

**DOKUZ EYLÜL UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**PUBMED ARTICLE RECOMMENDATION  
SYSTEM BASED ON COLLABORATIVE  
FILTERING**



**by**

**Mohammad Osama Salahaldeen BARAKAT**

**February, 2020**

**İZMİR**

**PUBMED ARTICLE RECOMMENDATION  
SYSTEM BASED ON COLLABORATIVE  
FILTERING**

**A Thesis Submitted to the  
Graduate School of Natural and Applied Sciences of Dokuz Eylül University  
In Partial Fulfillment of the Requirements for the Degree of Master of Science  
in Computer Engineering, Computer Engineering Program**

**by**

**Mohammad Osama Salahaldeen BARAKAT**

**February, 2020**

**İZMİR**

## M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “PUBMED ARTICLE RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING” completed by MOHAMMAD OSAMA SALAHALDEEN BARAKAT under supervision of ASSOC.PROF.DR.ADİL ALPKOÇAK and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Assoc.Prof.Dr.Adil ALPKOÇAK

Supervisor



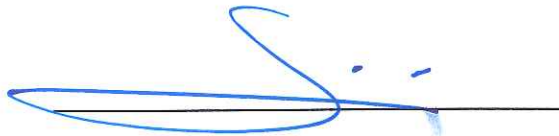
Prof. Dr. Alp KURT

(Jury Member)



Dr. Öğr. Üyesi PELİN YILDIRIM TAŞER

(Jury Member)



Prof.Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

## ACKNOWLEDGEMENTS

Foremost, I would like to thank my supervisor Assoc. Prof. Dr. Adil Alpkoçak Department of Computer Engineering, Dokuz Eylul University. As a foreign student who was living far away from his family and his hometown Assoc. Prof. Dr. Adil Alpkoçak was my family before being my supervisor. At the professional side his wide knowledge, experience, and personal guidance helped me constantly while writing my thesis. It has been a pleasure working under his supervision.

I would also like to thank my parents, and my brothers for this accomplishment, without their continuous spiritual support I wouldn't have been able to do it.

Mohammad Osama Salahladeen BARAKAT

# PUBMED ARTICLE RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING

## ABSTRACT

PubMed is one of the largest public databases on biological and medical sciences, it contains more than 30 million biomedical articles cited from several resources such as online books, conferences, and journals, the biggest percentage of citations comes from MEDLINE. In addition to the current articles, PubMed is being updated on a daily basis with new articles. Researchers are finding it very hard to cope with exponentially increasing numbers of biomedical literature, for that reason there is a need to design a recommendation system that helps researchers in finding materials that are relevant to them.

In this study we proposed a PubMed article recommendation system, PubGate. Our recommendation system is based on a hybrid approach using both content-based and collaborative approach with focus on the latter. For the collaborative filtering approach, we have used Jaccard similarity to compute the similarities between the users according to their liked articles and their keywords of interest, where we recommended articles that have been liked the most by similar users. Collaborative filtering usually suffers from the cold start problem, which is related to new users who have zero history. To overcome this problem, we integrated Elasticsearch engine to recommend articles to users based on their given keywords of interest. This thesis combines both content-based and collaborative approaches to recommend PubMed articles to the users.

**Keywords:** PubMed articles, recommender systems, collaborative filtering, content based, Jaccard similarity index, Elasticsearch

# İŞBİRLİKÇİ FİLTRELEME İLE PUBMED MAKALE ÖNERİ SİSTEMİ

## ÖZ

PubMed, her gün binlerce yeni makale ile güncellenen biyolojik ve tıp bilimleri hakkındaki en büyük erişime açık veri tabanlarından biridir. Ayrıca PubMed, MEDLINE, yaşam bilimleri dergileri ve çevrimiçi kitaplardan biyomedikal literatür için 30 milyondan fazla alıntı içeriyor. Araştırmacılar, artan sayıda biyomedikal literatürlerinde istedikleri yayınları bulmakta zorlanmaktadır. Bu nedenle, araştırmacıların özellikle ilgilendikleri materyalleri bulmalarına yardımcı olan bir öneri sistemi tasarlamak önemlidir.

Bu çalışmada PubGate isimli bir PubMed makale öneri sistemi geliştirdik. Öneri sistemimiz, içeriğe dayalı yaklaşımla birlikte daha çok işbirlikçi yaklaşımı kullanan karma bir yaklaşıma dayanmaktadır. İşbirliğine dayalı filtreleme yaklaşımı için, kullanıcılar arasında ilgilendikleri makalelere ve ilgi alanlarına göre anahtar kelimeleri kullanarak benzerlikleri hesaplamak için Jaccard benzerliği özneliğini kullandık ve benzer ilgi alanlarına sahip kullanıcılara ilgilenebilecekleri makaleleri tavsiye ettik. İşbirlikçi filtreleme, genellikle geçmiş olmayan yeni kullanıcılarla ilgili soğuk başlatma sorunundan muzdariptir. Bu sorunun üstesinden gelmek için, kullanıcılara anahtar kelimeleri temel alarak makaleler önermek için Elasticsearch motorunu entegre ettik. Bu tez, kullanıcılara PubMed makaleleri önermek için hem arama motorunu hem de işbirliği yaklaşımınıs birleştirir.

**Anahtar Kelimeler:** PubMed makaleleri, öneri sistemleri, işbirlikçi filtreleme yaklaşımı, içeriğe dayalı yaklaşım, Jaccard benzerlik endeksi, Elasticsearch

## CONTENTS

	<b>Page</b>
M.Sc. THESIS EXAMINATION RESULT FORM.....	ii
ACKNOWLEDGEMENTS .....	iii
ABSTRACT .....	iv
ÖZ .....	v
LIST OF FIGURES .....	viii
LIST OF TABLES .....	ix
<b>CHAPTER ONE - INTRODUCTION .....</b>	<b>1</b>
1.1 General .....	1
1.2 Contribution of thesis .....	2
1.3 Organization of the thesis .....	3
<b>CHAPTER TWO - LITERATURE REVIEW .....</b>	<b>4</b>
2.1 Recommender Systems .....	4
2.2 Content-Based Approach .....	4
2.3 Collaborative Filtering Approach.....	7
2.3.1 Memory-based collaborative filtering .....	8
2.3.2 Model-based collaborative filtering .....	9
2.3.3 Cold Start Problem .....	9
2.3.4 Related Work.....	10
2.4 Hybrid Approach .....	11
2.5 Similarity Measures.....	12
<b>CHAPTER THREE - METHOD.....</b>	<b>13</b>
3.1 System Overview .....	14
3.2 Data Collection.....	15

3.3 PubGate .....	16
3.4 Calculate similarities between the users.....	20
3.5 Articles Recommendation .....	22
3.5.1 Neighbor users.....	22
3.5.2 Elasticsearch Engine.....	25
<b>CHAPTER FOUR – EXPERIMENTS AND RESULTS.....</b>	<b>27</b>
4.1 Evaluation.....	27
4.2 Creating Benchmark Datasets .....	29
4.3 Results .....	31
4.4 Evaluation.....	33
<b>CHAPTER FIVE – CONCLUSION AND FUTURE WORK.....</b>	<b>35</b>
<b>REFERENCES.....</b>	<b>37</b>



## LIST OF FIGURES

	<b>Page</b>
Figure 2.1 Content-based approach.....	5
Figure 2.2 User-item matrix.....	7
Figure 2.3 Memory-based collaborative filtering approach.....	8
Figure 3.1 The design of our proposed model .....	14
Figure 3.2 Entity-Relationship Diagram for PubGate database .....	15
Figure 3.3 Homepage screen for PubGate .....	16
Figure 3.4 Keywords of interest screen.....	17
Figure 3.5 Screenshot for a dummy profile from the system.....	18
Figure 3.6 Screenshot for the HomePage screen .....	19
Figure 3.7 Screenshot for an article .....	20
Figure 3.8 Entity-Relationship Diagram between users, articles, and keywords.....	21
Figure 3.9 Similarity matrix.....	22
Figure 3.10 Example showing neighbor users .....	23
Figure 3.11 Screenshot for collaborative table in the database.....	24
Figure 3.12 Screenshot for recommended articles section at Homepage .....	25
Figure 3.13 Integration of Elasticsearch with our system.....	26
Figure 4.1 Confusion Matrix.....	27
Figure 4.2 The list of users we have created.....	29
Figure 4.3 Group one representation.....	30
Figure 4.4 Group two representation .....	30
Figure 4.5 Group three representation .....	31
Figure 4.6 User's matrix similarity .....	32
Figure 4.7 Adjusted user's matrix similarity.....	32
Figure 4.8 List of the recommended articles.....	33

## LIST OF TABLES

	<b>Page</b>
Table 2.1 User-item matrix with ratings .....	9
Table 4.1 Annotating the articles that have been liked by the users .....	33



# CHAPTER ONE

## INTRODUCTION

### 1.1 General

Over the last twenty years, the number of people using smartphones, tablets, or computers have increased rapidly, that increase was accompanied was an increase in the number of the applications created such as Facebook, Twitter, Instagram, Spotify, Amazon, Netflix, and many other applications. People using their devices can access whatever they want, spend hundreds of hours whether in searching for what interests them, or upload contents that express them. Scientific literature also had a part in the technology revolution, the number of published researches has increased dramatically, the internet and open source tools made it much easier for researchers to conduct their researchers and studies, one of those databases that witnessed that increase is PubMed. PubMed contains more than 30 million biomedical articles cited from several resources such as online books, confrences, and journals, the biggest percentage of citations comes from MEDLINE. The cited articles are usually not in the full-text format, most of the articles are only presented with their abstract, however articles usually contains direct links for their original resource which contains the full text.

PubMed is being updated daily with thousands of new articles. Unfortunately, researchers are not being able to cope with that dramatic increase in the number of new articles. Exploring PubMed is becoming an exhausting task for them due to the huge volume of data they have to go through or the huge amount of time they have to spend to find what they are looking for.

The motivation of this thesis comes here to use the super powers of Artificial Intelligence (AI). AI approaches are being widely used nowadays in the research areas of information filtering systems, text mining, and information retrieval. A great example of artificial intelligence approaches are recommender systems. Recommender systems are engines that aims to recommend items to the users that are related to them, items can be movies in a movie domain, books in a book domain, or products in an online selling website.

Recommender systems is not a new innovation, they have been applied in many different domains wither in the social network applications, or e-commerce applications. They are categorized mainly into three main approaches, *content-based approach* where the focus is on the characteristics of the items, *collaborative filtering approach* where the focus is at finding similar users who share the same taste, and finally the last approach, *hybrid approach* which combines more than one approach at the same time. We will speak about these three approaches more briefly in the later section.

One of the biggest drawbacks of collaborative filtering approach is the cold start problem. Cold start problem describes a problem of recommendation for new user, where there is no personnel network. A possible solution to the cold start problem is to use content-based filtering. Therefore, we integrated our approach with Elasticsearch engine to recommend articles to the new users based on their entered keywords. Elasticsearch engine was used as an external tool and there is no attention for us to include it in the calculations nor the evaluation part.

To evaluate our system, we have created 10 users, assigned keywords to them, and added articles to their liked articles lists which enabled us to calculate similarities between them, finally we recommend articles for the neighbor users. Then, we calculated the recall, precision, and f-measure metrics for the articles we have recommended, the outstanding value for precision metrics indicates that all of the articles we have recommended were relevant for the users.

## **1.2 Contribution of thesis**

In this thesis, we have proposed a system, which we called PubGate and it recommends users PubMed article based on collaborative filtering approach. We have developed a user-friendly web interface, where researchers can add their keywords of interest, follow other users, and like their articles of interest. Users are presented as a set of keywords and likes in our system, we calculated the similarities between them based on their liked articles and their keywords of interest. Finally, we have recommended the articles that have been liked the most among the similar users.

### **1.3 Organization of the thesis**

This thesis includes five chapters, they are organized as follows: Chapter 2 presents a literature review about recommender systems, and the related work that has been conducted using the three different approaches, content-based approach, collaborative filtering approach, and hybrid approach. Chapter 3 gives the details of our system PubGate, it includes which technologies we have used, how we have collected the data, how did we calculate similarities between the users, and finally how we recommended articles to the users. Chapter 4 presents the experiments we have conducted in addition to their results, finally Chapter 5 covers the conclusion of our proposed model and the future works that might be conducted.

## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2.1 Recommender Systems**

Recommender systems aim to recommend items that are related to the users based either on their previous history, or on users that share with them the same taste (Hristakeva, et al., 2017) they have been applied in several domains, whether in the e-commerce domain such as Amazon, Ebay, Aliexpress, Spotify, and Netflix, or in the social network domain such as Facebook, Twitter, and Instagram (Liu, Hu, Mian, Tian, & Zhu, 2014). Recommender systems in the research-paper field is not a new field nor a new study, in a literature survey conducted (Beel, Gipp, Langer, & Breitinger, 2016) the first research-paper recommender system was introduced in 1998 for a CiteSeer project, since that time till now there have been many articles published regarding research-paper recommendation approaches. What type of data is being collected, how it's being collected, and how it's being used, determines the approach of the recommender system. Recommender systems can be classified into three main approaches, content-based approach, collaborative filtering approach, and hybrid approach, described in sections 2.2, 2.3, and 2.4 respectively.

#### **2.2 Content-Based Approach**

One of the main approaches for recommender systems is the content-based approach, which recommends similar items to the user based on his/her previous likes or purchase history. In content-based approach the features or the characteristics of the items are extracted and compared with the profile of the user, for example in the movie domain the genres of the previously liked movies of the user are extracted, movies that belong to the same genre are recommended to the user, or in the book domain, the user might be recommended books that belong to the same authors of the books he liked in the past. Figure 2.1 shows how content-based approach works. In the first step the previous history of the user is fetched, where transactions can be as a form of like, watch, read, listen, or purchase. In the second step the similarities between the items are calculated, and items with the highest similarities are recommended to the user.

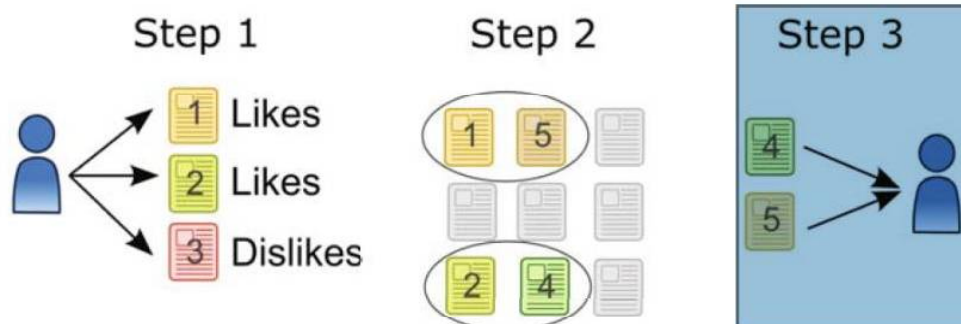


Figure 2.1 Content-based approach (Borges & Lorena, 2010)

Content-based approach is mostly used in the text-based domains such as in the news domain, book domain, and articles domain where the content is simply a text (Swapnil, 2012). There have been many content-based techniques applied for the text-based domains, the most popular one was the term frequency – inverse document frequency (TF-IDF).

Before introducing the term TF-IDF, the term TF should be introduced first. Assuming that we have a set of documents which it will be referred to as  $D$ , for every document  $d$  that belongs to that set  $D$ , all the terms are being extracted from the documents and an index vocabulary is being built. Terms which are frequently used like “is, of, the ...” which they are referred to as stop words are excluded from the index vocabulary because they exist in all the documents, therefore their value is really not important.

For every term  $t$  that belongs to the index vocabulary its term frequency (TF) for document  $d$  is simply calculated by counting how many times it appears in that document  $d$  and it is presented as  $tf(t, d)$ . Unfortunately, the drawback from relying on the term frequency technique is that it gives more importance for the frequently used terms, and less importance for the rare terms, to overcome this issue comes the term TF-IDF.

Inverse document frequency for term  $t$  is calculated as follows:

$$idf(t) = \log \frac{|D|}{d:t \in d} \quad (2.1)$$

Where  $|D|$  is the total number of documents in the set, and  $d:t \in d$  is the number of documents that contain the term  $t$ . Finally, TF-IDF( $t$ ) for term  $t$  is calculated as follows:

$$tf - idf(t) = tf(t, d) \times idf(t) \quad (2.2)$$

TF-IDF treats the term  $t$  globally and measures its importance within the collection rather than isolating it and treating it locally. To find the similarity between documents, documents are transformed in the form of vector space where their scalars are the values of the TF-IDF for the terms in the index vocabulary. After the vector space presentation, the distance between the vectors can be measured using one of the similarity measurements in section 2.5, documents with close distance means they are similar and are recommended to the user.

PURE, a PubMed article recommendation system based on content-based filtering developed by (Yoneya & Mamitsuka, 2007) in their recommender system they relied on the user's explicit feedback by asking users to select their favorite articles after registration. Authors used the tf-idf technique in addition to a learning probabilistic model on the preferred articles selected by the user to recommend the highly rated articles by the model, tf-idf technique was also used in Science Concierge (Achakulvisut, Acuna, Ruangrong, & Kording, 2016) a content-based recommendation system for literature search, their proposed model uses the votes of the users, users can determine whether a document is relevant or irrelevant to them, their proposed model used the tf-idf technique and topic modeling, tf-idf was used for vector presentations of the documents, where for topic modeling the latent semantic analysis (LSA) approach has been used. They tested their model on 15K scientific posters from the Society of Neuroscience Conference 2015.



Unlike the two previous studies (Kompan & Bielikova, 2010) in their proposed model they relied on the implicit feedback of the user, they have developed a content based recommender system for news domain, their vector article presentation was based on a several techniques such as term frequency and TF-IDF. The user model was created by extracting the logs of the user and analyzing the history of the previous visited articles. They have tested their proposed model over 10000 articles from the Slovak news portal SME.SK.

### 2.3 Collaborative Filtering Approach

The second main approach for the recommender systems is the collaborative filtering approach, it has become the most widely used approach for recommending items for user (Haifeng Liu, Zheng Hu, Ahmad Mian, Hui Tian, & Xuzhen Zhu, 2014) unlike the content based approach the collaborative filtering approach is not concerned with the characteristics nor the features of the items (Wei, He, Chen, Zhou, & Tang, 2017), the concept behind collaborative filtering approach is that users who had the same taste in the past will probably have the same taste in the future too (Cheng, Yin, Dong, Dong, & Zhang, 2016). Usually in the collaborative filtering approach the system is presented as a matrix where the users are represented as the rows, and the items are represented as the columns, and each cell in user-item matrix corresponds to a vote or like done by the user to the item, figure 2.2 shows an example for the user-item matrix for a movie domain, where the cells corresponds to ratings given by the users to the movies.

	Item 1	Item 2	Item 3	...	Item n
User 1	2	3	?	...	5
User 2	?	4	3	...	?
User 3	3	2	?	...	3
...	...	...	...	...	...
User m	1	?	5	...	4

Figure 2.2 User-Item matrix

Collaborative filtering approach can be categorized into two main approaches, memory-based collaborative filtering, and model-based collaborative filtering.

### 2.3.1 Memory-based collaborative filtering

Memory-based technique is somehow similar to the method used in content-based approach except that in the former we are not dealing with the features nor the characteristics of the items, in memory-based technique the transactions of the users are being collected, they can be in the form of clicks, votes, or likes, figure 2.3 shows how memory-based approach works, the previous history of the users is fetched in the first step, in the second step similarities between the users are being calculated using one of the similarity measures mentioned in section 2.5, close or neighbor users will be detected, finally in the third step items from the neighbor users will be recommended to the active user.

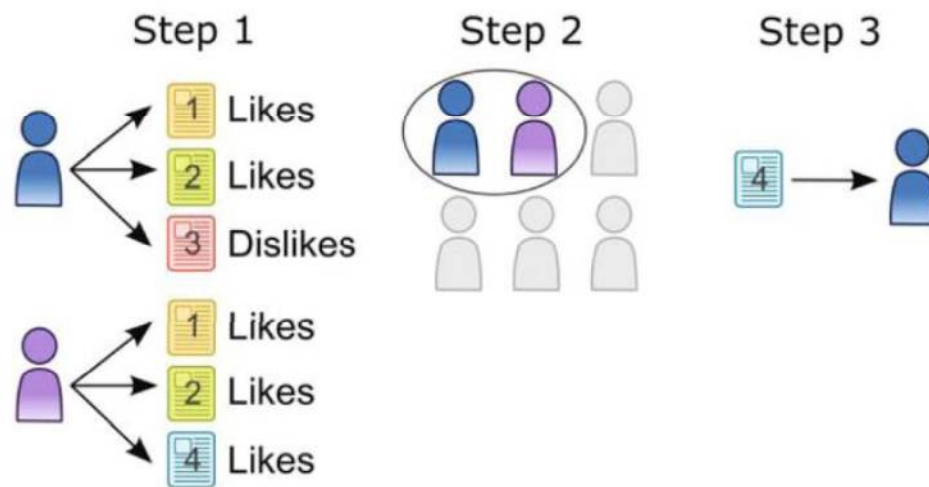


Figure 2.3 Memory-based collaborative filtering approach (Borges & Lorena, 2010)

The memory-based method is considered more accurate, its only drawback that with the increasing number of users or items the computing time will grow as well too. (Liu, Hu, Mian, Tian, & Zhu, 2014).

### 2.3.2 Model-based collaborative filtering

Unlike the memory-based approach where similarities are being calculated between the users to find neighbor users and then recommend their items for an active user, the mode-based approach tries to build a learning model using the previous ratings of the user to predict ratings for items the user didn't encounter with before, in other words the approach tries to predict the empty cells of table of 2.1.

Table 2.1 User-Item matrix with ratings

User / items	Item 1	Item 2	Item 3	Item n-1	Item n
User 1	5	5	?	5	1
User 2	2	1	1	4	2
User 3	3	4	2	3	3
User m-1	2	?	2	4	4
User m	1	1	1	2	4

Model-based approach relies on machine learning, and data mining techniques, of those popular techniques are Bayesian networks, Singular value decomposition (SVD), clustering models, decision trees, and Probabilistic latent semantic analysis (PLCA).

Despite that the memory-based approach is considered more accurate than the model-based approach, the model-based approach is way much faster than the memory-based approach were the process is being executed offline, and the predictions are created within a short period (Liu, Hu, Mian, Tian, & Zhu, 2014).

### 2.3.3 Cold Start Problem

Before proceeding with the related work section, there is a need to explain about one of the major drawbacks that the collaborative filtering approach suffers from, the cold start problem. The cold start problem occurs when the user or the item is new, that's why they refer to them with the term cold, new users or cold users usually have zero history or zero transactions with the system, which makes it hard for the model to

build a profile for them, or calculate the similarity between them and the rest of the users in the system.

#### ***2.3.4 Related Work***

Several studies have been conducted using the collaborative filtering approach, authors in (Sahoo, Pradhan, Barik, & Dubey, 2019) proposed a health recommender system based on item based collaborative filtering, they tested their proposed model on a 10k patients, patient's rating ranged from 1-5, their results showed that the values they got for root square mean error, and mean absolute error were much better when compared with other approaches.

The cold start problem which presents the major drawback that the collaborative filtering approach suffers from, was the center of the attention for some studies, in their proposed model (Bobadilla, Ortega, Hernando, & Bernal, 2012) they proposed a new similarity measure model by combining several simple similarity measurements, each similarity measure had a scalar associated with it, the value of scalars were determined by using neural networks, for the experiments part they have tested their proposed model on Netflix and Movielens databases, the results showed a good improvement in the mean of accuracy, precision and recall, also (Wei, He, Chen, Zhou, & Tang, 2017) they were able to address the recommendation problems for the cold start problem, their models combined a time-aware collaborative filtering (CF) model timeSVD++ with a deep learning architecture SDAE. The deep learning neural network SDAE is responsible for the extraction of item content features, while the timeSVD++ model is responsible for prediction of unknown ratings.

Other studies dealt with the disadvantages of the existing similarity measures, such as cosine, Pearson correlation coefficient, and mean squared difference, authors in (Liu, Hu, Mian, Tian, & Zhu, 2014) they have proposed a new similarity approach which is based on the proximity, impact, and popularity measure, known as PIP measure. Their proposed model had a better result when compared with the regular similarity measures.

Time was an important factor in some studies, in their model (Cheng, Yin, Dong, Dong, & Zhang, 2016) they calculated the similarities between the users using users' interest sequences. Interest Sequence of the user is described as the rating given by the user for items over different intervals of time. They assumed that users who have longer LCSIS (Longest Common Sub-IS) and more ACSIS (All Common Sub-IS) should also have more similarity in their preferences, while (Yingyuan, Pengqiang, Hsu, Hongya, & Xu, 2015) they proposed a time-ordered collaborative filtering recommendation algorithm (TOCF), which takes the time sequence characteristic of user behaviors into account. Besides, a new method to compute the similarity among different users, named time-dependent similarity, is proposed.

## **2.4 Hybrid Approach**

The last approach of the recommender systems is the hybrid approach which combines both of the previously mentioned approaches together, content-based approach with collaborative filtering approach. The purpose of this approach is to overcome the disadvantages of solely relying at one approach, for example (Nilashi, Ibrahim, & Bagherifard, 2018) they proposed a hybrid recommendation model based on collaborative filtering technique, in order to improve the scalability of their model they used the singular value decomposition as a dimensionality reduction technique which helps to find the most similar items and users in each cluster, and in order to improve the accuracy of the model they have used ontology.

In a different study (Hristakeva, et al., 2017) they showed how to minimise the cold start problem for collaborative approach by combining implicit feedback with collaborative approach. In their proposed model implicit feedback comes from the user's interactions, such as users adding documents to their personal libraries which allowed the model to calculate the similarities between the users according to what they have in their libraries. (Pessemier, Leroux, Vanhecke, & Martens, 2015) they have developed a hybrid recommender system for news domain, for the content-based part they used Lucene which was mainly a search engine, and for the collaborative filtering part they have used Mahout to exchange profile terms among neighboring users.

The inspiration for our contribution came from the previously mentioned studies, our contribution can be summarized as:

- Explicit user feedback: we have developed a user-friendly web application so researchers can add their keywords of interest, and like the articles they are interested in.
- Collaborative approach: we calculated the similarities between the users based on their liked articles and their keywords of interest. We recommended the liked articles of the neighbor users.
- Cold start problem: new users were recommended articles based on their entered keywords, the integration of the search engine Elasticsearch as a content based tool overcame the cold start problem.

## 2.5 Similarity Measures

In the previous sections, mainly 2.2 and 2.3.1, we spoke about the steps of their approach, calculating similarities were a common step whether in the content-based approach or in the memory-based filtering approach. There are three main popular measures used to calculate the similarities, Pearson's correlation, Cosine similarity, and Jaccard's similarity (Agarwal & Chauhan, 2017). Pearson correlation can be used in the memory-based filtering approach, given user  $a$  and user  $b$  their Pearson's correlation can be calculated as follow:

$$Pearson(a, b) = \frac{\sum_{i \in S_a \cap S_b} (r_{ai} - \bar{r}_a) \times (r_{bi} - \bar{r}_b)}{\sqrt{\sum_{i \in S_a \cap S_b} (r_{ai} - \bar{r}_a)^2 \sum_{i \in S_a \cap S_b} (r_{bi} - \bar{r}_b)^2}} \quad (2.3)$$

Where  $S_a$  and  $S_b$  are the set of items evaluated by user  $a$  and  $b$  respectively,  $r_{ai}$  and  $r_{bi}$  represents the rating given by user  $a$  and  $b$  for item  $i$  respectively, finally  $\bar{r}_a$  and  $\bar{r}_b$  are the averages of the ratings made by user  $a$  and  $b$ .

On the other hand, the cosine similarity does not take into consideration the average of the user ratings, for user  $a$  and  $b$  the cosine similarity in the memory-based filtering approach can be calculated as follows:

$$\text{Cosine}(a, b) = \frac{\sum_{\{i \in S_a \cap S_b\}} (r_{ai} - \bar{r}_a) \times (r_{bi} - \bar{r}_b)}{\sqrt{\sum_{\{i \in S_a \cap S_b\}} r_{ai}^2 \sum_{\{i \in S_a \cap S_b\}} r_{bi}^2}} \quad (2.4)$$

Cosine similarity is being calculated in a different way in the content-based approach, after transforming the documents into the vector space, the cosine similarity between document  $a$  and  $b$  is being calculated by dividing the dot product of the vectors by the their magnitude as follows:

$$\text{Cosine } \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \quad (2.5)$$

In summary Cosine similarity will show how much two documents are close to each other based on their angle rather their magnitude. Unlike Pearson's correlation and Cosine similarity which they don't take into consideration the rating of the of items outside the intersection set, Jaccard's similarity considers the difference between the two sets of items, given two users  $a$  and  $b$ , Jaccard's similarity can be calculated as follows:

$$\text{Jaccard}(a, b) = \frac{|S_a \cap S_b|}{|S_a \cup S_b|} \quad (2.6)$$

## CHAPTER THREE

### METHOD

#### 3.1 System Overview

We have developed a PubMed article recommendation system that aims to recommend articles for users. Figure 3.1 shows a brief explanation for the system we have designed. In the first step, we downloaded the articles from PubMed (Section 3.2) In the second step we stored the data and the transactions of the users, these data are entered through the web application that we have designed, PubGate (Section 3.3), using that data that we have collected and stored in the database, we calculated the similarities between the user's in the third step (Section 3.4), Finally in the last step, we have recommended the articles that have been liked by the neighbor users (Section 3.5)

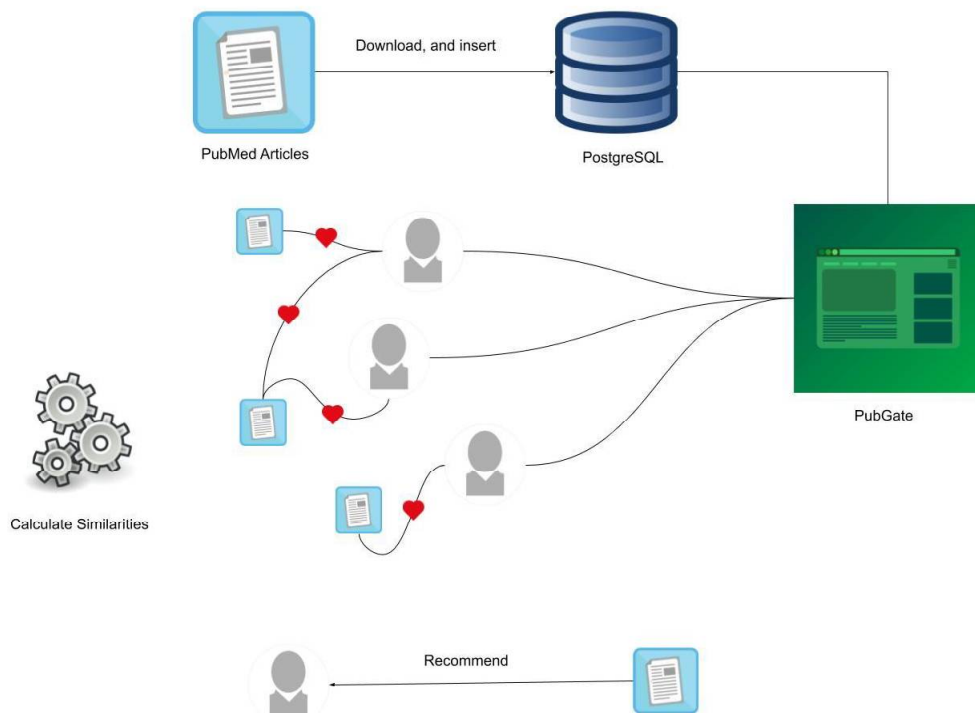


Figure 3.1 The design of our proposed model



### 3.2 Data Collection

To download the PubMed database, AsperaConnect (32) software was installed from the PubMed FTP page and the entire database was downloaded in .tar.gz file format. The OHDSI MedlineXmlToDatabase tool was launched to extract files downloaded via FTP from the .tar.gz compressed file format and transfer them to our local database, we excluded articles with empty abstracts, Figure 3.2 shows the Entity-Relationship Diagram for our database.

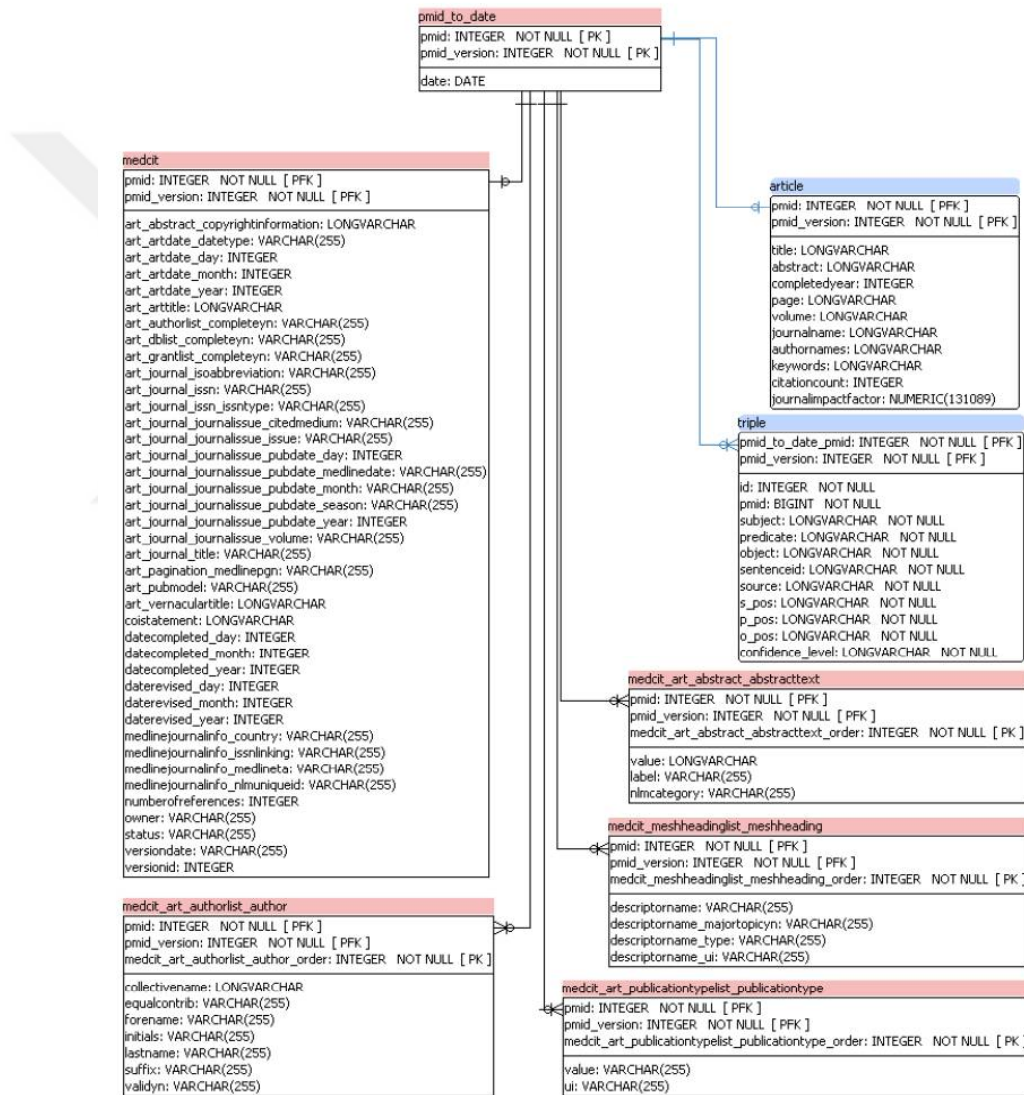


Figure 3.2 Entity-Relationship Diagram for PubGate database

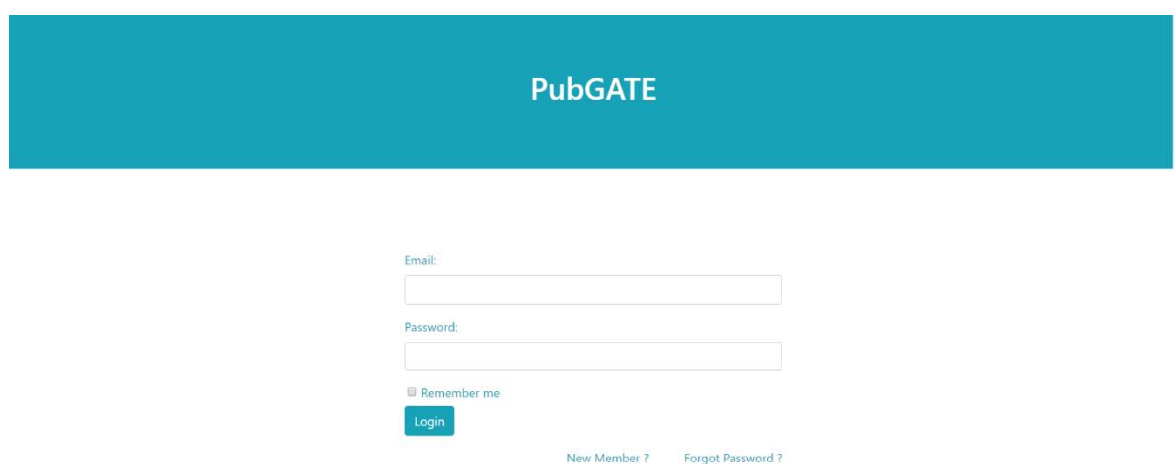
We have used PostgreSQL as our database, PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined

with many features that safely store and scale the most complicated data workloads. The origins of PostgreSQL date back to 1986 as part of the POSTGRES project at the University of California at Berkeley and has more than 30 years of active development on the core platform.

PubMed articles contains many fields as seen in Figure 3.2, we were only interested in certain fields, such as PMID, title, abstract, authors, keywords, and MeSH Terms. MeSH terms or Medical Subject Headings are manually assigned vocabularies by biomedical experts who scan each article, these vocabularies describe the main topic of each article.

### 3.3 PubGate

In addition to PostgreSQL database, we have deployed our web application, PubGate at the same apache server which is running under CentOS7 Operating system. We have used CodeIgniter for developing PubGate, CodeIgniter is a free, open-source, easy-to-use, object-oriented PHP web application framework, providing a ready-to-use library to use with your own PHP applications. We used NetBeans 8.2 as an integrated development environment (IDE). Figure 3.3 shows the homepage of PubGate.



The image shows the homepage of the PubGATE application. At the top, there is a teal banner with the text "PubGATE" in white. Below the banner, the login form is centered. It includes an "Email:" label above a text input field, a "Password:" label above another text input field, a "Remember me" checkbox, a teal "Login" button, and two links: "New Member ?" and "Forgot Password ?".

Figure 3.3 Homepage screen for PubGate

Users have to register in order to use PubGate, after successful registration users are asked to enter their keywords of interest, Figure 3.4 shows the screen of the keyword's section, we save these values in the database after being entered by the users.

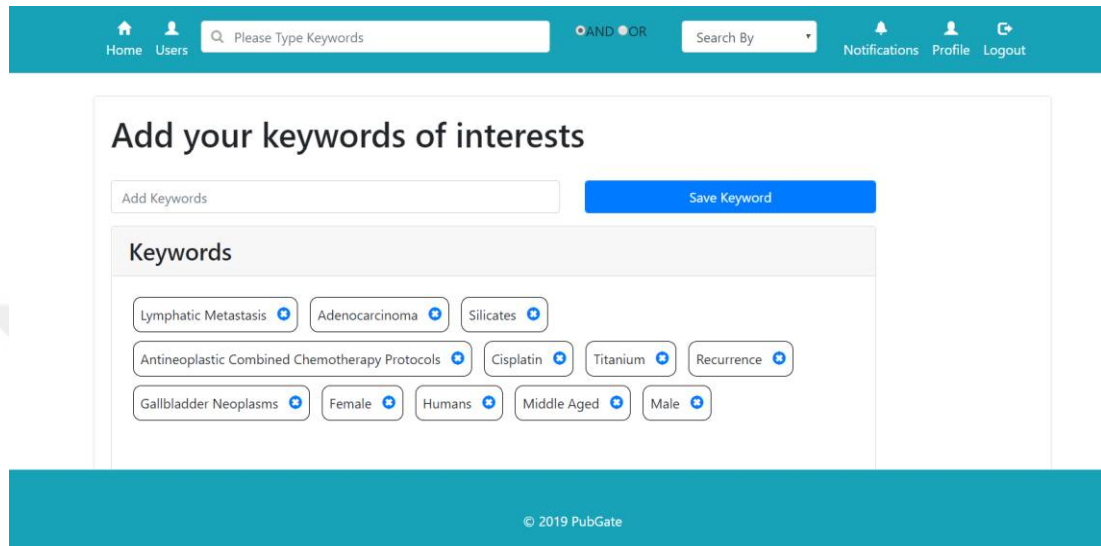


Figure 3.4 Keywords of interest screen

Using PubGate users are able to search for other users in the system and follow them, Figure 3.5 shows a screenshot for the profile screen for a dummy user.

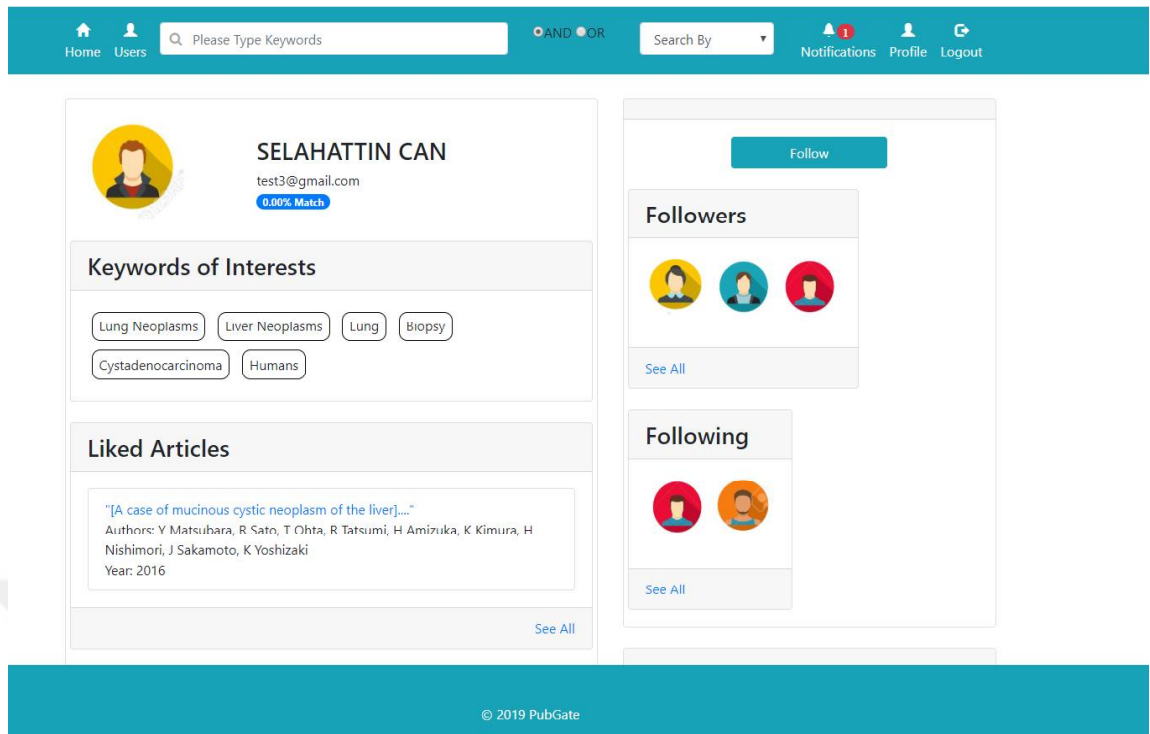


Figure 3.5 Screenshot for a dummy profile from the system

As seen the profile screen provides a valuable information about the users, their first name, last name, email, keywords of interests, followers list, following list, liked articles, and their favorite articles.

Once a user follow other users, their transactions will start to appear at the News Feeds section of the homepage screen, Figure 3.6 shows an example for these transactions.

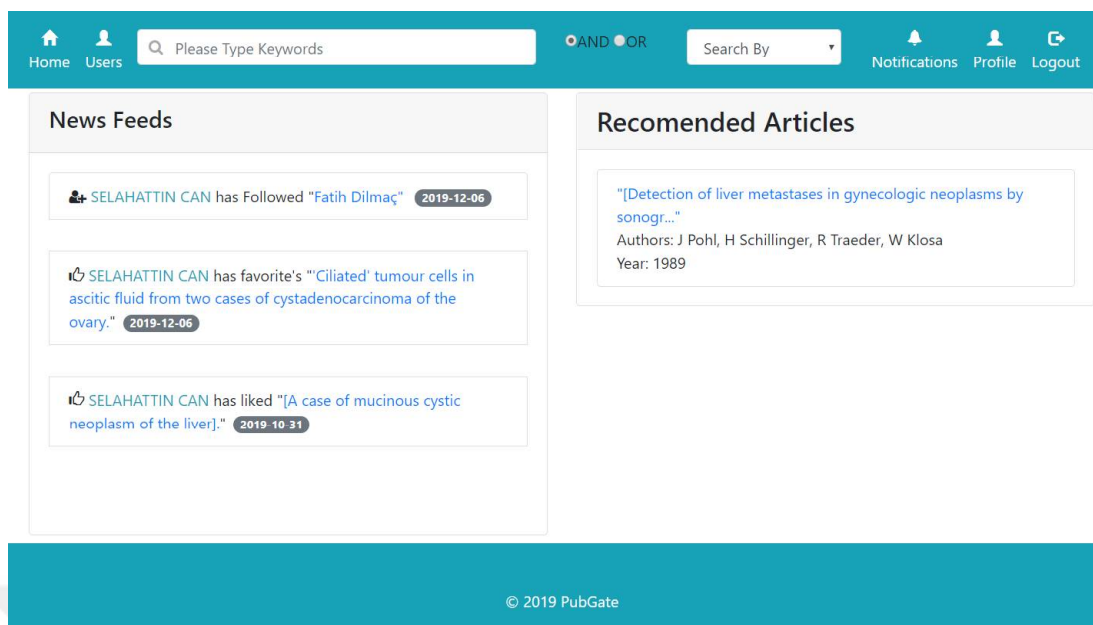


Figure 3.6 Screenshot for the HomePage screen.

PubGate also allows users to search for articles, and give them the option whether to like them, unlike them, favorite them, unfavorite them, or add them pre-created lists. Figure 3.7 shows a screenshot for a randomly selected article from the system, in addition to the title, abstract, PMID, authors, keywords, and MeSH terms we also display who liked this article. PubGate is a user friendly web application that aims to combine the features of social network applications such as exploring, searching, liking content , and following users, with scientific literature, in our case the biomedical articles in PubMed.

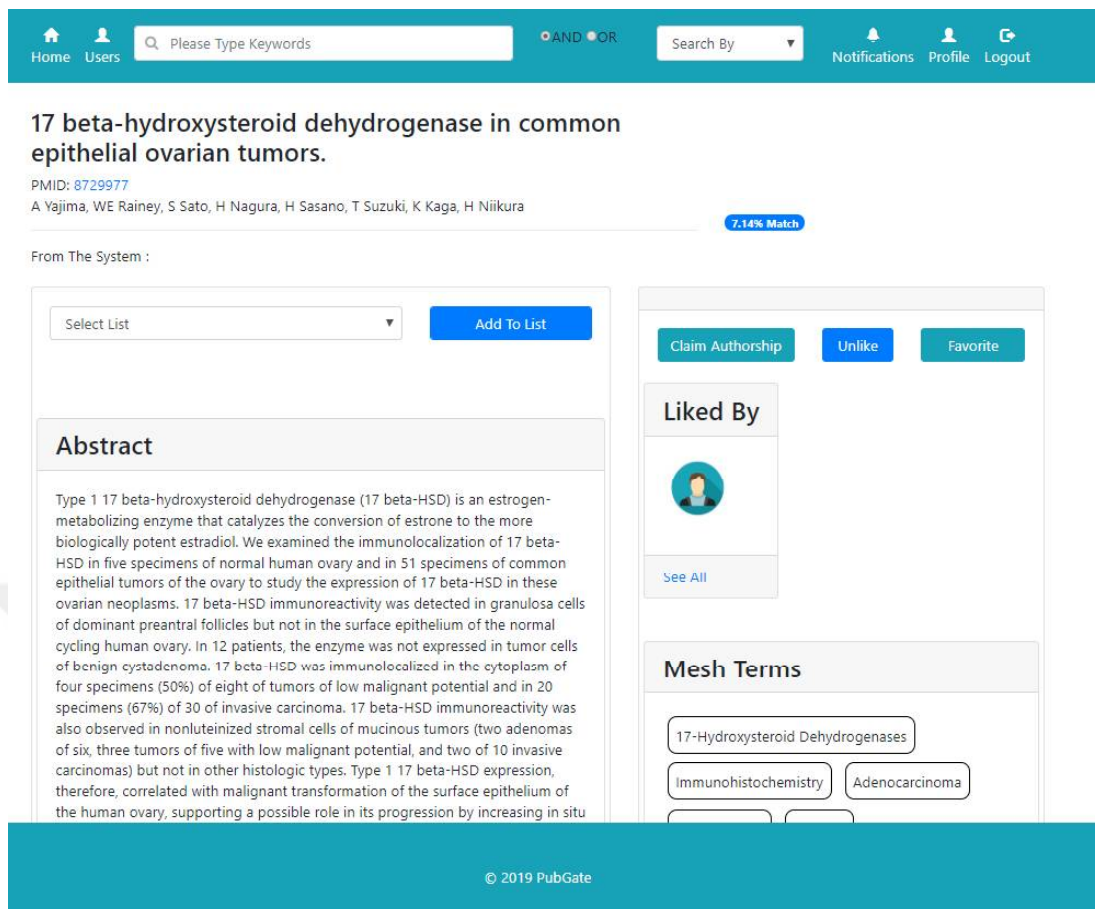


Figure 3.7 Screenshot for an article

### 3.4 Calculate similarities between the users

In the previous section we have explained the functionalities of PubGate, and how they can be used. Figure 3.8 shows the entity-relationship diagram between the users, their liked articles, and their entered keywords.

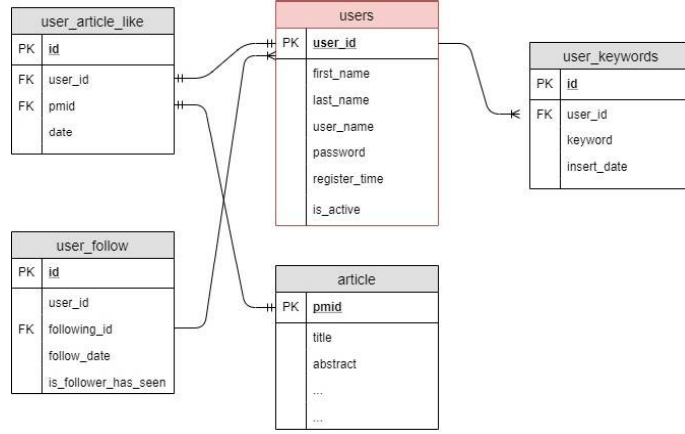


Figure 3.8 Entity-Relationship Diagram between users, articles, and keywords

In our proposed system users are presented as a set of keywords and likes. Similarity between two users  $u$ , and  $v$  is calculated as follows:

$$Sim(u, v) = W_L L(u, v) + W_k K(u, v) \quad (3.1)$$

Where  $L$  represents likes similarity,  $K$  represents keyword similarity. Besides  $W$  is the weight factor of each term in range of 0 and 1 were  $W_L + W_K \leq 1$ , likes similarity and keywords similarity is calculated as follows:

$$L(u, v) = \frac{L_u \cap L_v}{L_u \cup L_v} \quad (3.2)$$

$$K(u, v) = \frac{K_u \cap K_v}{K_u \cup K_v} \quad (3.3)$$

Where  $L_U$  is the set of articles liked by user  $u$ ,  $L_V$  is the set of articles liked by user  $v$ ,  $K_U$  is the set of keywords of user  $u$ , and  $K_V$  is the set of keywords of user  $v$ . Assuming that the number of the users in the system is  $n$  then the matrix presented in figure 3.9 represents the values of the similarities between  $n \times n$  users.

$$\begin{bmatrix} Sim(1,1) & Sim(1,2) & \dots & \dots & Sim(1,n-1) & Sim(1,n) \\ Sim(2,1) & Sim(2,2) & \dots & \dots & Sim(2,n-1) & Sim(2,n) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ Sim(n-1,1) & Sim(n-1,2) & \dots & \dots & Sim(n-1,n-1) & Sim(n-1,n) \\ Sim(n,1) & Sim(n,2) & \dots & \dots & Sim(n,n-1) & Sim(n,n) \end{bmatrix}$$

Figure 3.9 Similarity matrix

The values of the diagonal will be excluded from our calculations, there is no need to find the similarity between a user and himself, the result will always be equal to 1. The matrix we got is called a symmetric matrix, symmetric matrix contains two triangulars, upper and lower, since the two triangulars are similar, calculating one of them is enough, in addition to the diagonal we have also excluded the lower triangular from our calculation.

In our model we consider user  $u$ , and  $v$  to be a neighbor users or similar users if their similarity is higher than 0.6 in other words if  $Sim(u, v) \geq 0.6$ , in section 3.5.1 we explain how we use the neighbor users for articles recommendation.

### 3.5 Articles Recommendation

In the previous section we explained how we calculated the similarities between the users, in this section we explain how we recommend articles based in our two approaches.

#### 3.5.1 Neighbor users

After calculating the similarities between the users in the system and obtaining the matrix mentioned in section 3.4 we are able to identify the neighbor users. Figure 3.9 shows an example for the neighbor users for user  $A$ .



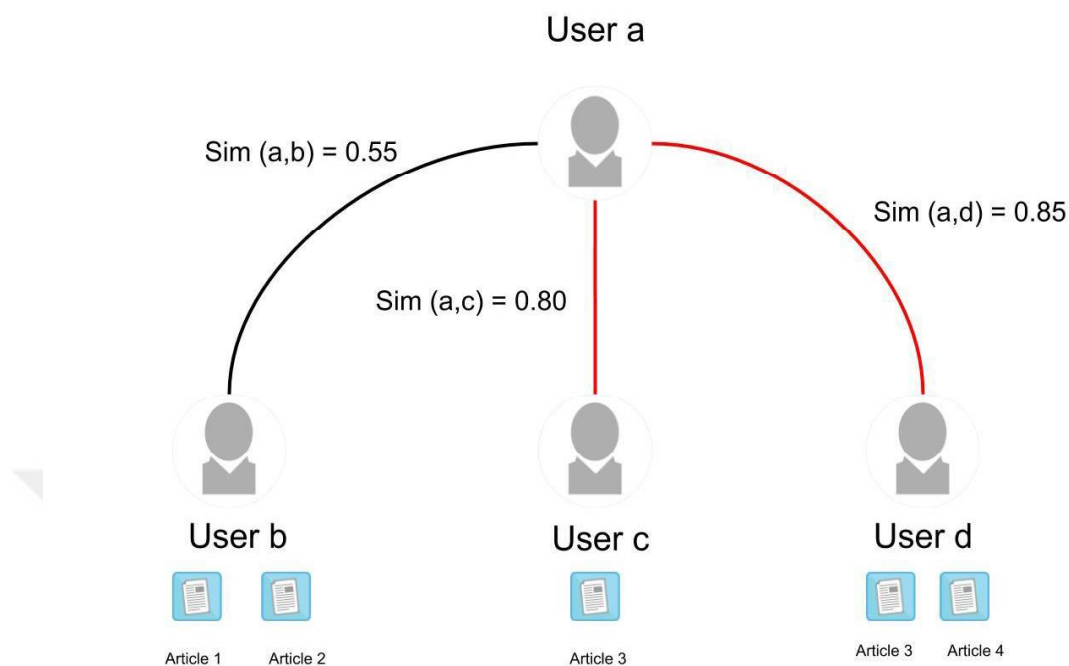


Figure 3.10 Example showing neighbor users

As seen from Figure 3.10 user  $c$ , and  $d$  are considered a neighbor user to user  $a$  since the calculated similarity is higher than 0.6, while user  $b$  is not considered a neighbor user to user  $a$  since the calculated similarity is less than 0.6. In our proposed model we give a priority for the articles that have been liked the most among the neighbor users, in our case article 3 will be at the top list of the recommendation list followed by article 4.

Once users open one of the articles that have been recommended to them, its `is_opened` value will be updated to 1 (TRUE), we make sure to not recommend the same article to the user twice. Figure 3.11 shows a screenshot for how the details of the recommended articles are stored in the database.

**+** `SELECT * FROM "colobrativ" LIMIT 50 (0.021 s)` Edit

<input type="checkbox"/> <u>Modify</u>	<u>id</u>	<u>user id</u>	<u>pmid</u>	<u>counter</u>	<u>is opened</u>
<input type="checkbox"/> <u>edit</u>	8	3	1655875	1	0
<input type="checkbox"/> <u>edit</u>	9	5	3589884	1	1
<input type="checkbox"/> <u>edit</u>	12	4	6024544	5	1
<input type="checkbox"/> <u>edit</u>	10	1	7894444	4	0
<input type="checkbox"/> <u>edit</u>	14	8	2211565	3	0
<input type="checkbox"/> <u>edit</u>	11	1	9888885	4	1
<input type="checkbox"/> <u>edit</u>	9	2	5987771	1	0
<input type="checkbox"/> <u>edit</u>	13	3	5447782	1	0

Figure 3.11 Screenshot for collaborative table in the database

Finally Figure 3.12 shows the section of the recommended articles at the homepage screen. Once the users login into their accounts they will immediately appear beside the news feeds section.

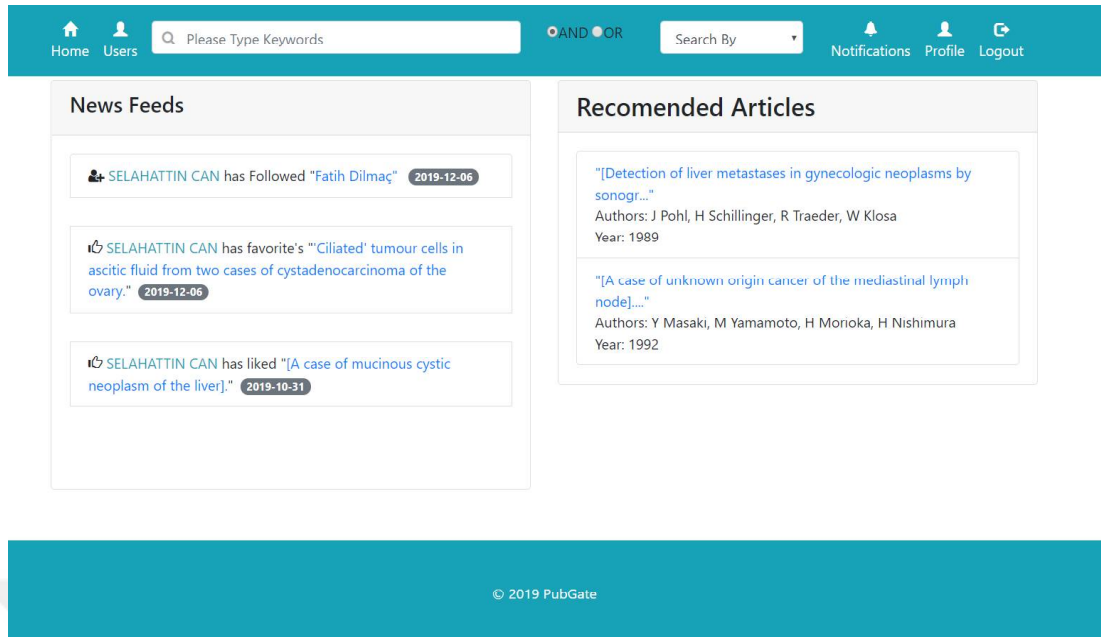


Figure 3.12 Screenshot for recommended articles section at Homepage

### 3.5.2 Elasticsearch Engine

One of the approaches to overcome the cold start problem that we mentioned in section 2.3.3 is to build a hybrid approach which combines both the collaborative filtering approach with any other approach that does not depend on the history nor the previous activities of the user, an example can be combining the collaborative filtering approach with a demographic filtering approach, in which the latter recommends items that has been liked the most in a certain area. In our model a cold user is a user who still does not have that enough number of likes for articles that allow the model to find him neighbor users.. In our proposed model we have decided to combine the collaborative filtering approach with a content-based approach, for that purpose for have integrated our system with Elasticsearch engine.

In our proposed model we have used Elasticsearch engine as a distributed NoSQL database, we integrated the Elasticsearch engine with our system as a content-based approach tool. Elasticsearch is a search engine based on the Lucene library, it's well known for its ability to provide a reliable and accurate results for text searching, thanks to its high mechanism which allows to find similarities between the texts in a very fast way. Figure 3.13 shows how we integrated Elasticsearch within our system.

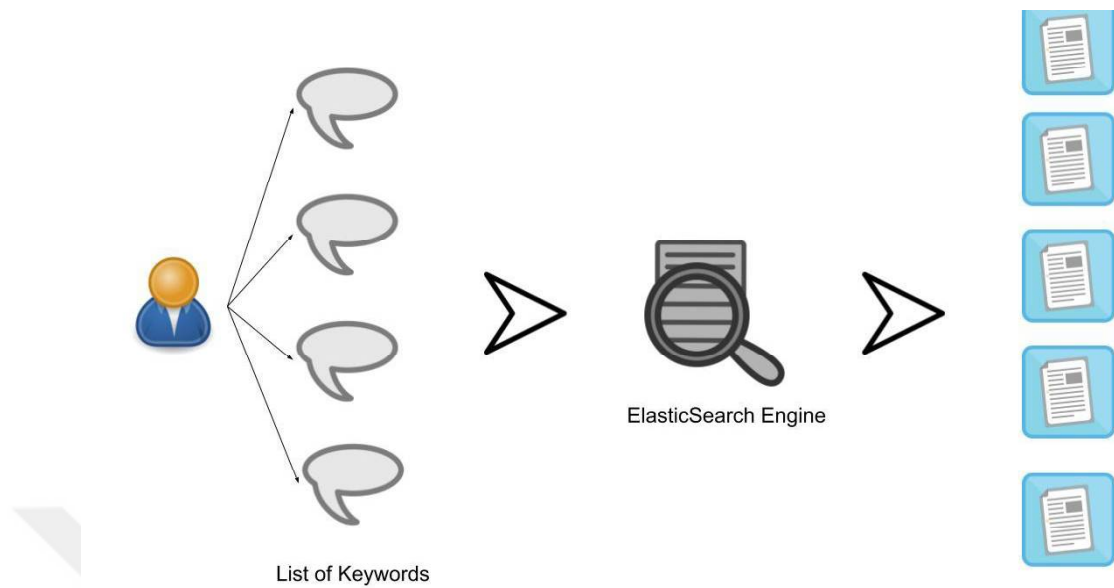


Figure 3.13 Integration of Elasticsearch with our system

As mentioned in section 3.3 users once upon registration they are asked to enter their keywords of interest through the web application PubGate, since the new users are cold users, Elasticsearch engine will use the entered keywords as a query to return a list of the top five articles that contain these keywords as a MeSH terms. The returned five articles will be recommended to the users at their homepage screen, thanks to Elasticsearch they had the capability to calculate the text similarity for big volumes of data. Our purpose was to use Elasticsearch as a helper tool to overcome the cold start problem that the collaborative filtering approach suffers from, there is no intention for us to include it in the calculation nor the evaluation part.

## CHAPTER FOUR

### EXPERIMENTS AND RESULTS

In this chapter we explain how we conducted the evaluation of our proposed model, starting from introducing the evaluation metrics we have used, to creating trivial benchmark datasets, and finally applying these evaluation metrics to the results we have got to evaluate our proposed model.

#### 4.1 Evaluation

To evaluate our proposed model, we used some metrics that are based on the confusion matrix as an evaluation method, Figure 4.1 shows the confusion matrix.

		Predicted Class	
		Yes	No
Actual Class	Yes	True Positive TR	False Negative FN
	No	False Positive FP	True Negative TN

Figure 4.1 Confusion matrix

The confusion matrix is also known as the error matrix, its commonly used to describe the performance of a classification model, in our case it's our proposed recommender system. From Figure 4.1 TP is true positive prediction, in which the recommended article belongs to the field of the user. FP is false positive prediction, in

which the recommended article does not belong to the field of the user. FN is false negative prediction, in which the articles that have not been recommended belong to the field of the user. TN is true negative, in which the articles that have not been recommended does not belong to the field of the user. From the confusion matrix we are able to calculate accuracy, precision, recall, and F-measure.

### **Accuracy**

Accuracy in literature means the quality or the state of being correct or precise, in our evaluation method its equal to the percentage of the correctness of the articles that we recommended. Accuracy is measured as follows:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (4.1)$$

### **Precision**

Precision is the measure of certainty or quality, precision is measured as follows:

$$Precision = \frac{TP}{TN+FP} \quad (4.2)$$

### **Recall**

Recall is the measure of completeness or quantity, recall is measured as follows:

$$Recall = \frac{TP}{TP+FN} \quad (4.3)$$

### **F-measure**

It measures the test's accuracy and by looking at its equation it can be considered as the average of recall and precision , the f-measure is calculated as follows:

$$f - measure = \frac{2 * recall * precision}{recall + precision} \quad (4.4)$$

## 4.2 Creating Benchmark Datasets

After defining our metrics for evaluation measurements in section 4.1, in this section we created the benchmark datasets in order to apply these metrics at our proposed model. As a first step we have created 10 users using dummy email, we assigned first name, last name, and photos for these users. Figure 4.2 shows a screenshot for the list of users we have created.

<input type="checkbox"/> Modify	user_id	first_name	last_name	user_name	password	register_time	is_active
<input type="checkbox"/> edit	1	Fatih	Dilmaç	test1@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	2	Serhat	Dernek	test2@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	3	SELAHATTIN	CAN	test3@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	4	Azad	Çağlayan	test4@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	5	Mustafa	İlhan	test5@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	6	Cansu	Meşe	test6@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	7	Elif	Oztürk	test7@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	8	Gamze	Kutay	test8@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	9	Mohammad	Barakat	test9@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1
<input type="checkbox"/> edit	10	Ezgi	Eren	test10@gmail.com	202cb962ac59075b964b07152d234b70	2019-10-31 13:03:31.150087	1

Figure 4.2 The list of users we have created

Our goal is to divide these 10 users into three different groups. The first group from 1-5 they are interested in lungs cancer, the second group from 6-8 they are interested in HIV, and the third group from 9-10 they are interested in Diabetes. The figures 4.3, 4.4, and 4.5 show the representation of group one, group two, and group three. We assigned a set of keywords, and articles to every user, the mentioned figures show the relation between the users, their liked articles, and their inserted keywords.

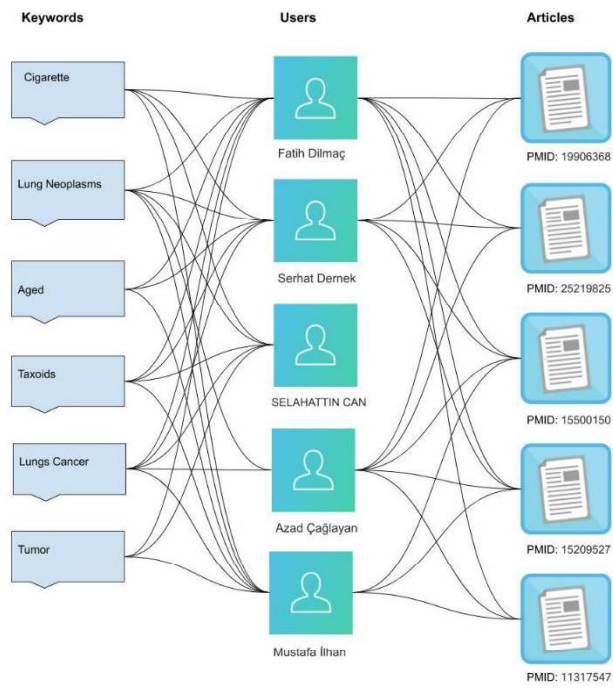


Figure 4.3 Group one representation

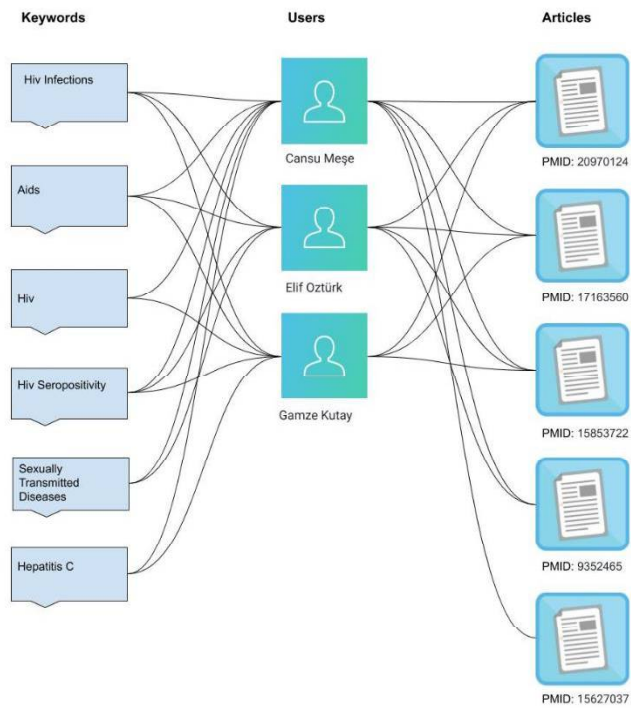


Figure 4.4 Group two representation



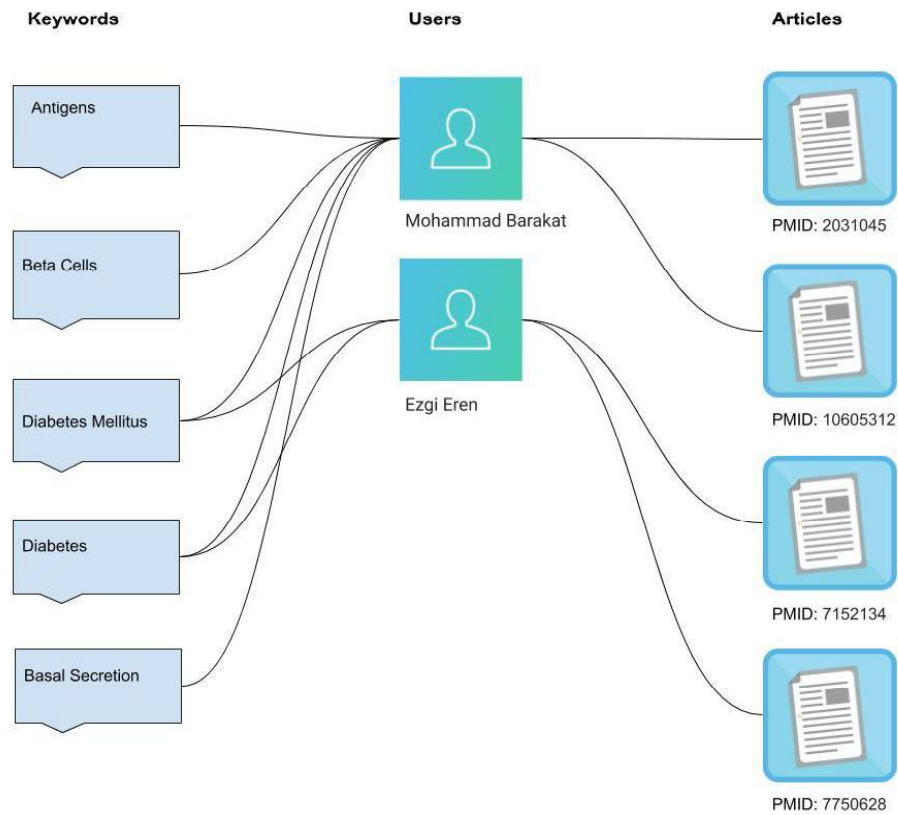


Figure 4.5 Group three representation

### 4.3 Results

In the first step of our results our proposed model calculated the similarities between the users using equation (3.1), the weight factors  $W_L$ , and  $W_K$  were given an equal value of 0.5 because we believe that both of the articles and keywords have the same importance. The matrix mentioned in figure 3.9 which represents the similarities between all the users in the system was also calculated as seen in Figure 4.6, since the number of users is 10 then the matrix hold a size of  $10 \times 10$ .

$$\begin{bmatrix}
 1 & 0.81 & 0.41 & 0.66 & 0.80 & 0 & 0 & 0 & 0 & 0 \\
 0.81 & 1 & 0.33 & 0.60 & 0.61 & 0 & 0 & 0 & 0 & 0 \\
 0.41 & 0.33 & 1 & 0.20 & 0.41 & 0 & 0 & 0 & 0 & 0 \\
 0.66 & 0.60 & 0.20 & 1 & 0.46 & 0 & 0 & 0 & 0 & 0 \\
 0.80 & 0.61 & 0.41 & 0.46 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0.73 & 0.71 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0.73 & 1 & 0.62 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0.71 & 0.62 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.20 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.20 & 1
 \end{bmatrix}$$

Figure 4.6 User's matrix similarity

As mentioned before this matrix is a symmetric matrix, in which the upper triangular and the lower triangular are equal, calculating one side is enough. Neighbor users are users whose similarity is equal to or greater than 0.6, so within this matrix we are only interested in the values that are equal to or greater than 0.6, for a better visualizing for what we have in the system, the rest of the values have been set to zero.

$$\begin{bmatrix}
 0 & 0.81 & 0 & 0.66 & 0.80 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0.60 & 0.61 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0.73 & 0.71 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.62 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}$$

Figure 4.7 Adjusted user's matrix similarity

As seen from the matrix shown in Figure 4.7, from group one, user one, two, four and five are neighbor users, from group two user six, seven, and eight are neighbor

users. Figure 4.8 shows a screenshot for the articles that have been recommended, we have stored these articles in the database to make sure no article is recommended twice for the same user, six articles have been recommended for four different users. PMID is the ID of the article that has been recommended, user\_id is to whom this articles has been recommended to.

<input type="checkbox"/> Modify	id	user_id	pmid	counter	is_opened
<input type="checkbox"/> edit	34	2	11317547	3	0
<input type="checkbox"/> edit	36	5	19906368	2	0
<input type="checkbox"/> edit	35	5	25219825	2	0
<input type="checkbox"/> edit	38	7	15627037	1	0
<input type="checkbox"/> edit	39	8	9352465	2	0
<input type="checkbox"/> edit	40	8	15627037	1	0

Figure 4.8 List of the recommended articles.

#### 4.4 Evaluation

For evaluating our model, we were able to only use the precision, recall, and f-measure metrics from the confusion matrix, we applied these three metrics at the six articles the model had recommended. Before starting with the calculations we have manually annotated the articles that have been liked by the ten users, table 4.1 shows the whole list of articles in addition to their field.

Table 4.1 Annotating the articles that have been liked by the users

Article ID	Field / Domain
19906368	Lungs Cancer
25219825	
15500150	
15209527	
11317547	

Table 4.1 continues

20970124 17163560 15853722 9352465 15627037	HIV
10605312 2031045 7750628 7152134	Diabetes

For our proposed model precision, recall, and f-measure is calculated as follows:

$$Precision = \frac{\# \text{ of recommendations that are relevant}}{\# \text{ of items we recommended}} = 6/6 = 1$$

$$Recall = \frac{\# \text{ of recommendations that are relevant}}{\# \text{ of all possible relevant items}} = 6/15 = 0.4$$

$$f - \text{measure} = \frac{2 * recall * precision}{recall + precision} = 0.57$$

We can see that our proposed model has an outstanding value for the precision, which means all the items that have been recommended are relevant for the users. For recall having a value of 0.4 is normal, users are only considered similar if their similarity value is greater than our threshold value which is 0.6, which means it's hard to recommend all the relevant items unless the similarity between the users is great.

## **CHAPTER FIVE**

### **CONCLUSION AND FUTURE WORKS**

In the era of technology and information, finding relevant items for users is becoming a complicated task, hundreds of new items are being added daily to the web, processing such amount of data manually is not only an exhausting task but also requires billions of hours, to mitigate this issue comes the part of artificial intelligence, mainly the recommender systems.

Recommender systems are algorithms that helps user to find what they are looking for by suggesting relevant items to them. In a scholarly domain where items are research articles, and researchers are the main users, recommender systems will recommend research articles for the researches. In the literature review we have conducted, recommender system approaches are mainly divided into three approaches, content-based, collaborative filtering, and hybrid approach. While the content-based approach focuses at the features of the items for computing similarities between the items, the collaborative filtering approach focuses at finding users with similar taste, the items of neighbor users are used for recommendation. Finally, the hybrid approach usually combines two or more approaches at the same time, such as combing the content-based approach with the collaborative filtering approach, the purpose of this approach is to overcome the drawbacks of solely relying at one approach.

In our study we have proposed a model that focuses at the collaborative filtering approach, in which we used the Jaccard's similarity to compute the similarities between the users according to their sets of liked articles, and keywords. Articles that have been liked the most by neighbor users were recommended first. Another important contribution were overcoming the cold start problem which the collaborative-filtering approach suffers from. We integrated our model with the Elasticsearch engine as a content-based tool to recommend articles for new users based on their entered keywords.

For the experimentation part we have created 10 users and assigned keywords, and articles to their liked libraries which enabled us to calculate the similarities between

the users, and recommend articles. The calculated precision value shows that all the articles the system have recommended were relevant.

As the researcher's interests may change over time, taking the time order into consideration can be considered as an option for improvement in recommendation model, tracing the sequence of the user's behavior is a future work improvement for our study.



## REFERENCES

- Achakulvisut, T., Acuna, D. E., Ruangrong, T., & Kording, K. (2016). Science Concierge: A fast content-based recommendation system for scientific publications. *PLOS One*, *11*(7),1-11.
- Agarwal, A., & Chauhan, M. (2017). Similarity measures used in recommender systems: a study. *International Journal of Engineering Technology Science and Research*, 2394-3386.
- Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2016). Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, *17*(4), 305-338.
- Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*, (26), 225-238.
- Borges, H. L., & Lorena, A. C. (2010). A survey on recommender systems for news data. In *Smart Information and Knowledge Management* (129-151). Berlin, Heidelberg: Springer.
- Cheng, W., Yin, G., Dong, Y., Dong, H., & Zhang, W. (2016). Collaborative filtering recommendation on users interest sequences. *PLOS One*, *11*(5), e0155739.
- Hassan, H. A. (2017). Personalized research paper recommendation using deep learning. *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, 327–330.
- Hristakeva, M., Kershaw, D., Rossetti, M., Knoth, P., Pettit, B., Vargas, S., & Jack, K. (2017). Building recommender systems for scholarly information. *Proceedings of the 1st Workshop on Scholarly Web Mining*, 25-32.

- Kompan, M., & Bielikova, M. (2010). Content-based news recommendation. *International Conference on Electronic Commerce and Web Technologies*, 61-72.
- Lin, J., & Wilbur, W. J. (2007). PubMed related articles: a probabilistic topic-based model for content similarity. *BMC Bioinformatics*, 8(1), 423-434.
- Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56, 156-166.
- Mohammad, S., Kylasa, S., Kollias, G., & Grama, A. (2016). Context-specific recommendation system for predicting similar PubMed articles. *In 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 1007-1014.
- Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications*, 92, 507-520.
- Pessemier, T. D., Leroux, S., Vanhecke, K., & Martens, L. (2015). Combining collaborative filtering and search engine into hybrid news recommendations. *In 3rd International Workshop on News Recommendation and Analytics (INRA 2015), in conjunction with the 9th ACM Conference on Recommender Systems (RecSys 2015)*, 13-18.
- Sahoo, A. K., Pradhan, C., Barik, R. K., & Dubey, H. (2019). DeepReco: deep learning based health recommender system using collaborative filtering. *Computation*, 7(2), 25.
- Swapnil, N. (2012). *A hybrid recommender: user profiling from tags/keywords and ratings*. Kansas: PhD thesis, Kansas State University, Kansas.



Wei, J., He, J., Chen, K., Zhou, Y., & Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69, 29-39.

Wei, W., Marmor, R., Singh, S., Wang, S., Demner-Fushman, D., Kuo, T.-T, Ohno-Machado, L. (2016). Finding related publications: extending the set of terms used to assess article similarity. *AMIA Summits on Translational Science Proceedings*, 225-234.

Yingyuan, X., Pengqiang, A., Hsu, C.-H., Hongya, W., & Xu, J. (2015). Time-ordered collaborative filtering for news recommendation. *China Communications*, 12(12), 53-62.

Yoneya, T., & Mamitsuka, H. (2007). Pure: a pubmed article recommendation system based on content-based filtering. *International Conference on Genome Informatics*, 18, 267-276.