Chapter 2

Background

> *"Information networks straddle the world. Nothing remains concealed. But the sheer volume of information dissolves the information. We are unable to take it all in."*
>
> -Gunter Grass (1927-2015)

In this chapter, we present an overview of the state-of-the-art in academic recommender systems. We mainly discuss journal and collaborator recommender systems, although we cover few literature from citation and reviewer recommendations related to our work. We have also discussed the principal evaluation metrics, methodologies, public datasets commonly used in the field.

2.1 Literature Review in Journal Recommendation

We provide here necessary background for journal recommender systems according to the taxonomy, as discussed in Section 1.2.

2.1.1 Collaborative Filtering-based Recommendation (CF)

Collaborative recommender systems (or collaborative filtering systems) predict the utility of items for a user based on the items previously rated by other users who have similar likings or tastes [4]. In the field of academic recommendations, Yang et al. [59] proposed a model to explore the relationship between publication venues and writing styles using three kinds of stylometric features: lexical, syntactic and structural. In another paper, Yang et al. [60] used a collaborative filtering model incorporating writing style and topic of papers to recommend venues. Yang et al. [30] proposed another joint multi-relational model (JMRM) of venue recommendation for author-paper pairs. This model utilized different tensors to represent relations among authors, venues, and papers in the academic environment that are highly coupled with each other.

Hyunh et al. [61] proposed a collaborative knowledge model (CKM) to organize collaborative relationships among researchers. The model quantified the collaborative distance, the similarity of actors before recommendations. Yu et al. [62] proposed a prediction model that used collaborative filtering for a personalized academic recommendation based on the continuity feature of a user's browsing content. Liang et al. [35] proposed a probabilistic approach consolidating user exposure that was modeled as a latent variable, inducing its incentive from data for collaborative filtering. Alhoori et al. [36] recommended scholarly venues taking into account a researcher's reading behavior based on personal references and the temporal factor as to when references were added. Trappey et al. [63] presented a new patent recommendation system based on clusters of users having similar patent search behaviors.

2.1.2 Content-based Recommendation (CBF)

In CBF, users are recommended items similar to the ones the user preferred in the past. In case of academic recommender systems, a user is recommended papers, collaborators, and/or venues *similar to* that the user liked earlier. The requirement to find *similarity* between journals and manuscripts started in the seventies, when Kochen et al. [64] proposed a way to recommend journals for authors' manuscripts based on relevance, acceptance rate, circulation, prestige, and publication lag of journals. Medvet et al. [49] considered the title and abstract of papers to recommend scholarly venues considering *n*-gram based Canvar-Trenkle, two-steps-LDA, and LDA+clustering to retrieve language profile, a subtopic of papers, and identification of the main topic as a research field.

Errami et al. [65] proposed a model called eTBLAST to recommend journals based on abstract similarity using the z-score of a set of extracted keywords and journal score. Schurmie et al. [66] proposed the Journal/ Author Name Estimator (Jane) [1] on biomedical database MEDLINE to recommend journals based on abstract similarity. They exploited

[1] http://jane.biosemantics.org

a weighted k-nearest neighbors and Lucene similarity score in order to rank articles. Similarly, Wang et al. [27] presented a content-based publication recommender system (PRS) for computer science articles using soft-max regression and chi-square based feature selection techniques.

Recently, few online services have started providing support for suggesting journals using keywords, title, and abstract matching. These services include Elsevier Journal Finder [2] [50] , Springer Journal Suggester [3], Edanz Journal Selector [4] and EndNote Manuscript Matcher [5] etc. Elsevier Journal Finder requires only the title and abstract of a paper and uses noun phrases as features and Okapi BM25+ to recommend journals. But, recommendations are restricted to Elsevier publishers only [50].

2.1.3 Hybrid Recommendation

Hybrid approaches combine collaborative and content-based methods avoiding certain limitations of content-based and collaborative systems. Wang et al. [27] proposed hybrid article recommendations incorporating social tag and friend information. Boukhris et al. [67] suggested a hybrid venue recommendation based on the venues of the co-citers, co-affiliated researchers, the co-authors of the target researcher. It is based on bibliographic data with citation relationships between articles.

Minkov et al. [68] introduced a method of recommending future events. Tang et al. [56] introduced a cross-domain topic learning (CTL) model to rank and recommend potential cross-domain collaborators. Xia et al. [69] proposed a socially aware recommendation system for conferences. Similarly, Cohen et al. [55] explored the domain of mining specific context in a social network to recommend collaborators.

2.1.4 Social Filtering-based Recommendation (SF)

On top of the above approaches, the approach based on a network representation of the input data has gained considerable attention in the recent past [38], mainly to alleviate the problems of the CF and CBF approaches. Here, a social graph is built among the

[2] http://journalfinder.elsevier.com

[3] http://journalfinder.com

[4] https://www.edanzediting.com/journal-selector

[5] http://endnote.com/product-details/manuscript-matcher

authors based on co-authorship. An edge exists between two authors if they co-author at least one paper [40, 70]. The venue having the highest count among the papers within n-hops from a given author-node is recommended.

Klamma et al. [47] proposed a Social Network Analysis (SNA) based method using collaborative filtering to recognize most similar researchers and rank obscure events by integrating the rating of most similar researchers for the recommendations. Silva et al. [71] proposed a three-dimensional research analytics framework (RAF), incorporating relevance, productivity, and connectivity parameters. Pham et al. [72] used the number of papers of a researcher in a venue to determine her rating for that venue using the clusters on social networks. Later, Pham et al. [73] presented clustering techniques on a social network of researchers to identify communities in order to generate venue recommendations. They also applied traditional CF calculations to provide the suggestions.

Chen et al. [74] introduced a model AVER to recommend the scholarly venues to a target researcher. Their approach utilizes a random walk with restart (RWR) model on the co-publication network incorporating author-author and author-venue relations. Later, Yu et al. [31] extended AVER to personalized academic venue recommendation model PAVE, where the topic distribution of researcher's publications and venues were utilized in LDA. Luong et al. [53] identified suitable publication venues by investigating the co-authorship network, most frequent conferences, normalized scores based on most successive conferences. Luong et al. [40] in another work, recommended suitable publication venues by investigating authors' co-authorship networks in a similar field. Xia et al. [69] provided venue recommendations using Pearson correlation and social information of conference participants to enhance smart conference participation.

2.1.5 Deep Learning-based Recommendation

Article recommendation using dynamic attention and deep learning-based model has been proposed by Wang et al. [75]. Models are trained to capture the underlying selection criteria for article selection. It is done by automatic representation learning of each article and its interaction with the metadata and adaptively captures changes in such criteria by hybrid attention-based deep learning model. Ebesu et al. [76] propose a novel neural probabilistic model that jointly learns the semantic representations of citation contexts and cited papers. The probability of citing a paper, given a citation is estimated by training a multi-layer neural network model.

Hassan et al. [77] present a personalized research paper recommendation system that recommend papers based on users' explicit and implicit feedback. The users are allowed to specify the papers of their interest explicitly. Users' viewing behavior, e.g., viewing abstracts only or full-text, will be further analyzed to enhance users' profile and quality of the recommendation further.

Yang et al. [78] presented a long-short-term memory (LSTM) based model for context-aware citation recommendation. The model first learns the distributed representations of the citation contexts and the scientific papers separately and then measure the relevance based on the learned features. Feng et al. [79] proposed a journal recommender system Pubmender to suggest suitable PubMed journals for biomedical literatures based on the paper's abstract. Pubmender uses pre-trained word2vec to construct a start-up feature space. Then this matrix is passed through a deep CNN model to obtain the high-level representation of abstract, and finally, a Softmax layer is added to generate the recommendations by choosing journals with the highest probabilities.

2.2 Literature Review in Collaborator Recommendation

We also provide here necessary background in the collaborator recommender systems.

2.2.1 Content-based Filtering Method (CBF)

Lee et al. [57] introduced a content-based recommendation system employing researchers' expertise and their professional social networks. Gollapalli et al. [80] proposed a content-based model for collaborator recommendation using the expertise profiles extracted from researchers' publications and academic homepages. Tang et al. [56] introduced a cross-domain topic learning (CTL) model to rank and recommend potential cross-domain collaborators.

Cohen et al. [55] proposed a keywords-based collaborator recommendation model, incorporating both researchers and a set of keywords as an input to the system. Yang et al. [81] proposed a weighted topic model for complementary collaborator recommendation.

They employed a greedy heuristic algorithm based on the probabilistic topic model. Liu et al. [34] proposed CAAR, which is designed by jointly representing scholars and research topics based on their mutual depedency and extracting scholars underlying characters for high-quality new collaborator recommendation.

2.2.2 Hybrid Method

Chen et al. [82] outlined a framework that makes suggestions of collaborators in view of a combination of the network structure similarity and the researcher's research interests between the source and target authors. Chaiwanarom et al. [83] suggested a collaborator recommendation in interdisciplinary areas of computer science using degrees of collaborative forces, temporal evolution of research interests, and similar seniority status. Kong et al. [37] proposed a system TNERec fusing both research interests and network structure. Yang et al. [84] proposed a model based on research expertise, researchers' institutional connectivity, and network proximity through SVM-rank fusion strategy.

Table 2.1: Comparison of research works on scholarly network analysis

Research	Meta-path Features	Dynamic Interest	Research Content	Scholarly-Aware Features	Hidden-Relationship	Network
Lopes [85]	No	No	No	No	No	CN
Xia [25]	No	No	No	No	No	CN
Kong [24]	No	No	Yes(Title)	No	No	CN
Xia [86]	No	No	No	Yes	No	CN
Kong [39]	No	Yes	Yes(Abstract)	No	No	CN
Zhou [54]	Yes	No	Yes(Title)	No	No	CN
DRACoR	Yes	Yes	Yes(Abstract+Title)	Yes	Yes	AAG

CN denotes Co-authorship Network (Definition 13), and AAG denotes Author-Author Graph(Definition 14)

2.2.3 Social Filtering-based Method (SF)

Co-authorship is one of the most tangible and well-documented forms of scientific collaboration [87]. Newman [88] studied many statistical properties of scientific collaboration networks, including the number of papers written by authors, numbers of authors per paper, numbers of collaborators that scientists have, and size of the component of connected scientists, and degree of clustering in the networks. Barabasi et al. [89] inferred that the dynamic and the structural mechanisms govern the evolution and topology of coauthor networks.

Liu et al. [90] conducted a coauthor network, for which he defined AuthorRank as an indicator of the impact of an individual author in the network. Their results show clear advantages of Page Rank and Author Rank over the degree, closeness, and betweenness centrality metrics. Lopes et al. [85] employed knowledge of co-researchers before publications and vector space models to recommend collaborators in an academic social network.

Random walk model is a popular model in recommendation systems. Mohsen et al. [91] investigated a random walk model combining the trust-based and collaborative filtering approaches for the recommendation. Konstas et al. [92] adopted Random Walk with Restart (RWR) integrating the rich information of both authors and co-authorship relations. Backstrom et al. [93] proposed a supervised random walk based on RWR, to predict and recommend links in social networks. Li et al. [86] proposed a system incorporating both authors and co-authorship to recommend collaborators for a target researcher.

Xia et al. [25] extended their previous model ACRec [86] and presented a system exploiting RWR approach to provide a recommendation. Zhou et al. [54] proposed a random walk with restart based collaborator recommendations in a heterogeneous bibliographic network to recommend collaborators. They used a set of meta path rules to simplify a heterogeneous network and used biased edge weighting to recommend collaborators. Kong et al. [39] used a topic clustering model on researchers' publications in each year and fixed the generated topic distribution by a time function to fit the dynamic change in interest. Kong et al. [24] used a topic clustering model on the title to identify academic domains of a researcher and then applied a random walk model to compute the researcher's feature vector.

Zhou et al. [94] proposed a model incorporating academic influence aware and multi-dimensional network analysis methods (AMIN). They mainly used activity-based collaboration relationships, specialty-aware connection, and topic-aware citation fitness for effective collaboration recommendations. Wang et al. [95] proposed a model named SCORE utilizing the weak tie relationships to provide a sustainable collaborator recommendation. They incorporated three perspectives: collaboration output, collaboration duration, and collaboration index to define collaboration sustainability. Sun et al. [96] proposed the Career Age-Aware Scientific Collaborator Recommendation (CAASCR) model consisting of three parts: authorship extraction, topic extraction, and career age-aware random walk for measuring scholar-scholar similarity.

We have discussed and analyzed the existing literature and concluded with a few state-of-the-art methods related to our proposed model DRACoR (Table 2.1).

2.3 Literature Review in Other Academic Recommendation

Over the past years, several researchers have worked on the task of paper-reviewer assignment and citation recommendation. Here we briefly introduce literature from reviewer and citation recommendation for the sake of completeness of discussion on academic recommender systems.

2.3.1 Reviewer Recommendation

Yun-hong et al. [97] use information retrieval and research analytics for reviewer recommendation by integrating three dimensions, i.e., connectivity, relevance, and quality. Tayal et al. [98] propose a novel approach to assign reviewers using type-2 fuzzy set to distribute the expertise of the reviewers across various factors. Liu et al. [99] take into account changes in the reviewer's interest trend, the relevance of the seed papers with reviewer, and authority of reviewers. Kou et al. [100] proposed a review assignment system (RAS) which automatically extracts the profiles of reviewers and submissions in the form of topic vectors. Liu et al. [101] propose an intelligent decision support approach for reviewer assignment exploiting heuristic knowledge of expert assignments and techniques of operations research.

Jin et al. [102] formulated the reviewer recommendation framework as an integer linear programming problem incorporating relevance between reviewer candidates and submissions, the interests trend of candidates, and the authority of candidates. Nguyen et al. [103] propose a decision support tool for conference review assignment using Ordered Weighted Averaging (OWA) to summarize information coming from different sources and rank each candidate reviewers. The WMD-CCA (Word Mover Distance-Constructive Covering Algorithm) was proposed by Zhao et al [104]. Moawad et al. [105] propose a novel framework MINARET exploiting the valuable information from scholarly websites such as Google Scholar, ACM DL, DBLP, etc. for identifying candidate reviewers. Jin et al. [106] propose a reviewer recommendation model based on reviewers publications. Jin et al. [107] presented a framework with the concept of integer programming problem that considers different indispensable aspects such as topical relevance, topical authority, and research interest to recommend reviewers for a group of submissions. Peng et al. [108] present a time-aware and topic-based reviewer assignment model.

Recently, few online services have started providing support for suggesting reviewers or experts using keywords, title, and abstract matching. These services include the Toronto paper matching system [109], submission sifting (SubSift) [110], the Microsoft conference management toolkit [6], the global review assignment processing engine (GRAPE) [111], Erie [112], advanced reviewer assignment system [100]. Protasiewicz et al. [113] proposed a content-based recommender system aimed at the selection of reviewers to evaluate research proposals.

2.3.2 Citation Recommendation

Gori et al. [114] proposed a random walk-based model for citation recommendation. In [115], a citation recommendation model was introduced utilizing paper content and author information. Nallapati et al. [116] introduced a joint model and reference graph incorporating topic model to recommend citations. Tang et al. [117], proposed a Boltzmann machine model with a two-layers specification for modeling article content and citation relationship.

The context-aware citation recommendation research by He et al. [118] divided the input research paper into contexts like global contexts and local contexts, and then recommended citation for a given paragraph. He et al. [119], extended their previous work with respect to lacking a bibliography by utilizing 4 different models like the language model to find citation contexts.

In [56], a cross-language citation recommendation system was introduced, incorporating context-aware features of research papers. Citation resolution systems are also

[6] https://cmt3.research.microsoft.com/Content/CMT.html

developed. One of the most prominent work in this area by Duma et al. [120], used context-based citation recommendations. Livne et al. [121], developed CiteSight, that supports contextual citation recommendation using differential search.

Liu et al. [28] presented a context-based collaborative filtering model for citation recommendation. Ren et al. [122] proposed a cluster-based reference recommendation framework with regards to heterogeneous bibliographic networks. Meng et al [123] introduced a unified graph-based model, exploring random walk. Liu et al. [124] utilized the pseudo relevance feedback (PRF) algorithm, exploiting various meta-paths on a heterogeneous bibliographic citation network. Huang et al. [29] proposed a neural network-based model for recommendation of citations.

2.4 Preliminaries

In this segment, we briefly describe some of the theoretical concepts, which have been used in further chapters.

2.4.1 Centrality Measures (Social Network Analysis)

A number of measures are used to find the central node, importance of a given node in the social network. We briefly introduce them below.

Betweenness Centrality (C_B)

C_B of a node quantifies how frequently the node shows up on different possible shortest paths between any two given nodes. Here C_B of a paper q is defined as

$$C_B(q) = \sum_{\substack{p,k,q \in V \\ p \neq k \neq q}} \frac{\sigma_{pk}(q)}{\sigma_{pk}} \tag{2.1}$$

where σ_{pk} denote the number of shortest paths from p to k and $\sigma_{pk}(q)$ denote the number of shortest paths from p to k via q.

Nodes with high betweenness act as potential deal makers [125].

Degree Centrality (C_D)

In a graph, the degree of a node is the number of edges that are adjacent to that node [126]. Higher the number of neighbors of a given node, the higher its impact is. Degree centrality of a paper p is defined as

$$C_D(p) = indeg(p) + outdeg(p) \tag{2.2}$$

where $indeg(p)$ is the number of research articles or papers citing to paper p and $outdeg(p)$ is the number of papers p is referring to.

Closeness Centrality (C_C)

The metric attempts to capture how centrally a node is located vis-a-vis other nodes and is measured as the inverse of total pair-wise distances from the node to all other nodes. Closeness centrality of a node p is defined as

$$C_C(p) = \frac{1}{\sum_{\substack{q \neq p \\ p \in V}} d_G(p, q)} \tag{2.3}$$

where $d_G(p, q)$ denotes the shortest distance between vertices p and q, i.e. the minimum length of any path connecting p and q in G.

Eigenvector Centrality (C_E)

It denotes the importance of a given node in a network based on the node's connections. A node is central to the extent that the node is associated with others who are central. It relies upon the quantity and quality of neighbor nodes that are straightforwardly associated with the node [127].

Table 2.2: Interpretation of centrality measures used in citation network [**?**]

Centrality measures	**Meaning**	**Interpretation in citation networks**
Degree	Node with most connection	How many papers can this article reach directly?
Betweenness	Connects disconnected groups	How likely is this papers to be the most direct route between two papers in the citation network?
Closeness	Rapid access to related paper-nodes	How fast can this paper reach everyone in the citation network?
Eigenvector	Connections to high-scoring nodes	How well is this paper connected to other well connected paper?
HITS	Directed weighted degree centrality	How is the content of the paper and the value of its link to other paper?

Eigenvector centrality measures not just how many papers are linked with a given paper, but additionally how many important papers are connected with the paper. Eigenvector centrality of a node p is

$$C_E(p) = \frac{1}{\lambda} \sum_{q \in B_p} a_{p,q} \times x_q \tag{2.4}$$

where a_{pq} is the (p, q)-th element in the adjacency matrix A of papers.

$$a_{pq} = \begin{cases} 1, & \text{if } q \text{ is linked to } p \\ 0, & \quad\text{otherwise} \end{cases} \tag{2.5}$$

x_q is the score of the eigenvector centrality of q, and λ is the eigenvalue of p. It measures the influence of set $q \in B_p$ consisting of all papers connected to paper p.

Hyperlink-induced Topic Search - HITS (C_H)

HITS is a link analysis algorithm based on hub and authority concept. Here, a good hub represents a paper that points to many other papers, and a good authority represents a paper that is linked by many different hubs [128].

For a graph G=(V,E), authority and hub-weights are given by $u^{(p)}$, $v^{(p)}$ respectively of a node p. The operation to update the u-weights is as follows.

$$u^{(p)} \leftarrow \sum_{q:(q,p) \in E} v^{(q)} \tag{2.6}$$

Similarly, the operation to update the v-weights are as follows.

$$v^{(p)} \leftarrow \sum_{q:(p,q) \in E} u^{(q)} \tag{2.7}$$

The set of weights $u^{(p)}$ is represented as a vector U with a coordinate for each page in G. Similarly, the set of weights $v^{(p)}$ as a vector V.

2.4.2 Information Retrieval Methods

Information Retrieval (IR) is a field of information science that finds documents from a large collection to satisfy user's information need. We briefly describe few IR techniques that have been used.

Okapi BM25+ (Probabilistic Retrieval)

Okapi BM25+ is based on the probabilistic retrieval framework [129,130], whose weighting based similarity score can be expressed as follows. Similarity between two pieces of text P_s, P_t is represented here as similarity (P_s, P_t)=

$$\sum_{t \in P_s \cap P_t} \ln\left(\frac{P - wf + 0.5}{wf + 0.5}\right)\left(\frac{(n_1+1).cf}{n_1(1 - r + r\frac{wl}{avwl}) + cf} + \delta\right)\frac{(n_3+1)qcf}{n_3 + qcf} \tag{2.8}$$

where, cf is the term t's frequency in testing paper (P_t), qcf is the term's frequency in seed paper (P_s), P is the total number of papers identified from each components, wf is the number of testing papers that hold the term t, wl is the length of abstract (in bytes), $avwl$ is the average length of papers in each components. There are few tuning parameters n_1 (between 1.0-2.0), r (usually 0.75), n_3 (between 0-1000), and δ (usually 1.0).

LDA (Latent Dirichlet Allocation)

Topic modeling is an unsupervised Bayesian model, which presents each document in a document set as a probability distribution with an unsupervised learning approach [131]. The main objective of topic model is to identify topics from large document collections by exploiting the word distribution in a corpus. It is a typical bag of words (BOW) model which assumes that a document is a collection of words and there is no order relationship between words. Here a topic is a probability distribution with all the words in the document as a support set, indicating how often the word appears in the topic.

LDA is capable of clustering words, documents, authors, and other related entities based on latent topics [132]. To be specific, given a document d, a multinomial distribution θ_d over topics T is sampled from a dirichlet distribution with parameter α. For each word w_{di} from document d_i, a topic t_{di} is picked from a topic multinomial distribution ψ_t sampled from a dirichlet distribution with parameter β. Thus, we can calculate the probability of a word w from a document d as follows:

$$P(w|d, \theta, \psi) = \sum_{t \in T} P(w|t, \psi_t) P(t|d, \theta_d) \tag{2.9}$$

Then, the likelihood of corpora C is

$$P(T, W, |\Theta, \Psi) = \prod_{d \in D} \prod_{d \in D} \theta_{dt}^{n_{dt}} \times \prod_{t \in T} \prod_{w \in W} \psi_{tw}^{n_{tw}} \tag{2.10}$$

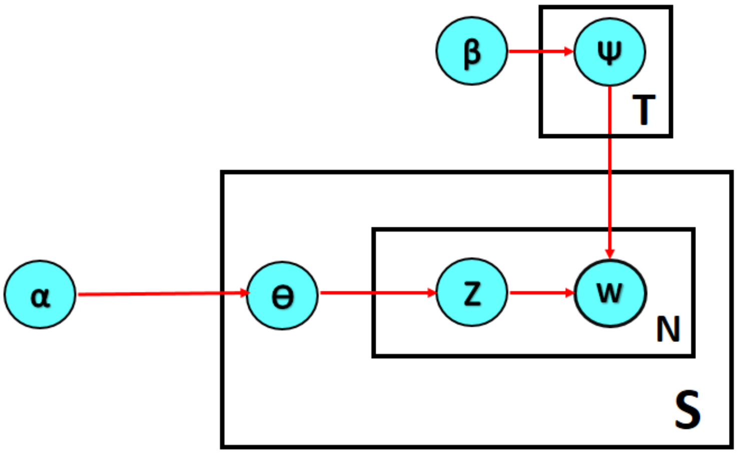

Figure 2.1: The graphical description of LDA

where n_{dt} is the number of times that the topic t has been mentioned in document d, N is the number of words in a given document (document i has N_i words) and n_{tw} represents the number of times that the word w has been associated with a topic t. The graphical representation can be seen from Fig. 2.1, where S denotes the whole document, and z denotes a specific topic.

NMF (Nonnegative Matrix Factorization)

It is a widely used tool for the analysis of high dimensional data as it automatically extracts sparse and meaningful features from a set of non-negative data vectors. Suppose we factorize a matrix V into two matrices W and H so that V$\approx WH$ (Here, the matrices to be non-negative). Given a set of multivariate n-dimensional data vectors, the vectors are placed in the columns of a $n \times m$ matrix V where m is the number of examples in the data set [133]. This matrix is then approximately factorized into an $n \times r$ matrix W and an $r \times m$ matrix H. Usually, r is chosen to be smaller than n or m, so that W and H are smaller than the original matrix V. This results in a compressed version of the original data matrix.

In text mining, let (i, j)th entry of the matrix V, could, for example, be equal to the number of times the ith word appears in the jth document in which case each column of V is the vector of word counts of a document [134]. Given such a matrix V and a factorization rank r, NMF generates two factors (W, H) such that, for all $1 \leq j \leq n$, we have

$$V(:,j) \quad \approx \quad \sum_{k=1}^{r} \quad W(:,k) \quad H(k,j), with \quad W \geq 0 \quad and \quad H \geq 0 \tag{2.11}$$

The columns of W can be interpreted as basis documents (bags of words). Assume they represent topics (sets of words found simultaneously in different documents). H tells us how to sum contributions from different topics to reconstruct the word mix of a given original document. Therefore, given a set of documents, NM identifies topics and simultaneously classifies the documents among these different topics.

2.4.3 Deep Learning Methods

Deep learning is a field of machine learning that is based on learning several layers of representations, typically by using artificial neural networks. Through the layered hierarchy of a deep learning model, the higher-level concepts are defined from the lower-level concepts. Due to multiple processing layers, deep learning models are able to learn multiple-abstract representations of data to capture both syntactic and semantic information [135]. One of the primary usages of deep learning techniques in recommender system is to enhance the accuracy of the overall recommendations. Since deep learning techniques are mainly used to extract hidden features, researchers utilize them to obtain latent factors.

In this section, we describe commonly used deep learning models. First, we introduce recurrent neural network (RNN), and then convolutional neural network (CNN) is discussed.

Recurrent Neural Network (RNN)

RNNs are a kind of feedforward neural networks which have a recurrent hidden state and the hidden state is activated by the previous states at a certain time. Therefore, RNNs can model the contextual information dynamically and can handle the variable length sequences. There are loops and memories in RNN to remember former computations. RNNs have been widely used in machine translation, speech recognition, and label generation.

When the RNN accepts a new input, it combines the implied state vector with the new input to produce an output that depends on the entire sequence. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations.

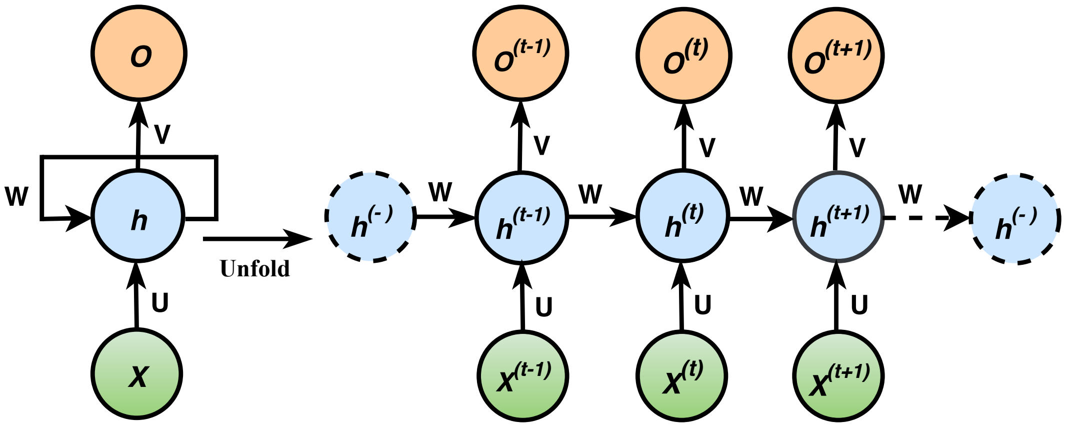

Figure 2.2: The standard RNN and unfolded

RNNs structure consists of input units, output units, and hidden units (Fig. 2.2). The most important feature of RNNs is that the nodes in the hidden layer are connected. It calculates the output of the hidden layer at the current time by obtaining the output of the input layer and the hidden layerstate at the previous time, that is, RNNs can remember the past information.

RNN can be applied directly to itself at the next timestamp, i.e., the input of the $i-th$ neuron at time t, except for the output of the $(i-1)$ layer neuron at time $t-1$, including its own input at time t. Here, x is the input, h is the hidden layer unit, o is the output, L is the loss function, and y is the label of the training set, t is the state time at time t. It should be noted that the performance of the decision unit h is determined not only by the input of this moment but also by the time before time t. V, W, and U are weights.

Long Short-Term Memory (LSTM) Method

Training conventional RNNs with gradient descent based backpropagation is difficult due to vanishing gradient and exploding gradients. To address this problem Long Short Term Memory (LSTM) [136] has been designed. LSTM is a kind of RNNs architecture and has become the mainstream structure of RNNs at present. It contains special units called memory blocks in the recurrent hidden layer [137]. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information as .

There are three gate controllers forget gate, input gate, and output gate (Fig. 2.3).

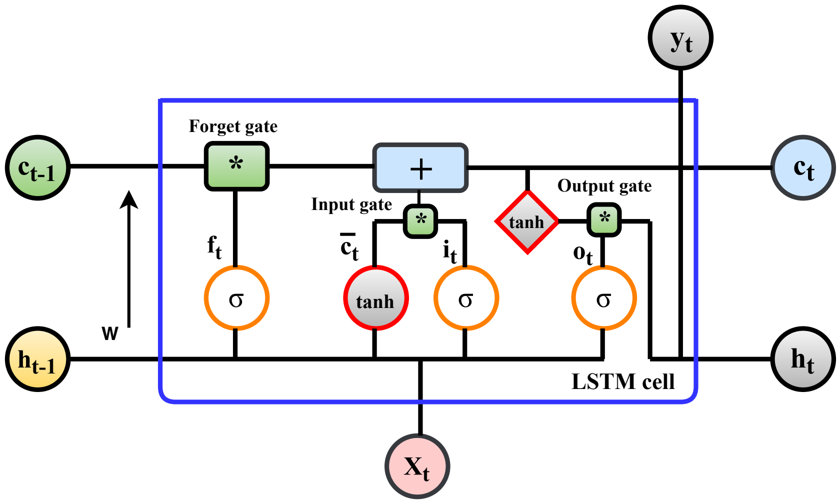

Figure 2.3: Single LSTM Cell

These gates control the short term and long term information to drop and which to store [138].

(i) **Forget gate**: It is controlled by f_t in the figure. It controls which parts of the information from long term state should be erased and which information should be kept to pass it to next long term state c_t .

(ii) **Input gate**: It is controlled by i_t in the figure. It controls which part of g_t i.e. input and previous short term state h_{t-1} should be passed to next long term state i.e. c_t. Hence input of current state is partially passed to long term memory.

(iii) **Output gate**: It is controlled by o_t. It controls which part of long term state c_{t-1} should be read and output at this time step (both to h_t) and y_t.

More formally, each cell in LSTM can be computed as follows:

$$\mathbf{i}_{(t)} = \sigma(W_{xi}^T \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hi}^T \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{2.12}$$

$$\boldsymbol{f}_{(t)} = \sigma(W_{xf}^T \cdot \mathbf{x}_{(t)} + \boldsymbol{W}_{hf}^T \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{2.13}$$

$$\boldsymbol{o}_{(t)} = \sigma(\boldsymbol{W}_{xo}^T \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{ho}^T \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \tag{2.14}$$

$$\mathbf{g}_{(t)} = \tanh(\boldsymbol{W}_{xg}^T \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hg}^T \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_g) \tag{2.15}$$

$$\boldsymbol{c}_{(t)} = \boldsymbol{f}_{(t)} \otimes \boldsymbol{c}_{(t-1)} + \boldsymbol{i}_{(t)} \otimes \boldsymbol{g}_{(t)} \tag{2.16}$$

$$\boldsymbol{y}_{(t)} = \boldsymbol{h}_{(t)} = \boldsymbol{o}_{(t)} \otimes \tanh\left(\boldsymbol{c}_{(t)}\right) \tag{2.17}$$

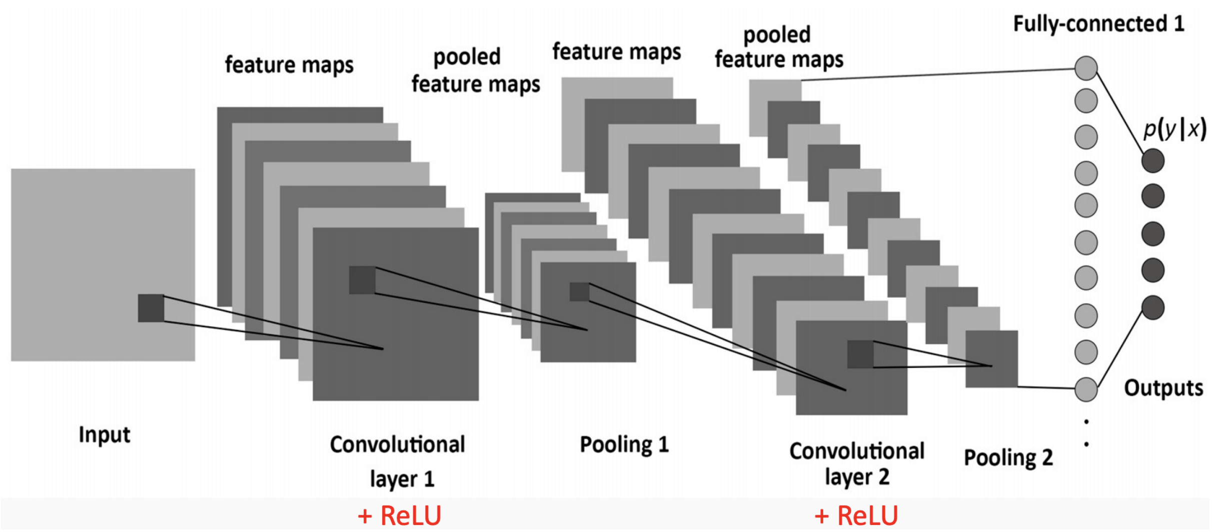

Figure 2.4: Example of CNN with two convolutional layers, two pooling layers, and a fully connected layer

Where, W_{xi}, W_{xf}, W_{xo}, W_{xg} are weight matrices of each of the four layers for their connections to the input vector x_t. W_{hi}, W_{hf}, W_{ho}, W_{hg} are weight matrices of each of the four layers for their connections to the previous short term state h_{t-1}. b_i, b_f, b_o, b_g are the bias terms for each four layers. This way LSTM can recognize an important input and store it in long term state and learn to extract whenever it is necessary.

Convolutional Neural Network (CNN)

A CNN is a type of feed-forward neural network which applies convolution operation in place of general matrix multiplication in at least one of its layers [139,140]. It can capture the global and local features and significantly enhancing the efficiency and accuracy. In recent years, CNN has been successfully applied in many difficult tasks like image and object recognition, audio processing, and self-driving cars [58]. A typical CNN consists of following components that transform the input volume into an output volume, namely, convolutional layers, Nonlinearity, pooling layers, and fully connected layers [141]. These layers are stacked to form convolutional network architecture as shown in Fig. 2.4[7]).

(i) **Convolution**: It aims to extract features from the input. Feature maps are obtained by applying convolution filters with a set of mathematical operations.

[7] https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

(ii) **Nonlinearity**: To introduce nonlinearities into the model, an additional operation, usually ReLU (Rectified Linear Unit), is used after every convolution operation.

(iii) **Pooling (Subsampling)**: Pooling reduces the dimensionality of the feature maps to decrease processing time.

(iv) **Classification**: The output from the convolutional and pooling layers represents high-level features of the input. These features can be used within the fully connected layers for classification [142].

2.5 Evaluation Strategy

We adopt following two kinds of evaluation to measure performances of the proposed system against other state-of-the-art methods.

(a) *Coarse-level or Offline Evaluation*: As the name suggests, it provides a raw-level quick idea as to how the proposed journal recommender system fares vis-a-vis other systems. We focus on the prediction accuracy to see whether the original publication venue for the test paper is predicted or not, and if yes, at what rank within some top N recommendations. Accuracy, MRR, and $F-measure_{macro}$ evaluation metrics are used during the evaluation (detailed below). We call this scenario *offline* because we can evaluate a system this way only when we have test data, of past records.

(b) *Fine-level or Online Evaluation*: This evaluation-scenario is more realistic as a researcher needs to have more than one venue recommendation from a system for her paper-in-writing that she wants to communicate. Here we go a little deeper and aim to see the relevance, usefulness, and quality of the recommended results. The system recommends an ordered list of venues that are assessed by experts in terms of graded relevance (Eqn. 3.8). Precision, nDCG, and average venue quality are used as evaluation metrics in the evaluation. We also call this type of evaluation 'online', as assessments are done post-facto.

2.6 Evaluation Metrics

We employ the following metrics that we find suitable to capture the necessary features for both types of evaluation.

(a) Accuracy@N: It is the ratio of no. of times a system correctly predicts the original entities within some fixed top N recommendations for a set of test items [49, 60]. Here we consider N = 3, 6, 9, 12 and 15 respectively.

$$Accuracy@N = \frac{\text{\# times the system correctly predicts venues within top N}}{\text{Total number of test papers}} \tag{2.18}$$

If N is small and/or the system is poor, it may fail to predict/recommend the original entity for a given item. Hence, we need to see it for a number of such items. Higher the number of papers, the better it will reflect the potential of the system. The score can be any real number between 0 and 1.

(b) Mean Reciprocal Rank (MRR): MRR is the arithmetic mean of a number of reciprocal ranks (RRs) where a RR is the inverse of the rank at which the first relevant item is retrieved in the ranked result [143].

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_{rel_i}} \tag{2.19}$$

where $rank_{rel_i}$ denotes the rank position of the first relevant item for the i-th query in a query set Q.

In offline evaluation, MRR is used to measure the capability of a system to predict the original entity of a test item. Although accuracy shows how often a system correctly predicts within a given rank, it does not focus at what rank. MRR plugs the gap here and incentivizes the system that predicts correctly at early ranks.

(c) Precision: Precision is the fraction of retrieved items that are relevant. In our context, it is the fraction of recommended venues that are relevant, as given below.

$$Precision = \frac{|\text{relevant items} \cap \text{recommended items}|}{\text{total number of recommended items}} \tag{2.20}$$

Precision@k means when k items are recommended, i.e.,

$$Precision@k = \frac{|\text{relevant items} \cap \text{recommended items}|}{k} \tag{2.21}$$

(d) Recall: It is another fundamental metric, and is the proportion of relevant items in the set of relevant items.

$$R = \frac{\text{relevant items retrieved}}{\text{number of relevant items}} \tag{2.22}$$

(e) $F - measure_{macro}$ (F_1): F_1 measure is defined as the balanced harmonic mean of precision and recall. Here we consider macro-averages for both precision and recall. The macro-average is the average of the same measures calculated for all classes. It treats all classes equally. For an individual class C_i (number of venues), if within-class true positives are tp_i , true negatives tn_i, false positives fp_i, and false negatives fn_i [144], then following are the definitions of necessary metrics.

$$Precision_{macro} = \frac{\sum_{i=1}^{N} \frac{tp_i}{tp_i+fp_i}}{N} \tag{2.23}$$

$$Recall_{macro} = \frac{\sum_{i=1}^{N} \frac{tp_i}{tp_i+fn_i}}{N} \tag{2.24}$$

$$F - measure_{macro} = \frac{2Precision_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}} \tag{2.25}$$

(f) Normalized Discounted Cumulative Gain (nDCG): It represents the ratio of discounted system gain and discounted ideal gain accumulated at a particular rank p, where gain at a rank p is the sum of relevance values from rank 1 to rank p [143]. Relevance value in our system (rel_{sj}) is a score (0, 1 or 2) assigned by a researcher to the venue at position j. Ideal vector is constructed hypothetically where all relevance scores (rel_{ij}) are ordered in decreasing order to ensure the highest gain at any rank.

$$DCG_{sp} = rel_{s1} + \sum_{j=2}^{p} \frac{rel_{sj}}{log_2(j)} \tag{2.26}$$

$$IDCG_p = rel_{i1} + \sum_{j=2}^{p} \frac{rel_{ij}}{log_2(j)} \tag{2.27}$$

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{2.28}$$

(g) Diversity (D): It is defined as the average dissimilarity (opposite of similarity) between all pairs of items in a result set [145, 146].

$$D = 2 * \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (1 - similarity(v_i, v_j))}{N(N-1)} \tag{2.29}$$

where N is the length of the recommendation, v_i and v_j are the venues appearing in the recommendation lists and Similarity(v_i, v_j) denotes the content (abstract, keywords) similarity among venues v_i and v_j.

(h) Stability: A recommender system is stable if the predictions do not change abruptly over a short period of time [42]. It is also called the mean absolute shift (MAS), designed to capture the internal consistency among predictions made by a given recommendation algorithm [3].

We adopt a two-phase approach to compute the stability of a recommendation algorithm. In phase 1, let P_1 be a set of recommendation based on training data R_1, where $P_1(u, i)$ represents a system-predicted rating for user u and item i. Then a set of hypothetical incoming rating is added to the original set of known ratings R_1. In phase 2, some subset S of predictions P_1 is added as the newly incoming known ratings. Thus, in phase 2, the set of known ratings becomes $R_2 = R_1 \cup$ S and the set of unknown ratings becomes $P_2 = P_1 \backslash S$. Based on R_2, predictions on unknown ratings P_2 are made. MAS is then defined as

$$Stability = MAS = \frac{1}{|P_2|} \sum_{(u,i) \in P_2} |P_2(u, i) - P_1(u, i)| \tag{2.30}$$

where P_1, P_2 are the predictions made in phase 1 and phase 2, respectively.

(i) Average-Venue Quality (Ave-quality): It evaluates the quality of the venues recommended by a system based on Google's h5-index [31].

$$\text{Average-venue quality} = \frac{\sum_{v \in V} H5_v}{|V|} \tag{2.31}$$

where V is the set of recommended venues and $H5_v$ is the h5-index of venue v. Higher the Ave-quality, we can claim, the better is the recommendation.

Precision captures the overall performance of the system in terms of how many relevant venues a system can recommend - a requirement often from a prospective researcher before sending her manuscript. However, precision only considers whether a venue is relevant or not. In reality, the relevance of a venue can be more fine-grained or graded like exactly relevant, partially or moderately relevant, not relevant, and so on. nDCG takes into consideration this subtlety and provides an idea of system performance with respect to an ideal system that ranks the recommendation in decreasing order of relevance. Both these metrics are bounded between 0 and 1 and are used for online evaluation.

2.7 Datasets

In this section, we present three datasets that are used in the experimental parts of this thesis. We used two datasets (MAG, DBLP) for journal recommendation tasks and two datasets (DBLP, hep-th) for the task of collaborator recommendation.

2.7.1 MAG (Microsoft Academic Graph)

Microsoft Academic Graph (MAG) dataset [44,147] is a heterogeneous dataset of scholarly publications publicly available and the most extensive dataset of open citation. It is currently being updated on a weekly basis. The dataset consists of various types of entities: publications, institutions (affiliations), authors, fields of study (FOS), venues (journals and conferences), events (specific conference instances), and the relations among these entities.

The dataset also contains metadata of the papers, such as title, DOI, and year of publication. It also has excellent coverage over various domains. All the FOS are constructed hierarchically into four levels (Level 0 to Level 3, with Level 3 being of the highest granularity). For our study, we use the version of MAG published on 5 February 2016. We process the dataset by retaining only papers related to the FOS of "Computer Science (CS)" and "Biology (BIO)" occurring at Level 0. For both, we have collected only the papers published in the year 1982-2016 (35 years' data).

Table 2.3: Statistics of both CS and BIO papers (Subset of MAG)

Type	No. of Records (CS)	Type	No. of Records (BIO)
Papers	15,641,658	Papers	14,785,486
Total FOS	14,417	Total FOS	10,522
FOS in Level 0	1 (CS)	FOS in Level 0	1 (Biology)
FOS in Level 1	35	FOS in Level 1	15
FOS in Level 2	685	FOS in Level 2	523
FOS in Level 3	13,696	FOS in Level 3	9,976

The major attributes of the dataset used are specified in Table 2.3. There are 13,696 fields of study (FOS) at Level 3 (e.g. "COBOL"), 685 at Level 2 (e.g. "Low-level programming language"), 35 at Level 1 (e.g. "programming language") and 1 at Level 0 (e.g. "Computer Science"). Similarly, in BIO there are 15 *FOS*s present at Level 1. The fields related to each other have a confidence score signifying relatedness among fields. The dataset does not have full-text or abstract of the publications. We have used a web-based crawler to extract the required set of abstracts by using the available title, year, URL, and DOI from the Web before applying abstract similarity.

2.7.2 DBLP-citation-network V10

We use a real-world dataset DBLP-citation-network V10 [8], the citation data extracted from DBLP, ACM, MAG (Microsoft academic graph), and other sources [148] to demonstrate the effectiveness of our proposed method. The tenth version contains 3,079,007 papers and 25,166,994 citations. Each paper is associated with abstract, authors, title, publishing year, venue, and references list. After removing duplicate papers, papers with missing fields, and inconsistent entries in the database, we are left with 2,236,968 papers.

We have used this dataset for the evaluation of both CNAVER (Chapter 4) as well as DRACoR (Chapter 6). Due to hardware constraints, only a subset of the original dataset is used for the experimentation of DRACoR. We performed our experiments on a subset of the dataset (data collected in the range 2000-2017) from Tang et al. [148]. In this work, we divided the dataset into two parts according to publication year: data during the years 2000-2012 as the training set and the rest as a testing set.

2.7.3 Hep-th (Theoretical High Energy Particle Physics)

The third dataset was hep-th (Theoretical High Energy Particle Physics) provided by KDD Cup 2003 [9]. After data preprocessing as above, we get 1,922 concurrent authors from 20,961 publications. Next, a modified heterogeneous network containing 2,395 terms, 12,018 cited papers, and 64 journals are constructed. We divided the dataset into two parts according to the year of publication: data before the year 1,999 as a training set and the rest as a testing set.

[8] https://aminer.org/citation

[9] https://www.cs.cornell.edu/projects/kddcup/datasets.html

2.8 Baseline Methods

In this section, we briefly discuss state-of-the-art techniques for both journal recommendation and collaborator recommendation respectively.

2.8.1 Baseline Methods for Journal Recommendation

To measure the performance of proposed journal recommender systems (DISCOVER, CNAVER, DeepRec), various state-of-the-art methods and freely available online services EJF and SJS are considered.

(a) Collaborative filtering models (CF): It is a memory-based implementation of collaborative filtering with a given paper-venue matrix. The underlying assumption is that there is a high probability for a paper to get published in venues where other similar papers have been published [60].

(b) Personal venue rating-based collaborative filtering models (PVR): It is based on the implicit rating given to individual venues, created from references of a researcher's publications and the papers which cited the researcher's past publications [36].

(c) Content-based filtering models (CBF): The main idea behind the approaches is to compute the similarity between researchers and venues. Here we have taken the researcher's publications and content of all publications at the venues as feature vectors computed by LDA model [49].

(d) Friend based model (FB): Friend based models recommend venues based on the number of neighbors like a researcher's co-author and co-author's co-author. If a venue is attached to many neighbors, the venue is recommended [149].

(e) Co-authorship network-based models (CN): This model creates a social network for each author and then recommends venues based on the reputation of the author's social network and other information such as venue name, venues sub-domain, number of publications [40].

(f) Random walk with restart models (RWR): It runs a random walk with restart model on a co-publication network with two types of nodes: authors and venues.

This model is similar to AVER, but the probability of skipping to the next neighbor node is equal in RWR [74].

(g) Hybrid approach (CF+CBF): We have mapped the citation web into a collaborative filtering rating matrix in such a way that a paper would represent a user, and a citation would represent an item. This method used an item-based collaborative filtering approach to identify a set of candidate papers in a given paper-citation matrix. Later on, we apply LDA on all extracted abstract, title to compute the similarity among a seed paper and candidate papers. Finally, the set of papers having high content similarity are identified, and their respective venues are recommended. We also tried to use the researcher-paper citation relationship to populate the rating matrix and other ways of combining CBF and CF, we chose the best performing method.

(h) Publication recommender system (PRS): It is based on a new content-based filtering (CBF) recommendation model using chi-square and softmax regression. It mainly consists of two modules, such as feature selection module and softmax regression module [33].

(i) Personalized academic venue recommendation models (PAVE): It is similar to the popular random walk model, however with additional feature of transfer matrix with bias. The probability of skipping to the next neighbor node is biased using co-publication frequency, relation weight, and the researcher's academic level in PAVE [31].

2.8.2 Baseline Methods for Collaborator Recommendation

To measure the effectiveness of the proposed system DRACoR, we compare our results with following state-of-the-art methods, as discussed below.

(a) CNRec: It is a common neighbors based recommendation model which is quite popular in recommendation based on social-networks. It is based on the assumption that if two researchers have a lot of co-authors in common, the high probability that they may collaborate in future [85].

(b) RWR: It involves a basic random walk with restart model on the whole co-author network to recommend collaborators. This model is similar to ACRec, but the probability of skipping to next neighbor node is equal in RWR [86].

(c) TBRec: This model is also called a content-based model that uses LDA on abstract and Doc2Vec on title in order to define the feature vector. This model is also a part of DRACoR without the inclusion of any other factors. We compute the content similarity among researchers by using Eqn. 6.14 to recommend personalized collaborators.

(d) MVCWalker: It is based on RWR-based recommendation, which uses three academic factors: co-author order, latest collaboration time, and times of collaboration within a co-author graph, to recommend personalized collaborators [25].

(e) CCRec: This model exploits both contents as well as the social network approach in order to recommend collaborators. They incorporate word2vec to identify the academic domains, as well as a random walk model to compute researchers' feature vectors [24].

(f) BCR: This model utilizes an integrated approach of three academic features: topic distribution of research interest, interest variation with time, and researchers' impact in collaborator network to provide beneficial collaborator recommendation [39].

(g) RWR-CR: This approach uses a heterogeneous bibliographic network with multiple types of nodes and links with a simplified network structure by removing the citing paper nodes. To weight edges in the network, both sequence importance and freshness importance are taken into consideration in order to bias the random walker's behaviors [54].