Expert Recommendation System for Scholarly Collaboration

A project report submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

of

Bachelor of Technology

in

Information Technology

by

Vasu Agarwal

Reg. No. 170911052

Under the guidance of

Dr. Tribikram Pradhan
Assistant Professor
Department of I & CT
Manipal Institute of Technology
Manipal, India

Mr. Santhosh Kamath
Assistant Professor - Selection Grade
Department of I & CT
Manipal Institute of Technology
Manipal, India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

SEPTEMBER 2021

My thesis is dedicated to my friends and family. My lovely parents, Vivek and Reena Agarwal, deserve special thanks for their unwavering encouragement and support. They never fail to inspire me. At the same time, I want to express my gratitude to my caring sister, Ishani Agarwal, who has always offered me guidance and has never left my side.

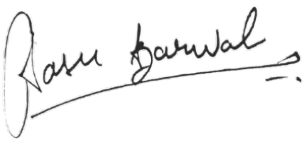
This dissertation is also dedicated to my numerous friends who have helped me during the process. I will always be grateful for what they have done for me.

DECLARATION

I hereby declare that this project work, titled **Expert Recommendation**, was completed by me in the Department of Information and Communication Technology at Manipal Institute of Technology, Manipal, under the supervision of **Dr. Tribikram Pradhan**, **Assistant Professor**, Department of Information and Communication Technology, M. I. T., Manipal. No part of this work has been submitted for the award of a degree or diploma either to this University or to any other Universities.

Place: Manipal

Date: 08-09-2021



Vasu Agarwal

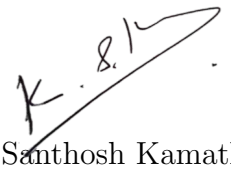
MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

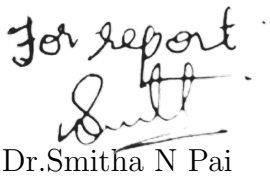
(A constituent unit of MAHE, Manipal)

CERTIFICATE

This is to certify that this project entitled **Expert Recommendation** is a bonafide project work done by **Mr. Vasu Agarwal (Reg.No.: 170911052)** at Manipal Institute of Technology, Manipal, independently under my guidance and supervision for the award of the Degree of Bachelor of Technology in Information Technology.



Mr. Santhosh Kamath
Assistant Professor - Selection Grade
Department of I & CT
Manipal Institute of Technology
Manipal, India



Dr.Smitha N Pai
Professor & Head
Department of I & CT
Manipal Institute of Technology
Manipal, India

ACKNOWLEDGEMENTS

I'd want to convey my heartfelt gratitude to Dr. Tribikram Pradhan for allowing me to undertake this project. I owe him a debt of gratitude for his assistance and advice in completing this job.

I'd also like to express my gratitude to Dr.Smitha N Pai for her unwavering support.

ABSTRACT

Researchers are recently publishing articles in such a large-scale that has never been seen before. In academic society, the age of scholarly big data has arrived, and researchers are having a difficult time locating necessary papers to study and meeting new scholars with whom to collaborate. To tackle this problem, we will be building an Expert Recommendation Model that assists the users in not only finding the right papers to study, but also use a co-author network to suggest potential researchers for collaboration.

For our model we look into how *Content-based* recommender system works for already available commercial products like movies recommendation on Netflix, books recommendation on Amazon and turn it towards **Academic recommendations**. We build the model by taking title and abstract of the papers as our parameters and lemmatizing them to extract the context out of these parameters. With this Context-aware recommender system, we can now recommend relevant academic papers to the users.

By taking into consideration the context of the research papers and coauthor network, we were able to find similar papers to recommend using cosine similarity. We also define two parameters, Mean squared error and Root mean square error to check the performance of our model. As a result, we can see that our model is able to predict and recommend relevant papers with a higher accuracy than other recommender systems.

[**Information systems**]: Information retrieval—Retrieval tasks and goals, Recommender systems

Contents

List of Tables

List of Figures

ABBREVIATIONS

RS	:	Recommender System
ML	:	Machine Learning
CN	:	Common Neighbors
WC	:	Word Cloud
SVD	:	Singular Value Decomposition
KNN	:	K-Nearest Neighbors
MSE	:	Mean Squared Error
RMSE	:	Root Mean Square Error

NOTATIONS

θ	:	Cosine angle between two vectors
β	:	Smoothing factor for topics
$\sum$	:	Sum of all the terms
y_i	:	Predicted value of ith$term$
$\tilde{y}_i$	:	Actual value of ith$term$

Chapter 1

Introduction

1.1 Introduction

In recent years, scholar have produced large-scale publications never seen before. The age of scholarly big data has arrived in academic society [1]. Due to the simple and easy access to a lot of scholarly data, the topic of information overload has received a lot of attention. Scholars have a hard time finding needed papers to study and meeting new people to collaborate with. In light of this, scholarly recommendation systems such as publication recommendation [2] and collaborator suggestion [3], have been created to assist scholars in finding needed material quickly and easily.

Our main goal is to thus build a *An Expert Scholar Recommendation System* that assists the users in not only finding the right papers to study, but also use a scholar *co-author network* to suggest potential scholars for collaboration.

1.2 Motivation

Recommender systems have grown in popularity and are attracting more and more attention from academics and industry [4]. However, fewer studies have looked into academic paper recommender systems or scholar recommendation than other recommender systems, such as those for movies, books, and music.

Researchers usually have to filter through a vast number of academic articles to find the ones that are relevant to their research. Because the quantity of published scholarly papers is increasing at an exponential rate, this screening is inconvenient and time-consuming. As a result, efficient academic paper recommender systems are in high demand [5].

In light of this, we aim to build a recommender system that instead of recommending movies or books, recommends academic paper for study. This project also aims in building a recommender system that would help the users in finding potential scholars for collaboration.

With the recommender model, the researchers are able to find proper and relevant academic articles instantaneously and could thus work and produce results at a much higher speed.

1.3 Objectives

The objective of this project is to to build a personalized recommender system that suggests not only relevant academic papers to study, but also recommends potential collaborators for scholarly research.

We aim to take the features embedded in a research paper like it's title and abstract and use it to recommend relevant academic paper for the study.

We also want to recommend potential scholars for collaboration based on co-author network. The purpose is to locate potential collaborators by comparing scholars' similarities. It has been demonstrated that similar scholars will likely collaborate [6]. As a result, the trick is to determine how comparable scholars are.

1.4 Organization of Report

1.4.1 Chapter 1

In this chapter we began by giving a general introduction of our project and our motivation for pursuing this topic. We also discuss about the previous work done and how our model looks to improve on them. In the end of Chapter 1, we also discuss our objective and what we are trying to achieve.

1.4.2 Chapter 2

The Chapter 2 covers the literature review. We begin by discussing the context and background of our project's topic. We also talk about the preliminary work that has been done on the field.

We then go over the results of prior studies and see what we can learn from them. We then move on to our work. Our objective is to build is model that is based on the previous RS models and then suggest potential collaborators.

1.4.3 Chapter 3

The design and technique of our project are discussed in this chapter. We talk about how we got to the end of the project and what measures we took to get there.

We will go over the process of building the RS model and how it takes the user input, calculates the *similarity score* and recommends paper along with coauthors.

We also go over the software and python libraries that were used in the project.

1.4.4 Chapter 4

In the chapter 4, we will see the implementation and result of our project. We will go through a few examples and how the model gives different recommendations based on different user inputs.

We will also see how the **cosine similarity** is used to calculate the similarity score and recommend potential collaborators.

Then the efficiency of our model will then be compared to that of other Machine Leaning models, and the results will be displayed.

1.4.5 Chapter 5

This chapter discusses the study's results and conclusions as well as the scope of future studies.

Chapter 2

Background

2.1 Literature Survey

In this section, we will look at two linked works, namely: *Information Retrieval Technique* in the form of PageRank and Common Neighbors algorithm. We will also take a deeper look into what is Recommender System and what different types of RS are available.

2.1.1 Information Retrieval technique

In order to increase the quality of web search engines, Google uses *PageRank* as one of the ways for determining the significance of web pages [7]. This approach has not just been used to rank web search results, but it has also been used to suggest scientific papers [8].

Google Scholar uses PageRank algorithms to find papers that are related to the topic of interest. Google Scholar's "related articles" tool presents a list of closely related papers, rated largely by how similar they are to the paper of interest, to assist with recommendations [9].

Although PageRank is a valuable tool for determining a paper's authority, it ranks papers mostly based on the number of citations they receive. As a result, even if a piece is regarded as notable literature, it is consistently placed

low.

This is a critical drawback because new papers can help researchers grasp current difficulties and choose future research objectives.

2.1.2 Common Neighbors Algorithm

One of the most important responsibilities in academic data mining is recommending scientific collaborations [10]. Academic collaboration seeks to identify possible collaborators by comparing scholars' similarity, with the idea that comparable scholars will interact in the future. The trick is to figure out how to exactly measure how similar scholars are.

Because co-authorships can be used to index scientific collaborations, it is possible to create a scientific collaboration network.

Common Neighbors(CN) is one of many link prediction-based systems that has been created, which encapsulates the idea that two strangers who share a buddy are more likely to be introduced than those who do not [11].

2.2 Recommender System

In a broad sense, recommender systems are algorithms that attempt to recommend relevant items to customers (items being movies to watch, text to read, products to buy etc.).

Amazon's product recommendations, Netflix's recommendations for movies and TV shows in your feed, YouTube's recommended videos, Spotify's music, Facebook's news feed, and Google Ads are all examples of recommender systems in action.

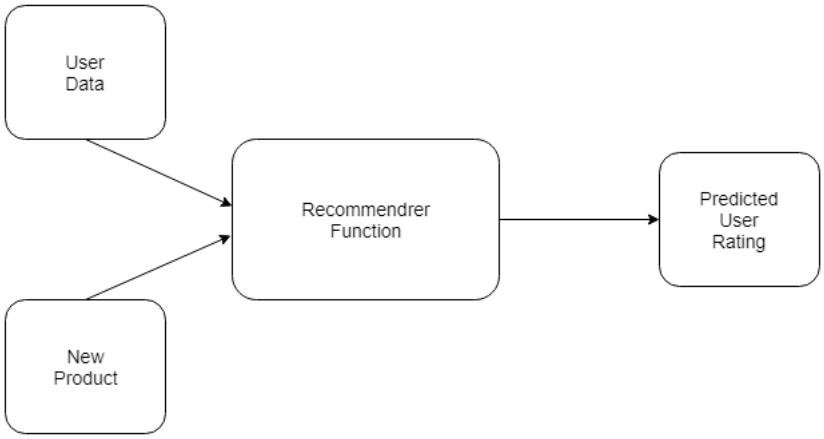


Figure 2.1: Recommender Function

Now we'll look into two primary recommender system paradigms, collaborative and content-based methods:

2.2.1 Collaborative Filtering

The *collaborative* approaches for recommender systems are methods that are totally based on past interactions between users and products in order to create fresh recommendations. The so-called "user-item interactions matrix" stores these interactions [12] as shown in figure 2.2.

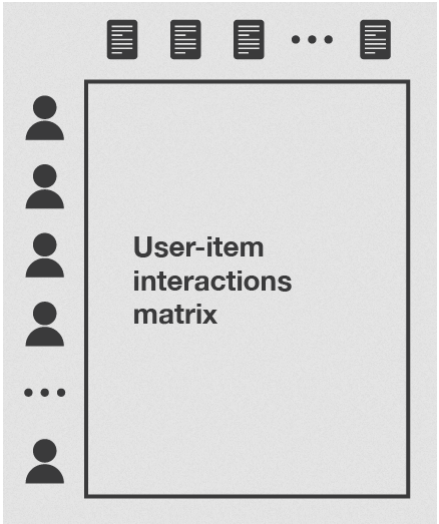


Figure 2.2: User-item interaction matrix

Then the primary assumption that governs collaborative approaches is that past user-item interactions are sufficient for detecting comparable people and/or similar objects, as well as making predictions based on these estimated proximity.

As a result, without enough initial ratings from users, collaborative filtering will not be able to provide accurate suggestions. Because obtaining user score information in digital libraries is challenging, this issue also arises in the sphere of academic articles.

2.2.2 Content-based Filter

Unlike collaborative methods that rely purely on user-item interactions, content-based approaches make use of extra information about users and/or items.

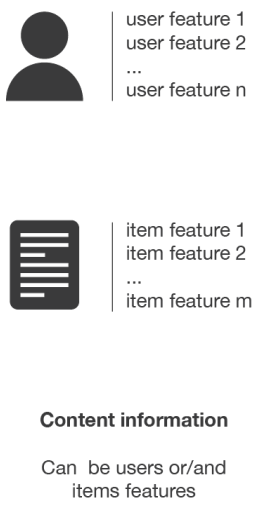


Figure 2.3: Content-based

Hence, users are recommended things based on their descriptions via content-based filtering algorithms [13]. Because text-based features are so good at classifying papers, academic article recommender systems that use content-based filtering rely on the ability to compare full texts or keywords [14]. An

example of content-based filtering technique is demonstrated by SCOPUS's "related documents" function.

Chapter 3

Design & Methodology

This section describes the dataset, the proposed model and steps taken to implement it:

3.1 Proposed Model

For our model we look into how *Content-based* recommender system works for already available commercial products like movies recommendation on Netflix, books recommendation on Amazon and turn it towards **Academic recommendations**.

To do this we first build a Content-based recommender system using **Title** and **Abstract** of papers as our parameters. But to enhance this, we use *Lemmatization* to not only match the content but also the *context* of the paper. With this we are able to overcome the issue of text features being conceptually similar but different in vocabulary.

Using our new **Context-aware** recommender system, we can now recommend relevant academic papers to the users.

We also want to recommend potential scholars for collaboration and for this we build a *co-author network* and find the Cosine similarity between each Author.

3.2 Data source

For this project, I utilised a dataset from the **arXiv.org** by Cornell University. There are 10 separate qualities and 41000 rows of data in this JSON file. [15] The attributes in our dataset are listed as below in figure 3.1:

	user_id	author_id	author	title	year	abstract	co_author	coauthor_id	title_id
0	1	1000	Wolfgang Bibel	Dual Recurrent Attention Units for Visual Ques...	1989	When two or more distinct organizations interc...	Jean-Marie Nicolas	1001	83
1	2	1001	Jean-Marie Nicolas	Sequential Short-Text Classification with Recu...	1989	Hosts are under EMCON condition, short for Emi...	J. B. Bocca	1005	49
2	3	1002	D. Chan	Multiresolution Recurrent Neural Networks: An ...	1989	In distributed systems surveillance protocols ...	M. Wallace	1003	21
3	4	1003	M. Wallace	Learning what to share between loosely related...	1988	The Versatile Message Transaction Protocol (VM...	J. C. Freytag	1004	57
4	5	1004	J. C. Freytag	A Deep Reinforcement Learning Chatbot	1988	The paper is concerned with efficient implemen...	J. B. Bocca	1005	57

Figure 3.1: Dataset used

3.2.1 Methodology

Below are the steps taken to build and test our model:

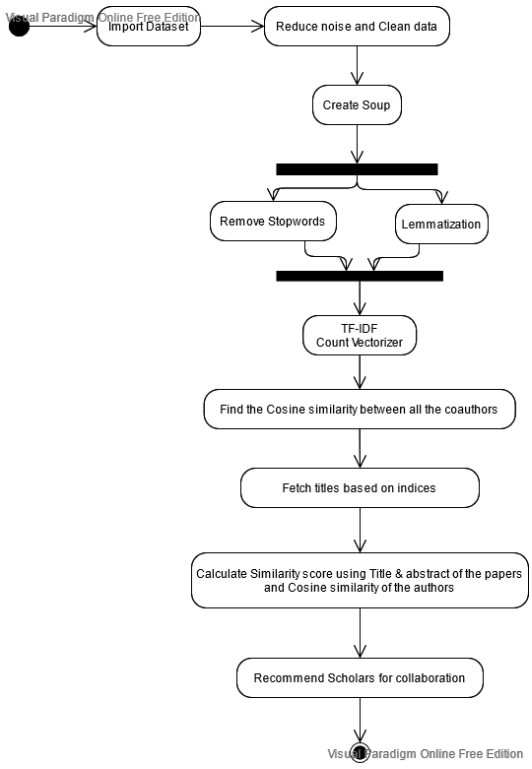


Figure 3.2: Activity diagram of our Methodology

1. **Import Dataset**

   We first import our data from arXiv dataset [15]. This dataset contains 41000 rows with 10 attributes. A snippet of the raw data can be seen in the figure 3.3.


```
{
    "author": "[{'name': 'Ahmed Osman'}, {'name': 'Wojciech Samek'}]",
    "day": 1,
    "id": "1802.00209v1",
    "link": "[{'rel': 'alternate', 'href': 'http://arxiv.org/abs/1802.00209v1', 'type': 'text/html'}, {'rel': 'related', 'href': 'http://arxiv.org/pdf/1802.00209v1', 'typ
    "month": 2,
    "summary": "We propose an architecture for VQA which utilizes recurrent layers to\ngenerate visual and textual attention. The memory characteristic of the\nproposed r
    "tag": "[{'term': 'cs.AI', 'scheme': 'http://arxiv.org/schemas/atom', 'label': None}, {'term': 'cs.CL', 'scheme': 'http://arxiv.org/schemas/atom', 'label': None}, {'t
    "title": "Dual Recurrent Attention Units for Visual Question Answering",
    "year": 2018
},
{
    "author": "[{'name': 'Ji Young Lee'}, {'name': 'Franck Dernoncourt'}]",
    "day": 12,
    "id": "1603.03827v1",
    "link": "[{'rel': 'alternate', 'href': 'http://arxiv.org/abs/1603.03827v1', 'type': 'text/html'}, {'rel': 'related', 'href': 'http://arxiv.org/pdf/1603.03827v1', 'typ
    "month": 3,
    "summary": "Recent approaches based on artificial neural networks (ANNs) have shown\npromising results for short-text classification. However, many short texts\noccur
    "tag": "[{'term': 'cs.CL', 'scheme': 'http://arxiv.org/schemas/atom', 'label': None}, {'term': 'cs.AI', 'scheme': 'http://arxiv.org/schemas/atom', 'label': None}, {'t
    "title": "Sequential Short-Text Classification with Recurrent and Convolutional\n  Neural Networks",
    "year": 2016
},
```


Figure 3.3: Raw Data

2. **Reduce Noise & Clean Data**

   We reduce noise by removing an attribute that doesn't contribute to our model.

   We also clean the data by removing spaces, special characters, numbers etc. and create two new columns, "clean_title" and "clean_abstract".

   The figure 3.3 shows the dataset with new attributes:

	user_id	author_id	author	title	year	abstract	co_author	coauthor_id	title_id	clean_title	clean_abstract
0	1	1000	Wolfgang Bibel	Dual Recurrent Attention Units for Visual Ques...	1989	When two or more distinct organizations interc...	Jean-Marie Nicolas	1001	83	dual recurrent attention units for visual ques...	when two or more distinct organizations interc...
1	2	1001	Jean-Marie Nicolas	Sequential Short-Text Classification with Recu...	1989	Hosts are under EMCON condition, short for Emi...	J. B. Bocca	1005	49	sequential short text classification with recu...	hosts are under emcon condition short for emis...
2	3	1002	D. Chan	Multiresolution Recurrent Neural Networks: An ...	1989	In distributed systems surveillance protocols ...	M. Wallace	1003	21	multiresolution recurrent neural networks an a...	in distributed systems surveillance protocols ...
3	4	1003	M. Wallace	Learning what to share between loosely related...	1988	The Versatile Message Transaction Protocol (VM...	J. C. Freytag	1004	57	learning what to share between loosely related...	the versatile message transaction protocol vmt...
4	5	1004	J. C. Freytag	A Deep Reinforcement Learning Chatbot	1988	The paper is concerned with efficient implemen...	J. B. Bocca	1005	57	a deep reinforcement learning chatbot	the paper is concerned with efficient implemen...

Figure 3.4: Cleaned

3. **Create Soup**

   Then we combine these two new columns to form a word *soup*.

	user_id	author_id	author	title	year	abstract	co_author	coauthor_id	title_id	clean_title	clean_abstract	soup
0	1	1000	Wolfgang Bibel	Dual Recurrent Attention Units for Visual Ques...	1989	When two or more distinct organizations interc...	Jean-Marie Nicolas	1001	83	dual recurrent attention units for visual ques...	when two or more distinct organizations interc...	dual recurrent attention units visual question...
1	2	1001	Jean-Marie Nicolas	Sequential Short-Text Classification with Recu...	1989	Hosts are under EMCON condition, short for Emi...	J. B. Bocca	1005	49	sequential short text classification with recu...	hosts are under emcon condition short for emis...	sequential short text classification recurrent...
2	3	1002	D. Chan	Multiresolution Recurrent Neural Networks: An ...	1989	In distributed systems surveillance protocols ...	M. Wallace	1003	21	multiresolution recurrent neural networks an a...	in distributed systems surveillance protocols ...	multiresolution recurrent neural networks appl...
3	4	1003	M. Wallace	Learning what to share between loosely related...	1988	The Versatile Message Transaction Protocol (VM...	J. C. Freytag	1004	57	learning what to share between loosely related...	the versatile message transaction protocol vmt...	learning share loosely related tasksthe versat...
4	5	1004	J. C. Freytag	A Deep Reinforcement Learning Chatbot	1988	The paper is concerned with efficient implemen...	J. B. Bocca	1005	57	a deep reinforcement learning chatbot	the paper is concerned with efficient implemen...	deep reinforcement learning chatbotthe paper c...

Figure 3.5: Dataset with Soup

The Word Cloud of the resulted soup along with the word count of most popular 25 words is shown below in the figure 3.5:

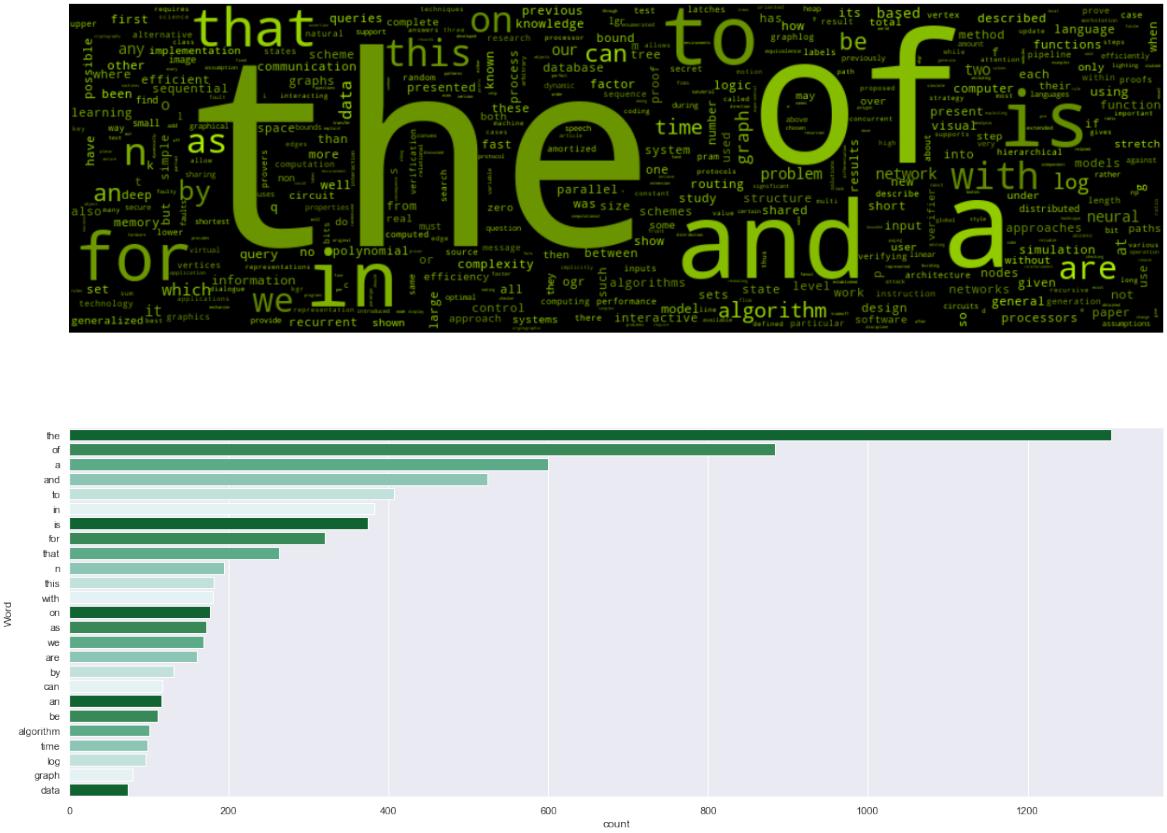


Figure 3.6: Word Cloud of the Soup

4. **Remove stopwords and Lemmatization**

As it can be seen from figure 3.5, the most popular words present in the

dataset are words like *the, of, and* etc.

Hence, in order to get actual relevant words, we need to remove all the stopwords. The WC after removing stopwords is shown in figure 3.6:

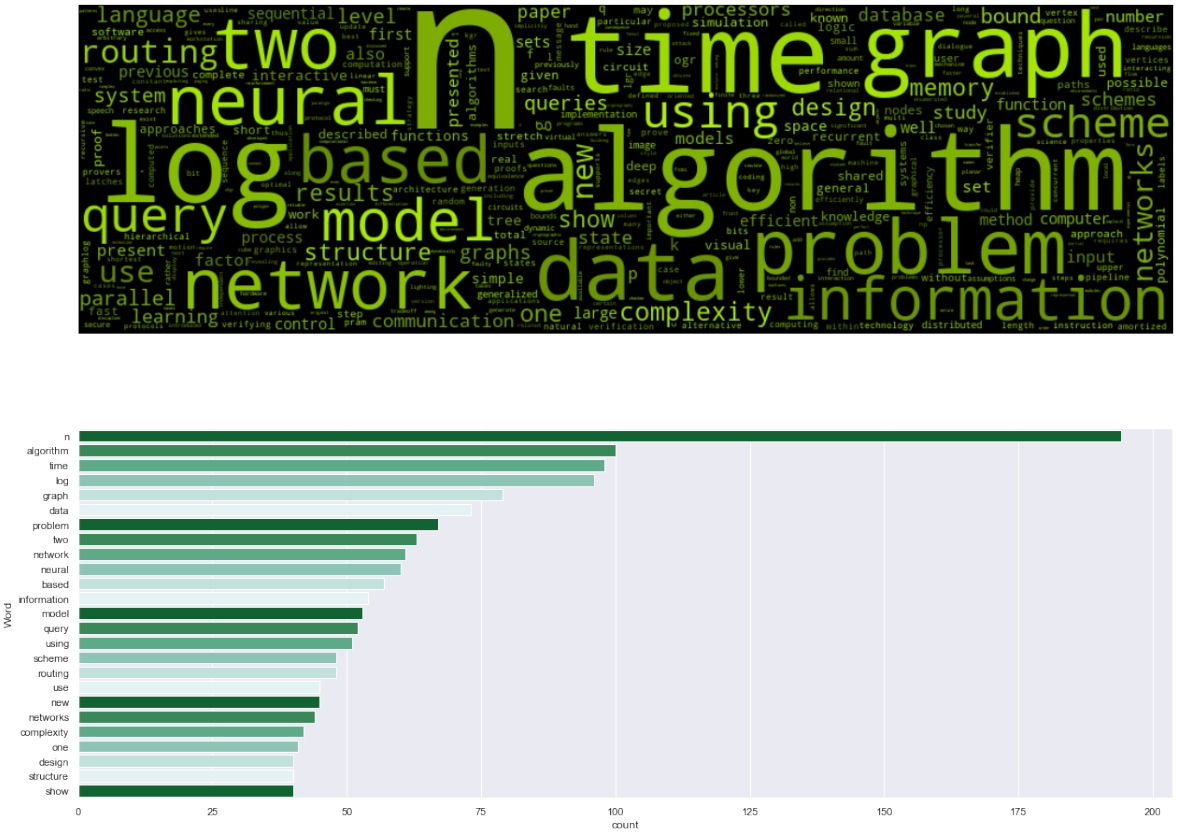


Figure 3.7: Updated WC after stopwords removal

We also use *Lemmatization* to match the words which maybe different in vocabulary but gives the same context. By doing this, we make our model work on not only *content* but also the *context* of the papers.

Definition 1 *"In linguistics, lemmatization is the process of assembling a word's inflected words so that they may be analysed as a single item, designated by the word's lemma, or dictionary form."*

5. **Count Vectorizer**

In this step we use the soup to form a count vectorizer or TF-IDF vectorizer. We do this by first calculating the *TF*, which is the number of

words in a sentence and then calculating the *IDF* by counting the total number of sentences in the document.

We then multiply TF and IDF to form the *TF-IDF vectorizer*.

6. **Cosine Similarity**

   We then calculate the θ of all the coauthors present in the dataset. Cosine similarity is a statistic for determining how similar publications are independent of size. It uses mathematics to calculate the cosine of the angle or θ created by two vectors projected in a multi-dimensional space. In this example, the two vectors I'm talking about are arrays holding the word counts of two documents.

   When plotted on a multi-dimensional space with each dimension corresponding to a word in the document, the cosine similarity captures the direction (the angle) of the texts rather than the amplitude.

$$Similarity = cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=0}^{n} A_i B_i}{\sqrt{\sum_{i=0}^{n} A_i^2}\sqrt{\sum_{i=0}^{n} B_i^2}}$$

   In this, θ is the angle between A & B, where A and B refers to two non-zero vectors.

7. **Fetch**

   We now fetch the titles of the papers present in our database. These papers are fetched based on the indices.

8. **Similarity Score**

   By using the title and abstract of the papers, along with the cosine similarity between different authors, we can calculate the final *similarity score*.

   This score is tells how similar two papers are by comparing the arrays of word count of both the papers. The score ranges from 0 to 1, with 0 meaning that the papers are the most similar to each other.

The code for the same can be seen in appendix A.5

9. **Recommend Collaborators**

   Now that we have built our model, we can start recommending relevant research papers and scholars to the users. An example input can be seen in the figure 3.8.

```
get_recommendations('Dual Recurrent Attention Units for Visual Question Answering', cosine_sim)
```

Figure 3.8: Sample Input

The list of libraries and other tools used in this project can be found in the appendix A.1

Chapter 4

Results

4.1 Result

As seen in the previous section, our Recommender model takes the *title* of the paper as an input and provide a list of similar relevant papers to study.

It also uses **cosine similarity** or θ along with the title input to provide the users with potential scholars for collaboration.

4.1.1 Recommendations result

Now let's take a look at some of the example inputs and how our model handled their results. :

1.

```
In [32]: get_recommendations('Dual Recurrent Attention Units for Visual Question Answering', cosine_sim)
         0
Out[32]:
```

author	title	co_author
Robert E. Tarjan	pix2code: Generating Code from a Graphical User Interface Screenshot	Jean-Marie Nicolas
Robert E. Tarjan	Correlational Neural Networks	Jean-Marie Nicolas
Jean-Marie Nicolas	Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks	J. B. Bocca
D. Chan	Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation	M. Wallace
M. Wallace	Learning what to share between loosely related tasks	J. C. Freytag
J. C. Freytag	A Deep Reinforcement Learning Chatbot	J. B. Bocca
J. B. Bocca	Generating Sentences by Editing Prototypes	J. C. Freytag
Andrew V. Goldberg	A Deep Reinforcement Learning Chatbot (Short Version)	Robert E. Tarjan
Robert E. Tarjan	Document Image Coding and Clustering for Script Discrimination	Daniel D. Sleator
Daniel D. Sleator	Tutorial on Answering Questions about Images with Deep Learning	Wolfgang Bibel

Figure 4.1: Output 1

Based on our input of title, we goal a similarity score of 0 as that title exists in our dataset. Similarly, we get a score for every title.

Now it can be seen in Figure 4.1, that the top paper recommended is *Generating Code from a GUI Screenshot* by **Robert E. Tarjan**. Hence, this paper must be most similar to our input title.

We can also see that it's coauthored by **Jean-Marie Nicolas**. If we look at the next entries in our output, we will not only find Robert E. Tarjan again, but also Jean-Marie Nicolas. We got papers written by him recommended to us based on our cosine similarity.

These coauthors are presented on the output as they can be potential collaborators to us based on their previous work. Similarly, we will find other co-authors such as J.B Bocca, M. Wallace etc. recommended to us.

2.

Similarly we can see our output 2 in Figure 4.2.

```
In [34]: get_recommendations('Learning what to share between loosely related tasks', cosine_sim)
         3
```

Out[34]:

author	title	co_author
J. B. Bocca	Generating Sentences by Editing Prototypes	J. C. Freytag
M. Wallace	Towards better decoding and language model integration in sequence to sequence models	J. C. Freytag
Jean-Marie Nicolas	Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks	J. B. Bocca
J. C. Freytag	A Deep Reinforcement Learning Chatbot	J. B. Bocca
Lawrence Koved	Domain Adaptive Neural Networks for Object Recognition	J. B. Bocca
Jean-Marie Nicolas	Monitoring Term Drift Based on Semantic Consistency in an Evolving Vector Field	J. B. Bocca
J. C. Freytag	Neural Machine Translation by Jointly Learning to Align and Translate	J. B. Bocca
Lawrence Koved	BlackOut: Speeding up Recurrent Neural Network Language Models With Very Large Vocabularies	J. B. Bocca
Robert L. Brown	Semantically Decomposing the Latent Spaces of Generative Adversarial Networks	Peter J. Denning
Peter J. Denning	Clustering with Deep Learning: Taxonomy and New Methods	Peter J. Denning

Figure 4.2: Output 2

In this example, we get *Generating Sentences by Editing Prototypes* by J.B Bocca as our top recommended paper. This paper is coauthored by *J.C. Freytag* and we can see in the 4th entry that we then get JC Freytag and his paper recommended to us.

3.

In our last example, as shown in Figure 4,3 we can again see the top recom-

mended paper and based on it, the coauthors suggested to us.

```
In [35]: get_recommendations('Gated Graph Sequence Neural Networks', cosine_sim)
         149
```

Out[35]:

author	title	co_author
Eli Upfal	A Factorization Machine Framework for Testing Bigram Embeddings in Knowledgebase Completion	David Peleg
David Peleg	Neural Networks for Joint Sentence Classification in Medical Paper Abstracts	David Peleg
Srinivas Devadas	De-identification of Patient Notes with Recurrent Neural Networks	David Peleg
David Peleg	Discourse-Based Objectives for Fast Unsupervised Sentence Representation Learning	David Peleg
Avi Wigderson	Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation	David Peleg
Eli Upfal	Transfer Learning for Speech Recognition on a Budget	David Peleg
Srinivas Devadas	Uncertainty Estimates for Efficient Neural Network-based Dialogue Policy Optimisation	David Peleg
David Peleg	Gated Graph Sequence Neural Networks	David Peleg
Avi Wigderson	Deep Reinforcement Learning in Large Discrete Action Spaces	David Peleg
Wolfgang Bibel	Dual Recurrent Attention Units for Visual Question Answering	Jean-Marie Nicolas

Figure 4.3: Output 3

4.2 Performance Metric

In this section, we attempted to assess our model's performance and compare it to other ML models.

Our model's performance depends upon the cosine similarity function we used to find the θ between different coauthors. Hence, we will be comparing it's performance with KNNBasic which is a basic collaborative filtering algorithm [16], and SVD algorithm, which is equivalent to Probabilistic Matrix Factorization [16].

To measure the performance of our model, we needed to follow few steps:

- Divide the scholar datastet into 2 parts, Actual dataset containing authors based on cosine similarity and a Predicted dataset.
- The actual dataset is then shuffled and converted into predicted. We now have a dataset with Actual cosine value results and one with our Predicted values. .Shuffling also prevents over-fitting.
- Both of these datasets are then passed to *Mean Squared Error*. The mean squared error (MSE) of a regression line reflects how close it is to

a set of points. By squaring the distances between the points and the regression line, it achieves this (these distances are the "errors").

$$MSE = 1/n \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

The average squared difference between the estimated and actual values is measured by Mean Squared Error or MSE. The variables y_i and $\tilde{y}_i$ indicate observed and anticipated values, respectively.

We also calculate Root Mean Square Error for our recommendation model. The formula for the same can be seen below:

$$RMSE = \sqrt{1/n \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2}$$

- We now define **SVD** and **KNNBasic** Algorithm as a benchmark against our defined model.

  We perform cross-validation on them and then measure the MSE & RMSE value of the two algorithms.

- We can now combine Cosine, SVD and KNNBasic algorithms and combine them to see how our model performs.

The table below shows the value of of MSE and RMSE of all three of the algorithms, Cosine similarity, SVD and KNNBasic:

Table 4.1: Performance Metric

Algorithm	test_mse	test_rmse
KNNBasic	108.385814	10.410816
SVD	108.388243	10.410938
Cosine	40.379386	6.354478

The same can be seen in the graphs below:

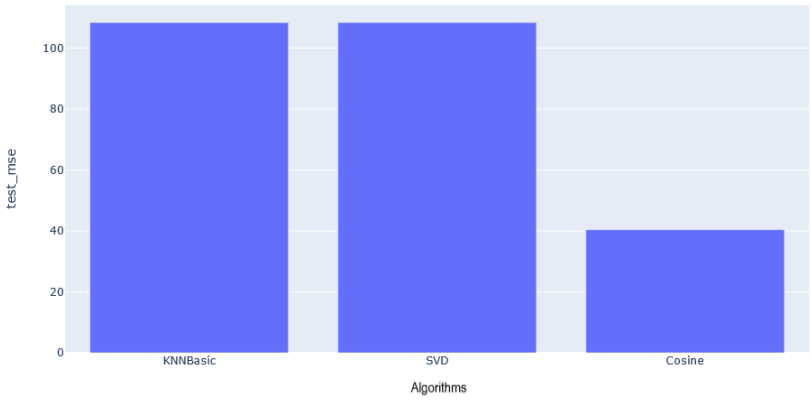


Figure 4.4: Algorithms v/s test_mse

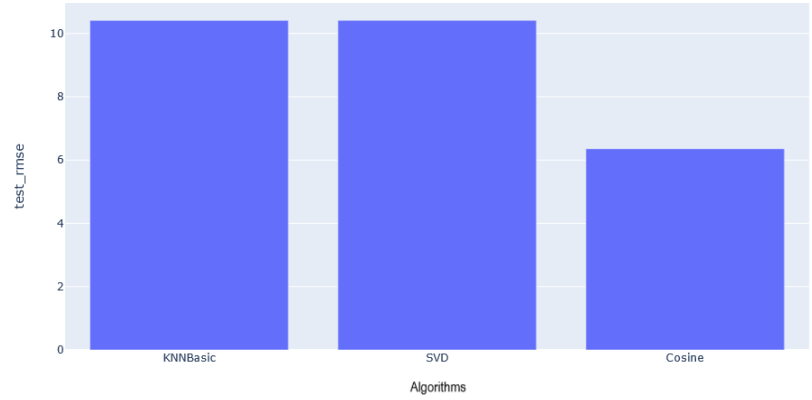


Figure 4.5: Algorithms v/s test_rmse

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

In recent years, Scholars have produced large-scale publications never seen before. The age of scholarly big data has arrived in academic society [1]. And due to the simple and easy access to a lot of scholarly data, the topic of information overload has received a lot of attention. Scholars have a hard time finding needed papers to study and meeting new people to collaborate with.

Our main goal was to thus build a *An Expert Scholar Recommendation System* that assists the users in not only finding the right papers to study, but also use a scholar *co-author network* to suggest potential scholars for collaboration.

To achieve this, we built a recommender system that recommends appropriate papers to study based on Count Vectorizer and also recommends potential scholars for collabortaion by building a co-author network and calculating the Cosine similarity between each coauthor.

First the title and abstract of all the papers in our dataset were taken, cleaned and merged to form a word soup. All the English stopwords were then removed from this soup and it was also lemmatized.

After this, TF-IDF was calculated and using that we found out the count vectorizer. Similarly, θ between all coauthors was calculated.

By using the title and abstract of the papers, along with the cosine similarity between different authors, we were able to calculate the similarity score.

Experimental results and performance metrics, MSE and RMSE indicates that our model performs exceptionally better than already established recommender algorithms like SVD and KNNBasic.

Hence, our model not only recommends research papers and potential scholars accurately, it also recommends them at a much better performance. In some cases, we could see an improvement of almost 300% in our recommendation model.

5.2 Future Scope

Due to the vast amount of material already available, researchers are finding it difficult to identify academic publications that are most relevant to their current work or the study subject in which they are interested.

As a result, a lot of research is being done on how to create successful academic paper recommender systems, which are support systems that help users find publications.

Information retrieval techniques such as Google's PageRank or Common Neighbours are used in the past by scholars to find the relevant papers to study. But all of these come with their drawbacks, especially considering the information overload in the academic society.

To solve this problem, we built a Context-aware Recommender System to navigate the scholarly big data and assist the users in not only finding the right papers to study, but also use a scholar co-author network to suggest potential scholars for collaboration.

This model can now be refined and used by scholars to find relevant papers. But the dataset only captures a snapshot of all the available papers at a given time. This reduces the accuracy of our model as it misses the latest papers

published. In the future, we wish to record the academic papers in real-time in order to make sure that we are recommending the best and most relevant papers to our users. In order to achieve this, we can extract papers from academic networks like *Google Scholar* or *Microsoft Academic*.

In the case of scholar recommendation for collaboration, we are using the coauthor network extracted from the academic papers published by them. In this method, we are only considering the network topology without considering scholars' academic information. But the performance of collaborator recommender can increase significantly by using network embedding in academic scholar networks like Linkedin, Research Gate etc.

Appendices

Appendix A

Code of Various Functions

A.1 Libraries needed

```
# Standard Libraries
import pandas as pd
import numpy as np
import json

# Data Preprocessing & NLP
import nltk
import re
import string
import gensim
from textblob import Word

from gensim.utils import simple_preprocess
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer, SnowballStemmer
from nltk.stem.porter import *
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import MultiLabelBinarizer
# nltk.download('all')
# nltk.download('punkt')

# Models
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import (RandomForestClassifier, GradientBoostingClassifier)
from sklearn.neighbors import NearestNeighbors
from sklearn.model_selection import GridSearchCV
from sklearn.metrics.pairwise import linear_kernel
```

A.2 Data Cleaning & Creating Soup

```
# Data Cleaning
def clean_text(text):
    # remove everything except alphabets
    text = re.sub("[^a-zA-Z]", " ", text)
    # remove whitespaces
    text = ' '.join(text.split())
    text = text.lower()

    return text
```

```
# creating clean text feature
features = ['title', 'abstract']
for feature in features:
    arxivData['clean_' + feature] = arxivData[feature].apply(clean_text)
```

A.3 Lemmatization and Creating Word Cloud

```
def random_color_func(word=None, font_size=None, position=None,
                      orientation=None, font_path=None, random_state=None):
    h = int(360.0 * 55.0 / 255.0)
    s = int(100.0 * 255.0 / 255.0)
    l = int(100.0 * float(random_state.randint(70, 120)) / 255.0)
    return "hsl({}, {}%, {}%)".format(h, s, l)

def freq_words(x, terms = 30):
    all_words = ' '.join([text for text in x])
    all_words = all_words.split()

    freq_dist = nltk.FreqDist(all_words)
    words_df = pd.DataFrame({'word':list(freq_dist.keys()), 'count':list(freq_dist.values())})

    fig = plt.figure(figsize=(21,16))
    ax1 = fig.add_subplot(2,1,1)
    wordcloud = WordCloud(width=1000, height=300, background_color='black',
                          max_words=1628, relative_scaling=1,
                          color_func = random_color_func,
                          normalize_plurals=False).generate_from_frequencies(freq_dist)

    ax1.imshow(wordcloud, interpolation="bilinear")
    ax1.axis('off')

    # select top 20 most frequent word
    ax2 = fig.add_subplot(2,1,2)
    d = words_df.nlargest(columns="count", n = terms)
    ax2 = sns.barplot(data=d, palette = sns.color_palette('BuGn_r'), x= "count", y = "word")
    ax2.set(ylabel= 'Word')
    plt.show()
```

```
# Lemmatization process
'''
Words in the third person are changed to first person and verbs in past and future tenses are changed into the present by the
lemmatization process.
'''
lemmatizer = WordNetLemmatizer()

def tokenize_and_lemmatize(text):
    # tokenization to ensure that punctuation is caught as its own token
    tokens = [word.lower() for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(sent)]
    filtered_tokens = []

    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    lem = [lemmatizer.lemmatize(t) for t in filtered_tokens]
    return lem
```

A.4 TF-IDF Vectorizer

```
# TfIdf matrix transformation on clean_summary column
tfidf_matrix = tfidf_vec.fit_transform(predicted)
# Compute the cosine similarity
pred_cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
pred_cosine_sim
```

A.5 Function to get Recommendations

```
def get_recommendations(title, similarity):

    idx = indices[title]
    print(idx)
    # pairwsie similarity scores
    sim_scores = list(enumerate(similarity[idx]))
    # sorting
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:11]

    article_indices = [i[0] for i in sim_scores]
    # Return the top 10 most related articles
    return arxivData[['author', 'title', 'co_author']].iloc[article_indices].style.hide_index()
```

Appendix B

Performance Metric

B.1 Defining Base Algorithms

```
def calculate_ratings(title_id, user_id):
    if title_id in df_ratings:
        cosine_scores = similarity_matrix_df[user_id] #similarity of id_user with every other user
        ratings_scores = df_ratings[title_id]
        index_not_rated = ratings_scores[ratings_scores.isnull()].index
        ratings_scores = ratings_scores.dropna()
        cosine_scores = cosine_scores.drop(index_not_rated)
        ratings_similarity_score = np.dot(ratings_scores, cosine_scores)/cosine_scores.sum()
    else:
        return 2.5
    return ratings_similarity_score
```

```
#evaluates on test set
def score_on_test_set():
    title_coauthor_pairs = zip(arxivData['title_id'], arxivData['user_id'])
    print(title_coauthor_pairs)
    predicted_ratings = np.array([calculate_ratings(title_id, user_id) for (title_id, user_id) in title_coauthor_pairs])
    true_ratings = np.array(arxivData['coauthor_id'])
    mse_score = mean_squared_error(true_ratings, predicted_ratings)
    mse_score = sklearn.metrics.mean_squared_error(true_ratings, predicted_ratings)
    rmse_score = math.sqrt(mse_score)
    return mse_score, rmse_score
test_set_score = score_on_test_set()
print(test_set_score)
```

B.2 Plotting the graphs

```
df = list(surprise_results['test_mse'].values/10000)
cosine_rmse = test_set_score[0]
df.append(cosine_rmse)
```

```
import plotly.express as px
name = ['KNNBasic','SVD', 'Cosine']
fig = px.bar(df, x=name, y='test_mse')
fig.show()
```

```
df = list(surprise_results['test_rmse'].values/100)
cosine_rmse =test_set_score[1]
df.append(cosine_rmse)
```

```
import plotly.express as px
name = ['KNNBasic','SVD', 'Cosine']
fig = px.bar(df, x=name, y='test_rmse')
fig.show()
```

References

[1] S. Khan, X. Liu, K. Shakil, and M. Alam, "A survey on scholarly data: From big data perspective," *Information Processing & Management*, vol. 53, pp. 923–944, 07 2017.

[2] X. Cai, J. Han, W. Li, R. Zhang, S. Pan, and L. Yang, "A three-layered mutually reinforced model for personalized citation recommendation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 6026–6037, Dec. 2018.

[3] X. Kong, H. Jiang, Z. Yang, Z. Xu, F. Xia, and A. Tolba, "Exploiting publication contents and collaboration networks for collaborator recommendation," *PLoS ONE*, vol. 11, 2016.

[4] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923615000627

[5] M. A. Domingues, A. M. Jorge, and C. Soares, "Dimensions as virtual items: Improving the predictive ability of top-n recommender systems," *Information Processing & Management*, vol. 49, no. 3, pp. 698–720, 2013, personalization and Recommendation in Information Access. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457312001069

[6] D. Liben-nowell and J. Kleinberg, "The link prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, 01 2003.

[7] Y. Liang, Q. Li, and T. Qian, "Finding relevant papers based on citation relations," in *Web-Age Information Management*, H. Wang, S. Li, S. Oyama, X. Hu, and T. Qian, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 403–414.

[8] B. Gipp, J. Beel, and C. Hentschel, "Scienstein: A Research Paper Recommender System," in *Proceedings of the International Conference on Emerging Trends in Computing (ICETiC'09)*, Kamaraj College of Engineering and Technology India. Virudhunagar, India: IEEE, Jan. 2009.

[9] N. Ford, *The essential guide to using the web for research.* Sage, 2011.

[10] Z. Liu, X. Xie, and L. Chen, "Context-aware academic collaborator recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1870–1879. [Online]. Available: https://doi.org/10.1145/3219819.3220050

[11] "The Common Neighbors algorithm - Neo4j Graph Algorithms User Guide," 2021-08-15. [Online]. Available: https://neo4j.com/docs/graph-algorithms/current/labs-algorithms/common-neighbors/

[12] C. Cechinel, M. Ángel Sicilia, S. Sánchez-Alonso, and E. García-Barriocanal, "Evaluating collaborative filtering recommendations inside large learning object repositories," *Information Processing & Management*, vol. 49, no. 1, pp. 34–50, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457312000921

[13] T.-P. Liang, Y.-F. Yang, D.-N. Chen, and Y.-C. Ku, "A semantic-expansion approach to personalized knowledge recommendation," *Decision Support Systems*, vol. 45, no. 3, pp. 401–412, 2008, special Issue Clusters. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923607000814

[14] F. M. Belém, E. F. Martins, J. M. Almeida, and M. A. Gonçalves, "Personalized and object-centered tag recommendation methods for web 2.0 applications," *Information Processing & Management*, vol. 50, no. 4, pp. 524–553, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457314000181

[15] C. B. Clement, "arxiv." [Online]. Available: https://arxiv.org/abs/1905.00075

[16] "Building and Testing Recommender Systems," 2021-08-20. [Online]. Available: https://towardsdatascience.com/building-and-testing-recommender-systems-with-surprise-step-by-step-d4ba702ef80b/

Table B.1: Project Detail

Student Details

Student Name	Vasu Agarwal		
Registration Number	170911052	Section/Roll No.	A/01
Email Address	vki.vasu@gmail.com	Phone No.(M)	9113694237

Project Details

Project Title	Expert Recommendation System for Scholarly Collaboration		
Project Duration	6 Months	Date of Reporting	10-02-2021

Internal Guide Details

Faculty Name	Dr. Tribikram Pradhan
Designation	Assistant Professor
Full Contact Address with PIN Code	Department of Information and Communication Technology, Manipal Institute of Technology, Manipal-576104
Email Address	tribikram.pradhan@manipal.edu
Faculty Name	Mr. Santhosh Kamath
Designation	Assistant Professor - Selection Grade
Full Contact Address with PIN Code	Department of Information and Communication Technology, Manipal Institute of Technology, Manipal-576104
Email Address	santosh.kamath@manipal.edu

August 4, 2021

Vasu Agarwal
K.G−119, Kavi Nagar
Ghaziabad Uttar Pradesh
201002

Dear Vasu Agarwal,

Further to your recent meetings and discussions with us, we are pleased to offer you employment with Optum Global Solutions (India) Pvt. Ltd. (formerly known as QSSI Technologies India Pvt. Ltd.) ("the Company") a UnitedHealth Group Company, in the position of **Software Engineer** at [redacted] Your work location shall be at Company's office located at **Bangalore,Marathahalli**. The terms and conditions of your employment are set out hereinafter:

EMPLOYMENT

We are pleased to extend this offer to you basis the selection process administered. Your effective date of joining shall be no later than **August 9, 2021**. Your employment with the Company shall be subject to the timely submission of the following listed mandatory documents for background verification purposes, to be submitted prior to or latest by your Start Date. Successful pre and/or post−employment background checks, accuracy of the testimonials and information provided by you and your being free from any contractual restrictions preventing you from accepting this offer or starting work with us on the above−mentioned date, are required for your employment with the company:

(i) Highest Degree Certificate
(ii) PAN Card OR Passport
(iii) Relieving Letter/ Experience Letter from all the organizations worked in last 5 years, except for the immediate last employer for which you will be granted 45 days from your start date

You, if so asked by the Company, shall disclose on your own behalf and, if married, on your spouse's behalf full details of any external directorships held and any personal business interests including partnerships, shareholdings and trusteeships; involvement in any other business ventures involving unlimited liability; personal liabilities in connection with business activities; and involvement in other positions external to the Company and your employment will be subject to acceptance by the Company of those external interests.

Please note that if during the pre or post−employment background checks, the background checking agency gives a negative report or in the event of unsatisfactory result of your pre or post−employment background checks, this letter of appointment shall stand revoked automatically (whether you have accepted it or not) and, if you have already commenced employment with the Company, such employment shall automatically terminate without giving rise to any claim for compensation or damages in your favor, but without prejudice to Company's rights and remedies against you.

PROBATION

You shall serve a minimum probation period of **180 days** from the date of your joining the Company ("Probation") following which you shall get confirmed into the Company by default unless you receive a letter for confirmation extension. The Company reserves the right to extend the probation period for an additional Ninety (90) days in the event that your performance is not up to expectation.

Your performance shall be evaluated according to your efficiency, punctuality, conduct, maintenance of discipline and in accordance with the Company's regulations or policies existing now or in future. It shall be your responsibility to read, peruse and follow Company's regulations/policies, hardcopies which shall be made available to you upon request, but which otherwise are available on the Company's website.

During the period of Probation, either the Company or you may at any time terminate your employment without cause by giving in writing to the other party, Thirty (30) days notice or in lieu thereof a sum equal to the amount or pro−rated amount of salary which would have accrued to you during the period or remaining period of notice. You shall not be entitled to any notice pay if your employment is terminated in accordance with condition 7.6 of the Appendix 3 to this letter of appointment.

PLACE OF POSTING

Your initial place of posting shall be at the Company's office located at **Bangalore,Marathahalli**. The Company works across different geographies providing services to its clients and you may be required to go through appropriate induction and orientation along with necessary training programme. The training is given to ensure that you are compliant with the best practices followed by the Company on a worldwide basis. However, your services are transferable and you may be assigned/ transferred in India or outside India to serve the Company in any of its existing or future offices or any of its group companies or associates. It is a condition of your employment that you comply with any such requirements of the Company. The transfer arrangement shall not deem to constitute a change in your conditions of service.

Notwithstanding the above, you may however be required to work at any other place that the Company may deem fit and as may be required from time to time. You may also be seconded, deputed or transferred to any other person/company associated with the Company whether in India or abroad. In such a case your relocation expenses shall be borne by the Company and your reimbursement shall be as per the relocation policy of the Company.

Your place of work shall change in case of any relocation of the Company's offices, for which you shall be entitled to reimbursement in consonance with the relocation policy of the Company.

The Company operates on a 24X7 basis and is open for 365 days in a year.

PERFORMANCE OF DUTIES

You shall be assigned with all the duties and responsibilities of the **Software Engineer** and such other duties on behalf of the Company, as may be reasonably assigned from time to time by the Company's management.

COMPENSATION

As compensation for services to be rendered, you shall be paid an annual fixed salary of [redacted]. Your cost to the Company (CTC) shall be [redacted] per annum. A detailed compensation structure is provided along with this letter of appointment.

The salary shall be payable on a monthly basis in arrears on or about the last working day of each calendar month, but in no case later than the 7th day of the succeeding calendar month. Please note that your salary details are highly confidential and should not be disclosed inside or outside the Company by you in any manner whatsoever and any failure on your part to adhere to this obligation shall be considered as serious breach of the terms of this letter of appointment.

DEFERRED SIGN−ON BONUS

You shall be entitled to a total sign on bonus of [redacted]. Amount of [redacted] will be payable to you at the time of payment of your first salary and remaining amount of [redacted] will be paid to you post completion of **12** months of service. In the event, your employment with the Company is terminated either by you or by the Company for any reason whatsoever, prior to completion of 1 (one) year from the date of each pay out, you will be required to repay the Company amount of sign−on bonus due as on date of termination forthwith. In case the total sign−on bonus amount or part thereof is not repaid to the Company by you, Company reserves the right to settle it against your full and final settlement amount. Sign−on bonus shall be governed by the applicable Company policy.

*Withholding taxes as applicable would be deducted from the above.

REWARDING RESULTS PLAN

You shall be eligible to participate in the Rewarding Results Plan in accordance with the terms and conditions of the Company, as amended from time to time. In this Rewarding Results Plan, you may be eligible to earn an annual performance−based incentive in addition to your basic salary. Your initial annual target incentive is **15%** of the fixed salary. It is clarified that no payment under this plan is guaranteed, and is subject to attainment of corporate and business unit's financial performance thresholds as well as individual performance ratings attained for the year on the Company's discretion. Basis this, your annual incentive payout could range from **0%− 15%** of the fixed salary. Any annual or other bonus payments are discretionary, non−binding and revocable for future years. Kindly refer to the rewarding results plan policy for any information regarding eligibility, payouts or any other terms and conditions associated with this plan.

The payment of all compensation and bonus / incentive, if any, shall be made in accordance with the relevant policies of the Company in effect from time to time, including normal payroll practices, and shall be subject to income tax deductions at source, as applicable. All requirements under Indian tax laws, including tax compliance and filing of tax returns, assessment etc. of your personal income, shall be fulfilled by you.

By accepting this letter of appointment you authorize the Company to deduct from your remuneration on termination of employment (including salary, salary in lieu of notice, sign on bonus, notice pay out etc.) all debts owed by you to the Company or any of its group companies/associates or any fine imposed by the Company as a discretionary penalty pursuant to the Company's disciplinary procedure.

TERMINATION OF EMPLOYMENT

During the Probation period, either Company or you may at any time terminate this letter of appointment without cause by giving in writing to the other party, Thirty (30) days notice. Company reserves the right either to accept your pay and allowance / towards the notice period or demand for actual service during the notice period. You shall not be entitled to any notice pay if your employment is terminated in accordance with condition 7.6 of the Appendix 3 to this letter of appointment.

After completion of the Probation period, either Company or you may at any time terminate this letter of appointment without cause by giving in writing to the other party, 90 days notice. The Company reserves the right either to accept your pay and allowance / towards the notice period or demand for actual service during the notice period. You shall not be entitled to any notice pay if your employment is terminated in accordance with condition 7.6 of the Appendix 3 to this letter of appointment.

However, notwithstanding the above, the Employee must refer to the Company's Separation Policy (as available on Company's intranet link) for the notice period days applicable to them based on their entity, grade and employment status at the time of resignation.

The notice period matrix, as provided under the Company's Separation Policy, shall be applicable with the change in employee job family, job role and employment status. The provisions of the notice period matrix, as provided under the Company's Separation Policy, shall over −ride the notice period as stipulated in the appointment contract or any other document issued before this date. No separate individual employee consent shall be necessary for applicability of this clause.

In case of any conflict pertaining to the notice period between this Offer letter and the prevalent Separation Policy of the Company, the contents of the Separation Policy shall take precedence over the terms of this offer letter and shall be binding on the employee.

Your employment shall also be governed by the standard terms and conditions, which are annexed hereto as Appendix 3 and the same shall form an integral part of this letter of appointment.

Your employment is conditional upon your acceptance of the standard terms and conditions and the specific provisions contained in Appendix 3.

Kindly sign and return the duplicate copy of this letter of appointment along with the Appendixes, as a token of your acceptance of the terms and conditions set out herein. Also, please initial each page of this letter of appointment and the Appendixes.

Please note that by signing this letter of appointment, you have agreed to accept the employment with the Company on the terms and conditions set out herein. Upon your signature and return to us, this letter of appointment shall be treated as an employment agreement and the terms and conditions of this letter of appointment shall govern your employment with the Company.

This letter of appointment shall automatically stand revoked in the event you do not join the Company on or before the effective date mentioned in this letter of appointment.

It is a pleasure to welcome you as a part of **Optum Global Solutions (India) Pvt. Ltd.,** We are confident that your employment with the Company shall prove mutually beneficial and rewarding and we look forward to having you join us.

Congratulations and welcome to **Optum Global Solutions (India) Pvt. Ltd.** You shall be receiving an e−mail communication from us shortly for your new hire orientation training. You are requested to attend the same on your first day of reporting along with the documents as mentioned in the Appendix '2'. Should there be a change in your start date, it is mandatory that the same be communicated to us a week in advance.

Vasu Agarwal, we thank you for considering **Optum Global Solutions (India) Pvt. Ltd.** as your future employer! We have bold objectives:

- Improve the lives of others;
- Change the landscape of health care forever;
- Leave the world a better place than we found it.

Joining us, shall put you amongst a team that is committed to excellence in everything we do. We are passionate, energetic and focused. You'll be sharing a culture of leadership and excitement as you begin to do **your life's best work.**[SM]

For Optum Global Solutions (India) Private Limited

Sumek Gopal
Vice President I Human Capital

I accept this letter of appointment on the terms and conditions as described herein.

ACKNOWLEDGEMENT:

Vasu Agarwal

Vasu Agarwal

Date: 05-August-2021

Expert recommendation System

ORIGINALITY REPORT

14%	7%	4%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Amrita Vishwa Vidyapeetham Student Paper	4%
2	Jieun Son, Seoung Bum Kim. "Academic paper recommender system using multilevel simultaneous citation networks", Decision Support Systems, 2018 Publication	2%
3	Submitted to Manipal University Student Paper	2%
4	Submitted to Cardiff University Student Paper	1%
5	Submitted to University College London Student Paper	1%
6	apps.dtic.mil Internet Source	1%
7	Submitted to Universiti Teknologi Malaysia Student Paper	1%
8	www.scribd.com Internet Source	<1%

9	Submitted to Indiana University Student Paper	<1 %
10	Submitted to Texas A & M University, Kingville Student Paper	<1 %
11	www.nrsp.org Internet Source	<1 %
12	Submitted to Birkbeck College Student Paper	<1 %
13	Rahul Budhraj, Pooja Kherwa, Shreyans Sharma, Sakshi Gill. "Chapter 50 Efficient Recommendation System Using Latent Semantic Analysis", Springer Science and Business Media LLC, 2022 Publication	<1 %
14	hal.archives-ouvertes.fr Internet Source	<1 %
15	irep.ntu.ac.uk Internet Source	<1 %
16	Yicong Liang. "Finding Relevant Papers Based on Citation Relations", Lecture Notes in Computer Science, 2011 Publication	<1 %
17	www.mdpi.com Internet Source	<1 %

18	Tribikram Pradhan, Sukomal Pal. "A hybrid personalized scholarly venue recommender system integrating social network analysis and contextual similarity", Future Generation Computer Systems, 2019 Publication	<1 %
19	dc.uwm.edu Internet Source	<1 %
20	docear.org Internet Source	<1 %
21	worldcomp-proceedings.com Internet Source	<1 %
22	www.coursehero.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 3 words
Exclude bibliography	On		