Social Paper Recommender System Based on Distributed Representation

Songkai Tang
SEIEE
Shanghai Jiao Tong University

Pengfei Chen
SEIEE
Shanghai Jiao Tong University

To help generate relevant suggestions for researchers, recommendation systems have started to leverage the latent interests in the publication profiles of the researchers themselves. While using such a publication citation network has been shown to enhance performance, the network is often sparse, making recommendation difficult.

To alleviate this sparsity, we examine the effect of modeling a researcher's past works in recommending scholarly papers to the researcher. Our hypothesis is that an author's published works constitute a clean signal of the latent interests of a researcher. A key part of our model is to enhance the profile derived directly from past works with information coming from the past works' referenced papers as well as papers that cite the work.

In our experiments, we differentiate between junior researchers who have only published one paper and senior researchers that have multiple publications.

We also design a new papre2vec model, in which we apply Doc2Vector method, use pre-trained the distributed word representation to build a improved feature vector, and use DeepWalk to find the associating paper with a certain recommended paper. Such a new model greatly improves the performance.

I. Introduction

Digital libraries (DLs) are raising at a high speed: today, much of the world's new knowledge is now largely captured in digital form and archived within a digital library system. However, these trends also lead to information overload, where users find an overwhelming number of publications that match their search queries but are largely irrelevant to their latent information needs.

In order to solve such problem, past researchers have focused their attention on finding better ranking algorithms for paper search. In such area, the PageRank algorithm [1] has been employed to induce a better global ranking of search results. A drawback of this approach is that it is a global search and does not induce better rankings that are personalized for the specific interests of the user.

To handle with this issue, digital libraries such as Elsevier, PubMed, SpringerLink all have systems that can send out email alerts or provide RSS feeds on paper recommendations that match user interests. These systems make the DL more proactive, sending out matched articles in a timely fashion. Unfortunately, these require the user to state their interests explicitly, either in terms of categories or as saved searches, and take up valuable time on the part of the user to set up.

We aim to address this problem by providing recommendation results by using information about the user's research interests that exists in their publication list. A researcher's publication list both a historical and current list of research interests and requires little to no effort on the part of the user to provide. The aim of our work in this paper is to study and assess the effectiveness of different models in representing this information in their user profile. Our main contribution in this work is in developing the whole model which accounts for information contained not only in the papers that are published by an author, but also in papers that are referenced by or that cite the author's work. We extend this paradigm in modeling candidate papers to recommend, enriching their representation to also include their referenced work and works that cite them. We show that modeling these contexts is crucial for obtaining higher recommendation accuracy.

This kind of system was first introduced by Kazunari Sugiyama [2], and has been improved in our project by introducing our papre2vec model, which applies Doc2Vector model using pre-trained the distributed word representation to build an improved feature vector, and uses DeepWalk method. With such model, we contribute to higher accuracy, and are able to find more recommend paper.

Word2vec can be view as word-level distributed represent, Doc2Vector can be view as sentence-level distributed represent, and DeepWalk can be view as paragraph-level distributed representation.

Other works have differentiated their analysis of scholarly paper recommendation by user experience. For example, Torres et al [3] investigated recommendation satisfaction according to two types of researcher's level, students (masters and PhD students) and professionals (researchers and professors). Kazunari Sugiyama [2] further divide researcher's level by their published paper and their citation count.

Similarly, Kazunari Sugiyama [2] also hypothesize that the set of topics represented by all of a senior researcher papers is not representative of a researcher's current interests. To further improve recommendation accuracy, we use a distributed representation from text instead of traditional TD-IDF topic model.

II. Related Work

Recommendation can be viewed as periodic searching of a digital library. PageRank algorithm is a widely adopted search ranking algorithm used by Google. We then review work on recommendation systems in the environment of scholarly DLs, As finding potential papers also based on Doc2Vector method, we also briefly review these parts of work. We conclude our review with a discussion on the representations that have been used to construct a robust user profile for use in recommendation.

A. Improving Ranking in Digital Libraries

The PageRank algorithm [1] simulates a user navigating the Web at random, by choosing between jumping to a random page with a certain probability (referred to as the damping factor d), and following a random hyperlink. While this algorithm has been most famously applied to improve ranking of Web search results, it has also been applied to the digital library field in two ways: (1) in improving the ranking of search results, and (2) in measuring the importance of scholarly papers.

(a) Ranking Search Results

Since the beginning of bibliometric analysis, scholars have been measuring the count of other publications that refer to a particular author or work. This notion of citation count is widely used in evaluating the importance of a paper because it has been shown to strongly correlate with academic document impact [4]. The Thomson Scientific ISI impact factor (ISI IF) is the representative approach using citation count [5], which factors citation counts with a moving window to calculate the impact of certain journals. The advantages of citation count are (1) its simplicity of computation; and (2) that it is a proven method which has been used for many years in scientometrics. However, citation counting has well-known limitations: Citing papers with high impact and ones with low impact are treated equally in standard citation counting.

In order to overcome the above shortcomings of impact factor, Sun and Giles [6] noted that conference venues in computer science are a prime vehicle for impact calculation that are neglected by the ISI impact factor. They proposed to remedy this problem by incorporating the popularity factor to consider venue as an information cue and to reflect the influence of a publication venue. This popularity factor is defined based on citation analysis of publication venues and the PageRank algorithm.

When PageRank is introduced to scholarly papers, the rank of a paper can decrease if the paper contains a large quantity of outgoing links. In some ways, this is counter-intuitive, as a well-referenced paper may better contextualize its contributions with respect to existing work, and would thus be a mark of higher quality work. Krapivin and Marchese [7] proposed "Focused PageRank" algorithm to alleviate this problem. In their approach, a reader of an article (referred to as a "focused surfer") may follow the references with different probabilities, so their random surfer model becomes focused on some of the references in the article.

While PageRank can estimate authority of the article, one of its problems is that it ranks articles based on the prior popularity (number of citations) or prior prestige (PageRank score). Therefore, recent articles always obtain lower scores. However, it is important for researchers to be able to find such recent articles because they can discover new research directions, solutions and approaches, and digest new work that is relevant to their current interests. As such, Sayyadi and Getoor [8] proposed "FutureRank" which computes the expected future PageRank, focusing on citation network of scholarly papers.

(b) Measuring the Importance of Scholarly Papers

PageRank has also been applied to measure the importance of scholarly papers. Unlike the studies in Section (a), these works define the importance of a paper among a certain given set.

The ISI Impact Factor is also flawed in that its rankings are biased towards popularity. In order to overcome this problem, Bollen et al. [9] compared the rankings of journals obtained by the following approaches: (1) ISI Impact Factor, (2) weighted PageRank, and (3) their contribution called Y -factor, that is a product of (1) and (2). Journal ranking obtained by their Y -factor showed that the top ranked journal closely matched personal perception of importance.

Chen et al. [10] applied the PageRank algorithm to the scientific citation networks. They found that some classical articles in physics domain have a small quantity of citations but also a very high PageRank. They called these papers scientific gems and concluded that existence of such gems is caused by the PageRank model, which captures not only the total citation count but also the rank of each of the citing papers.

B. Recommendation in Digital Libraries

Recommendation systems provide a promising approach to ranking scholarly papers according to a user's interests. Such systems are classified by their underlying method of recommendation.

Collaborative filtering [11, 12, 13] is one of the most successful recommendation approaches that works by recommending items to target users based on what other similar users have previously preferred. This method has been used in e-commerce site such as Amazon.com, Ebay and so on. However, it suffers from "cold-start problem," in which it cannot generate accurate recommendations without enough initial ratings from users. Recent works alleviate this problem by introducing pseudo users that rate items [14] and imputing estimated rating data using some imputation technique [15].

Content-based filtering [16, 17, 18] is also widely used in recommender systems. This approach provides recommendations by comparing candidate item's content representation with the target user's interest representation. This method has been applied mostly in textual domains such as news recommendation [19] and hybrid approaches with collaborative filtering [20, 21, 22].

We now examine recommendation systems in the field of scholarly digital libraries. McNee et al. [23] proposed an approach to recommending citations using collaborative filtering. Their approach extended Referral Web [24] by exploring ways to directly apply collaborative filtering to social networks that they term as the "Citation Web," a graph formed by the citations between research papers. This data can be mapped into a framework of collaborative filtering and used to overcome the cold-start problem. Expanding this approach, Torres et al. [25] proposed a method for recommending research papers by combining collaborative filtering and content-based filtering. However, the final ranking scheme obtained by merging the output from collaborative filtering and content-based filtering is not performed as the authors claim that pure recommendation algorithms are not designed to receive input from another recommender algorithm. Gori and Pucci [26] devised a PageRank-based method for recommending research papers. But in their approach, a user have to prepare initial set of relevant articles to get better recommendation, and the damping factor d that affects the score of PageRank is not optimized. Yang et al. [27] presented a recommendation system for scholarly papers that used a ranking-oriented collaborative filtering approach. Although their system overcomes the cold-start problem by utilizing implicit behaviors extracted from a user's access logs, Web usage data are noisy and not reliable generally as pointed out in [28]. In addition, their predefined criteria and parameters to select effective data are not investigated in detail.

C. Doc2Vector Model

When we apply machine learning in the problem of natural language understanding, the first step is to find a way to translate these symbols into vector. In NLP, the most intuitive, and so far, the most commonly used word representation is one-hot Representation, which represents each word as a very long vector. The dimension of this vector is the size of the word list. Most of the elements are 0, and only one dimension has a value of 1. This dimension represents the current word. Of course, there is an important problem in this way of expression, that is, the word gap: any two words are isolated. There is no relation between the two words. Even the synonyms cannot survive.

Since this easy representation has such a serious defect, there is a need for a word-to-vector representation that can both represent the word itself and consider the semantic distance. Thus, Hinton proposed distributed representation in 1986 [29]. It is a low dimensional real number vector, and the distance between related or similar words is closer.

Word2Vector is an algorithm that has just started in recent years to train the N-gram language model by neural network machine learning algorithm and to find the method of word corresponding to vector during the training process. Word2vec is an efficient algorithm model that characterizing a word as a real value vector. By using the thought of depth learning, it can simplify the processing of text content into vector operations in K dimensional vector space by training, and the similarity in vector space can be used to represent the semantic similarity of the text.

Word2Vec is actually composed of two different ways: Continuous Bag of Words (CBOW) and Skip-gram. The goal of CBOW is to predict the probability of current words based on the above and below words. Skip-gram is just the opposite: predicting the probability of context based on the current words.

However, even if the above models deal with word vectors on average, we still ignore the influence of alignment between words on sentiment analysis. That is, the above word2vec is based on the semantic dimension of word, but does not have the ability of "semantic analysis" in context.

As a summative method for dealing with variable length text, Quoc Le and Tomas Mikolov propose Doc2Vec method

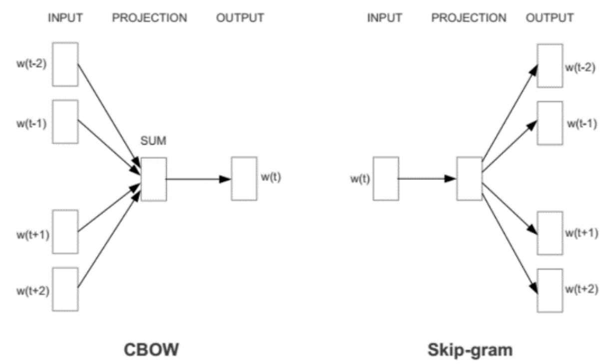

Figure 1: CBOW and Skip-gram.

[30]. In addition to adding a paragraph vector, this method is almost equivalent to Word2Vec. Like Word2Vec, there are two ways in this model: Distributed Memory (DM) and Distributed Bag of Words (DBOW). DM attempts to predict the probability of a word in a given context and paragraph vector. In the training process of a sentence or document, paragraph ID remains the same, sharing the same paragraph vector. DBOW predicts the probability of a group of random words in paragraphs only when a paragraph vector is given.

III. PROPOSED METHOD

To sum up, systems employing PageRank have demonstrated improved ranking of search results. However, since PageRank is a global ranking scheme, the user's research interests are not considered in the ranking and the generated ranking is thus not customized to the user. When we examined the commonalities of the recommendation systems, we note that they consider a user's interests in only a limited sense, by virtue of using metadata or collaborative filtering. Building a user profile derived directly from user content is thus most relevant to our scenario. However, the existing methods for constructing a robust user profile consider only click-through data for Web page recommendation. We believe this method is possibly too ephemeral for research interests, which are more long term in general. For scholarly paper recommendation, we should utilize the textual nature of the papers themselves.

To address these shortcomings, we propose recommending papers based on an individual's recent research interests as modeled by a profile derived from their publication list. We hypothesize that this will result in high recommendation accuracy as we believe that a user's research interests are reflected in their prior publications.

We first construct each researcher's profile using their list of previous publications, and then recommend papers by comparing the profiles with the contents of candidate papers. Unlike research studies described in the previous section, our approach is novel because it directly addresses each user's research interest using their publication history. A key aspect of our approach is that we include Doc2Vector and DeepWalk method. While simple TF-IDF model fail to use the literature feature of words in the paper, we introduce Doc2Vector method to make the distance between similar paper closer, and use DeepWalk to find more candidate papers.

A. User Profile Construction

We first divide researchers into (i) junior researchers, and (ii) senior researchers. This is because the two types of researchers' publication lists exhibit different properties. We define junior researchers as having only one recently published paper, which has yet to attract any citations (i.e., no citation papers). Senior researchers differ in having multiple past publications, where their past publications may have attracted citations. This is shown graphically in Figures 2 and 3.

Our representations of the user profile are based on foundation of a paper represented as a feature vector. For each paper p on the publication list of a researcher, we transform p into a feature vector as follows:

$$\boldsymbol{f}^p = (w_{t_1}^p, w_{t_2}^p, \cdots, w_{t_m}^p), \qquad (1)$$

Where m is the number of distinct terms in the paper, and t_k denotes each term. Using term frequency (TF).

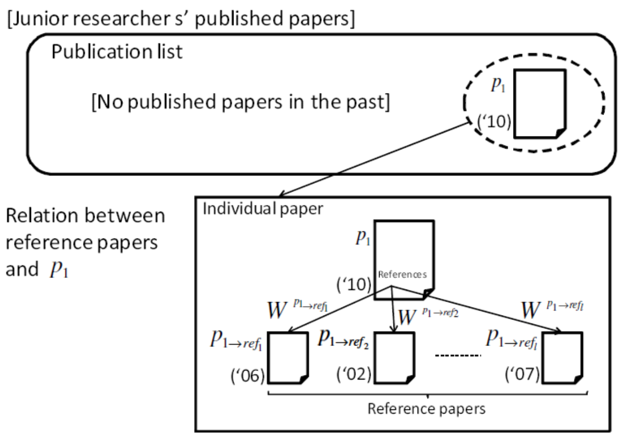

Figure 2: Publication list by junior researchers.

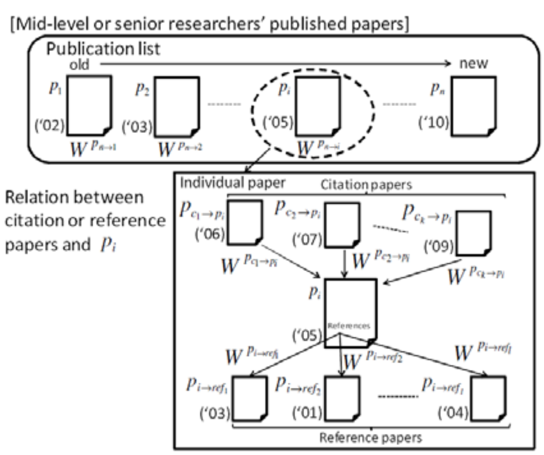

Figure 3: Publication lists by senior researchers.

TF-IDF Model: For simple TF-IDF model, we define each element $w_{t_k}^p$ of $\boldsymbol{f}^p$ in Equation (1) as follows:

$$w_{t_k}^p = \frac{tf(t_k, p)}{\sum_{s=1}^{m} tf(t_s, p)},$$

where $tf(t_k, p)$ is the frequency of term t_k in a paper p. We prefer TF rather than adopting the standard TF-IDF scheme, as the small number of papers in a researcher's publication list may adversely affect the IDF score calculation.

Paper2vector Model: In our new model, given a sequence of training words $w_1, w_2, w_3, \ldots, w_T$, the objective of the word vector model is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=k}^{T-k} \log p(w_t | w_{t-k}, \ldots, w_{t+k})$$

The prediction task is typically done via a multiclass classifier, such as softmax. There, we have:

$$p(w_t | w_{t-k}, \ldots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

Each of y_i is un-normalized log-probability for each output word i, computed as

$$y = b + Uh(w_{t-k}, \ldots, w_{t+k}; W) \qquad (2)$$

where U, b are the softmax parameters. h is constructed by a concatenation or average of word vectors extracted from *W*.

Here we use Doc2Vector method (see Figure 4), every paragraph is mapped to a unique vector, represented by a column in matrix D and every word is also mapped to a unique vector, represented by a column in matrix W. The paragraph vector and word vectors are averaged or concatenated to predict the next word in a context. In the experiments, we use concatenation as the method to combine the vectors.

More formally, the only change in this model compared to the word vector framework is h is constructed from W and D.

The paragraph token can be thought of as another word. It acts as a memory that remembers what is missing from the current context – or the topic of the paragraph.

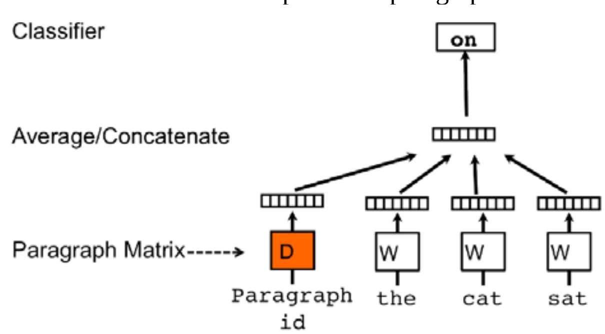

Figure 4: Doc2Vector model

From this, we are able to construct $\boldsymbol{f}^p$ in our model.

Based on the set of feature vectors $\boldsymbol{f}^p$, we can then construct the user profiles for junior researchers, and senior researchers by using the relations between each researcher's published paper and its citation and reference papers.

In our framework, we assign weights to modify the influence of citation and reference papers. Let $W^{p_{u \to v}}$ be the multiplicative coefficient used to integrate the target paper with a source paper . We explore the following two different weighting schemes for this framework.

(W1) Linear Combination (LC)

This baseline weighting scheme simply combines papers u and v. In other words, we define $W^{p_{u\to v}}$ as follows:

$$W^{p_{u\to v}} = 1. \tag{3}$$

This method treats each neighboring paper v on a parity with the researcher's own paper u.

(W2) Cosine Similarity (COS)

Here we employ the cosine similarity between papers u and v as the weighting scheme. Applying Equation (1), let f^u and f^v be feature vector of papers u and v, respectively. Then the similarity $\text{sim}(f^u, f^v)$ between these two feature vectors is computed by Equation (4), and we use $\text{sim}(f^u, f^v)$ as $W^{p_{u\to v}}$.

$$W^{p_{u\to v}} = sim(\boldsymbol{f}^u, \boldsymbol{f}^v) = \frac{\boldsymbol{f}^u \cdot \boldsymbol{f}^v}{|\boldsymbol{f}^u| \cdot |\boldsymbol{f}^v|}. \tag{4}$$

This approach strengthens the signal from a researcher's paper u by emphasizing papers that are more similar among its citation and reference papers.

With weighting schemes now defined, we can construct user profiles for the two classes of researchers.

User Profile for Junior Researcher

As shown in Figure 2, junior researchers have only one paper p_1 that is most recently published (e.g., '10). The paper has reference papers $p_1 \to ref_y$ $(y = 1, \dots, l)$ (published older than '10). However, there are no papers that cite the paper p_1 because p_1 is just published recently. Therefore, the weights are only applicable to reference papers. In terms of that, user profile $\mathbf{P}_{user}$ is defined as follows:

$$\boldsymbol{P}_{user} = \boldsymbol{f}^{p1} + \sum_{y=1}^{l} W^{p1\to ref_y} \boldsymbol{f}^{p1\to ref_y}, \tag{5}$$

where $W^{p_1 \to ref_y} (y = 1, \dots, l)$ denotes each weight assigned to paper $p_1 \to ref_y$ $(y = 1, \dots, l)$ computed on the basis of paper p_1, defined by a choice among (W1) and (W2).

User Profile for Junior Researcher

As shown in Figure 3, senior researchers have several published papers $p_i (i = 1, \dots, n-1)$ in the past (e.g., '02, '03, …) as well as the most recently published paper p_n (e.g., '10). With the exception of the most recent paper p_n, each past published paper p_i may be cited by other papers $p_{c_x \to p_i} (x = 1, \dots, k)$. In addition, each paper p_i has reference papers. Therefore, we first construct the feature vector $\mathbf{F}^{p_i}$ for each paper, using its feature vectors for citation papers $f^{c_x \to p_i}$ and their weights $W^{p_{c_x \to p_i}}$, and its feature vectors for reference papers $f^{p_i \to ref_y}$ and their weights $W^{f^{p_i \to ref_y}}$, as follows:

$$\begin{aligned}\boldsymbol{F}^{p_i} = \boldsymbol{f}^{p_i} &+ \sum_{x=1} W^{p_{c_x \to p_i}} \boldsymbol{f}^{p_{c_x \to p_i}} \\ &+ \sum_{y=1}^{l} W^{p_i \to ref_y} \boldsymbol{f}^{p_i \to ref_y}.\end{aligned} \tag{6}$$

Then, using Equation (6), the user profile $\mathbf{P}_{user}$ for a senior researcher is defined as follows:

$$\begin{aligned}\boldsymbol{P}_{user} &= W^{p_{n\to 1}} \boldsymbol{F}^{p1} + W^{p_{n\to 2}} \boldsymbol{F}^{p2} \\ &\qquad + \cdots + W^{p_{n\to n-1}} \boldsymbol{F}^{n-1} + \boldsymbol{F}^{p_n} \\ &= \sum_{z=1}^{n-1} W^{p_{n\to z}} \boldsymbol{F}^{p_z} + \boldsymbol{F}^{p_n},\end{aligned} \tag{7}$$

where $W^{p_n \to z} (z = 1, \dots, n-1)$ denotes each weight assigned to paper $p_n \to z$ computed on the basis of the most recent paper p_n, defined by a choice among (W1) and (W2).

As these researchers have multiple prior publications, we also employ an additional forgetting factor (W4) that gives a larger weight (close to 1) to more recent papers and smaller weight (close to 0) to older papers, under the assumption that a user's research interest gradually decays as years pass.

B. Feature Vector Construction for Candidate Papers

Unlike the TF representation of papers used in the user profile, we employ TF-IDF for the calculation of the feature vector $f^{P_{rec}}$ of a candidate paper P_{rec} to be considered for recommendation. Identical to Equation (1), we first define the feature vector $f^{P_{rec}}$ of P_{rec} as follows:

$$\boldsymbol{f}^{p_{rec}} = (w_{t_1}^{p_{rec}}, w_{t_2}^{p_{rec}}, \cdots, w_{t_m}^{p_{rec}}), \tag{8}$$

where m is the number of distinct terms in the paper, and $t_k (k = 1, 2, \dots, m)$ denotes each term. Using TF-IDF, each element $w_{t_k}^{P_{rec}}$ of $f^{P_{rec}}$ in Equation (8) is defined as follows:

$$w_{t_k}^{p_{rec}} = \frac{tf(t_k, p_{rec})}{\sum_{s=1}^{m} tf(t_s, p_{rec})} \cdot \log \frac{N}{df(t_k)}, \tag{9}$$

where $tf(t_k, P_{rec})$ is the frequency of term t_k in the target paper p, N is the total number of papers to recommend in the collection, and $df(t_k)$ is the number of papers in which term t_k appears. We favour TF-IDF here rather than pure TF for candidate papers, as the pool for candidate papers is usually much larger. In our experiments as we describe later, our candidate paper base consists of several hundreds of papers, making IDF more reliable and consistent. Critically, our dataset also contains clean citation information that allows us to construct correct citation and reference papers. Therefore, we also use this information to characterize a candidate paper better and obtain high recommendation accuracy: Let $\mathrm{F}^{P_{rec}}$ be the feature vector for paper to recommend, as well as Equation (6), this is denoted as follows:

$$\begin{aligned}\boldsymbol{F}^{p_{rec}} = \boldsymbol{f}^{p_{rec}} &+ \sum_{x=1}^{k} W^{p_{c_x \to p_{rec}}} \boldsymbol{f}^{p_{c_x \to p_{rec}}} \\ &+ \sum_{y=1}^{l} W^{p_{rec \to ref_y}} \boldsymbol{f}^{p_{rec \to ref_y}}.\end{aligned} \tag{10}$$

where $P_{c_x \to P_{rec}} (x = 1, \dots, k)$ and $P_{rec \to ref_y} (y = 1, \dots, l)$ denote papers that cite and papers that refers, respectively.

C. Recommendation of Papers

Using the user profile defined by Equation (5) or (7), and feature vector for the candidate paper to recommend defined by Equation (10), our system computes $sim(P_{user}, \mathrm{F}^{P_{rec}})$ between P_{user} and $\mathrm{F}^{P_{rec}}$ by Equation (11):

$$sim(\boldsymbol{P}_{user}, \boldsymbol{F}^{p_{rec}}) = \frac{\boldsymbol{P}_{user} \cdot \boldsymbol{F}^{p_{rec}}}{|\boldsymbol{P}_{user}| \cdot |\boldsymbol{F}^{p_{rec}}|}, \tag{11}$$

and ranks the set of candidate papers in order of decreasing similarity.

In our new paper2vec model, we use Doc2Vec to form semantic feature vector $P^{Semantic}$, and DeepWalk to form structure feature vector $P^{Structure}$. We combine these two as:

$$t - Sim(p1, p2) = cos < P_1^t, P_2^t >. t$$
$$= \{Semantic, Structure\}$$

For candidate paper, we get id from:

Candidate paper id = $argmax(SemanticSim(F^{User}, P^{Paper}))$

The similarity between papers is defined as:

$$AdvancedSim(p1,\ p2) = SemanticSim(p1, p2) + StructureSim(p1, p2)$$

And our final paper list is consisting of candidate paper and top k of similar paper in ACL list. The usage of DeepWalk is possible here because of the high accuracy of our model, and the recommend paper is high likely to meet the need of researchers. When the user finds the recommend paper fit its interest, it may want to read more paper like this, and our advanced similarity can successfully handle this.

IV. EXPERIMENTS

A. *Experimental Data*

We use publication lists of 27 researchers who have been engaged in natural language processing and information retrieval, and have publication lists in DBLP6. As DBLP lists many important venues in computer science, we believe that a researcher's DBLP list is representative of their main interests. In the experiment, we use papers and references (P+R) to represent the situation of junior researchers, and after adding citations (P+R+C) to represent the situation of senior researchers.

We construct the user profile for each researcher using their respective publication list in DBLP. Table 1 shows the statistics about these researchers. Since we focus on recommending scientific paper only, we removed references to Web sites, books, and other URLs for our experiments.

The candidate papers to recommend is the ACL Anthology Reference Corpus (ACL ARC). The ACL ARC is constructed from a significant subset of the ACL Anthology, a digital archive of conference and journal papers in natural language processing and computational linguistics. The ACL ARC consists of 10,921 articles from the February 2007 snapshot of the ACL Anthology.

Among them, 597 full papers published in ACL 2000-2006 were selected. Each of junior and senior researchers were asked to mark papers relevant to their recent research interest. This corpus features information about citation and reference papers for each paper. We use this information to construct feature vectors for these papers as described in Section III.

Average number of DBLP papers	10.74
Average number of relevant papers	14.06
Average number of citation papers	15.81
Average number of reference papers	24.14

Table 1: Some statistics about researchers.

B. *Evaluation Measure*

As in standard information retrieval (IR), the top ranked documents are the most important to get correct, since users check these ranks more often. To properly account for this effect, we employ IR evaluation measures: (1) normalized discounted cumulative gain (NDCG) [31], and (2) mean reciprocal rank (MRR) [32].

Normalized Discounted Cumulative Gain (NDCG)

Discounted cumulative gain (DCG) is a measure that gives more weight to highly ranked documents and incorporates different relevance levels (relevant, and irrelevant) through different gain values.

$$DCG(i) = \begin{cases} G(1) & \text{if } i = 1 \\ DCG(i-1) + \frac{G(i)}{log(i)} & \text{otherwise,} \end{cases}$$

where *i* denotes the i^{th} ranked position. In our work, the relevance level depends on just a binary notion of relevance: whether recommended papers are relevant or not to the user. We use $G(i) = 1$ for relevant search results and $G(i) = 0$ for irrelevant search results. The average normalized DCG over all users is selected to show the accuracy of recommendation. As a typical recommendation system will just recommend a few items, we are only concerned about whether the top ranked results are relevant or not. Therefore, in this work, we use NDCG@N (N=5,10) for evaluation where N is the number of top-N papers recommended by our proposed approaches.

Mean Reciprocal Rank (MRR)

Mean reciprocal rank (MRR) indicates where in the ranking the first relevant item is returned by the system, averaged over all users. This measure provides insight in the ability of the system to return a relevant item at the top of the ranking.

C. *Experimental Results*

In this section, we show our experimental results. In our experiments, we construct feature vectors for the candidate papers to recommend using the target paper only (ACL ARC paper, denoted as P), the target paper and its citation papers (denoted as P+C), the target paper and its reference papers (P+R), and the target paper and both citation and reference papers (P+C+R).

We evaluate the recommendation accuracy of our approach using these feature vectors for candidate papers and user profile described in the following.

we compare recommendation accuracy obtained by user profile constructed by the past paper only (P), the past paper and its citation papers (P+C), the past and its reference papers (P+R), and the past paper and both citation and reference papers (P+C+R). Table 2 shows the recommendation accuracy evaluated with NDCG@5, 10, and MRR, as shown in Table 2.

According to these results, regarding user profile, the recommendation accuracy obtained by user profile (P+C+R) outperforms that obtained by user profile (P), in most cases.

Comparison of TF-IDF and Our Approach

In the Table 2, we also show the results of using simple TF-IDF model (without Doc2Vector and DeepWalk) for recommendation. In all cases, the results obtained by TF-IDF score give much lower accuracy than that of our proposed approaches. As simple TF-IDF cannot use the literature similarity message of paper, TF-IDF loses many useful features. And in our paper2vec model, as it has a high accuracy, DeepWalk can make sense.

NDCG@5	Weight	P	P+C	P+R	P+R+C
TF-IDF	LC	0.325	0.334	0.390	0.401
	COS	0.325	0.351	0.399	0.406
DOC2VEC	LC	0.832	0.822	0.833	0.840
	COS	0.832	0.822	0.831	**0.843**

NDCG@5

NDCG@10	Weight	P	P+C	P+R	P+R+C
TF-IDF	LC	0.305	0.323	0.451	0.362
	COS	0.305	0.346	0.365	0.362
DOC2VEC	LC	0.853	0.849	0.850	0.857
	COS	0.853	0.850	0.848	**0.858**

NDCG@10

MRR	Weight	P	P+C	P+R	P+R+C
TF-IDF	LC	0.621	0.657	0.670	0.709
	COS	0.621	0.696	0.688	0.709
DOC2VEC	LC	0.806	0.802	0.804	0.813
	COS	0.806	0.802	0.801	**0.814**

MRR

Table 2: Recommendation accuracy for researchers evaluated with NDCG@5, NDCG@10 and MRR

Summary of Obtained Results

Our key result is that incorporating the context of a paper, in the form of references and, when available, citations, results in improved recommendation accuracy. Our new model, based on Doc2Vector and DeepWalk method, give a great accuracy and efficiency improvement.

V. CONCLUSION

We have proposed a generic model towards recommending scholarly papers relevant to a researcher's interests by capturing their research interests through their past publications.

Compared to the old model, we introduce Doc2Vector method while getting the representation vectors for papers, and thanks to the high accuracy, the usage of DeepWalk to find more recommend paper becomes possible.

In our analysis, we have verified the effectiveness of our approach for two classes of researchers: junior researchers, and senior researchers. We evaluated recommendation accuracy using NDCG and MRR, and achieve consistent results using both metrics.

VI. ACKNOWLEDGMENT

We thank tutors and classmates who volunteered their time to offer suggestions for this study. Without their help, this work would not have been possible.

REFERENCES

[1] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report SIDL-WP-1999-0120, Stanford Digital Library Technologies Project, 1998.

[2] Kazunari Sugiyama, and Min-Yen Kan. Scholarly Paper Recommendation via User's Recent Research Interests. ACM 978-1-4503-0085-8/10/06, 2010

[3] R. Torres, S. M. McNee, M. Abel, J. A. Konstan, and J. Riedl. Enhancing Digital Libraries with TechLens. In Proc. of the 4th ACM/IEEE Joint Conference on Digital Libraries (JCDL 2004), pages 228–236, 2004.

[4] F. Narin. Evaluative Bibliometrics: The Use of Publication and Citation Analysis in the Evaluation of Scientific Activity. Cherry Hill, N.J.: Computer Horizons, 1976.

[5] E. Garfield. Citation Indexing: Its Theory and Application in Science, Technology, and Humanities. New York: John Wiley and Sons, 1979.

[6] Y. Sun and C. Giles. Popularity Weighted Ranking for Academic Digital Libraries. In Proc. of the 29th European Conference on Information Retrieval (ECIR 2007), pages 605–612, 2007.

[7] M. Krapivin and M. Marchese. Focused PageRank in Scientific Papers Ranking. In Proc. of the 11th International Conference on Asian Digital Libraries (ICADL 2008), Lecture Notes in Computer Science (LNCS), Vol. 5362, pages 144–153, 2008.

[8] H. Sayyadi and L. Getoor. FutureRank: Ranking Scientific Articles by Predicting their Future PageRank. In Proc. of the 9th SIAM International Conference on Data Mining, pages 533–544, 2009.

[9] J. Bollen, M. A. Rodriguez, and H. V. D. Sompel. Journal Status. Scientometrics, 69(3):669–687, 2006.

[10] P. Chen, H. Xie, S. Maslov, and S. Redner. Finding Scientific Gems with Google's PageRank Algorithm. Journal of Informetrics, 1(1):8–15, 2007.

[11] D. Goldberg, D. Nichols, B. M. Oki, and D. B. Terry. Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM, 35(12):61–70, 1992.

[12] P. Resnick, N. Iacovou, M. Suchak, and J. R. P. Bergstorm. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proc. of the ACM 1994 Conference on Computer Supported Cooperative Work (CSCW '94), pages 175–186, 1994.

[13] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, 40(3):77–87, 1997.

[14] S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste. Naïve Filterbots for Robust Cold-Start Recommendations. In Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD'06), pages 699–705, 2006.

[15] X. Su and T. M. Khoshgoftaar and R. Greiner. Imputed Neighborhood Based Collaborative Filtering. In Proc. of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2008), pages 633–639, 2008.

[16] J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proc. of the 14th Conference on Uncertanity in Artificial Intelligence (UAI '98), pages 43–52, 1998.

[17] C. Basu, H. Hirsh, and W. Cohen. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In Proc. of the 15th National Conference on Artificial Intelligence (AAAI '98), pages 714–720, 1998.

[18] B. M. Sarwar, G. Karypis, and J. A. Konstan. Analysis of Recommendation Algorithms for E-commerce. In Proc. of the 2nd ACM Conference on Electronic Commerce (EC '00), pages 158–167, 2000.

[19] W. Chu and S.-T. Park. Personalized Recommendation on Dynamic Content Using Predictive Bilinear Models. In Proc. of the 18th International World Wide Web Conference (WWW2009), 2009. 691-700.

[20] M. Balabanovic and Y. Shoham. Fab: Content-Based, Collaborative Recommendation. Communications of the ACM, 40(3):66–72, 1997.

[21] C. Basu, H. Hirsh, and W. Cohen. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In Proc. of the 15th National Conference on Artificial Intelligence (AAAI '98), pages 714–720, 1998.

[22] P. Melville, R. J. Mooney, and R. Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations. In Proc. of the 18th National Conference on Artificial Intelligence (AAAI2002), pages 187–192, 2002.

[23] S. M. McNee, I. Albert, D. Cosley, S. L. P. Gopalkrishnan, A. M. Rashid, J. S. Konstan, and J. Riedl. Predicting User Interests from Contextual Information. In Proc. of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02), pages 116–125, 2002.

[24] H. Kautz, B. Selman, and M. Shah. Referral Web: Combining Social Networks and Collaborative Filtering. Communications of the ACM, 40(3):63–65, 1997.

[25] R. Torres, S. M. McNee, M. Abel, J. A. Konstan, and J. Riedl. Enhancing Digital Libraries with TechLens. In Proc. of the 4th ACM/IEEE Joint Conference on Digital Libraries (JCDL 2004), pages 228–236, 2004.

[26] M. Gori and A. Pucci. Research Paper Recommender Systems: A Random-Walk Based Approach. In Proc. of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006), pages 778–781, 2006.

[27] D. Yang, B. Wei, J. Wu, Y. Zhang, and L. Zhang. CARES: A Ranking-Oriented CADAL Recommender System. In Proc. of the 9th ACM/IEEE Joint Conference on Digital Libraries (JCDL 2009), pages 203–211, 2009.

[28] C. Shahabi and Y.-S. Chen. An Adaptive Recommendation System without Explicit Acquisition of User Relevance Feedback. Distributed and Parallel Databases, 14(3):173–192, 2003.

[29] G.E. Hinton. 1986. Learning distributed representations of concepts. In Proceedings of the eighth annual conference of the cognitive science society, pages 1–12. Amherst, MA.

[30] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference onMachine Learning (ICML 2014), pages 1188–1196,Beijing, China

[31] K. Järvelin and J. Kekäläinen. IR Evaluation Methods for Retrieving Highly Relevant Documents. In Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), pages 41–48, 2000.

[32] E. M. Voorhees. The TREC-8 Question Answering Track Report. In Proc. of the 8th Text REtrieval Conference (TREC-8), pages 77–82, 1999.