A

Dissertation Report on

Content-based Research Paper Recommendation (C-RPR) System using Deep Learning

Submitted

in partial fulfilment of the requirements for the degree of

Master of Technology

in

Computer Science & Engineering

by

Miss Mane Seema Ramchandra

Roll No. 2030002

Under the Supervision of

Prof. Ashwini Patil

DEPARTMENT OF COMPUTER ENGINEERING

K.E. Society's

Rajarambapu Institute of Technology, Rajaramnagar

(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

2021-2022

K.E. Society's

Rajarambapu Institute of Technology, Rajaramnagar

(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

CERTIFICATE

This is to certify that, Miss Mane Seema Ramchandra (Roll No-2030002) has successfully completed the dissertation work and submitted dissertation report on "Content-based Research Paper Recommendation (C-RPR) System using Deep Learning" for the partial fulfillment of the requirement for the degree of Master of Technology in Computer Science and Engineering from the Department of Computer Science and Engineering, as per the rules and regulations of Rajarambapu Institute of Technology, Rajaramnagar, Dist: Sangli.

Date:

Place: RIT, Rajaramnagar

Prof. Ashwini Patil

Sign of Supervisor

Dr. L.L. Kumarwad	Dr. Sachin S. Patil
Sign of External Examiner	Sign of Head of Program
Dr Nagraj V. Dharwadkar	Dr. S. S. Gavade
Sign of Head of Department	Sign of PG Convener

DECLARATION

I declare that this report reflects my thoughts about the subject in my own words. I have sufficiently cited and referenced the original sources, referred or considered in this work. I have not misrepresented or fabricated or falsified any idea/data/fact/ source in this my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute.

Place: RIT, Rajaramnagar Name of Student: Miss Mane Seema Ramchandra

Date: Roll No: 2030002

ACKNOWLEDGEMENTS

I must mention several individuals and organizations that were of enormous help in the development of this work. I would like to express my sincere gratitude to my supervisor Prof Ashwini Patil, for his guidance, nurturing, encouragement and support in every stage of my study. His knowledge, kindness, patience, open-mindedness, and vision have provided me with lifetime benefits.

I also very thankful to HoP, HoD for their valuable suggestions, critical examination of work during the progress, I am indebted to them.

In addition, very energetic and competitive atmosphere of the Computer Science and Engineering Department had much to do with this work. I acknowledge with thanks to faculty, teaching and non-teaching staff of the department, Central library and Colleagues.

I sincerely thank to Dr. Mrs. S.S. Kulkarni, for supporting me to do this work and I am very much obliged to her. Last but not the least I thankful to my parents who constantly supported me in all aspects.

Place: RIT, Rajaramnagar   Name of Student:Miss Mane Seema Ramchandra
Date: Roll No:2030002

ABSTRACT

Recommender system is one of the most critical research areas in today's world because it helps users to find their interest in the internet. Due to the exponential growth of data every day on the internet, it has become a pervasive problem of information overload and finding relevant information. In recent years, it has become a widespread technique used by many e-commerce applications, such as article recommendations, movie recommendations, product recommendations, music recommendations to provide the right information to their customers In general, recommendation systems have been classified into collaborative and content-based filtering.

The number of papers that have been published has grown significantly each year, involving more contributors and a larger range of vocations. In order to reduce the effects of information overload, it may be helpful to suggest research papers to active researchers and research students. So, it becomes challenging for researchers to identify the most similar research papers and choose the best venue to publish them as a result of the large number of papers being submitted to various venues. The proposed recommendation system uses a content-based filtering approach using a deep neural network and helps researchers to submit their manuscripts to the most suitable venue and also recommends the most similar research paper to do their research in a smooth way. The C-RPR model uses a natural language processing technique where TF-IDF is used for vectorization, and the cosine similarity technique is used as a similarity measure to recommend similar papers. Also, Bi-LSTM and GRU is used to recommend an appropriate venue by training a model on a research publication dataset for computer science journals with attributes such as ID, title, abstract, author, venue, etc. Compared to other models that predominantly make use of machine learning algorithms and feature selection techniques, the proposed model produced better results with 74.55% accuracy using Bi-LSTM and 82.70% accuracy using GRU mechanism for batch size 32. For batch size 64, the proposed model produced better results with 69.72% accuracy using Bi-LSTM and 75.83% accuracy using GRU mechanism.

Keywords: Recommendation System, Content-Based Filtering, Bi-LSTM, TF-IDF, Cosine Similarity.

Contents

List of Figures

List of Tables

ABBREVIATIONS

C-RPR	Content-based Research Paper Recommendation
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
Bi-LSTM	Bi-directional Long Short Term Memory
GRU	Gated Recurrent Unit

Chapter 1

Introduction

The recommender system acts as an information filtering tool; they provide appropriate information as per the user's choice and interest. In recent years, it has become a widespread technique that is being used in many applications. In general, recommendation systems have been classified into collaborative and content-based filtering. Due to its significance, the recommender system has become one of the most important research areas in today's context. Though recommendations systems are being used for a quite long time, many research challenges and issues in the design of effective recommendation systems are yet to be addressed in an effective manner. Recommendation systems (RS) are a rapid transformation in e-commerce and play a very important role in many areas, such as product recommendation, movie recommendation, news recommendation, and book recommendation. Over the past few years, scientific article recommendations have become increasingly popular. It is getting more and harder for scholars to identify pertinent articles and suitable venues to submit their papers as the number of scholarly publications in various sorts of journals and conferences grows tremendously. A lot of journals and conferences are receiving a variety of articles. Therefore, recommending relevant research articles to busy researchers will help them stay current with their field of study and avoid information overload [1]. Recommender systems are software applications that suggest or recommend items or products (in the case of ecommerce) to users. These systems use user's preferences or interests (supplied as inputs) and an appropriate algorithm in finding the relevant or desired items or products. Recommender systems deal with information overload problems by filtering items that potentially may match the users' preferences or interests. These systems aid users to efficiently overcome the problem by filtering irrelevant information when users search for desired information. Recommendation methods help users in deal with the information overload problem. In the academic as well as the educational domain, the application of intelligent recommending technologies enhances the utilization efficiency of academic resources to a great extent, especially for newcomers and students in this area. Scientific papers are experiencing a new era named "big scholarly data" the number of published paper has sharply increased year after year, involving more participants and wider vocations. Therefore, recommending research papers to busy researchers and research students may reduce the impact caused by information overload [32].

The recommender system gathers data from various resources to make recommendations. These recommendations are made by considering the interests and previous history of user. This paper is proposed to recommend similar papers and appropriate venues. Many authors outperformed recommendations based on keywords and titles. The recommendation system using abstracts for authors is still under research. The proposed system performs recommendations of top N research papers and venues based on the abstract. The recommender system helps users choose items based on their interests by filtering products. Many recommendation algorithms emerge from the recommendation system, guiding users to find their interests in a large space [10, 11, 12].

The primary motive of any recommender system is to infer customers' interests using various data sources. These systems aim to extract the most relevant information from collected data for each user. While doing this recommender system pays attention to the user's personal preferences and interests. These systems find the match among people and products, which endorses their similarities. Depending on the information utilized to make recommendations, these systems can be divided into Content-based (CB) filtering or Collaborative Filtering (CF). CB techniques are based on profile of the target user's past interests the description of items. A CF technique is generally based on the notion that persons with similar preferences about certain items are likely to have alike preferences for other items [6], [32].

1.1 Types of Recommendation System

The three primary classifications of recommendation systems are Content-Based Filtering (CBF), Collaborative Filtering (CF), Graph-Based Method, and Hybrid Recommendation.

1. Content-Based Filtering (CBF): CBF is a traditional recommender system that suggests items with similar properties [6, 24, 25]. CBF is based on the item's description and takes into account previous user behavior. It generates a user's profile based on their history. For example, CBF extracts keywords from candidate papers, generates a user profile and calculates similarity [13]. High similarity scores are used to determine which papers are most similar to the user's search and which ones should be recommended to them. For example, if a user like area such as 'Machine Learning' then we can recommend him the context of titles related to 'Machine Learning'. WebPages, news, and publishing recommendation systems all utilize CBF to a large extent. The recommendations are calculated using cosine similarity measures using TF-IDF, Bag of Words etc.

   The recommendation process is performed in three steps: analyzing the content, learning the user profile, and filtering the recommendations. In the first step, data is pre-processed to extract important information. As part of data preprocessing, cleaning of the data is done using various approaches like imputing missing values, deleting stop words, deleting non-alphanumeric characters, lemmatized columns, etc. Later feature extraction is carried out to focus the attention on the most vital part of the information. In the second step, user preferences' data is collected and generalized to construct user profiles using machine learning techniques. The generated user profile is compared with stored profiles using various similarity metrics in the last step. This comparison is made using different similarity measurement techniques such as Cosine similarity, Euclidian distance, Pearson correlation coefficient, etc. The system also has a feedback component that helps to evaluate the usefulness of recommendations and take further corrective actions accordingly. In the feedback process, users' responses towards recommended items are collected.

2. Collaborative Filtering: CF is widely used in movie and music recommendation. In CF, the preferences and interests of two or more similar users are taken into account. The collaborative filtering method collects and explores information about users' past behaviour in terms of likes and dislikes. This includes monitoring the user's online behaviour and forecasting their interests by observing similarities of two users. In order to make recommendations for related products in the future, CF algorithms keep track of customer evaluations and ratings from the past. One such technique is matrix factorization. For example, if the watch histories of users 1 and 2 are highly similar, it is likely that if user 1 watches a movie, user 2 will watch the same or something close to it [15]. The CF method is based on collecting and exploring information about users' past behavior in terms of their likes and dislikes. This includes monitoring the user's online activities and predicting what they will like according to their similarity with other users. CF recommendation systems are easy to create and use; further domain knowledge is not required for giving recommendations in these systems. These systems offer unanticipated and diverse recommendation. The CF algorithms are divided into two classes memory-based approach and model-based approach. Memory-based approach remembers the likes and dislikes of every user; it gives personalized recommendations based on these likes and dislikes. Memory-based approaches can be further categorized as User-based approaches and Item-based approach. The system aims to compute resemblances between items and users. The similarity ratings are assigned to each item and user according to the computed resemblance. The recommender system finds neighbors for a given user or item using previously calculated ratings. User-based CF finds similarities between different users and each pair of users is given a rating. This rating indicates the correlation of one user with another user. The pair of users having higher ratings are considered neighbors. This system gives recommendations on basis of feedback given to an item by users who are neighbors of the target user. In item-based recommender system likenesses between different items are found. It assumes that items with similar user scores are likely to be of similar types [34], [35].

3. Hybrid recommendation: Hybrid recommendation system combines Content-

Based filtering (CBF) and Collaborative filtering to suggest a broader range of items to users. The idea of this approach is to combine the aforementioned recommendation techniques to make use of the advantages of one approach and fix the disadvantages of another approach. For instance, CF usually faces a cold-start problem triggered when a new item is added to the system and has no user ratings, whereas CB can tackle this issue since the prediction for new items is generally based on available descriptions of these items. A hybrid recommendation system increases system performance by addressing the shortcomings of each separate algorithm.

1.2 Techniques to Design a Recommendation System

There are lots of techniques used to design a recommendation system which simply takes input from user and give response to the user as per the interest and past behaviors.

1. Machine Learning and Deep Learning: The future of personalized content and product recommendations will be heavily shaped by deep learning (DL) and its ability to predict the next natural item or product for a visitor. In the last decade, the scientific world has been thrilled by the tremendous success of deep learning (DL) methods, playing a major role in how artificial intelligence (AI) has flourished over the past few years. The most notable of these revolutions have been made possible by DL in computer vision and natural language processing (NLP). The architecture of all deep learning models includes a multiple-layers structure, with every layer consisting of nodes that perform a particular mathematical operation, which is called an activation function. Various types of deep learning models include ANN, Convolution Neural Network (CNN), Recurrent Neural Network (RNN) are used to design Recommendation system [1]. There are many variations of RNN like LSTM, GRU are used to improve the presentation of the Recommendation system.
   The proposed research work uses sequential architecture for designing a Recommendation system. Earlier research work focuses on choosing appropriate hyper-parameters to improve the performance of the Recommendation system using LSTM (Long Short Term Memory) sequential model [31]. Bi-directional LSTM contains few hidden layers; all information from sentences

is passed through those layers. The proposed research work aims to have further improvements in accuracy of Recommendation systems and generate high-quality responses. The proposed model improves performance of Recommendation system by replacing LSTM model by using GRU with Beam search decoding. The working of GRU is the same as LSTM, but it uses a simplified structure. Unlike the LSTM, GRU has only three gates as the update gate, reset gate, and current memory gate. These gates are used to choose what data ought to be passed to the yield [20], [21], [23]. Both LSTM and GRU models are useful to overcome vanishing and exploding gradient problems.

2. NLP: Natural Language Processing (NLP) is one of the key components in Artificial Intelligence (AI), which carries the ability to make machines understand human language. NLP allows machines to understand and extract patterns from such text data by applying various techniques such as text similarity, information retrieval, document classification, entity extraction, clustering. This is where the concepts of Bag-of-Words (BoW) and TF-IDF come into play. Various word embedding techniques are being used i.e., Bag of Words, TF-IDF, word2vec to encode the text data.

   The proposed system implements TF-IDF technique in natural language processing to deal with textual data and convert it into vectors by multiplying the term frequency and inverse document frequency, which generates feature vector space. These feature vectors are given as input to the cosine function to calculate the angle between two vectors. By computing the cosine angle between two papers, one can estimate how similar two documents are.

1.3 Research Challenges

There are many challenges faced while building Recommendation system.

1. Existing system [1] used NLP and machine learning techniques to recommend top K journals which suffers with less accuracy. So, the research gap found is that the accuracy can further be improved.

2. Accuracy needs to be improved by use of deep learning techniques such as LSTM and GRU instead of machine learning techniques.

3. Improvement in accuracy by various hyper-parameter tuning and comparison of them.

1.4 Motivation of the Present Work

The main aim of Recommender System is to recommend relevant papers and most suitable journal to users. These systems use dataset provided by user to perform experimentation. Recommender systems deal with information overload problems by filtering papers that potentially may match the title given as input. The main goal of proposed work is that when an author provides text query, then the model represents set of papers similar to the context provided and also recommend most suitable journal to submit the paper in a good journal.

1.5 Objectives of Present Work

1. To perform literature survey on content based recommendation system for research paper publications and identify gap for new work.
2. To study and implement an existing NLP content based recommendation system for computer science publications.
3. Proposing a novelty in existing NLP content based recommendation system for computer science publications to achieve improvement using deep learning.
4. To perform the performance comparison of existing NLP content based recommendation system for computer science publications and the proposed content based research paper recommendation systems using deep learning.

1.6 Layout of the Thesis

This work is arranged as follows: Chapter 1 gives brief overview of Recommendation system, its types, techniques, some research challenges and objectives of present work. In chapter 2 discuss about the history of the Recommendation system and literature survey related to deep learning based Recommendation system. Chapter 3 discusses the theoretical concept of RNN algorithm and its variations like LSTM and GRU. Chapter 4 briefly discuss about methodology of proposed

Recommendation system. Chapter 5 describes the comparative experimental results of LSTM and GRU. Finally the chapter 6 describes the conclusion about proposed Recommendation system.

1.7 Closure

This section discusses the Recommendation system. Chapter 1 gives information about two characters of the Recommendation system and also gives an overview of some challenges faced while designing the Recommendation system. This section briefly discusses different Recommendation designing techniques. The research motivation and objective are also included in chapter 1.

Chapter 2

Literature Review

2.1 Introduction

For research purpose literature survey is very important part. It performs critical analysis on existing Recommendation system. Researchers can perform comparative study with the help of literature survey. It helps to find out gap or limitation of existing Recommendation system. It also helps to find out the problem statement. This Chapter covers different reviews carried out by the researchers for designing Recommendation framework to improve the presentation of the system. Following literature survey is based on various technique used for building Recommendation system such as Neural Network, Recurrent Neural Network and its variations.

The recommendation system is used in many areas, including research article recommendation, product recommendation, movie recommendation, restaurant recommendation, news recommendation, and book recommendation. A lot of these technologies are built on the idea of data mining, which extracts useful patterns from large data sets.

P Kumar et al. [33] conducted a survey of various recommendation systems based on filtering methods, challenging applications, and evaluation measures. The goal of the work is to familiarize practitioners and researchers with the many attributes and potential filtering methods of recommendation systems.

Wang et al. [1] represented a publication recommender system using a feature selection module to extract features and to generate feature vector space, and a softmax regression module to produce recommendations. The proposed recommendation system for articles in computer science covers 66 leading publishing venues, including IEEE, Springer, ACM, AAAI, and SIAM, across five digital libraries.

Mayank et al. [2] developed a recommendation system and proposed a ConfAssist framework having 91.6% accuracy of classification, which categorizes conferences and venues with similarity to other already categorized conferences.

Hassan et al. [3] implemented a recommendation system to recommend papers by considering the user's implicit and explicit feedback by using a recurrent neural network by finding latent semantic features of papers in the PubMed dataset.

Titipat et al. [4] worked on article content. The author used the Latent Semantic Analysis (LSA) model to analyze scientific posters. The Nearest Neighbor algorithm is utilized to obtain proposed elements while the Ricchio algorithm is used to make suggestions.

Hamed et al. [8] implemented one of the most popular techniques called LDA (Latent Dirichlet Analysis) for topic modelling and used it to fine tune relationships between topics. On the DBLP dataset, the author applied the Gibbs sampling algorithm.

By Muhammad et al. [9], they proposed a CFP recommender system to find an appropriate conference to submit papers to by generating a user profile and a cosine similarity technique is used for recommendation.

Braja et al. [16] used the approach of data reusability and implemented literature recommendations by adopting the Information Retrieval paradigm with MEDLINE articles from GEO datasets. The authors recommended top similar papers by using the cosine similarity technique and implemented the vectorization techniques including TF-IDF, BM25, word2vec, doc2vec, Latent Semantic Analysis, and Latent Dirichlet Allocation. As a result, BM25 performed better by showing good accuracy compared to other vectorization techniques.

Ebesu et al. [4] implemented a model of encoder-decoder architecture with the attention mechanism. By using a 1-dimension convolution across all conceivable word windows in a given context, the author used the TDNN in our encoder to capture long-term dependencies. Relu serves as a convolutional layer's nonlinear activation layer. The gated recurrent unit prevents the gradient problem from vanishing or exploding (GRU).

W. Huang et al. [32] proposed the distributed semantic representations of the terms and the cited articles focused on the local citation recommendation (or context-based recommendation). Author used neural network to forecast the citation for given the context.

Many authors outperformed recommendations based on keywords and titles to recommend similar papers. The use of machine learning algorithms to forecast the venues for a given abstract has been shown to be ineffective. This research is currently ongoing. With the use of a deep learning, this paper provides venue recommendations based on abstracts.

Chapter 3

Theoretical Background

This chapter introduces some theoretical background about deep learning technique used in Recommendation system. It includes in depth information about RNN based Recommendation system. To overcome the problem of vanishing exploding gradients problem, Bi-LSTM and GRU is introduced.

3.1 Recurrent Neural Network (RNN)

A RNN is type of neural system where the contribution of the current advance is yield from the past advance. In standard neural network all inputs and outputs are not dependent of each other. The inputs and outputs in conventional neural networks are all independent of one another, but in situations when it's necessary to anticipate the next word in a phrase, it's necessary to remember the prior words as well. RNN was developed as a result, and it utilised a Hidden Layer to address this problem. Hidden state, which retains some information about a sequence, is the primary and most crucial component of RNNs. The Recurrent Neural Network (RNN) has a ”memory” that retains every piece of information regarding the calculations that have been made. The same task is performed on all of the inputs or hidden layers to produce the output, and the same parameters are used for each input. Mostly RNN is use to make successive data. The fundamental element of RNN is hidden state, which recollects some data about information succession. This RNN architecture is used in encoder-decoder model. The RNN has major drawback known as gradient vanishing problem.

3.2 Sequential model (Sequential)

For a simple stack of layers with precisely one input tensor and one output tensor for each layer, a sequential approach is appropriate. The objective of sequence modelling is to foretell the subsequent entity in an input sequence (word or letter) based on the preceding entities. Deep learning's standard ANN or CNN are unsuitable for such inputs, which sparks the development of algorithms like RNN, LSTM, and Transformers, among others. When processing sequences of integers, a sequential model embeds each integer into a 64-dimensional vector before processing the vector series using an LSTM layer. The input sequence handled by algorithms such as RNN, LSTM, and GRU.

3.3 Long Short Term Memory (LSTM)

The LSTM depends on its gated structure. In the proposed model any dialogue or sentence is a pass to the LSTM encoder. Each word from a given input sentence is converted with respect to its vector form.

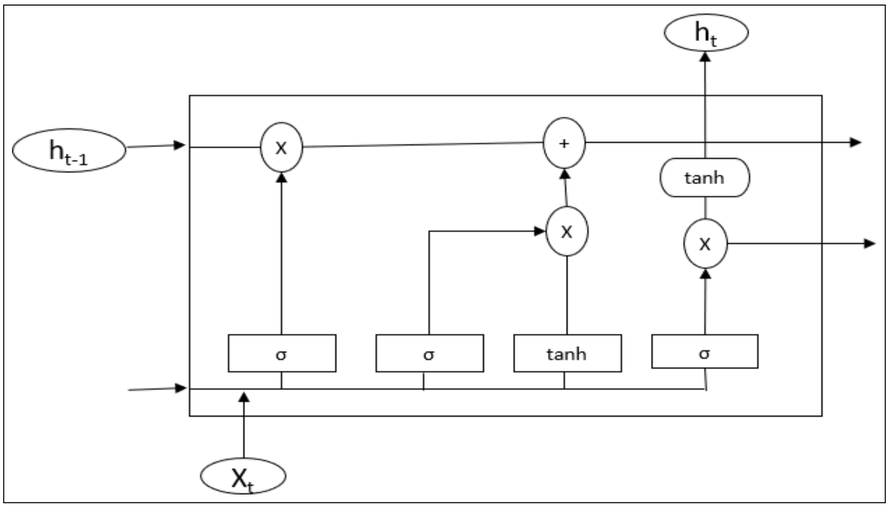


Figure 3.1: Long Short Term Memory Architecture

This vector form is a pass to the LSTM first layer which is known as the forget gate layer. Here LSTM encoder has to decide which information is going to be considered. Above figure 3.1 forget gate layer considerX_h, h_{t-1} is current step input and previous step output respectively. Then the output of the forget gate layer pass to the next input gate layer and output gate layer respectively. As show in above figure 3.1 sigmoid and threshold activation function used to calculate vector form of inputs. Finally, the LSTM encoder generates an output vector, which is an intermediate vector. This intermediate vector passes to the LSTM decoder which contains all information about the input sequence. The LSTM decoder uses an intermediate vector to generate the target sequence. LSTM stops decoding when the end of sentence token arrived. The LSTM decoder gives response as per the input given to the encoder.

3.4 Bi-Long Short Term Memory (LSTM)

In the LSTM architecture, there are three gates incorporated to memorise long-term dependencies. Though the LSTM models provide a strong prediction, they need to improve the forecast further. To achieve this, the Bi-LSTM model is introduced to offer additional training. It traverses the input data in both directions, forward and backward as shown in figure 3.2.

Recurrent neural networks that are bidirectional are basically just two separate RNNs combined. The networks may access both forward and backward information about the sequence thanks to this structure at each time step.

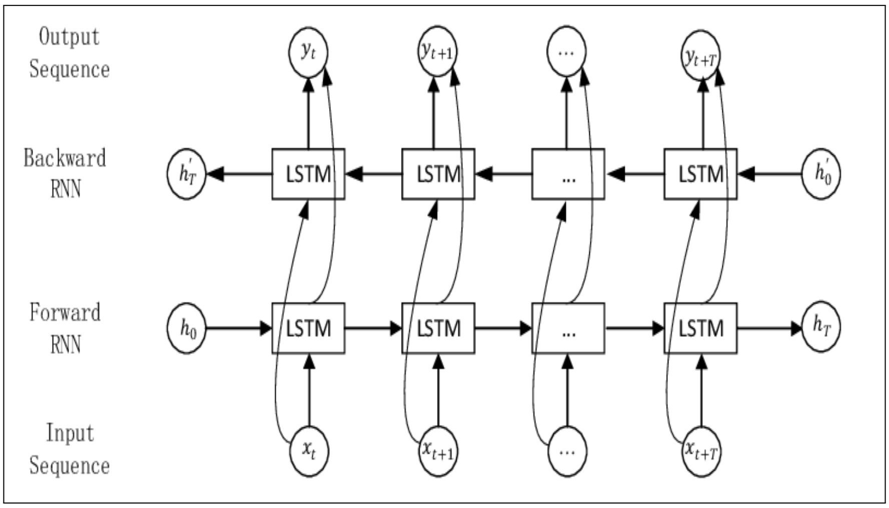


Figure 3.2: Bi-directional Long Short Term Memory Architecture

When using bidirectional, your inputs will be processed in two different directions: one from the present to the future and the other from the future to the present. This method differs from unidirectional in that information from the future is preserved in the LSTM that runs backward, and by combining the two hidden states, we can preserve data from both the present and the future at any given time.

3.5 Gated Recurrent Neural Network (GRU)

The vanishing exploding gradients problem is effectively resolved using GRU. The working of GRU and LSTM is similar but there is a difference in gate structure. Here vector form of the input sentence given to the GRU first layer. The first layer of the GRU encoder is the update gate layer. The update gate performs concatenation between the current input and previous output. This can be done using the following equation [33].

$$Z_t = (W^Z \times x_t + U^Z \times h_{t-1}) \tag{3.1}$$

Here x_t is current input given to our proposed model GRU encoder with respect to time t. The previous output with respect to time $t - 1$ is h_{t-1}. In the update, gate multiplication performs with respect to its corresponding weights. Here the weights are W^Z and U^Z. After performing concatenation the output of the equation is a pass to the activation function, which gives the output as Z_t. The update entryway encourages the proposed model to decide the amount of the past data regarding now is the ideal time should be passed along to what's to come. With the help of the update gate the proposed model eliminates the imperil of vanishing gradient problem. The output of the update gate is a pass to the reset gate. This gate is utilized to determine which information should be removed from the previous in the proposed model [33]. To calculate the output of the reset gate following equation is used.

$$r_t = (W^r \times x_t + U^r \times h_{t-1}) \tag{3.2}$$

The above equation is the same as the update gate equation. Here weights are W^r and U^r is multiplied with respect to x_t and h_{t-1}. This gate also removes unnecessary data from the proposed GRU encoder model. Here we need to store relevant information from the past using the below equation in GRU encoder with respect to its current content. In the above equation, the output of the reset gate is used as input and then in the proposed model tangent activation function applied [19].

$$h_t = tanh(W \times x_t + r_t \times U \times h_{t-1}) \tag{3.3}$$

$$h_t = Z_t \times h_{t-1} + (1 - Z_t) \times h_t \tag{3.4}$$

Finally a single vector is generated, which is the output of the proposed GRU encoder. The output of this equation depends on all the previous gate outputs. Then GRU decoder generates proper response as per input given. In the proposed model update entryway and reset door is utilized to choose what data ought to be passed to the yield. The GRU encoder-decoder model has the ability to keep all the data as long as feasible.

3.6 Closure

This chapter discuss in depth information about RNN based Recommendation system. The performance of the Recommendation system decreases due to vanishing exploding gradients problem. To overcome this issues this chapter discuss about Bi-LSTM and GRU.

Chapter 4

Methodology of Present Work

4.1 Introduction

This chapter discuss about proposed architecture and mathematical model of Bi-LSRM and Gated Recurrent Unit (GRU). This chapter gives detailed explanation of proposed Recommendation system.

4.2 Overview of proposed System

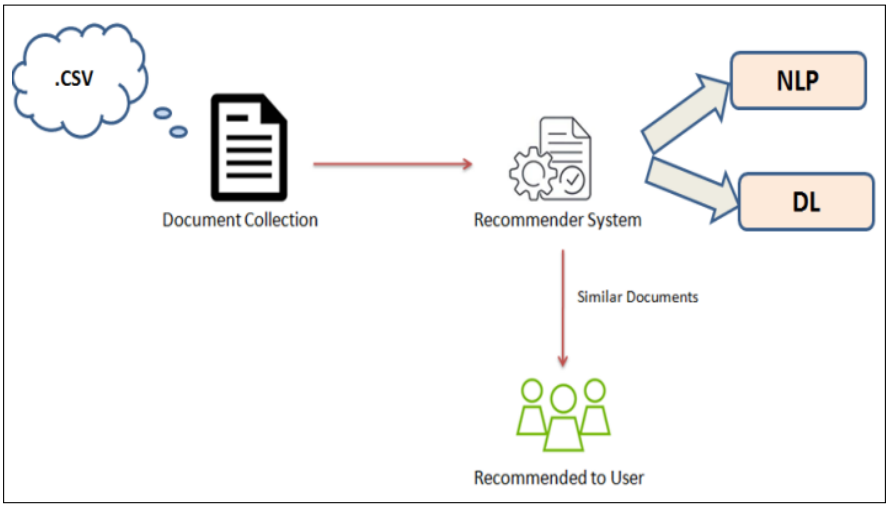


Figure 4.1: Proposed System (C-RPR) Architecture

Figure 4.1 show the proposed system architecture. The proposed system uses CBF techniques where the recommendation is given by considering similar characteristics of data. Document categorization is the most important topic in computer science publications [1], where the dataset consists of various venues (journals/conferences) where research papers are submitted. Here, documents are sequences of words, or these are the sentences.

In this architecture, NLP (Natural Language Processing) techniques are used for document processing to recommend similar research papers and a DL (Deep Learning) method is used to classify the venue by training the model as shown in Fig. 4.1.

The C-RPR Model consists of two modules: the feature vector similarity module and the Bi-LSTM (Bidirectional Long Short Term Memory) module. Feature vector similarity is generated using TF-IDF (Term Frequency Inverse Document Frequency) and calculates the cosine angle between those vectors. The model is trained using Bi-LSTM and GRU, which also predicts the ideal venue. First, each document is subjected to document preprocessing such as tokenization, stopword elimination, and stemming. Secondly, feature vectors FV are created, and finally, these feature vectors are used to calculate similarity and to train the model. Bi-LSTM and GRU is an extended recurrent neural network and is trained for recommending journals as it remembers long-term dependencies. This section introduces details of both modules.

4.2.1 Feature Vector Similarity Module

This module consists of the TF-IDF technique in natural language processing to deal with textual data and convert it into vectors by multiplying the term frequency and inverse document frequency, which generates feature vector space. These feature vectors are given as input to the cosine function to calculate the angle between two vectors.

TF-IDF:

TF-IDF is natural language processing techniques mostly used to handle textual data. It gives numerical weightage to words and recognizes the most important words from the corpus [1, 16]. The multiplication of TF and IDF generates the TF-IDF score for a certain word. TF is the number of times a word w occurs in a document d as shown in Eq. 4.1. The IDF measures the weight of rare words as shown in Eq. 4.2. The words that occur rarely in the document have a high IDF score.

$$tf = T/L \tag{4.1}$$

Where, the total number of terms in a document is L, and T is the number of times the term t appears in the document.

$$idf = log\frac{D}{D_i + 1} \tag{4.2}$$

D stands for the overall number of documents and for the overall number of documents that include the term t.

Then,

$$tf = tf \times idf \tag{4.3}$$

By multiplying the term frequency and inverse document frequency scores for a phrase, the TF-IDF score is determined shown in Eq. 4.3. Strong relationships to the document are shown by words with high TF-IDF scores [18, 19].

COSINE SIMILARITY:

A similarity metric called cosine similarity is frequently employed in text analysis to assess document similarity. By computing the cosine angle between two papers, one can estimate how similar two documents are, as illustrated in Fig 4.2, following data preprocessing and the determination of the TF-IDF score.

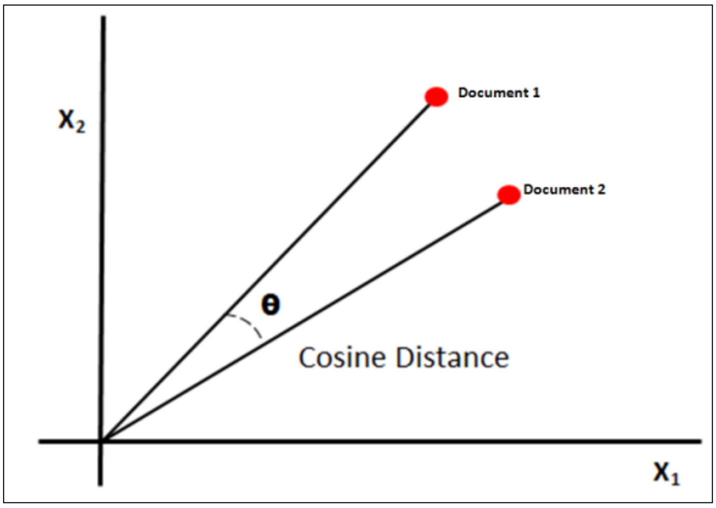


Figure 4.2: Cosine Similarity

The cosine angle is measured using the following formula shown in Eq. (4.4).

$$\text{Cosine Similarity} = cos(\theta) = \frac{V_q.V_d}{||V_q||||V_d||} \tag{4.4}$$

V_q and V_d are two documents represented in the form of vectors. V_q and V_d

are vectors corresponding to the query and document. The two documents are exactly similar when the cosine angle value is 1. If the angle is 0, then it shows the dissimilarity of documents. And, the intermediate values show intermediate similarity [18, 19].

4.2.2 Bi-LSTM:

The proposed model is based on a recurrent neural network model to deal with sequential data in textual format. As shown in Figure 4.3, Bi-LSTM is a bidirectional LSTM that comes under a recurrent neural network and is an extension of the traditional LSTM which processes sequential data consisting of two LSTMs. One takes input X at timestep T in a forward direction and the other in a backward direction at every time step to store information from the past and future [23].

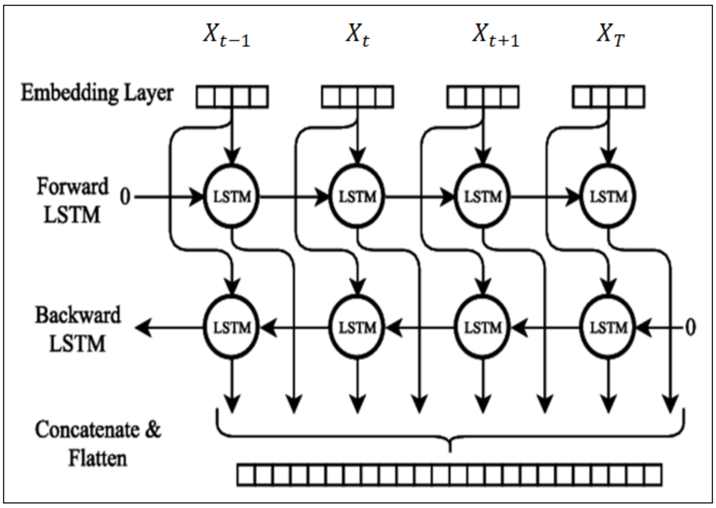


Figure 4.3: Bi-LSTM Model

In the LSTM architecture, there are three gates incorporated to memorise long-term dependencies. Though the LSTM models provide a strong prediction, they need to improve the forecast further. To achieve this, the Bi-LSTM model is introduced in this paper to offer additional training. It traverses the input data in both directions, forward and backward. It has been found that Bi-LSTM models for document categorization offer better predictions than conventional LSTM models. Document categorization is the most important topic in computer science publications [1], where the dataset has 20 venues (journals/conferences) such as IEEE, ACM, Springer, and SIAM as classes where research papers are submitted. Here, documents are sequences of words, or these are the sentences. This paper includes first-document preprocessing, feature vector construction using feature weighting methods, and prediction of venues by training a bi-LSTM neural network using abstract.

4.3 Gated Recurrent Unit (GRU)

The vanishing exploding gradients problem is effectively resolved using GRU [33]. The working of GRU and LSTM is similar but there is a difference in gate structure. Here vector form of the input sentence given to the GRU first layer. The first layer of the GRU encoder is the update gate layer. The update gate performs concatenation between the current input and previous output. This can be done using the following equation.

$$Z_i = (W^Z \times x_t + U^Z \times h_{t-1}) \tag{4.5}$$

Here x_t is current input given to our proposed model GRU encoder with respect to time t. The previous output with respect to time $t - 1$ is $ht - 1$. In the update, gate multiplication performs with respect to its corresponding weights. Here the weights are W^Z and U^Z. After performing concatenation the output of the equation is a pass to the activation function, which gives the output as Z_t. The update entryway encourages the proposed model to decide the amount of the past data regarding now is the ideal time should be passed along to what's to come. With the help of the update gate the proposed model eliminates the imperil of vanishing gradient problem. The output of the update gate is a pass to the reset gate. This gate is utilized to determine which information should be removed from the previous in the proposed model [33]. To calculate the output of the reset gate following equation is used.

$$r_t = (W^r \times x_t + U^r \times h_{t-1}) \tag{4.6}$$

The above equation is the same as the update gate equation. Here weights are W^r and U^r is multiplied with respect to x_t and $ht - 1$. This gate also removes unnecessary data from the proposed GRU encoder model. Here we need to store relevant information from the past using the below equation in GRU encoder with respect to its current content. In the above equation the output of the reset gate used as input and then in the proposed model tangent activation function applied [33].

$$h_t = tanh(W \times x_t + r_t \times U \times h_{t-1}) \tag{4.7}$$

$$h_t = Z_t \times h_{t-1} + (1 - Z_t) \times h_t \tag{4.8}$$

Finally a single vector is generated, which is the output of the proposed GRU encoder. The output of this equation depends on all the previous gate outputs. Then GRU decoder generates proper response as per input given. In the proposed model update entryway and reset door is utilized to choose what data ought to be passed to the yield. The GRU encoder-decoder model has the ability to keep all the data as long as feasible.

4.4 Natural Language Processing (NLP)

Text must be preprocessed before a model is trained for almost all Natural Language Processing (NLP) tasks. It is up to us researchers to sanities the text since deep learning models cannot use raw text directly. The preprocessing techniques can vary depending on the nature of the work. The most popular preprocessing method that may be used with many NLP jobs using NLTK. For use with NLP in Python, there is a toolkit called NLTK. It offers us a variety of text processing libraries and a large number of test datasets. Using NLTK, a range of operations can be performed, including tokenization, lower-case conversion, stop words removal, stemming, lemmatization, parse tree or syntax tree generation, and POS tagging.

4.5 Training Proposed Model

The main goal of the proposed model in the training state is to reduce the difference between actual output and predicted output generated by this model. During the training of the proposed model, the predicted answer is given to the next time step, this makes training process slow. To speed up this, proposed model gives the actual output sentence to the decoder. At each time step the decoder produces yield utilizing a lot of vocabulary. Here the vocabulary size in this proposed model is 5000.

4.6 Closure

In this chapter proposed methodology explained. The architecture and mathematical structure of proposed model described briefly in this chapter.

Chapter 5

Experimental Environment

5.1 Introduction

This chapter discusses experimental environment such as software requirements for proposed Recommendation system, dataset and some hyper-parameter used to enhance performance of the proposed system. This chapter gives information about experimental results and analysis. There are two recommendations provided by the proposed system: 1) Recommendation of similar papers based on the title of the paper or a keyword as a query using TF-IDF and cosine metric and 2) Recommendation of a three-class (Top3) journal/conference to submit the researcher's paper based on the abstract of the paper as input. This paper consists of 20 venues and is classified using the Bi-LSTM and GRU model.

5.2 Experimental Environment

1. Operating System: For proposed Recommendation system Windows operating system is used.
2. Programming Language: Python programming language is used to implement proposed Recommendation system. This language contains some build-in packages for machine learning. For implementation purpose python 3.5 version is used.
3. Tensorflow: It is open source platform for AI. With the help of tensorflow we can easily build and deploy any machine learning model. For implementation of the proposed Recommendation system we use tensorflow 1.0.0 version.

4. Google Colabaratory: For faster training the proposed system requires GPU. The Google Colaboratory provides free GPU services for limited period of time.

5.2.1 Dataset

The papers from different publications are considered from the DBLP dataset along with different features such as id, title, author, abstract, venue, and link of paper. The dataset has 9348 records. 90% of the data is used to train the model, and the remaining 10% is used for testing. There are 20 venues across four digital libraries such as IEEE, SIAM, Springer, and ACM that are considered labels for prediction. With the help of these labels, prediction of venues is made where researchers can submit their paper by giving the abstract as input.

5.2.2 Parameter Details

This research work explored bi-directional LSTM and GRU. A neural network mechanism has been applied to improve performance. Here 32/64 numbers of samples used before model getting updated with respect to hidden layers which contains 128 vectors. In the proposed model we want global minima so we set the learning rate as 0.001 and we are using 2 hidden layers. The word embedding is nothing but converting text into numbers so we are using embedding size as 64.

Hyper-parameter Name	**Value**
Vocabulory size	660
Batch size	16
Number of Epochs	10
Size of Layer	128
Number of Layers	2
Embedded size	64
Learning size	0.001
Encoder Type	Bidirectional

Table 5.1: Training Hyper-parameters

5.3 Experimental Work

5.3.1 Data Preprocessing

Due to the design of the proposed model the users input cannot direct fed to the encoder. There are some data preprocessing step required to prepare dataset and transform the sequential data into numerical form is known as word embedding as show in figure 5.1.

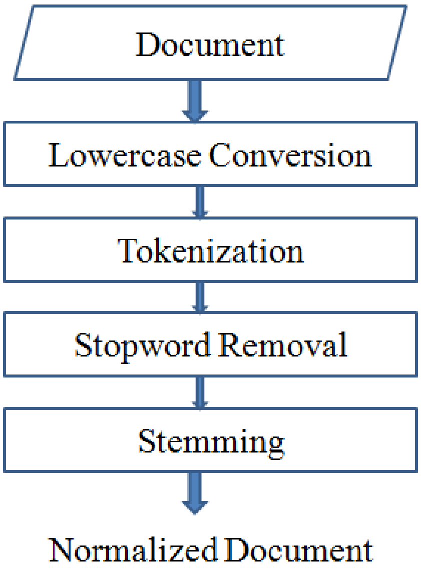


Figure 5.1: Data Preprocessing

Following are some data preprocessing steps used in the proposed Recommendation system.

1. Tokenization: It is important part of the data preprocessing. Length of the input and the output sentence are differ thus tokens are added.
2. Text vocabulary: In proposed model most frequent 4882 words in training dataset are kept as vocabulary size.
3. Remove word which is less frequent.

5.3.2 Algorithm

Algorithm (Bi-LSTM Model)

Step 1: Start

Step 2: Documents in category i

Step 3: Text Preprocessing - Stopword Removal, Tokenization, pad_sequencing

Step 4: Normalized Document

Step 5: Build a Neural Network - Bi-directional LSTM

Step 6: Setting hyper parameters - vocabulary_size, embedding_dim, max_length, layers, activation, dropout, optimizer.

Step 7: Calculate Accuracy

Step 8: Top k venues

Step 9: end

5.4 Experimental Results

This section discusses the response generated by the Recommendation system using LSTM and GRU. This section also summarizes all experimental results and also discusses comparison between LSTM and GRU.

During experimentation, activation functions such as softmax, relu, LeakyReLU, and tanh were tested with different optimizers such as Adam, SGD, adagrad, and RMSprop. Table 5.2 and Table 5.3 show performance for batch sizes of 32 and 64 with different activation functions and optimizers consisting of 10 epochs with vocabulary_size = 5000, embedding_dimensions = 64, max_length = 200, dropout = 0.2, and loss = sparse_categorical_crossentropy.

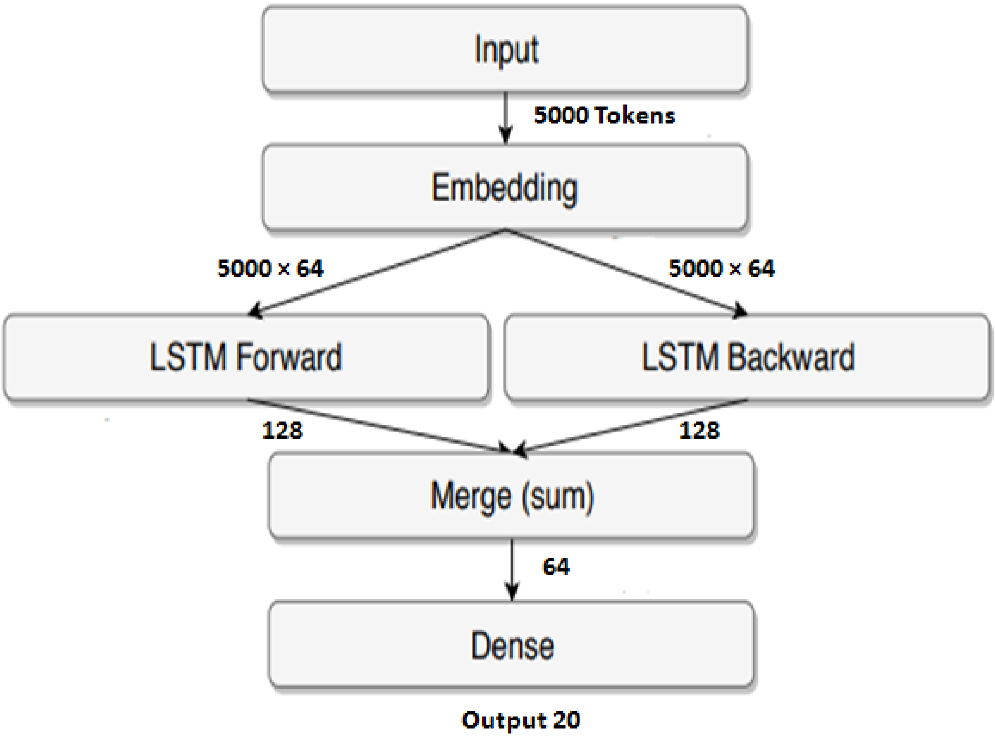


Figure 5.2: Bi-LSTM architecture in proposed system

The proposed Bi-LSTM architecture is built with three layers as shown in Fig. 5.2, where 5000 tokens are passed as input. The embedding layer is an input layer. Each token is transformed by this layer into an array of 64 dimensions. The Bi-LSTM contains two hidden layers, forward and backward, with 128 memory units. Relu is used as an activation function in hidden layers. The merging of two hidden LSTMs generates output. Finally, the dense layer uses a softmax activation function with 20 output neurons.

Table 5.2 and Table 5.3 show the performance improvement using five Bi-LSTM models with different activation functions and optimizers. The results are analyzed for batch sizes of 32 and 64. During experimentation, it is observed that softmax is only the activation function that performs better at the output layer as compared to others, as this problem is a multiclassification problem. Relu is performing well in the hidden layers. At the output layer, Adam gives better results as compared to others. The accuracy of all five models is calculated using accuracy metric to evaluate the recommender system, which is defined in Eqs. (5.1).

$$Accuracy = f(x) = \frac{\sum_{i=1}^{N} |P_i \cap G_i|}{\sum_{i=1}^{N} |G_i|} \tag{5.1}$$

Where Pi is predicted samples in ith category and Gi is the labeled samples as ith category.

By observing the Table 5.2, the accuracy is 86.05% for training dataset and 74.55% is for testing dataset with batch size 32. In Table 5.3, the accuracy is 79.48% for training dataset and 69.72% is for testing dataset with batch size 64.

Table 5.2: Accuracy of Five Bi-LSTM models on batch size 32

Batch Size 32					
Hyperparameters	**Bi-LSTM Architecture 1**	**Bi-LSTM Architecture 2**	**Bi-LSTM Architecture 3**	**Bi-LSTM Architecture 4**	**Bi-LSTM Architecture 5**
Activation function (Hidden Layer)	Relu	LeakyReLU	LeakyReLU	Tanh	Relu
Activation function (Output Layer)	Softmax	Softmax	Softmax	Softmax	Softmax
Optimizer	RMSprop	RMSprop	Adam	Adam	Adam
Training accuracy (%)	68.56	73.30	82..20	80.61	**86.05**
Testing accuracy (%)	63.61	64.89	73.03	73.28	**74.55**

Table 5.3: Accuracy of Five Bi-LSTM models on batch size 64

Batch Size 64					
Hyperparameters	**Bi-LSTM Architecture 1**	**Bi-LSTM Architecture 2**	**Bi-LSTM Architecture 3**	**Bi-LSTM Architecture 4**	**Bi-LSTM Architecture 5**
Activation function (Hidden Layer)	Relu	LeakyReLU	LeakyReLU	Tanh	Relu
Activation function (Output Layer)	Softmax	Softmax	Softmax	Softmax	Softmax
Optimizer	RMSprop	RMSprop	Adam	Adam	Adam
Training accuracy (%)	52.70	56.69	57.68	63.69	**79.48**
Testing accuracy (%)	37.40	49.36	49.62	58.27	**69.72**

5.4.1 Accuracy Comparison on Various Epoch (Bi-LSTM)

Figure 5.3 and Figure 5.4 shows the graphical results of accuracy comparison on various epoch for LSTM. As epoch increases, loss of LSTM decreases. As epoch increases, accuracy of LSTM increases.

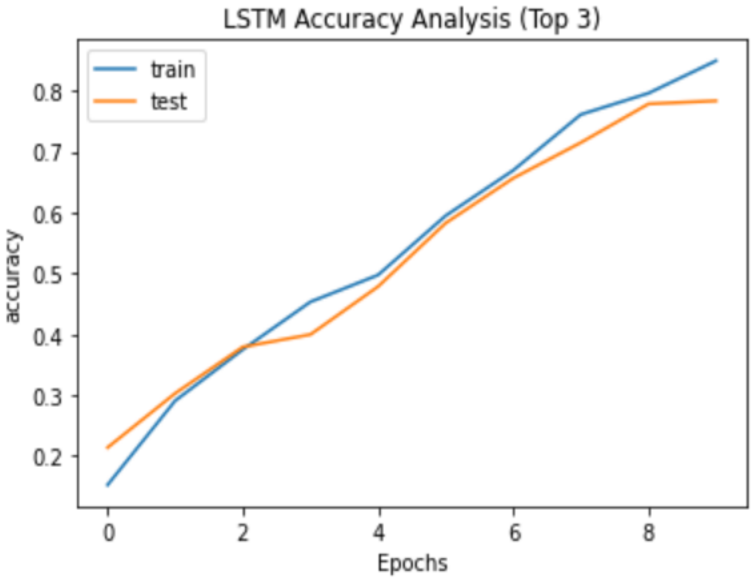


Figure 5.3: Bi-LSTM Accuracy Analysis (Top 3) for batch size=32

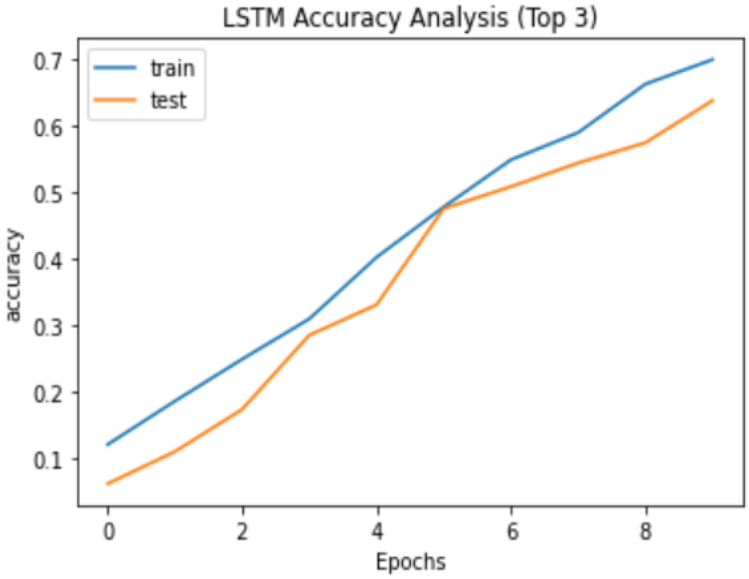


Figure 5.4: Bi-LSTM Accuracy Analysis (Top 3) for batch size=64

It has been shown that, with the use of machine learning algorithms and feature selection techniques, forecasting the venues for a given abstract is ineffective [1]. The proposed system makes use of a Bi-LSTM to provide top-3 venue recommendations based on abstracts as input with better accuracy as compared to machine learning models. The respective accuracy graph is shown in figure 5.3 and figure 5.4.

5.4.2 Accuracy Comparison on Various Epoch (GRU)

By observing the Table 5.4, the accuracy is 85.91% for training dataset and 78.37% is for testing dataset with batch size 32. In Table 5.5, the accuracy is 86.15% for training dataset and 75.83% is for testing dataset with batch size 64.

Figure 5.5 and Figure 5.6 shows the graphical results of accuracy comparison on various epoch for GRU. As epoch increases, loss of GRU decreases. As epoch increases, accuracy of GRU increases.

The proposed system makes use of a GRU to provide top-3 venue recommendations based on abstracts as input with little bit improved accuracy in Bi-LSTM and as compared to machine learning models. The respective accuracy graph is shown in Figure 5.5 and Figure 5.6.

Table 5.4: Accuracy of Five GRU models on batch size 32

Batch Size 32					
Hyperparameters	**GRU Architecture 1**	**GRU Architecture 2**	**GRU Architecture 3**	**GRU Architecture 4**	**GRU Architecture 5**
Activation function (Hidden Layer)	Relu	LeakyReLU	LeakyReLU	Tanh	Relu
Activation function (Output Layer)	Softmax	Softmax	Softmax	Softmax	Softmax
Optimizer	RMSprop	RMSprop	Adam	Adam	Adam
Training accuracy (%)	69.48	67.86	85.38	**90.14**	79.48
Testing accuracy (%)	57.51	57.25	75.57	**82.70**	69.72

Table 5.5: Accuracy of Five GRU models on batch size 64

Batch Size 64					
Hyperparameters	**GRU Architecture 1**	**GRU Architecture 2**	**GRU Architecture 3**	**GRU Architecture 4**	**GRU Architecture 5**
Activation function (Hidden Layer)	Relu	LeakyReLU	LeakyReLU	Tanh	Relu
Activation function (Output Layer)	Softmax	Softmax	Softmax	Softmax	Softmax
Optimizer	RMSprop	RMSprop	Adam	Adam	Adam
Training accuracy (%)	58.53	59.17	74.789	**86.15**	75.94
Testing accuracy (%)	44.78	51.91	62.85	**75.83**	63.61

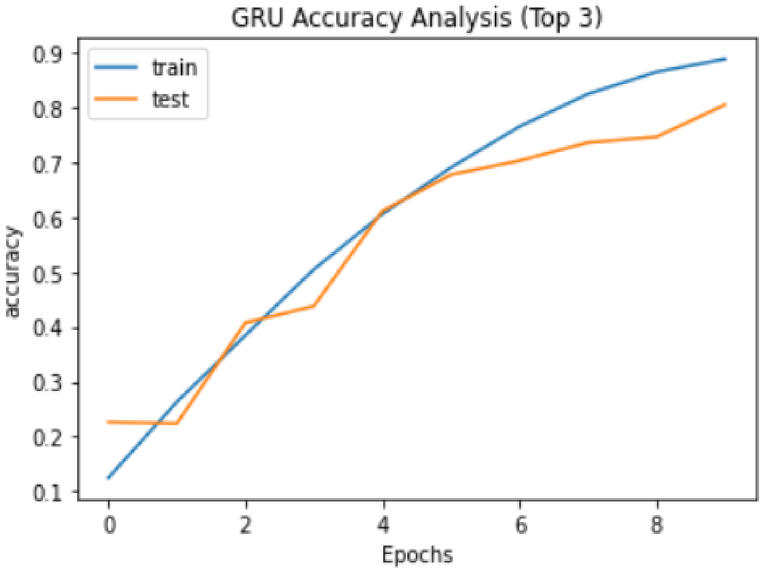


Figure 5.5: GRU Accuracy Analysis (Top 3) for batch size=32

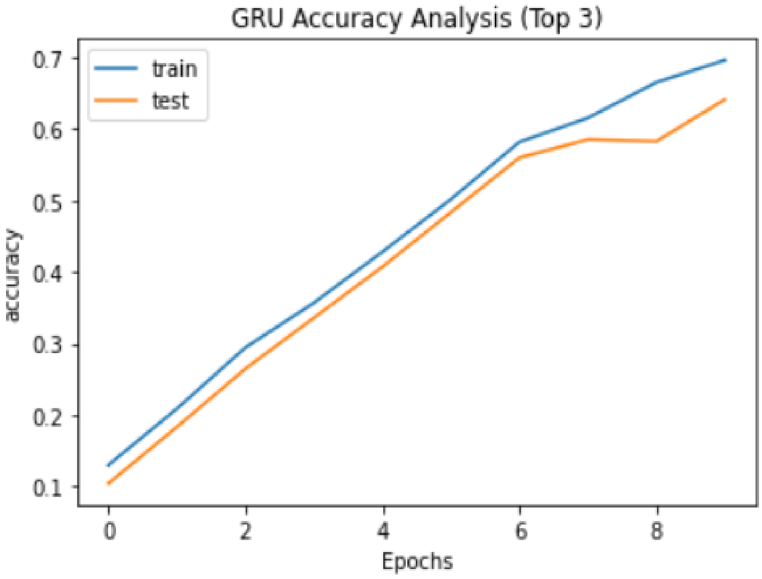


Figure 5.6: GRU Accuracy Analysis (Top 3) for batch size=64

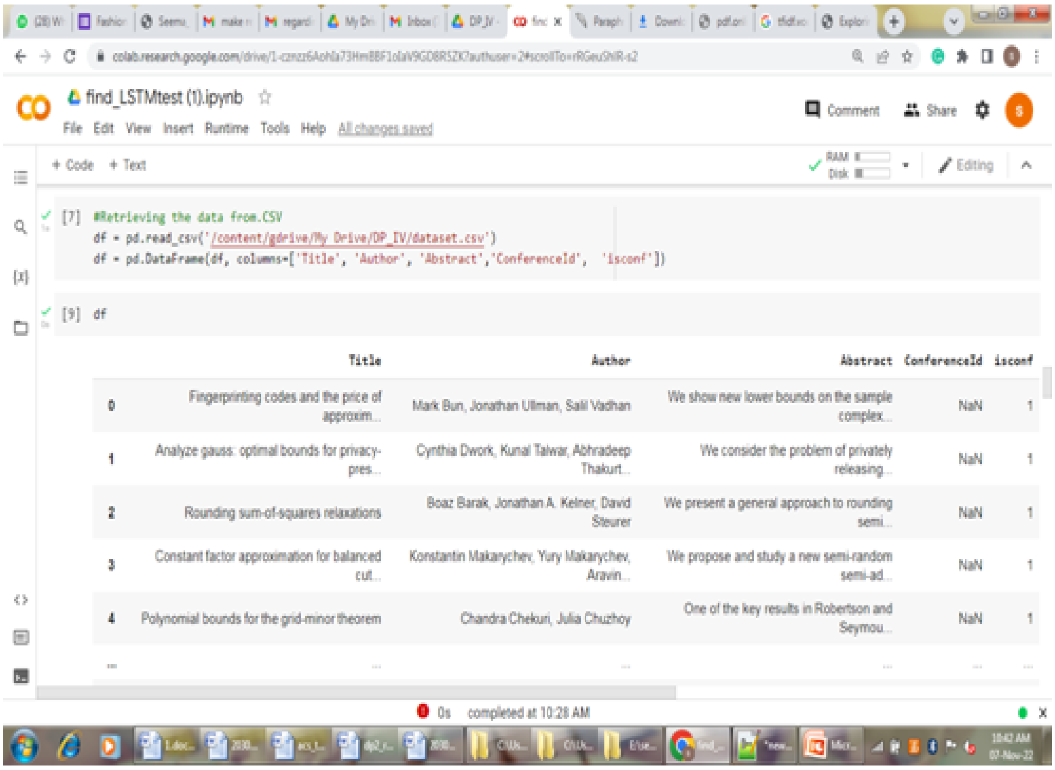
It is observed that the accuracy is 88.73% for training dataset and 80.41% is for testing dataset with batch size 32. The accuracy is 69.62% for training dataset and 64.12% is for testing dataset with batch size 64.

5.5 Generating Recommendations

Used TFIDF vectorizer and cosine similarity to preselect the most similar titles. It generates recommendations for a query using title as input and generates similar titles and journals.

Step 1 : Dataset

The dataset is downloaded from Kaggle and has a total of 3225 records and 20 journal and conference publications across 5 digital libraries (IEEE, ACM, SIAM, Springer, AAI). Total of 8 features are included in the dataset such as ID, Title, Author, Abstract, ConferenceID, Link, Isconf, and venue as shown in Figure 5.7.



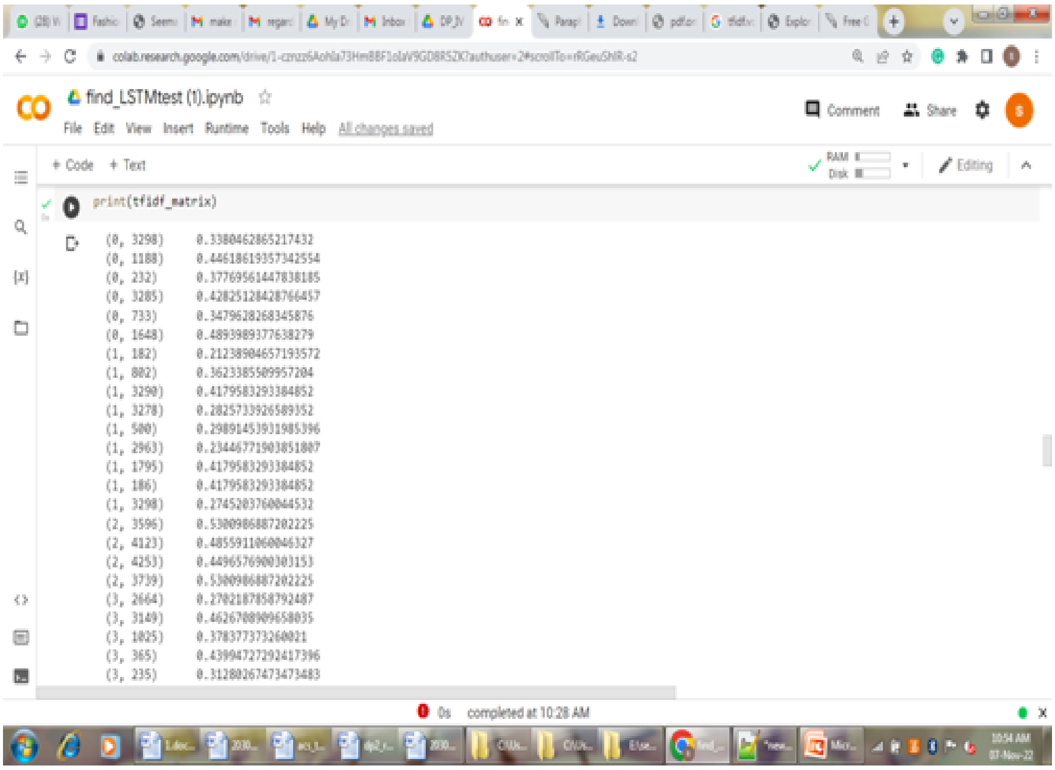
```
[7] #Retrieving the data from.CSV
    df = pd.read_csv('/content/gdrive/My Drive/DP_IV/dataset.csv')
    df = pd.DataFrame(df, columns=['Title', 'Author', 'Abstract','ConferenceId',  'isconf'])

[9] df
```

	Title	Author	Abstract	ConferenceId	isconf
0	Fingerprinting codes and the price of approxim...	Mark Bun, Jonathan Ullman, Salil Vadhan	We show new lower bounds on the sample complex...	NaN	1
1	Analyze gauss: optimal bounds for privacy-pres...	Cynthia Dwork, Kunal Talwar, Abhradeep Thakurt...	We consider the problem of privately releasing...	NaN	1
2	Rounding sum-of-squares relaxations	Boaz Barak, Jonathan A. Kelner, David Steurer	We present a general approach to rounding semi...	NaN	1
3	Constant factor approximation for balanced cut...	Konstantin Makarychev, Yury Makarychev, Aravin...	We propose and study a new semi-random semi-ad...	NaN	1
4	Polynomial bounds for the grid-minor theorem	Chandra Chekuri, Julia Chuzhoy	One of the key results in Robertson and Seymou...	NaN	1
...	...	...	...	...	...

Figure 5.7: Dataset

Step 2 : TFIDF Vocabulary

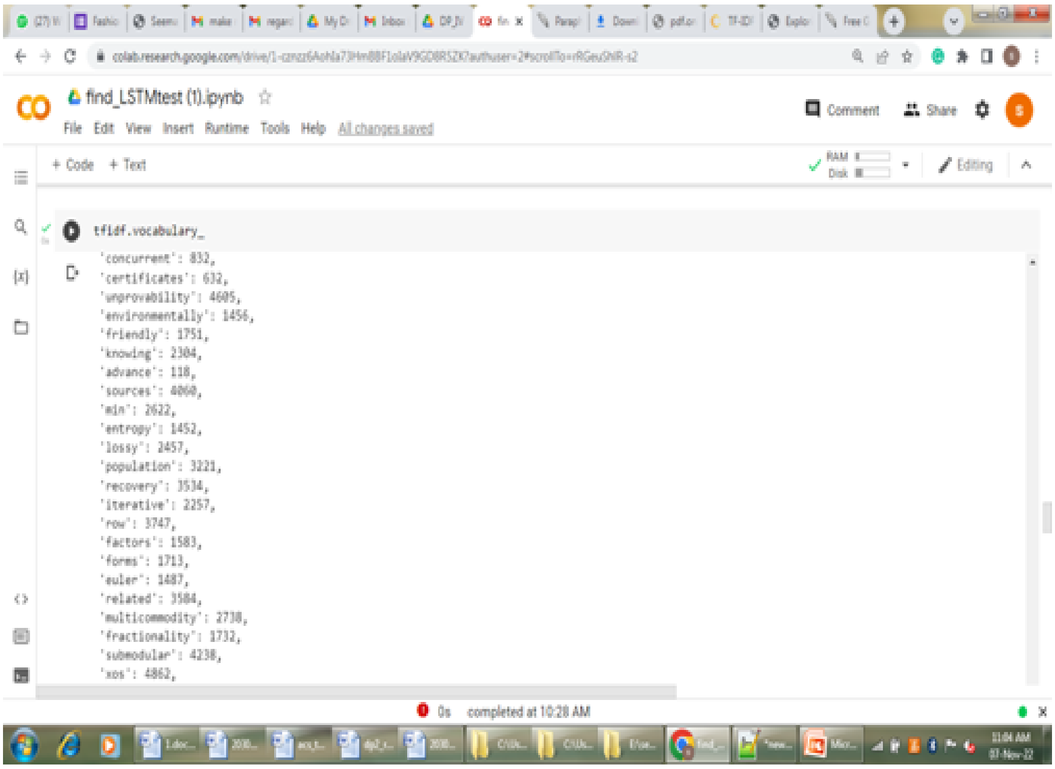
The TF-IDF Vectorizer object is defined and all English stop words such as "the" and "a" is removed. Finally, we constructed the required TF-IDF matrix as shown in Fig. 5.8 by fitting and transforming the data to tell which word is more significant. In Fig. 5.9, words are converted to numbers.



```
print(tfidf_matrix)
```

```
  (0, 3298)	0.3380462865217432
  (0, 1188)	0.44618619357342554
  (0, 232)	0.37769561447838185
  (0, 3285)	0.42825128428766457
  (0, 733)	0.3479628268345876
  (0, 1648)	0.4893989377638279
  (1, 182)	0.21238904657193572
  (1, 802)	0.3623385509957204
  (1, 3290)	0.4179583293384852
  (1, 3278)	0.2825733926589352
  (1, 500)	0.29891453931985396
  (1, 2963)	0.23446771903851807
  (1, 1795)	0.4179583293384852
  (1, 186)	0.4179583293384852
  (1, 3298)	0.2745203760044532
  (2, 3596)	0.5300986887202225
  (2, 4123)	0.4855911060046327
  (2, 4253)	0.4496576900303153
  (2, 3739)	0.5300986887202225
  (3, 2664)	0.2702187858792487
  (3, 3149)	0.4626708909658035
  (3, 1025)	0.378377373260021
  (3, 365)	0.43994727292417396
  (3, 235)	0.31280267473473483
```

Figure 5.8: TFIDF Matrix

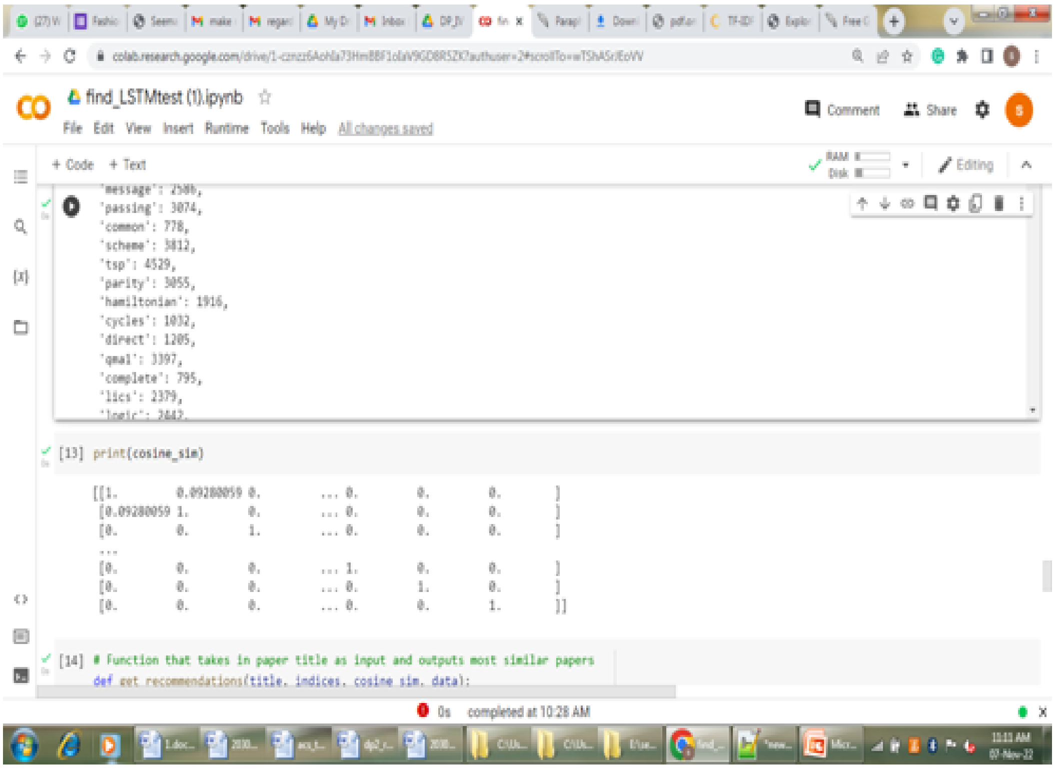


```
tfidf.vocabulary_
```

```
 'concurrent': 832,
 'certificates': 632,
 'unprovability': 4605,
 'environmentally': 1456,
 'friendly': 1751,
 'knowing': 2304,
 'advance': 118,
 'sources': 4060,
 'min': 2622,
 'entropy': 1452,
 'lossy': 2457,
 'population': 3221,
 'recovery': 3534,
 'iterative': 2257,
 'row': 3747,
 'factors': 1583,
 'forms': 1713,
 'euler': 1487,
 'related': 3584,
 'multicommodity': 2738,
 'fractionality': 1732,
 'submodular': 4238,
 'xos': 4862,
```

Figure 5.9: TFIDF Vocabulary

Step 3 : Cosine Similarity Matrix



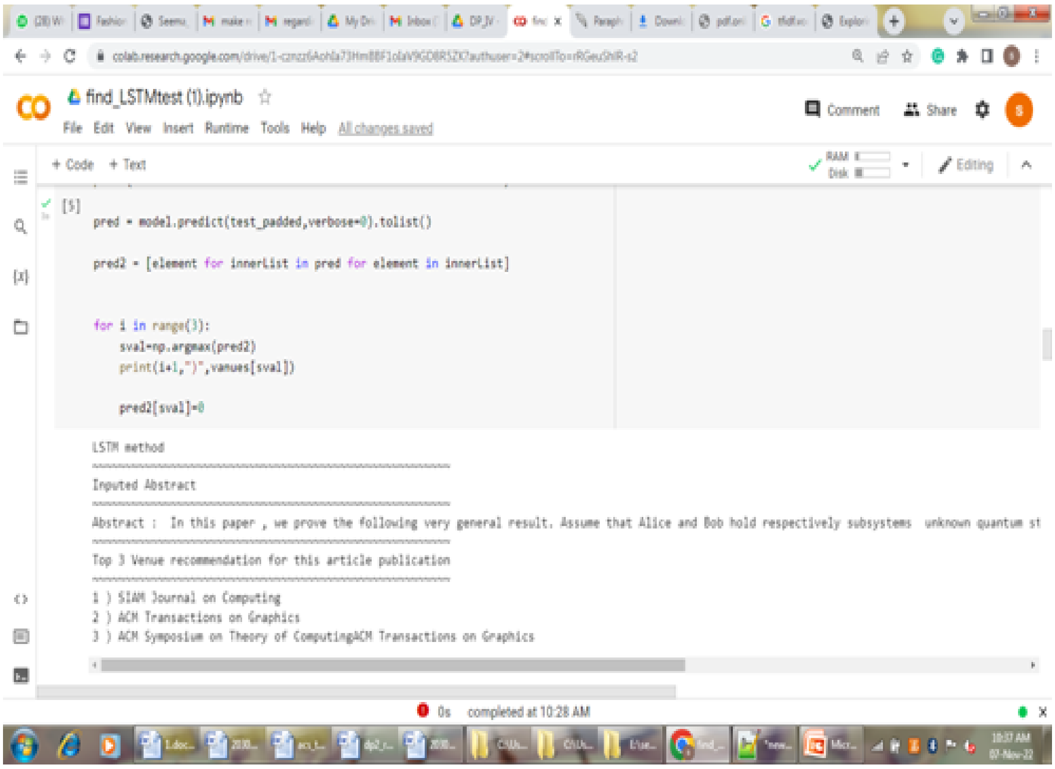
```
 'message': 2586,
 'passing': 3074,
 'common': 778,
 'scheme': 3812,
 'tsp': 4529,
 'parity': 3055,
 'hamiltonian': 1916,
 'cycles': 1032,
 'direct': 1205,
 'qmal': 3397,
 'complete': 795,
 'lics': 2379,
 'logic': 2442,
```

```
[13] print(cosine_sim)

[[1.         0.09280059 0.         ... 0.         0.         0.        ]
 [0.09280059 1.         0.         ... 0.         0.         0.        ]
 [0.         0.         1.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 1.         0.         0.        ]
 [0.         0.         0.         ... 0.         1.         0.        ]
 [0.         0.         0.         ... 0.         0.         1.        ]]
```

```
[14] # Function that takes in paper title as input and outputs most similar papers
     def get_recommendations(title, indices, cosine_sim, data):
```

Figure 5.10: Cosine Similarity Matrix

Step 4 : Top 3 Venues

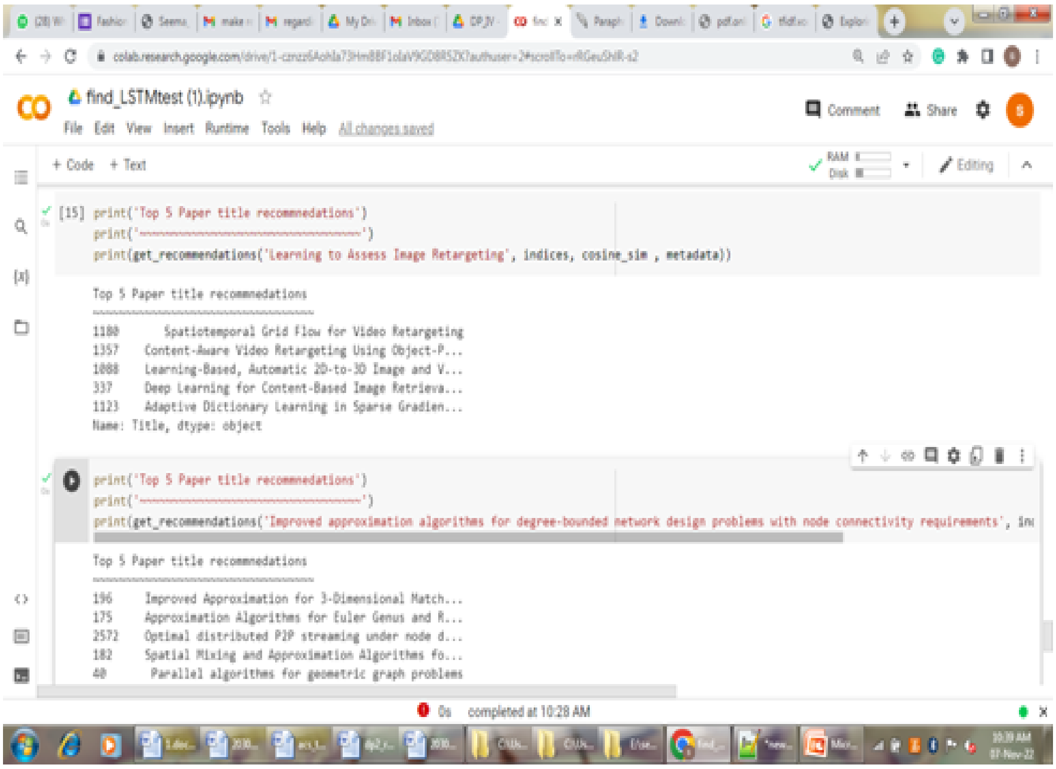


```
[5]
pred = model.predict(test_padded,verbose=0).tolist()

pred2 = [element for innerList in pred for element in innerList]


for i in range(3):
    sval=np.argmax(pred2)
    print(i+1,")",vanues[sval])

    pred2[sval]=0
```

```
LSTM method
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Inputed Abstract
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Abstract :  In this paper , we prove the following very general result. Assume that Alice and Bob hold respectively subsystems  unknown quantum st
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Top 3 Venue recommendation for this article publication
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 ) SIAM Journal on Computing
2 ) ACM Transactions on Graphics
3 ) ACM Symposium on Theory of ComputingACM Transactions on Graphics
```

Figure 5.11: Top-3 Venues for article publication

Step 5 : Top 5 Similar Papers



```
[15] print('Top 5 Paper title recommedations')
     print('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~')
     print(get_recommendations('Learning to Assess Image Retargeting', indices, cosine_sim , metadata))

Top 5 Paper title recommedations
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1180      Spatiotemporal Grid Flow for Video Retargeting
1357    Content-Aware Video Retargeting Using Object-P...
1088    Learning-Based, Automatic 2D-to-3D Image and V...
337     Deep Learning for Content-Based Image Retrieva...
1123    Adaptive Dictionary Learning in Sparse Gradien...
Name: Title, dtype: object
```

```
print('Top 5 Paper title recommedations')
print('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~')
print(get_recommendations('Improved approximation algorithms for degree-bounded network design problems with node connectivity requirements', in

Top 5 Paper title recommedations
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
196     Improved Approximation for 3-Dimensional Match...
175     Approximation Algorithms for Euler Genus and R...
2572    Optimal distributed P2P streaming under node d...
182     Spatial Mixing and Approximation Algorithms fo...
40       Parallel algorithms for geometric graph problems
```

Figure 5.12: Top-3 Venues for article publication

Chapter 6

Conclusion

The suggested system is based on a content-based recommendation system that was implemented on the DBLP dataset, which consists of 20 venues from publishers including IEEE, Springer, and ACM. It is constructed on a deep memory neural network called Bi-LSTM. The TF-IDF approach is used to first preprocess and vectorize the dataset's textual data. Using the cosine metric, these vectorized values are used to assess how similar two publications are. The Bi-LSTM suggests the top three places for researchers to submit their work to the journals and conferences that are the most pertinent to their work.

The proposed model produced better results with 74.55% accuracy using Bi-LSTM and 78.37% accuracy using GRU mechanism for batch size 32. For batch size 64, the proposed model produced better results with 69.72% accuracy using Bi-LSTM and 75.83% accuracy using GRU mechanism. We believe that by leveraging GRU with attention mechanisms and by using word embedding techniques such as word2vec, doc2vec, and BERT, it can be increased much more further, which will be the focus of our future study.

REFERENCES

[1] [1] Wang, D., Liang, Y., Xu, D., Feng, X., & Guan, R. (2018). A content-based recommender system for computer science publications. Knowledge-Based Systems, 157, 1-9. doi:10.1016/j.knosys.2018.05.001

[2] Singh, M., Chakraborty, T., Mukherjee, A., & Goyal, P. (2015, June). ConfAssist: A Conflict resolution framework for assisting the categorization of Computer Science conferences. In Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries (pp. 257-258). doi:10.1145/2756406.2756963

[3] Hassan, H. A. M. (2017, July). Personalized research paper recommendation using deep learning. In Proceedings of the 25th conference on user modeling, adaptation and personalization (pp. 327-330). doi:10.1145/3079628.3079708

[4] 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), e0158423.
doi:10.1371/journal.pone.0158423.

[5] Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., & Xia, F. (2019). Scientific Paper Recommendation: A Survey. IEEE Access, (pp. 9324-9339). doi: 10.1109/ACCESS.2018.2890388

[6] Kumar, P., & Thakur, R. S. (2018). Recommendation system techniques and related issues: a survey. International Journal of Information Technology,10(4), 495-501. doi:10.1007/S41870-018-0138-8

[7] Beel, J., Langer, S., Genzmehr, M., Gipp, B., Breitinger, C., & Nürnberger, A. (2013, October). Research paper recommender system evaluation: a quantitative literature survey. In Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation (pp. 15-22). doi:10.1145/2532508.2532512

[8] Jelodar, H., Wang, Y., Xiao, G., Rabbani, M., Zhao, R., Ayobi, & Masood, I. (2021). Recommendation system based on semantic scholar mining and topic modeling on conference publications. Soft Computing, 25(5), 3675-3696. doi:10.1007/s00500-020-05397-3

[9] Asim, M., & Khusro, S. (2018). Content Based Call for Papers Recommendation to Researchers. (2018)12^{th} International Conference on Open Source Systems and Technologies (ICOSST), (pp.42-47). doi:10.1109/ICOSST.2018.8632174

[10] Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., & Xia, F. (2019). Scientific Paper Recommendation: A Survey. IEEE Access, (pp.9324-9339). doi:10.1109/ACCESS.2018.2890388

[11] Kumar, P., Thakur, R. S. (2018). Recommendation system techniques and related issues: a survey. International Journal of Information Technology, 10(4), 495-501. doi:10.1007/S41870-018-0138-8

[12] Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., & Reiterer, S. (2013). Toward the next generation of recommender systems: applications and research challenges. Multimedia services in intelligent environments, 81-98. doi:10.1007/978-3-319-00372-6_5

[13] Sun, J., Ma, J., Liu, Z., & Miao, Y. (2014). Leveraging content and connections for scientific article recommendation in social computing contexts. The Computer Journal, 57(9), 1331-1342

[14] Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS), 22(1), 143-177. doi:10.1145/963770.963776.

[15] Moghaddam, M. G., Mustapha, N., Mustapha, A., Sharef, N. M., & Elahian, A. (2014, May). AgeTrust: A new temporal trust-based collaborative filtering approach. In 2014 International Conference on Information Science & Applications (ICISA) (pp. 1-4). IEEE. doi:10.1109/ICISA.2014.6847352.

[16] Patra, B. G., Maroufy, V., Soltanalizadeh, B., Deng, N., Zheng, W. J., Roberts, K., & Wu, H. (2020). A content-based literature recommendation system for datasets to improve data reusability–a case study on gene expression omnibus (geo) datasets. Journal of Biomedical Informatics, 104, 103399. doi:10.1016/j.jbi.2020.103399

[17] Ebesu, T., & Fang, Y. (2017, August). Neural citation network for context-aware citation recommendation. In Proceedings of the 40th international ACM

SIGIR conference on research and development in information retrieval (pp. 1093-1096). doi:10.1145/3077136.3080730

[18] Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016, April). Cosine similarity to determine similarity measure: Study case in online essay assessment. In 2016 4th International Conference on Cyber and IT Service Management (pp. 1-6). IEEE. doi:10.1109/CITSM.2016.7577578

[19] Alodadi, M., & Janeja, V. P. (2015, October). Similarity in patient support forums using TF-IDF and cosine similarity metrics. In 2015 International Conference on Healthcare Informatics (pp. 521-522). IEEE. doi:10.1109/ICHI.2015.99.

[20] Wang, J., Zhu, L., Dai, T., & Wang, Y. (2020). Deep memory network with Bi-LSTM for personalized context-aware citation recommendation. Neurocomputing, 410, 103-113. doi:10.1016/j.neucom.2020.05.047.

[21] Braz, F. A., da Silva, N. C., de Campos, T. E., Chaves, F. B. S., Ferreira, M. H., Inazawa, P. H. & Peixoto, F. H. (2018). Document classification using a Bi-LSTM to unclog Brazil's supreme court. arXiv preprint arXiv:1811.11569.

[22] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735.

[23] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019, December). The performance of LSTM and BiLSTM in forecasting time series. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 3285-3292). IEEE. doi:10.1109/BigData47090.2019.9005997.

[24] Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., & Xia, F. (2019). Scientific paper recommendation: A survey. Ieee Access, 7, 9324-9339. doi:10.1109/ACCESS.2018.2890388.

[25] Philip, S., Shola, P., & Ovye, A. (2014). Application of content-based approach in research paper recommendation system for a digital library. International Journal of Advanced Computer Science and Applications, 5(10). doi:10.14569/IJACSA.2014.051006.

[26] E. Portilla Olvera and D. Godoy(2012). Evaluating Term Weighting Schemes for Content-based Tag Recommendation in Social Tagging Systems. IEEE

Latin America Transactions, 10. 1973-1980.
doi:10.1109/TLA.2012.6272482.

[27] Roy, P. K., Chowdhary, S. S., & Bhatia, R. (2020). A Machine Learning approach for automation of Resume Recommendation system. Procedia Computer Science, 167(2019), 2318–2327.
doi:10.1016/j.procs.2020.03.284.

[28] Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. Proceedings of the ACM International Conference on Digital Libraries, 195–204.
doi:10.1145/336597.336662.

[29] Meteren, R. Van, & Someren, M. Van. (2000). Using Content-Based Filtering for Recommendation. ECML/MLNET Workshop on Machine Learning and the New Information Age, 47–56.

[30] Abduljabbar, Rusul & Dia, Hussein & Tsai, Pei-Wei. (2021). Unidirectional and Bidirectional LSTM Models for Short-Term Traffic Prediction. Journal of Advanced Transportation. 2021.
1-16. 10.1155/2021/5589075.

[31] Huang, W., Wu, Z., Liang, C., Mitra, P., & Giles, C. L. (2015). A neural probabilistic model for context based citation recommendation. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI 2015 and the 27th Innovative Applications of Artificial Intelligence Conference, IAAI 2015 (pp. 2404-2410). doi:10.1609/aaai.v29i1.9528.

[32] F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, 16(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005.

[33] El-Amir, H., & Hamdy, M. (2019). A Tour Through the Deep Learning Pipeline.

[34] Afoudi, Y., Lazaar, M., & Al Achhab, M. (2021). Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. Simulation Modelling Practice and Theory, 113(6), 102375. https://doi.org/10.1016/j.simpat.2021.102375.

[35] Sang, A., & Vishwakarma, S. K. (2018). A ranking based recommender system for cold start & data sparsity problem. 2017 10th International Conference on Contemporary Computing, 1–3. https://doi.org/10.1109/IC3.2017.8284347.

LIST OF PUBLICATIONS ON PRESENT WORK

[1] Sandeep A. Thorat, Ashwini Gavade, Seema Mane and Sourabh Bakshi “Research Challenges, Opportunities and Applications in Collaborative Filtering and Content-based Recommendation System”, National Conference on Advances in Science, Engineering and Technology for Sustainable Development (NCASETSD - 2022), (Status: Best Paper Award and extended to IEEE)

[2] Ashwini Patil, Seema Mane, “Content-based Research Paper Recommendation (C-RPR) System using Bi-LSTM Neural Network”, Mapana Journal of Sciences. (Status: Under Review)

Keywords: Recommendation System, Content-based, Collaborative Filtering, Research Challenges, Applications

Sandeep A. THORAT[0000-0002-1641-0376]*,

Ashwini GAVADE[0000-0003-4104-845X]*,

Seema MANE[0000-0001-8377-974X]*,

Sourabh BAKSHI[0000-0001-7245-9230]*

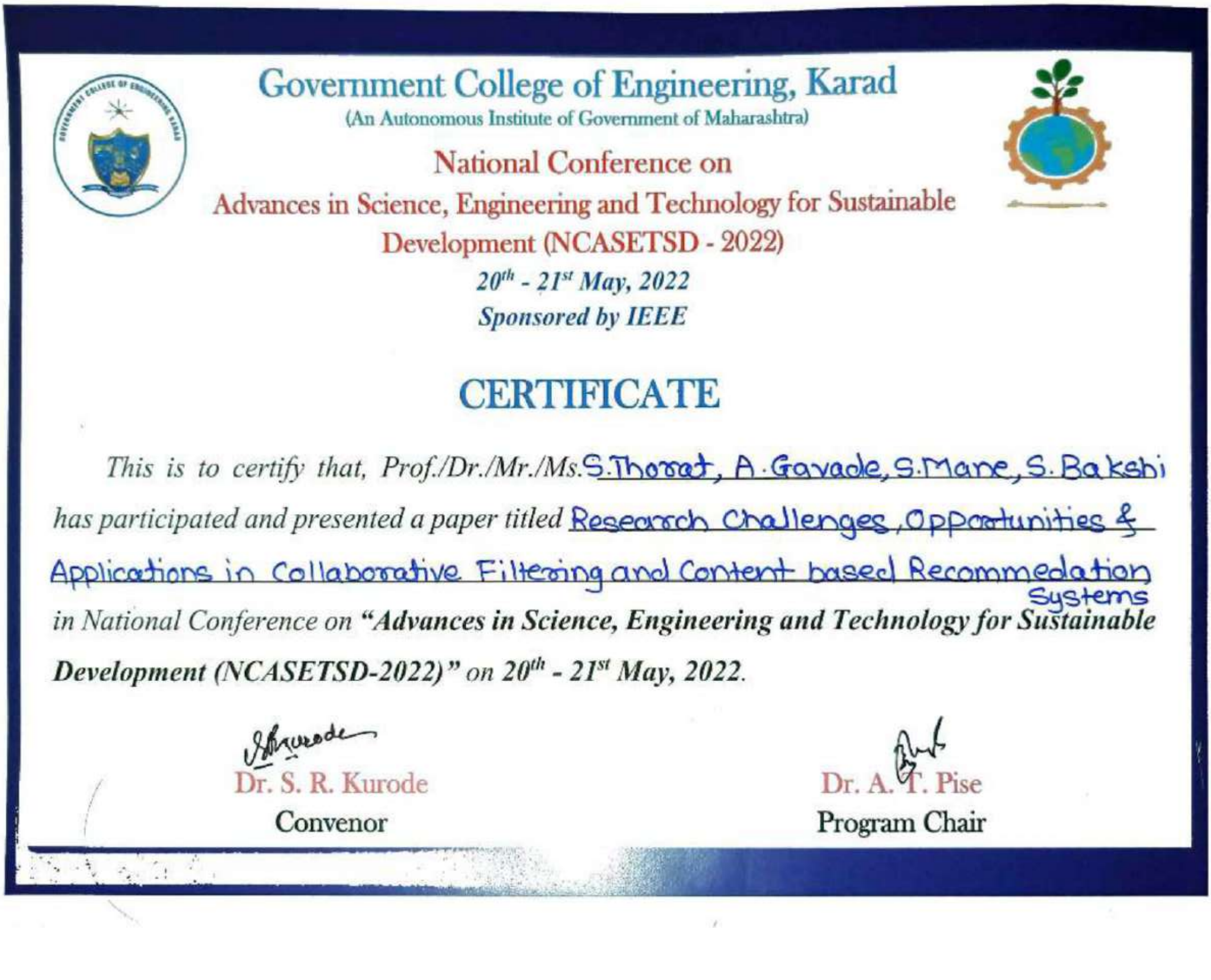
Research Challenges, Opportunities, and Applications in Collaborative Filtering and Content-based Recommendation Systems

Abstract

Due to the exponential growth of data on Internet, providing correct information to users in a reasonable amount of time has become a challenge. The recommender system acts as an information filtering tool; they provide appropriate information as per the user's choice and interest. In recent years, it has become a widespread technique that is being used in many applications. In general, recommendation systems have been classified into collaborative and content-based filtering. Due to its significance, the recommender system has become one of the most important research areas in today's context. Though recommendations systems are being used for a quite long time, many research challenges and issues in the design of effective recommendation systems are yet to be addressed in an effective manner. The purpose of this paper is to familiarize people with the research challenges and opportunities in recommender systems and their available solutions. The collaborative filtering approach stills suffer from many drawbacks, including data sparsity, gray sheep, cold start problem, and scalability. The content-based filtering approach suffers from reciprocity, sparsity and limited content analysis issues. This paper reviews existing research approaches to overcome these challenging issues. We have also discussed future research directions in collaborative filtering and content-based recommendation systems. Various application domains have listed out where recommendation systems can be improved, such as healthcare, agriculture, etc. The paper describes possible future extensions in all these applications. Overall this paper will act as a guide for those who are interested in doing research in the recommendation system.

* Computer Science Department, Rajarambpau Institute of Technology, Sangli, India,

sandip.thorat@ritindia.edu, ashwinihakke@gmail.com, seema.chavan1012@gmail.com, bakshisourabh@gmail.com



Government College of Engineering, Karad

(An Autonomous Institute of Government of Maharashtra)

National Conference on

Advances in Science, Engineering and Technology for Sustainable Development (NCASETSD - 2022)

20th - 21st May, 2022

Sponsored by IEEE

CERTIFICATE

This is to certify that, Prof./Dr./Mr./Ms. S.Thorat, A.Gavade, S.Mane, S.Bakshi *has participated and presented a paper titled* Research Challenges, Opportunities & Applications in Collaborative Filtering and Content based Recommedation Systems *in National Conference on* ***"Advances in Science, Engineering and Technology for Sustainable Development (NCASETSD-2022)"*** *on* ***20th - 21st May, 2022***.

Dr. S. R. Kurode
Convenor

Dr. A. T. Pise
Program Chair

Content-based Research Paper Recommendation (C-RPR) System using Bi-LSTM Neural Network

Seema Mane[1], Sandeep Thorat[2] and Ashwini Patil[3]

[1] Computer Science Department, Rajarambpau Institute of Technology, Sangli, India,

E-mail: seema.chavan1012@gmail.com

[2] Computer Science Department, Rajarambpau Institute of Technology, Sangli, India,

E-mail: sathorat2003@gmail.com

[3] Computer Science Department, Rajarambpau Institute of Technology, Sangli, India,

E-mail: ashwini.patil@ritindia.edu

Abstract

Due to the exponential growth of data every day on the internet, it has become a pervasive problem of information overload and finding relevant information. In recent years, it has become a widespread technique used by many e-commerce applications, such as article recommendations, movie recommendations, product recommendations, music recommendations, etc. It becomes challenging for researchers to identify the most similar research papers and choose the best venue to publish them as a result of the large number of papers being submitted to various venues. The proposed recommendation system uses a content-based filtering approach using a deep neural network and helps researchers to submit their manuscripts to the most suitable venue and also recommends the most similar research paper to do their research in a smooth way. The C-RPR model uses a natural language processing technique where TF-IDF is used for vectorization, and the cosine similarity technique is used as a similarity measure to recommend similar papers. Also, Bi-LSTM is used to recommend an appropriate venue by training a model on a research publication dataset for computer science journals with attributes such as ID, title, abstract, author, venue, etc. Compared to other models that predominantly make use of machine learning algorithms and feature selection techniques, the proposed model produced better results with 74.55% accuracy using Bi-LSTM.

Keywords:

Recommendation System, Content-Based Filtering, Bi-LSTM, TF-IDF, Cosine Similarity.

1. INTRODUCTION

Recommendation systems (RS) are a rapid transformation in e-commerce and play a very important role in many areas, such as product recommendation, movie recommendation, news recommendation, and book recommendation. Over the past few years, scientific article recommendations have become increasingly popular. It is getting more and harder for scholars to identify pertinent articles and suitable venues to submit their papers as the number of scholarly publications in various sorts of journals and conferences grows tremendously. A lot of journals and conferences are receiving a variety of articles. Therefore, recommending relevant research articles to busy researchers will help them stay current with their field of study and avoid information overload [1].

The recommender system gathers data from various resources to make recommendations. These recommendations are made by considering the interests and previous history of user. This paper is proposed to recommend similar papers and appropriate venues. Many authors outperformed recommendations based on keywords and titles. The recommendation system using abstracts for authors is still under research. The proposed system performs recommendations of top N research papers and venues based on the abstract. The recommender system helps users choose items based on their interests by filtering products. Many recommendation algorithms emerge from the recommendation system, guiding users to find their interests in a large space [10, 11, 12]. The three primary classifications of recommendation systems are Content-Based Filtering (CBF), Collaborative

Search mail

Thank u...

Journals Christ University <journals@christuniversity.in>
to me

Dear Author

Greetings from Centre for Publications, CHRIST (Deemed to be University)!

3661 - "Content-based Research Paper Recommendation (C-RPR) System using Bi-LSTM Neural Network" submitted on 13 Aug 2

Currently the submitted papers are being reviewed for Oct-Dec 2022 Issue, On or Before 30/9/22, you would get to know the status of your Paper.

Thank you for your Patience.

Warm Regards

seema chavan <seema.chavan1012@gmail.com>
to Journals

Thank you for your response.

UGC-CARE List

You searched for "**Sciences"**. Total Journals : **448**

Search: 0975-3303

Sr.No.	Journal Title	Publisher	ISSN	E-ISSN	UGC-CARE coverage year	Details
333	Mapana Journal of Sciences	CHRIST Deemed to be University	0975-3303	NA	from April - 2022 to Present	View

Showing 1 to 1 of 1 entries (filtered from 448 total entries)

Previous 1 Next

K.E. Society's

Rajarambapu Institute of technology, Rajaramnagar

An Autonomous Institute

(Affiliated to Shivaji University)

SYNOPSIS OF M.TECH DISSERTATION

1. **Name of Program** : M.Tech (Computer Science & Engineering)
2. **Name of Student** : Seema Ramchandra Mane (2030002)
3. **Date of Registration** : June 2021
4. **Name of Guide** : Dr. S. A. Thorat
5. **Sponsored details (if any)**
6. **Proposed Title** : **"Content Based Research Paper Recommendation System using Deep Learning"**
7. **Synopsis of dissertation work:**

7.1 Relevance

Recommender system (RS) is undergoing rapid transformation in almost all aspects. These Systems have been applied to many areas, such as movie recommendations, music recommendations, news recommendations, web page and document recommendations. Many companies have employed and benefited from recommender systems, such as the book recommendation of Amazon, music recommendation of Apple Music, and product recommendation. There are majorly three types of recommender systems which work primarily in the Media and Entertainment industry: Collaborative Recommender system, Content-based recommender system and Knowledge based recommender system. Today, most of the RSs are developed based on the first two approaches which are Content-based filtering and collaborative based filtering. Content-based filtering methods are based on a description of the item and a profile of the user's preferences. These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user. Content-based recommenders learn a based on an item's features. Collaborative methods for recommender systems are methods that are based solely on the past interactions recorded between users and items in order to produce new recommendations. These interactions are stored in the so-called "user-item interactions matrix". Knowledge based recommendation works on functional knowledge: they have knowledge about how a particular item meets a particular user need, and can therefore reason about the relationship between a need and a possible recommendation. This type of recommender system attempts to suggest objects based on inferences about a user's needs and preferences.

7.2 Techniques

Deep Learning:

The future of personalized content and product recommendations will be heavily shaped by deep learning (DL) and its ability to predict the next natural item or product for a visitor. In the last decade, the scientific world has been thrilled by the tremendous success of deep learning (DL) methods, playing a major role in how artificial intelligence (AI) has flourished over the past few years. The most notable of these revolutions have been made possible by DL in computer vision and natural language processing (NLP). The architecture of all deep learning models includes a multiple-layers structure, with every layer consisting of nodes that perform a particular mathematical operation, which is called an activation function.

NLP:

Natural Language Processing (NLP) is one of the key components in Artificial Intelligence (AI), which carries the ability to make machines understand human language. NLP allows machines to understand and extract patterns from such text data by applying various techniques such as text similarity, information retrieval, document classification, entity extraction, clustering. This is where the concepts of Bag-of-Words (BoW) and TF-IDF come into play. Various word embedding techniques are being used i.e., Bag of Words, TF-IDF, word2vec to encode the text data.

Bag of Words (BoW):

The Bag of Words (BoW) model is the simplest form of text representation in numbers. Like the term itself, we can represent a sentence as a bag of words vector (a string of numbers). Bag of Words just creates a set of vectors containing the count of word occurrences in the document. In this approach, scores in the document-term matrix are simply the number of occurrences of the terms in each document. The order of the words does not matter; it just counts the number of terms. But as an alternative, we can use the n-gram model to force the model to care about the order of words. E.g. we can use 2 sets of words as a term in bi-gram.

TF-IDF:

"Term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus." Term Frequent (TF) is a measure of how frequently a term t appears in a document. IDF is a measure of how important a term is. We need the IDF value because computing just the TF alone is not sufficient to understand the importance of words. We can calculate the IDF values for each word. By computing the TF-IDF score for each word in the corpus, Words with a higher score are more important, and those with a lower score are less important. TF-IDF model contains information on the more important words and the less important ones as well. TF-IDF usually performs better in machine learning models.

Cosine similarity:

From the similarity score, a custom function needs to be defined to decide whether the score classifies the pair of chunks as similar or not. Cosine similarity returns the score between 0 and 1 which refers 1 as the exact similar and 0 as the nothing similar from the pair of chunks. In regular practice, if the similarity score is more than 0.5 than it is likely to similar at a somewhat level.

7.3 Present Theories and Practices

In [1], paper presents a preliminary survey of different recommendation system based on filtering techniques, challenges applications, and evaluation metrics. The existence of recommender system had been identified in the late 1970s, ever since many researchers have proposed various approaches to develop efficient recommender system. The motive of work is to introduce researchers and practitioner with the different characteristics and possible filtering techniques of recommendation systems. Various methods are being proposed to develop an effective recommendation system out of them; two form the basis for the development of other approaches. These methods are content filtering, collaborative filtering. There are many techniques used to compute the similarity between the users like Pearson Correlation Similarity, Cosine Similarity. Challenges like data sparsity, cold start, scalability, shilling attack and gray sheep are discussed. The quality of recommendation system is measured through various types of evaluation metric based on the accuracy of prediction and coverage. The selection of metric depends on filtering technique, features of data set, and the task of recommendation system. Evaluation metrics are categorized as prediction accuracy metrics (MAE, RMSE) and classification accuracy metrics (precision, recall, F-measures).

In [2], author proposed work aims to develop a recommendation system for scholarly use. There are two data components of research: the datasets and the articles for recommendation. For the datasets, author collected metadata (title and summary) of datasets from GEO. For the articles, they collected the archived MEDLINE article details (title, abstract, Medical Subject Headings (MeSH) terms, date of publication) from PubMed8 for recommendation. Vector space model (VSM) techniques were utilized to convert each dataset and article into vectors. Similar articles for a given dataset were collected using the cosine similarity. For evaluation, distributional vectors constructed using multiple techniques, such as term frequency and inverse document frequency (TF-IDF), BM25, latent Dirichlet allocation (LDA), latent semantic analysis (LSA), word2vec, and doc2vec. The study proposes a similarity-based literature recommendation system for datasets collected from GEO. This work is a first step towards enhancing the experience of data reusability by recommending literature for datasets. An information retrieval paradigm was applied for literature recommendation and the top-performing literature recommendation technique obtained.

In [3], an author proposed a recommender system on computer science publications referred to as the Publication Recommender System (PRS). This system is based on a new content-based filtering (CBF) recommendation model using chi-square and softmax regression, which are combined to construct a real-time online system. The contributions of the recommendation system is: (1) PRS is a non-profit driven recommender system covering 66 top computer science publication venues across more than five digital libraries, such as Springer, IEEE, ACM, AAAI and SIAM. It can simultaneously recommend top journals and conferences using the input of abstract or the whole manuscript. (2) Considering the continually expanding field of computer science and daily updated corpus of the ever-changing, the recommendation ability for cutting edge research areas and topics is essential. The model can automatically update once a new training set is formed. (3) PRS could respond to user in real time and easily be deployed on a web server. In order to make PRS more practical, all methods used in PRS are designed to be less computational complexity. Publication Recommender System consists of two modules: feature selection module and softmax regression module. Feature vector space is generated in feature selection module and feature vectors are used to train softmax regressor in softmax regression module. Term frequency and inverse document frequency (TF-IDF) can recognize the important words or phrases of articles. The chi-square statistic measures the dependence between the term t and a category c (such as, in case of computer journals or conferences). Softmax regression is chosen to be the classifier because there are many journals and conferences in the recommender system and all journals and conferences need to be ranked for recommending according to the classification scores.

In this paper [4] author propose a type of CBF that uses a multiattribute network (MN), which comprises entire attribute information for different items. Many possible attributes of CBF can play significant roles in determining the quality of the recommendation results, because they may provide sufficient information for measuring sophisticated similarities. This approach can address the sparsity problem and the over-specialization problem that frequently affect recommender systems. The overall process of the CBF–MN algorithm is described as: (1) acquiring item attributes, (2) calculating item similarities, (3) generating an MN containing all the items, (4) network clustering to group the items, and (5) calculating a score that reflects the importance of each item selected in the cluster, and using this score for recommendation. Modularity clustering is algorithms among network-based clustering techniques identify the proper number of clusters by considering the link structures between observations.

In this paper [5] author proposed model is based on the encoder-decoder architecture with the attention mechanism. In encoder they leverage the TDNN designed to capture long-term dependencies with a 1-dimension convolution over all possible word windows for a given context. Relu act as Convolutional layer which is nonlinear activation function. The phrase level representation obtained by the TDNN provides a trade-off, between capturing semantics and computational time. Gated Recurrent Unit (GRU) utilized to help prevent the vanishing or exploding gradient problem. The recurrent neural network decoder consults this representation when determining the optimal paper to recommend based solely on its title.

In this paper [6] system calculates byte frequency distribution using 1-gram based approach to make payload profile. The packet payload length has a strong impact on the byte frequency distribution, so multiple profiles are created for different payload lengths. Due to this, number of profiles for a particular service becomes very large. To minimize the complexity of profile comparisons, profiles are clustered together. The system is trained in an unsupervised way for profile creation. In the testing phase the system captures incoming payloads and compares the payload with stored normal profiles. If the new payload profile does not match with any stored profile for the same service, then an alert is generated indicating a suspicious packet.

In this paper [7] the number of users' data is always large-scale, traditional algorithms cannot effectively cope with e-commerce personalized recommendation tasks. Author proposed an e-commerce product personalized recommendation system based on learning clustering representation. Traditional KNN method has limitation in selecting adjacent object set. Thus, introduced neighbor factor and time function and leverage dynamic selection model to select the adjacent object set and combined RNN as well as attention mechanism to design the e-commerce product recommendation system.

7.4 Proposed Work

The work is proposed to study existing methods used for developing content based recommendation systems and perform comparative analysis of these methods. As a part of this work we will select an effective method from existing system and develop a recommendation system. This work will attempt to design a content based recommendation system using deep learning and compare the performance of same with existing systems.

7.5 Objectives

1. To perform literature survey on content based recommendation system for research paper publications and identify gap for new work.
2. To study and implement an existing NLP content based recommendation

system for computer science publications.

3. Proposing a novelty in existing NLP content based recommendation system for computer science publications to achieve improvement using deep learning.
4. To perform the performance comparison of existing NLP content based recommendation system for computer science publications and the proposed content based research paper recommendation systems using deep learning.

Possible Outcomes

1. Identification and study of various methods for recommendation system
2. Implementation of novel content based research paper recommendation system using deep learning which gives better experience to the user.

Proposed work is planned in following phases.

Phase I - Literature Survey and Synopsis Preparation:
Duration:Nov 2021 – Dec 2021

In this phase we aim to do the Literature survey by collecting different journal papers on that basis:

- Identify and study various methods of content based recommendation system for research paper publications.
- Identifying various challenges of current content based recommendation system for research paper publications
- Using literature survey, preparing the synopsis for the dissertation work

Phase II - Implement an existing content based research paper recommendation system, Report writing and submission.
Duration: Jan 2022 – Feb 2022

In this phase we plan to carry out following task.

1. To find existing NLP techniques to design content based recommendation system for computer science publications.
2. To perform critical analysis and identify merits and demerits of the system and work on possible refinements in these system

Phase III- Implementation of the proposed system and collecting experimental observations.
Duration: March 2022 – May 2022

In this phase we plan to carry out following task.

1. To implement content based research paper recommendation system using deep learning.
2. To collect experimental observations.

Phase IV- Comparison and analysis of experimental results of proposed method and earlier method. Report writing and submission.
Duration: June 2022–July 2022

In this phase we plan to carry out following task.

1. Comparison and analysis of experimental results of proposed content based research paper recommendation system using deep learning and existing NLP content based recommendation system for computer science publications.

8. Facilities Required for Project:

To carry out dissertation work, facilities that will be needed are mentioned below:

- Operating System: Windows
- Programming Language: Python

9. Expected Date for Completion of Work: July2022

10. Approximate Expenditure: Nil

Date: Seema Mane

Place: RIT, Rajaramnagar Student

Dr. A. C. Adamuthe
Guide,
Dept. of CSE
R.I.T. Rajaramnagar

Dr. S. S. Patil
HOP,
Dept. of CSE
R.I.T. Rajaramnagar

Dr. N. V. Dharwadkar
HOD,
Dept. of CSE
R.I.T. Rajaramnagar

6. Bibliography

1. Recommendation Systems: Techniques, Challenges, Application, and Evaluation [Raghuwanshi, Sandeep K. and Rajesh Kumar Pateriya. SocProS (2017)].

2. A content-based literature recommendation system for datasets to improve data reusability – A case study on Gene Expression Omnibus (GEO) datasets [Braja Gopal Patraa, Vahed Maroufya, Babak Soltanalizadeha, Nan Denga published in April 2020, publisher Journal of Biomrdical Informatics]

3. A content-based recommender system for computer science publications [Donghui Wang, Yanchun Liang, Dong Xu, Xiaoyue Feng, Renchu Guan in Oct 2018, publisher: Knowledge-Based Systems]

4. Content-based filtering for recommendation systems using multiattribute networks [Jieun Son, Seoung Bum Kim published in Dec. 2017,publisher: Expert Systems with Applications]

5. Neural Citation Network for Context-Aware Citation Recommendation [Travis Ebesu and Yi Fang published in Aug 2017]

6. Payload Content based Network Anomaly Detection Sandeep A. Thorat Amit K. Khandelwal Bezawada Bruhadeshwar K. Kishore published in Aug 2008 , publisher: IEEE]

7. A E-commerce personalized recommendation analysis by deeply-learned clustering [Kai Wang, Tiantian Zhang, Tianqiao Xue published in Aug 2020 publisher: Journal of Visual Communication]

PAPER NAME

plagarism_2030002_Report.docx

AUTHOR

Seema Mane

WORD COUNT 11811 Words	CHARACTER COUNT 66451 Characters
PAGE COUNT 61 Pages	FILE SIZE 1.5MB
SUBMISSION DATE Nov 23, 2022 12:51 PM GMT+5:30	REPORT DATE Nov 23, 2022 12:52 PM GMT+5:30

● 9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 9% Publications database
- Crossref Posted Content database
- Crossref database

● Excluded from Similarity Report

- Internet database
- Bibliographic material
- Cited material
- Submitted Works database
- Quoted material
- Small Matches (Less then 25 words)

9% Overall Similarity

Top sources found in the following databases:

- 9% Publications database
- Crossref database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Donghui Wang, Yanchun Liang, Dong Xu, Xiaoyue Feng, Renchu Guan. "... Crossref	2%
2	Braja Gopal Patra, Vahed Maroufy, Babak Soltanalizadeh, Nan Deng, W.... Crossref	<1%
3	Sandeep A. Thorat. "Payload Content based Network Anomaly Detectio... Crossref	<1%
4	"Soft Computing for Problem Solving", Springer Science and Business ... Crossref	<1%
5	Jieun Son, Seoung Bum Kim. "Content-based filtering for recommenda... Crossref	<1%
6	Kai Wang, Tiantian Zhang, Tianqiao Xue, Yu Lu, Sang-Gyun Na. "E-com... Crossref	<1%
7	Benard Magara Maake, Sunday O. Ojo, Tranos Zuva. "chapter 6 A Surve... Crossref	<1%
8	Benard M. Maake, Sunday O. Ojo, Tranos Zuva. "chapter 5 Information ... Crossref	<1%
9	Tham Vo. "An integrated network embedding with reinforcement learni... Crossref	<1%

10	P. Venkateswara Rao, A. P. Siva Kumar. "The societal communication ... Crossref	<1%
11	Akshi Kumar, Simran Seth, Shivam Gupta, Shubham. "Sentiment-Enha... Crossref	<1%
12	Jyoti Shokeen, Chhavi Rana. "An Application-oriented Review of Deep ... Crossref	<1%
13	"Intelligent Techniques for Web Personalization", Springer Science and... Crossref	<1%
14	Matjaž Kragelj, Mirjana Kljajić Borštnar. "Automatic classification of ol... Crossref	<1%
15	Muhammad Asim, Shah Khusro. "Content Based Call for Papers Reco... Crossref	<1%
16	"Design, User Experience, and Usability. Design for Contemporary Inter... Crossref	<1%
17	"Big Data and Networks Technologies", Springer Science and Business ... Crossref	<1%
18	"Trends in Artificial Intelligence Theory and Applications. Artificial Intel... Crossref	<1%
19	Kristian Wahyudi, Johanes Latupapua, Ritchie Chandra, Abba Suganda ... Crossref	<1%

Content-based Research Paper Recommendation (C-RPR) System using Deep Learning

by Seema Mane

General metrics

66,451	11,811	535	29 min 52 sec	48 min 22 sec
characters	words	sentences	reading time	speaking time

Score

86

This text scores better than 86% of all texts checked by Grammarly

Writing Issues

217	✓	217
Issues left	Critical	Advanced

Unique Words

Measures vocabulary diversity by calculating the percentage of words used only once in your document

20%

unique words

Seema Satyendra Chavan

E-mail:
seema.chavan1012@gmail.com Mob:
9860305680

OBJECTIVES

To be a part of organization where I can learn new things and utilize my knowledge, skills and abilities up to maximum extent for benefit of mine and organization both.

WORKING EXPERIENCE (5 yrs)

Post held - Visiting Lecturer
Institution - Govt. women's residence polytechnic, Tasgaon.
Duration - 17/1/2013 to 19/10/2013

Post held - Visiting Lecturer
Institution - Govt. women's residence polytechnic, Tasgaon.
Duration - 8/8/2016 to 19/03/2020

Post held - Lecturer
Institution - DKTE's Yashwantrao Chavan Polytechnic, Ichalkaranji.
Duration - 9/12/2021 to till date

EDUCATIONAL QUALIFICATION

- Persuing MTech in Computer Science & Engineering department at Rajarambapu Institute of Technology, Islampur.
- B.E (CSE) in May 2012 from Dr. J. J. Magdum college of Engineering with 63.47%.
- HSC in May 2008 from Jaysingpur College, Jaysingpur with 59.33%.
- S.S.C in May 2006 from Shri. Jaysingrao Ghatge Vidyamandir & Highschool, Kagal with 74.26%.

TECHNICAL SKILLS

• Language	–	C, C++, JAVA, Python, Android
• Markup & Scripting Language	–	HTML , CSS
• Tools & IDE	–	Anaconda (Spyder), Visual Studio Code, Eclipse Oracle 10g
• Databases	–	MySQL, Oracle 10g

PROJECT DESCRIPTION

Proposed Project Details (B.E):

- **Title:** Implementation of Evolving Email Clustering Method for Email Grouping.
- **Description:**

  This paper presents the design and implementation of a new system to manage email messages using email evolving clustering method with unsupervised learning approach to group emails base on activities found in the email messages, namely email grouping.
  This project is based on IEEE paper 2010 "Implementation of Evolving Email Clustering Method for Email Grouping".

Proposed Project Details (MTech miniproject – SEM-I):

- **Title:** Heart Disease Prediction System Using Machine Learning.
- **Description:**

  Heart Disease Prediction System (HDPS) is developed using different Machine learning algorithms to classify and predict the patient with heart disease. The proposed work predicts heart disease by exploring the four classification algorithms and does performance analysis. The objective of this study is to effectively predict if the patient suffers from heart disease. This project shows the analysis of various machine learning algorithms, the algorithms that are used in this are SVM, K nearest neighbors (KNN), Logistic Regression, Naïve Baye's and Decision Tree Classifiers used for accurately diagnose Heart Disease.

Proposed Project Details (MTech miniproject – SEM-II):

- **Title:** Scientific Paper Recommendation using Deep Learning.
- **Description:**

  Scientific Paper Recommendation System is developed using Recurrent Neural Network to recommend a list of high-quality candidate papers based on query given to reduce amount of time researchers spend in searching for existing work and to reduce the problem of information overload. The Recommendation System implemented on standard dataset of arxiv having 502353 records. It uses TF-IDF mechanism to vectorize the text data and Encoder-Decoder mechanism, GRU with attention mechanism to train the model.

EXTRACURRICULAR ACTIVITY

- Certification in “**Introduction to Machine Learning**” by Coursera
- Certification in “**Mobile Application Development”** (Android).
- Certification in Online Faculty Development Program on **“Machine Learning”.**
- Certification in Online Faculty Development Program on **“Data Analytics”.**
- Certification in Online Faculty Development Program on “**Cyber Security**”
- Certification in Online Faculty Development Program on “**Research Methodology**”
- Certification in Online Faculty Development Program on “**Machine Learning with Python**”
- Participated in & 7 Day course on “**Android Apps Development**”

PAPER PUBLISHED

- ➢ Got Best Paper award for the Paper on “Research Challenges, Opportunities and Applications in Collaborative Filtering and Content-based Recommendation System” in National Conference on Advances in Science, Engineering and Technology for Sustainable Development (NCASETSD -2022).
- ➢ Paper published on Privacy Preserving Search Over Encrypted Data with Secure and Dynamic Operation in Cloud Computing in IJCSE
- ➢ DECENTRALIZE ELECTRONIC VOTING SYSTEM USING BLOCKCHAIN in IJCST
- ➢ Heart Disease Prediction System Using Machine Learning at (RIT – RAIET) Conference

LEISURE INTEREST

Making greeting cards (handmade), Singing a song.

STRENGTH

- o Ability to work in team
- o Ability to meet deadline.