Combining Text Embedding and Knowledge Graph Embedding Techniques for Academic Search Engines

Gengchen Mai, Krzysztof Janowicz, and Bo Yan

STKO Lab, University of California, Santa Barbara, CA, USA
http://stko.geog.ucsb.edu/

Abstract. The past decades have witnessed a rapid increase in the global scientific output as measured by published papers. Exploring a scientific field and searching for relevant papers and authors seems like a needle-in-a-haystack problem. Although many academic search engines have been developed to accelerate this retrieval process, most of them rely on content-based methods and feature engineering. In this work, we present an entity retrieval prototype system on top of IOS Press LD Connect which utilizes both textual and structure information. Paragraph vector and knowledge graph embedding are used to embed papers and entities into low dimensional hidden space. Next, the semantic similarity between papers and entities can be measured based on the learned embedding models. Two benchmark datasets have been collected from Semantic Scholar and DBLP to evaluate the performance of our entity retrieval models. Results show that paragraph vectors are effective at capturing the *similarity* and *relatedness* among papers and knowledge graph embedding models can preserve the inherent structure of the original knowledge graph and hence assist in link prediction tasks such as co-author inference.

Keywords: Entity Retrieval · Paper Recommender System · Paragraph Vector · Knowledge Graph Embedding .

1 Introduction

The global scientific output almost doubles every nine years[1]. In the presence of such a tremendous growth of scientific literatures, searching for relevant papers and authors seems unsustainable. Hence, developing methods to accelerate the retrieval process is an active research topic [1]. In fact, several academic search engines have been established to facilitate this process such as Google Scholar, Microsoft Academic Search, Semantic Scholar, DBLP, and so forth. After indexing literature based on their textual content, authors, publication year, and citation information, these academic search engines provide paper-level (and sometimes author-level) recommendations. A core question for such

[1] http://blogs.nature.com/news/2014/05/global-scientific-output-doubles-every-nine-years.html

academic search engines is how to define and measure *similarity* and *relatedness* among research papers, authors, potential funding sources, and so forth. The conventional way is using feature engineering which extracts features from textual content, citation networks, and co-author networks.

Semantic Web technologies play an increasing role in the field of academic publishing, libraries, and bibliographic metadata more broadly where they are used to ease publishing, retrieving, interlinking, and integrating datasets, often across outlets and publishers. Examples for this growing influence are Linked Data portals such as Springer Nature SciGraph[2], the DBLP SPAQRL endpoint[3], IOS Press LD Connect[4], as well as Linked Scientometrics [4]. The availability of these bibliography knowledge graphs makes it possible to bring entity retrieval and content-based paper recommendations together. In fact, the IOS LD Connect portal does not only provide bibliographic data for all authors, papers, journals, and institutions that have published with IOS over the past 30 years as Linked Data, it also provides document embeddings extracted from the *full* text of each paper and knowledge graph embeddings for all entities in the graph.[5]

In this paper, we present an entity retrieval prototype on top of IOS LD Connect which utilizes both textual information and structure information. **The research contributions of our work are as follows:** 1) We developed an entity retrieval system based on paragraph vectors and knowledge graph embeddings. As far as we know, our system is the first entity retrieval system in the bibliography field which uses both techniques. 2) We establish a paper similarity benchmark dataset from Semantic Scholar and empirically evaluate the learned embedding models. 3) Another benchmark dataset from DBLP is constructed and used to evaluate the performance of the learned knowledge graph embedding model.

The rest of this paper will be structured as follows. In Section 2, we first discuss the pros and cons of the existing paper/reviewer recommender systems. Next, in Section 3, the entity retrieval system we developed on top of IOS Press LD Connect is presented and two benchmark datasets are collected from Semantic Scholar and DBLP to evaluate our model. Finally, we conclude our work in Section 4.

2 Related Work

Existing work on paper and author/reviewer recommender systems can be roughly divided into two categories: 1) research focusing on developing new methods and algorithms for enhancing recommendation capabilities, and 2) research focusing on mining and analyzing scholarly data and publishing trends. The first category is largely related to developments in information retrieval,

[2] https://www.springernature.com/gp/researchers/scigraph
[3] http://dblp.rkbexplorer.com/sparql/
[4] http://ld.iospress.nl/
[5] Currently, we serve pre-trained models using the Doc2Vec for the full text and the TransE for the kowledge graphs.

such as semantic similarity measurements, ranking algorithms, and recommendation methods. For example, by combining terms used by citing documents and terms from the document itself, researchers have shown better performance than standard indexing in scientific literature search systems [13]. Mooney et al. [9] devised a content-based book recommending system using information extraction for text categorization. Others have put more weight on providing more capable and intuitive user interface. For instance, Hu et al. developed a Linked-Data-Driven web portal to assist the interactive exploration of scholarly content [5]. The second research direction takes advantage of the enormous amount of scholarly data, such as academic papers, institutions, and researchers, and applies data mining and machine learning approaches to gain insights that could potentially connect the dots and serve the whole research community. One subfield in this direction is scientometrics which deals with analyzing the impact of researchers, research articles and their interplay [3]. In order to analyze the dynamics of diachronic topic-based research communities, a hybrid semantic approach has been developed by Osborne et al [12]. In an attempt to gain insights of future research trends and technologies, Osborne et al. [11] also proposed a technology-topic framework that uses a semantically-enriched topic model to forecast the propagation of technologies to different research areas. Wang et al. [16] present the idea of *linked* document embeddings which jointly learns the textual information as well as the citation network information. The learned document embeddings are further applied to a document classification task to demonstrate the effectiveness of this approach. However, citation networks are only one part of the structured information from scholar data. Other structured information such as the author-to-paper, author-to-organization relationships are also very important for paper and reviewer recommender systems. In this work, we focus on the intersection of both categories outlined above. Our end-user interface and retrieval system correspond to the first direction which emphasizes the information retrieval aspect while our co-author inference component corresponds to the second direction which emphasizes the scholarly data mining aspect.

3 Entity Retrieval System

In this section, we will first describe the used dataset. Next, we will discuss the methods we use to develop an entity retrieval prototype. Finally, we will present two evaluations of our models.

3.1 Dataset

We use the new IOS Press LD Connect platform as main dataset in this work. This knowledge graph encodes the information about all the papers published by IOS Press until now. All metadata about papers are serialized and published as Linked Data by following the bibliographic ontology[6] and a SPARQL endpoint[7]

[6] `http://bibliontology.com/#sec-sioc-rdf`
[7] `http://ld.iospress.nl:3030`

as well as a dereference interface[8] are provided. We also created document and knowledge graph embeddings for use by the broader research community. The document embeddings are learned from the full texts of all PDF papers and will enable researchers to analyze papers and the corpus without having to expose the full text directly (due to copyright limitations).

Table 1 shows the number of entities within LD Connect which includes publishers (`prov:Publisher`), journals (`bibo:Journal`), journal series (`bibo:Series`), periodicals (`bibo:Periodical`), journal issues (`bibo:Issue`), conference papers (`bibo:Chapter`), journal papers (`bibo:AcademicArticle`), authors (`foaf:Person`), organizations (`foaf:Organization`) and their geographic locations, and author lists per paper (`rdf:Seq`). Note that if a person authored multiple papers, (s)he will have a Uniform Resource Identifier (URI) for each paper and `owl:sameAs` is used to connect these URIs to indicate they refer to the same person. Simply put, this reflects the difference between a *creator* role and the person playing this role (while possibly being at different institutions).

Table 1. An overview of LD Connect as of 05/2018

Class Name	# of Instances
prov:Publisher	1
bibo:Journal	125
bibo:Series	41
bibo:Periodical	2255
bibo:Issue	8891
bibo:Chapter	46915
bibo:AcademicArticle	80891
foaf:Person	385272
foaf:Organization	168360
rdf:Seq	109309

3.2 Textual Embedding

Paragraph vectors [6], specifically the Distributed Bag of Words version of Paragraph Vector (PV-DBOW), are used to encode textual information of each paper into low dimensional vectors. Word embedding [8] was first proposed as a two layer neural network architecture to encode each word into a dense continuous vector. The learned word vectors have been shown to preserve syntactic and semantic word relationships. As a successor of word embedding, paragraph vector embeds each piece of text with arbitrary length into a continuous vector space such that the learned vectors preserve the semantics of the text.

The PV-DBOW model is similar to the skip-gram model in Word2Vec in that they both aim to capture semantics as an indirect result of a contextual prediction task [6]. In this prediction task, the model aims to maximize the average log probability of predicting a word given the paragraph. As shown in

[8] `http://ld.iospress.nl/ios/ios-press`

Equation 1, PV-DBOW calculates such average log probability for a sequence of training words $w_1, w_2, ..., w_T$ in paragraph pg_i.

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|pg_i) \tag{1}$$

The prediction is done by means of a softmax classifier shown in Equation 2.

$$p(w_t|pg_i) = \frac{exp(y_{w_t})}{\sum_j exp(y_j)} \tag{2}$$

Each y_j defined by:

$$y = Uh(pg_i; D) + b \tag{3}$$

where U and b are the weights and bias in the softmax layer respectively, D is the embedding matrix for paragraphs, and h is a lookup operation to map the one-hot vectors of paragraphs to their respective embeddings from D.

Compared with the vector space model [14], paragraph vector encode each piece of text into a lower dimension vector. It is assumed that cosine similarity between two paragraph vectors represents the semantic similarity between the corresponding texts.

In this work, all 117,835 PDF documents are parsed and mapped to entities in the knowledge graph. After some text preprocessing steps such as tokenization and lemmatization, the preprocessed texts of each paper are fed into PV-DBOW model. The gensim Doc2Vec package is used and the hyperparameters are set as: 200 for vector dimension; 10 for scan window size; 100 for minimum word frequency; and 0.025 for learning rate.

3.3 Structure Embedding

Textual information is of great importance for paper similarity. However, an entity retrieval system for a bibliographic dataset should go beyond simple similar paper search. A user might also be interested in finding similar researchers to himself/herself and in searching for organizations, e.g., departments or labs, which work on similar topics for potential collaborations. Moreover, co-author networks and other relationship encoded in the metadata might also help to improve the performance of paper searching. Finally, editors may be interested in reviewer recommendations. Thanks to the increasing adoption of Semantic Web technologies, all these author-to-paper, paper-to-keyword, author-to-organization, etc. relationships are encoded in the knowledge graph. However, the symbolic representations of knowledge graphs prohibit the usage of probabilistic models which are widely used in many kinds of machine learning applications including entity retrieval systems [15]. Hence, a core question becomes how to *transform* the components of these heterogeneous networks into numerical representations such that they can be easily utilized in an entity retrieval system. Now the importance of knowledge graph embedding comes to the front.

Similar to word embedding, KG embedding aims at learning distributional representations for components of a knowledge graph while preserving the inherent structure of the original knowledge graph. Several knowledge graph embedding models have been proposed which can be classified into two groups: 1) *translation-based models* (e.g. TransE [2], TransH [17], and TransR [7]) and 2) *semantic matching models* (e.g. RESCAL, HolE [10], and DisMult [18]). In this work, we will utilize the more widely studied translation-based models because they have a clear geometric interpretation. Specifically, we use the TransE model for three reasons: 1) TransE is very efficient to run on a large knowledge graph such as LD Connect which contains 6351700 triples; 2) TransE has a very intuitive geometric interpretation which will help us understand the embedding results; 3) TransE embeds all entities and relations in the same low-dimensional vector space which is important for property path reasoning.

Given a knowledge graph G which contains a collection of triples/statements (h_i, r_i, t_i)[9], TransE embeds the entities and relations in a knowledge graph into the same low-dimensional space. Here, in a triple (h_i, r_i, t_i), h_i stands for the head entity (subject), r_i stands for the relation (predicate), and t_i is the tail entity (object). TransE treats each relation r_i as a transformation operation from the head entity h_i to the tail entity r_i. In order to set up a learning problem, a plausibility scoring function $d(h_i, r_i, t_i)$ is defined on each triple/statement (h_i, r_i, t_i) which measures the accuracy of the translation operation (See Equation 4). Here, $\mathbf{h_i}, \mathbf{r_i}, \mathbf{t_i}$ stands for the corresponding embedding of h_i, r_i, t_i which have the same dimension and $\| \, . \, \|$ represents L_1- or L_2-norm. Equation 4 implies that a correct triple observed from G will have a low plausibility score while an unobserved triple will have a relatively high score.

$$d(h_i, r_i, t_i) = \| \, \mathbf{h_i} + \mathbf{r_i} - \mathbf{t_i} \, \| \tag{4}$$

Finally, a margin-based loss function $\mathcal{L}$ is defined to set up an optimization problem (See Equation 5). Similar to word embedding and paragraph vector, negative sampling is used to accelerate the learning process. Here, G^+ represents the original knowledge graph which is a set of triples where $G^-_{(h_i, r_i, t_i)}$ stands for a set of corrupt triples from (h_i, r_i, t_i) in which either h_i or r_i is replaced with a random entity. In order to learn meaningful representations of entities and relations, the margin-based loss is minimized while the total plausibility of the observed triples is maximized. To prevent the loss from being trivially minimized by enlarging the norms of the embeddings of entities, L_2-normalization is applied on the entity embedding matrix [2].

$$\mathcal{L} = \sum\nolimits_{(h_i, r_i, t_i) \in G^+} \sum\nolimits_{(h_i', r_i', t_i') \in G^-_{(h_i, r_i, t_i)}} \left[\gamma + d(h_i, r_i, t_i) - d(h_i', r_i', t_i') \right]_+ \tag{5}$$

TransE has been applied to the entire LD Connect graph to learn the embeddings for all entities and relations. Note that each person will have one URI

[9] TransE only considers object-type properties.

for each paper (s)he authored and all these URIs are linked to each other by `owl:sameAs` relations as explained above. The same logic has been applied to organizations. We conflate these `owl:sameAs` entities to one entity before running the TransE model.

3.4 Retrieval Systems

We developed two retrieval systems based on the textual embedding model and structured embedding model. The first one implements a similar paper search interface[10] based on the learned PV-DBOW model. Users can enter text in the search bar and the interface will dynamically send a SPARQL SELECT query to the LD Connect endpoint with a *contains* filter to search for entities of type `bibo:Chapter` or `bibo:AcademicArticle` which contain the users' query in their titles. The result are visualized as a list of papers from which the user can select. The search functionality computes the cosine similarity between the paragraph vector of the query paper with all papers in the corpus and return top 20 most similar papers. Fig. 1 shows an example search. The table shows similar papers found via the PV-DBOW model and their normalized similarity. We can see all of the search results are about Semantic web and Linked Data. A quantitative evaluation of this will be discussed in Section 3.5.

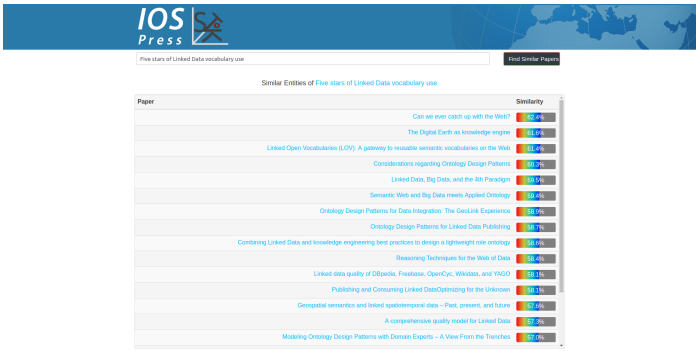

Fig. 1. Paper similarity search interface

The second retrieval system[11] is based on the TransE model which provides the option of searching different types of entities like papers, authors, journals, and organizations. After the user selects the type of entities, (s)he can enter texts in the search bar and select the entity from the list like the first system. The system will return top 20 entities with the selected type based on cosine similarity. Fig. 2 shows the result of searching for *Pascal Hitzler* who is one of the authors of the paper above. The resulting author list contains a lot of co-authors and n-degree co-authors[12] of Pascal. Moreover, the person that has more

[10] `http://stko-testing.geog.ucsb.edu:3000/ios/qe/paper`

[11] `http://stko-testing.geog.ucsb.edu:3000/ios/qe/entity`

[12] If one person p_i has a co-author relationships with both person p_j and person p_k, then we define person p_j and person p_k have a two-degree co-author relationship.

co-authored papers with the searched person should be generally ranked higher. Now, if TransE would just reveal existing co-authorship, it may be a convenient tool for look-up tasks, but not very useful as a general purpose retrieval and recommender system. However, as argued above, the system also returns other authors based on relationships between authors, between affiliations, and between outlets such as journals. for example, authors that published in the same outlets will become more similar. Put differently, TransE does preserve some of the inherent structure of the original knowledge graph. A formal evaluation of entity similarity will be discussed in Section 3.6.

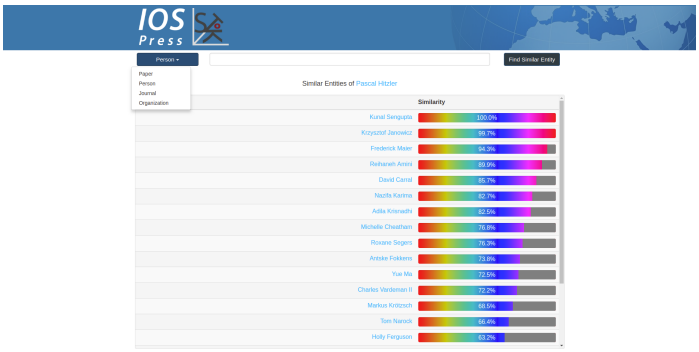

Fig. 2. Entity similarity search interface

3.5 Paper Similarity Evaluation

Next, we explore the possibility of combining these two models for a similar paper search task. Given a paper q_i within the IOS Press corpus, the paper similarity ranking task requires to fetch the top K most similar papers d_k where $k \in 1, 2, ..., K$ and rank them based on a similarity metric.

To do so we need to establish a paper similarity benchmark dataset to evaluate the ranking algorithm. In this work, we utilize the paper search API[13] from Semantic Scholar to collect a benchmark dataset. Evaluating paper similarity by hand, i.e., by asking domain experts, is a very difficult and subjective task. Hence, major search engines often rely on understanding how people search for papers and which papers they click on and download. Such massive log data is not yet available to us. Hence, by using Semantic Scholar as baseline, we can at least demonstrate that our results are in line with a major commercial product. We used the title of every paper in the IOS Press corpus to search for the top 500 similar papers in Semantic Scholar[14]. In total, 106,705 papers have been used to search for similar papers. After the search results are obtained, the DOIs and the titles of the papers in each search result list are co-referenced to the papers

[13] `https://www.semanticscholar.org/api/1/search`
[14] We filter out papers with titles containing fewer than 4 words.

in the IOS Press document corpus. As we are working within the LD Connect corpus, we filter out similar papers that are not in the corpus (as they could not have been suggested by our system) as well as those that only have two or less similar papers in our corpus. After these collection and co-referencing steps, there are 33,871 paper search results left and on average 4.96 relevant papers for each search paper.

Since the paper similarity ranking results are collected from Semantic Scholar which is also based on a machine learning approach, we cannot directly argue that the rank information itself would reflect human judgment. Instead, we treat this benchmark dataset as a binary classification results in which papers that appear in the search result are the positive samples. In order to have a balanced training dataset, the same number of papers are randomly selected from the rest of the corpus and labeled as negative samples.

The established benchmark dataset is split into training (80%) and testing datasets (20%), and a logistic regression model is applied on the training dataset. The training features of the logistic regression model are derived from the textual embedding and structure embedding model. To be more specific, given a query paper q_i and a list of papers d_k ($k \in 1, 2, ..., 2K$) where $d_1, d_2, ..., d_K$ are positive samples and $d_{K+1}, d_{K+2}, ..., d_{2K}$ are negative samples, their corresponding paragraph vectors are fetched and cosine similarity between the embeddings of q_i and d_k are computed to represent their textual similarity PV_{ik}. The same logic is applied to the learned TransE embeddings to get the structured similarity KG_{ik} between q_i and d_k. PV_{ik} and KG_{ik} are used as features to train a logistic regression model. The baseline will be models which use only one feature PV_{ik} or KG_{ik} in the logistic regression.

Table 2. The evaluation results of paper similarity binary classification task

	Precision	Recall	F1
Combined Model	0.8790	0.8372	0.8576
PV-DBOW	0.8770	0.8345	0.8552
TransE	0.6747	0.6817	0.6782

The trained logistic regression models are evaluated on the test datasets. Precision, recall, and F1 score are used as evaluation metrics. Table 2 shows the evaluation results. The logistic regression model using PV-DBOW achieves a strong performance (over 80% for each metric) while the TransE-only model shows a weaker performance and also does not add much to the combined model. Two possible explanations can be provided based on this result: 1) in the knowledge graph, each paper has relatively few incoming and outgoing degrees compared with other types of entities (such as authors) and this link sparsity issue results in insufficient learning of TransE embeddings of papers; 2) The established benchmark dataset is biased towards the textual similarity and neglects the structure similarity because of the paper similarity algorithm used by Se-

mantic Scholar. The key result is that our PV-DBOW based model yields results that are in line with commercial state-of-the-art systems.

3.6 Co-author Inference Evaluation

At first glance, the paper similarity evaluation results make the TransE model seem useless. However, it is simply designed to fulfill a complimentary task. Compared with PV-DBOW which only has embeddings for papers, TransE can obtain embeddings for every entities and relations in the knowledge graph. In order to better understand what TransE does and how the resulting embedding can be used, we performed another evaluation that infers co-authorship.

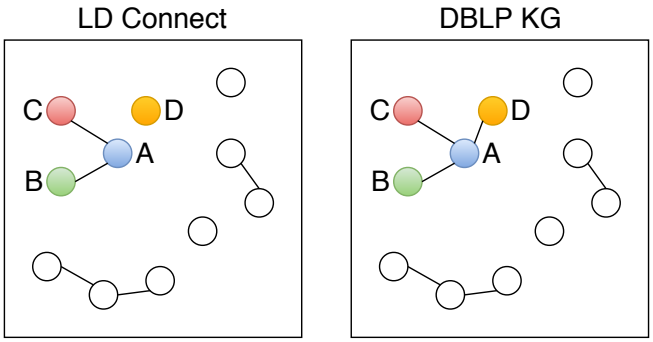

Fig. 3. An illustration of co-author inference evaluation

Fig. 3 provides an illustration of the idea of co-author inference. Node A, B, C, and D refer to four authors in two different knowledge graphs. Here, we use LD Connect and DBLP as an example. The links between nodes represent the co-author relationship. Note that two people might have more than one co-authored paper. The link in Fig. 3 represents a binary relationship. Person A has a co-author relationship with Person B, C, and D. However all knowledge graphs only store overlapping/partial information. As shown in Fig. 3, LD Connect does not have a link between A and D but DBLP does. Our hypothesis is that a similarity search on the trained TransE model for author A will likely also yield author D even tough their co-author relationship is missing in IOS Press LD Connect. Simply put, the chance of having a co-authored paper that we do not know about with a similar author is more likely than with a dissimilar author. We call this task co-authorship prediction. To the best of our knowledge it has not been tested in such setup before.

To validate our hypothesis, we collect a co-author dataset from DBLP as follows:

1. We randomly select 10,000 authors from the conflated LD Connect corpus;
2. Based on the TransE embeddings, for each selected author p_i, we obtain the top 10 similar authors p_{ik} where $k \in 1, 2, .., 10$ who have not co-authored any paper with p_i according to LD Connect;

3. For each pair of authors (p_i, p_{ik}), we search for the number of co-authored papers they have in DBLP KG which forms author pair dataset C;
4. For each selected author p_i, we also *randomly* select 10 authors $p_{ik}^{'}$ where $k \in 1, 2, .., 10$ from the conflated LD Connect;
5. For each pair of authors $(p_i, p_{ik}^{'})$, we also search for the number of their co-authored papers in DBLP KG which forms author pair dataset $C^{'}$;
6. We compute the ratio of co-author relationship for these person pairs in C and $C^{'}$ and compare them. Intuitively there should be more matching co-authors in C than $C^{'}$.

According to our experiment, there are 5.511 percent of author pairs in C which have co-author relationships in DBLP KG while there are only 1.537 percent for the randomly selected author pair dataset $C^{'}$. This result validates our assumption that the TransE model can help predict the missing co-author relationship between authors based on the observed graph structure. To put these numbers into perspective, we have shown that we can predict potential co-authorship based on author similarity. Of course, in most cases, authors are similar without having co-authored papers. In a corpus of such size two people working on, say, Semantic Web technologies will be more similar to each other in comparison to an author pair working on Alzheimer's disease and Internet Of Things.

4 Conclusion

In this work, we presented an entity retrieval system utilizing LD Connect based on textual embedding and structure embedding techniques. The retrieval model is evaluated by two benchmark datasets collected from Semantic Scholar and DBLP. In the first evaluation on paper similarity, two features derived from PV-DBOW and TransE are extracted and a binary classification model is trained on datasets collected from Semantic Scholar. Results show that TransE does not have a huge impact on improving the performance of paper similarity classification. This might be caused by the fact that the paper similarity algorithm adopted by Semantic Scholar focuses on textual similarity rather than structural similarity. As a second step, a novel co-author inference evaluation is carried out to show the effectiveness of the TransE knowledge graph embedding models for entity retrieval. A co-author pair benchmark dataset is collected from DBLP KG to demonstrate the ability of TransE for co-author inference based on the observed triples in a bibliographic dataset.

In the future, more advanced sequence models like LSTM can be used instead of PV-DBOW to capture richer information from text content. In addition, instead of learning textual embedding and structured embedding separately, we want to build a joint learning model which will help both of the embedding learning processes. In addition, instead of using a generic knowledge graph embedding model such as TransE which can be applied on any type of knowledge graphs, we want to explore ways to build a structure embedding model which specifically focuses on bibliographic knowledge graphs.

References

1. Beel, J., Gipp, B., Langer, S., Breitinger, C.: Research-paper recommender systems: A literature survey. International Journal on Digital Libraries **17**, 305–338 (2016)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems. pp. 2787–2795 (2013)
3. Breitinger, C., Gipp, B., Langer, S.: Research-paper recommender systems: a literature survey. International Journal on Digital Libraries **17**(4), 305–338 (2015)
4. Hu, Y., Janowicz, K., McKenzie, G., Sengupta, K., Hitzler, P.: A linked-data-driven and semantically-enabled journal portal for scientometrics. In: International Semantic Web Conference. pp. 114–129. Springer (2013)
5. Hu, Y., McKenzie, G., Yang, J.A., Gao, S., Abdalla, A., Janowicz, K.: A linked-data-driven web portal for learning analytics: Data enrichment, interactive visualization, and knowledge discovery. In: LAK Workshops (2014)
6. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188–1196 (2014)
7. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI. vol. 15, pp. 2181–2187 (2015)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
9. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the fifth ACM conference on Digital libraries. pp. 195–204. ACM (2000)
10. Nickel, M., Rosasco, L., Poggio, T.A., et al.: Holographic embeddings of knowledge graphs. In: AAAI. pp. 1955–1961 (2016)
11. Osborne, F., Mannocci, A., Motta, E.: Forecasting the spreading of technologies in research communities. In: Proceedings of the Knowledge Capture Conference. pp. 1–8. ACM (2017)
12. Osborne, F., Scavo, G., Motta, E.: A hybrid semantic approach to building dynamic maps of research communities. In: International Conference on Knowledge Engineering and Knowledge Management. pp. 356–372. Springer (2014)
13. Ritchie, A., Teufel, S., Robertson, S.: Using terms from citations for ir: some first results. In: European Conference on Information Retrieval. pp. 211–221. Springer (2008)
14. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18**(11), 613–620 (1975)
15. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017)
16. Wang, S., Tang, J., Aggarwal, C., Liu, H.: Linked document embedding for classification. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 115–124. ACM (2016)
17. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI. vol. 14, pp. 1112–1119 (2014)
18. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of International Conference on Learning Representations (2015)