MICROBLOG BASED PERSONALIZED NEWS RECOMMENDATION USING HYBRID APPROACH

A Project Report

submitted by

MOULI R **2017103555**

LOGESH J **2017103550**

KARTHIKEYAN M **2017103541**

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY, CHENNAI 25

MAY 2021

ANNA UNIVERSITY, CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report titled **" MICROBLOG BASED PERSONALIZED NEWS RECOMMENDATION USING HYBRID APPROACH "** is the bonafide work of " **MOULI R (2017103555), LOGESH J (2017103550)** and **KARTHIKEYAN M (2017103541) "** who carried out the project work under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering.

Place: Chennai
Date:

SIGNATURE

Dr. S. Renugadevi
Assistant Professor,
Department of Computer Science and Engineering,
Anna University,
Chennai – 600 025

COUNTERSIGNED

SIGNATURE

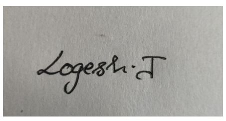
Dr. Valli S
Head of the Department,
Department of Computer Science and Engineering,
Anna University, Chennai – 600 025.

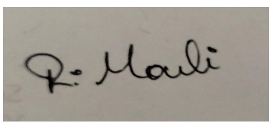
ACKNOWLEDGEMENT

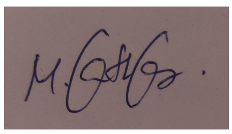
We express our deep gratitude to our guide, Dr. S. Renugadevi,Assistant Professor, Department of Computer Science and Engineering, for guiding us through every phase of the project. We appreciate her thoroughness,tolerance and ability to share his knowledge with us. Apart from adding her own input, she has encouraged us to think on our own and give form to our thoughts. We owe her for harnessing our potential and bringing out the best in us.

We are extremely grateful to Dr. S. Valli, Professor and Head of the Department of Computer Science and Engineering, Anna University, Chennai - 25, for extending the facilities of the Department towards our project and for her unstinting support.

We express our thanks to the panel of reviewers Dr.S. Bose, Professor,Department of Computer Science and Engineering and Dr. V. Mary AnitaRajam, Professor, Department of Computer Science and Engineering, for their valuable suggestions and critical reviews throughout the course of our project.We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.







. Logesh J Mouli R Karthikeyan M

ABSTRACT

Recommender systems are built to help us to easily find the most proper information on the internet. Unlike the search engines, recommender systems bring the information to the user without any manual search effort. This is achieved by using the similarities between users and/or items.A personalised news recommendation system collects news from multiple press releases and presents the recommended news to the user.

In this work we propose a news recommendation model combines collaborative filtering-based and content-based filtering methods. The existing models works only on the existence of historical user news read behaviour of the users which leads to the inability of the system to make recommendations for a new user due to the unavailability of the historical read behaviour of the user.

The proposed model aims in solving the cold start issue by collecting the user information from microblogging sites such as twitter and make use of that information in addressing the cold start issue and make recommendations for the new users.At the same time the model aims in improving the accuracy and diversity score

of the existing model by undertaking certain changes in the existing model.

திட்டப்பணி சுருக்கம்

இணையத்தில் மிகவும் சரியான தகவல்களை எளிதில் கண்டுபிடிக்க எங்களுக்கு உதவும் வகையில் பரிந்துரை அமைப்புகள் கட்டடப்பட்டுள்ளன. தேடுபொறிகள் போலன்றி, பரிந்துரைக்கும் அமைப்புகள் எந்த ஒரு கையேடு தேடலும் இல்லாமல் பயனருக்கு தகவலைக் கொண்டு வருகின்றன. பயனர்களுக்கும் / அல்லது உருப்படிகளுக்கும் இடையிலான ஒற்றுமையை பயன்படுத்துவதன் மூலம் இது அடையப்படுகிறது. தனிப்பயனாக்கப்பட்ட செய்தி பரிந்துரை அமைப்பு பல செய்தி வெளியீடுகளிலிருந்து செய்திகளைச் சேகரித்து பரிந்துரைக்கப்பட்ட செய்திகளை பயனருக்கு அளிக்கிறது.

இந்த வேலையில் ஒரு செய்தி பரிந்துரை மாதிரி கூட்டு வடிகட்டுதல் அடிப்படையிலான மற்றும் உள்ளடக்க அடிப்படையிலான வடிகட்டுதல் முறைகளை ஒருங்கிணைக்கிறது. தற்போதுள்ள மாதிரிகள் பயனர்களின் வரலாற்று பயனர் செய்தி வாசிப்பு நடத்தை இருப்பதில் மட்டுமே செயல்படுகின்றன, இது பயனரின் வரலாற்று வாசிப்பு நடத்தை கிடைக்காததால் புதிய பயனருக்கான பரிந்துரைகளை செய்ய கணினியின் இயலாமைக்கு வழிவகுக்கிறது.

முன்மொழியப்பட்ட மாதிரியானது ட்விட்டர் போன்ற மைக்ரோ பிளாக்கிங் தளங்களிலிருந்து பயனர் தகவல்களைச் சேகரிப்பதன் மூலம் குளிர் தொடக்க சிக்கலைத் தீர்ப்பதை நோக்கமாகக் கொண்டுள்ளது மற்றும் குளிர் தொடக்க சிக்கலைத் தீர்ப்பதில்

அந்தத் தகவலைப் பயன்படுத்துவதோடு புதிய பயனர்களுக்கான பரிந்துரைகளையும் செய்யுங்கள்.அதே நேரத்தில், மாதிரி சில மேம்பாடுகளைச் செய்வதன் மூலம் அசல் மாதிரியின் நிலைத்தன்மையையும் பன்முகத்தன்மை மதிப்பையும் மேம்படுத்த முயல்கிறது.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

OC	Ordered Clustering
CF	Collaborative Filtering
CB	Content Based Filtering
FLANN	Fast Library For Approximate Nearest Neighbours
HR	Hotness Rate
RC	News Recency
STP	Short Term Profile
NR Matrix	News Read Matrix
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

CHAPTER 1

INTRODUCTION

This chapter gives an outline of the forms related to recommendation systems and news recommendations. This work elaborates the challenges that are to be considered in the development of the proposed system.

1.1 RECOMMENDATION SYSTEMS

Recommender Systems or recommendation engines form or work from a specific type of information filtering system technique that attempts to recommend information items that are likely to be of interest to the user.Typically, a recommender system compares a userprofile to some reference characteristics, and seeks to predict the 'rating' that a user would give to an item they had not yet considered by the user.

News perusing has changed with the progress of the World Wide Web, from the conventional demonstration of news utilization by means of physical daily paper membership to getting to thousands of sources by means of the web. News websites, like Google News and Yahoo! News, collect news from different sources and give a total view of news from around the world. A basic issue with news benefit websites is that the volumes of articles can be overpowering to the clients. The challenge is to assist clients discover news articles that are curiously to read.

1.1.1 Collaborative Filtering Methods

Collaborative methods for recommender systems are methods that are based solely on the past interactions recorded between users and items in order to produce new recommendations. These interactions are stored in the so-called "user-item interactions matrix"

The main advantage of collaborative approaches is that they require no information about users or items and, so, they can be used in many situations. Moreover, the more users interact with items the more new recommendations become accurate: for a fixed set of users and items, new interactions recorded over time bring new information and make the system more and more effective.

1.1.2 Content-based filtering

Another common approach when designing recommender systems is content-based filtering. Content-based filtering methods are based on a description of the item and a profile of the user's preferences. These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user. Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on an item's features.

1.1.3 Hybrid Recommender Systems

Most recommender systems now use a hybrid approach, combining collaborative filtering, content-based filtering, and other approaches. There is no reason why several different techniques of the same type could not be hybridized. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model.

These methods can also be used to overcome some of the common problems in recommender systems such as cold start and the sparsity problem.

1.2 PROBLEM STATEMENT

A news item is particular in nature and is diverse from other items to recommend.A news item may have to be placed in more than one news category. Apart from that, a news item has a short lifetime and may expire in a little term of time.Recency is the foremost commonly utilized property to decide a news lifetime based on the time span of the first time the news is distributed. Another property is popularity, which appears the number of times a news thing is read by the clients all through its lifetime.

The cold-start issue is one of the major problems in all recommendation systems based on collaborative filtering. The issue raises when a new client joins the system and doesn’t have any historical information, there's no data about the client to recommend items.

The cold-start problem becomes more seriously within the news domain because new users’ visit, after an event has happened or users who sometimes visit news locales based on the expected news articles to be published online. It is additionally referred as first rater, ramp up or early rater problem.

The existing models works only on the existence of historical user news read behaviour of the users which leads to the inability of the system to make recommendations for a new user due to the unavailability of the historical read behaviour of the user.

1.3 OBJECTIVES

The main aim of this research work is to propose a personalised news recommendation system that would attempt to solve the cold start issue in recommendation of news items and at the same time improve the accuracy of the recommendation.

The proposed framework incorporates both the collaborative filtering (CF)-based and content-based filtering methods along the following contributions:

(i) Make use of microblogging sites like twitter to extract user information get the news tweets from the official news handles , get the user read behaviour form information like retweets and use them to build the user read behaviour information.

(ii) Maintain long term user profile and short term profile for the users. The long term profile is used for collaborative filtering while the short term profile is used for content based filtering of users.

(iii) A news metadata model that incorporates ReadingRate and Hotness. And a property, hotnessRate , is used to attain submodularity.

CHAPTER 2

RELATED WORKS

In recent years, there has been much focus on the design and development of personalised news recommendation systems that monitor and learn users' reading behaviours and generate news set based on these behaviours. Common news recom- mendation systems are often based on collaborative filtering (CF), content-based filtering (CB) or in some cases, hybrid methods.

The CF-based news recommendation systems gen- erate personalised recommendations for users based on their behaviours in news reading. In this method, similar users are clustered in a group based on their similarities in news access behavioural patterns. Such behaviours are expressed in the form of binary votes or numerical ratings on each news item. Nonetheless, CF algorithms have difficulty in generating reliable recommendations when data are sparse, and they cannot recommend news items that have no rating from the users, which is often known as cold-start recommendation. Google News[7] , GroupLens , and DRN are examples of CF-based methods.

On the other hand, a content-based news recommendation system recommends news items based on content similarities between the news items and user’s profile. It considers a given user’s reading behaviour and analyses the content of the newly-published news before presenting it to the users. This type of system computes similarity between newly-published news items and the user’s content-based profile and rates them. The news items with high rates are then recommended to the users. However, content-based meth- ods cannot recommend accurately to a new user with low access in news reading.

Aside from the cold-start and data sparseness problems, scalability is one of the major issues in news recommendation that requires elegant algorithms to effectively deal with large news corpus [9], [8]. Several strategies can be used to address the scalability issue such as the MinHash algorithm [9] and clustering.

Several news recommendation frameworks have been proposed in an attempt to increase the recommendation accuracy, overcome the large volume of data, and recommend diverse news items [4], [5], [8], [9] do not make an attempt to filter the number of news items to recommend. These systems recommend the same number of news items to the users, i.e. they are unable to recommend the appropriate number of news items to each user based on the individual user behaviour in news reading.

SCENE[9] employed sub-modularity modelling and exper- imented how news sets can be matched to the users' interests as much as possible while maintaining highest diversity of news. This is achieved by constructing a rich news metadata and user profiles that subsequently affect news selection, hence the accuracy of news recommendation [9]. Overall, news selection requires a new strategy in utilising rich user profiles and news metadata to assist the news recommendation system in achieving accurate and diverse recommendation of news items.

This paper proposes a framework for news recommendation system named MicroBlog based Personalised News Recommendation using Hybrid Approach. This framework is a hybrid recommendation framework, which combines Collaborative Filtering (CF) based technique and Content-based technique. It consists of three components, which are User and News Clustering, News Selection, and Personalised News Recommendation. In the first component, User and News Clustering, news metadata is generated from the newly-published news articles . In order to support this component Ordered Clustering (OC) is used. The second component, News Selection, compares a given user's behaviour to the other similar users and matches the user's profile with the news metadata, to select the recommendable news set. Finally, the third component, Personalised News Recommendation, prioritises and ranks the pruned news articles to recommend the final news set to the user.

CHAPTER 3

REQUIREMENTS ANALYSIS

This chapter discusses the technologies and tools that were employed in the development of this project.

3.1 HARDWARE

The model was implemented , developed and deployed in Lenovo IdeaPad 310 with Intel V Core(TM) i5 -2710 CPU @ 2.65 GHz with 8 GB RAM in Ubuntu 64-Bits platform.

Machine Specifications :

CPU @ 2.65GHz

RAM: 8.00 GB

ROM: 1TB

Graphics Card : Nvidia 820 mx

Operating System : Ubuntu

No other special hardware interface was required/used for the successful implementation of the system.

3.2 SOFTWARE

(i) CELERY

Celery is a simple, flexible, and reliable distributed system to process vast amounts of messages, while providing operations with the tools required to maintain such a system.

It's a task queue with focus on real-time processing, while also supporting task scheduling. Celery is Open Source and licensed under the BSD License.

(ii) PostgreSQL

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

PostgreSQL database is used to store the dataset of the project. This database is chosen because of its reliability , faster query results , performance , robustness and the ability to handle large amounts of data in ease.

(iii) VISUAL STUDIO CODE

In this project the Microsoft visual studio is used as an IDE. Visual Studio Code combines the simplicity of a source code editor with like IntelliSense code completion and debugging. Visual Studio Code supports macOS, Linux, and Windows. With support for hundreds of languages, VS Code contains features like syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, etc. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings helps in easy navigation. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

(iv) REDIS SERVER

Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker. Redis provides data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes, and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions, and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.

In our work the REDIS server is used as a message broker which acts as a broker between the main thread and the celery task scheduler .

(v) TWEEPY

Tweepy is a python class which can provide access to twitter's rest API. Each method can accept various parameters and return responses. This Tweepy is essential for extraction of twitter information that is essential for making recommendations in our system.

This API provides access to a variety of different resources including the following:

- Tweets
- Users
- Retweets
- Lists

- Trends
- Media

(vi) PANDAS

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.

(vii) DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django is used to build the User Interface of the project that is used to display the final recommended news to the user . All the model functions are developed inside a django environment and runs inside that environment.

(viii) FLANN

FLANN is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. It contains a collection of algorithms we found to work best for nearest neighbor search and a system for automatically choosing the best algorithm and optimum parameters depending on the dataset. FLANN is written in C++ and contains bindings for the following languages: C, MATLAB, Python, and Ruby.

(ix) SPACY

spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python. spaCy is designed specifically for production use and helps you build applications that process and "understand" large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning.

CHAPTER 4

SYSTEM DESIGN

This chapter discusses the detailed working of all the modules that are present in the system.The system consists of many modules that are interdependent on each other (output of one module becomes the input of another module) and their functions combine to give a successful model.

4.1 WORKING

The proposed system consists of five modules :

- Tweets and News Data Scraper
- Collaborative Filter Module
- Content Based Filter Module
- Additional Collaborative Module using FLANN
- Personalized News Recommendation

The ‘Tweets and News Data Scraping module’ is the module that performs all the data scraping works like twitter data from twitter using tweepy and scrapping the news data from the news websites using beautifulsoup .All the scraped data are stored into postgreSQL database.Last 30 days tweets of news handles are extracted and the information related to the retweets are taken. Every user who has retweeted for a news tweet is considered to have read the news. This module also includes all the news scraping functionalities the scraps nws from the websites provided the url of the news article.

The main functionality of the Collaborative filter module is to find the user to user similarity between the users. From the news read behaviour that has been obtained in the previous module a news matrix called News-Read Matrix is constructed. The NR matrix is a binary matrix between the users and the news items. A field in the NR matrix corresponding to a news item is 1 if the user has read the news item else if he has not read the news item the field is 0. Now from the NR matrix a user-user similarity matrix is constructed. This user similarity matrix is used to find the user vs user similarity. To find user vs user similarity Jaccard Similarity is used.

From the obtained User similarity matrix Ordered clustering is performed on the users and the users are clustered into various clusters. And from the clusters for a given users the higher order similar users are found. The news read by higher order users are taken and score for the news is calculated based on the similarity score and are sorted based on the score and hotness of the news and the resultant news is the news from Collaborative Filter Module.

The First step in the Content based filtering module is to perform named entity recognition on all the news items using spacy pre-trained model.Named entities are recognised in all the news items and from the entities the users short term profile is created. The user’s short term profile is created by collection of all the entities recognised by the spacy model and the entities are added to the short term profile if the user has read the news which can be known from the NR matrix.The Short term Profile and the news items are then compared and similarity scores are calculated using Cosine Similarity.The resulting similarity scores are used to construct Profile Similarity matrix.This similarity matrix is given as input to the ordered clustering which then clusters users short term profile and news items.

After clustering the user profiles and news items given the particular user all the higher order news relative to the given user's profile are chosen and are considered as content related news to the user. Apart from selecting content related news all the higher order user profiles of the given users are also taken they are considered as similar profiles to the user and news items are chosen from their sides also.

Finally the recommended news items are sorted based upon the similarity scores of the news and the hotness of the news and the resultant news is the news from Collaborative Filter Module.

Personalised News Recommendation modules combines the resultant news of the Collaborative filter module and Content based filter module. The combination of both set of news is controlled by a factor (alpha) which ranges from 0 to 1 which determines the percentage of news to be taken from Collaborative filter part and Content based filter part.These two set of news are combined and prioritised to make final recommendation to the user

Incase a news user uses the system the historical information of the new user is not available ,hence in this case the tweets made by the user is extracted from twitter. The obtained tweets are preprocessed and vectorised using spacy model.

Using the FLANN model all the existing user’s short term profile are plotted in a multi dimensional graph and then the new users explicit profile vectorised form is given as input to the FLANN model to get the nearest neighbour among the existing user. The nearest neighbour is identified and the recommendations of the nearest neighbour is recommended to the new user.

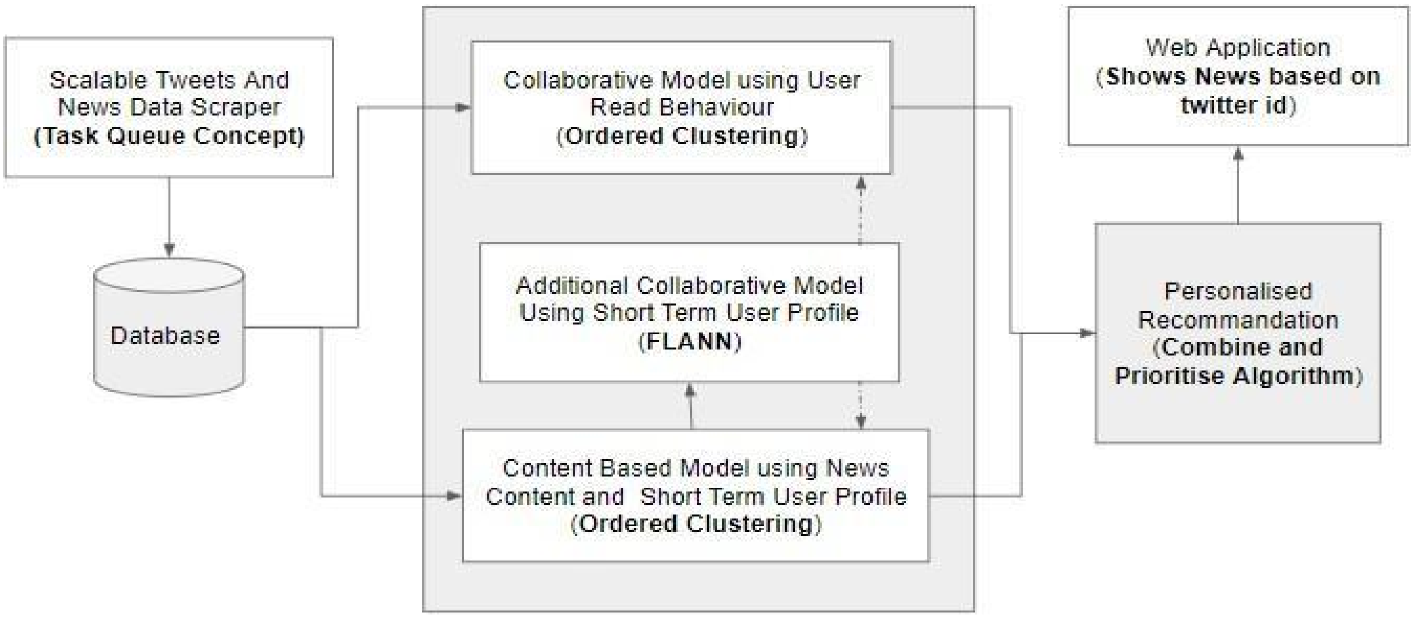


Figure 4.1 Overall Architecture Diagram of the Proposed System

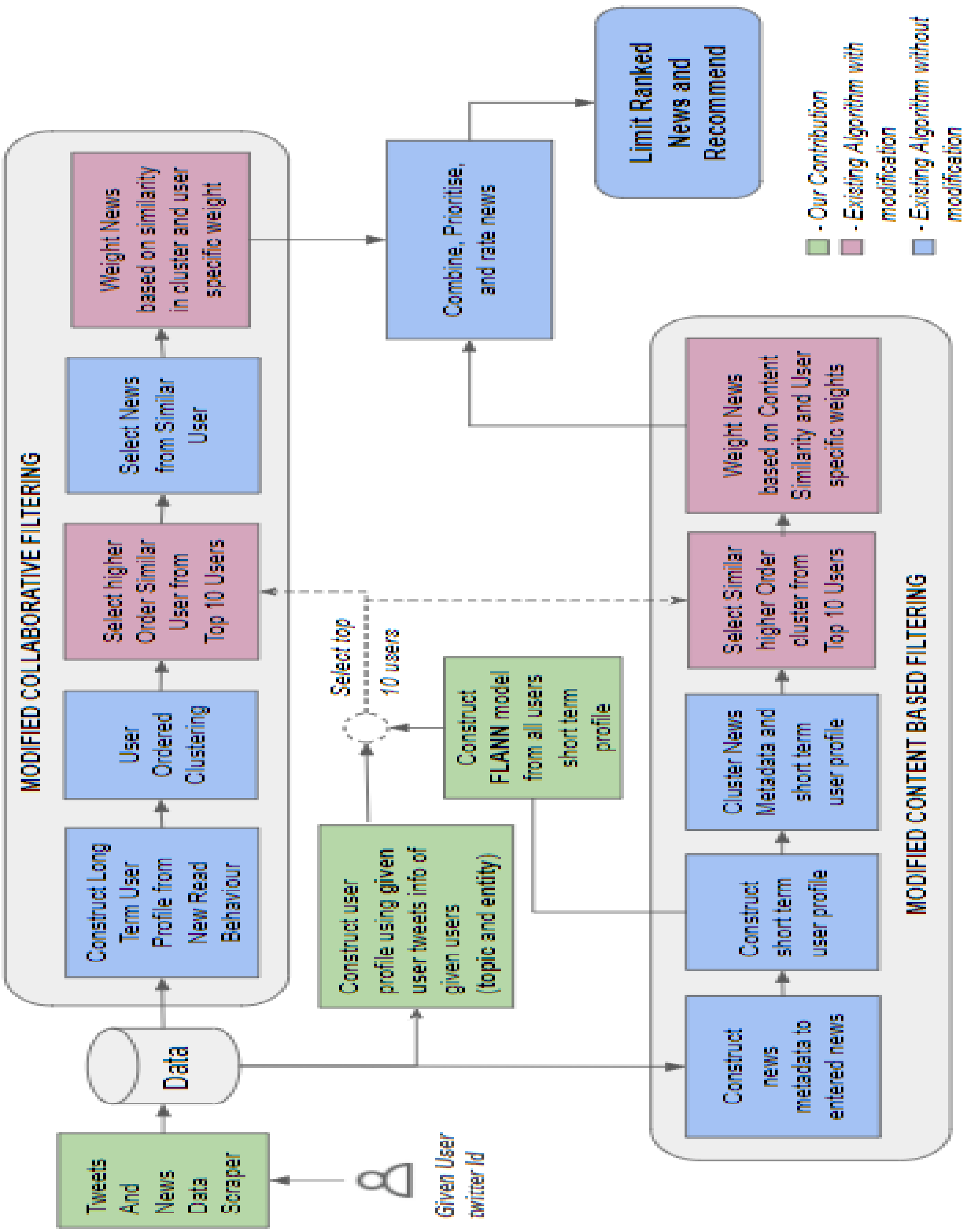


Figure 4.2 Detailed Architecture Diagram of the Proposed System

4.2 Tweets and News Data Scraper

(i) Extraction of tweets from official twitter handles of News Channels

The tweets of official news channels are traced and extracted from the twitter and other related information like urls of the news, hashtags used in the tweet are also extracted along with the tweet.

Further, information like time of tweet, number of retweets, user-Id of the retweeter and the time of the retweet are taken from this part and these information can be of use in order to determine the hotness of the news, reading rate of the user which will be useful in the future.

(ii) Extraction of tweets from user and user related followers

The Tweets of the user are extracted in order to find the area of interest of the user which can be useful in making meaningful recommendations for the user. In case the user is not an active user of twitter then in such a case the tweet of the followers can be much helpful in finding the area of interest of the user.

(iii)Scrapping of News Articles

News articles are scrapped from the news links and the content of the news articles are processed to get the title metadata, content ,news published time etc.

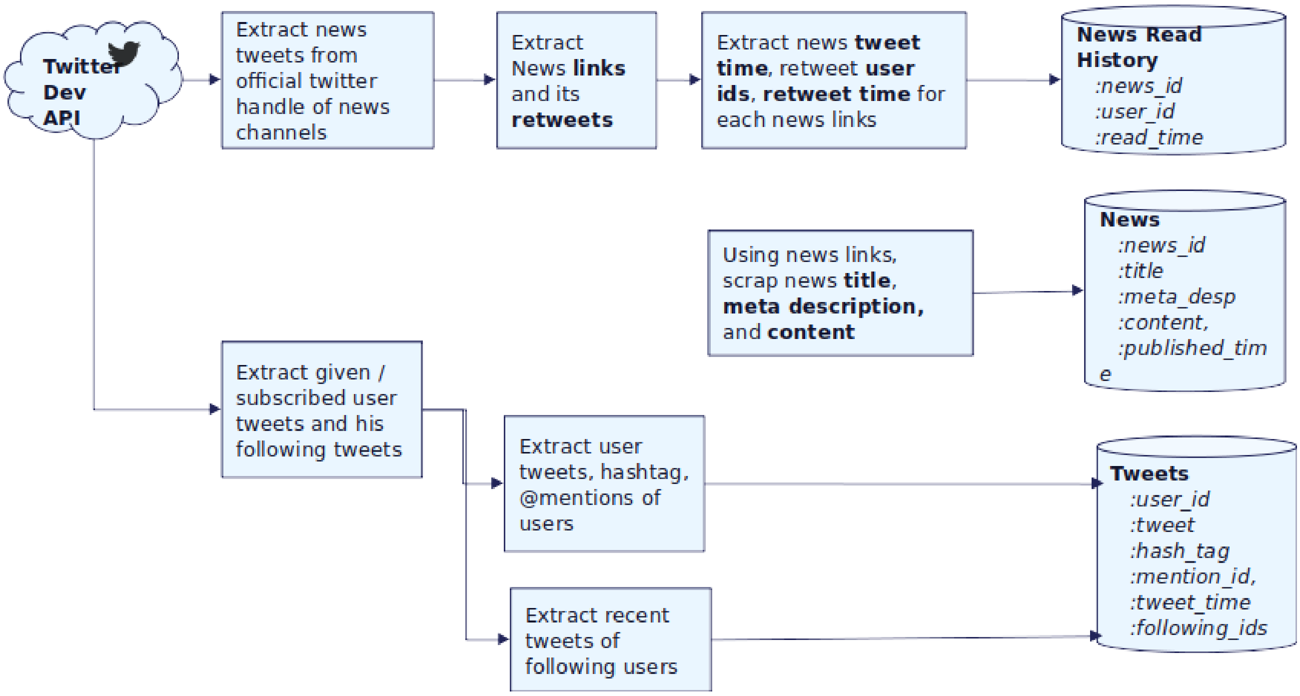


Figure 4.3 Scraper Module Architecture Diagram

4.3 Collaborative Module and Long-Term User Profile

(i) Construction of Long-Term Profile

The long-term user profile is parameterised with a three-dimensional tuple

$$L = \{Us, R, Hr\},$$

where:

1) Us represents a set of users and the similarity ratios to a given user u-i which are computed by utilising the Binary Jaccard Similarity

2) R is the Reading Rate . Because the number of news articles that a user reads daily is different from the other users, this behaviour should be considered in the news selection process.

3) Hotness Rate (Hr) is the average value of Hotness of a news article which a user likes to read.

(ii) User Clustering and News Selection

Similarity Matrix is created between the users using the Jaccard Similarity and the similarity matrix is given as input to the Ordered clustering algorithm .The output of the ordered clustering is a cluster of similar users and a Cluster Matrix.

Now based on the user clusters the news items are weighed and the news items that are related to the users are selected.

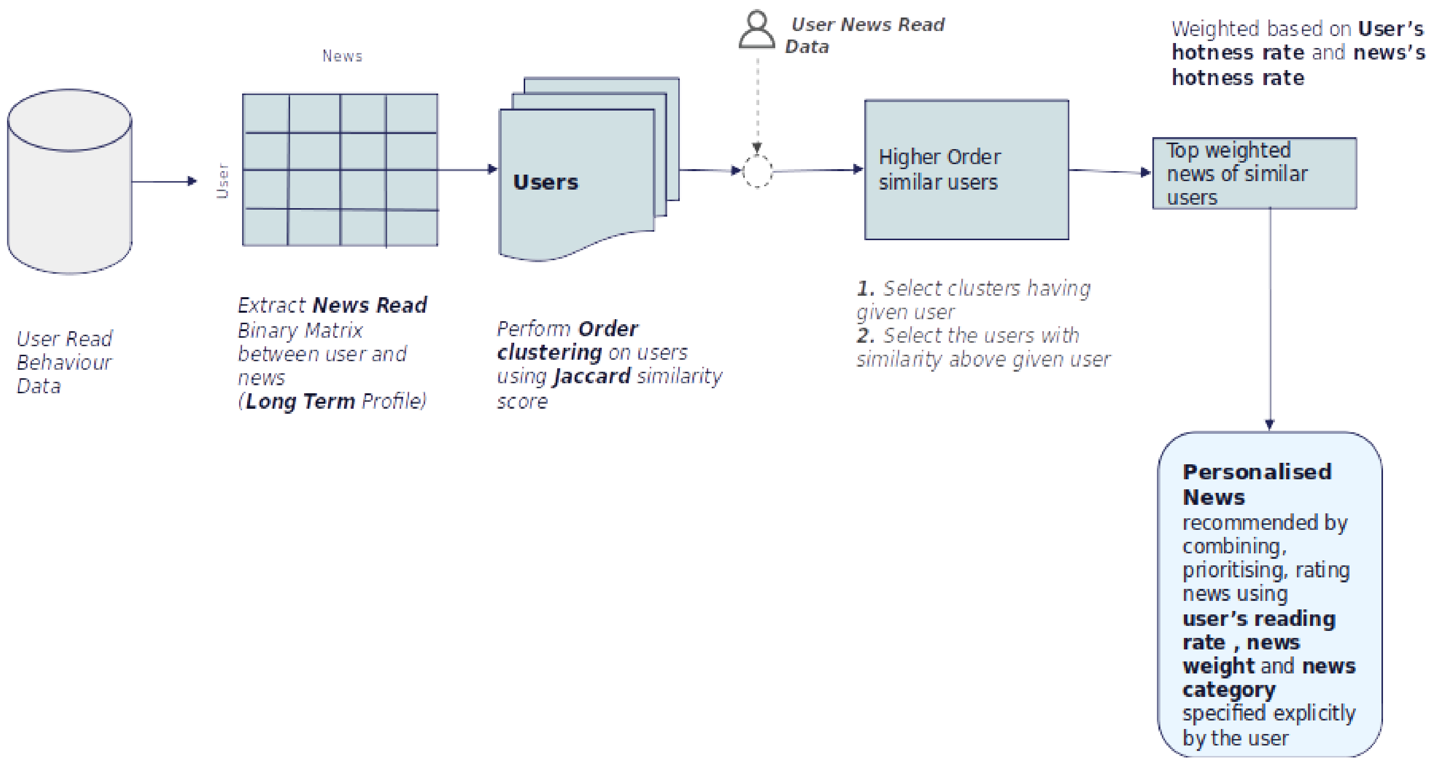


Figure 4.4 Architecture Diagram - Collaborative Module

4.4 Content Based filter Module

(i) Construction of Short-Term Profile

The short-term user profile is parameterised with a two-dimensional tuple

$$S = \{T, E\},$$

where:

1) T represents the named entities with their relevance tags that are extracted from the topics of an accessed news article,

$$T = \{\{t_1,tr_1\}, \{t_2,tr_2\}, \ldots, \{t_m,tr_m\}\}$$

where t_i represents the named entity and tr_i represents the relevance tag of tr_i , and they are gathered from the user's accessed news topics.

2) E is a set of named entities and their relevance tags that are extracted from the read news content,

$$E = \{\{e_1,er_1\}, \{e_2,er_2\}, \ldots, \{e_m,er_m\}\}$$

where e_i represents the named entity and er_i represents the relevance tag of e_i , and they are gathered from the user's accessed news content.

(ii) Construction of News Meta-Data

The news metadata N is parameterised with a five-dimensional tuple,

$$N = \{T,E,P,Rc,H\}$$

where:

1) T denotes a set of named entities and their relevance tags that are extracted

from the news topic,

$$T = \{\{t_1,tr_1\}, \{t_2,tr_2\}, \ldots, \{t_m,tr_m\}\}$$

where ti represents the named entity and tri represents the relevance tag of ti.

2) E represents a set of named entities and their relevance tags that are extracted from the news content,

$$E = \{\{e_1, er_1\}, \{e_2, er_2\}, \ldots, \{e_m, er_m\}\}$$

where e_i represents the named entity and eri represents the relevance tag of e_i.

3) P is the news popularity and it represents the number of times a news article is read by the users.

4) Rc is the news recency and it is a score that is computed based on :

$$Rc = NewsReadTime – NewsPublishedTime$$

Equation No. **4.1** Recency

5) H is the Hotness of a news article. In other words, it represents the interestingness of the news article. Hotness is computed as:

$$H = Popularity/Recency$$

Equation No. **4.2** Hotness

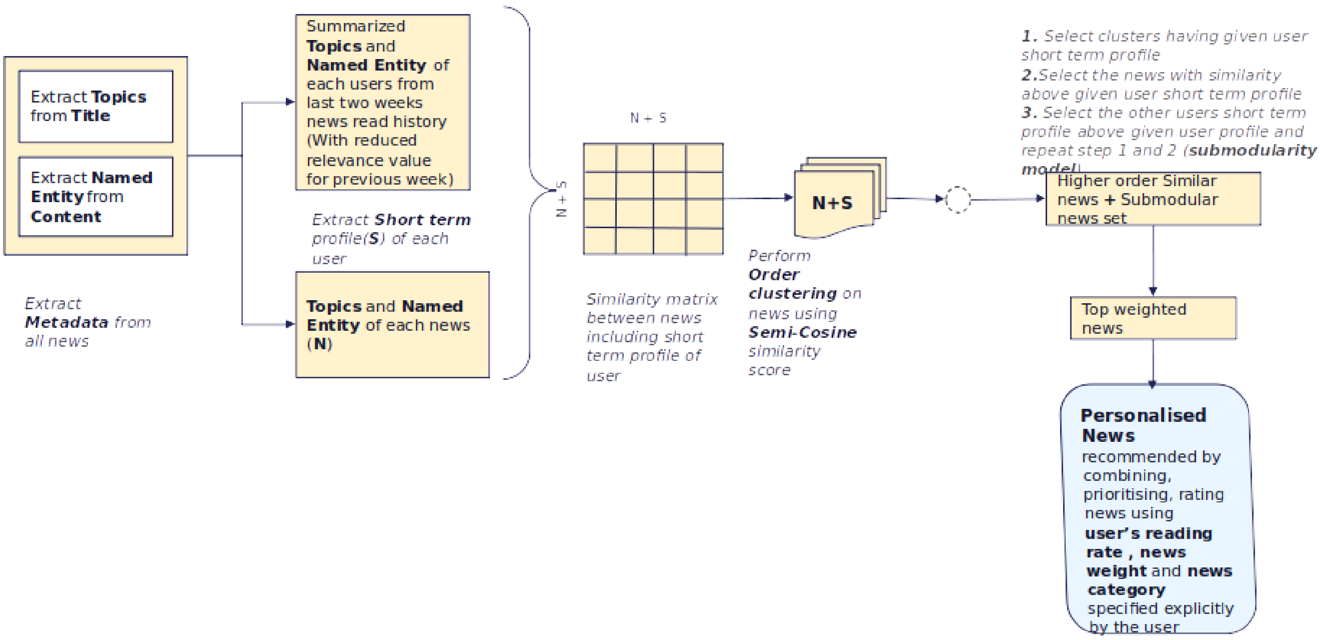


Figure 4.5 Architecture Diagram - Content Based Module

4.5 Additional Collaborative model using Short-Term Profile

Using the FLANN model all the existing user's short term profile are plotted in a multi dimensional graph and then the new users explicit profile vectorised form is given as input to the FLANN model to get the nearest neighbour among the existing user. The nearest neighbour is identified and the recommendations of the nearest neighbour is recommended to the new user.

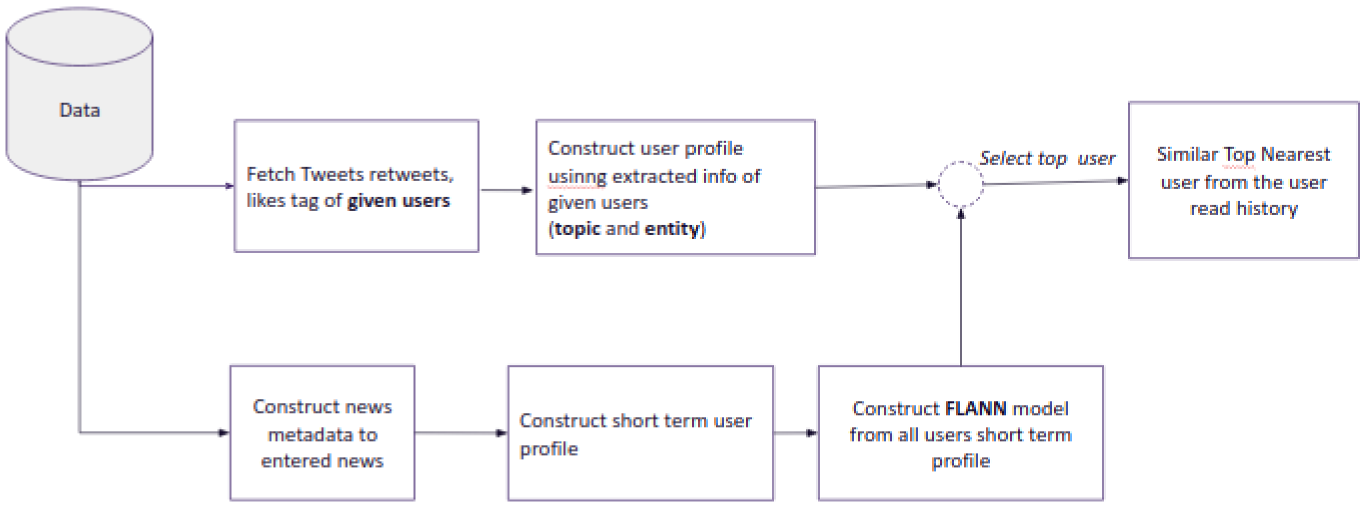


Figure 4.6 Architecture Diagram - Additional Collaborative Module

4.6 Personalised News Recommendation Module

(i) Combine, Prioritise and Rate News

The selected news sets from both of the CF-based and the Content-based methods of the system are combined to generate the final news set to recommend. An approach is proposed to prioritise the combined news set to finalise the recommended news set. The proposed approach is performed as follows.

Firstly, the two sets of the selected news, namely: the CF-based news set (News CF) and the Content-based news set (News CB) as well as the explicit user profile are passed to this procedure as inputs. Secondly, the News CF and News CB are combined and prioritised to produce the final news set, News FINAL, as shown below:

$$\text{New}_{\text{Final}} = \alpha\text{News}_{\text{CF}} + \beta\text{News}_{\text{CB}}$$

Equation No. 4.3

where α and β are parameters to control how we trust the corresponding CF-based and Content-based methods.

(i) LIMIT RANKED NEWS AND RECOMMEND

Reading Rate determines an average number of news items which a user prefers to read per day. Each user has a different behaviour in news reading and the number of daily news reading varies based on the user's interest and behaviour in news reading. The number of recommended news articles is computed as a coefficient of Reading Rate.

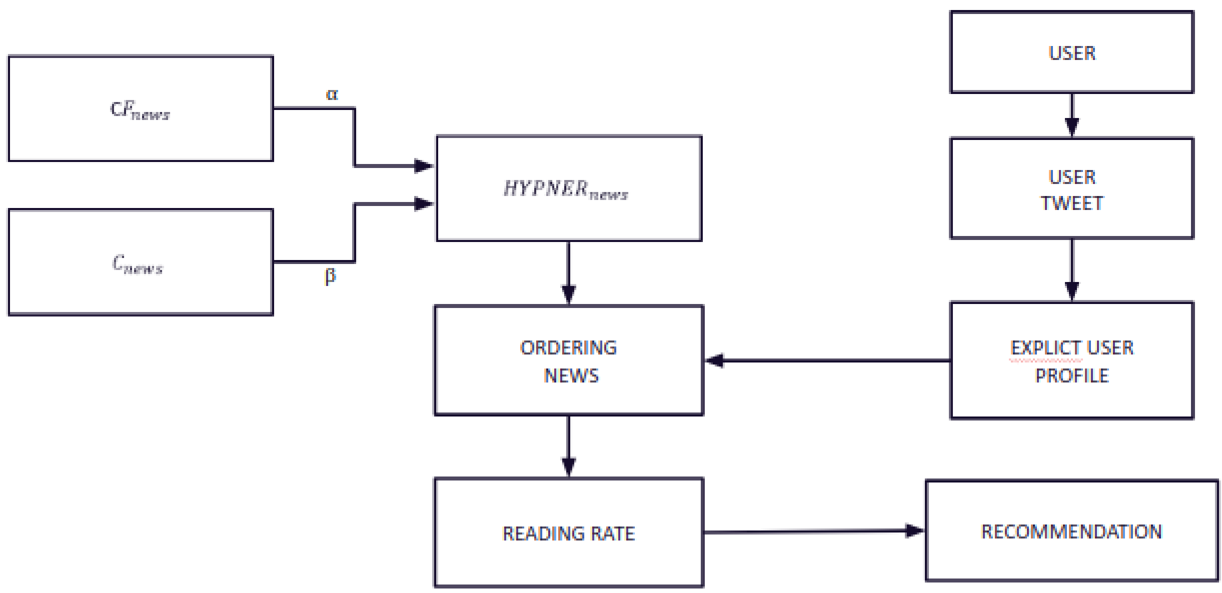


Figure 4.7 Architecture Diagram - Personalised News Recommendation Module

CHAPTER 5

SYSTEM DEVELOPMENT

This chapter discusses the various steps during the implementation of the Proposed system.

5.1 EXTRACTION OF NEWS CHANNEL TWEET DATA

```
-[ RECORD 1 ]-------------------------
id              | 13
tweet_id        | 1367856190928154624
text            | YouTube has removed channels from broadcasters run by Myanmar's military following a dramatic escalation of viol
enc_ https://t.co/aCkrVANhoj
retweet_count   | 99
created_at      | 2021-03-05 20:45:04+05:30
likes           | 413
url             | https://twitter.com/i/web/status/1367856190928154624
expanded_url    | https://edition.cnn.com/2021/03/05/tech/youtube-tiktok-myanmar-military-videos-intl-hnk/index.html?utm_source=tw
CNN&utm_term=link&utm_content=2021-03-05T15%3A15%3A03&utm_medium=social
news_channel_id | 1
-[ RECORD 2 ]-------------------------
id              | 18
tweet_id        | 1367837309715030020
text            | Britain's Prince Philip is transferred to a private hospital in London following a "successful" heart procedure,
 Bu_ https://t.co/waMpMc1yFe
retweet_count   | 39
created_at      | 2021-03-05 19:30:02+05:30
likes           | 275
url             | https://twitter.com/i/web/status/1367837309715030020
expanded_url    | https://edition.cnn.com/2021/03/05/uk/prince-philip-hospital-transfer-gbr-intl/index.html?utm_medium=social&utm_
content=2021-03-05T14%3A00%3A01&utm_term=link&utm_source=twCNN
news_channel_id | 1
```

Figure 5.1 News Tweet Data of News Handles

Figure 5.1 shows the tweets data that is collected from news handles that consists of tweet text,likes,news url,retweet count time of tweet tweet id etc.

```
-[ RECORD 1 ]-------------------------
id             | 1
tweet_id       | 1367909240342536194
created_at     | 2021-03-06 00:15:52+05:30
news_tweet_id  | 1
users_id       | 1
-[ RECORD 2 ]-------------------------
id             | 2
tweet_id       | 1367909120788082695
created_at     | 2021-03-06 00:15:23+05:30
news_tweet_id  | 1
users_id       | 2
SELECT 2
```

Figure 5.2 Retweet information of News Tweets

Figure 5.2 shows the information regarding the retweets for the tweets in Figure 5.1 which is further used to construct the NR Matrix

```
-[ RECORD 1 ]-------------------------
id              | 10
twitter_id      | GeminiBull
reading_rate    | 32
hotness_rate    | 23.289762721627305
friends_count   | 499
followers_count | 773
-[ RECORD 2 ]-------------------------
id              | 11
twitter_id      | ibrayoosuf
reading_rate    | 36
hotness_rate    | 11.033569979956319
friends_count   | 113
followers_count | 427
SELECT 2
Time: 0.003s
```

Figure 5.3 User Information

The information regarding the user who has retweeted on the news is also collected and stored.

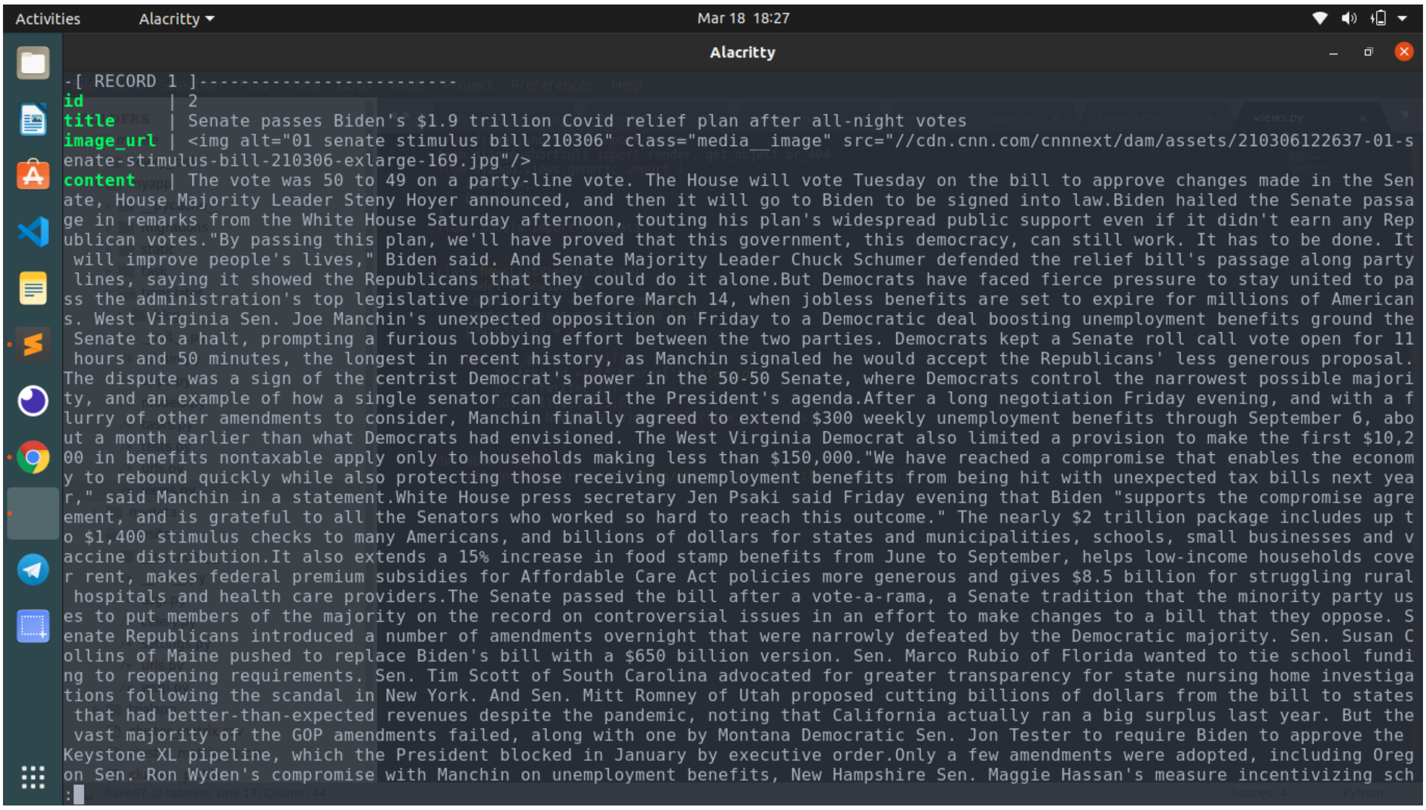
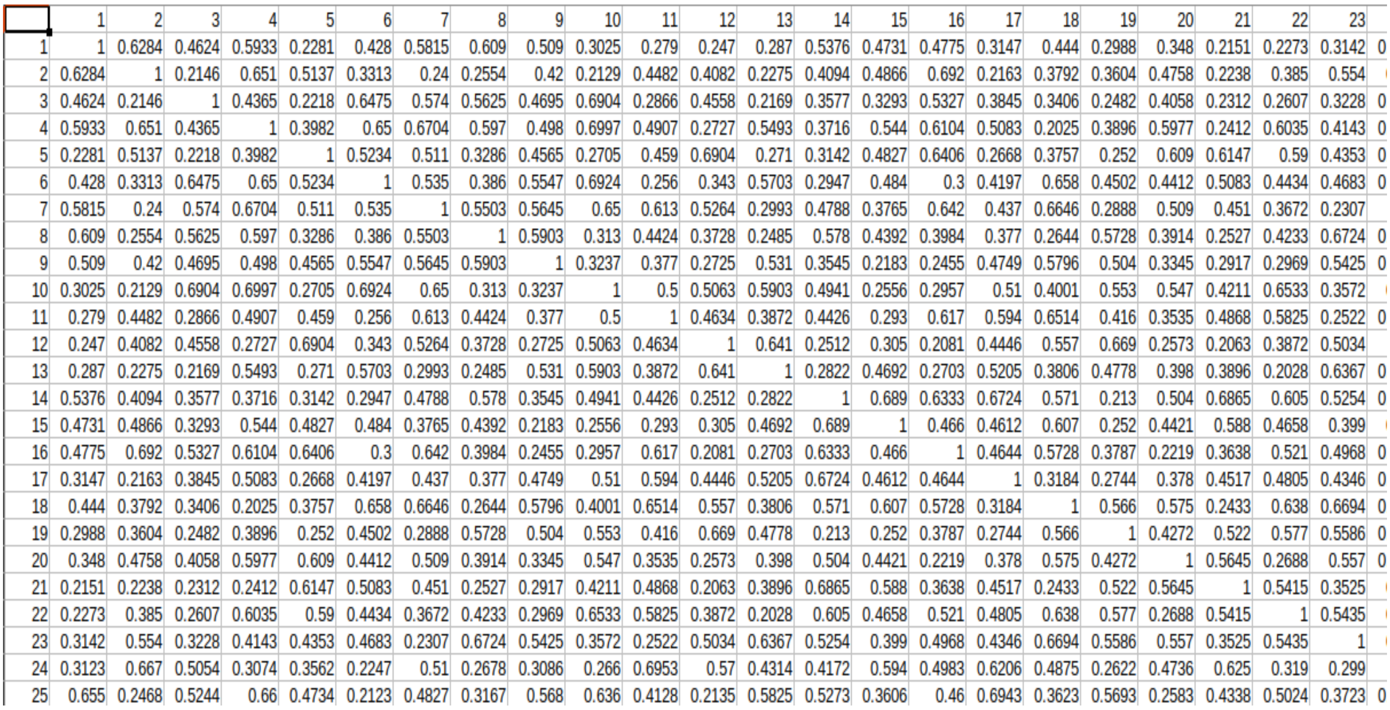


Figure 5.4 News Information

5.2 COLLABORATIVE FILTER AND LONG TERM PROFILE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX
1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
2	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
6	5	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
7	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
8	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
9	8	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
10	9	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
11	10	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0
12	11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
13	12	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0
14	13	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
15	14	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	0
16	15	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
17	16	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
18	17	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1
19	18	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	19	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	21	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0
23	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	24	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
26	25	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

Figure 5.5 News Read Matrix



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	1	0.6284	0.4624	0.5933	0.2281	0.428	0.5815	0.609	0.509	0.3025	0.279	0.247	0.287	0.5376	0.4731	0.4775	0.3147	0.444	0.2988	0.348	0.2151	0.2273	0.3142	0
2	0.6284	1	0.2146	0.651	0.5137	0.3313	0.24	0.2554	0.42	0.2129	0.4482	0.4082	0.2275	0.4094	0.4866	0.692	0.2163	0.3792	0.3604	0.4758	0.2238	0.385	0.554	
3	0.4624	0.2146	1	0.4365	0.2218	0.6475	0.574	0.5625	0.4695	0.6904	0.2866	0.4558	0.2169	0.3577	0.3293	0.5327	0.3845	0.3406	0.2482	0.4058	0.2312	0.2607	0.3228	0
4	0.5933	0.651	0.4365	1	0.3982	0.65	0.6704	0.597	0.498	0.6997	0.4907	0.2727	0.5493	0.3716	0.544	0.6104	0.5083	0.2025	0.3896	0.5977	0.2412	0.6035	0.4143	0
5	0.2281	0.5137	0.2218	0.3982	1	0.5234	0.511	0.3286	0.4565	0.2705	0.459	0.6904	0.271	0.3142	0.4827	0.6406	0.2668	0.3757	0.252	0.609	0.6147	0.59	0.4353	0
6	0.428	0.3313	0.6475	0.65	0.5234	1	0.535	0.386	0.5547	0.6924	0.256	0.343	0.5703	0.2947	0.484	0.3	0.4197	0.658	0.4502	0.4412	0.5083	0.4434	0.4683	0
7	0.5815	0.24	0.574	0.6704	0.511	0.535	1	0.5503	0.5645	0.65	0.613	0.5264	0.2993	0.4788	0.3765	0.642	0.437	0.6646	0.2888	0.509	0.451	0.3672	0.2307	
8	0.609	0.2554	0.5625	0.597	0.3286	0.386	0.5503	1	0.5903	0.313	0.4424	0.3728	0.2485	0.578	0.4392	0.3984	0.377	0.2644	0.5728	0.3914	0.2527	0.4233	0.6724	0
9	0.509	0.42	0.4695	0.498	0.4565	0.5547	0.5645	0.5903	1	0.3237	0.377	0.2725	0.531	0.3545	0.2183	0.2455	0.4749	0.5796	0.504	0.3345	0.2917	0.2969	0.5425	0
10	0.3025	0.2129	0.6904	0.6997	0.2705	0.6924	0.65	0.313	0.3237	1	0.5	0.5063	0.5903	0.4941	0.2556	0.2957	0.51	0.4001	0.553	0.547	0.4211	0.6533	0.3572	
11	0.279	0.4482	0.2866	0.4907	0.459	0.256	0.613	0.4424	0.377	0.5	1	0.4634	0.3872	0.4426	0.293	0.617	0.594	0.6514	0.416	0.3535	0.4868	0.5825	0.2522	0
12	0.247	0.4082	0.4558	0.2727	0.6904	0.343	0.5264	0.3728	0.2725	0.5063	0.4634	1	0.641	0.2512	0.305	0.2081	0.4446	0.557	0.669	0.2573	0.2063	0.3872	0.5034	
13	0.287	0.2275	0.2169	0.5493	0.271	0.5703	0.2993	0.2485	0.531	0.5903	0.3872	0.641	1	0.2822	0.4692	0.2703	0.5205	0.3806	0.4778	0.398	0.3896	0.2028	0.6367	0
14	0.5376	0.4094	0.3577	0.3716	0.3142	0.2947	0.4788	0.578	0.3545	0.4941	0.4426	0.2512	0.2822	1	0.689	0.6333	0.6724	0.571	0.213	0.504	0.6865	0.605	0.5254	0
15	0.4731	0.4866	0.3293	0.544	0.4827	0.484	0.3765	0.4392	0.2183	0.2556	0.293	0.305	0.4692	0.689	1	0.466	0.4612	0.607	0.252	0.4421	0.588	0.4658	0.399	
16	0.4775	0.692	0.5327	0.6104	0.6406	0.3	0.642	0.3984	0.2455	0.2957	0.617	0.2081	0.2703	0.6333	0.466	1	0.4644	0.5728	0.3787	0.2219	0.3638	0.521	0.4968	0
17	0.3147	0.2163	0.3845	0.5083	0.2668	0.4197	0.437	0.377	0.4749	0.51	0.594	0.4446	0.5205	0.6724	0.4612	0.4644	1	0.3184	0.2744	0.378	0.4517	0.4805	0.4346	0
18	0.444	0.3792	0.3406	0.2025	0.3757	0.658	0.6646	0.2644	0.5796	0.4001	0.6514	0.557	0.3806	0.571	0.607	0.5728	0.3184	1	0.566	0.575	0.2433	0.638	0.6694	0
19	0.2988	0.3604	0.2482	0.3896	0.252	0.4502	0.2888	0.5728	0.504	0.553	0.416	0.669	0.4778	0.213	0.252	0.3787	0.2744	0.566	1	0.4272	0.522	0.577	0.5586	0
20	0.348	0.4758	0.4058	0.5977	0.609	0.4412	0.509	0.3914	0.3345	0.547	0.3535	0.2573	0.398	0.504	0.4421	0.2219	0.378	0.575	0.4272	1	0.5645	0.2688	0.557	0
21	0.2151	0.2238	0.2312	0.2412	0.6147	0.5083	0.451	0.2527	0.2917	0.4211	0.4868	0.2063	0.3896	0.6865	0.588	0.3638	0.4517	0.2433	0.522	0.5645	1	0.5415	0.3525	
22	0.2273	0.385	0.2607	0.6035	0.59	0.4434	0.3672	0.4233	0.2969	0.6533	0.5825	0.3872	0.2028	0.605	0.4658	0.521	0.4805	0.638	0.577	0.2688	0.5415	1	0.5435	
23	0.3142	0.554	0.3228	0.4143	0.4353	0.4683	0.2307	0.6724	0.5425	0.3572	0.2522	0.5034	0.6367	0.5254	0.399	0.4968	0.4346	0.6694	0.5586	0.557	0.3525	0.5435	1	
24	0.3123	0.667	0.5054	0.3074	0.3562	0.2247	0.51	0.2678	0.3086	0.266	0.6953	0.57	0.4314	0.4172	0.594	0.4983	0.6206	0.4875	0.2622	0.4736	0.625	0.319	0.299	
25	0.655	0.2468	0.5244	0.66	0.4734	0.2123	0.4827	0.3167	0.568	0.636	0.4128	0.2135	0.5825	0.5273	0.3606	0.46	0.6943	0.3623	0.5693	0.2583	0.4338	0.5024	0.3723	0

Figure 5.6 User Similarity Matrix

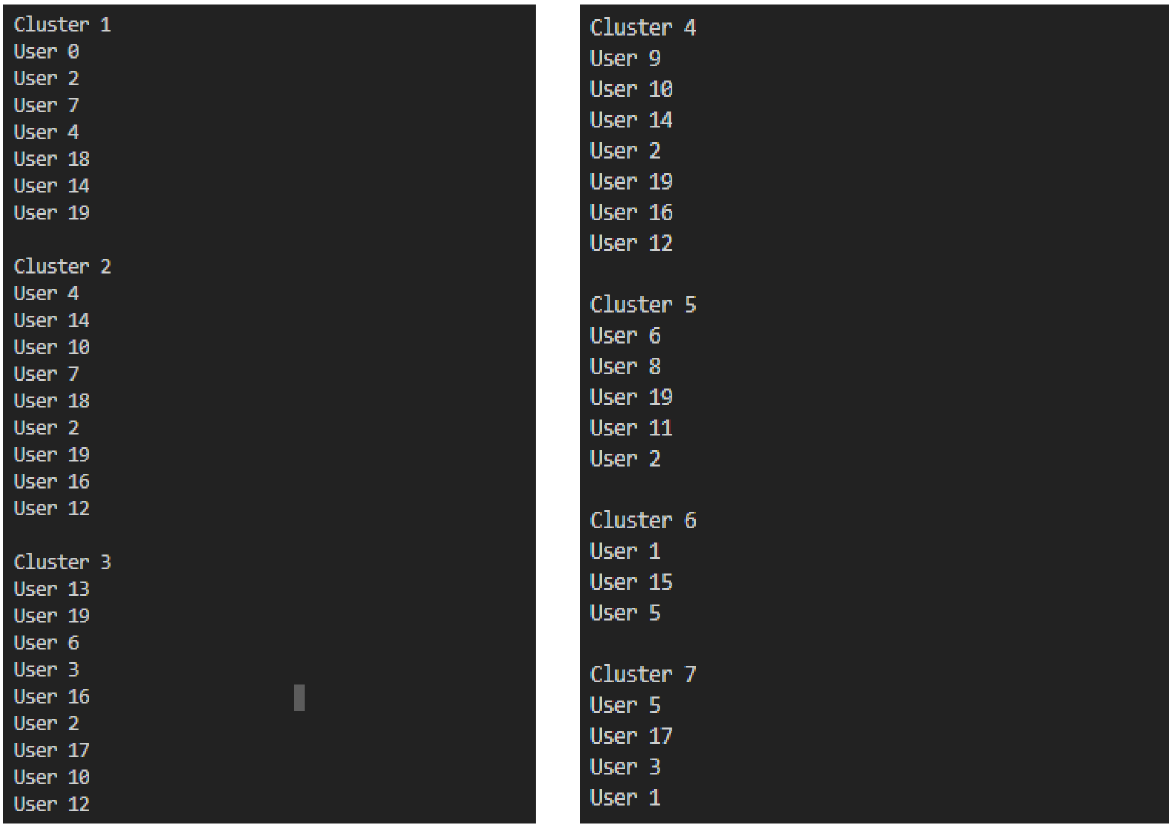


Figure 5.7 User Cluster

The News Read matrix(Figure 5.5) is constructed from user read behaviour from which similarity matrix(Figure 5.6) is constructed using jaccard similarity. The users are clustered by performing ordered clustering on the users similarity matrix.The users clusters obtained by ordered clustering is shown in figure 5.7.

```
>>> from myapp.task.collaborative_filter import *

>>> similar_user = find_similar_users(44)

>>> similar_news = find_colaborative_similar_news(similar_user)
FINDING SIMILAR NEWS ...
CALCULATING NEWS SCORE ...
>>>
>>> similar_news[:15]
[[3, 13.156700000000003], [894, 12.9513], [87, 12.9325], [77, 11.96830000000000
1], [29, 11.7828], [104, 11.219200000000003], [761, 11.071399999999999], [544,
11.040000000000001], [498, 10.943100000000001], [78, 10.8428], [789, 10.6322000
00000001], [562, 10.621899999999998], [161, 10.5101], [901, 10.411399999999999]
, [35, 10.386600000000001]]
>>>
```

Figure 5.7 Resultant News of Collaborative Module

5.3 CONTENT BASED FILTER AND SHORT TERM PROFILE

```
df
```

	Entities	Labels	Position_Start	Position_End
0	(amic, sol, pare, camp, molt)	ORG	82	105
1	(covid, han, fet, mental)	PERSON	126	146
2	(inaugur, cup)	EVENT	43	54
3	(liber)	PERSON	75	80
4	(democrat)	NORP	81	89
...	...	...	...	...
817	(cox)	PERSON	234	237
818	(covid)	PERSON	321	326
819	(first)	ORDINAL	422	427
820	(week)	DATE	86	90
821	(three)	CARDINAL	190	195

822 rows × 4 columns

Figure 5.8 Entity Identificaion in News Content

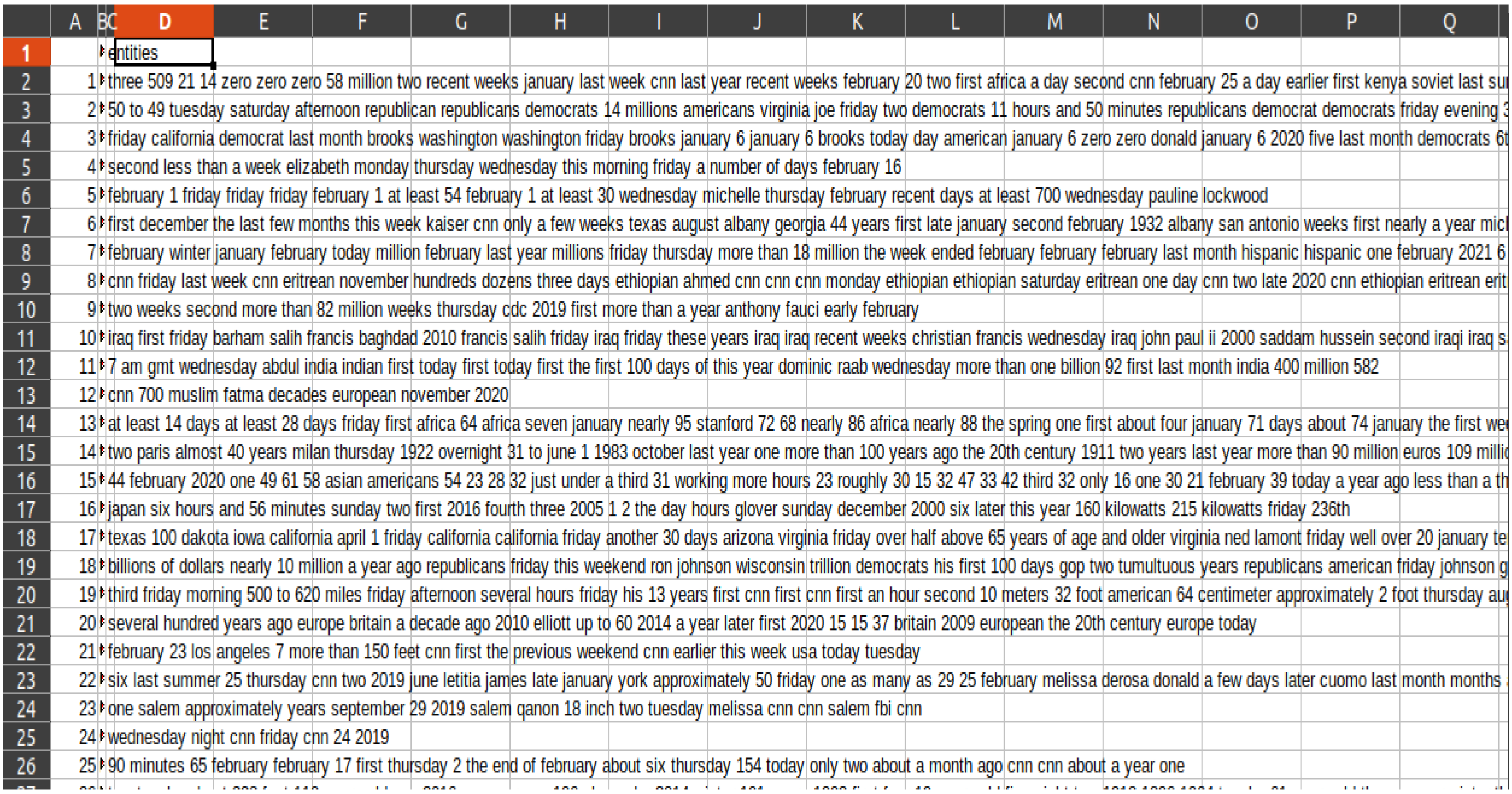
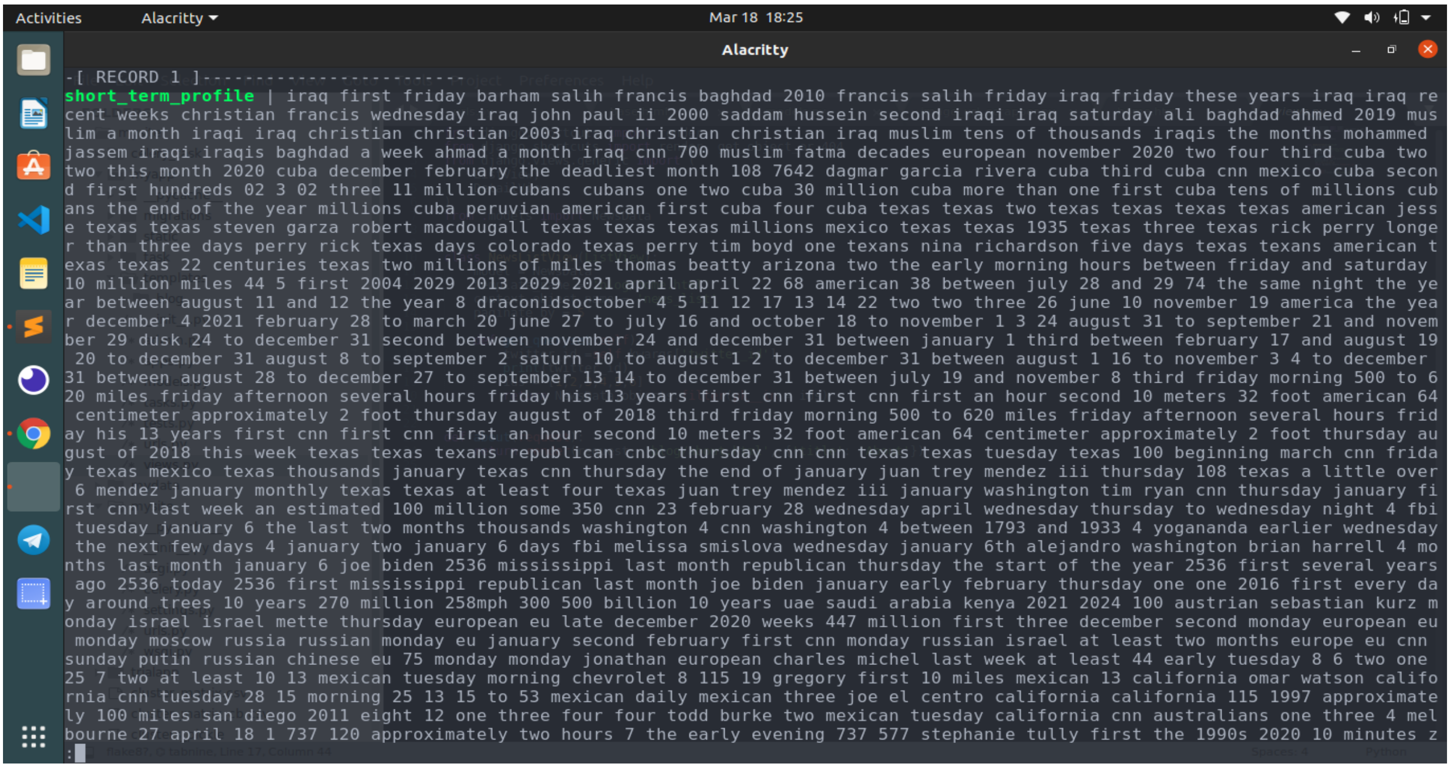


Figure 5.9 News Meta Data

On all the news items named entity recognition is performed and all the entities constitute to form the news meta data ,such metadata are stored for further use.



. **Figure 5.10** User Short Term Profile

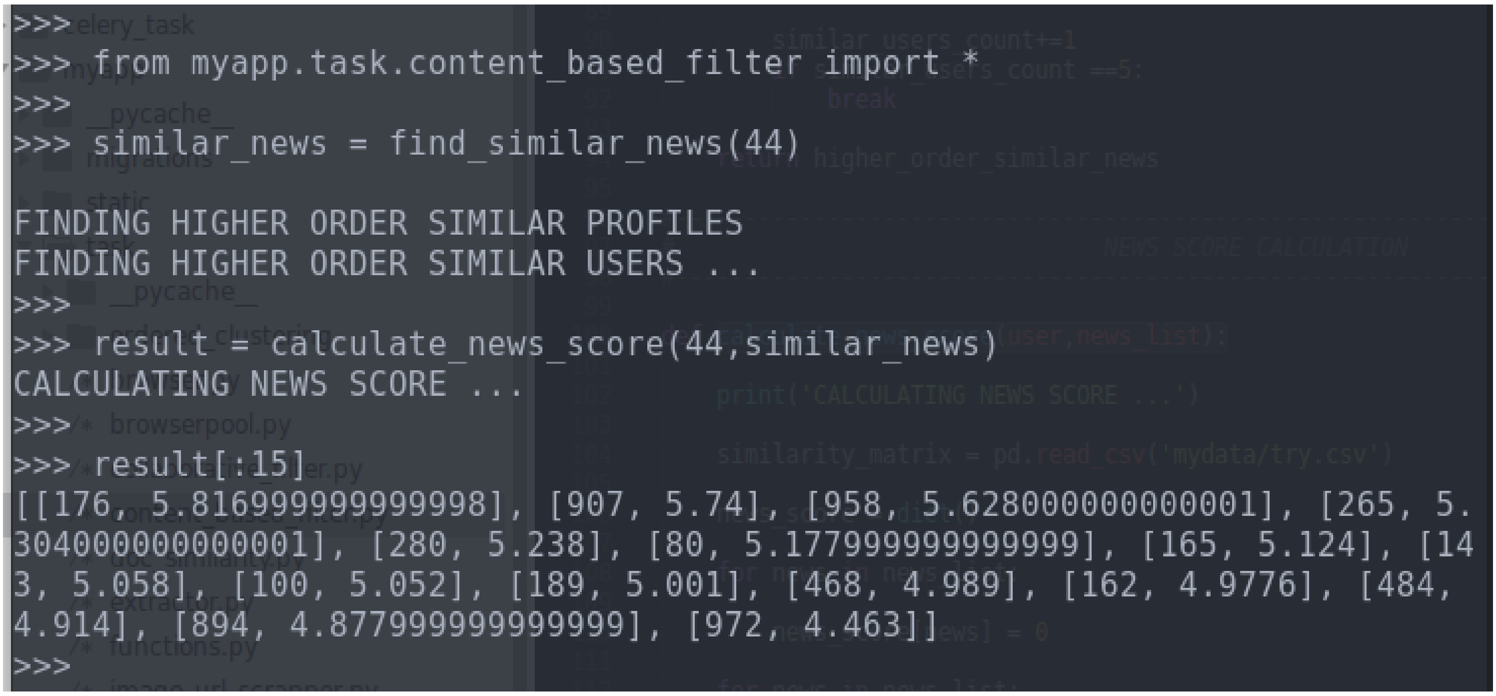
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	1	0.6284	0.4624	0.5933	0.2281	0.428	0.5815	0.609	0.509	0.3025	0.279	0.247	0.287	0.5376	0.4731	0.4775	0.3147	0.444	0.2988	0.348	0.2151	0.2273	0.3142	0
2	0.6284	1	0.2146	0.651	0.5137	0.3313	0.24	0.2554	0.42	0.2129	0.4482	0.4082	0.2275	0.4094	0.4866	0.692	0.2163	0.3792	0.3604	0.4758	0.2238	0.385	0.554	
3	0.4624	0.2146	1	0.4365	0.2218	0.6475	0.574	0.5625	0.4695	0.6904	0.2866	0.4558	0.2169	0.3577	0.3293	0.5327	0.3845	0.3406	0.2482	0.4058	0.2312	0.2607	0.3228	0
4	0.5933	0.651	0.4365	1	0.3982	0.65	0.6704	0.597	0.498	0.6997	0.4907	0.2727	0.5493	0.3716	0.544	0.6104	0.5083	0.2025	0.3896	0.5977	0.2412	0.6035	0.4143	0
5	0.2281	0.5137	0.2218	0.3982	1	0.5234	0.511	0.3286	0.4565	0.2705	0.459	0.6904	0.271	0.3142	0.4827	0.6406	0.2668	0.3757	0.252	0.609	0.6147	0.59	0.4353	0
6	0.428	0.3313	0.6475	0.65	0.5234	1	0.535	0.386	0.5547	0.6924	0.256	0.343	0.5703	0.2947	0.484	0.3	0.4197	0.658	0.4502	0.4412	0.5083	0.4434	0.4683	0
7	0.5815	0.24	0.574	0.6704	0.511	0.535	1	0.5503	0.5645	0.65	0.613	0.5264	0.2993	0.4788	0.3765	0.642	0.437	0.6646	0.2888	0.509	0.451	0.3672	0.2307	
8	0.609	0.2554	0.5625	0.597	0.3286	0.386	0.5503	1	0.5903	0.313	0.4424	0.3728	0.2485	0.578	0.4392	0.3984	0.377	0.2644	0.5728	0.3914	0.2527	0.4233	0.6724	0
9	0.509	0.42	0.4695	0.498	0.4565	0.5547	0.5645	0.5903	1	0.3237	0.377	0.2725	0.531	0.3545	0.2183	0.2455	0.4749	0.5796	0.504	0.3345	0.2917	0.2969	0.5425	0
10	0.3025	0.2129	0.6904	0.6997	0.2705	0.6924	0.65	0.313	0.3237	1	0.5	0.5063	0.5903	0.4941	0.2556	0.2957	0.51	0.4001	0.553	0.547	0.4211	0.6533	0.3572	
11	0.279	0.4482	0.2866	0.4907	0.459	0.256	0.613	0.4424	0.377	0.5	1	0.4634	0.3872	0.4426	0.293	0.617	0.594	0.6514	0.416	0.3535	0.4868	0.5825	0.2522	0
12	0.247	0.4082	0.4558	0.2727	0.6904	0.343	0.5264	0.3728	0.2725	0.5063	0.4634	1	0.641	0.2512	0.305	0.2081	0.4446	0.557	0.669	0.2573	0.2063	0.3872	0.5034	
13	0.287	0.2275	0.2169	0.5493	0.271	0.5703	0.2993	0.2485	0.531	0.5903	0.3872	0.641	1	0.2822	0.4692	0.2703	0.5205	0.3806	0.4778	0.398	0.3896	0.2028	0.6367	0
14	0.5376	0.4094	0.3577	0.3716	0.3142	0.2947	0.4788	0.578	0.3545	0.4941	0.4426	0.2512	0.2822	1	0.689	0.6333	0.6724	0.571	0.213	0.504	0.6865	0.605	0.5254	0
15	0.4731	0.4866	0.3293	0.544	0.4827	0.484	0.3765	0.4392	0.2183	0.2556	0.293	0.305	0.4692	0.689	1	0.466	0.4612	0.607	0.252	0.4421	0.588	0.4658	0.399	
16	0.4775	0.692	0.5327	0.6104	0.6406	0.3	0.642	0.3984	0.2455	0.2957	0.617	0.2081	0.2703	0.6333	0.466	1	0.4644	0.5728	0.3787	0.2219	0.3638	0.521	0.4968	0
17	0.3147	0.2163	0.3845	0.5083	0.2668	0.4197	0.437	0.377	0.4749	0.51	0.594	0.4446	0.5205	0.6724	0.4612	0.4644	1	0.3184	0.2744	0.378	0.4517	0.4805	0.4346	0
18	0.444	0.3792	0.3406	0.2025	0.3757	0.658	0.6646	0.2644	0.5796	0.4001	0.6514	0.557	0.3806	0.571	0.607	0.5728	0.3184	1	0.566	0.575	0.2433	0.638	0.6694	0
19	0.2988	0.3604	0.2482	0.3896	0.252	0.4502	0.2888	0.5728	0.504	0.553	0.416	0.669	0.4778	0.213	0.252	0.3787	0.2744	0.566	1	0.4272	0.522	0.577	0.5586	0
20	0.348	0.4758	0.4058	0.5977	0.609	0.4412	0.509	0.3914	0.3345	0.547	0.3535	0.2573	0.398	0.504	0.4421	0.2219	0.378	0.575	0.4272	1	0.5645	0.2688	0.557	0
21	0.2151	0.2238	0.2312	0.2412	0.6147	0.5083	0.451	0.2527	0.2917	0.4211	0.4868	0.2063	0.3896	0.6865	0.588	0.3638	0.4517	0.2433	0.522	0.5645	1	0.5415	0.3525	
22	0.2273	0.385	0.2607	0.6035	0.59	0.4434	0.3672	0.4233	0.2969	0.6533	0.5825	0.3872	0.2028	0.605	0.4658	0.521	0.4805	0.638	0.577	0.2688	0.5415	1	0.5435	
23	0.3142	0.554	0.3228	0.4143	0.4353	0.4683	0.2307	0.6724	0.5425	0.3572	0.2522	0.5034	0.6367	0.5254	0.399	0.4968	0.4346	0.6694	0.5586	0.557	0.3525	0.5435	1	
24	0.3123	0.667	0.5054	0.3074	0.3562	0.2247	0.51	0.2678	0.3086	0.266	0.6953	0.57	0.4314	0.4172	0.594	0.4983	0.6206	0.4875	0.2622	0.4736	0.625	0.319	0.299	
25	0.655	0.2468	0.5244	0.66	0.4734	0.2123	0.4827	0.3167	0.568	0.636	0.4128	0.2135	0.5825	0.5273	0.3606	0.46	0.6943	0.3623	0.5693	0.2583	0.4338	0.5024	0.3723	0

Figure 5.11 Profile Similarity Matrix

```
Cluster  5 :

Profile  47
News  50

Cluster  6 :

Profile  52
Profile  53
News  42
Profile  15

Cluster  7 :

News  22
Profile  27
Profile  1
News  45

Cluster  8 :

Profile  46
News  53
News  41
News  29
```

```
Cluster  1 :

News  54
News  36
Profile  24

Cluster  2 :

Profile  17
Profile  18
News  16
News  40
News  15

Cluster  3 :

News  51
Profile  15
News  39
Profile  1
News  31
Profile  26
```

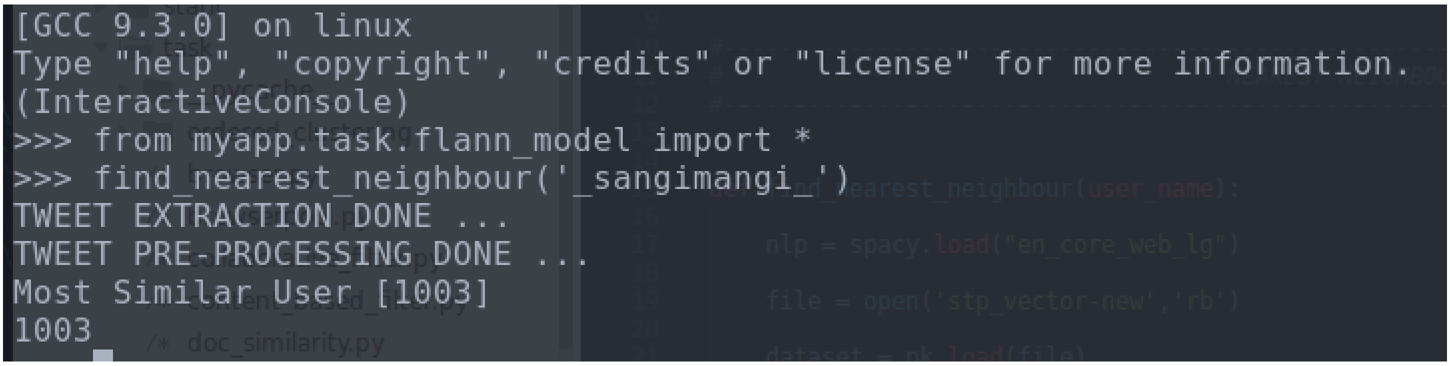
Figure 5.12 Profile and News Clusters



```
>>>
>>> from myapp.task.content_based_filter import *
>>>
>>> similar_news = find_similar_news(44)

FINDING HIGHER ORDER SIMILAR PROFILES
FINDING HIGHER ORDER SIMILAR USERS ...
>>>
>>> result = calculate_news_score(44,similar_news)
CALCULATING NEWS SCORE ...
>>>
>>> result[:15]
[[176, 5.8169999999999998], [907, 5.74], [958, 5.628000000000001], [265, 5.
304000000000001], [280, 5.238], [80, 5.1779999999999999], [165, 5.124], [14
3, 5.058], [100, 5.052], [189, 5.001], [468, 4.989], [162, 4.9776], [484,
4.914], [894, 4.8779999999999999], [972, 4.463]]
>>>
```

Figure 5.13 Resultant Recommendation of Content based Module

5.4 NEAREST NEIGHBOUR SEARCH



```
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from myapp.task.flann_model import *
>>> find_nearest_neighbour('_sangimangi_')
TWEET EXTRACTION DONE ...
TWEET PRE-PROCESSING DONE ...
Most Similar User [1003]
1003
```

Figure 5.14 Nearest Neighbour Finding using Flann

5.5 COMBINE AND PRIORITISE FINAL RECOMMENDATION

```
>>> Model.predict('joan_kem')
PREDICTING FOR USER : joan_kem
FINDING SIMILAR NEWS ...
CALCULATING NEWS SCORE ...
FINDING HIGHER ORDER SIMILAR PROFILES
FINDING HIGHER ORDER SIMILAR USERS ...
CALCULATING NEWS SCORE ...
[[679, 5.015000000000001], [335, 5.0116], [914, 4.8097], [104, 4.7374], [35, 4.732600000000001], [87, 4.6578], [974, 4.6284], [365
, 4.588000000000001], [593, 4.4514], [336, 4.4109], [86, 4.2729], [991, 4.1843], [674, 4.1601], [596, 4.1446], [683, 4.12599999999
99994], [611, 4.1212], [394, 4.077], [761, 4.0295], [863, 4.024799999999999], [746, 3.9113], [584, 3.9075], [534, 3.90479999999999
94], [957, 3.805], [147, 3.7788999999999997], [550, 3.7474], [940, 3.7110999999999996], [796, 3.7002], [262, 3.6887999999999996],
[537, 3.6653000000000002], [614, 3.6652000000000005], [103, 3.6633999999999998], [39, 3.6597], [640, 3.6541], [621, 3.6476], [985,
 3.6167000000000002], [28, 3.6131], [59, 3.6041999999999996], [154, 3.5937], [650, 3.5776], [200, 3.4853000000000005], [936, 3.477
2], [783, 3.3997], [393, 3.3988], [917, 3.3761], [697, 3.3758000000000004], [542, 3.3678999999999997], [978, 3.3594], [597, 3.3501
], [606, 3.3386], [623, 3.3357], [722, 3.3173], [511, 3.3118999999999996], [723, 3.2977], [102, 3.2741], [673, 3.2652], [78, 3.264
9], [645, 3.249], [272, 3.2401], [319, 3.2281999999999993], [522, 3.215], [299, 3.1975999999999996], [750, 3.1822], [626, 3.1635],
 [420, 3.1597999999999997], [682, 3.1487], [627, 3.1433999999999997], [943, 3.1316], [830, 3.1111999999999997], [887, 3.1085000000
000003], [692, 3.0468], [143, 5.9472], [556, 5.322000000000001], [712, 5.298], [552, 5.226000000000001], [257, 5.124], [580, 5.004
], [679, 4.983600000000001], [48, 4.9799999999999995], [165, 4.968], [116, 4.495], [344, 4.495], [350, 4.475], [479, 4.45], [299,
4.445], [61, 4.445], [98, 4.4385], [606, 4.4385], [180, 4.42], [148, 4.41], [282, 4.3774999999999995], [844, 4.375], [259, 4.37],
[145, 4.363], [890, 4.36], [942, 4.355], [154, 4.355], [988, 4.35], [547, 4.348], [376, 4.343500000000001], [262, 4.3335]]
>>>
```

Figure 5.15 Combining & Prioritising Final Recommendations

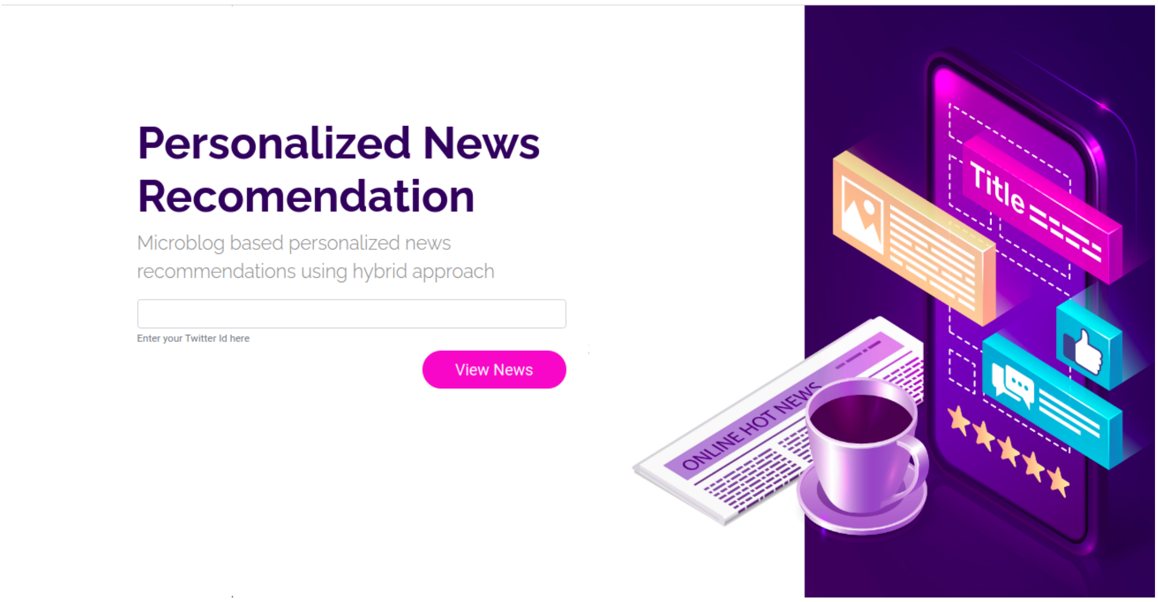


Figure 5.16 User Interface

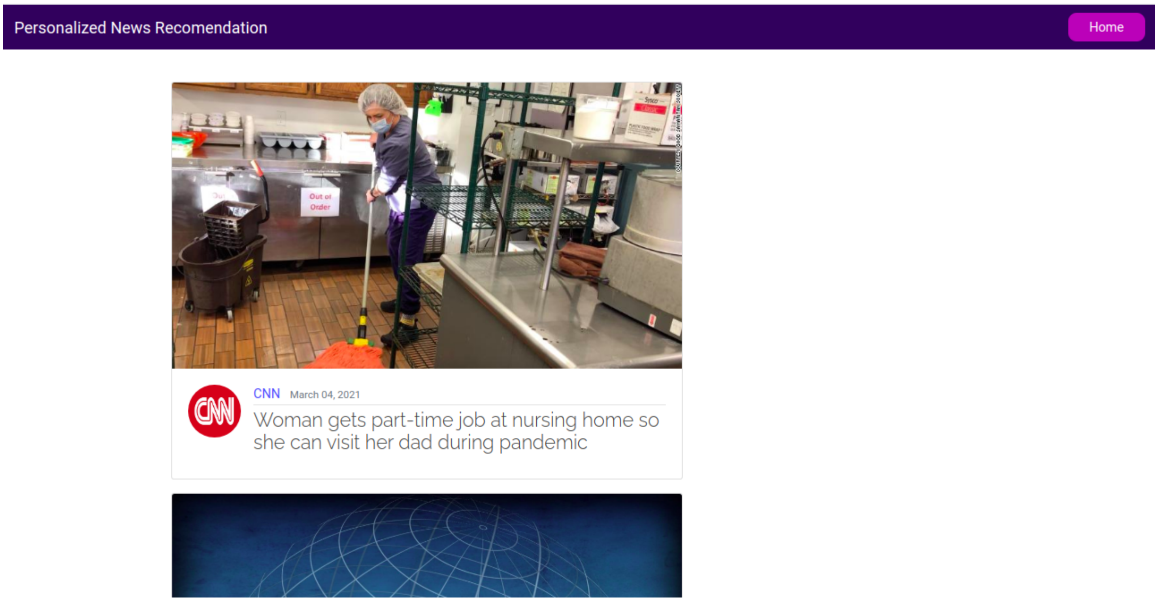


Figure 5.17 Final Recommendations in the UI

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 PERFORMANCE METRICS

6.1.1 Diversity

Diversity is defined as the average dissimilarity between news items that are recommended to a given user.

$$\text{diversity} = \Sigma_{ni\in N}\Sigma_{nj\in N, ni\neq nj}(1 - \text{Sim}(n_i, n_j))$$

Equation No. 6.1 Diversity

The Average Diversity-Score of the proposed system is obtained is 0.7312

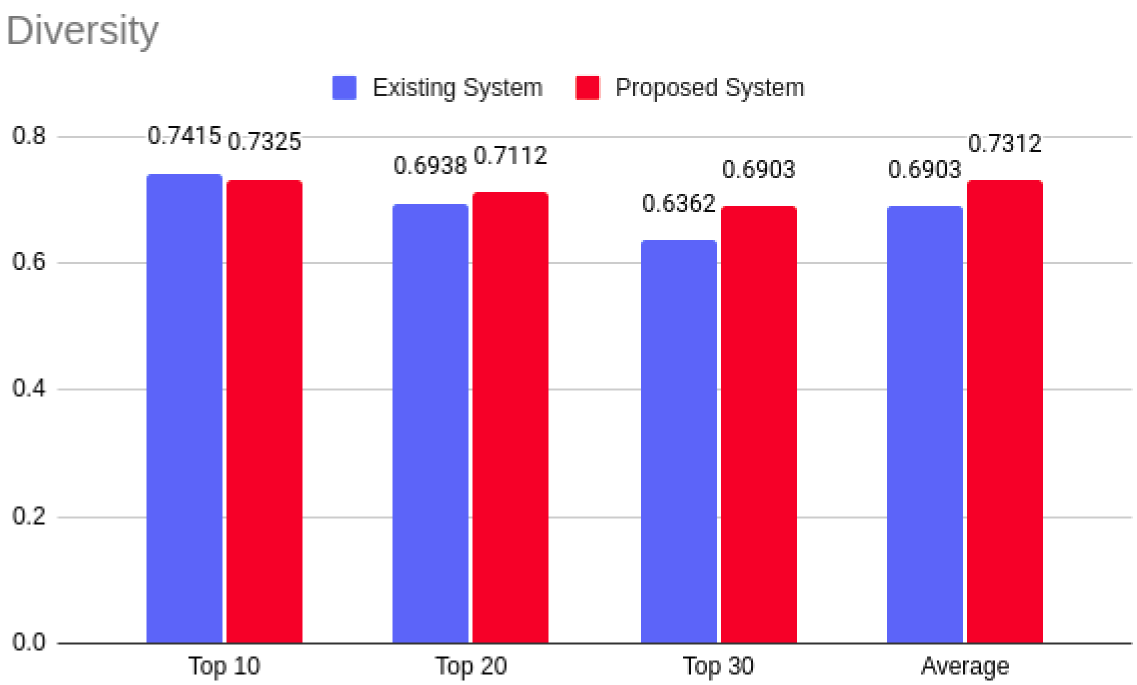


Figure 6.1 Diversity Score Comparison Graph

METHODS	TOP @ 10	TOP @ 20	TOP @ 30	AVERAGE
Existing System	0.7415	0.6938	0.6362	0.6903
Proposed System	0.7325	0.7112	0.6903	0.7312

Table 6.1 Diversity Score Evaluation

ALL CASES	RELEVANT MEANING
TRUE POSITIVE	News is relevant to the user and is recommended
TRUE NEGATIVE	News that is related to the user but not recommended
FALSE POSITIVE	News is recommended to the user but is not relevant
FALSE NEGATIVE	News that is not related to the user and is not recommended

Table 6.2 Confusion Matrix Cases - 1

	RECOMMENDED	RELEVANT
TRUE POSITIVE	YES	YES
TRUE NEGATIVE	NO	YES
FALSE POSITIVE	YES	NO
FALSE NEGATIVE	NO	NO

Table 6.3 Confusion Matrix Cases - 2

6.1.2 Precision

Precision is defined as the portion of recommended items that is in fact relevant to the user.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Equation No. 6.2 Precision

The Precision Score obtained in the proposed system is obtained is 0.6522

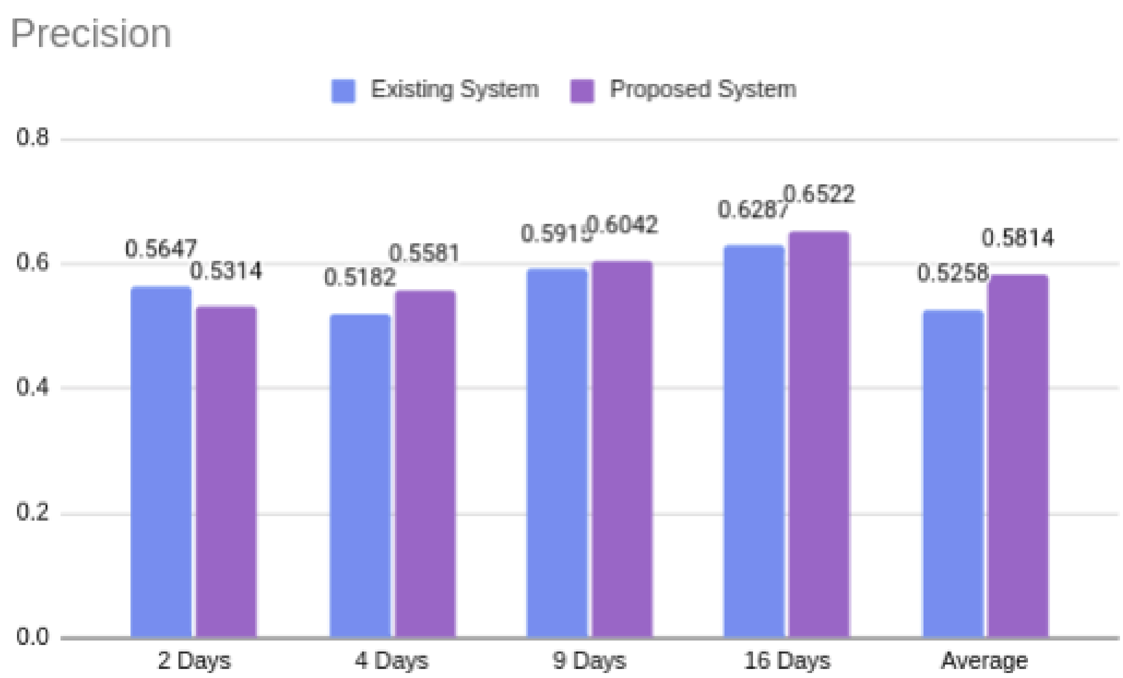


Figure 6.2 Precision Score Comparison Graph

6.1.3 Recall

Recall is defined as the portion of relevant items that is recommended to the active user .

$$\text{Recall} = \frac{TP}{TP + FN}$$

Equation No. 6.3 Recall

The Recall Score obtained in the proposed system is 0.8205

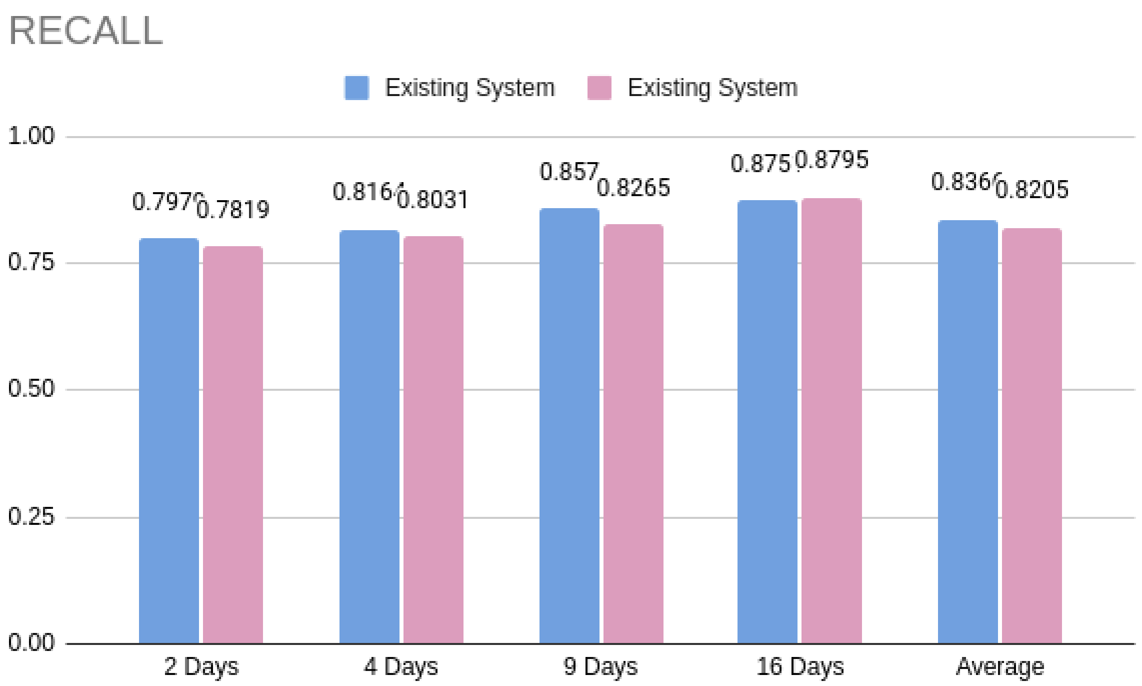


Figure 6.3 Recall Score Comparison Graph

6.1.4 F1-Score

F1-Score is defined as the harmonic mean of precision and recall.

$$\text{F1 - Score} = 2 \times \frac{\text{Precision X Recall}}{\text{Precision + Recall}}$$

Equation No. 6.4 F1-Score

The Highest F1- Score recorded in the proposed system is 0.7489.

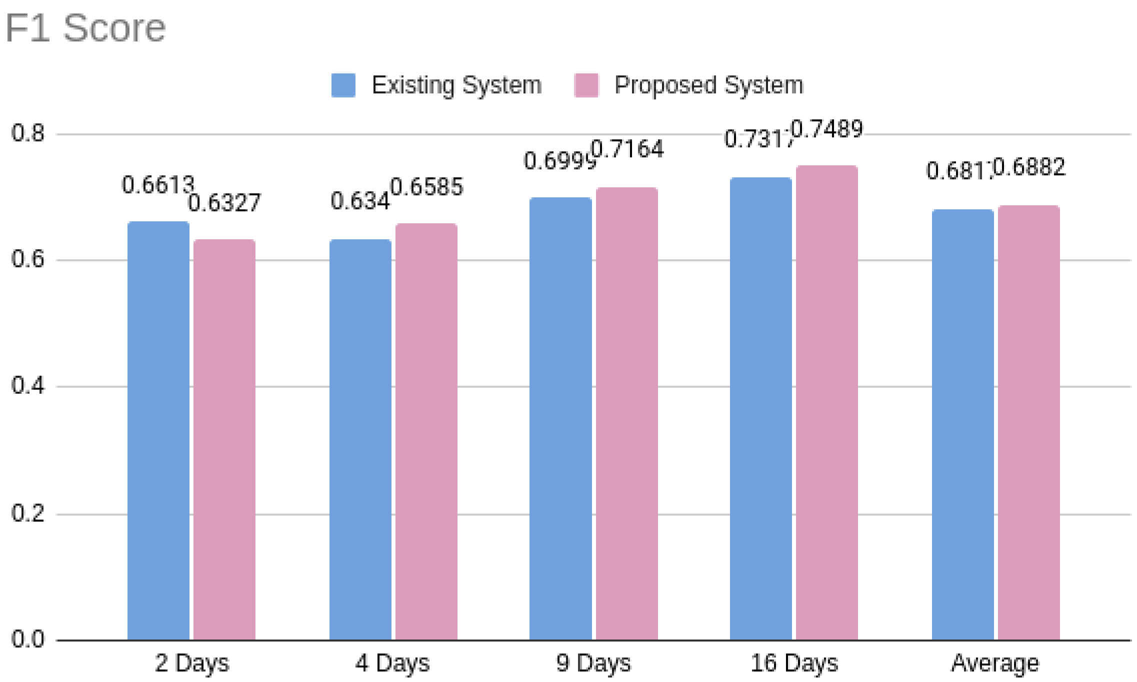


Figure 6.4 F1- Score Comparison Graph

6.2 FLANN PERFORMANCE

6.2.1 Time vs Branching

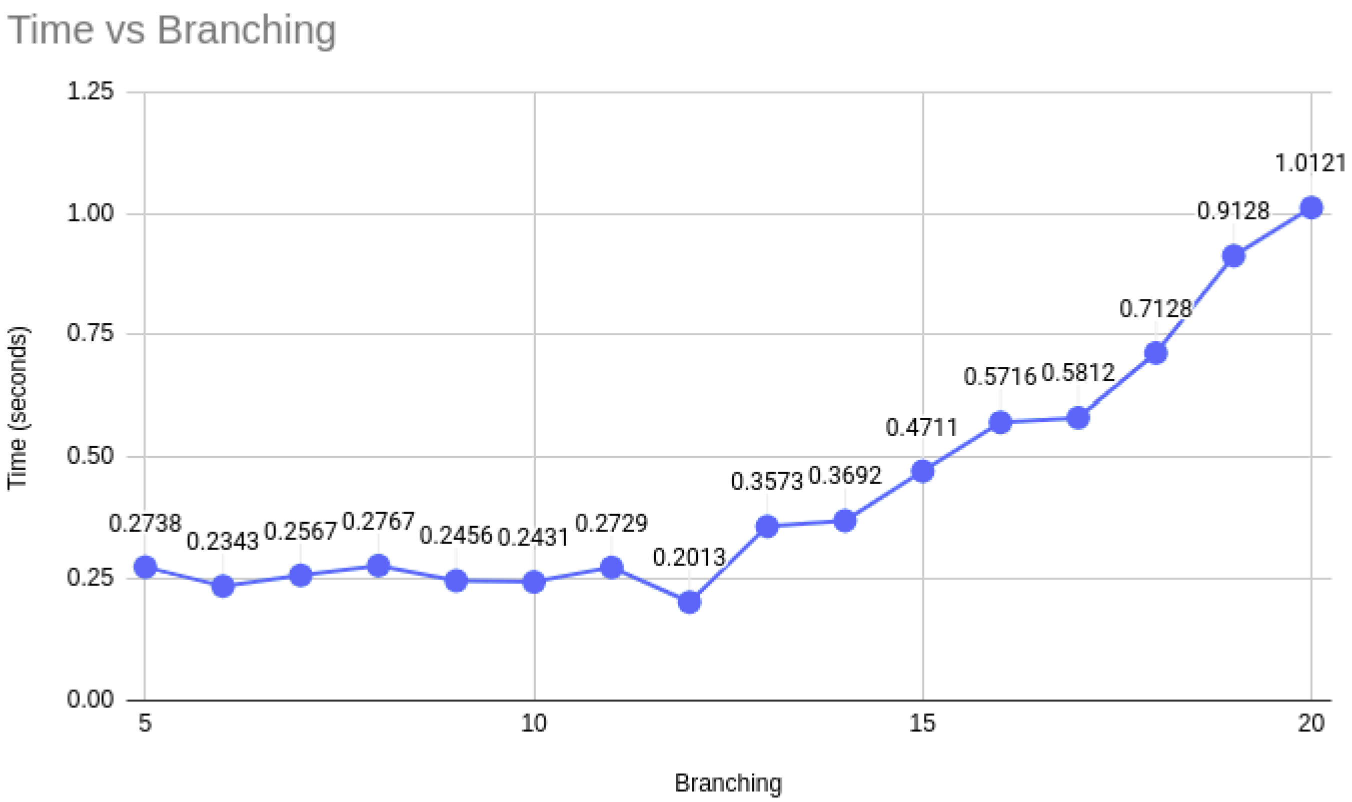


Figure 6.5 Time vs Branching Graph

The time taken to find the nearest neighbour using FLANN is plotted against the branching parameter the resultant graph is shown in Figure 6.5 . It can be seen that the minimum time i.e Nearest Neighbours are found faster when branching parameter is 12 with time duration of 0.2013 seconds (approx).

6.2.2 Time vs Checks

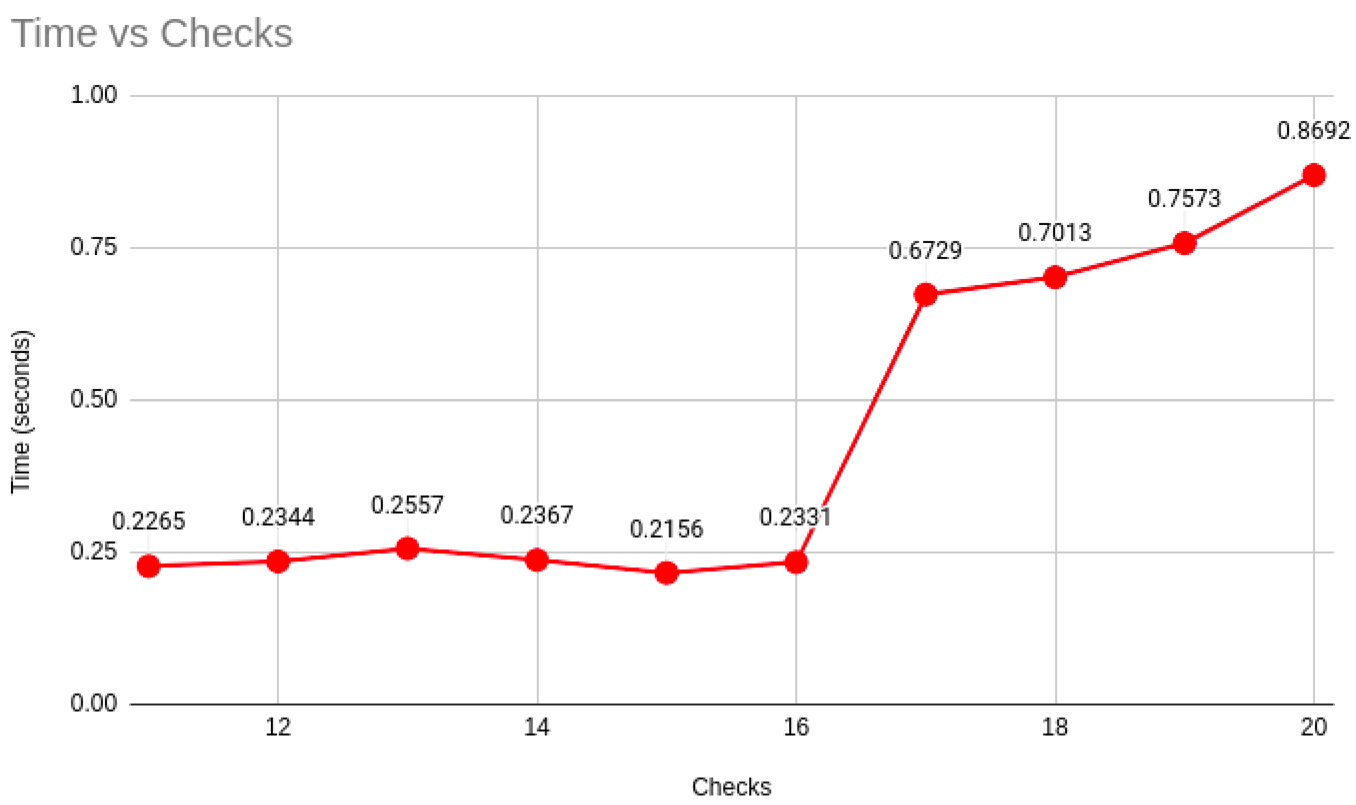


Figure 6.6 Time vs Checks Graph

The time taken to find the nearest neighbour using FLANN is plotted against the Checks parameter and the resultant graph is shown in Figure 6.6 . It can be seen that the minimum time to find the nearest neighbour i.e Nearest Neighbours are found faster when Checks parameter is 15 with time duration of 0.2156 seconds (apprx), but as increasing the number of checks improvises the precision of FLANN in finding the nearest neighbour 16 is chosen as the CHECKS parameter value.

6.3 DISCUSSION

The final news set to recommend by the proposed system has a significant diversity on topic categories. Multiple memberships in Ordered Clustering help to arrange news items in diverse distributions. The proposed model increased diversity in news recommendation based on the existing system. Table 6.1 shows diversity evaluation on the recommended news set by both the existing system and the proposed system. The main observation from the results is that increasing the recommended news set improved the diversity because news selection is performed within similar topic categories. Overall, HYPNER improved diversity on average by 5.80%.

CONCLUSION AND FUTURE WORK

CONCLUSION

News recommendation system is an automated approach built to provide the most appropriate information from the vast amount of data on the Internet. The main aim of a news recommendation system is to recommend news items that suit with the user's needs without manual exertion from the users.

This paper was set to address the cold start issue in news recommendation and at the same time to improve accuracy in news recommendation by highlighting the issues of clustering, news and user modelling, news rating, and news selection.The results has shown that the proposed model has achieved 5.80% improvement in terms of diversity of the news and 2.50% improvement in terms of F1-Score.The solutions can be further investigated on other items of recommendation systems such as music, video or documents.

FUTURE WORKS

The Future works include making the system suitable for other items such as music , videos, images posts etc. The current system has taken 1013 users and 4203 news items into processing.

The future aim is to increase the number of users and the number of users and news count into large scale or implement multiple instances of the system for each news channel or each news category and connect all the instances by some means of connection to make recommendation to users across different instances.

REFERENCES

[1] A. Darvishi, H. Ibrahim, F. Sidi and A. Mustapha, "HYPNER: A Hybrid Approach for Personalized News Recommendation," in IEEE Access, vol. 8, pp. 46877-46894, 2020.

[2] C. Feng, M. Khan, A. U. Rahman and A. Ahmad, "News Recommendation Systems - Accomplishments, Challenges & Future Directions," in IEEE Access, vol. 8, pp. 16702-16725, 2020.

[3] D. Wu, M. Zhang, C. Shen, Z. Huang and M. Gu, "BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery," in IEEE Access, vol. 8, pp. 32215-32225, 2020.

[4] D. Wu, M. Zhang, C. Shen, Z. Huang and M. Gu, "BTM and GloVe Similarity Linear Fusion-Based Short Text Clustering Algorithm for Microblog Hot Topic Discovery," in IEEE Access, vol. 8, pp. 32215-32225, 2020.

[5] G. De Souza Pereira Moreira, “CHAMELEON: A meta architecture for news recommender systems,” in Proc. 12th ACM Conf. Recommender System. (RecSys), 2018, pp. 578-583.

[6] A. S. Das, M. Datar, A. Garg, and S. Rajaram, ‘‘Google news personalization: Scalable online collaborative filtering,’’ in Proc. 16th Int. Conf. World Wide Web (WWW), 2017, pp. 271–280.

[7] D. Khattar, V. Kumar, M. Gupta, and V. Varma, “Personalized news recommendation: A review and an experimental investigation,” in Proc. NewsIR Workshop, 2018, pp. 45–50.

[8] L. Zheng, L. Li, W. Hong, and T. Li, “PENETRATE: Personalized news recommendation using ensemble hierarchical clustering,” Expert Syst. Appl., vol. 40, no. 6, pp. 2127–2136, May 2013.

[9] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan, “SCENE: A scalable two-stage personalized news recommendation system,” in Proc. 34th. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2011, pp. 125–134.