A novel content-based recommendation approach based on LDA topic modeling for literature recommendation

Mr. Dhiraj Vaibhav Bagul
School of Computer Engineering and Technology, MIT Academy of Engineering
Alandi(D), Pune – 412105, Maharashtra, India.
dvbaghul@mitaoe.ac.in

Dr. Sunita Barve
School of Computer Engineering and Technology, MIT Academy of Engineering
Alandi(D), Pune – 412105, Maharashtra, India.
ssbarve@comp.mitaoe.ac.in

***Abstract*—In an application such as literature recommendation, we require a comprehensive recommender model that can generate relevant recommendations similar to the literature provided in the input query. In this paper, we have proposed a novel content-based recommender system based on Latent Dirichlet Allocation (LDA) and Jensen-Shannon distance, which can be used specifically for the task of literature recommendations. We have compared this model with the standard cosine-similarity based approach for its use to generate scientific publication recommendations, in which recommend suitable journals/conferences to publish a research work based on the abstract of the user's manuscript as an input. We evaluated the results of both the proposed model and standard cosine-similarity based approach over unseen documents and achieved a precision score of 62.58% while the standard cosine-similarity based approach achieved a precision of only around 48%.**

Keywords—Recommender Systems, literature recommendation, content-based recommendation, Natural Language Processing, Topic modeling.

I. INTRODUCTION

Due to its advantages in providing a customized user experience, recommender systems have become an integral part of many online platforms such as Amazon, YouTube, Spotify, Netflix, etc. Many of the potential applications of recommender systems include Music Recommendations [1-2], Movies Recommendations [3], Book Recommendations [4-5], etc.

In the recent period, some research to develop recommender systems for literature recommendations [6- 7] has been carried out. Models based on existing approaches in the Content-Based recommendation and novel approaches involving topic modeling and deep learning are proposed to develop a literature recommendation system [7-8]. One of the issues in some of these models lies in the ability to scale. These models fail to scale with the growing size of the corpus which makes these models ineffective over time. Another issue with most of these proposed approaches is comprehensiveness. Most models fail to generate precise recommendations on a diverse corpus, and some of them fail to capture the complex topical relationship present in the corpus. For the case of literature recommendations, we require a model that can scale at large and perform well on a broader range of domains. Considering this, we began working with the objective of developing a recommender model that will be apt for the application of literature recommendations and has the ability to capture subtle topical relationships thereby addressing the issue of low precision over a diverse corpus. In this paper, we propose a novel content-based recommendation system based on LDA topic modeling and Jensen-Shannon distance for the task of literature recommendations. We also test this proposed model by its use for the task of generating recommendations of scientific publications to publish the manuscript by using the abstract of the manuscript as input from the user. We have developed a dataset for the same by web-crawling and evaluated the results generated by the proposed model over both seen and unseen manuscripts. We also compare the results of our model with the results of the traditional cosine-similarity based recommendation approach for generating *top-n* recommendations for unseen documents.

The rest of this article is organized as follows: Section 2 discusses the recent developments and recommender systems' applications. Section 3 explains our proposed LDA-based content recommender model, followed by experimental analysis and results in section 4 and the conclusion in section 5.

II. BACKGROUND

Most of the recommendation systems are implemented using either Content-Based filtering or collaborative filtering. Content-based recommender systems recommend a certain Content-Based on similarity with other content that other users liked in the past [4]. Collaborative filtering provides a recommendation based on the similarity between two or more users. Collaborative filtering filters out items that the current user might like based on reactions given by similar users. It selects the most similar users to the current user and recommends the content liked by or viewed by these similar users [9]. In X. Wu et al. [10], a novel similarity calculation

method is proposed for collaborative filtering recommendation. The similarity between the users and items is calculated using the overlap of items used by two users and the overlap of the similar items rated by the same users. Item similarity is calculated similarly.

In recent years, many applications of recommender systems have been proposed and deployed. Pradeep Kumar Roy et al. in [11] proposed a Resume recommendation using classification and matching is for a given job description. TF-IDF is used for feature extraction from a candidate resume. These extracted features are used to predict the category of resume using four different classifiers, out of which Linear Support Vector Machine is more effective in correctly capturing the resume insights as per the job description. The recommendation of resumes matching closest to the job description is ranked using two approaches, Content-Based recommendation using Cosine-similarity and k-Nearest Neighbor. The performance of the machine learning model is restricted to 78.53% accuracy for classification. In 2016, Achakulvisut T et al. proposed a Content-Based filtering approach to provide recommendations of relevant scientific papers [12]. It is designed to give feedback as fast as possible by providing a vote for a single document for its relevance and irrelevance using Rocchio Algorithm and Nearest Neighbor Assignment. Terms are weighted considering each document's local structure using different weighting schemes like simple term frequency, term frequency and inverse document frequency (TF-IDF), and log-entropy. Donghui Wang et al. [8] proposed a content-based publication recommender for the computer science domain using chi-square feature selection and Softmax regression. Chi-square statistics are measured for each unique term and then sorted to select feature vector. Softmax regression is used as a classifier to provide three class recommendation. In addition to Chi-Square, the performance of recommendation is verified using Mutual Information and Information Gain. Chi-Square outperforms the Mutual Information and Information Gain feature selection model with an accuracy of 61.37%.

In recent research, topic model based techniques are used to perform document classification and develop recommender systems. Jobin Wilson et al. in [13] proposed a collaborative filtering model based on topic modeling to improve the performance of the collaborative filtering. The feature vectors are represented using the document-topic probability distribution of corpus item documents using Latent Dirichlet Allocation (LDA). The similarity score between the users is calculated using latent topic space of the item documents and rating overlap based similarity. This hybrid recommendation using LDA refined the neighborhood formation for improving the quality of recommendation for contextual data in text form. Z. Li et al. in [14] proposed a news text classification model. This model uses latent topic information of the news to reduce the high dimensional space and select the relevant features in the new text. The Latent Dirichlet Allocation based probabilistic topic model not only discover the topic category to which the news text belongs but extend it to multi-class classification by softmax regression algorithm. Kunlun Li et al. in [15] used LDA and SVM for Multi calls text categorization. Underlying semantic Structures are discovered using Latent Dirichlet Allocation (LDA) for improving the text performance of multi-class text categorization. The LDA based feature selection identifies the semantic structure and integrates relation between the feature words. The probabilistic distribution of the feature word is used to create a mapping of text to a topic. This Topic-Text mapping is used to train the support vector machine for text classification.

LDA topic modeling and Jensen-Shannon divergence have been used for various tasks such as Text mining and calculating text similarity. In [16], Zhou Tong et al. demonstrated the use of LDA topic modeling to generate topic distributions of Wikipedia articles and using Jensen-Shannon divergence to calculate the similarity between these distributions. Minglai Shao et al. in [17], have proposed a novel algorithm to computer similarity between texts. This algorithm uses LDA topic modeling for the purpose of text clustering and generating probability distributions, and to compute the similarity between any two texts by finding semantic relation through analyzing word co-occurrence of these texts. Masaki Uto et al. in [18] have developed a reports recommendation system to enhance the writing skills of learners by recommending reports that are similar to the learner's current subject at hand. LDA topic modeling is used to model previous reports and Jensen-Shannon distance to compare topic models of these reports and generate recommendations.

III. METHODOLOGY

The proposed recommender system consists of three different algorithmic stages. First is preprocessing and vector representation of documents. Then, create a topic model based on LDA, and at last, a recommendation module will provide recommendations based on Jensen-Shannon distance [19]. Figure 1 depicts the flow of the proposed recommendation model.

A. Data preprocessing and vector representation

In this stage, we use different Natural Language Processing techniques to perform text preprocessing and feature weighing schemes to generate a vector model for each of the documents. First, we remove stop words from the documents then apply the Porter stemming algorithm [20] to stem the words. Now that we have cleaned the document, we use frequency-based embedding techniques to generate a vector representation for the documents.

1) Count-vector:

Consider a corpus C contains documents $\{d1, d2, d3 \ldots, dp.\}$ and $N\{n1, n2, n3, \ldots, nq\}$ is a set of unique tokens where each token occurs in at least one document in C, then count-vector representation for the corpus is a matrix of

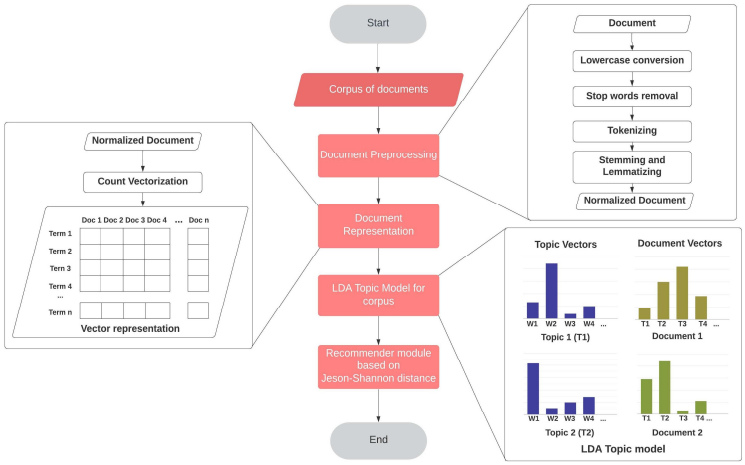


Fig. 1. Flowchart of proposed recommendation approach

dimensions $p \times q$ where row i represents the frequency of a particular token in a document d_i.

2) TF-IDF Vectorization:

TF-IDF reflects how important a word or a phrase is to a document in a set of documents [21]. TF-IDF prepares feature vectors for the set of text documents, which can then be used to do further text analysis. For document d, the term frequency of a word w (or a phrase) is the number of times the word w occurred in document d. It is normalized in order to avoid hassle over a considerable length of documents.

$$tf(w,d) = T/L \tag{1}$$

Here, T is the frequency of a word w in d, and L is the number of unique words in d. Inverse document frequency of a word or a phrase is a measure of how much information it provides about a document d. Consider a word w occurs in D_i number of documents, and D is the total number of documents on the logarithmic scale, then idf of a word w is:

$$idf(w,D) = \log \frac{D}{(D_i+1)} \tag{2}$$

Now,

$$tf - idf_{(w,d)} = tf(w,d) \times idf(w,D) \tag{3}$$

We can see that $tf - idf$ of a word w is directly proportional to the term frequency in document d and inversely proportional to the frequency over documents i.e. D_i. From this, we can infer that if a particular word is infrequent in the corpus but has high occurrence inside a particular document in a corpus then it will have a $tf - idf$ higher value and it will weigh more in the feature vector for that particular document. In contrast, for a word that is frequent in both corpus and document, it will weigh less in feature vector for the documents in which it occurs. These facts can be used to reduce the dimensionality of the document-term matrix by eliminating terms that have a lower contribution to the overall corpus.

For preparing a topic model LDA requires a bag-of-words representation of each document present in a corpus and a dictionary for all unique terms occurring at least once throughout the corpus. We can obtain both the bag-of-words representations and dictionary through the above-discussed embedding techniques. The major difference between the results of these two techniques lies in the dimensionality of the generated results. Count-vector representation does not take into account the occurrence of a term throughout the corpus. This may assign more weight to such terms thereby increasing the unique terms for a corpus and hence the dimensionality. On the other hand, as $tf - idf$ weighs the occurrence of a term throughout the corpus it provides the ability to reduce the dimensionality by eliminating such terms with less information to provide about a particular document. Both of these term weighting schemes are commonly used, and we generate a topic model by using both of them. At last, we compare the results and propose a model with a term-weighing scheme that provides better results.

B. *Preparing Topic Model with Latent Dirichlet Allocation*

In this section, we will introduce the Latent Dirichlet Allocation (LDA), which is a non-linear topic modeling technique that we have used for the proposed recommendation system approach.

LDA is a widely used non-linear topic modeling technique proposed by David Blie, et al. in [22]. It is a three-layer Bayesian probability model composed of document, topic, and words. LDA works under the assumption that words occurring in documents carry out important semantic information and the set of documents that are on a similar topic will contain the same set of words. Based on this idea, latent topics are discovered by finding a group of words from the corpus which occur frequently in documents. Furthermore, each document

from the corpus is modeled as a finite mixture of a set of these underlying topics.

For the case of literature recommendations systems, we expect the recommendations to be content similar, and hence it is safe to assume that literature on the same topic will roughly use a similar set of words. Thus, we can utilize the LDA topic modeling technique to group literature documents that are on similar topics.

LDA follows a generative approach where words in documents are generated through a stochastic process. Given a corpus with bag-of-words representation and a vocabulary *V* for documents in the corpus, hidden topics for these documents are discovered as distributions of the words in the vocabulary [23]. These topics are modeled as a latent random variable and the words are modeled as an observed random variable. The joint distribution for it can be defined after establishing this generative process and later probability distributions over the latent variables that are conditioned with the observed variables can be computed through statistical inference. Figure 2, depicts the graphical plate representation of the generative LDA model for the entire corpus. The documents in the corpus are modeled as Dirichlet prior distribution of the latent topics and these latent topics are modeled as polynomial distribution of the words. Consider θ as a set of joint distribution for a set of *D* documents. Then, the Dirichlet distribution is given by:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta^{\alpha_i - 1} \quad (4)$$

And, the joint distribution over θ and the set of *N* topics *Z* is given by:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \quad (5)$$

Here, α is *K*-dimensional hyperparameter, and it stays constant for all the documents present in *D*. β is also a multinomial hyperparameter used in generating word distributions over all the vocabulary words in *V*. To understand the learning procedure of this hyperparameter in-depth please refer to the original paper [22].

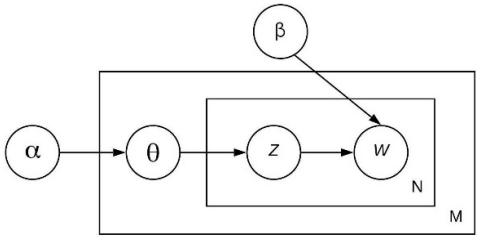


Fig. 2. Probabilistic graphical model of LDA

1) Generating probability distributions for training data with LDA:

We obtain the Vocabulary and bag-of-words representations for the documents in the corpus at the end of the preprocessing stage. Both of these are then used to train the LDA and generate a topic model based on which we can obtain topic distribution for each of the input documents, i.e., scientific publications. To obtain a fine-tuned model with the highest possible precision, we use a set of topics and different hyperparameter values to generate a topic model. In Algorithm 1, we depict the generative process that is used to produce a probabilistic topic model for a corpus D consisting of M unique publications with each length $\boldsymbol{N_i}$.

Algorithm 1 Generative process to develop a topic model

Input: *D*: Corpus *M* of documents in sparse-matrix representation
Output: $\boldsymbol{\theta_C}$: Topic distribution for documents in corpus *D*
$\boldsymbol{\varphi_K}$: Word distribution for each topic $\boldsymbol{k\ in\ K}$

1: For each i in *M*: Choose $\boldsymbol{\theta_i \sim Dir(\alpha)}$
2: For each $\boldsymbol{k\ in\ K(set\ of\ topics)}$: Choose $\boldsymbol{\varphi_k \sim Dir(\beta)}$.
3: For each of the word positions i, j, where i $\in \{\boldsymbol{1, \dots, M}\}, \boldsymbol{and\ j} \in \{\boldsymbol{1, \dots, N_i}\}$:
(1) Choose a topic $\boldsymbol{z_{i,j} \sim Multinomial(\theta_i)}$.
(2) Choose a topic $\boldsymbol{w_{i,j} \sim Multinomial\left(\varphi_{z_{i,j}}\right)}$
4: return $(\boldsymbol{\theta_C}, \boldsymbol{\varphi_K})$

Here,
M denotes the number of documents
N is the number of words in a given document (document i has $\boldsymbol{N_i}$ words)
α is the parameter of the Dirichlet-prior on the per-document topic distributions
β is the parameter of the Dirichlet-prior on the per-topic word distribution
$\boldsymbol{\theta_i}$ is the topic distribution for a document i
$\boldsymbol{\varphi_k}$ is the word distribution for the topic $\boldsymbol{k}$
$\boldsymbol{z_{i,j}}$ is the topic for the j[th] word in document i

C. Recommendation stage based on Jensen-Shannon distance

In this section, we will introduce the recommendation stage of the proposed model, which uses the Jensen-Shannon distance to generate the recommendation results for the given input.

1) Jensen-Shannon distance:

Jensen-Shannon divergence (JSD) is a Kullback-Leibler divergence based symmetrical metric to measures the "difference" between two probability distributions. Furthermore, the square root of the Jensen-Shannon divergence is Jensen-Shannon distance [19].

At the end of the LDA topic modeling stage, we have a set of two probability distributions, mainly topic distributions and word distributions for all the documents in the training corpus. These are essentially the documents that we want to recommend to the user based on the provided text input. We can obtain the join distribution for the text input provided by the user, and we can prepare the content-similar recommendation for the input by comparing this newly generated joint distribution with the joint distributions of all the documents in the corpus.

This task requires a metric that can evaluate the similarity between two probability distributions. Jensen-Shannon distance is one such metric that is useful to compare two probability distributions. Given the two probability distributions, Jensen-

Shannon distance provides a large positive value to indicate the extent to which the two provided probability distributions differ from each other. The larger the value less similar will bet the content represented by these distributions. The smaller positive value indicates that the two probability distributions are consistent with each other and, hence, have potentially similar content. Consider two discrete probability distributions P and Q, then Jensen-Shannon Divergence of Q from P is defined as:

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M) \quad (6)$$

Where,

$$M = \frac{1}{2}(P + Q) \quad (7)$$

And, D is the Kullback-Leibler divergence of probability distribution P from the probability distribution Q, which is given by:

$$D(P||Q) = \sum_i P(i)\log(\frac{P(i)}{Q(i)}) \quad (8)$$

Now, the Jensen-Shannon distance for P from Q can be obtained as,

$$JSD_{dist} = \sqrt{JSD(P||Q)} \quad (9)$$

2) Getting publication recommendations:

Algorithm 2 depicts the steps to generate *top-10* recommendations of the scientific publications. Input to this stage is textual data for which we want to find recommendations and the topic model generated for training corpus. In case of publication recommendation, the user will provide the abstract of the manuscript as text input and we will have a topic model for the training corpus containing each publication as one document.

Upon receiving text input from the user, we generate a joint probability distribution for it as per our training corpus. Afterward, the similarity between each of the documents present in the corpus and the provided input is calculated by using Jensen-Shannon distance. We store these results to identify documents that have the least Jensen-Shannon distance from the provided input data and are most similar to the provided content. By sorting these obtained results we recommend the *top-10* documents for the given input in order of most similar to least similar.

Algorithm 2 Recommending publication for the manuscript

Input: A: Abstract of the manuscript,
θ_C: Set of topic distribution for publications in corpus C
Output: *Top-10* recommendations of the scientific publications

1: Choose $\theta_A \sim Dir(\alpha)$ for A
2: pScores = [] (an empty 2-D array to maintain the JSD obtained of the abstract from each document.)
3: for each θ_p publication θ_C:
distance = JSD(θ_p, θ_A) (calculating Jensen-Shannon distance between of θ_A from θ_p)
append(p, distance) to pScores
end for
4: sort (pScores, ascending)
5: return (pScores[:10])

IV. Experimental Results and Analysis

In this section, we discuss the development and evaluation of our proposed recommendation approach for the application of publication recommendation to publish research work. We also compare the recommendation results obtained for the proposed system with the recommendations of the cosine-similarity based approach. Figure 3 is the block diagram of the proposed system, which depicts different modules involved in the proposed model and its relation.

A. Dataset

We developed a corpus of 100 publications from 10 different domains for our recommendation system. To represent the content of these publications in our corpus we used the abstracts and other metadata of previously published research papers in that particular corpus. We obtained the abstracts and other metadata of papers published in various scientific publications are through a web crawler for training purposes. At first, we scraped the SCIMAGO website to collect various scientific publications' titles. Figure 4 shows us a snippet of a few of the publications that are present in the corpus and their domain represented by the domain code. These gathered titles of around 100 publications from 10 different domains were then used to get the metadata of papers published in that particular publication. By using *arcas* library in python, we collected the metadata of around 500 papers for each of the publications. Out of which, 90% of the papers were used for generating the

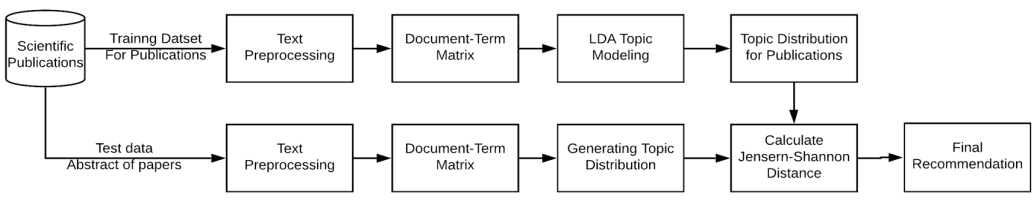


Fig. 3. Block diagram for scientific publication recommendation system

training corpus, and the remaining 10% were used to perform the evaluation. In Figure 5, we can see a snippet of titles and metadata of some of the publications present in the corpus.

	Name	Domain Code	Type
0	Annual Review of Plant Biology	1100	Journal
1	Genome Biology	1100	Journal
2	Annual Review of Pathology: Mechanisms of Disease	1100	Journal
3	Molecular Biology and Evolution	1100	Journal
4	Trends in Ecology and Evolution	1100	Journal

Fig. 5. Snippet of dataset containing all publications over all domains

	title	raw_data
0	Annual Review of Plant Biology	Green Algae as Model Organisms for Biological...
1	Genome Biology	Mass-spectrometry of single mammalian cells q...
2	Annual Review of Pathology: Mechanisms of Disease	Extracting Angina Symptoms from Clinical Note...
3	Molecular Biology and Evolution	Selscan: an efficient multi-threaded program ...
4	Trends in Ecology and Evolution	Stochastic models of population extinction ...

Fig. 4. Snippet of dataset of publications and crawled metadata

B. Text Preprocessing

Text preprocessing is an essential part of development, with which we can improve the feature space for our training set and hence, the performance. Our dataset contains a corpus of documents that are in a raw text format. Before preparing the LDA topic model, we need to generate bag-of-words representation and the Vocabulary of unique words. Hence, we follow standard text-processing steps such as tokenization, stemming, stop-words removal, and, at last, we use both count-vector representation and TF-IDF vector representations techniques to generate bag-of-words and Vocabulary of unique words. The Vocabulary generated from both these algorithms is of different size as the algorithmic approaches of both are different. We compare the LDA topic model generate through both techniques in the upcoming section.

C. Implementing LDA based content recommender

Now that we have bag-of-words representation and vocabulary, we use the genism library in python [24], to generate the LDA topic model for the publications through the metadata of previously published research papers. With this obtained LDA-model for the corpus of publications, we prepare a recommendation function that takes the text input from the user which can be an abstract of the manuscript for which a user wants to find out the publication which is suitable to publish the same manuscript. For this text input, the function then generates joint distribution. After that, we use Jensen-Shannon as a metric to determine *top-10* publications from the corpus, similar to the provided text input.

D. Experimental results

As discussed earlier, some of the previous recommendation systems fail to capture subtle topical relationships and hence perform poorly on a diverse corpus. In our experiment, we tested the LDA-based content recommender by evaluating its precision score for recommendations generated over a diverse corpus. We evaluated its overall precision score on seen documents as well as unseen documents. We consider the recommendation provided by the model successful recommendation if the actual publication for that paper appears in the *top-10* recommendations. In every other case, we regard the generated recommendations as unsuccessful. Based on this, the precision score for the system can be defined as:

$$Precision = \frac{Total\ sucessful\ recommendation}{Total\ provided\ recommendation} \quad (10)$$

In this section, at first, we compare precision scores on both seen and unseen documents of our proposed model developed by use of two different vector representation techniques i.e. TF-IDF vectorization and count-vector representation. Furthermore, we evaluate the performance of both these models with a cosine-similarity based recommendation approach for unseen documents. At last, we also evaluate the performance of our count-vector based LDA model for each of the domains present in the corpus.

1) Precision score for count-vector based and TF-IDF vector based proposed model:

For our proposed recommendation system we use both count-vector representation and TF-IDF vector representation to generate the bag-of-words for the documents in the corpus and to generate the vocabulary of unique words in the corpus. As the TF-IDF algorithm weighs down the terms that are frequent in the overall corpus it provides us the opportunity to reduce the overall unique terms present in the vocabulary by setting extremes. Due to this, the output generated from the TF-IDF technique contains fewer vocabulary terms compared to vocabulary generated through count-vector representation. After developing the model by using both these techniques, we can see their performance over unseen and seen documents compared in Figures 6 & 7 to generate the *top-10* recommendations. In order to generate an optimized model, we experimented with the number of topics, and model hyperparameters α and β and evaluated the performance generated for both seen and unseen documents.

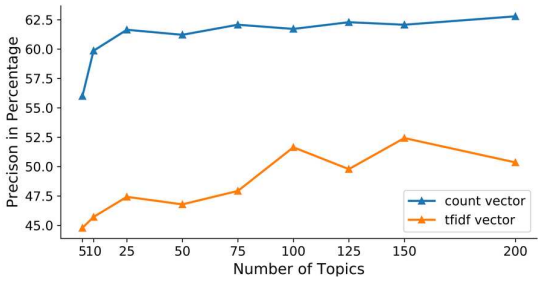


Fig. 6 Precision score for unseen documents over varied topics

In both cases, we can clearly see that the model that uses the count-vector representation to generate bag-of-words representation outperforms the model generated by using the TF-IDF technique. For both unseen and seen documents, we

TABLE I. PRECISION SCORE ACHIEVED BY COUNT-VECTOR BASED LDA MODEL FOR EACH DOMAIN

Domain Name	Total test sample	Relevant recommendations obtained	Precision score
Computer Science	990	686	0.693
Agriculture and Biological Sciences	870	486	0.558
Engineering	940	478	0.508
Earth and Planetary Sciences	760	540	0.71
Chemistry	990	618	0.624
Environmental Science	730	438	0.6
Business, Management	920	592	0.643
Mathematics	1000	580	0.58
Physics and Astronomy	980	669	0.682
Social Sciences	1000	660	0.66

can see that there is a positive trend between precision score and number of topics for the count-vector based model, after which there is a slight decrease in precision score. In the case of TF-IDF based model, we can see an increasing trend in the precision score, but it is not consistent compared to the count-vector-based model. The highest precision score achieved by the TF-IDF based model is slightly more than 52% and that of the count-vector based model is around 62.5% for the unseen manuscripts. For seen documents, TF-IDF based model has the highest precision score slightly below 60%, and that for the count-vector based model is around 68%. From these results, we can conclude that the count-vector based LDA model is

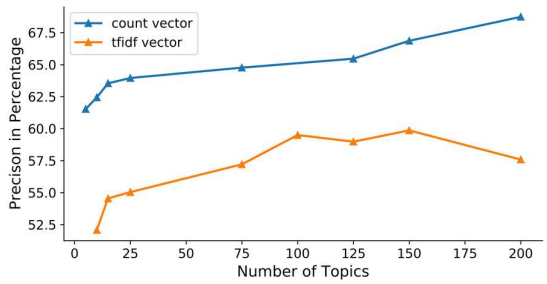


Fig. 7 Precision score for seen documents over varied topics

more suited for the case of literature recommendation if compared to the TF-IDF based model.

2) Comparioson between the proposed model and cosine-similarity based model:

Our main objective of the research was to prepare a model that can perform better on a diverse corpus when compared to previous approaches. To evaluate this objective, we compare the recommendation results of the proposed model with the results of the standard cosine-similarity based approach. For the same corpus, we have developed a TF-IDF based cosine-similarity model and we compare its recommendation results over unseen documents with both TF-IDF based and count-vector LDA model. In Figure 8, we can see the comparison between all three approaches to generate *top-10, 15, 20, 25, and 30* recommendations for unseen documents. For each case, our proposed approach outperforms the cosine-similarity based model. The highest precision score amongst all three approaches is achieved by the count-vector based LDA model.

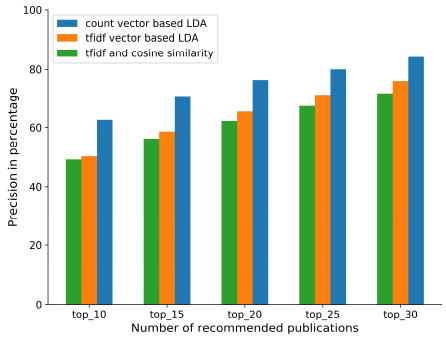


Fig. 8 Comparison between three approaches over unseen documents

3) Performance of count-vector based model over each domain:

As stated earlier, we gathered publications from 10 different domains ranging from Biochemistry to Engineering. This will help us to evaluate the proposed model's performance for a specific domain. Through Table 1, we can see the precision score of our model over each domain present in the corpus. It can be inferred that for each of the domains, at least half of the generated recommendations are successful. The model provides the best recommendation results over unseen papers for publications in the domain of Earth and Planetary Sciences and performs weekly in case of unseen papers from the engineering domain. This variation can be a result of the quality of content that is present in the corpus. With content that is poor to be generalized, the model performs poorly over unseen documents and performs quite well over the content that is best suitable for generalization. Hence, to develop a model that provides better recommendations it is needed to analyze the quality of the training data used to prepare the model.

V. CONCLUSION

In this paper, we proposed a novel content-based recommendation approach based on LDA topic modeling and Jensen-Shannon distance. We also developed a scientific publication recommendation system where we take the abstract

of the manuscript from the user as an input and provide *top-10* publications suitable for publishing the manuscript based on content similarity. We received the highest precision score of 62.58% for unseen data in *top-10* recommendations for our proposed LDA model with count-vector representation. Our test results show that the model can capture subtle topical relationship over a diverse corpus and perform better over the cosine-similarity based approach. Hence, this model can be used for applications in literature recommendations where we expect the system to generalize over a large and diverse corpus to provide suitable recommendations over both seen and unseen documents. In future work, we will focus on improving the system's performance potentially by using different similarity measurement metrics than the Jensen-Shannon distance.

REFERENCES

[1] M. Schedl, D. Hauger, "Tailoring music recommendations to users by considering diversity, mainstreaminess, and novelty," In Proceedings of the 38th international acm sigir conference on research and development in information retrieval, August 2015, pp. 947-950.

[2] M. Kaminskas, F. Ricci, M. Schedl, "Location-aware music recommendation using auto-tagging and hybrid matching," In Proceedings of the 7th ACM conference on Recommender systems, October 2013, pp. 17-24.

[3] CA Gomez-Uribe, N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," ACM Transactions on Management Information Systems (TMIS), December 2015, 6(4):1-9.

[4] RJ Mooney, L. Roy, "Content-based book recommending using learning for text categorization," In Proceedings of the fifth ACM conference on Digital libraries, June 2000, pp. 195-204.

[5] CM. Chen, "An intelligent mobile location-aware book recommendation system that enhances problem-based learning in libraries," Interactive Learning Environments, October 2013, 21(5):469-95.

[6] BG Patra, V Maroufy, B Soltanalizadeh, N Deng, WJ Zheng, K Roberts, H. Wu, "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, March 2020, 103399.

[7] C. Wang, DM Blei., "Collaborative topic modeling for recommending scientific articles," In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, August 2011, pp. 448-456.

[8] D Wang, Y Liang, D Xu, X Feng, R. Guan, "A content-based recommender system for computer science publications," Knowledge-Based Systems, October 2018, 157:1-9.

[9] G Linden, B Smith, J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet computing, January 2003, 7(1):76-80.

[10] X Wu, Y Huang, S. Wang, "A New Similarity Computation Method in Collaborative Filtering based Recommendation System," In 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), September 2017, pp. 1-5.

[11] PK Roy, SS Chowdhary, R. Bhatia, "A Machine Learning approach for automation of Resume Recommendation system," Procedia Computer Science, Jan 2020, 167:2318-27.

[12] T Achakulvisut, DE Acuna, T Ruangrong, K Kording, "Science Concierge: A fast content-based recommendation system for scientific publications," PloS one, July 2016, 11(7):e0158423.

[13] J Wilson, S Chaudhury, B. Lall, "Improving collaborative filtering based recommenders using topic modeling," In 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), August 2014, Vol. 1, pp. 340-346.

[14] Z Li, W Shang, M. Yan, "News text classification model based on topic model," In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), June 2016, pp. 1-5.

[15] K Li, J Xie, X Sun, Y Ma, H. Bai, "Multi-class text categorization based on LDA and SVM," Procedia Engineering, January 2011, 15: 1963-7.

[16] Z Tong, H. Zhang, "A text mining research based on LDA topic modelling," In International Conference on Computer Science, Engineering and Information Technology, May 2016, pp. 201-210.

[17] M Shao, L. Qin, "Text similarity computing based on LDA topic model and word co-occurrence," In 2014 2nd International Conference on Software Engineering, Knowledge Engineering and Information Engineering (SEKEIE 2014), March 2014, Atlantis Press, pp. 193-203.

[18] M Uto, S Louvigné, Y Kato, T Ishii, Y Miyazawa, "Diverse reports recommendation system based on latent Dirichlet allocation" Behaviormetrika, July 2017, 44(2):425-44.

[19] CD Manning, H. Schütze, "Foundations of statistical natural language processing," MIT press; May 1999.

[20] MF. Porter, "An algorithm for suffix stripping," Program, July 1980, 14(3):130-7.

[21] CD Manning, H. Schütze, P. Raghavan, "Introduction to information retrieval," Cambridge university press, 2008.

[22] DM Blei, AY Ng, MI. Jordan, "Latent dirichlet allocation," Journal of machine learning research, Jan 2003, 993-1022.

[23] DJ. Hu, "Latent dirichlet allocation for text, images, and music," University of California, San Diego. Retrieved April. 2009;26:2013.

[24] R Řehůřek, P. Sojka, "Gensim—statistical semantics in python" Retrieved from genism. org. August 2011.

[25] Shakya P, Shakya S. Critical Success Factor of Agile Methodology in Software Industry of Nepal. Journal of Information Technology. 2020;2(03):135-43.

[26] Raj JS. Machine Learning Implementation in Cognitive Radio Networks with Game-Theory Technique. Journal: IRO Journal on Sustainable Wireless Systems June. 2020;2020(2):68-75.