Query Resolution of Literature Knowledge Graphs using Hybrid Document Embeddings

Iqra Muhammad, Frans Coenen, Carol Gamble, Anna Kearney, and Paula Williamson

The University of Liverpool, Liverpool, L693BX, UK
iqra@liverpool.ac.uk

Abstract. Literature Knowledge Graphs play a critical role in helping domain experts carry out query resolution for finding relevant articles in published literature. Such knowledge graphs are usually in the form of Curated Document Databases (CDDs). Domain Experts and researchers typically query such literature knowledge graphs using some form of query-resolution mechanism. Machine learning techniques can be used to automate query-resolution. This paper presents a document query-resolution mechanism, given a query and set of documents in a knowledge graph, based on a hybrid word embedding that combines knowledge graph embeddings with "traditional" embeddings. A query-document data set extracted from a clinical trials CDD (the ORRCA CDD) was used. Three "traditional" word embeddings were considered: CBOW, BERT and SciBERT. The evaluation demonstrated that hybrid embeddings produced better results than when the embedding models were used in isolation. A best Mean Average Precision of 0.486 was obtained when using a CBOW and random walk knowledge graph hybrid embedding.

Keywords: Query Resolution, Word embedding, Document ranking

1 Introduction

The number of published papers in the scientific domain has increased year-on-year. As a consequence researchers find it increasingly cumbersome to find relevant literature. Researchers typically find relevant publications using a query-resolution mechanism directed at some document repository such as Google Scholar or PubMed. The query resolution process can be made more effective if a more domain specific document repository is used. The fundamental idea underpinning query-resolution is that, given a search query, potential documents matched to the query can be ranked according to how well they match the query and the top k documents returned because these are considered to be the most relevant to the query. This requires that the query and documents are represented in a way that facilitates matching. The most common approach is to use some form of word embedding. A word embedding is a learned text representation whereby each word or phrase in a document or query is represented by a numerical vector. A document embedding for each document in document repository

(CDD) can then be generated by averaging the individual word embeddings. A query embedding can be generated in a similar manner.

The query-resolution process can be made even more effective if the document repository (CDD) is encapsulated in the form of a literature knowledge graph, as opposed to the traditional relational database typically adopted, because knowledge graphs impose a structure on the data that avoids the need for exhaustive searching when responding to queries.

One example of a CDD represented as a literature knowledge graph is the Online Resource for Recruitment research in Clinical trials[1]. The ORRCA CDD was developed to bring together scientific literature focused on the topic of clinical trials. There are various techniques, based on word embeddings, to support query-resolution using literature knowledge graphs. Some recent examples can be found in [3, 4, 28]. However, these examples all used "traditional" embeddings.

Recently, many word embedding methods based on deep learning neural networks have been used for query and document representation so as to facilitate the effective scoring of documents with respect to query resolution [2, 9]. Examples include: (i) Continuous Bag of Words (CBOW) [7] (ii) Bidirectional Encoder Representations from Transformers (BERT) embedding [9] (iii) Sci-BERT embedding [2]. However, when applied to literature knowledge graph represented CDDs these embedding techniques ignore the "knowledge" that is inherently available as a consequence of the knowledge graph structure.

The central hypothesis that the work presented in this paper seeks to address is that query resolution can be made more effective if a hybrid embedding is used whereby an established word embedding is combined with a literature knowledge graph embedding [1, 14, 22]. More specifically a random walk knowledge graph embedding generated by conducting a random walk over a literature knowledge graph is advocated, as suggested in [11][13][25]. Experiments were conducted using three different "traditional" embeddings (CBOW, BERT and Sci-Bert) combined with a random walk embedding; and when using these embeddings in isolation.

The remainder of this paper is structured as follows. A literature review of query-resolution mechanisms is first given in Section 2. Then, in Section 3, a review of the proposed query resolution approach is given. Section 4 gives a review of Random Walk Knowledge Graph Embeddings. The conducted evaluation of the approach is reported on in Section 5. The paper is concluded in Section 6 with the main findings.

2 Literature Review

The work presented in this paper is directed at a hybrid embedding approach, where two embedding are used to represent documents stored in a literature knowledge graph and user queries directed at that graph. The idea is to combine

[1] https://www.orrca.org.uk/

a graph embedding, that captures the information within a literature knowledge graph, and a more traditional word embedding. Word embeddings are typically generated using a deep learning embedding model [24, 10]. However, to train an embedding model requires a large amount of data and consequently significant processing power, which means that the generation of a dedicated embedding model for a specific application domain, including the clinical trails domain considered in this paper, is not a viable option. The solution is the adoption/adaptation of an existing embedding model. There are many embedding models that have been reported on in the literature and three are considered in this paper: (i) CBOW, (ii) BERT and (iii) Sci-BERT. Word embedding can be categorised as being either: (i) contextual or (ii) non-contextual. Non-contextualized word embeddings do not take into account the surrounding word context of a word whereas a contextualized embedding does. CBOW embeddings are an example of the first. BERT and Sci-BERT embedding are examples of the second. All of these word embedding models can be used in the context of transfer learning. Further detail concerning non-contextualized embeddings are presented in Sub-section 2.1, whilst contextualized embedding models are considered in Sub-section 2.2. The section is concluded, in Sub-section 2.3, with a discussion concerning existing work on knowledge graph embedding models.

2.1 Non-Contextualized Embedding Models - CBOW

Non-contextualized embedding models do not consider an individual word's context within a document. A popular class of such embedding model is the Word2Vec model. The input to Word2Vec is a word and the output is an embedding. Some examples of Word2Vec models are the Continuous Bag Of Words (CBOW) model and the Skip-gram model [7]. The biggest benefit of using these techniques is that they can be used at scale, in real world settings, without requiring a significant amount of time to tune to a specific domain of interest (not the case when using contextualized embedding models like BERT). For the work presented in this paper the CBOW model was considered because it is exemplar of a non-contextualized embedding model and because of the good performance reported in the literature [29]. CBOW is trained by considering each word in each document in sequence using a sliding window and produces an embedding for each input word. Once training is complete the CBOW system is no longer required. Examples of reported work where CBOW embeddings have been used for query resolution can be found in [6, 12, 20].

2.2 Contextualized Embedding Models - BERT and SciBERT

Contextualized word embedding techniques are based on deep learning neural networks. The benefit of using contextualized word embedding models is that they take into account the surrounding context of a word. The difference between contextualized and non-contextualized models can be explained by considering the following two sentences:

The man was accused of robbing a bank.
The man went fishing by the bank of the river.

A non-contextualized embedding model would generate the same word embedding for the word “bank” in both cases, whereas a contextualized embedding system would generate different word embedding depending on the context of the word “bank”. As noted above, the advantage of non-contextualized embedding models over contextualized models is that they are easy to train and can be easily deployed at scale. However, contextualized models can be shown to produce embeddings that better reflect a given text [9, 27]. With respect to the work presented in this paper, BERT and Sci-BERT were considered as exemplars of contextualized models. Sci-BERT is a variation of BERT directed at scientific applications, and thus it was considered to be suited to the clinical trials application domain used as a focus for the work presented in this paper.

2.3 Knowledge Graph Embedding Models

There are various algorithms for the generation of knowledge graph embeddings used with respect to question answering and document/query representation in document retrieval. Some of such well-known knowledge graph embedding algorithms are Deep Walk [16], LINE [21] and Node2Vec [5]. With respect to the work in this paper, *Node2Vec* was used because of its ease of use and it being scalable for larger knowledge graphs as seen in recent literature [22, 26]. A random walk consists of simulating a walk over a set of vertices in a knowledge graph. The output of a random walk is a set of sentences that are then given as an input to a natural language processing model like ‘bag of words” model or a “skip gram” model. The most well-known work on knowledge graph embedding models used particularly for document retrieval and ranking can also be found in [13][11][18][19][25].

3 The Hybrid Query-resolution Approach

This section gives an overview of the proposed query-resolution approach to literature knowledge graphs using a hybrid representation that combines a “traditional” embedding and a knowledge graph embedding. A schematic outlining the proposed approach is presented in Figure 1. The input (top of the Figure) is a query Q and a document collection $\mathbf{D} = \{D_1, D_2, \dots D_i\}$. The whole document collection D is referenced by a knowledge graph. Each document $D_i \in \mathbf{D}$ consists of n terms such that $D_i = \{d_1, d_2, \dots d_n\}$. From Figure 1 it can be seen that the proposed approach has four main stages.

Stage I: Pre-processing
Stage II: Generation of Word embeddings.
Stage III: Knowledge graph embedding and word embedding concatenation.
Stage IV: Measuring similarity between query embedding and document embeddings, and document ranking.

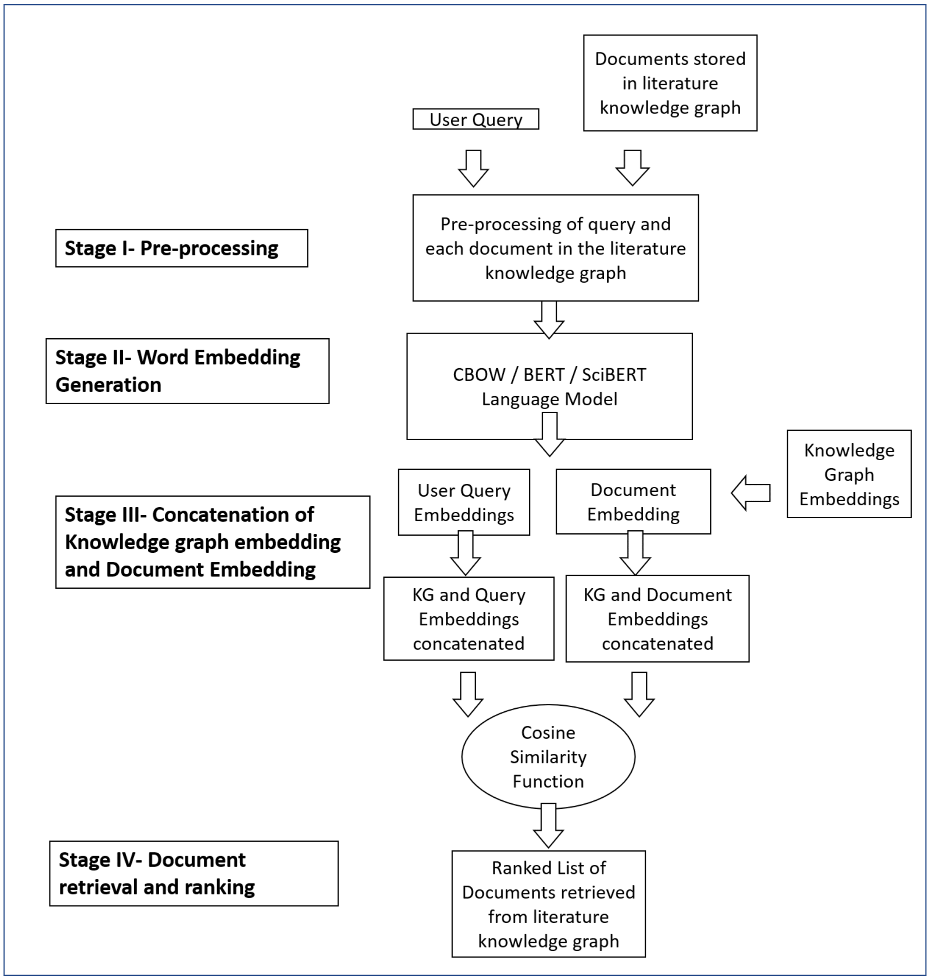

Fig. 1. Schematic of the adopted literature knowledge graph query resolution process.

During the first stage, Stage I, the input query Q and document collection $\mathbf{D}$, are pre-processed. The nature of the pre-processing depends on the nature on the language model used. For the evaluation of the proposed approach, and as noted earlier and indicated in the figure, three word embedding models were considered: CBOW, BERT and Sci-BERT. The pre-processing for the CBOW model entails punctuation and stop-word removal to give Q' and $\mathbf{D}'$. The Python Natural language toolkit[2] was used for stop word removal. The pre-processing for BERT was conducted by the BERT default tokenizer which does all the required pre-processing of the query Q and document collection D. BERT also adds special "classification" (CLS) and "separating" sentence (SEP) tokens to the start and end of each sentence during pre-processing. The pre-processing when using Sci-

[2] https://www.nltk.org/

BERT embedding is similar to when using BERT embedding. The result of the pre-processing, regardless of which embedding model was adopted, is a cleaned version of Q and $\mathbf{D}$, Q' and $\mathbf{D}' = \{D'_1, D'_2, \dots\}$.

During the second stage, Stage II, of the proposed query-resolution approach, as shown in Figure 1, a selected language model is used to generate word embeddings for query Q' and each document $D'_i \in \mathbf{D}'$. Recall that a word embedding is expressed as a numeric vector of a constant length.

The third stage, Stage III, of the proposed query-resolution mechanism, takes the word embedding generated from the second stage, and concatenates the generated word embedding with a random walk knowledge graph embedding. The intuition here was that random walk knowledge graph embeddings when combined with a "traditional" embedding, would provide additional information leading to a better query resolution performance than would otherwise be attained as suggested in [13, 23]. In Figure 1 the "traditional" embedding is referred to as the "left hand embedding" and the knowledge graph embedding as the "right hand embedding". The left-hand and right-hand embeddings are concatenated to produce a new hybrid document embedding for the query and each document in the literature knowledge graph. Further detail regarding the generation of random walk literature knowledge graph embeddings is provided in subsection 4 below.

The fourth stage of the proposed query-resolution mechanism takes the document embeddings and query embedding from the third stage and determines the similarity scores. For the evaluation presented later in this paper, and as indicated in Figure 1, cosine similarity was used. Cosine similarity, is the cosine of the angle between two vectors x and y calculated using Equation 1. Similarity scores are generated for each document in the literature knowledge graph which are then used to create a ranked list of documents from which the top k can be selected. Experiments were conducted using $k = 5$ and $k = 10$ with consultation from domain experts for the ORRCA database.

$$S_{cos}(x, y) = \frac{x.y}{||x|| \times ||y||} \tag{1}$$

4 Random Walk Knowledge Graph Embedding

The random walk knowledge graph (right-hand) embedding was generated using a random walk technique applied over a knowledge graph G. The basic idea of random walk generation was presented in [17][5]. The advantage of concatenating an embedding generated from random walk knowledge graph to a general word embedding (such as CBOW, BERT, or Sci-BERT) is that the graph embedding will capture the knowledge held in the literature graph which, it was conjectured, would provide for a better word embedding. The proposed random algorithm used a set of random walks (paths) over G, such that $\mathbf{R} = \{R_1, R_2, ...\}$. Each random walk $R_i \in \mathbf{R}$ is of the form $[v_1, v_2, \dots, v_{rw}]$, where v_j is a concept vertex in G and rw is the length of the walk. Note that no two values for v_j are the same. Each $R_i \in \mathbf{R}$ thus comprises a sequence of vertices representing concepts in a knowledge graph. Each random walk across G can be referred to as a "sentence". This means that various kinds of NLP models, such as a "bag of words" model or a "skip-gram" model [7] can be applied to the generated sentences from such random walks. For the evaluation presented later in this paper, the Node2vec Framework was used to simulate random walks over G and for the generation of random walk embeddings. A value of $rw = 3$ was used for the experiments in this paper because similar values have been used in the literature in the context of the literature knowledge graph generated from ORRCA [15].

5 Evaluation

This section reports on the evaluation of the proposed hybrid query-resolution mechanism. The objectives of the reported evaluation were:

- To compare the operation of CBOW, BERT and Sci-BERT embeddings when combined with random walk knowledge embeddings and when used in isolation.
- To identify an appropriate setting for k, the number of documents returned (rank threshold). Experiments were conducted using $k = 5$ and $k = 10$.

For the random walk generation the number of random walks generated was set to 100 as such a value has been used in the literature as well [11]. This was because it represented an appropriate trade off between the execution time required to generate the knowledge graph random walk embeddings and coverage. Note that considerable computational resource is required to generate random walks. The ORRCA query-document [8] data set was used, which comprised 45 search queries.

The evaluation metrics used were Mean Average Precision (MAP) at k, for $k = 5$ and $k = 10$, calculated as shown in Equation 2. This metric was used because the data set did not have a ground truth ranking, hence metrics like Normalized Cumulative Gain (NDCG) could not be used. In Equation 2: (i) k is the rank threshold, (ii) Q is an evaluation query data set and (iii) ap_{jk} is Average Precision at rank k for query $j \in Q$ calculated as shown in Equation 4. In Equation 4: (i) p_{ji} is the ranked precision for query j at rank i. (i) p_{ji} is defined as the ranked precision for query j at rank i. (ii) m is equal to the number of relevant documents retrieved. Ranked precision is defined as the fraction of relevant documents for a query q_j retrieved from the total number of documents retrieved at (up to) rank i. Ranked precision is calculated as shown in Equation 3, where: (i) tp_{ji} is the number of "true positives" at rank i (the number of documents that should have been retrieved in response to a query j, and were retrieved up to rank i), and (ii) fp_{ji} is the number of "false positives" at rank i (the number of documents that should not have been retrieved in response to a query q_j, but were retrieved up to rank i).

$$MAP(k) = \frac{1}{|Q|} \sum_{j=1}^{j=|Q|} ap_{jk} \quad (2) \qquad p_{ji} = \frac{tp_{ji}}{tp_{ji} + fp_{ji}} = \left(\frac{relevant}{retrieved}\right) \quad (3)$$

$$ap_{j,k} = \frac{1}{m} \sum_{i=1}^{i=k} p_{ji} \quad (4)$$

The MAP results obtained are presented in Table 1; the best results are highlighted in bold font. From the Table, it can be seen that the hybrid CBOW and random walk knowledge graph embedding produced the best results. The suggested reason for this was that the CBOW model vocabulary was best suited to the ORRCA application domain. The results obtained when using CBOW, BERT and SciBERT in isolation seems to support this suggestion. The experiments where each of the embedding models were used in isolation also demonstrated that the knowledge graph random walk embedding performed well; thus supporting the conjecture that knowledge graph random walk embedding provides beneficial additional knowledge, which in turn increases the effectiveness of the proposed query resolution approach.

Table 1. $MAP@k$ Table for BERT, SciBERT and CBOW when combined with Random Walk embeddings, and when used in isolation

Embedding Model	MAP@5	MAP@10
CBOW and KG embeddings	**0.486**	**0.313**
BERT and KG embedding	0.420	0.256
SciBERT and KG embedding	0.414	0.252
SciBERT only embedding	0.393	0.186
BERT only embedding	0.409	0.256
CBOW only embedding	0.433	0.259
Random Walk KG only embedding	**0.458**	**0.271**

Inspection of Table 1 also indicates that better results were obtained using $k = 10$ than $k = 5$ in that better results were returned. Tables 2 to 4 present the $AP@k$ results obtained using: CBOW and random walk embeddings and CBOW used in isolation, BERT and random walk embeddings and BERT used in isolation, SciBERT and random walk embeddings and SciBERT used in isolation. The tables present the $AP@k$ results for each of the 45 queries in the ORRCA query-document data set. The search queries that perform the best are highlighted in bold font. Inspection of the Tables indicates that Queries 31, 32, 33, and 34 gave the best results from all the search queries. It was conjectured that this was a function of the query size; these queries featured more keywords than other queries. The number of keywords in a search query affects the precision. Search queries with a greater number of keywords tend to achieve better results compared to search queries with fewer keywords.

6 Conclusion

This paper proposed a query resolution mechanism for queries directed at Curated Document Databases (CDDs) stored as a literature knowledge graph. A

Table 2. *AP@k* results for combined CBOW and random walk embeddings, in comparison with CBOW used in isolation

Search Code	CBOW + Random Walk		CBOW only	
	P@5	P@10	P@5	P@10
Search1	0.4	0.4	0.0	0.3
Search2	0.4	0.3	0.4	0.3
Search3	0.2	0.2	0.6	0.5
Search4	0.6	0.6	0.6	0.6
Search5	0.4	0.2	0.0	0.0
Search6	0.2	0.4	0.8	0.7
Search7	0.8	0.9	0.6	0.6
Search8	0.0	0.0	0.0	0.0
Search9	0.4	0.4	0.4	0.5
Search10	0.6	0.6	0.6	0.6
Search11	1.0	0.9	1.0	0.9
Search12	0.4	0.7	0.6	0.7
Search13	0.4	0.0	0.6	0.0
Search14	0.8	0.7	1.0	0.7
Search15	0.4	0.5	0.2	0.3
Search16	0.6	0.4	0.4	0.4
Search17	0.0	0.0	0.0	0.0
Search18	0.6	0.5	0.4	0.4
Search19	1.0	1.0	1.0	1.0
Search20	0.4	0.2	0.2	0.2
Search21	0.8	0.5	0.8	0.5
Search22	0.6	0.5	0.2	0.3
Search23	0.6	0.8	0.4	0.5
Search24	1.0	0.9	0.0	0.0
Search25	0.0	0.0	0.0	0.0
Search26	0.0	0.0	0.0	0.0
Search27	0.8	0.8	1.0	0.8
Search28	0.6	0.7	0.8	0.7
Search29	0.8	0.0	0.8	0.0
Search30	1.0	0.0	1.0	0.0
Search31	**1.0**	**1.0**	**1.0**	**1.0**
Search32	**1.0**	**0.9**	**1.0**	**0.9**
Search33	**1.0**	**1.0**	**0.8**	**0.9**
Search34	**1.0**	**1.0**	**1.0**	**0.9**
Search35	0.0	0.0	0.0	0.0
Search36	0.4	0.2	0.6	0.3
Search37	0.4	0.0	0.4	0.0
Search38	0.4	0.2	0	0.4
Search39	0.0	0.0	0.2	0.1
Search40	0.4	0.4	0.2	0.3
Search41	0.6	0.4	0.4	0.3
Search42	0.0	0.0	0.0	0.0
Search43	0.2	0.1	0.2	0.1
Search44	0.6	0.0	0.8	0.0
Search45	0.0	0.0	0.0	0.2

Table 3. *AP@k* results for combined BERT and random walk embeddings, in comparison with BERT used in isolation

Search Code	BERT + Random Walk		BERT only	
	P@5	P@10	P@5	P@10
Search1	1.0	1.0	0.0	0.0
Search2	0.2	0.3	0.0	0.2
Search3	0.6	0.5	0.6	0.4
Search4	0.4	0.6	0.6	0.6
Search5	0.0	0.0	0.0	0.0
Search6	0.2	0.4	0.2	0.5
Search7	0.4	0.7	0.0	0.4
Search8	0.0	0.0	0.0	0.0.1
Search9	0.0	0.2	0.2	0.3
Search10	0.6	0.8	0.8	0.8
Search11	1.0	0.9	1.0	0.9
Search12	0.6	0.6	0.8	0.7
Search13	0.6	0.0	0.6	0.0
Search14	0.8	0.8	0.8	0.8
Search15	0.6	0.4	0.4	0.4
Search16	0.4	0.4	0.6	0.5
Search17	0.0	0.0	0.0	0.0
Search18	0.4	0.4	0.2	0.3
Search19	1.0	0.9	1.0	0.9
Search20	0.2	0.3	0.2	0.2
Search21	0.4	0.5	0.6	0.5
Search22	0.6	0.6	0.4	0.4
Search23	0.2	0.5	0.6	0.7
Search24	0.4	0.7	0.6	0.7
Search25	0.0	0.0	0.0	0.0
Search26	0.0	0.0	0.0	0.0
Search27	0.6	0.7	0.6	0.7
Search28	0.6	0.8	0.8	0.7
Search29	1.0	0.0	0.6	0.0
Search30	1.0	0.0	1.0	0.0
Search31	**1.0**	**1.0**	**1.0**	**1.0**
Search32	**1.0**	**0.9**	**1.0**	**0.9**
Search33	**1.0**	**0.9**	**1.0**	**1.0**
Search34	**1.0**	**0.9**	**1.0**	**0.9**
Search35	0.0	0.0	0.0	0.0
Search36	0.4	0.2	0.0	0.0
Search37	0.4	0.0	0.4	0.0
Search38	0.2	0.1	0.2	0.1
Search39	0.2	0.1	0.2	0.2
Search40	0.0	0.1	0.2	0.3
Search41	0.2	0.2	0.2	0.2
Search42	0.0	0.0	0.0	0.0
Search43	0.2	0.1	0.0	0.0
Search44	0.6	0.0	0.6	0.0
Search45	0.0	0.1	0.0	0.1

Table 4. $AP@k$ results for combined SciBERT and random walk embeddings, in comparison with Sci-BERT used in isolation

Search Code	Sci-BERT + Random Walk		Sc-BERT only	
	P@5	P@10	P@5	P@10
Search1	0.0	0.3	0.0	0.3
Search2	0.2	0.2	0.0	0.3
Search3	0.4	0.3	0.2	0.5
Search4	0.6	0.6	0.6	0.6
Search5	0.0	0.1	0.0	0.1
Search6	0.2	0.1	0.2	0.3
Search7	0.4	0.6	0.0	0.5
Search8	0.0	0.0	0.0	0.0
Search9	0.6	0.4	0.6	0.4
Search10	0.8	0.7	1.0	0.8
Search11	1.0	0.9	1.0	1.0
Search12	0.2	0.3	0.8	0.7
Search13	0.6	0.0	0.4	0.0
Search14	0.4	0.5	0.8	0.8
Search15	0.4	0.2	0.6	0.4
Search16	0.0	0.1	0.2	0.4
Search17	0.0	0.0	0.0	0.0
Search18	0.4	0.4	0.4	0.4
Search19	1.0	1.0	1.0	0.9
Search20	0.0	0.0	0.0	0.2
Search21	0.4	0.5	0.4	0.5
Search22	0.8	0.6	0.4	0.2
Search23	0.4	0.6	0.6	0.5
Search24	0.6	0.6	0.6	0.7
Search25	0.2	0.1	0.0	0.1
Search26	0.0	0.0	0.0	0.0
Search27	0.4	0.5	1.0	0.0
Search28	0.6	0.7	0.8	0.7
Search29	0.6	0.0	0.8	0.0
Search30	1.0	0.0	0.0	0.0
Search31	**1.0**	**1.0**	**1.0**	**1.0**
Search32	**1.0**	**0.9**	**1.0**	**0.9**
Search33	**1.0**	**0.9**	**1.0**	**0.9**
Search34	1.0	0.9	1.0	0.0
Search35	0.0	0.0	0.0	0.0
Search36	0.0	0.0	0.0	0.1
Search37	0.4	0.0	0.2	0.0
Search38	0.2	0.4	0.0	0.1
Search39	0.0	0.1	0.0	0.0
Search40	0.2	0.2	0.4	0.2
Search41	0.4	0.2	0.4	0.0
Search42	0.0	0.0	0.0	0.0
Search43	0.2	0.1	0.2	0.1
Search44	0.6	0.0	0.5	0.0
Search45	0.0	0.0	0.0	0.2

hybrid document embedding was proposed that combined a "traditional" embedding with a knowledge graph embedding for queries and documents. Three kinds of word "traditional" embedding were considered: CBOW, BERT and SciBERT. The evaluation indicated that the proposed hybrid embedding resulted in better $MAP@k$ results than when the various embeddings were used in isolation. A best $MAP@k$ value of 0.486 was obtained when using a combination of CBOW and the proposed random walk knowledge graph embedding.

References

1. Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. Construction of the literature graph in semantic scholar. *NAACL HLT 2018*, pages 84–91, 2018.
2. Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, 2019.
3. Chuming Chen, Karen E Ross, Sachin Gavali, Julie E Cowart, and Cathy H Wu. Covid-19 knowledge graph from semantic integration of biomedical literature and databases. *Bioinformatics*, 37(23):4597–4598, 2021.
4. Jens Dörpinghaus, Andreas Stefan, Bruce Schultz, and Marc Jacobs. Context mining and graph queries on giant biomedical knowledge graphs. *Knowledge and Information Systems*, 64(5):1239–1262, 2022.
5. A Grover and J Leskovec. Node2vec: scalable feature learning for networks. kdd 2016: 855–864, 2016.
6. Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 55–64, 2016.
7. Derry Jatnika, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. Word2vec model analysis for semantic similarities in english words. *Procedia Computer Science*, 157:160–167, 2019.
8. Anna Kearney, Nicola L Harman, Anna Rosala-Hallas, Claire Beecher, Jane M Blazeby, Peter Bower, Mike Clarke, William Cragg, Sinead Duane, Heidi Gardner, et al. Development of an online resource for recruitment research in clinical trials to organise and map current literature. *Clinical Trials*, 15(6):533–542, 2018.
9. Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pretraining of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.
10. Md Kowsher, Md Shohanur Islam Sobuj, Md Fahim Shahriar, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. An enhanced neural word embedding model for transfer learning. *Applied Sciences*, 12(6):2848, 2022.
11. Xiaomin Liang, Daifeng Li, Min Song, Andrew Madden, Ying Ding, and Yi Bu. Predicting biomedical relationships using the knowledge and graph embedding cascade model. *PLoS One*, 14(6):e0218264, 2019.
12. SK Liji and P Muhamed Ilyas. Semantic malayalam dialogue system for covid-19 question answering using word embedding and cosine similarity. In *2021 International Conference on Advances in Computing and Communications (ICACC)*, pages 1–6. IEEE, 2021.

13. Zheng-Hao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *ACL (1)*, 2018.
14. Gengchen Mai, Bo Yan, Krzysztof Janowicz, and Rui Zhu. Relaxing unanswerable geographic questions using a spatially explicit knowledge graph embedding model. In *International Conference on Geographic Information Science*, pages 21–39. Springer, 2019.
15. Iqra Muhammad, Danushka Bollegala, Frans Coenen, Carrol Gamble, Anna Kearney, and Paula Williamson. Document ranking for curated document databases using bert and knowledge graph embeddings: Introducing grab-rank. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 116–127. Springer, 2021.
16. Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.
17. Shengtian Sang, Zhihao Yang, Xiaoxia Liu, Lei Wang, Hongfei Lin, Jian Wang, and Michel Dumontier. Gredel: A knowledge graph embedding based method for drug discovery from biomedical literatures. *IEEE Access*, 7:8404–8415, 2018.
18. Sonam Sharma. Fact-finding knowledge-aware search engine. In *Data Management, Analytics and Innovation*, pages 225–235. Springer, 2022.
19. Longxiang Shi, Shijian Li, Xiaoran Yang, Jiaheng Qi, Gang Pan, and Binbin Zhou. Semantic health knowledge graph: semantic integration of heterogeneous medical knowledge and services. *BioMed research international*, 2017, 2017.
20. Alfredo Silva and Marcelo Mendoza. Improving query expansion strategies with word embeddings. In *Proceedings of the ACM Symposium on Document Engineering 2020*, pages 1–4, 2020.
21. Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.
22. Qingyun Wang, Manling Li, Xuan Wang, Nikolaus Parulian, Guangxing Han, Jiawei Ma, Jingxuan Tu, Ying Lin, Haoran Zhang, Weili Liu, et al. Covid-19 literature knowledge graph construction and drug repurposing report generation. *arXiv preprint arXiv:2007.00576*, 2020.
23. Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.
24. Shirui Wang, Wenan Zhou, and Chao Jiang. A survey of word embeddings based on deep learning. *Computing*, 102(3):717–740, 2020.
25. Colby Wise, Miguel Romero Calvo, Pariminder Bhatia, Vassilis Ioannidis, George Karypus, George Price, Xiang Song, Ryan Brand, and Ninad Kulkani. Covid-19 knowledge graph: Accelerating information retrieval and discovery for scientific literature. In *Proceedings of Knowledgeable NLP: the First Workshop on Integrating Structured Knowledge and Neural Networks for NLP*, pages 1–10, 2020.
26. Tong Wu, Yunlong Wang, Yue Wang, Emily Zhao, Yilian Yuan, and Zhi Yang. Representation learning of ehr data via graph-based medical entity embedding. *arXiv preprint arXiv:1910.02574*, 2019.
27. Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*, 2019.
28. Tong Yu, Jinghua Li, Qi Yu, Ye Tian, Xiaofeng Shun, Lili Xu, Ling Zhu, and Hongjie Gao. Knowledge graph for tcm health preservation: Design, construction, and applications. *Artificial intelligence in medicine*, 77:48–52, 2017.

29. Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian document computing symposium*, pages 1–8, 2015.