# A Study of the Similarities of Entity Embeddings Learned from Different Aspects of a Knowledge Base for Item Recommendations

Guangyuan Piao(✉) and John G. Breslin

Insight Centre for Data Analytics, Data Science Institute,
National University of Ireland, Galway, Ireland
guangyuan.piao@insight-centre.org, john.breslin@nuigalway.ie

**Abstract.** The recent development of deep learning approaches provides a convenient way to learn entity embeddings from different aspects such as texts and a homogeneous or heterogeneous graph encoded in a knowledge base such as DBpedia. However, it is unclear to what extent domain-specific entity embeddings learned from different aspects of a knowledge base reflect their similarities, and the potential of leveraging those similarities for item recommendations in a specific domain has not been explored. In this work, we investigate domain-specific entity embeddings learned from different aspects of DBpedia with state-of-the-art embedding approaches, and the recommendation performance based on the similarities of these embeddings. The experimental results on two real-word datasets show that recommender systems based on the similarities of entity embeddings learned from a homogeneous graph via the `dbo:wikiPageWikiLink` property provides the best performance compared to the ones learned from other aspects.

**Keywords:** Deep learning · Semantic similarity · Knowledge base
Entity embeddings · Recommender systems · Knowledge graph

## 1 Introduction

Knowledge bases (KBs) such as DBpedia [12] and Wikidata [29] have received great attention in the past few years due to the embedded knowledge which is useful for a wide range of tasks including recommender systems [3]. For example, Linked Open Data-enabled recommender systems (LODRS) aim to utilize the background knowledge about items (entities) from linked datasets such as DBpedia for improving the quality of recommendations [6,7]. However, most previous studies on LODRS view a KB as a heterogeneous knowledge graph (KG) based on the domain-specific entities and properties defined in an ontology (e.g., DBpedia ontology). Take DBpedia as an example, the heterogeneous KG can be


A. Gangemi et al. (Eds.): ESWC 2018 Satellite Events, LNCS 11155, pp. 345–359, 2018.
https://doi.org/10.1007/978-3-319-98192-5_52

seen as one aspect of a knowledge base, and a KB can contain several aspects of knowledge with respect to entities (see Fig. 1) such as:

- *Textual knowledge*: This type of knowledge denotes textual knowledge about entities, e.g., the abstracts of movies via `dbo`[1]`:abstracts` property.
- *Knowledge from a homogeneous graph*: This type of knowledge denotes the inherited knowledge from Wikipedia[2] based on the `dbo:wikiPageWikiLink` property, which provides a set of connected entities via the same property.
- *Knowledge from a heterogeneous graph*: This type of knowledge is powered by the heterogeneous graph, which consists of domain-specific entities and other nodes connected to those entities via different properties defined in the ontology of a KB, and has been widely used for extracting background knowledge about items (entities) for LODRS.
- *Visual knowledge*: This denotes visual information about entities, e.g., the thumbnails of movies via `dbo:thumbnail` property.

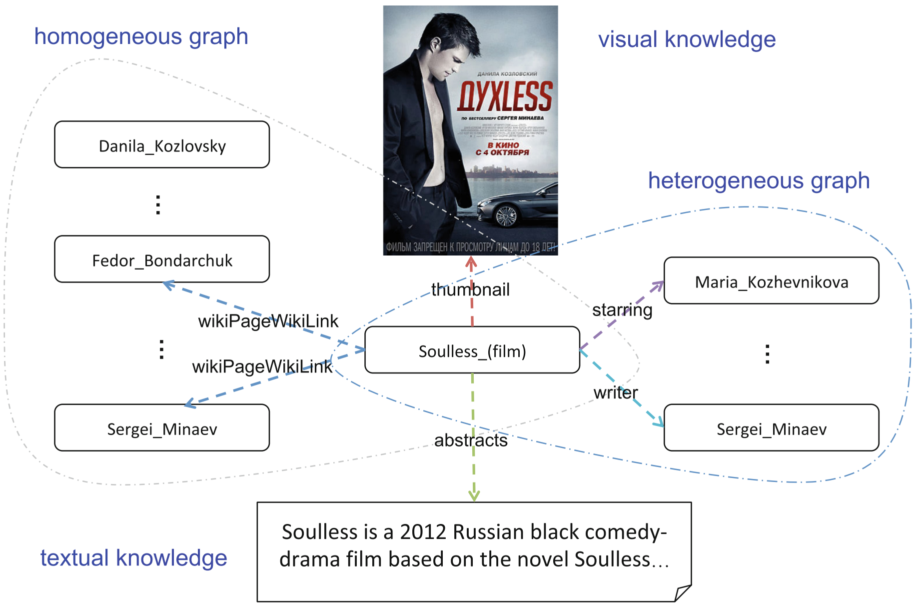



**Fig. 1.** Knowledge about the movie entity `Soulless_(film)` from different aspects of DBpedia.

Recently, a great number of studies have been proposed to learn entity embeddings in a KG for the KG completion task [4] or for other classification or recommendation tasks using those low-dimensional representations of entities as

[1] The prefix `dbo` denotes http://dbpedia.org/ontology/.
[2] https://www.wikipedia.org.

features [26,27] based on deep learning approaches. While related work reveals several insights regarding the entity embeddings learned from the heterogeneous graph of a KB, there exists little research on understanding the similarities between those entity embeddings learned from other aspects of KBs. There has been considerable semantic similarity/distance measures which were designed for measuring the similarity/distance between entities in the same domain in linked datasets such as DBpedia for various purposes such as item recommendations in a cold start. This preliminary work can be seen as being in the same direction as these studies but with the focus on investigating the similarities between entity embeddings learned from embedding approaches using deep learning or factorization models with domain knowledge.

In this preliminary work, we aim to investigate the semantic similarities of entity embeddings learned from different aspects of a KB, and evaluate them in the context of item recommendations in the music and book domains. Specifically, we focus on the textual knowledge, knowledge from a homogeneous or heterogeneous graph based on dedicated embedding approaches including deep learning techniques. Deep learning approaches have been proved to be effective on learning the latent representations of various forms of data such as images, texts, as well as nodes in networks. Therefore, we use `Doc2Vec` [10] and `Node2Vec` [8] to learn the entity embeddings based on the textual knowledge and the knowledge from a homogeneous graph, and use an embedding model for knowledge graph completion to learn the entity embeddings based on the heterogeneous graph of a KB. We use DBpedia as our knowledge base in this work. In particular, we are interested in investigating the following research questions with results in Sect. 5:

- *How do those entity embeddings learned from different aspects of a KB reflect the similarities between items (entities) in a specific domain in the context of item recommendations in a cold start?*
- *Do those entity embeddings learned from different aspects complement each other?*

To the best of our knowledge, this is the first work on investigating the semantic similarities between entity embeddings learned from different aspects of a KB, and exploring their usages in the context of recommender systems. A shorter version of this paper was published in the CEUR proceedings of the 1st Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies [23].

## 2 Related Work

Here we review some related work on linked data similarity/distance measures for measuring the similarity/distance between two entities in a specific domain for recommendation purposes, and the approaches exploring entity embeddings for item recommendations.

### 2.1 Linked Data Similarity/Distance Measures

`LDSD` [19,20] is one of the first approaches for measuring the linked data semantic distance between entities in a linked dataset such as DBpedia. Leal et al. [11] proposed a similarity measure which is based on a notion of *proximity*. This method measures how connected two entities are (e.g., based on the number of paths between two entities), rather than how distant they are. Piao et al. [21] revised `LDSD` in order to satisfy some fundamental axioms as a distance-based similarity measure, and further improved it based on different normalization strategies [22]. More recently, Alfarhood et al. [1] considered additional resources beyond the ones one or two hops away in `LDSD`, and the same authors also proposed applying link differentiation strategies for measuring the linked data semantic distance between two entities in DBpedia [2]. In contrast to aforementioned approaches, Meymandpour et al. [14] proposed a information content-based semantic similarity measure for measuring the similarity between two entities in linked open data cloud, which can consider multiple linked datasets for measuring the similarity. In this work, we are interested in the similarities of entity embeddings learned from different aspects of a knowledge base, and compare those similarities with one of the semantic similarity/distance measures [22].

### 2.2 Exploring Entity Embeddings for Item Recommendations

Recently, entity embeddings learned from a knowledge graph using deep learning approaches have been used for item recommendations. In [27], the authors proposed `RDF2Vec`, which runs random walks on a heterogeneous RDF[3] graph in DBpedia, and then applies `Word2Vec` [15,16] techniques by treating the sequences of triples as sentences. The learned entity embeddings based on the whole KG were then used to find the $k$-nearest neighbors of items. Afterwards, those neighbors were used as side information for factorization machines [25] for providing item recommendations. In contrast to [27] which uses the whole KG for learning entity embeddings, we learn domain-specific entity embeddings from different aspects of a KB. The entity embeddings learned from the whole KG might reflect relatedness of entities instead of their similarities as they are learned by incorporating all properties and nodes from other domains. However, related entities are not always similar, e.g., a musical artist and his/her spouse are related but not similar.

Zhang et al. [30] proposed collaborative knowledge base embedding, which jointly learn the latent representations in collaborative filtering for item recommendations as well as the ones for a knowledge base. However, those entity embeddings were used as features and the similarities between them were not investigated. Palumbo et al. [18] used domain-specific triples from DBpedia for learning entity embeddings with `Node2Vec` for item recommendations. In order to use `Node2Vec` for the heterogeneous graph based on domain-specific properties, the authors applied `Node2Vec` to each heterogeneous graph which consists

[3] https://www.w3.org/RDF/.

of all triples based on a single property. Afterwards, those property-specific similarity scores were used as features for a learning-to-rank framework with the training dataset. In contrast, we are interested in the entity embeddings learned from the heterogeneous graph and the similarities between those embeddings.

## 3 Learning Entity Embeddings from Different Aspects of DBpedia

In this section, we discuss three state-of-the-art embedding/vectorization approaches that we adopted for learning entity embeddings based on different aspects of knowledge from DBpedia.

### 3.1 Entity Embeddings with Textual Knowledge

`Doc2Vec` [10], which is inspired by `Word2Vec` [15,16], was devised for learning embeddings for larger blocks of text such as documents or sentences. This model uses document vectors and contextual word vectors to predict the next word, which is a multi-class classification task. Figure 2 shows the `Doc2Vec` model, where each document/paragraph is mapped to a latent vector which is a column in a document/paragraph matrix $D$, and each word has its embedding which is a column in a word embedding matrix $W$. As we can see from the figure, the document vector and word vectors are concatenated to predict the next word in a context. The document and word vectors can be learned by optimizing the classification error in a given set of documents. For example, with a window size 8, the model predicts the 8th word based on the document and 7 contextual word vectors. We used the `gensim` [24] implementation of `Doc2Vec` for our experiment.

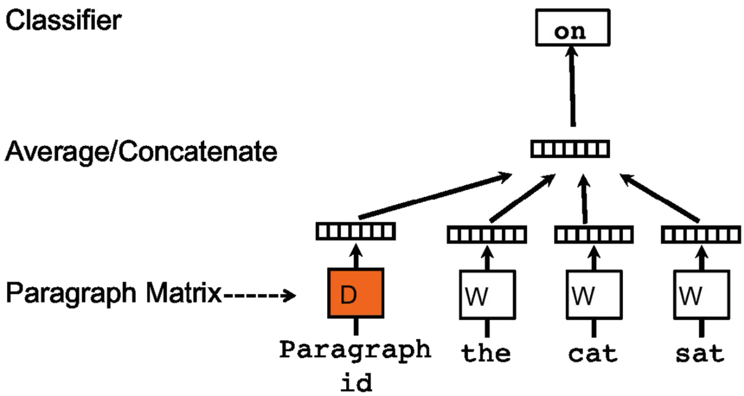



**Fig. 2.** `Doc2Vec` model for learning document/paragraph vector [10].

In our experiment, each abstract of an entity in a specific domain is a document, which is provided by the `dbo:abstracts` property, and the set of all abstracts is used for learning entity embeddings in a specified domain with the `Doc2Vec` model. The window size is set to 8 in the same way as [10].

### 3.2 Entity Embeddings with a Homogeneous Graph

`Node2Vec` [8], which is also inspired by `Word2Vec` [15], aims to learn latent representations of nodes for a homogeneous network. It extends the Skip-gram architecture (see Fig. 3) to networks, and optimizes the (log) probability of observing a network neighborhood for each node. To apply the Skip-gram model for networks, `Node2Vec` first executes random walks based on a defined searching strategy, and the sequence of nodes obtained via the search is used for the Skip-gram model. We used the author's implementation[4] for our experiment.

In our study, we treat the graph which consists of all items in a specific domain and other connected nodes to those items via the `dbo:wikiPageWikiLink` property as the homogeneous graph from DBpedia, and apply `Node2Vec` to learn the entity embeddings based on this homogeneous graph.

*Parameters.* We choose smaller values for some parameters compared to the settings in the original paper as there is a great number of `dbo:wikiPageWikiLink` relationships, which takes a long time for training the model due to its expensiveness. Our settings for the main hyperparameters of `Node2Vec` are as follows:

– `walk_length=10`: The length of walk for each node.
– `num_walks=10`: The number of walks per node.
– `p=q=1`: `p` and `q` denote the return and in-out hyperparameters for random walks, respectively.
– `window_size=5`: The context size for optimization.

### 3.3 Entity Embeddings with a Heterogeneous Graph

`TransE` [4] is a translation-based model for knowledge graph completion by learning the embeddings of entities and their relationships. In short, `TransE` learns those embeddings in order to satisfy $E(s)+E(p) \approx E(o)$ for a valid triple $(s, p, o)$ in a knowledge base, where $E(x)$ denotes $x$'s embedding. Although `TransE` has been used for learning entity embeddings for KG completion by considering all triples in a KG, for item recommendations in a specific domain, most previous studies extract the domain-specific DBpedia graph which consists of all entities in that domain and incoming or outgoing nodes via domain-specific properties [19,22]. Therefore, to learn domain-specific entity embeddings, we extract all triples for the entities/items in that domain with relevant properties. In consistence with a previous work [22], we used the top-15 properties for each domain in order to obtain all triples for the subjects in that domain. Table 1 shows those properties we used to extract domain knowledge about items for our experiment in Sect. 4.

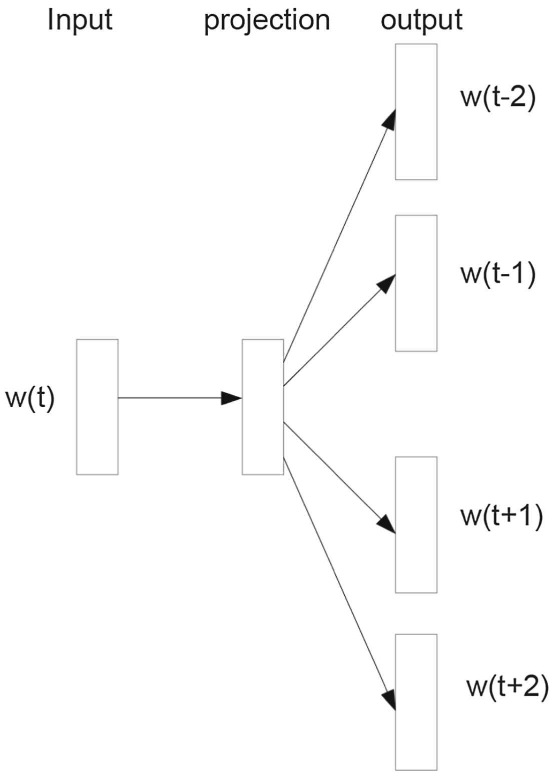



**Fig. 3.** The Skip-gram model architecture from [16], which aims to learn word vector representations that are good at predicting the nearby words.

The dimensionality of entity embeddings is set to 100 for all three approaches, and the trained embeddings are available at https://github.com/parklize/DL4KGS. For our experiment, we used the HDT [5] dump for the DBpedia 2016-04 version, which is available at http://www.rdfhdt.org/datasets/.

## 4 Experiment Setup

We evaluate the similarities of entity embeddings learned from different aspects of DBpedia in the context of cold-start scenarios in recommender systems where the top-N items are recommended based on the cosine similarities between entity embeddings, which are learned from different aspects of DBpedia.

[4] http://snap.stanford.edu/node2vec.

**Table 1.** The top-15 domain-specific properties used for extracting valid triples from DBpedia and for training `TransE`.

| Music | Book |
|---|---|
| – dct:subject | – dct:subject |
| – dbo:genre | – dbo:author |
| – dbo:associatedBand | – dbo:publisher |
| – dbo:associatedMusicalArtist | – dbo:literaryGenre |
| – dbo:instrument | – dbo:mediaType |
| – dbo:recordLabel | – dbo:subsequentWork |
| – dbo:occupation | – dbo:previousWork |
| – dbo:hometown | – dbo:country |
| – dbo:bandMember | – dbo:series |
| – dbo:formerBandMember | – dbo:nonFictionSubject |
| – dbo:currentMember | – dbo:coverArtist |
| – dbo:influencedBy | – dbo:illustrator |
| – dbo:pastMember | – dbo:genre |
| – dbo:associatedAct | – dbo:translator |
| – dbo:influenced | – dbo:recordLabel |

### 4.1 Datasets

We use two real-world datasets in the music and book domains for our experiment. The first dataset is a last.fm dataset from [17], which consists of 232 musical artists, and the top-10 similar artists for each of the 232 artists obtained from last.fm. Those top-10 similar artists provided in last.fm for each artist are used as the ground truth. The second dataset is a dbbook dataset[5] in the book domain, which consists of 6,181 users and 6,733 items which have been rated by at least one user. We randomly selected 300 users who have liked at least 10 books for our experiment. For each user, we randomly chose one item and recommended items similar to the chosen one based on their similarities. Therefore, the other books liked by each user except the chosen one are used for our ground truth here. For both datasets, all items in each dataset are considered as candidate items for recommendations.

To learn domain-specific entity embeddings, we extracted background knowledge from DBpedia for all entities/items in two domains: the music and book domains. The subjects in the music domain are the entities that have their `rdf:type`(s) as `dbo:MusicalArtist` and `dbo:Band`, and the subjects in the book domain are the ones that have their `rdf:type`(s) as `dbo:Book`. After obtaining all subjects, we further obtain their abstracts, connected nodes (entities and categories) via the `dbo:wikiPageWikiLink` property, and the connected nodes

[5] http://challenges.2014.eswc-conferences.org/index.php/RecSys#DBbook_dataset.

via those properties defined in Table 1. Table 2 shows the details of the domain knowledge with respect to the music and book domains.

**Table 2.** The statistics of background knowledge about items from different aspects of DBpedia.

| | Music | Book |
|---|---|---|
| # subjects | 171,812 | 76,639 |
| # abstracts | 131,622 | 70,654 |
| # wikiPageWikiLinks | 5,480,222 | 2,340,146 |
| # triples | 1,481,335 | 316,969 |

### 4.2 Evaluation Metrics

The recommendation performance is evaluated by the evaluate metrics below:

- **P@N**: Precision at rank N (P@N) is the proportion of the top-N recommendations that are relevant to the user, which is measured as follows:

$$P@N = \frac{|\{relevant\ items@N\}|}{N}$$

- **R@N**: Recall at rank N (R@N) represents the mean probability that relevant items are successfully retrieved within the top-N recommendations.

$$R@N = \frac{|\{relevant\ items@N\}|}{|\{relevant\ items\}|}$$

- **nDCG@N**: nDCG (Normalized Discounted Cumulative Gain) takes into account rank positions of the relevant items. nDCG@N can be computed as follows:

$$nDCG@N = \frac{1}{IDCG@N} \sum_{k=1}^{N} \frac{2^{\hat{r}_{uk}} - 1}{\log_2(k+1)}$$

  where $\hat{r}_{uk}$ is the relevance score of the item at position $k$ with respect to a user $u$ in the top-N recommendations, and the normalization factor IDCG@N denotes the score obtained by an ideal top-N ranking.

We used the paired t-test in order to test the statistical significance where the significance level is set to 0.05.

### 4.3 Compared Methods

We compare the similarity measures below to evaluate the similarities of item embeddings based on different aspects of DBpedia:

– `Resim` [22]: This is a semantic distance/similarity measure for LOD dataset such as DBpedia, which measures the similarity based on the direct and indirect properties between two entities. We use the implementation from our previous work[6] for our experiment.
– $\texttt{Cos}(\texttt{V}_{\texttt{tk:Doc2Vec}})$: This method uses the cosine similarity measure for the entity embeddings learned from textual knowledge of entities from DBpedia using `Doc2Vec`.
– $\texttt{Cos}(\texttt{V}_{\texttt{hmk:Node2Vec}})$: This method uses the cosine similarity measure for the entity embeddings learned from homogeneous graph knowledge of entities from DBpedia using `Node2Vec`.
– $\texttt{Cos}(\texttt{V}_{\texttt{htk:TransE}})$: This method uses the cosine similarity measure for the entity embeddings learned from heterogeneous graph knowledge of entities from DBpedia using `TransE`.
– $\texttt{Cos}([\texttt{V}_\texttt{x}, \texttt{V}_\texttt{y}])$: This method uses the cosine similarity measure for the concatenated entity embeddings learned from several aspects of entities from DBpedia. For example, $\texttt{Cos}([\texttt{V}_{\texttt{htk:TransE}}, \texttt{V}_{\texttt{tk:Doc2Vec}}])$ denotes the method using the cosine similarity measure for the concatenated entity embeddings based on `TransE` and `Doc2Vec`, and `Cos([all])` denotes the concatenated ones based on all embedding approaches.

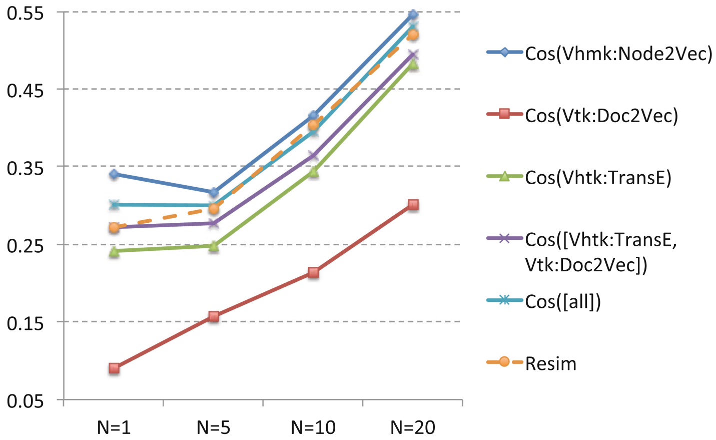
## 5 Results

Figures 4 and 5 show the nDCG@N results and the precision-recall curve of item recommendations based on the similarities of different entity embeddings in the music and book domains. Overall, we observe that the recommendations based on the entity embeddings with `Node2Vec` provide the best performance followed by the ones with `TransE` and `Doc2Vec`.

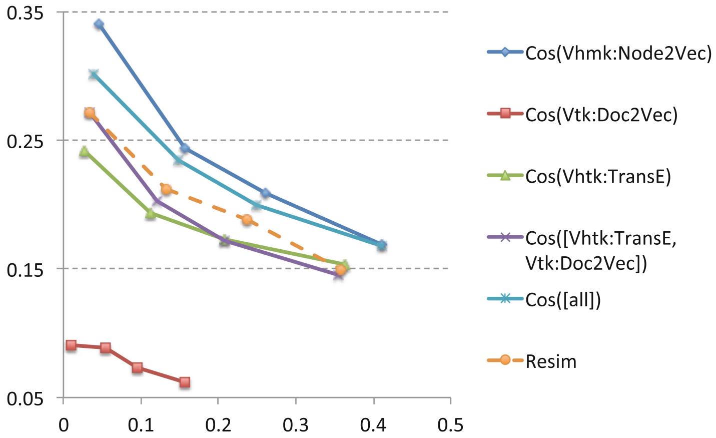
In both datasets, the results using the embeddings learned from `Node2Vec` significantly outperform the ones learned from `TransE` and `Doc2Vec`, which show that the great amount of information provided by `dbo:wikiPageWikiLink` reflects the similarities between entities better than other aspects of DBpedia. We also observe that combining the embeddings based on `TransE` and `Doc2Vec` improves the recommendation performance significantly compared to using the embeddings learned from `TransE` or `Doc2Vec`. However, combining all embeddings learned from the three different aspects do not provide further improvement on the recommendation performance. Also, the concatenated embeddings with `Node2Vec` and other embeddings do not provide better performance compared to using the ones learned from `Node2Vec` alone, and the results are omitted from Figs. 4 and 5 for clarity.

[6] https://github.com/parklize/resim.

In the last.fm dataset, we observe some significant improvement of `Node2Vec` and `Cos([all])` over `Resim`. For example, the recommendation performance is improved by 25.4% and 11.1% with `Node2Vec` and `Cos([all])` compared to using `Resim`. In contrast, there is no statistical difference between the recommendation performance using those embeddings and using `Resim` in the dbbook dataset. This might be due to the relatively small size of subjects in the book domain and their related aspects for training those embeddings.

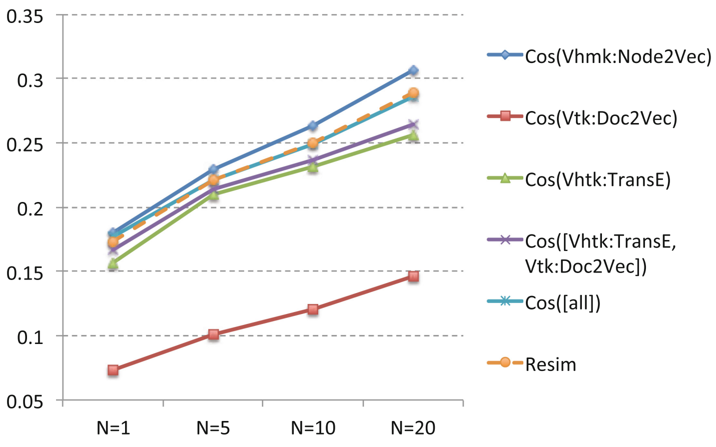



(a) nDCG@N

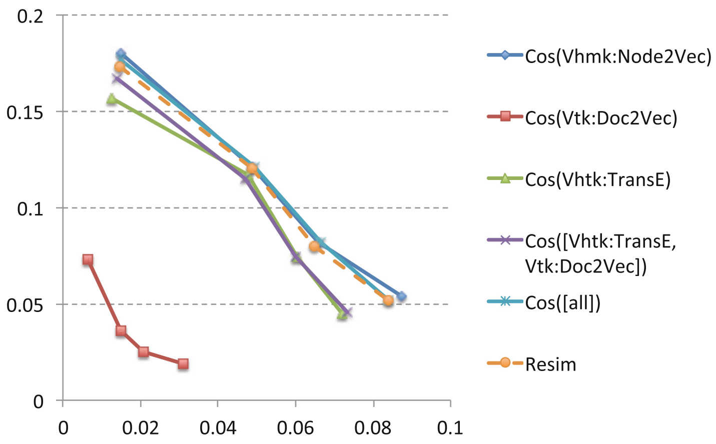



(b) P@N (y-axis) and R@N (x-axis) curve when N = 1, 5, 10, 20.

**Fig. 4.** The performance of item recommendations on the last.fm dataset with all methods compared.




(a) nDCG@N




(b) P@N (y-axis) and R@N (x-axis) curve when N = 1, 5, 10, 20.

**Fig. 5.** The performance of item recommendations on the dbbook dataset with all methods compared.

## 6 Conclusions and Future Work

In this paper, we investigated the embeddings learned from three different aspects of DBpedia using state-of-the-art deep learning and embedding-based approaches, and the recommendation performance based on the similarities captured by those embeddings in two real-world datasets. The preliminary results indicate that the entity embeddings learned from the homogeneous graph powered by the `dbo:wikiPageWikiLink` property provide the best performance in the context of item recommendations compared to the ones learned from other aspects of DBpedia. We further explored potential synergies that exist by combining those embeddings learned from different aspects. The concatenated embeddings with the ones learned from textual knowledge (using `Doc2Vec`) and the heterogeneous graph (using `TransE`) significantly improves the performance. This preliminary study can be seen as a first step towards investigating the similarity between entity embeddings learned from different aspects of a knowledge base for item recommendations, and also poses many research questions for future work.

First, although we used state-of-the-art approaches for learning entity embeddings from different aspects, there are many other state-of-the-art alternatives for learning entity embeddings such as `Tweet2Vec` [28] for learning entity embeddings based on their abstracts, and `ETransR` [13] for learning the embeddings based on the heterogeneous graph. A further investigation of using other deep learning and embedding-based approaches for learning entity embeddings for different aspects of a knowledge base is required.

Secondly, how to choose domain-specific triples out of all triples in the knowledge base is a remaining question. Using triples extracted with domain-specific properties might lead to a smaller number of triples for those embedding-based approaches to learn good entity embeddings. In contrast, using the whole heterogeneous graph might lead to general entity embeddings which tend to capture their relatedness instead of the similarities. Further research is needed to confirm the hypothesis, and a recent approach such as [9] for extracting domain-specific subgraphs can be further explored for extracting domain-specific triples for training the entity embeddings in that domain.

Finally, the results of this study showed that concatenating all embeddings does not further improve the performance, and those results suggest more research is needed for combining those entity embeddings which are learned from different aspects of a knowledge base.

**Acknowledgments.** This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 (Insight Centre for Data Analytics).

## References

1. Alfarhood, S., Labille, K., Gauch, S.: PLDSD: propagated linked data semantic distance. In: 2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 278–283 (2017)
2. Alfarhood, S., Gauch, S., Labille, K.: Employing link differentiation in linked data semantic distance. In: Różewski, P., Lange, C. (eds.) KESW 2017. CCIS, vol. 786, pp. 175–191. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69548-8_13
3. Blomqvist, E.: The use of Semantic Web technologies for decision support - a survey. Semant. Web **5**(3), 177–201 (2014). https://doi.org/10.3233/SW-2012-0084
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data, pp. 2787–2795 (2013)

5. Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary RDF representation for publication and exchange (HDT). Web Semant. Sci. Serv. Agents World Wide Web **19**, 22–41 (2013). http://www.sciencedirect.com/science/article/pii/S1570826813000036
6. Figueroa, C., Vagliano, I., Rodríguez Rocha, O., Morisio, M.: A systematic literature review of Linked Data-based recommender systems. Concurrency Comput. **27**(17), 4659–4684 (2015)
7. de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-aware content-based recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 119–159. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_4
8. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. CoRR abs/1607.0 (2016). http://arxiv.org/abs/1607.00653
9. Lalithsena, S., Kapanipathi, P., Sheth, A.: Harnessing relationships for domain-specific subgraph extraction: a recommendation use case. In: IEEE International Conference on Big Data, Washington D.C. (2016)
10. Le, Q.V., Mikolov, T.: Distributed Representations of Sentences and Documents. CoRR abs/1405.4 (2014). http://arxiv.org/abs/1405.4053
11. Leal, J.P., Rodrigues, V., Queirós, R.: Computing semantic relatedness using DBpedia (2012)
12. Lehmann, J., et al.: DBpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semant. Web J. **6**(2), 167–195 (2013)
13. Lin, H., Liu, Y., Wang, W., Yue, Y., Lin, Z.: Learning entity and relation embeddings for knowledge resolution. Procedia Comput. Sci. **108**, 345–354 (2017)
14. Meymandpour, R., Davis, J.G.: A semantic similarity measure for linked data: an information content-based approach. Knowl. Based Syst. **109**, 276–293 (2016)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR abs/1301.3 (2013). http://arxiv.org/abs/1301.3781
16. 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)
17. Oramas, S., Sordo, M., Espinosa-Anke, L., Serra, X.: A semantic-based approach for artist similarity. In: ISMIR, pp. 100–106 (2015)
18. Palumbo, E., Rizzo, G., Troncy, R.: Entity2Rec: learning user-item relatedness from knowledge graphs for top-N item recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, pp. 32–36. ACM, New York (2017). https://doi.org/10.1145/3109859.3109889
19. Passant, A.: dbrec — music recommendations using DBpedia. In: Patel-Schneider, P.F., et al. (eds.) ISWC 2010. LNCS, vol. 6497, pp. 209–224. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17749-1_14
20. Passant, A.: Measuring semantic distance on linking data and using it for resources recommendations. In: Proceedings of the AAAI Spring Symposium: Linked Data Meets Artificial Intelligence. vol. 77, pp. 93–98 (2010). files/129/display.html
21. Piao, G., Ara, S., Breslin, J.G.: Computing the semantic similarity of resources in dbpedia for recommendation purposes. In: Qi, G., Kozaki, K., Pan, J.Z., Yu, S. (eds.) JIST 2015. LNCS, vol. 9544, pp. 185–200. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31676-5_13
22. Piao, G., Breslin, J.G.: Measuring semantic distance for linked open data-enabled recommender systems. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing, 04–08 April 2016, pp. 315–320. ACM, Pisa (2016)

23. Piao, G., Breslin, J.G.: A study of the similarities of entity embeddings learned from different aspects of a knowledge base for item recommendations. In: 1st Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies at the 15th Extended Semantic Web Conference (2018). CEUR-WS.org
24. Radim Rehurek, P.S.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta, May 2010
25. Rendle, S.: Factorization machines with libFM. ACM Trans. Intell. Syst. Technol. **3**(3), 57:1–57:22 (2012)
26. Ristoski, P., Paulheim, H.: RDF2Vec: RDF graph embeddings for data mining. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9981, pp. 498–514. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46523-4_30
27. Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., Paulheim, H.: RDF2Vec: RDF graph embeddings and their applications. Semant. Web J. (2018). http://www.semantic-web-journal.net/system/files/swj1495.pdf
28. Vosoughi, S., Vijayaraghavan, P., Roy, D.: Tweet2Vec: learning tweet embeddings using character-level CNN-LSTM encoder-decoder. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016, pp. 1041–1044. ACM, New York (2016). https://doi.org/10.1145/2911451.2914762
29. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
30. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 353–362. ACM, New York (2016)