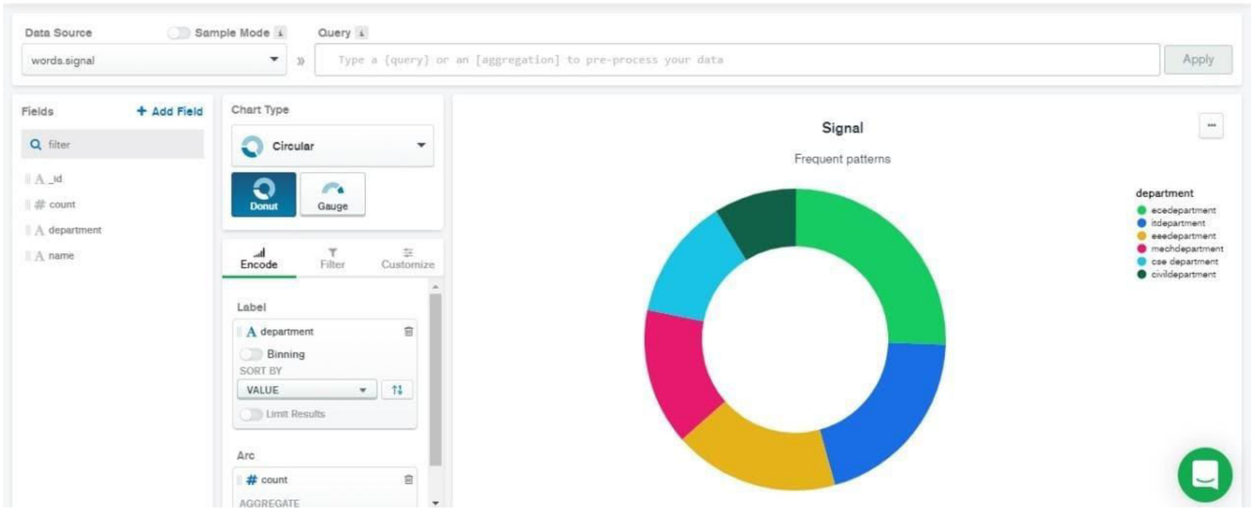


Fig 6: *BarChart of the word "ED"*



Data Source: words.signal

Query: Type a {query} or an [aggregation] to pre-process your data

Fields: _id, count, department, name

Chart Type: Circular

Label: department
Binning
SORT BY: VALUE
Limit Results

Arc: count
AGGREGATE

department
- 🟢 ecedepartment
- 🔵 itdepartment
- 🟡 eeedepartment
- 🔴 mechdepartment
- 🔵 cse department
- 🟢 civildepartment

Fig 7: *PieChart of the word "Signal"*

V. CONCLUSION AND FUTURE SCOPE

We presented a general strategy for rating frequent patterns in relation to the data that is both accurate and repeatable. This makes it easier to interpret and retrieve data from the website. As unstructured data is turned to semi-structured data, this makes it easier for the user to retrieve the information. Because unstructured data contains irregularities and ambiguities that make retrieving data difficult for the user, the data is converted to semi-structured data, which removes some of the irregularities and ambiguities. This aids pharmaceutical companies, educational institutions, finance, and marketing departments, among others, in locating the data they seek. This project can be utilized and customized to meet the demands of the user.

REFERENCES

1. Researchgate: ttps://www.researchgate.net/publication/283236813_Performance_Analysis_of_NoSQL_Databases
2. Rupali Arora, Rinkle Rani Aggarwal, Modeling and Querying Data in MongoDB, International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013
3. Bo Zhao, Web Scraping, Springer International Publishing AG (outside the USA) 2017L.A. Schintler, C.L. McNeely (eds.), Encyclopedia of Big Data, DOI 10.1007/978-3-319-32001-4_483-1
4. Sai Jyothi Bolla and Jyothi S, Doc-Based Modelling for Medical Big Data, 2018 IADS International Conference on Computing, Communications & Data Engineering (CCODE) 7-8 February
5. Niteshwar Datt Bhardwaj, Comparative Study of CouchDB and MongoDB –NoSQL Document Oriented Databases, International Journal of Computer Applications (0975 – 8887) Volume 136 – No.3, February 2016

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING