A HYBRID ARTICLE RECOMMENDATION SYSTEM BASED ON DEEP LEARNING AND CO-PUBLICATION NETWORK ANALYTICS

BÜŞRA ATLANEL

BOĞAZİÇİ UNIVERSITY

2019

A HYBRID ARTICLE RECOMMENDATION SYSTEM BASED ON DEEP LEARNING AND CO-PUBLICATION NETWORK ANALYTICS

Thesis submitted to the

Institute for Graduate Studies in Social Sciences

in partial fulfillment of the requirements for the degree of

Master of Arts

in

Management Information Systems

by

Büşra Atlanel

Boğaziçi University

2019

DECLARATION OF ORIGINALITY

I, Büşra Atlanel, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature.....

Date 13.09.2019

ABSTRACT

A Hybrid Article Recommendation System Based On

Deep Learning and Co-Publication Network Analytics

In recent years, with the rapid development of world wide web, researchers are spending more effort and time to reach the most relevant academic work for their studies because of the information overload. Preventing users from being distracted by a tremendous amount of publications and simplification of the research process makes recommendation systems more valuable. Traditional recommendation systems generally suffer from limited coverage, data sparsity, and cold start problem. In order to tackle these problems and achieve better performance, many recommender systems started to use neural network models. Being an effective neural network model, deep learning technology can transform article titles and abstract information into text embeddings and capture non-linear relationships between these text embeddings. In addition to deep learning on text embeddings, the relationship between authors has a huge effect on their future preferences. The research of co-publication relationship with social network analysis improves the performance of the recommendation systems. In this study, the aim is to propose a hybrid article recommendation system that incorporates deep learning for article text similarity using Deep Siamese BiLSTM and social network analysis through node embeddings using co-publication and citation networks to exploit the network structure to provide benefit for recommender systems. Experiments conducted in this research show that the proposed model achieved a prediction rate of 7% on average when the number of articles to be recommended is taken as 100.

ÖZET

Derin Öğrenme ve Ortak Yayın Ağı Analitiklerine Dayalı Bir Hibrit Bilimsel Makale Öneri Sistemi

Son yıllarda internetin gelişmesiyle, internetteki bilgi ve kaynak fazlalığından ötürü akademik araştırmacılar kendi çalışmalarına ve ilgi alanlarına yönelik en uygun makaleyi bulabilmek için daha fazla zaman ve enerji harcamaktadır. Araştırmacıların internetteki bilgi yığını içinde kaybolmaması ve araştırma sürecinin kolaylaştırması açısından makale öneri sistemleri daha da değerli hale gelmiştir. Geleneksel öneri sistemleri veri seyrekliği, yeni gelen bir makale ile ilgili az verinin olması vb. problemlerden ötürü etkili çalışmamaktadır. Bu problemlerin üstesinden gelebilmek ve daha etkili sonuçlar alabilmek için, öneri sistemlerinde son yıllarda yapay sinir ağı modelleri kullanılmaya başlandı. Etkili bir yapay sinir ağı modeli olan derin öğrenme ile makalelerin başlık ve özet bilgileri metin vektörlerine çevrilerek, makaleler arasındaki doğrusal olmayan ilişkiler tespit edilebilmektedir. Ek olarak, makale yazarlarının birbirleri arasındaki ilişki, yazarların ilerideki çalışmalarında kullanacakları makale tercihlerinde büyük etki yaratmaktadır. Makale yazarlarının birlikte ortak yayın çıkardığı yazarlar, bu yazarların yazdıkları diğer makaleler ya da referans gösterdikleri makaleler arasındaki ilişkinin sosyal ağ analizleri ile incelenmesi öneri sistemlerinin performansını arttırmaktadır. Bu çalışmada ise Siamese BiLSTM derin öğrenme algoritması kullanılarak makaleler arasındaki metin benzerlikleri ile Node2Vec sosyal ağ analizi kullanılarak makale yazarları arasındaki benzerlik değerlerini analiz eden hibrit bir makale öneri sistemi geliştirilmiştir. Gerçekleştirilen denemelerde, önerileri sayısının 100'e ulaştığı durumda tahmin doğruluğunun ortalama olarak 7% seviyesine ulaştığı görülmüştür.

ACKNOWLEDGMENTS

First, I would like to express my profound gratitude to my thesis advisor, Asst. Prof. Ahmet Onur Durahim, for his endless patience and for sharing his vast knowledge.

In addition, I would like to thank other thesis committee members, Prof. Dr. Aslı Sencer and Prof. Dr. Erkay Savaş, for their insightful evaluations and contributions to this study.

I would also like to thank my precious friends who gave me a magical touch and never left me alone in this process.

Last but not least, I would like to present my sincere thanks to my beloved family, for their moral and material support and for being with me all the time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In recent years, researchers spend more effort and time to find the most relevant articles for their studies because of the information overload (Zhang & Chang, 2016). Simplifying the process of searching for related work and preventing users from being distracted by the excessive amount of research papers make recommendation systems more valuable. Traditional recommendation systems, especially content-based and collaborative filtering recommendation systems, were used in many domains such as movie, music, news and article recommendation. They were effective in these domains to some extent. But they suffered from limited coverage, data sparsity, and cold start problem. In order to surpass these challenges, many recommender systems started to use text information. Relatedly, two mainstream approaches can be referred on text analysis. Nevertheless, both have some limitations on further improvements: (1) The bag of words model can only capture a superficial understanding of a paragraph because it just considers the existence of a word. Contextual information such as word orders, phrase and paragraph-level extensions are ignored by this model. (2) Deep learning methods can capture contextual information effectively, but it increases the complexity and runtime of the model (Xie, et al., 2019). Paying attention to learning semantic meanings from the textual content, neural network-based recommendation systems have shown promising accuracy performance (Wu, Sun, Hong, Ge, & Wang, 2018).

In addition to neural networks, the relationship between researchers has a huge effect on their preferences. In scientific article recommendation systems,

authors are influenced by the studies of their co-authors. These co-publication relationships can improve the accuracy of the recommendation system.

On the other hand, authors' published works point out their latent interests. Sugiyama and his colleagues proposed an article recommendation system via the user's recent research interests (Chen & Ban, 2016). Analysis of past citing papers may lead to their further preferences.

To sum up, academic literature is expanding at a rate that requires smart algorithms for research and navigation. However, technology for finding influential and relevant articles is in its early development (West, Wesley-Smith, & Bergstrom, 2016).

The main purpose of this study is to propose an approach that incorporates deep learning for article text similarity using Deep Siamese BiLSTM and social network analysis for co-publication similarity using Node2Vec to provide benefit for recommender systems.

This study includes five major chapters which are Introduction, Literature Review, Methodology, Results & Findings, and Conclusion & Future Work. Chapter 1 is the Introduction chapter which contains a general overview of the subject and the purpose of this study. Chapter 2 is the Literature Review chapter that gives details about current studies on scientific article recommendation. Chapter 3 is the Research Methodology chapter describing deep learning and social network analysis. Chapter 4 is the Results & Findings chapter that involves evaluation methods. Chapter 5 is the Conclusion & Future Work chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Article recommender system

With the prevalence of recommendation systems, finding effective methods for recommendation tasks takes an important place in the academic world. In a scientific article recommendation problem, principal methods generally focus on article similarity. These methods are collaborative filtering (CF) approach, meta-data-based approach and content-based (CB) approach.

2.1.1 Collaborative filtering approach

Collaborative filtering is one of the most common methods in traditional recommendation systems. In the study by Liou (2016), the rating scores of the articles are gathered by a survey which is conducted among the students in a university. The drawback of this study was that a limited number of articles were asked to students. So, this may lead to the cold start problem. Meanwhile, rating information may not be available in alternative datasets. In the study by Liu and his colleagues (2015) an enhanced collaborative filtering recommendation system that takes citing and reference papers corresponding to the users has been utilized. Using the citation network, it creates an article-citation matrix. In the same way, this approach also suffers from cold start problem because recommended articles are obtained from citations by other articles. When a new article is selected as an article of interest, it must be a reference of at least one article.

2.1.2 Meta-data based approach

Doerfel and his colleagues (2012) find similarity between articles comparing their domain information which are the title, keywords, authors and publication year. The essential benefit of this approach is the availability of meta-data even if the article itself is published in paid journals. Nevertheless, this approach does not make accurate recommendations all the time. For an author to publish a paper other than his usual interest is a problem for this method.

2.1.3 Content-based approach

In the study by Ding and his co-workers (2014), the textual citation information (e.g., in which position a reference is mentioned) of two articles are compared to find the relationship between them. Compared to the collaborative filtering and meta-data-based approaches, it gives improved results and seems to be a preferable option. Primary concerns of using this method are that finding the whole textual content of the articles in digital libraries is troublesome. Also, storing this textual information can be costly and it can take a lot of time to make the matching process.

2.1.4 Hybrid systems

Hybrid systems generally integrate collaborative filtering and content-based approaches to cover up each other's deficiency. For instance, absence of a rating value for a newcomer item leads to cold start problem for collaborative filtering method. However, content-based approach meets this deficit by processing feature information of the item.

In another study, Burke (2007) measures the effects of four diverse recommendation systems with seven different hybridization methods. They apply

various hybrid combination techniques and compare with each other. They claim that their augmented hybrids give desired results.

Another study about hybrid recommendations system is conducted by Tsolakidis and his colleagues (2016) on scientific article recommendation. They proposed a hybrid approach using both the collaborative filtering and content-based approaches. In the content-based approach, articles written by the authors were indexed and TF-IDF algorithm was implemented to measure the weights for each indexed word. In collaborative filtering part, they assumed that authors tend to have similar preferences with the authors with akin behavior. Therefore, their contribution applies graph-based analysis to emphasize the effect of each indexed item.

2.1.5 Network analysis

Traditional methods like CF and CBA have their own drawbacks. In order to improve the recommendation performance, finding the latent patterns between the articles and authors is crucial.

In the research by Waheed and his co-workers (2019), they developed a recommendation system that generates a multilevel citation network that considers the relationship between the articles to acquire significant articles and an author collaboration network to find key authors from those articles using centrality measures. They compared their method with Google Scholar by evaluating NDCG metric.

What is more, Ren and his colleagues (2014) proposed a cluster-based citation recommendation model. They assumed that citations tend to be clustered into interest groups based on several types of relationships in the citation network.

According to these interest groups, they predicted citations for a given query of articles.

Additionally, Wang and some researchers (2018) proposed a hybrid approach that combines CBF and CF with social information. Social tag and social friend information which has a significant effect on recommendations systems were integrated to CBF and CF to improve accuracy.

Finally, West and his colleagues (2016) proposed a citation-based method called Eigenfactor-Recommends. They combined the citation network that represents the hierarchical structure of scientific venues, domains, fields and so forth with importance scoring based upon a network centrality measure. They used hierarchical clustering in order to find the relevance between the articles and then recommended the articles based on their relevance scores among these clusters.

2.1.6 Neural networks

In a notable study, Huang and his coworkers (2015) proposed a framework that learns the semantic representations of a citation context. The model was trained to use a multi-layer neural network to estimate the probability of citing a paper. While most citation recommendation models focus on global recommendation which recommends a list of citations for a given context, this study focuses on local recommendation which recommends references for a particular manuscript where a citation should be made.

Du and his colleagues (2018) introduced the POLAR model which integrates one-shot learning with neural networks. POLAR performs an attention-based CNN to determine the similarity score between articles. One-shot learning finds personalized scores for each author using click data of articles from RARD dataset. This model

outperforms many existing neural network methods among recommendation systems.

2.2 Node2Vec

Node2vec is an analytical framework that learns continuous element representations for nodes in networks. These representations are applicable to numerous machine learning tasks (Grover & Leskovec, 2016).

The objective of node2vec is to maximize the probability of protecting neighborhoods of nodes by learning low-dimensional representations of features. The algorithm discovers diverse neighbor nodes by using random walk. Principally, node2vec presents a perspective to modify the examination-manipulation trade-off that activates representations obeying a range of equivalences from homophily to constitutional equivalence.

The study by Zhang, Yin, Zhu, and Zhang (2018) presents an extensive review of today's literature on graph representation learning by relating to the machine learning field. They propounded new taxonomies to summarize the latest network representation learning practices. They also performed experimental research to compare the performance of representative algorithms given the datasets and analyzed their complexity.

2.3 Siamese LSTM

Siamese LSTM (Long Short-Term Memory) is an adaptation of LSTM networks to Siamese architecture for learning semantic representations of textual information. It gives a new impulse to the information extraction task in terms of measuring the semantic similarity of textual information. Unlike a classical neural network,

Siamese networks include two or more generic sub-networks. So, it is much easier to train a model because it shares weights on both sides.

Mueller and Thyagarajan (2016) proposed a very explicit approach called Siamese MaLSTM ("Ma" stands for Manhattan distance) to the common problem of sentence similarity. They designed a model for labeled data composed of pairs of variable-length series. They provided word embedding vectors in pairs with synonymous information to LSTMs, which utilized a fixed size vector to encode the semantic meaning stated in a corpus (regardless of a certain wording or syntax). By limiting consecutive operations to rely on a simple Manhattan distance metric, they forced the sentence representations learned by the model to develop a highly structured space that reflects complex semantic relationships. Thus, the learned model could utilize the hidden units to encode diverse characteristics of each sentence.

In the study of Neculoiu and his colleagues (2016), the model associates a batch of character-level bidirectional LSTM's with a Siamese structure. Normally an LSTM structure stores past information and processes inputs which have already been run through from the hidden state. However, bidirectional LSTM manages both future and past context by processing the reverse of the input through a separate recurrent neural network (RNN). At each step, the outputs are simply concatenated from the forward and backward networks. For instance, the task is to predict the next word of a sentence like "She's gone to….". While forward network uses past information, backward network processes future information in "… and she dropped her coffee to the ground". In particular, BiLSTM models demonstrate better results.

2.4 Doc2Vec

The history of Doc2Vec belongs to Word2Vec that generates and learns from word embeddings (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). Immediately after, Le and Mikolov (2014) developed another model called Doc2Vec which analyses vectors of sentences or paragraphs. Doc2Vec is an unsupervised method for learning the distributed representations of documents and sentences. Word2Vec uses one vector for each word whereas Doc2Vec utilizes one vector for each paragraph or sentence. It uses "Distributed Memory" model that links a memory vector that intends to capture the subject of the document.

In the study of Le and Mikolov (2014), they performed a sentiment analysis process using Doc2Vec method over the IMDB dataset. They trained the model with the embeddings of movie reviews. They stated that the model improved the result by 1.3% compared to the best previous result.

Another study conducted by Dai, Olah, and Le (2015) preserves a comparison between Doc2Vec and other algorithms such as LDA (Latent Dirichlet Allocation). They run their models on arXiv and the Wikipedia dataset. They demonstrated that Doc2vec algorithm produces better results than the other methods.

2.5 PageRank

Google developed an algorithm called PageRank (PR) in order to rank websites in the Google Search results. PageRank utilizes hyperlinks of webpages to find the importance score of them. PageRank considers the number and the quality of the links and figures out how important and valuable a webpage is. It is assumed that the more a website receives links, the more it is important.

Several applications related with PageRank is available both in literature and in businesses. PageRank algorithm was reviewed and associated to some updated previous algorithms in the field of information retrieval (Franceschet, 2010).

In another study, Wang, Liu and Zhao (2012) proposed a method to estimate the power of a person in a specific group and optimize the function of personal preferences. Their algorithm depends on seeking the potential needs of the users. In the end, the results showed that their model improves the prediction accuracy of the group recommendation.

In addition, the study by Şora (2015) uses PageRank to classify the important objects in a group. Their model is based on dependencies structure of the system. Different dependency styles are identified in the model and the model finds the optimal way of generating a system graph.

CHAPTER 3

METHODOLOGY

In this study, a new recommendation system is proposed for recommending potential articles to researchers. The proposed method leverages deep neural networks for encoding article text data to embedding vectors, specifically Siamese Bidirectional LSTM and combines this approach with social network analysis to utilize co-publication and citation networks using Node2Vec method.

To begin with, ArnetMiner (AMiner) dataset from aminer.org was chosen to perform the analysis due to the massive size of the data and the variety of the content (Tang, et al., 2008). It contains useful information such as textual information, citation and co-publication relationship. A ready data set is an advantage, but it needs to be cleaned and formatted. After preprocessing operations, the data was split into three parts. The first part of the dataset is allocated to the training data for training the model, the second part is the validation data which will soon be used for the optimization of hyperparameters. Lastly, the third part is allocated for the test data in order to evaluate the model.

3.1 Data preprocessing and preparation

3.1.1 Data acquisition: AMiner dataset

AMiner dataset is freely available on the web (aminer.org) and just designed for research purposes (Tang, et al., 2008). Figure 1 shows a web screenshot of dataset. Each paper in the dataset includes title, abstract, year of publication, authors, and venue information. The citation data is obtained from ACM, DBLP and other similar sources. Basically, article information consists of the following topics: Information

Fusion, Machine Learning, Database Systems, Web Mining, Web Services, Data Mining, Association Rules, Description Logics, Semantic Web, XML Data, Information Retrieval. The dataset can be used for text analysis like topic modeling, finding the most effective articles, clustering with the network and side information.

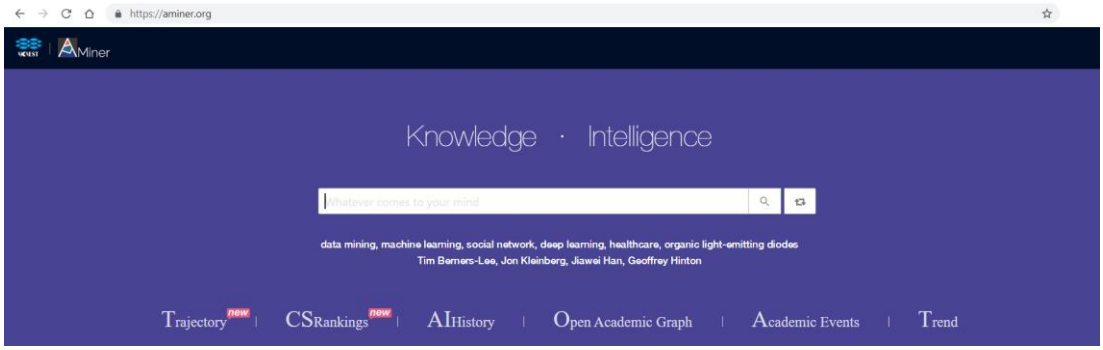

Figure 1. AMiner dataset acquisition

3.1.2 Data structure

AMiner dataset contains four data files which includes article information, citation network, co-publication network and article-author network. Table 1 denotes article information that contains all text information (title and abstract) related with an article. Table 2 shows the citation network. The first article column denotes the citing article and the second column is cited article. Table 3 demonstrates the co-publication network. The author in the first column is in a collaboration relationship with the author in the second column. The number of the articles published by the same pair of authors are considered as “Score” in the table. These scores will be normalized according to min-max normalization later on. Table 4 shows article-author network. It denotes the list of articles published by authors.

Table 1. An Example of Article Information Data

Article_ID	Title	Year	Abstract
316504	Bluetooth revealed: the insider's guide to an open specification for global wireless communication	2001	The authoritative guide to Bluetooth! From two contributors to the Bluetooth...
320182	Does Code Decay? Assessing the Evidence from Change Management Data	2001	A central feature of the evolution of large software systems is that change...
320198	Toward a Mathematical Foundation of Software Engineering Methods	2001	The development of large software systems consists of a sequence of modeling...
641546	Branch Classification to Control Instruction Fetch in Simultaneous Multithreaded Architectures	2002	Advances in semiconductor technology have several impacts on processor desi...
998993	Application of the variational iteration method to the regularized long wave equation	2007	This paper applies the variational iteration method to the regularized long...
998994	Comparison between the homotopy perturbation method and the sine-cosine wavelet method for solving linear integro-differential equations	2007	This paper compares the homotopy perturbation method with the sine-cosine w...

Table 2. An Example of Citation Network Data

Article_ID_1	Article_ID_2
320428	320425
320791	320780
320845	320848
320853	489137
323738	323756
323756	323738
324474	341704
324529	402070
324550	719935
324587	324637

Table 3. An Example of Co-Publication Network Data

Author_ID_1	Author_ID_2	Score
356	264972	3
356	286911	2
356	740193	10
611	909811	1
611	1247163	1
708	307040	21
2826	1020336	5
2927	1215646	2

Table 4. An Example of The Articles Published by Authors Data

Article_ID	Author_ID
316504	216235
320182	62687
320182	127427
320182	585809
320185	338624
320189	1098173
320198	859703
320199	143234
320199	902979
320415	641067

3.1.3 Data cleaning

Some of the articles in the dataset lack of abstract and citation information. Since the abstracts will be used as an input in the developed model, the articles with no abstract value were removed from dataset in the first place. Also, the records with missing values of year and author were removed.

The articles in the dataset was published between 1951 and 2014. In this study, the articles published after 2001 were selected and used due to excessive amount of missing data and lack of citation information before that time period.

The dataset was split into three parts: the articles published between 2001 and 2009, inclusive, were defined as the training set, the articles published in 2010 were

defined as the validation set, finally the articles published after 2010 till 2014 were defined as the test set.

For the training data, the first elimination is done by selecting only the articles that has at least one cited article. Since the citation network will be used for training the Siamese-LSTM architecture, the articles that had never cited any of the articles cannot be used in the model. Afterwards, the authors whose publications have at least 6 citations in total were selected within the articles published in between 2001 and 2009. Let's assume that an author published 5 articles in the training set and the number of the citations of each article are {0, 1, 3, 7, 2} respectively. The first article is removed at the first elimination operation with zero number of citations. Since the total number of cited articles was 13 which is greater than 6, we kept the last 4 articles. Table 5 denotes the number of records for each attribute in the cleaned training dataset.

Table 5. Number of the Records in the Cleaned Training Dataset

Description	Value
Number of Authors	163,468
Number of Articles	218,271
Number of Author-Author Pairs	886,213
Number of Article-Article Pairs	796,695
Number of Article-Author Pairs	1,243,776

3.2 Approach

The recommendation task is to acquire an ordered set of potential articles R = {d1, d2, …, dk} to an author-of-interest.

First of all, an author-of-interest is selected. While determining the author-of-interest, one needs to be sure that this author must have published articles in the training, validation and the test data. According to the past publications of the author, potential articles similar to these past publications are found based on the text similarity scores obtained by the Siamese BiLSTM algorithm. Likewise, potential articles might have been published by some of his co-authors. It is known that authors with akin behavior is more likely to be influenced with each other. In that case, an additional co-publication similarity score from Node2Vec algorithm can be attached to the model. On the other hand, some articles could be more important than the others due to the number and importance of their citing articles. To emphasize that state, Node2Vec and PageRank algorithms will be applied to the citation network and the similarity scores will be added to the model. Ranked total of these text-embedding, co-publication and citation similarity scores gives the ordered set of potential articles for the author-of-interest. The coefficients used to combine these different scores are determined considering the success of the recommendations based on the validation data. The evaluation of this approach will be estimated by checking whether the author-of-interest cited any of these recommended articles in the future (test dataset). More detailed information is adverted in the following sections. Figure 2 shows the flow diagram.

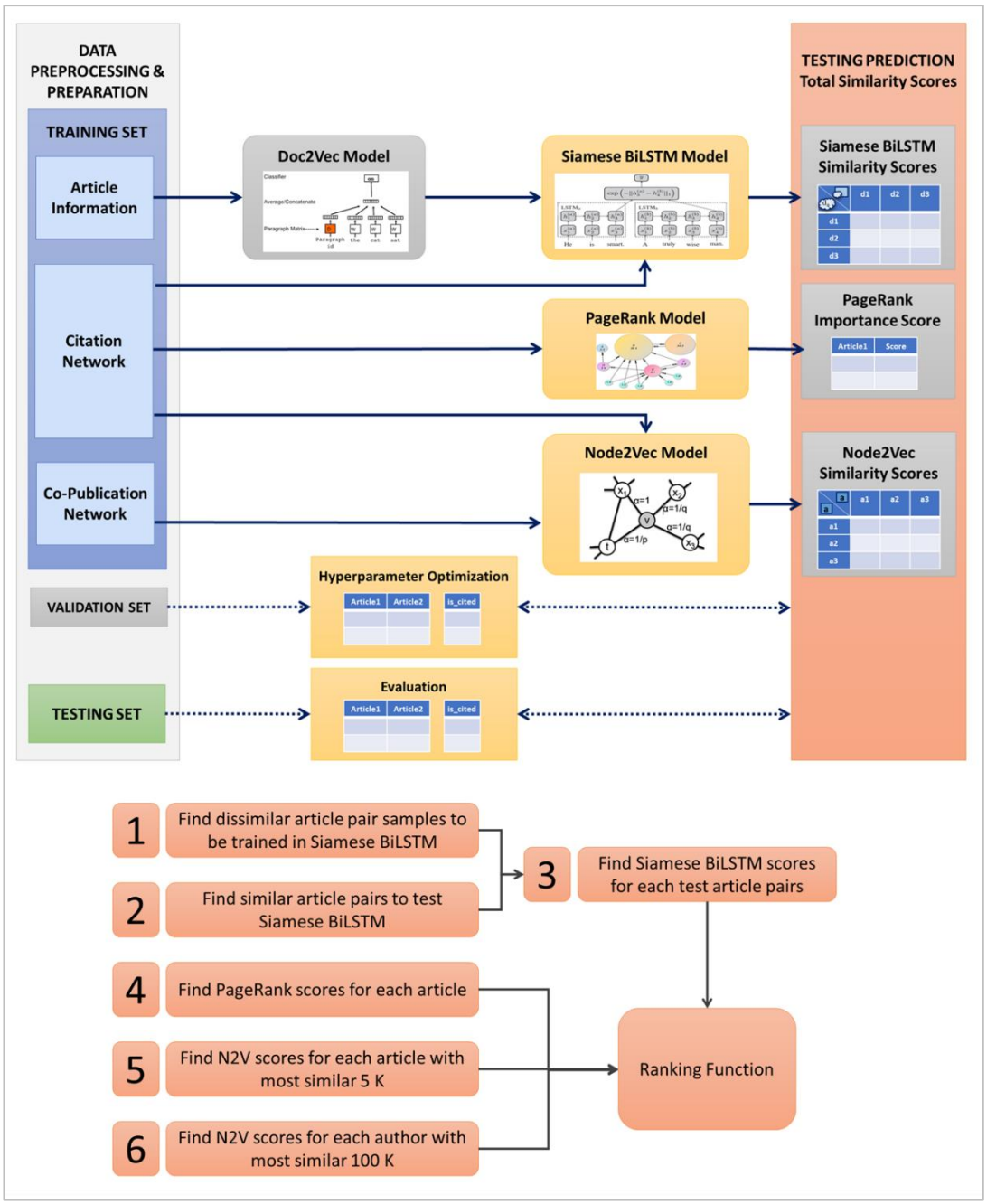

Figure 2. Flow diagram of the proposed model

3.3 Experimental setting

As it is mentioned in the previous sections, data was split into three parts as training, validation and test. Figure 3 illustrates the splitting operation.

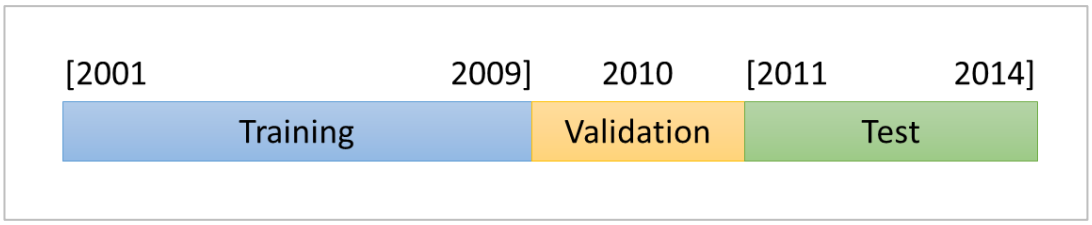

Figure 3. Data splitting

The aim of the study is to recommend articles from training set to an author-of-interest. So, before running all the models, selection of the author-of-interest group is crucial for running the validation and test phases of the algorithms. In order to determine the author-of-interest group the following steps were taken:

1. To evaluate the proposed model, authors that had published articles in both of the training, validation and test datasets must be selected in the first place.
2. These authors must have a substantial number of cited articles in validation and test sets to make a proper evaluation. To ensure that state;
    a. First, cited articles in the validation set are determined for each author. Some of these articles might have been cited by the same author in the training set as well. So, these already cited articles are extracted from the cited articles list of the validation set. The number of the cited articles in validation set are calculated for each author. The authors that has given citations to 15 to 30 articles that are in the training period, in their articles written in the validation period are selected. Since the validation data includes articles written in just one-year, 2010, lower and upper limits are set as 15 and 30, respectively.
    b. Second, cited articles in the test set are determined for each author. Some of these articles might have been cited by the same

author in the training and validation periods as well. So, these articles are extracted from cited articles list of the test data set. The number of cited articles in the test set are calculated for each author and the authors that have given citations to 70 to 100 articles in the training period only are selected. Since the test data includes the data between 2011 and 2014, lower limit is set as 70 while upper limit kept at 100.

c. Finally, authors that meet both aforementioned conditions are selected.

Ultimately 136 authors were obtained according to these conditions and chosen as the author-of-interest group. Figure 4 illustrates the determination process of author-of-interest group.

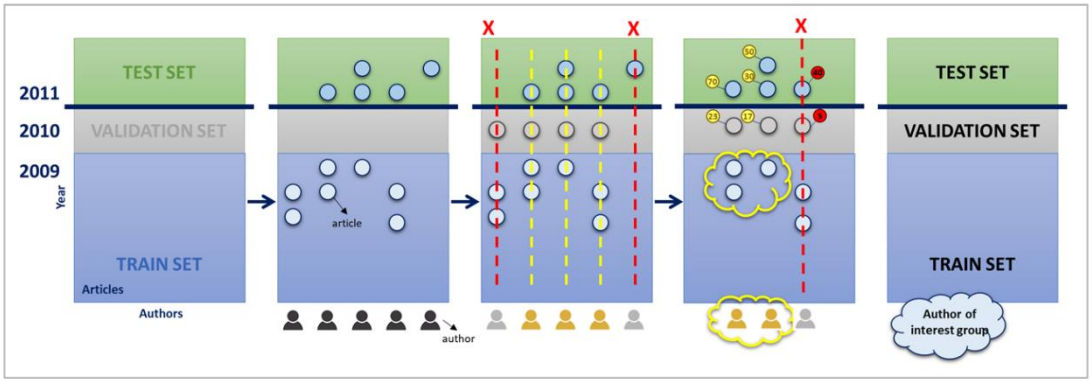

Figure 4. Determination of author-of-interest group

136 authors published 1,855 articles in total in the training period. They had cited 3,070 articles in validation set and 10,981 articles in test set. 3,070 articles are used to optimize hyperparameters of the ranking function, which will be explained in detail in the model validation section. 10,981 articles are used in comparison with the recommended articles in model evaluation phase.

Figure 5 shows the data structure that is used in the models. The number of the records in Table 5 can be seen in Figure 5 as training inputs. The data of the author-of-interest group is in the test inputs part. Data icons in grey color belongs to test data set and the blue icons are from training data set.

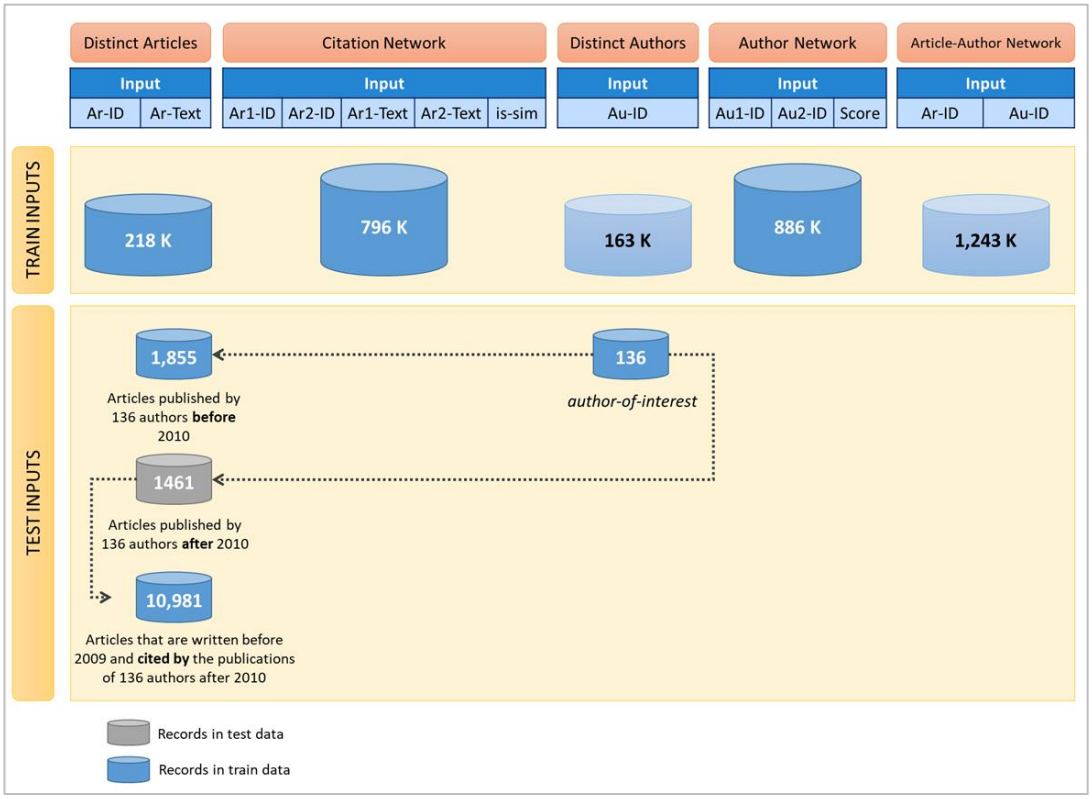

Figure 5. Model inputs

3.3.1 Siamese BiLSTM network

Siamese BiLSTM is a type of neural network that includes two generic sub-networks. Generic statement refers to that these twin sub-networks have the same composition with the same weights and parameters. Siamese BiLSTM is implemented on keras library to capture the article similarities using embedding vectors.

The architecture of Siamese BiLSTM model is shown in Figure 6. The model has four layers including the text input layer, embedding layer, BiLSTM encoder layer and Siamese dense layer. Text inputs are given to the model in pairs. Then,

they are transformed to the document vectors in the embedding layer. In BiLSTM layer, the pair of input documents are passed through the twin networks respectively and turned into two different feature vectors. So, the distance of these two feature vectors can be calculated in Siamese Layer (Cui, Pan, & Liu, 2019).

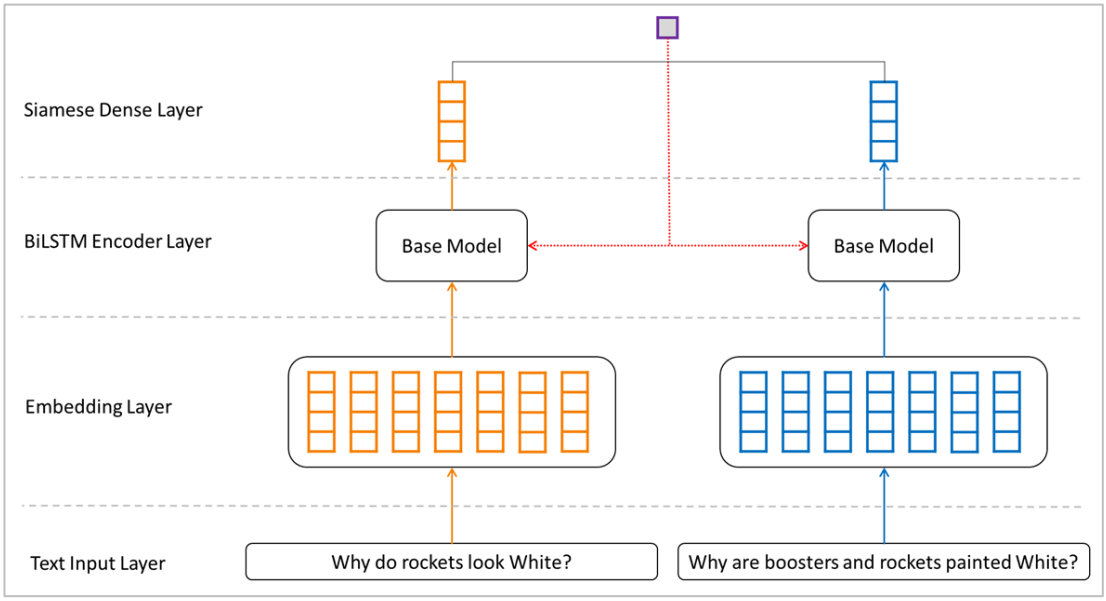

Figure 6. Siamese BiLSTM model architecture

Siamese BiLSTM model processes the inputs in pairs. Figure 7 is an example of input data used in the previous studies. “Sentence_1” and “Sentence_2” are text inputs. Boolean “is_similar” field indicates whether these two sentence pairs are similar or not. This parameter should be labeled by someone beforehand as one or zero.

In this study, text inputs are the concatenation of the article title and abstract information which includes 130 words on average. Since the citation network gives an idea about the similarity of the article pairs, a pair of citing and cited articles are assumed as similar. However, similar inputs fall short of running the model. Dissimilar article pairs are also needed. Therefore, Doc2Vec algorithm is used for finding the dissimilar samples to be used as input in the Siamese BiLSTM algorithm.

Sentence_1	Sentence_2	is_similar
Shoul I buy tiago?	What keeps children active and far from phone?	0
How can I be a good physiologist?	What should I do to be a great physiologist?	1
How do I read any of my facebook posts?	How can I see all of my facebook posts?	1
Which fish can survive in sweet water?	What is web application?	0
What is best way to make Money online?	What is best way to ask for Money online?	0

Figure 7. Siamese BiLSTM model input

3.3.2 Finding dissimilar samples using Doc2Vec

Doc2Vec is an unsupervised method for learning the distributed representations of documents and sentences. Unlike Siamese BiLSTM, it is not necessary to label the input data as similar or dissimilar. The input is the list of documents. Doc2Vec represents the documents as a vector and calculates the similarity scores for each document in the similarity matrix using the cosine similarity metric. Figure 8 illustrates the model.

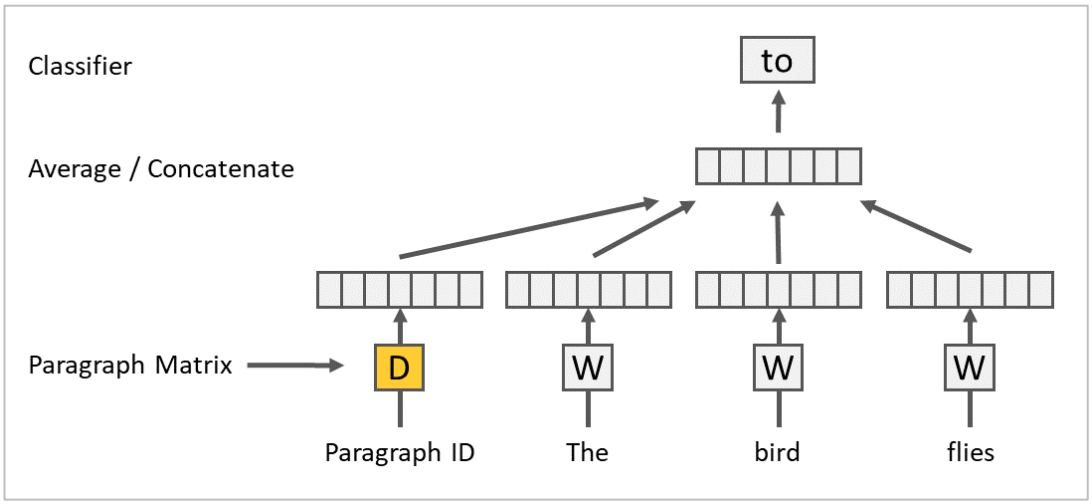

Figure 8. Doc2Vec model architecture

Since, Doc2Vec is capable of calculating the similarities for each document combination. The pairs with minimum similarity score could be chosen as dissimilar samples to be used to train the Siamese BiLSTM algorithm.

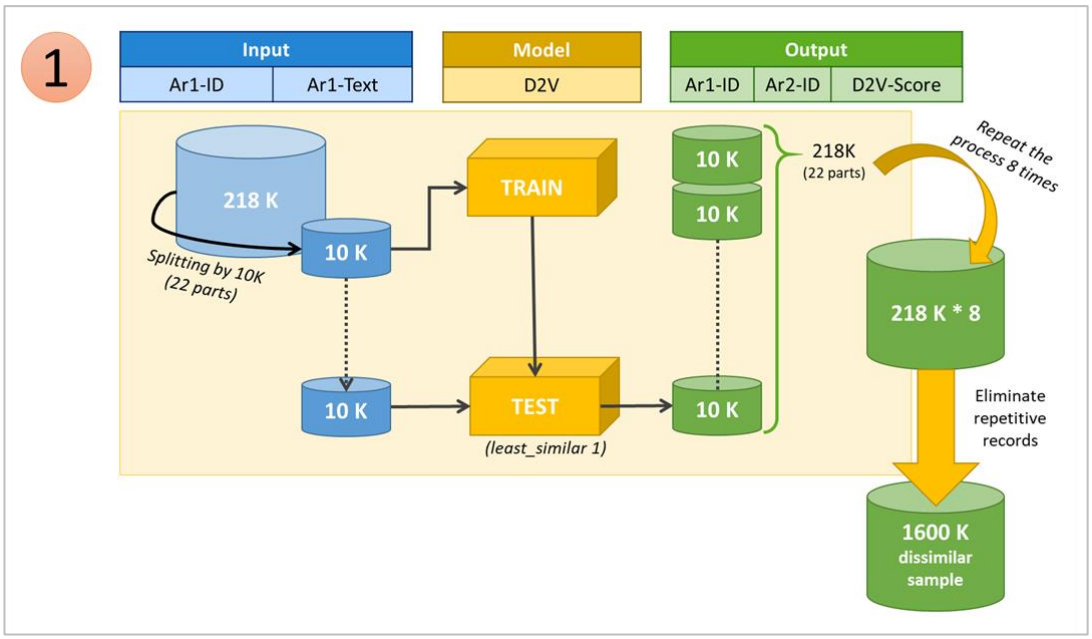

Figure 9. The process of finding dissimilar samples using Doc2Vec

Figure 9 shows the diagram of obtaining dissimilar article pairs. 218K distinct article text input was used to train the Doc2Vec model. Doc2Vec has the function of getting the "most similar (n)" items. Unfortunately, there is no function to find the least similar articles. By using the similarity matrix, the pairs with the least similarity score could be found.

Since the size of the article list was quite big, running the model to create a 218K x 218K matrix took a long time. So, the data of 218K article was split into 22 parts with the size of 10K articles. Doc2Vec model was separately trained for each 10K-article text data to get 10K x 10K matrix with the similarity scores and the least similar 1 article was found for each article. Figure 10 illustrates the matrix presentation. To protect diversity, instead of choosing the least similar(n) article among 10K-input data, the least similar articles in the other 10K-article groups were investigated. The order of 218K articles was randomly ranked and split into 22 parts again. This splitting operation was performed 8 times and approximately 1,700K dissimilar article pairs were obtained in total. Due to the random sorting of the input

data, some duplicate records occurred. After the removal of the duplicate records, finally, approximately 1,600K dissimilar pair inputs were ready to be fed into the Siamese BiLSTM algorithm to train the model.

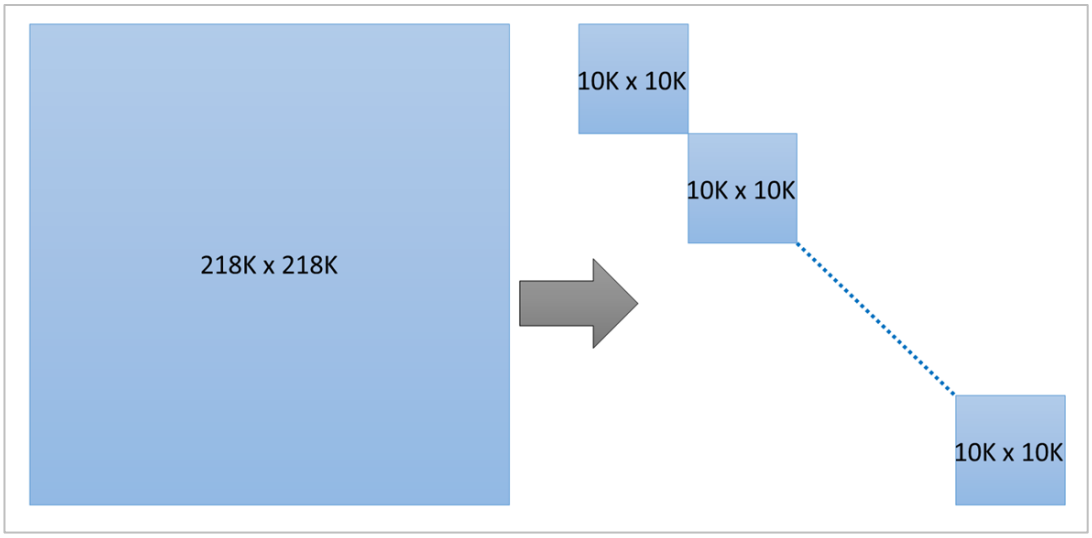

Figure 10. Representation of similarity matrixes

3.3.3 Finding potential similar article pairs using Doc2Vec

Siamese BiLSTM model does not include the most similar function. The article pairs which might have been similar, was tested within the model. Therefore, it was necessary to find the potential articles similar to the articles published by the author-of-interest group. To achieve this task, Doc2Vec algorithm was used. As mentioned before, Doc2Vec contains most_similar(n) function and does not require input pairs. So, 218K articles with text information were trained in the Doc2Vec model and 1,855 articles were tested with the number of 5K most similar articles. Finally, 9.3M potential pairs were ready to be sent to test Siamese BiLSTM model. Figure 11 summarizes the process.

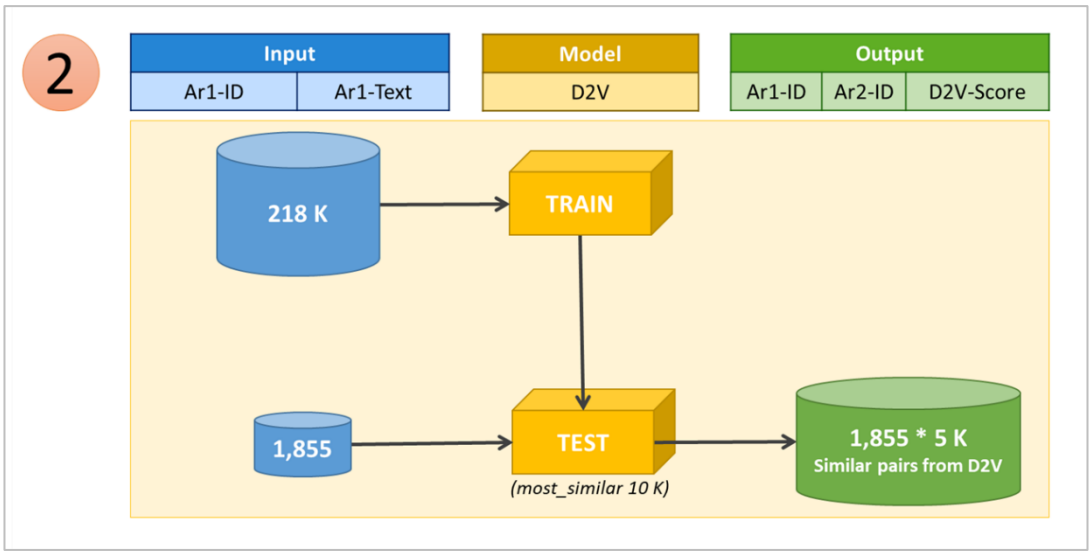

Figure 11. The process of finding potential similar article pairs using Doc2Vec

3.3.4 Finding text similarity scores using Siamese BiLSTM

Citing and cited articles in the citation network are used as similar (is-sim = 1) pairs in the training input. But all citations of an article may not be similar to it. For example, while some core articles are referred to several times in an article, some exists just once. Unfortunately, entire text document of an article does not exist in AMiner dataset. So, according to the PageRank algorithm, as mentioned in section 3.3.2, articles whose importance scores are less than 0.03 were eliminated. Hence, the number of citation record dropped from 796K to 730K.

Dissimilar article pairs (is-sim = 0) are gathered from Doc2Vec model by searching the least similar value in similarity matrix. Table 6 shows an example of training input.

Table 6. Training Input Example of Siamese BiLSTM Algorithm

Ar1-ID	Ar1-Text	Ar2-ID	Ar2-Text	is-sim
316504	An open specification for global wireless communication	641546	Branch Classification to Control Instruction Fetch…	1
320182	Assessing the Evidence from Change Management Data	998993	Application of the variational iteration method to the…	0
320198	Toward a Mathematical Foundation of Software…	998994	Comparison between the homotropy perturbation…	1
998993	Application of the variational iteration method to the…	316504	An open specification for global wireless…	0
641546	Branch Classification to Control Instruction Fetch…	320182	Assessing the Evidence from Change Management Data	0
998993	Application of the variational iteration method to the…	320198	Toward a Mathematical Foundation of Software…	1

Before model building, some preprocessing operations were performed on training data such as removal of stop words, stemming and tokenization. After that, 2,330K text inputs were transformed to the embedding vectors. Embedding vectors were delivered to Siamese BiLSTM network and model was trained with the following parameters: Layer_size = 128 and validation_split = 0.1

9.3M outputs gathered from Doc2Vec model for finding potential articles were delivered to Siamese BiLSTM model to get text similarity scores for each article pair. Finally, the text similarity scores of the articles were provided. Figure 12 summarizes the process.

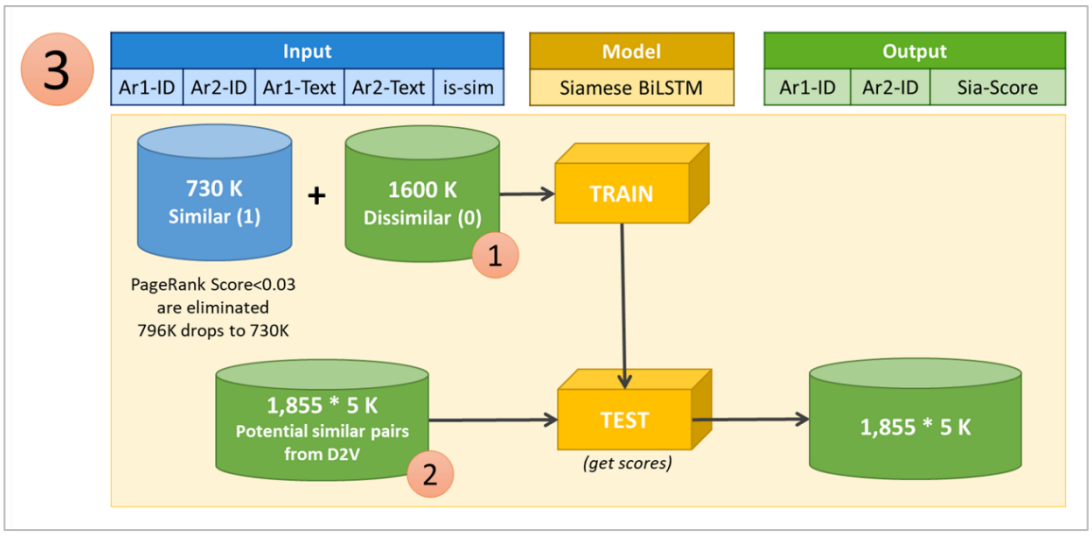

Figure 12. The process of finding text similarity scores using Siamese BiLSTM

3.3.5 Finding PageRank scores

The importance of an article can be determined by the number and importance of the articles referring to it. PageRank algorithm utilizes the relationship between articles using the citation network.

The algorithm considers the number and the quality of citations and figure out how important and valuable an article is. Figure 13 illustrates the process.

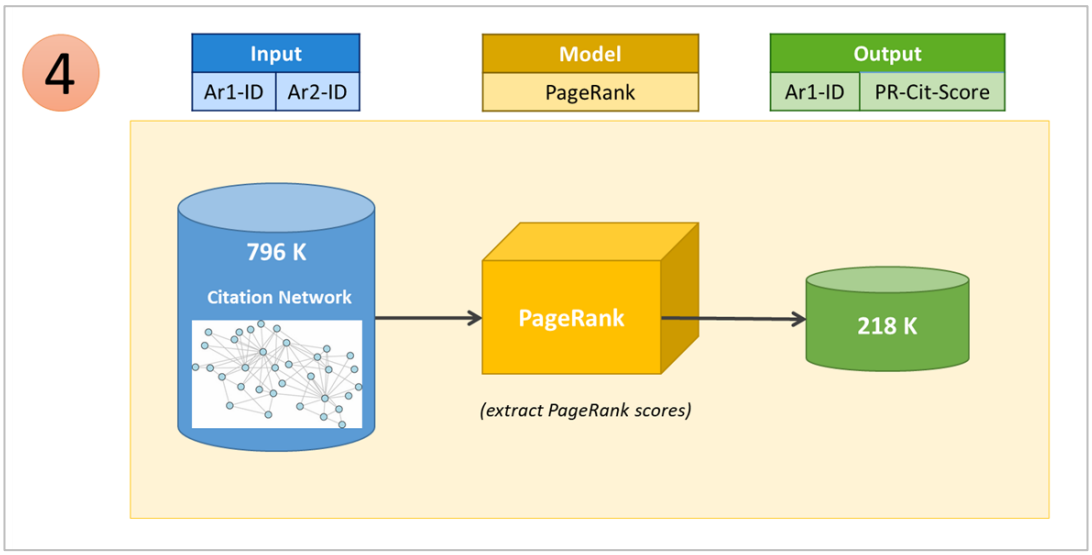

Figure 13. The process of finding importance scores of articles using PageRank

The entire citation network with the number of 796K links was delivered to PageRank model. Right after that, the model was trained, and PageRank scores were acquired for all distinct articles (218K articles). These scores are normalized according to min-max normalization afterwards.

3.3.6 Finding article network similarity scores utilizing the citation network

Node2Vec is an analytical framework that learns continuous element representations for nodes in networks. In order to maximize the probability of protecting the network neighborhoods of the nodes, Node2Vec learns to map the nodes to a low-dimensional space of features.

The entire citation network with 796K links among the articles was trained within the Node2Vec algorithm and then the most similar 10K articles were captured. Finally, 18M article pairs with Node2Vec scores were obtained. Table 7 shows an example of the model input. Additionally, Figure 14 illustrates the process.

Table 7. Training Input Example of Node2Vec Algorithm (Article)

Ar1-ID	Ar2-ID	is-sim
316504	641546	1
320182	995993	1
320198	998994	1
198993	316304	1
641246	325182	1
998993	320198	1

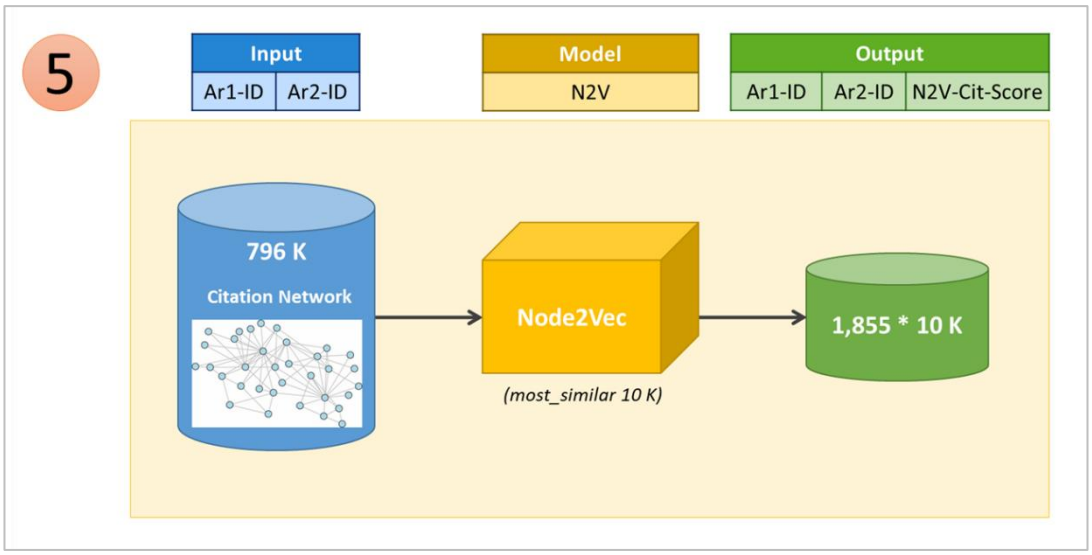

Figure 14. The process of finding article similarity scores using Node2Vec

3.3.7 Finding author network similarity scores utilizing the co-publication network

In this part, the underlying assumption is that authors tend to have similar tastes with their co-authors. So, a similarity score can be calculated between authors using the Node2Vec algorithm. The co-publication network includes 886K author relationship records. Table 8 shows an example of the model input. The score field stands for the number of the articles published by both authors. Before training the model, these scores were normalized according to min-max normalization. After this operation Node2Vec model was trained. Figure 15 illustrates the process.

Table 8. Training Input Example of Node2Vec Algorithm (Author)

Au1-ID	Au2-ID	Score
862457	432432	0.987
195645	862552	0.568
433465	343862	0.232
139786	238625	0.743
764532	186255	0.111
244565	986200	0.100

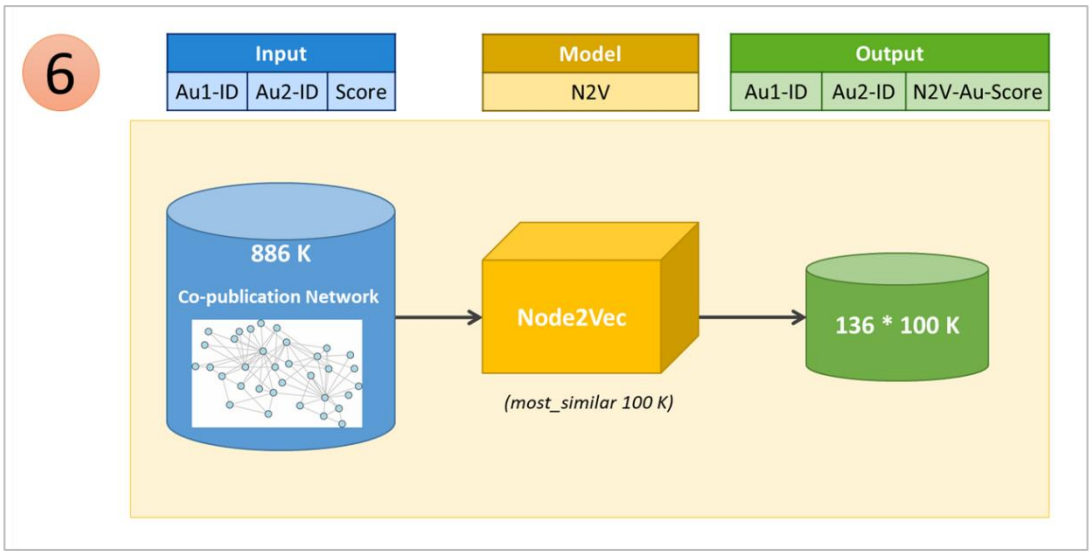

Figure 15. The process of finding author similarity scores using Node2Vec

3.4 Model aggregation

All model outputs were gathered into a single score. Since the main focus was on text similarity scores from Siamese BiLSTM model, aggregation process was built on text similarity output. First, Node2Vec similarity scores from the citation network joined by left outer method. Secondly, the authors of citing articles were joined. Thirdly, the authors of the cited articles were joined to the aggregate output. In the fourth step, Node2Vec similarity scores of each author pairs were joined. Lastly, the importance calculated by the PageRank algorithm were added to the model. Figure 16 shows the steps of the model aggregation.

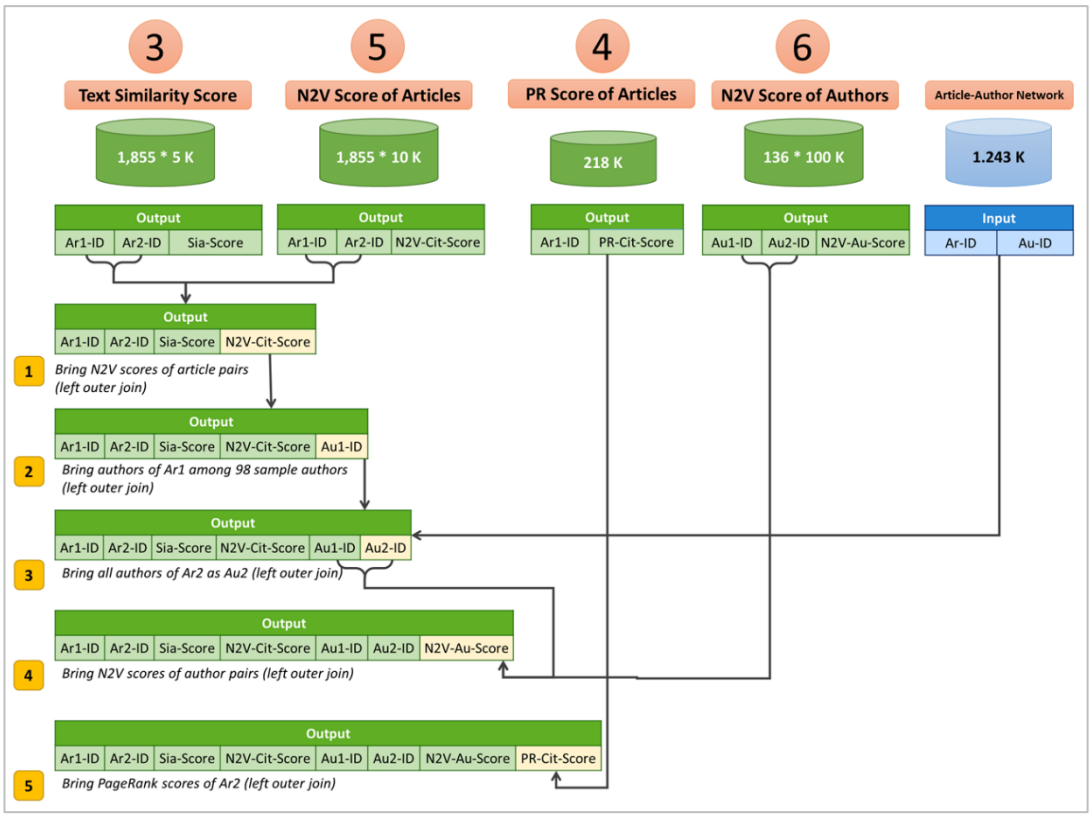

Figure 16. Steps of model aggregation

In the aggregate output table, the author-of-interests (Au1), their published articles before 2010 (Ar1), potential similar articles to these articles (Ar2) and their authors (Au2) were denoted. Since the recommendation task is to find potential articles for author-of-interest group, Ar1 and Au2 columns can be extracted. These columns were used to get similarity scores to the aggregate table and there is no need to keep them. Output table was grouped by according to Ar2 and Au1 and the scores with the maximum value were selected for each group. For instance, article with ID "1015533" was recommended to an author-of-interest "8723" due to two different articles published by him (1965289 and 55535). When Ar1 column is deleted, article "1015533" exists twice in the data table. So, maximum Sia-Score and N2V-Cit-Score were selected (0.889 and 0.791 respectively). Table 9 denotes the aggregate table.

Table 9. An Example of Aggregation Table

Ar1-ID	Ar2-ID	Sia-Score	N2V-Cit-Score	Au1-ID	Au2-ID	N2V-Au-Score	PR-Cit-Score
882606	3411153	0.987	0.773	8723	5232	0.773	0.561
1965289	1015533	0.786	0.791	8723	4338	0.837	0.781
55535	1015533	0.889	0.556	8723	4546	0.005	0.781
747168	18145	0.433	0.298	1088	5645	0.000	0.001
994937	700422	0.908	0.000	1088	3435	0.609	0.902
911896	5300435	0.998	0.753	1088	5232	0.773	0.719

After grouping operation, maximum values of similarity scores were stated in the final aggregate table. Table 10 denotes an example of a final aggregate table. Figure 17 summarizes the grouping by process.

Table 10. An Example of Final Aggregation Table

Ar2-ID	Max (Sia-Score)	Max (N2V-Cit-Score)	Au1-ID	Max (N2V-Au-Score)	PR-Cit-Score
3411153	0.987	0.773	8723	0.773	0.561
1015533	0.889	0.791	8723	0.837	0.781
18145	0.433	0.298	1088	0.000	0.001
700422	0.908	0.000	1088	0.609	0.902
5300435	0.998	0.753	1088	0.773	0.719

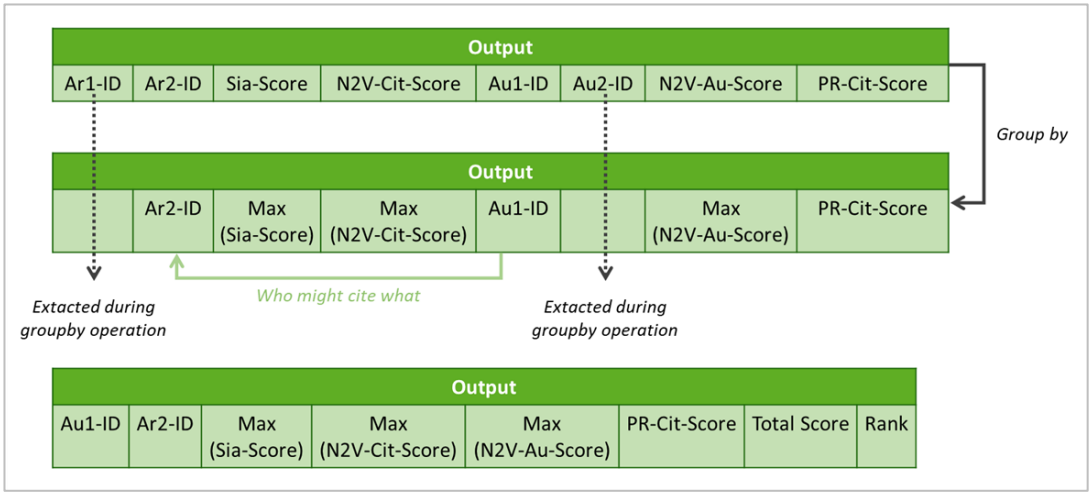

Figure 17. Group by process on aggregate table

3.5 Model validation

According to the final aggregate table, a total score was calculated. According to this total score, top-N potential articles are selected and recommended to authors. Total Score depends on text similarity score (Sia-Score), co-publication similarity score (N2V-Au-Score), citation similarity score (N2V-Cit-Score) and the importance score of the articles (PR-Cit-Score). All four of these scores have already been calculated in aggregate table. On the other hand, it is worth mentioning that all scores except PageRank score were between 0 and 1. Therefore, PageRank score was normalized according to min-max normalization method. In order to calculate Total Score, these scores were multiplied with their corresponding coefficients that are going to be optimized, as is shown in the formula (1) below:

$$\text{Total Score} = a \times \text{Max(Sia-Score)} + b \times \text{Max(N2V-Au-Score)} + c \times \text{Max(N2V-Cit-Score)} + d \times \text{PR-Cit-Score} \quad (1)$$

Here, the hyperparameter (coefficient) of text similarity score (Sia-Score) is “a”, the hyperparameter of co-publication similarity score (N2V-Au-Score) is “b”, the hyperparameter of citation similarity score (N2V-Cit-Score) is “c” and the hyperparameter of importance score of the articles (PR-Cit-Score) is “d”.

The combination of two hyperparameter sets {1, 2, 3, 4} and {1, 2, 3, 10} are used to find the best hyperparameter combination that gives the highest recall rate while recommending 100 articles for each author. According to these parameters 24 different combinations for both hyperparameter sets are generated. For instance, a combination list for the hyper parameter set {1,2,3,4} and {1,2,3,10} is shown below.

[(1,2,3,4), (1,2,4,3), (1,3,2,4), (1,3,4,2), (1,4,2,3), (1,4,3,2), (2,1,3,4), (2,1,4,3), (2,3,1,4), (2,3,4,1), (2,4,1,3), (2,4,3,1), (3,1,2,4), (3,1,4,2), (3,2,1,4), (3,2,4,1), (3,4,1,2), (3,4,2,1), (4,1,2,3), (4,1,3,2), (4,2,1,3), (4,2,3,1), (4,3,1,2), (4,3,2,1), (1,2,3,10), (1,2,10,3), (1,3,2,10), (1,3,10,2), (1,10,2,3), (1,10,3,2), (2,1,3,10), (2,1,10,3), (2,3,1,10), (2,3,10,1), (2,10,1,3), (2,10,3,1), (3,1,2,10), (3,1,10,2), (3,2,1,10), (3,2,10,1), (3,10,1,2), (3,10,2,1), (10,1,2,3), (10,1,3,2), (10,2,1,3), (10,2,3,1), (10,3,1,2), (10,3,2,1)]

48 different Total Score values were calculated for all authors in author-of-interest list. With the help of the validation data, the best hyperparameter combination that gave the best result was.

Table 11. Determination of Total Score with Optimized Hyperparameters

Ar2-ID	Au1-ID	Max (Sia-Score)	Max (N2V-Cit-Score)	Max (N2V-Au-Score)	PR-Cit-Score	Total Score
336874	1468309	0.93	0.07	0.28	0.47	6.58
572314	1468309	0.91	0.40	0.43	0.28	5.77
875064	1468309	0.99	0.82	0.94	0.02	5.61
1206687	1468309	1.00	0.82	0.93	0.01	5.57
344447	1468309	0.99	0.09	0.00	0.44	5.56
1688287	1468309	1.00	0.83	0.86	0.02	5.50
810101	1468309	1.00	0.83	0.86	0.02	5.43
1733302	1468309	1.00	0.83	0.83	0.02	5.34
1016135	1468309	1.00	0.64	0.87	0.04	5.33
654442	1468309	1.00	0.42	0.88	0.08	5.31

Table 11 shows top-10 recommended article list according to Total Score values calculated for a single hyperparameter combination (a=1, b=2, c=3, d=10) for a single author-of-interest “1468309”. This calculation is implemented for all authors in author-of-interest list and the success result was confirmed by the validation data. After calculating Total Score with 48 different combination of hyperparameters, the best hyperparameter combination that maximizes the recall rate at the level of 24% is

obtained as {a=1, b=2, c=3, d=10}. The final formula for calculating Total Score is as follows:

$$\text{Total Score} = 1 \times \text{Max(Sia-Score)} + 2 \times \text{Max(N2V-Au-Score)} + 3 \times \text{Max(N2V-Cit-Score)} + 10 \times \text{PR-Cit-Score} \quad (2)$$

3.6 Model evaluation

With the help of validation process, hyperparameters of Total Score formula were determined. Total Score was calculated for each article in final aggregate table and ranked in descending order for each author-of-interest as shown in Table 12.

In this table, the top-10 recommendations for an author-of-interest are listed as an example. As it is seen in the table, Au1-ID denotes the author-of-interest and Ar2-ID denotes the recommended article with similarity scores and Total Score. Total Score is ranked in descending order and top-20 articles are selected and recommended to the author-of-interest.

Table 12. Model Output

Ar2-ID	Au1-ID	Max (Sia-Score)	Max (N2V-Cit-Score)	Max (N2V-Au-Score)	PR-Cit-Score	Total Score	Rank
576214	48804	0.997	0.457	0.193	0.375	6.511	1
1719401	48804	0.999	0.885	0.677	0.052	5.539	2
1105452	48804	0.999	0.963	0.705	0.018	5.483	3
1412517	48804	0.998	0.948	0.707	0.021	5.471	4
424010	48804	0.999	0.463	0.000	0.298	5.370	5
1193147	48804	0.999	0.921	0.705	0.018	5.357	6
525120	48804	0.997	0.644	0.600	0.122	5.355	7
1098363	48804	0.989	0.941	0.677	0.015	5.323	8
904968	48804	0.996	0.912	0.705	0.016	5.304	9
1193146	48804	0.999	0.899	0.724	0.014	5.287	10
1915963	48804	0.999	0.862	0.741	0.016	5.236	11
1128858	48804	0.999	0.879	0.701	0.018	5.223	12
1732706	48804	0.999	0.851	0.653	0.034	5.208	13
782613	48804	0.999	0.842	0.708	0.021	5.159	14
1727943	48804	0.999	0.833	0.741	0.014	5.127	15
1390037	48804	0.999	0.961	0.533	0.016	5.115	16
1670096	48804	0.990	0.935	0.580	0.015	5.108	17
1154800	48804	0.999	0.908	0.600	0.017	5.098	18
1707071	48804	0.996	0.776	0.688	0.038	5.085	19
1195800	48804	0.999	0.826	0.705	0.019	5.083	20
1195800	48804	0.999	0.826	0.705	0.019	5.083	20

In order to evaluate the model, top-N recommended articles are compared with the articles cited by an author-of-interest in test data set. If a recommended article is cited by an author-of-interest's publication that is written after 2010 (test period), this recommendation is assumed as successful.

Figure 18 summarizes the determination of successfully recommended articles. First, top-N articles were recommended to an author-of-interest. Second, successfully recommended articles are determined according to citations in the test set.

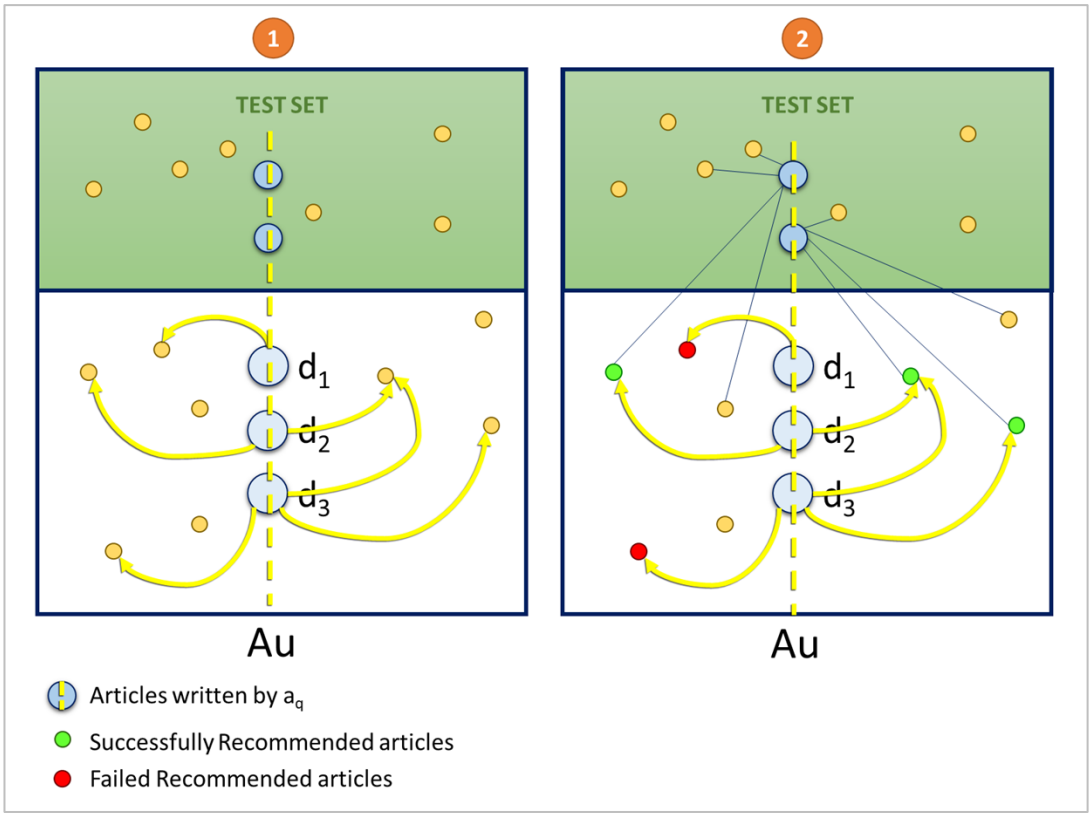

Figure 18. Determination of successfully recommended articles

There are two metrics used in this study to measure the performance of the model, Precision and Recall. Precision and Recall rates are calculated by the formulas (3) and (4):

Recall = [The number of successfully recommended articles] / [The number of citations in the test set] (3)

Precision = [The number of successfully recommended articles] / [The number of recommended articles] (4)

These metrics are calculated for each author. For instance, assume an author-of-interest "58616" had cited 70 articles in the set. Let's assume that N is chosen as 10 and thus 10 articles are recommended. And seven of them are actually cited by the publications of this author in the test set. So, recall rate is calculated as 7 / 70 = 10% and precision rate is calculated as 7 / 10 = 70%.

In order to calculate recall and precision rate of the entire model, instead of averaging recall and precision scores of each author-of-interest, recall and precision rate are calculated according to total number of recommended and cited articles.

CHAPTER 4

RESULTS AND FINDINGS

As mentioned before, Recall and Precision metrics were used for evaluation of the proposed model. Recall metric is the ratio between successfully predicted recommendations and the number of articles cited by an author-of-interest in test set. Precision metric is the ratio between successfully predicted recommendations and the number of the recommended articles. The recommended article list was selected in different ranges which are Top 20, Top 50 and Top 100 potential article counts.

A high recall ratio with a lower N indicates a better recommendation system. Table 13 shows Recall rates of the proposed model, the proposed model without author similarity score and the proposed model without any network score. Table 14 shows the precision rates.

Table 13. Recall Rates of Proposed Model

	Top-20	Top-50	Top-100
Proposed Model	2.2%	4.2%	6.8%
N2V-Au-Score is extracted	1.7%	3.7%	6.0%
N2V-Au-Score and N2V-Cit-Score are extracted	1.0%	1.6%	2.7%

Table 14. Precision Rates of Proposed Model

	Top-20	Top-50	Top-100
Proposed Model	8.8%	6.8%	5.5%
N2V-Au-Score is extracted	6.8%	6.0%	4.8%
N2V-Au-Score and N2V-Cit-Score are extracted	3.2%	2.6%	2.2%

Considering the results, in terms of the precision and recall metrics, it can be explicitly seen that integration of deep learning (Siamese BiLSTM) and node embedding (Node2Vec) methods outperformed other models where Node2Vec similarity scores were extracted. In another view, this shows that authors with akin behavior tend to have similar preferences, because existence of author similarity score in the model increases the success rate.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The use of recommender systems for extracting related papers have become vital due to the recent challenge of handling big data. Many of the article recommendation approaches have their own drawbacks. The quality of the recommended articles is generally compromised as only citation counts. The proposed approach considers both text similarity between the articles and the relationship between the authors and articles via networks. Total similarity score was calculated for each paper by using deep learning and social network analysis tools. Total Scores were ranked for each author-of-interest. Top-N articles were selected and recommended to the author-of-interest. Recommendation list was compared with the citations in the test list. The number of successfully predicted articles were obtained. In order to evaluate the model, Precision and Recall measures were calculated for each top-N choice. According to the results, adding network similarity scores to the model shows higher performance compared to the other experiments.

It is expected that as the number of recommended articles increases, precision will decrease and recall measure will increase. In many studies recommending the maximum number of items generally limited to 100. According to the results obtained, it is obvious that co-publication relationship effects authors' preferences. Utilizing the Node2Vec algorithm increased the recall rate by 7% at the N level of 100 compared to the model that just used Siamese BiLSTM and PageRank similarity scores.

Future work may include separating title and abstract texts and getting text embeddings separately. Because the effect of a title is bigger on selecting a newbie

article. On the other hand, it is known that the articles that were published in the reputable venues are highly preferable. Many studies use the power of the venues for their recommendation task. Venue information or article-venue network might have a positive effect on the recommendation systems' performances.

This study can be applied to the models that operates input pairs. Comments in Yemek Sepeti application can be analyzed using Siamese BiLSTM network. In order to make a comment in Yemek Sepeti, it is necessary to score the restaurant. Comments with the scores greater than 3 can be assumed as similar and the rest can be assumed as dissimilar pairs in a scale of 5. This approach can be beneficial for recommending restaurants or foods.

REFERENCES

Burke, R. (2007). Hybrid web recommender systems. Berlin, Heidelberg: Springer.

Chen, J., & Ban, Z. (2016). Literature recommendation by researchers' publication analysis. *2016 IEEE International Conference on Information and Automation (ICIA)* (pp. 1964-1969). Ningbo: IEEE.

Cui, Z., Pan, L., & Liu, S. (2019). Hybrid BiLSTM-Siamese network for relation extraction. *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (pp. 1907-1909). Montreal: International Foundation for Autonomous Agents and Multiagent Systems.

Dai, A. M., Olah, C., & Le, Q. V. (2015). Document embedding with paragraph vectors. *ArXiv, abs/1507.07998.*

Ding, Y., Zhang, G., Chambers, T., Song, M., Wang, X., & Zhai, C. (2014). Content-based citation analysis: The next generation of citation analysis. *Journal of the Association for Information Science and Technology, 65*(9), 1820-1833.

Doerfel, S., Jäschke, R., Hotho, A., & Stumme, G. (2012). Leveraging publication metadata and social data into folkrank for scientific publication recommendation. *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web* (pp. 9-16). New York: ACM.

Du, Z., Tang, J., & Ding, Y. (2018). POLAR: Attention-based CNN for one-shot personalized article recommendation. *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 675-690). Cham: Springer.

Franceschet, M. (2010). PageRank: Standing on the shoulders of giants. *ArXiv, arXiv:1002.2858.*

Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 855-864). New York: ACM.

Huang, W., Wu, Z., Liang, C., Mitra, P., & Giles, C. L. (2015). A neural probabilistic model for context based citation recommendation. *Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 2404-2410). Austin: AI Access Foundation.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *Proceedings of International Conference on Machine Learning*, (pp. 1188-1196).

Liou, C. H. (2016). Personalized article recommendation based on student's rating mechanism in an online discussion forum. *2016 49th Hawaii International Conference on System Sciences (HICSS)* (pp. 60-65). Washington: IEEE.

Liu, H., Kong, X., Bai, X., Wang, W., Bekele, T. M., & Xia, F. (2015). Context-based collaborative filtering for citation recommendation. *IEEE Access*(3), pp. 1695-1703.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pp. 3111-3119.

Mueller, J., & Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. *Proceedings of the 30th AAAI Conference on Artificial Intelligence* (pp. 2786-2792). Phoenix: AAAI Press.

Neculoiu, P., Versteegh, M., & Rotaru, M. (2016). Learning text similarity with siamese recurrent networks. *Proceedings of the 1st Workshop on Representation Learning for NLP* (pp. 148-157). Berlin: Association for Computational Linguistics.

Ren, X., Liu, J., Yu, X., Khandelwal, U., Gu, Q., Wang, L., & Han, J. (2014). Cluscite: Effective citation recommendation by information network-based clustering. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 821-830). New York: ACM.

Şora, I. (2015). A PageRank based recommender system for identifying key classes in software systems. *2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics* (pp. 495-500). IEEE.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). Arnetminer: extraction and mining of academic social networks. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 990-998). New York: ACM.

Tsolakidis, A., Triperina, E., Sgouropoulou, C., & Christidis, N. (2016). Research publication recommendation system based on a hybrid approach. *Proceedings of the 20th Pan-Hellenic Conference on Informatics* (p. 78). New York: ACM.

Waheed, W., Imran, M., Raza, B., Malik, A. K., & Khattak, H. A. (2019). A hybrid approach toward research paper recommendation using centrality measures and author ranking. *IEEE Access*(7), pp. 33145-33158.

Wang, G., He, X., & Ishuga, C. I. (2018). HAR-SI: A novel hybrid article recommendation approach integrating with social information in scientific social network. *Knowledge-Based Systems*(148), pp. 85-99.

Wang, J., Liu, Z., & Zhao, H. (2012). Group recommendation based on the PageRank. *JNW, 7*(12), 2019-2024.

West, J. D., Wesley-Smith, I., & Bergstrom, C. T. (2016). A recommendation system based on hierarchical clustering of an article-level citation network. *IEEE Transactions on Big Data, 2*(2), 113-123.

West, J. D., Wesley-Smith, I., & Bergstrom, C. T. (2016). A recommendation system based on hierarchical clustering of an article-level citation network. *IEEE Transactions on Big Data, 2*(2), 113-123.

Wu, L., Sun, P., Hong, R., Ge, Y., & Wang, M. (2018). Collaborative neural social recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Xie, J., Zhu, F., Huang, M., Xiong, N., Huang, S., & Xiong, W. (2019). Unsupervised learning of paragraph embeddings for context-aware recommendation. *IEEE Access*(7), pp. 43100-43109.

Zhang, D., Yin, J., Zhu, X., & Zhang, C. (2018). Network representation learning: A survey. *IEEE transactions on Big Data*.

Zhang, L., & Chang, W. K. (2016). A study on scientific article recommendation system with user profile applying TPIPF. *Journal of the Korean Society for information Management, 33*(1), pp. 317-336.