IJAMEC
e-ISSN: 2147-8228

INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS ELECTRONICS AND COMPUTERS

www.dergipark.org.tr/ijamec

International Open Access

Volume 11
Issue 01

March, 2023

Research Article

A Novel Article Recommendation System Empowered by the Hybrid Combinations of Content-Based State-of-the-Art Methods

İlya KUŞ [a,b], ***Sinem BOZKURT KESER*** [b,*], ***Savaş OKYAY*** [b]

[a]*Karamanoğlu Mehmetbey University, Department of Computer Engineering, Karaman, Türkiye*
[b]*Eskişehir Osmangazi University, Department of Computer Engineering, Eskişehir, Türkiye*

ARTICLE INFO

Article history:
Received 6 November 2022
Accepted 9 March 2023

Keywords:
Content-based filtering
Latent dirichlet allocation
Recommender system
Word embedding algorithm

ABSTRACT

The initial literature reviewing step is of great importance during any scientific reporting. Nevertheless, finding relevant papers grows tough as the number of online scientific publications rapidly increases. Correspondingly, the need for article recommendation systems has emerged, which aim to recommend new papers suitable for the researchers' interests. Using these systems provides researchers access to related publications quickly and effectively. In this study, a novel article recommendation system, which is empowered by the hybrid combinations of content-based state-of-the-art methods, is proposed. Various methods are utilized comparatively for an in-depth analysis, and user profiles are evaluated. 41,000 articles collected from the ARXIV dataset are used in the performance evaluation. In the experiments in which Word2vec and LDA are combined, Precision@50, Recall@50, and F1-score@50 achieve the highest performance with .206, .791, and .498 values, respectively. The in-depth analysis and the numerical findings justify that the proposed system is strong and promising compared to the literature.

1. Introduction

With the advancements in information technologies and scientific developments, recommendation systems have been developed rapidly in every field. These include movies [1], news [2], music [3], books [4], and webpage recommendations [5]. In addition to these systems, article recommendation systems are also becoming popular in scientific development. Due to the increased amount of information in the literature, and the inability to follow, it is getting more difficult for researchers to get accurate information. Therefore, various article recommendation systems are developed to lessen the time spent on finding relevant articles according to researchers' interests. These systems are of immense importance to broaden the horizons of young researchers with limited or no experience in publication and to reach information quickly according to their research interests. Moreover, they offer articles that correspond with the research interests of experienced academics with publishing records and without wasting time.

In the literature, recommendation systems generally use three methods: content-based, collaborative filtering, and hybrid recommendation systems. Content-based systems are developed with user ratings or data provided by clicking a link. Researchers are making recommendations with the user profiles they create using this data [7]. As the amount of information obtained by users increases, the accuracy of the results also increases. Content-based recommendation systems are the most appropriate methods to be used in cases where the user and item information is limited or unavailable. On the other hand, the collaborative filtering method performs the operations on the given ratings while creating the user profile. In this process, the items with maximum similarities are recommended to the user. Hybrid recommender systems are created by combining several recommendation techniques [6]. Further, one content-based technique may include more than one method, thereby constructing a hybrid combination [20].

In this study, a novel recommendation system is developed by combining word embedding algorithms of

* Corresponding author. E-mail address: sbozkurt@ogu.edu.tr
DOI: 10.18100/ijamec.1199886

content-based stated-of-the-art-methods. Comparative analyses are made by using user profiles during the evaluation phase. In the proposed system, title and summary information of the articles are extracted from the ARXIV dataset. The research areas of the articles are used to create user profiles that check the accuracy of the recommendation lists. The ARXIV dataset doesn't include information that can be interacted with, such as user ratings, historical profile information, or the number of clicks. Therefore, we have inferred the relationship between users and articles without considering the user's past actions. Thus, the accuracy of the proposed system is calculated according to the interaction of the article's research area. This way, article-research field interactions are created for each article and profile. Experiments are performed in three steps. In the first step, the TF-IDF (*Term Frequency–Inverse Document Frequency*) and LDA (*Latent Dirichlet Allocation*) techniques are combined on a method basis. In the second step, the Word2vec and LDA techniques are combined on a method basis. Then, The Word2vec and LDA algorithms are brought together methodically in the next stage. Finally, the performance of the two hybrid methods is compared by using Precision, Recall, and F1-score a in the offline evaluation. As a result, a comparative analysis of the recommended list is done in accordance with the knowledge of the articles' research area.

The highlights and the contributions of our work are as follows:

- We construct a novel article recommendation system empowered by hybrid combinations of state-of-the-art methods for the literature recommendations task.
- Paper profiles are created by extracting the information on the articles' research areas from the ARXIV dataset. The title and summary information of the article and the research areas in the user profile are utilized in the proposed system.
- Various hybrid perspectives comparatively take action: We combined the Word2vec word embedding algorithm with the LDA technique. Then, we combined the TF-IDF algorithm and the LDA technique.
- The user profile is used only in the evaluation phase of the proposed system. The user profile contains the article's research area information.
- The comparative results demonstrate the effectiveness of the proposed hybrid method for paper recommendations.

The remainder of this paper is organized as follows. Relevant studies in the literature are briefly discussed in Section 2. In Section 3, we present the architecture of the methodology is presented in detail. The dataset, evaluation metrics, and the basic steps of the architecture are given in this section. The evaluation results are reported in Section 4. Finally we conclude the study by summarizing the conclusions and thinking about the future work ahead in section 5.

2. Related Work

Scientific paper recommendation systems make recommendations in line with the request of researchers. The several techniques used in these systems can be categorized as follows: content-based filtering, collaborative filtering, and hybrid recommendation systems [8]. This study uses a content-based filtering method. Thus, a comparative analysis of the studies using the content-based filtering method in the literature is shown in Table 1.

The purpose of content-based recommendation systems is to list recommendations based on the user's preferences. Generally, it is based on the concept that items with similar attributes will be rated similarly. Content-based method tries to generate recommendations based on the number of similarities. In general, in these systems, user profiles are created with the items that users interact with. These interactions can be listed as purchasing items, reading in the past, clicking links, evaluating, and downloading. The most descriptive properties are used to model an item and users. The similarity coefficient between the item and the user is checked to make recommendations for the created user model. A recommendation list is created, especially with the items having high coefficients. Dhanda and Verma developed a recommender system using an Efficient Incremental High-Utility Itemset Mining algorithm [9]. In that study, article recommendations are listed according to the interests of the users with the help of the inputs from the user. Likewise, Al Alshaikh et al. presented a new content-based recommendation system for the research paper field using the ACM dataset [10]. Researchers developed a content-based article recommendation system using the Pudmed dataset [11]. As a result, articles were recommended based on the explicit and implicit feedback found in the user profile. Similarly, researchers utilized a content-based method for recommending citations in an academic paper draft [12]. The researchers achieved the best results on PubMed and DBLP (*Digital Bibliography and Library Project*) datasets using the Word2Vec technique and cosine similarity. Wang et al. proposed a content-based system for computer science publications [13]. This system helps researchers submit their papers based on chi-square feature selection and the SoftMax regression model for achieving interactive online responses. They emphasized that the chi-square feature selection with the 0.23 value of the F measurement is much better than the results of other feature selection methods. The other feature selection methods — mutual information and information gain, were measured as 0.18 and 0.21 in the F-score, respectively.

Other researchers developed a recommendation system based on TF-IDF and LDA probabilistic subject modeling techniques [14]. With LDA model, the researchers extracted keywords from the title and summary information in the articles.

Table 1. Comparative Analysis in the Literature

Ref.	Year	Methods	Dataset	Performance Metrics	Evaluation Method	Input Metadata	Output
[9]	2016	EIHI algorithm and PLSA algorithm	ACL Anthology Network	Citation frequency	**Offline**	publishing date	Top 10 paper
[10]	2017	Cosine Similarity, **TF-IDF**	ACM and CiteSeerX dataset	Accuracy, Average Precision, MAP	**Offline**	Title, keywords, abstract	Top N paper
[11]	2017	**Word2vec**, RNN	Pudmed Dataset	-	-	Title, abstract	Personalized Recommendation
[12]	2018	Nearest neighbor, **Word2Vec**, **Cosine Similarity**	Pubmed, DBLP, OpenCorpus	MRR and F1	**Offline**	Title, key, venue, authors, abstract	Top 3 paper recommendation
[13]	2018	Chi-square feature selection, **TF-IDF**	The abstract of papers from distinct journals and conferences	Accuracy, F-score, TPR, FPR, and ROC	Online	Title, abstract, author, link of paper	Top 3 paper
[14]	2019	**TF-IDF**, **LDA**	-	Precision, Recall F-score	-	Title, abstract, keywords	Research paper classification
[15]	2019	**TF-IDF**, SVD, **LDA**, Oklid distance	The dataset is created from the prepared daily records	-	**Offline**	Title, keywords, aim, scope	Journal Recommendation
[16]	2019	**TF-IDF**, **Cosine similarity**	DBLP	Ranks, Low medium, high distance	Online	Title, year, author, conference	Nine papers
[31]	2019	**Word2vec, TF-IDF**	the publicly available dataset	Precision, NDCG	Online	Title, keywords, and abstract	Top N paper recommendation
[20]	2020	**TF-IDF**, **Cosine similarity**	the publicly available dataset	Precision, Recall, F-score, MAP, MRR	**Offline**	Title, abstract, keywords, references	Top N paper recommendation
[18]	2020	**Doc2vec**, **LDA**, RWR	DBLP, ACM, MAG	Accuracy, Precision, NDCG, MRR, F1, Diversity, Stability	**Offline**	Title, abstract	Top N venues
[29]	2020	**Word2vec**, **Cosine Similarity**	**ARXIV dataset**, Aminer dataset	Accuracy, Precision, Recall, F-score	**Offline**	Title, abstract	Paper classification
[30]	2020	**Word2vec**, BM25, **LDA**, **Doc2vec**, LSA	Geo dataset, MEDLINE dataset,	Recall	**Offline**	Title, abstract	Top N paper recommendation
[19]	2021	**TF-IDF**, **Cosine Similarity**	Articles collected from the ARXIV	-	-	Title, abstract	Top N paper recommendation
[20]	2021	**LDA**, **TF-IDF**, JSD	100 publications from 10 domains	Precision	**Offline**	Title	Top N paper recommendation
[21]	2021	**TF-IDF**, **Cosine Similarity**, ED, PC, LK, SK	Articles collected from the NIPS and ARXIV	-	-	Title, abstract	Top N paper recommendation
[22]	2022	**Doc2vec**, XGBoost	DBLP	Accuracy, Precision, Recall, F1	**Offline**	Title, abstract, keywords	Top N paper recommendation

Patra et al. developed a system that allows researchers to use their time efficiently and recommends relevant articles for datasets [30]. The proposed system's recall@10 values are 0.3687, 0.4752, and 0.5168 for LDA, Word2vec, and Doc2vec, respectively. The recommendations are listed according to the weighted values obtained with TF-IDF. Kanakia et al. developed a hybrid recommendation system that was evaluated with a user study of 40 participants [31]. As a result, precision@10 in co-citation and content-based methods are evaluated as 0.315 and 0.226, respectively. In another study, a system was developed to solve the problem of researchers searching for publication locations [15]. The purpose of utilizing LDA is size reduction, and TF-IDF was used to find the semantic analysis. Olshanniko et al. developed a recommendation system based on topic modeling. In this study, researchers' perceptions of the relevance of potential collaborators were evaluated [16]. Haruna et al. used research articles with general contextual metadata, regardless of the research area and user expertise [17]. This way, a new content-based and collaborative relationship-based approach is proposed to customize recommendations based on the hidden relationships between articles. Another study, recommending a place for publication is the content and network-based recommendation system called CNAVER by Pradhan and Pal [18]. DBLP dataset was utilized in the experiments. In the system proposed by Gündoğan and Kaya, an article was classified using the Word2vec technique [29]. Öz et al. developed a prototype for the content-based article recommendation system [19]. The researchers were introduced a system that lists the articles most similar to any article that takes the title and abstract information as input. The researchers created a system based on LDA probabilistic subject modeling and Jensen distance [20]. Deniz et al. proposed a content-based recommendation model based on title and summary similarity to recommend appropriate papers in the field of computer science [21]. Researchers used the TF-IDF method, Linear Kernel, Sigmoid Kernel, Euclidean Distance, Pearson Correlation, and Cosine Similarity. Besides, mutual analysis was utilized for similarity measurements. On the other hand, journal recommendation systems extract the relationship between the articles and the topics accepted by the journals. Therefore, the study by ZhengWei et al. extracted the bibliographic information of each article to obtain a text representation of the bibliographic information of each scientific paper [22]. Then, the model was trained for article and journal matching classification according to the topics accepted by the journals. As a result, they utilized article and journal matching classifications and achieved high accuracy.

In most of the above approaches, recommendation lists are created by inferring the association of users or articles. Generally, user or article profiles are used to develop recommendation lists. Recommendation lists are created using the general metadata in the data set. By using both title and summary information, the most overlapping articles are recommended to each other. Unlike content-based recommendation systems in literature, user profiles are used to measure the accuracy of the recommendation lists. Most importantly, using user profiles only during the evaluation phase prevents the recommendations from being dependent on user profiles. In this way, we recommend similar articles in terms of content to researchers working on different subjects.

3. Material and Methods

In content-based recommendation systems, features such as items that users interact with, users' ratings, and the number of clicks, are used to recommend similar things to each other. In article recommendation systems, users are researchers, and items are articles. In these systems, similar relationships are extracted, and similar papers are recommended to each other with the help of these relationships. Firstly, the researchers' articles are collected in the content-based article recommendation systems. Then, user profiles are created. User profiles are used to find similar users. These profiles may include the researcher's preferences, interests, information such as citations by researchers, research areas of the article, and keywords used in the article content. These profiles are used in content-based systems to determine the relationship between articles. By finding similarities in content-based article recommendation systems, similar articles are suggested to researchers working on the same topics. As a result, most related articles recommend each other based on the user profiles created in the article recommendation systems.

In this study, a content-based recommendation system is developed that allows researchers to recommend the necessary articles quickly and accurately. Firstly, the words in the dataset should be converted into meaningful data to create a similarity between the articles. In this system, a recommended list is created in order of similarity using the title and summary information of the articles.

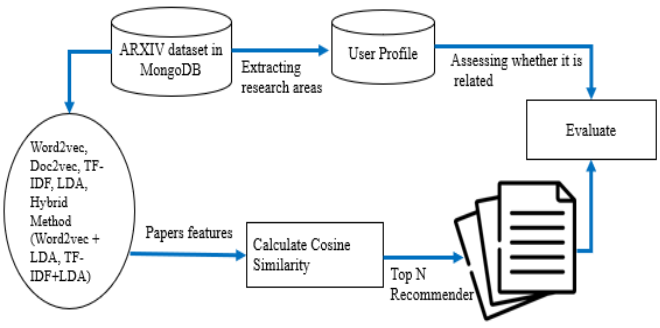

Figure 1. Article Recommendation System Basic Steps

In addition, the relationships between the research areas of the papers are used to create the paper profiles. Thus, Word2vec, TF-IDF, Doc2Vec, and LDA methods are used to calculate the similarity between the articles. Then, comparative experiments are conducted for better analysis of hybrid and other word embedding techniques. The basic

steps of the content-based article recommendation system are presented in Figure 1.

Firstly, we transformed all the articles into a meaningful form with word embedding techniques. Then, user profiles are created with the knowledge of the research interests of the articles, and features were extracted with word embedding techniques. After that, the cosine similarity technique is utilized to calculate the similarities of the articles to each other. Finally, the top 10 related articles are recommended according to the input article. In the evaluation of performance, tests are implemented considering the research areas in the article profiles and the research areas of the proposed articles. In the proposed system, it is concluded whether the articles are related to each other considering the research areas. In this way, the first N articles are listed by similarity score. Figure 2 shows the detailed architecture of the proposed recommendation system. All the enumerations in Figure 2 match the corresponding subsection.

3.1. Dataset

ARXIV dataset contains the publication lists of 24,707 researchers with different research interests, such as computer science, economics, mathematics, physics, electrical engineering and systems science, and statistics [32]. The dataset includes 41,000 records of titles, abstracts, authors, article links, and research areas. Three-fifths of the dataset was used for training, and the rest for test data. Table 2 gives the contents of the ARXIV dataset.

Table 2. ARXIV Dataset

Total number of articles	41,000
Total number of researchers	24,707
Total number of research areas	8
Total number of sub-research areas	52
Year range of articles published	1993-2018

In this study, we first imported the ARXIV dataset in JSON (JavaScript Object Notation) format to the MongoDB database application. After adding the dataset in JSON format to the application, we connected it to the MongoDB database. On the other hand, in the user profile stage, user profiles were created by extracting research interests from the MongoDB database. At this stage, article profiles containing the research area information are created, which are considered user profiles. In content-based recommendation systems, user profiles include information such as items the user has liked in the past, their click history, and what users have previously viewed and rated.

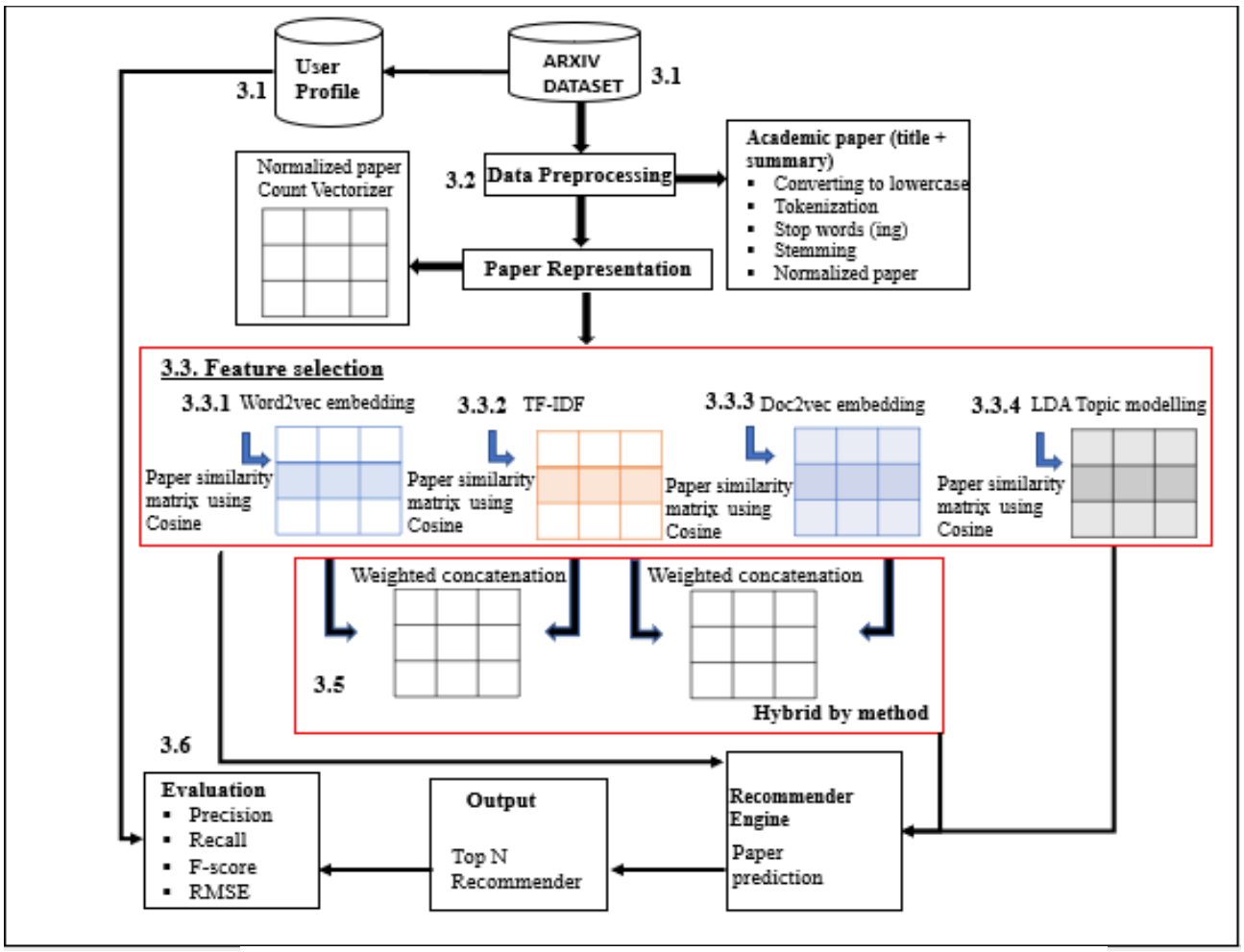

Figure 2. The Architecture of the Proposed Academic Article Recommendation System

The dataset used in the study does not contain data such as user ratings, click history or previously read articles. In content-based recommendation systems, researchers need these profiles because recommendations are made using user profiles. Therefore, with this data, it is necessary to establish a connection between articles or create profiles that show the relationship between the paper and the user. In this study, the research subjects of the articles are used to establish a relationship between the papers. Firstly, the research area information of each article is extracted from the dataset. If the research area of one article and the research area of the other article are the same, there is accepted that it is a relationship between these two articles. Thus, article profiles are created by extracting the relationships between the articles. Each article in the dataset has up to three research areas. The dataset includes eight research area topics and 52 sub-research topics. The similarities are calculated after extracting the relationships with the help of these research and sub-research areas. As a result, the articles are listed with the highest similarity.

3.2. Data Preprocessing

Natural language processing is an artificial intelligence application that enables a computer to understand and interpret human language. For the text data to work in the computer environment using this application, it must go through some preprocessing steps. Thus, the complexity of the model is reduced with preprocessing steps and high-performance recommendations [13]. First, the process is applied to remove punctuation and lowercase to conversion. On the other hand, tokenization is a method that divides enormous amounts of text data into smaller pieces. In the third part, it is the stage where common words in English that do not have a meaning in a sentence, such as prepositions, conjunctions, and pronouns removed, words known as stop words in texts. The next stage is the stemming stage. Stemming usually reduces the word to a root by removing the derivational suffixes. As a result, normalized data gets with the preprocessing steps of natural language processing to increase the performance system before feature selection.

3.3. Feature Selection Module

In the literature, recommendation systems use natural language processing methods and word embedding techniques to list recommendations. This section briefly introduces the different techniques used in the proposed content-based recommendation system. In the study, extracting the semantic representations of the title and summary information of the articles in the dataset applies word embedding techniques such as Word2vec, TF-IDF, Doc2vec, and the subject modelling technique LDA.

3.3.1. Word2vec

Mikolov et al. proposed a neural network-based model that generates semantic representations of words in 2013 [23]. This model represents words as vectors by embedding words in vector spaces. The input in the model is a two-layer neural network that generates a hundred-dimensional vector space. This model is not a neural network but converts the input text into a digital form that deep neural networks can process as input. The Word2vec technique includes parameters shown in Table 3.

Table 3. Word2vec Parameters

Parameters	Description
sentences	list of preprocessed words
vector_size	Number of tokens used to represent words
window	Maximum distance between context word and neighboring word
min_count	Ignores words with low total frequency
workers	Number of threads used while training the model
sg	If sg is 0 then CBOW algorithm, if 1 is Skip-gram algorithm
alpha	Gives the initial learning rate (0.001 to 0.005)
min_alpha	As training progresses, the learning rate decreases linearly to this value.

In the Word2vec technique, the lower the frequency in the text for a word, the more insignificant the word. This technique has two algorithms, Skip-gram and CBOW (*Continuous Bag of Words*). If *sg* value is *0,* Skip-gram algorithm is used. If *sg* value is *1*, then the CBOW algorithm is utilized in the model. In conclusion, Word2vec is a word embedding technique used to learn relationships or dependencies between words and to solve various natural language processing problems. For this study, *window* 100, *min_count* 3, *workers* 32, *negative* 10, *alpha* 0.03, *min_alpha* 0.0007, and *sg* value is 1. The vectorial provisions of the words were determined using the skip-gram algorithm. The *W(t)* shown in this figure is the central word in a sentence. The words to the right of the central word and in the hall are the words from W(t-2) to W(t+2). In the example in Figure 3, the working logic of the architecture is shown in detail.

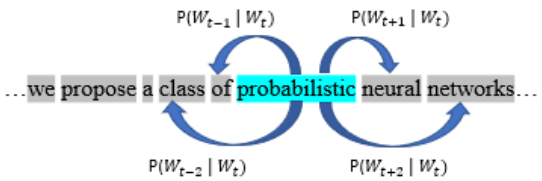

Figure 3. Skip-gram Source Text

In this study, firstly, the title and summary information of the articles in the dataset extracts from the MongoDB database. To work effectively with text data, the words normalizing by going through preprocessing. Then the vocabulary containing all the unique words is built. After that, we created a new database containing only the research

area of the articles. The Word2vec model is created by setting the parameter values. A two-dimensional word2vec matrix was obtained after training the model. We calculated the similarities between the resulting word2vec matrix and the articles using cosine similarity. As a result, the top 10 papers that are most like the target article are listed. In the performance evaluation stage, article profiles were used to measure the performance of the systems. We created the article-research relationship according to the research interests of the articles. Finally, we measured performance with the evaluation metrics.

3.3.2. TF-IDF

TF-IDF determines how often a word occurs in an article and how important that word is for that article. Simply *TF* is the word frequency in a document, as shown in Equation 1. *IDF* is the inverse of document frequency among all documents, as shown in Equation 2. The importance of the word in the article increases in proportion to the number of occurrences in the paper. TF-IDF method is calculated as shown in the Equation 3, given below. Here, the frequency of a t_i word in document d is measured. As a result, that is seeing that the frequency of words in long articles is higher than in others with short ones. For this reason, term frequency is normalized using the length of the articles [10].

$$TF_{(t_{ij})} = \frac{\text{Number of times term } t_i \textit{ appears in document } d}{\text{Total number of terms in } d_j \textit{ document}} \quad (1)$$

$$IDF_i = \log(\frac{Number\ of\ all\ documents\ found\ in\ training}{Numer\ of\ documents\ containing\ the\ term\ t_i\ in\ the\ training\ set} \quad (2)$$

As a result, the TF-IDF formula is given in Equation 3.

$$TF - IDF_{(t_{ij})} = TF_{(t_{ij})} \times IDF_{\text{i}} \quad (3)$$

The weighted terms obtained with TF-IDF are calculated between *0* and *1* for each paper in the training set. Because of this result, all concepts for representing the article and measuring the similarity vector between articles are associated with the training set with weighted terms obtained by TF-IDF.

The purpose of this study is to recommend articles with the same research topics to each other. To find related articles, TF–IDF subtracts the weights of terms numerically. Likewise, firstly, the articles pulled from the ARXIV dataset were added to the MongoDB database. Then, the title and summary information of the articles in the dataset extracts from the MongoDB database. To work effectively with text data, the words normalizing by going through preprocessing. After that, in the second stage, we created the TF-IDF model. Thanks to this stage model created the TF-IDF matrix. The weights of the words from the TF-IDF technique articles calculates according to the frequency. Then, we calculated the similarity of the articles to the content. User profiles are created to measure the system's performance. Finally, we measured the performance of the systems according to the article-research field relationship in the article profiles. As a result, recommending articles containing the same research areas to each other shows that the system worked correctly. The TF-IDF technique includes parameters shown in Table 4.

Table 4. TF-IDF Parameters

Parameters	Definition	Value
stop_words	ineffective words	english
strip_accents	'ASCII' works on characters with ASCII mapping, while 'unicode' works on whichever character.	unicode
ngram_range	the n-value range for different n-grams to subtract	(1, 2)
analyzer	whether the attribute should be made of words or n-grams of characters.	word
min_df	terms with document frequency lower than the given threshold	0.003
max_df	terms with document frequency higher than the given threshold	0.5

3.3.3. Doc2vec

In the Word2vec technique, each word is represented by a vector, while in the Doc2vec technique, each document exemplifies a vector. This technique creates a digital representation of documents without considering the length of the papers. The Doc2vec model was developed by the same researchers one year after the Word2vec [24]. This model is a deep learning algorithm used for the vectorization of texts. Although similar to the Word2vec technique, Doc2vec has a sentence vector in the input layer in the neural network [22]. In general, the Doc2vec model is used to determine the meaning, grammar, and word order of the sentence, besides converting it to a fixed-size vector. In this way, the similarity of the vectors calculating and applied recommendation algorithm. In addition, the CBOW and skip-gram algorithms of the Word2vec model correspond to the DM (*Distributed Memory*) and DBOW (*Distributed bag of words*) algorithms in the doc2vec architecture. In this study, distributed memory algorithm is used.

The Doc2vec model trains enormous text data in a completely unsupervised manner, without the need for averaging word vectors or private label datasets, unlike n-gram models [26]. In this study, the parameters used in the Doc2vec method with the values of *vector_size* 100, *alpha* 0.025, *min_alpha* 0.00025, *min_count* 2, and *dm* 1. As in the other two methods, in the Doc2vec model, the title and summary information of the articles got from the MongoDB database. Then, the title and summary information went through the data preprocessing steps. After that, the vocabulary was created for the model and trained in 10 epochs. In this way, we created the doc2vec matrix. Doc2vec creates a representation of the documents. Therefore, the similarity of each document to the other is calculated by

cosine similarity. As a result, the ten most similar article recommendations were made, as in the TF-IDF and Word2vec methods. As in other methods, the article-research field relationship got with the help of article profiles containing research areas during the evaluation phase. Then we used this relationship to calculate the accuracy of the first ten articles. In this way, it shows that high accuracy evaluations obtain if the articles in the same research area recommend each other based on the research area information in the article profiles.

3.3.4. LDA

We used the LDA probability model to match a topic model with the article data. This model conceptualizes each word in a document as string of N words [25]. Thus, each word belongs to all topics but has different probabilities, and all topics are given different weights in each paper. LDA is a nonlinear subject modeling technique proposed by Blei et al [27]. In this model, each of the nodes shown here is a random variable and is labeled differently according to the role of the process. LDA assumes that every document of several words can represent a Dirichlet probability distribution on confidential matters. In the LDA model, the output consists of groups of keywords, each of which falls under a specific topic category. Then these keyword groups are tagged with unique topic titles.

Recommendations made in content-based article recommendation systems should be similar in terms of content. For this reason, it is a thing that articles containing the same topics will use a similar word group in terms of content. For this reason, the LDA technique uses the article recommendation system to group articles containing the same topics. The words with the highest probability in each subject give us information about the scope of the topics. In this way, LDA estimates probabilities. With LDA modelling, more than one topic extracts from the content of an article. At this stage, the purpose is to identify the issues with high probability and to make recommendations based on those issues. In this algorithm, the title and summary information of the articles extracts from the MongoDB database and passed through the preprocessing steps. Then, the dictionary and corpus created required for subject modelling. These help us find the frequency of words in documents. Then, we created the LDA model in the Gensim library. We used various parameters while creating this model.

Table 5. LDA Parameters

Parameters	Definition
num_topic	Number of hidden topics needed to extract all builds
id2word	Used for word sizing, debugging, and topic printing
chunksize	The number of documents to be used in each training
passes	The number of passes through the corpus in training
corpus	Consists of the ids of the words and the frequencies in the text data

Table 5 gives these parameters. With these parameters, we calculate the weights of the topics. Then we choose the topics with high probability. As in word embedding techniques, the LDA technique, the article-research relationship got according to the created article profiles. Then, the first 10 article recommendations are according to the input article. The article-research field relationship in the article profiles uses to measure the performance.

3.1. Weighted Concatenation using hybrid models

Besides methods such as Word2vec, TF-IDF, Doc2vec, and LDA used in the study, tested hybrid by method approaches. In the first approach, we combined TF-IDF and LDA techniques. A new matrix was created by taking the weighted average of the formed matrix. Similarly, cosine similarity was used to measure similarity between articles, as in other methods. In the other approach, we combined Word2vec and LDA techniques. We took the weighted average of the matrix obtained with Word2vec and the matrix obtained by LDA modelling. Likewise, cosine similarity was used to calculate similarity. In both hybrid approaches, the first 10 article recommendations make as in the other methods. The method-based evaluations in the system were effective in making comparisons with the basic models. We discussed the advantages of the two methods used in both experiments and aimed to achieve higher accuracy performance.

3.2. Content-Based Recommendation

Cosine Similarity measurement is shown in equation 4 to extract the similarities between the target article and the proposed article.

$$\cos(\theta) = \frac{A \times B}{||A|| \times ||B||} == \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (4)$$

A_i and B_i represent the feature vectors of the articles.

In hybrid tests, operations are performed by taking the weighted average of the similarity matrices obtained from the Word2vec technique and LDA probability-based subject modeling methods. Likewise, processes were performed with the weighted average of the TF-IDF and LDA methods with the help of similarity measurement. In this way, article recommendations are listed according to the similarity scores. As a result, articles that are like each other with the cosine similarity are recommended to researchers. Moreover, word embedding techniques and LDA topic modelling are used.

3.3. Evaluation Metrics

There are different approaches used in the evaluation of recommendation systems. Researchers should use the system architecture-appropriate evaluation method to demonstrate that the systems they developed are working true. The proposed content-based academic article recommendation system is evaluated offline. One of the most important issue to be considered in recommendation systems is that the relevant items are at the top. Therefore, rank-based metrics are used in this study. Rank-based metrics evaluate the recommended list up to a certain threshold of N. In this study, Precision@N, Recall@N, and F1-Score@N metrics are used. The Precision@N metric calculates the precision for the first N values of the recommendation list. The aim is to find errors for the first N values of the list. Likewise, Recall@N calculates recall for a subset of N recommendations. F1-score@N calculates the harmonic mean of the precision and recall values of the Top N recommendation. Equation 5 is the Precision formula used in recommendation systems. The precision formula measures the accuracy of the articles recommended to the user. The recall formula in Equation 6, measures the ratio of the first N recommended articles to all papers. The F1 score given in Equation 7 is a harmonic mean of precision and recall values [28].

$$Precision = \frac{Number\ of\ relevant\ papers}{Total\ number\ of\ recommend\ papers} \quad (5)$$

$$Recall = \frac{Number\ of\ relevant\ papers}{Total\ number\ of\ relevant\ papers} \quad (6)$$

$$F1 - score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \quad (7)$$

As a result of the evaluation metrics, were compared word embedding techniques. In addition, the performance of the systems created as a hybrid based on the method evaluated.

4. Results and Analysis

This section presents the experimental results of the performance evaluation of the proposed academic article recommendation system. ARXIV data set was used in the content-based article recommendation system. This data set consists of many sub-disciplines and articles covering many fields including mathematics, statistics, electrical engineering, and quantitative biology. The research areas of the articles include a maximum of three sub-areas. It obtains high accuracy in the article recommender system containing a single research area. For this reason, it is a useful data set for researchers who specialize both in a single field or more than one field. In this study, the performance is measured by performing offline experiments. The purpose of the experiments is to compare the performance of word embedding algorithms used in this article's recommendation system. Precision, recall, and F1-score metrics are used to compare the performance of the methods.

Table 6. Comparative Analysis of Methods

	Precision					Recall					F1-Score				
	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50	@10	@20	@30	@40	@50
Word2vec	.131	.157	.169	.169	.169	.541	.595	.622	.649	.649	.336	.376	.396	.409	.409
TF-IDF	.131	.131	.143	.163	.169	.541	.541	.594	.675	.702	.336	.336	.369	.419	.436
Doc2vec	.134	.144	.154	.175	.175	.541	.625	.708	.708	.750	.337	.384	.431	.441	.462
LDA	.134	.144	.185	.185	.185	.436	.509	.587	.587	.587	.285	.326	.386	.386	.386
Word2vec+LDA	.144	**.185**	**.195**	.206	**.206**	**.625**	**.791**	**.791**	**.791**	**.791**	**.384**	**.488**	**.493**	**.498**	**.498**
TF-IDF+LDA	**.164**	**.185**	**.195**	**.195**	.195	.541	.583	.666	.666	.666	.353	.384	.431	.431	.431

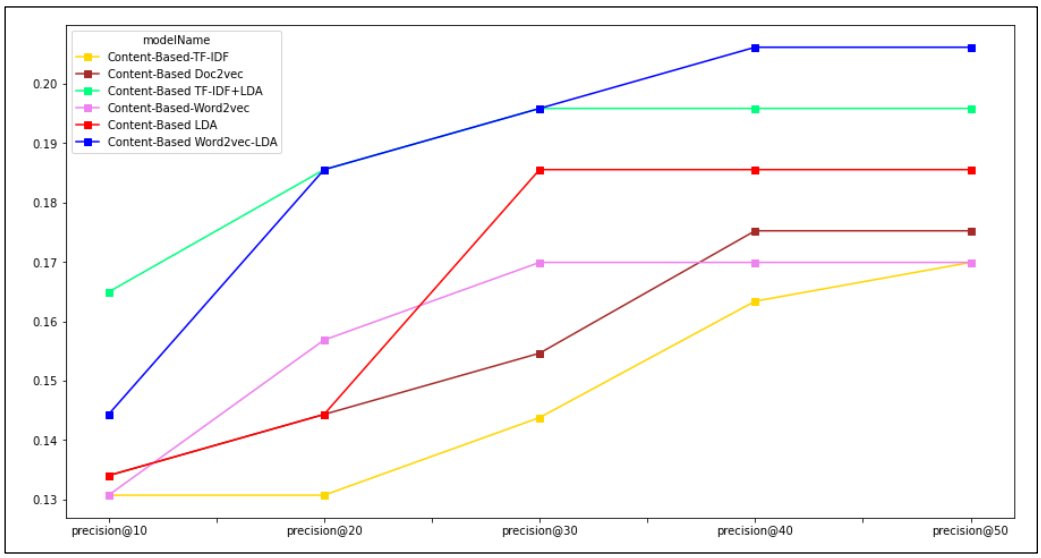

Figure 4. Precision of Top-N recommendation

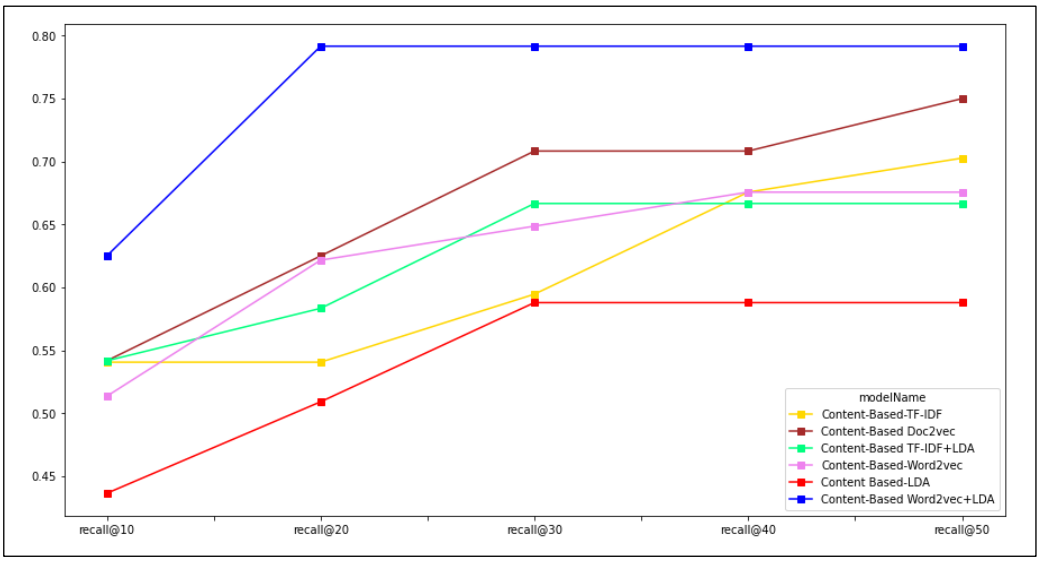

Figure 5. Recall of Top-N recommendation

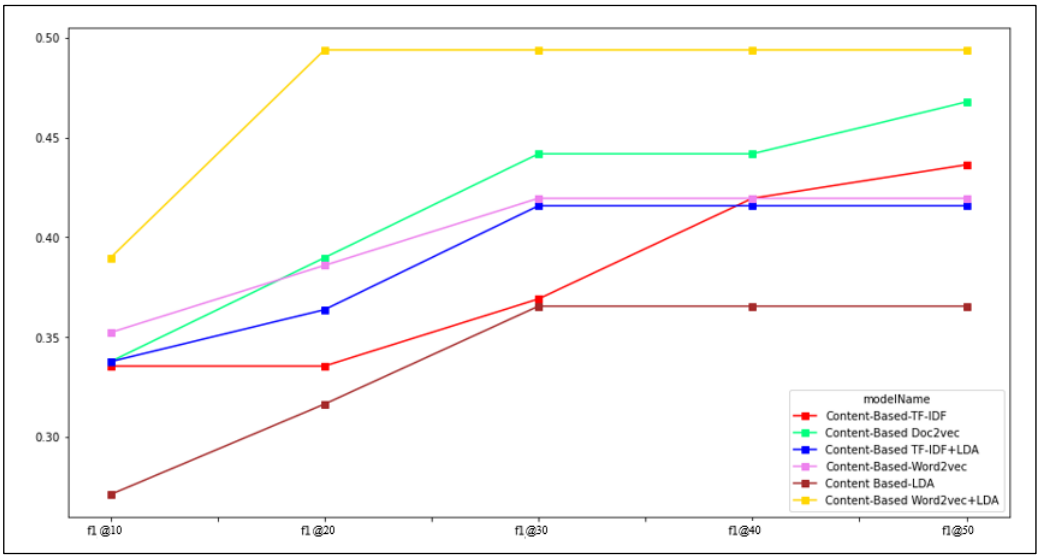

Figure 6. F1-Score Top-N recommendation

As a result of the experiments, the highest scores are obtained with Word2vec+LDA techniques. The performance values of the algorithms are shown in Table 6. In Table 6, the method with the highest accuracy is marked in bold. Figures 4, 5, and 6 show a graph of Precision@N, Recall@N, and F1-score@N, respectively. In general, as the N numbers increase, the performance in the system also increases. It is because the higher the number of recommended articles, the more likely it is to find the right publication. The highest performance is found with the Word2vec + LDA technique. Primarily, a comparison of Precision@N values gives in Figure 4. While the highest Precision@N value was 0.20619 with Word2vec + LDA, the Precision@N value was 0.19587 with TF-IDF + LDA approach. We achieved less accuracy when using only Word2vec or LDA techniques. But by combining both methods and using the advantages of the algorithms, we achieve high performance results in the hybrid technique. Likewise, we achieved less accuracy when using only TF-IDF or LDA techniques. But by combining both techniques, we have achieved higher performance outputs. Therefore, the experimental results show that the proposed approach can bring the relevant papers back to the user more than others. Figure 5 shows a comparative analysis of the Recall@N values. In this evaluation criterion, we found the highest accuracies with the techniques considered hybrid. Figure 6 shows the F1-score@N values.

In the results obtained, the results of the Top 50 recommendations are higher than the results of the Top 10 recommendations. The reason for the high performance, the Top 50 can give a wider solution area. With the increase in the solution area, the probability of the proposed articles being relevant also increases. As a result, we achieved the highest performance among Word2vec,

TF-IDF, Doc2vec, and LDA techniques with the Doc2vec technique with a recall value of 0.75. In the experiment where Word2vec and LDA techniques handled as hybrids, we achieved a higher output of 0.7919 than Doc2vec and other algorithms. In general, precision@50, recall@50, and F1-score@N gets significantly better results than other methods in Word2vec + LDA and TF-IDF + LDA approaches, which we created as hybrids, respectively. The results obtained in this way showed an improvement over other method.

5. Conclusion

In this study, a hybrid system is proposed based on a content-based method with the help of user profiles containing the research areas of the articles. We made an in-depth analysis of the proposed system with word embedding algorithms and topic modelling techniques. As a result of the increase in the initial N values, there was a significant increase in all values of the recommended articles. In the study, besides the research areas of the articles, we also considered the similarities of the articles in terms of content. Thus, the most similar paper recommendations are listed according to the target article. The reason the performance of the system gives high results is that the two techniques are used together. In future work, we will try various hybrid architectures on the word embedding techniques. We will also aim to increase the performance of systems with experiments in which we will also use deep learning algorithms.

Nomenclature

Ave-quality : Average Venue-quality
AUC : Area Under the ROC Curve
ED : Euclidean Distance
EIHI : Efficient Incremental High-Utility Itemset Mining algorithm
FPR : False Positive Rate
JSD : Jensen Shannon Divergence
LK : Linear Kernel
LSA : Latent Semantic Analysis
MAP : Mean Average Precision
MRR : Mean Reciprocal Rank
NDCG : Normalized Discounted Cumulative Gain
NIPS : Neural Information Processing Systems
PC : Pearson's Correlation
RNN : Recurrent Neural Network
ROC : Receiver Operating Characteristic
RWR : Random Walk with Restart
SK : Sigmoid Kernel
SVD : Single-Value Decomposition
TPR : True Positive Rate
XGBoost : Extreme Gradient Boosting

Acknowledgment

This study is supported by the decision of the Scientific Research Projects Commission (ESOGU-BAP) dated 14.01.2021 and dated 61690618-622.03 (project number 202115003) dated 14.01.2021. Thank you for your contributions.

References

[1] Reddy, S. R. S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. *Content-based movie recommendation system using genre correlation.* Singapore, In Smart Intelligent Computing and Applications*, 2019,* p. 391-397.

[2] Qi, T., Wu, F., Wu, C., & Huang, Y. *Personalized news recommendation with knowledge-aware interactive matching.* In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021. p. 61-70.

[3] Schedl, M. *Deep learning in music recommendation systems*. Frontiers in Applied Mathematics and Statistics, 2019. p. 44.

[4] Devika, P., Jisha, R. C., & Sajeev, G. P. *A novel approach for book recommendation systems*. In 2016 IEEE international conference on computational intelligence and computing research (ICCIC), 2016. p. 1-6

[5] Gopalakrishnan, T., Sengottuvelan, P., Bharathi, A., & Lokeshkumar, R. *An approach to webpage prediction method using variable order Markov model in recommendation systems.* Journal of Internet Technology, 2018. **19**(2): p. 415-424.

[6] Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., & Xia, F. *Scientific paper recommendation: A survey.* IEEE Access, 2019. **7***:* p. 9324-9339.

[7] Das, J., Majumder, S., Dutta, D., & Gupta, P. *Iterative use of weighted voronoi diagrams to improve scalability in recommender systems*. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2015. p. 605-617.

[8] Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. *Recommender system application developments: a survey*. Decision Support Systems, 2015. **74***:* p. 12-32.

[9] Dhanda, M., & Verma, V. *Recommender system for academic literature with incremental dataset*. *Procedia Computer Science*, 2016. **89**: p. 483-491.

[10] Al Alshaikh, M., Uchyigit, G., & Evans, R. *A research paper recommender system using a Dynamic Normalized Tree of Concepts model for user modelling*. In 2017 11th International Conference on Research Challenges in Information Science (RCIS), 2017. p. 200-210.

[11] Hassan, H. A. M. *Personalized research paper recommendation using deep learning*. In Proceedings of the 25th conference on user modeling, adaptation and personalization, 2017. p. 327-330.

[12] Bhagavatula, C., Feldman, S., Power, R., & Ammar, W. *Content-based citation recommendation*. In Proc. of NAACL, 2018.

[13] Wang, D., Liang, Y., Xu, D., Feng, X., & Guan, R. *A content-based recommender system for computer science publications*. Knowledge-Based Systems, 2018. p. 157, 1-9.

[14] Kim, S. W., & Gil, J. M. *Research paper classification systems based on TF-IDF and LDA schemes*. Human-centric Computing and Information Sciences, 2019. **9**(1):p. 1-21.

[15] Jain, S., Khangarot, H., & Singh, S. *Journal recommendation system using content-based filtering*. In Recent developments in machine learning and data analytics, 2019. p. 99-108.

[16] Olshannikova, E., Olsson, T., Huhtamäki, J., & Yao, P. *Scholars' Perceptions of Relevance in Bibliography-Based People Recommender System*. Computer Supported Cooperative Work (CSCW), 2019. **28**(3):p. 357-389.

[17] Haruna, K., Ismail, M. A., Qazi, A., Kakudi, H. A., Hassan, M., Muaz, S. A., & Chiroma, H. Research paper recommender system based on public contextual metadata. Scientometrics, 2020. 125(1):p. 101-114.

[18] Pradhan, T., & Pal, S. *CNAVER: A content and network-based academic venue recommender system*. Knowledge-Based Systems, 2020. **189:** p. 105092.

[19] Öz, V. K., Deniz, E., Keser, S. B., Kartal, Y., & Okyay, S. *Yeni Bir İçerik-Tabanlı Akademik Makale Tavsiye Sistemi Prototipi Geliştirilmesi*. Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Bilişim Dergisi, 2021. **2**(2): p. 6-11.

[20] Bagul, D. V., & Barve, S. *A novel content-based*

recommendation approach based on LDA topic modeling for literature recommendation. In 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021 p. 954-961.

[21] Deniz, E., ÖZ, V. K., Keser, S. B., Okyay, S., & Kartal, Y. *İçerik tabanlı bilimsel yayın öneri sisteminde benzerlik ölçümlerinin incelenmesi*. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 2021, **12**(2): p. 221-228.

[22] ZhengWei, H., JinTao, M., YanNi, Y., Jin, H., & Ye, T. *Recommendation method for academic journal submission based on doc2vec and XGBoost*. Scientometrics, **127(**5): p. 2381-2394.

[23] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. *Distributed representations of words and phrases and their compositionality*. Advances in neural information processing systems, 2013. p. 26.

[24] Le, Q., & Mikolov, T. *Distributed representations of sentences and documents*. In International conference on machine learning, 2014 p. 1188-1196.

[25] Morsomme, R., & Alferez, S. V. *Content-Based Course Recommender System for Liberal Arts Education*. *International Educational Data Mining Society*, 2019.

[26] Nandi, R. N., Zaman, M. A., Al Muntasir, T., Sumit, S. H., Sourov, T., & Rahman, M. J. U. *Bangla news recommendation using doc2vec*. In 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), 2018. p. 1-5.

[27] Blei, D. M., Ng, A. Y., & Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, 2003. p. 993-1022.

[28] Sakib, N., Ahmad, R. B., & Haruna, K. *A collaborative approach toward scientific paper recommendation using citation context,* 2020. **8:** p. 51246-51255.

[29] Gündoğan, E., & Kaya, M. *Research paper classification based on Word2vec and community discovery.* In 2020 International Conference on Decision Aid Sciences and Application (DASA), 2020. p. 1032-1036.

[30] Patra, B. G., Maroufy, V., Soltanalizadeh, B., Deng, N., Zheng, W. J., Roberts, K., & Wu, H. 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, 2020. p.104, 103399.

[31] Kanakia, A., Shen, Z., Eide, D., & Wang, K. *A scalable hybrid research paper recommender system for microsoft academic*. In The world wide web conference. 2019. p. 2893-2899.

[32] https://arxiv.org/category_taxonomy