



Using Knowledge Graphs to enhance the utility of Curated Document Databases

Thesis submitted in accordance with the requirements of the University of Liverpool for
the degree of Doctor in Philosophy by

Iqra Muhammad

November 2022

Abstract

The research presented in this thesis is directed at the generation, maintenance and querying of Curated Document Databases (CDDs) stored as literature knowledge graphs. Literature knowledge graphs are graphs where the vertices represent documents and concepts; and the edges provided links between concepts, and concepts and documents. The central motivation for the work was to provide CDD administrators with a useful mechanism for creating and maintaining literature knowledge graph represented CDDs, and for end users to utilise them. The central research question is “What are some appropriate techniques that can be used for generating, maintaining and utilizing literature knowledge graphs to support the concept of CDDs?”. The thesis thus addresses three issues associated with literature knowledge graphs: (i) their construction, (ii) their maintenance so that their utility can be continued, and (iii) the querying of such knowledge graphs. With respect to the first issue, the Open Information Extraction for Knowledge Graph Construction (OIE4KGC) approach is proposed founded on the idea of using open information extraction. Two open information extraction tools were compared, the RnnOIE tool and the Leolani tool. The RnnOIE tool was found to be effective for generation of triples from clinical trial documents. With respect to the second issue two approaches are proposed for maintaining knowledge graph represented CDDs; the CN approach and the Knowledge Graph And BERT Ranking (GRAB-Rank) approach. The first proposed approach used a feature vector representation; and the second a unique hybrid domain specific document embedding. The hybrid domain-specific document embedding combines a Bidirectional Encoder Representations from Transformers embedding with a knowledge graph embedding. This proposed embedding was used for document representation in a LETOR model. The idea was to rank a set of potential documents. The Grab-Rank embedding based LETOR approach was found to be effective. For the third identified issue the standard solution is to represent both the query to be addressed and the documents in the knowledge graph in a manner that will allow the documents to be ranked with respect to the query. The solution proposed for this was to utilize a hybrid embedding for query resolution. Two forms of embedding are utilized for query resolution: (i) a Continuous Bag-Of-Words embedding was combined with graph embedding and (ii) for the second BERT and Sci-BERT embedding were combined with graph embedding. The evaluation indicates that the CBOW embedding combined with graph embedding was found to be effective.

Acknowledgements

I would like to thank my first supervisor Prof. Frans Coenen for his support, and co-operation throughout my PhD journey. I really appreciate his positive energy especially during the difficult days. I feel very honoured to have this experience with him. My sincere thanks are also to my secondary supervisors, Dr. Paula Williamson and Prof Danushka Bollegala for their valuable feedback and constructive assessment of my research. I am also thankful to Anna Kearney from the Bio-statistics Department, at the University of Liverpool, who provided unlimited support concerning the collection of the ORRCA CDD, providing valuable recommendations for the dataset used in this research. My sincere and profound gratitude also goes to my friends who have been supporting me so that I can follow my dreams.

Contents

Abstract	i
Acknowledgements	ii
Contents	v
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 Overview	1
1.2 Research Question and Issues	3
1.3 Research Methodology	5
1.3.1 Contributions and Publications	6
1.4 Thesis Outline	8
1.5 Summary	9
2 Literature Review	10
2.1 Introduction	10
2.2 Knowledge graphs	12
2.2.1 Domain Specific Knowledge Graphs	12
2.2.2 Literature knowledge graphs	14
2.3 Literature Knowledge Graph Construction	15
2.3.1 Rule-based OIE for knowledge graph construction	16
2.3.2 Supervised OIE for Knowledge graph construction	16
2.4 Literature Knowledge Graph Maintenance	17
2.4.1 Statistical Document Representation for Ranking Models for Updating Literature Knowledge Graphs	18
2.4.2 Semantic Document Representation for Ranking Models for Updating Literature Knowledge Graphs	19

2.4.3	Knowledge Graph Embedding Representation for Ranking Models for Updating Literature Knowledge Graphs	21
2.5	Literature knowledge graph query resolution	22
2.5.1	Semantic Document Ranking Models for Literature Knowledge Graph Query Resolution	23
2.5.2	Knowledge Graph Document Ranking Models for Literature Knowledge Graph Query Resolution	23
2.5.3	Summary	24
3	Evaluation Dataset	25
3.1	Introduction	25
3.2	Background to ORRCA	26
3.3	Review of ORRCA Datasets	27
3.4	Summary	28
4	Knowledge Graph Generation	30
4.1	Introduction	30
4.2	Problem Definition	32
4.3	The Open Information Extraction For Knowledge Graph Generation (OIE4KGC) Approach	33
4.3.1	Triple Extraction (OIE4KGC Stage 1)	36
4.3.2	Triple Filtering (OIE4KGC Stage 2)	37
4.3.3	Linking of Clinical Concepts to UMLS (OIE4KGC Stage 3)	38
4.3.4	Knowledge Graph Population (OIE4KGC Stage 4)	39
4.4	Evaluation	41
4.4.1	Comparison of OIE Tools	42
4.4.2	Qualitative Analysis of Open Information Extraction Tools	43
4.5	Summary	45
5	Maintenance of literature knowledge graph using document ranking	47
5.1	Introduction	47
5.2	Problem Definition	51
5.3	The Proposed LETOR Framework	52
5.4	The CN Approach	54
5.5	The GRAB-Rank Approach	57
5.5.1	BERT Embedding	58
5.5.2	Knowledge Graph Embedding	59
5.6	Evaluation	61
5.6.1	Evaluation Data Set	62
5.6.2	Evaluation Metrics	62
5.6.3	Determination of The Most appropriate Value for sigma	63

5.6.4	Determination of The Most appropriate Value for the random walk length	66
5.6.5	Combined BERT and Knowledge Graph Embedding Versus Single Embedding	67
5.6.6	Comparative effectiveness	68
5.6.7	Time Savings Gained	68
5.7	Summary	72
6	Literature Knowledge Graph Query Resolution	73
6.1	Introduction	73
6.2	The Query Resolution Process	74
6.3	Contextual and Non-contextual Embedding Systems	79
6.3.1	Non-Contextual Embedding Systems (CBOW)	80
6.3.2	Contextual Embedding Systems (BERT and SciBERT)	81
6.4	Similarity Measurement and Ranking	83
6.5	Evaluation	84
6.5.1	Evaluation Dataset	85
6.5.2	Evaluation Dataset Collection Process	85
6.5.3	Evaluation Metrics	89
6.5.4	Results and Discussion	90
6.5.5	Empirical study for knowledge graph query resolution	100
6.6	Conclusion	105
7	Conclusion and Future Work	110
7.1	Introduction	110
7.2	Summary of Thesis	110
7.2.1	Chapter 1 key findings	110
7.2.2	Chapter 2 key findings	111
7.2.3	Chapter 3 key findings	112
7.2.4	Chapter 4 key findings	112
7.2.5	Chapter 5 key findings	113
7.2.6	Chapter 6 key findings	113
7.3	Main Findings and Contributions	114
7.4	Future Work	117
	References	121
A	Appendix 1	137

List of Figures

1.1	ORRCA papers and articles by year, 1976-2017, illustrating the rapid growth of the number of publications directed at recruitment strategies for clinical trials.	2
1.2	A toy example of a literature knowledge graph generated using OIE4KGC	4
2.1	Schematic of a simple Knowledge Graph, $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\} \in C$ and $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\} \in R$	13
2.2	A schematic diagram for a domain-specific knowledge graph	13
2.3	Schematic of a simple Literature Knowledge Graph, $\{d_1, d_2, d_3, d_4\} \in D$, $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\} \in C$ and $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\} \in R$	14
4.1	A Schematic showing the document processing stages involved in the construction of a literature knowledge graph using the OIE4KGC approach	34
4.2	A schematic diagram for a predicate-argument sequence labelling problem	36
4.3	An example of a literature knowledge graph generated using OIE4KGC	39
5.1	The Proposed LETOR Framework	53
5.2	The Proposed CN Approach	55
5.3	The GRAB-Rank LETOR process	58
5.4	Example graph for explaining the concept of random walk	60
5.5	Some example random walks generated from the graph given in Figure 5.4	60
5.6	Precision-recall curve for ORRCA 2015 dataset using CN approach, decision thresholds σ given on the x-axes, and precision and recall on the y-axes	64
5.7	Precision-recall curve for ORRCA 2017 dataset using CN approach, decision thresholds σ given on the x-axes, and precision and recall on the y-axes	64
5.8	Precision-recall curve for ORRCA 2015 dataset using GRAB-Rank approach, decision thresholds σ given on the x-axes, and precision and recall on the y-axes	65
5.9	Precision-recall curve for ORRCA 2017 dataset using GRAB-Rank approach, decision thresholds σ given on the x-axes, and precision and recall on the y-axes	65

5.10	The 2015 update effort-recall curve using CN algorithm	69
5.11	The 2017 update effort-recall curve using CN algorithm	70
5.12	The 2017 update effort-recall curve using GRAB-Rank algorithm	71
5.13	The 2015 update effort-recall curve using GRAB-Rank algorithm	72
6.1	Schematic of the adopted literature knowledge graph query resolution process.	76
6.2	A schematic diagram for CBOW model word embedding generation	80
6.3	A screenshot of the Online Resource for Research in Clinical trials (ORRCA) main search interface	87
6.4	A screenshot of the Online Resource for Research in Clinical trials (ORRCA) advance search interface	88
6.5	A bar chart showing the recorded $MAP@5$ values for the document embedding techniques considered	96
6.6	A bar chart showing the recorded $MAP@10$ values for the document embedding techniques considered	97
6.7	Bar graph showing number of queries against $AP@5$ values when using CBOW+RandomWalk document embeddings	98
6.8	Bar graph showing number of queries against $AP@10$ values when using CBOW+RandomWalk document embeddings	99
6.9	Bar graph showing number of queries against $AP@5$ values when using CBOW only document embeddings	100
6.10	Bar graph showing number of queries against $AP@10$ values when using CBOW only document embeddings	101
6.11	Bar graph showing number of queries against $AP@5$ values when using BERT+RandomWalk document embeddings	102
6.12	Bar graph showing number of queries against $AP@10$ values when using BERT+RandomWalk document embeddings	103
6.13	Bar graph showing number of queries against $AP@5$ values when using BERT only document embeddings	104
6.14	Bar graph showing number of queries against $AP@10$ values when using BERT only document embeddings	105
6.15	Bar graph showing number of queries against $AP@5$ values when using SciBERT+Random Walk document embeddings	106
6.16	Bar graph showing number of queries against $AP@10$ values when using SciBERT+Random Walk document embeddings	107
6.17	Bar graph showing number of queries against $AP@5$ values when using SciBERT only document embeddings	107
6.18	Bar graph showing number of queries against $AP@10$ values when using SciBERT only document embeddings	108
6.19	Bar graph showing number of queries against $AP@5$ values when using Random Walk document only embeddings	108

6.20	Bar graph showing number of queries against $AP@10$ values when using Random Walk only document embeddings	109
6.21	A bar graph representing the relevance scores from the KG query resolution empirical study on x-axis and number of irrelevant/relevant documents on y-axis	109

List of Tables

3.1	Statistical overview of the ORRCA evaluation data sets	28
4.1	Symbol table for Chapter 4	32
4.2	Performance of RnnOIE using the ORRCA and ReVerb dataset	44
4.3	Performance of Leolani using the ORRCA and ReVerb datasets	44
4.4	Example triples extracted using the RnnOIE triple extraction tool applied to sentences in the ORRCA dataset	45
4.5	Example triples extracted using the Leolani triple extraction tool applied to sentences in the ORRCA dataset	46
5.1	Symbol table for Chapter 5	51
5.2	The performance of GRAB-Rank using a range of values for rw , the random walk length (best results in bold font).	67
5.3	Comparison, in terms of precision and recall, using the GRAB-Rank approach, and using BERT and knowledge graph embeddings in isolation (best results in bold font).	68
5.4	The performance of GRAB-Rank and CN approaches in comparison with the Okapi BM25 approach (best results in bold font).	69
6.1	Symbol table for Chapter 6, Utilisation of Literature Knowledge Graphs using Query Based Document Ranking	75
6.2	Fragment of ORRCA Query-document Evaluation Data set	86
6.3	$AP@k$ results for combined CBOW and random walk embeddings, in comparison with CBOW used in isolation	91
6.4	$AP@k$ results for combined BERT and random walk embeddings, in comparison with BERT used in isolation	92
6.5	$AP@k$ results for combined SciBERT and random walk embeddings, in comparison with Sci-BERT used in isolation	93
6.6	$AP@k$ results for Random Walk embeddings used in isolation	94
6.7	$MAP@k$ Table for BERT, SciBERT and CBOW when combined with Random Walk embeddings, and when in isolation	95

6.8	Results from empirical study	103
A.1	Table showing the search queries in the ORRCA query-document dataset mentioned in Chapter 6.	137
A.2	Table showing documents returned by the proposed system as part of the Empirical Study for the query “bereaved” and their relevance label as discussed in Chapter 6	141
A.3	Table showing documents returned by the proposed system in Empirical Study for the query “palliative” and their relevance label as discussed in Chapter 6	142
A.4	Table showing documents returned by the proposed system in Empirical Study for the query “facebook” and their relevance label as discussed in Chapter 6	143
A.5	Table showing documents returned by the proposed system in Empirical Study for the query “obesity” and their relevance label as discussed in Chapter 6	144

Chapter 1

Introduction

1.1 Overview

The volume of scientific literature is increasing at a rapid rate on a year-on-year basis [56, 81]. This rate of growth has led to an increase in the cumulative domain knowledge that researchers need to access. This has created a challenge for the scientific community in terms of the resources required to manually search and analyse relevant documents in the literature. One solution is the usage of Curated Document Databases (CDDs), specialised document collections that bring together published work, in a defined domain, into a single scientific literature repository. One example of such a CDD, and that used both as a focus and for illustrative purposes throughout this thesis, is the Online Resource for Recruitment research in Clinical trials (ORRCA) CDD [58]. The ORRCA CDD¹ brings together abstracts of papers concerned with the highly specialised domain of recruitment strategies for clinical trials, and serves to limit the documents that clinical trials researchers need to access. However, the use of CDDs only offers a partial solution as the volume of available literature continues to grow. For instance, at time of writing, the ORRCA CDD had a year-on-year increase in publications as shown in Figure 1.1.

Another way of minimising the difficulty researchers face when reviewing the volume of scientific literature that is available is by harnessing advances in Artificial Intelligence (AI) and Machine Learning (ML) [30]. The tools and techniques of AI and ML have made significant advances over recent decades. This has led to the automation of many tasks that once required significant human resource. One example, and one of particular relevance

¹<https://www.orrca.org.uk/>

to this thesis, is the automation of search methods for scientific literature [106, 44, 129]. Various research studies have been published since 2006, referencing the need for screening automation when reviewing scientific literature and the benefits that can be realised [92]. However, with respect to CDDs these advantages have yet to make significant “in roads”. This is largely because the concept of CDDs was initially seen as presenting a solution to the information overload challenge associated with the volume of scientific literature available. However, as demonstrated in Figure 1.1, the concept of CDDs has only offered a “stop-gap” solution. As the volume of documents held in CDDs continues to increase there is a corresponding need for the application of AI and ML-based automated search methods; this challenge was highlighted in [42] in the context of the ORRCA CDD.

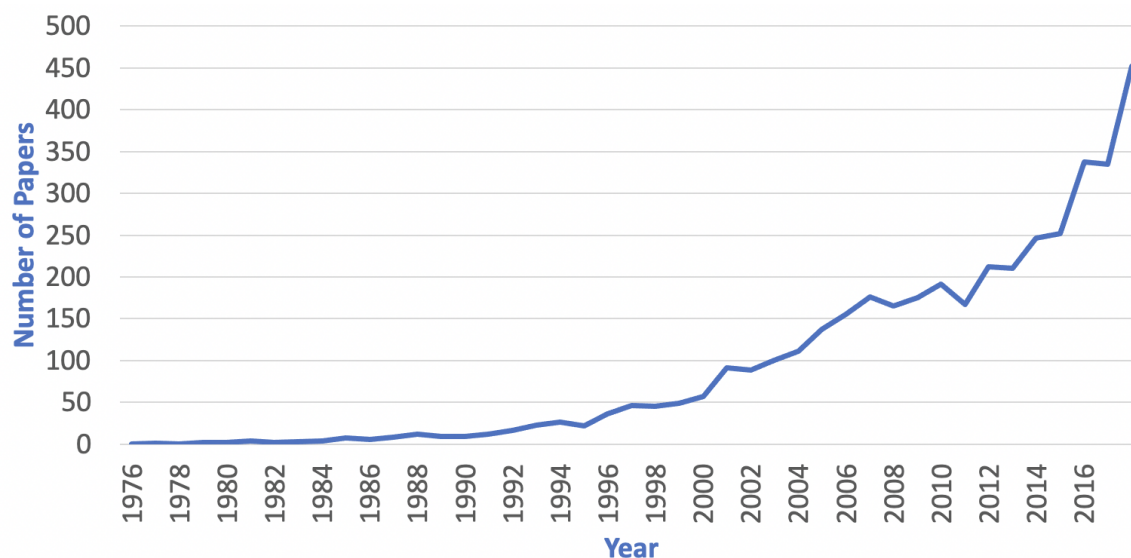


Figure 1.1: ORRCA papers and articles by year, 1976-2017, illustrating the rapid growth of the number of publications directed at recruitment strategies for clinical trials.

In the context of CDDs the screening of scientific literature is not just about finding appropriate documents within the CDD, it is also concerned with the creation of CDDs and their maintenance. The manual process for creating CDDs is frequently referred to as the “systematic review process” [91, 58]. Systematic reviews are carried out by those responsible for the provision of CDDs so as to update and maintain such databases. The systematic review process is resource intensive. It is estimated that carrying out a single systematic review can take from several months to a whole year. Automation of the process,

using the tools and techniques of AI and ML, therefore has clear benefits. Various types of AI and ML technique have been reported on for analysing and extracting information from scientific documents. A state-of-the-art AI technique for representing knowledge is the idea of knowledge graphs. A knowledge graph is a graph where the vertices represent entities of some kind, the knowledge we are interested in representing, and the edges the relationships between these entities. As such they are frequently used with respect to linked open data in the context of the “semantic web”. However, they are also used by internet search engines, such as Google and Yahoo, and by question-answering services such as those provided by Apple’s Siri and Amazon’s Alexa. In addition they play a role with respect to the facilities provided by social network platforms, such as LinkedIn and Facebook. Where a knowledge graph is used to represent documents, the term literature knowledge graph (or sometimes document knowledge graph) is used. In a literature knowledge graph the entities are documents and/or concepts referenced in documents, and the edges the connections between them. A toy example literature knowledge graph is shown in Figure 1.2. In the figure the blue vertices indicate concepts and the yellow vertices documents. There are two kinds of edge in the figure:

1. Edges linking Concepts (blue colour)
2. Edges linking Concepts and Documents (red colour)

The advantage offered by literature knowledge graphs is that their usage speeds up the process of query resolution and consequently information retrieval [129, 18]. Literature knowledge graphs have clear potential in the context of CDDs. This observation is the central motivation underpinning the work presented in this thesis.

1.2 Research Question and Issues

Given the foregoing motivation the fundamental objective of the work presented in this thesis is to investigate how best the benefits offered by the idea of literature knowledge graphs can be realised in the context of CDDs. The central research question that this thesis seeks to address is thus formulated as follows:

1. What are some suitable techniques that can be used for generating, maintaining and utilizing literature knowledge graphs to support the concept of CDDs?

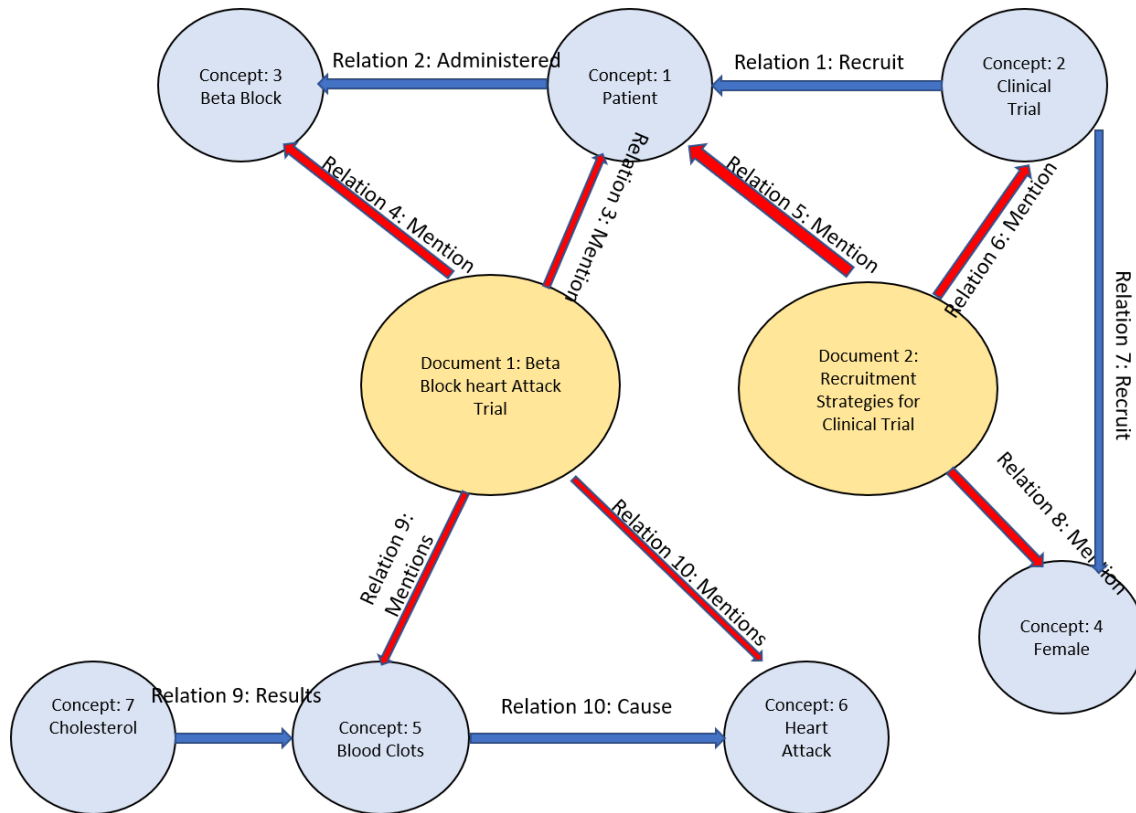


Figure 1.2: A toy example of a literature knowledge graph generated using OIE4KGC

The resolution of this overriding research question involves the resolution of a number of Subsidiary Research Questions (SRQs):

1. Given a collection of documents within a CDD, represented using traditional relational database technology, how can these best be processed so that they form a literature knowledge graph.
2. Given an existing CDD, represented as a literature knowledge graph, how can this knowledge graph best be maintained to ensure that it is up to date.
3. Given an existing CDD, represented as a literature knowledge graph, how can this knowledge graph best be queried so as to retrieve relevant documents.
4. Assuming that the maintenance and querying of literature knowledge graphs will entail some kind of document ranking what is a suitable mechanism for deriving a ranked list of documents and what would this mechanism entail?

5. In the context of document ranking can the concepts within a literature knowledge graph be utilized to improve a document ranking mechanism and how would this operate?
6. Given the foregoing SRQ, can the embeddings implicit within a literature knowledge graph be used to provide an answer to a query in the context of document retrieval?

The significance of SRQ2 is that a CDD, however it is represented, is only as good as its last update; it needs to be maintained. SRQ4 to SQR6 are all related to SRQ2 and SRQ3 in that any solution to literature knowledge graph maintenance and document retrieval is likely to entail document ranking.

1.3 Research Methodology

The research methodology that was adopted to provide answers to the above SRQs, and consequently the overriding research question, comprised three Phases:

1. Literature Knowledge Graph Construction
2. Literature Knowledge Graph Maintenance
3. Literature Knowledge Graph Querying

The start point for Phase 1, was to hand generate a literature knowledge graph using ten records from the ORRCA CDD. This was a proof of concept exercise to demonstrate the viability of using literature knowledge graphs in the context of CDDs. The next step was to develop an automated literature knowledge graph construction mechanism. The central idea was to use the concept of Open Information Extraction (OIE), the established RnnOIE (Recurrent Neural Network OIE) tool was adopted for this purpose. The result was the Open Information Extraction for Knowledge Graph Construction (OIE4KGC system). This was evaluated using the bespoke ORRCA data sets and the ClauseIE ReVerb benchmark dataset.

For Phase 2 a number of techniques were considered, with a focus on two. The first considered the use of a n-gram support vector regression based learning to rank model to update curated document collections. The second considered two kinds of embedding:

1. BERT word embeddings

2. Knowledge graph concept embedding

The result was the Knowledge Graph And BERT Ranking (GRAB-Rank) approach to the updating of CDDs. In both cases evaluation was conducted using the pre-2015 ORRCA data set, and tested using the 2015 and 2017 update data sets. Good results were obtained. The update dataset had been labelled in a binary manner with positive labels indicating the documents to be included in the CDD and negative labels indicating the documents not to be included in the CDD. The metrics used to measure effectiveness were precision and recall, calculated as given in Equations 1.1 and 1.2 where:

1. TP is the number of true positives
2. FP is the number of false positives
3. FN us the number of false negatives

A true positive is an outcome where the model correctly predicts the positive class. A true negative is an outcome where the model correctly predicts the negative class. A false negative is an outcome where the model incorrectly predicts the negative class.

$$Precision = TP/(TP + FP) \tag{1.1}$$

$$Recall = TP/(TP + FN) \tag{1.2}$$

Phase 3 commenced with the query-resolution for knowledge graph approach. For this approach, there was a need for an appropriate query-document dataset. This required participation of end users. To this end £20,000 of grant funding was obtained from the Medical Research Council - National Institute for Health Research (MRC-NIHR) Trials Methodology Research Partnership.

1.3.1 Contributions and Publications

A number of proposed techniques were developed as a consequence of the work presented in this thesis. Of note are the following:

1. The Open Information Extraction for Knowledge Graph Construction (OIE4KGC) system for constructing a literature knowledge graph given a corpora of documents to be included [84].

2. The Knowledge Graph And BERT Ranking (GRAB-Rank) based on a hybrid document embedding updating a literature knowledge graph represented CDD [83].
3. Utilizing a hybrid document embedding for query-resolution based on a combination of knowledge graph and

A further contribution of the work is the ORRCA CDD document extraction benchmark data set, which has been made available to the community.

The work presented in this thesis has resulted in a number of publications as follows:

1. Muhammad, I., Coenen, F., Gamble, C., Kearney, A. and Williamson, P. (2020). Knowledge graph construction using open information extraction. Proc. the 2nd International Workshop on Machine Learning and Knowledge Graphs (MLKG2020).
This paper presents the OIE4KGC (Open Information Extraction for Knowledge Graph Construction) approach. Central to the approach is the concept of Open Information Extraction (OIE), using the established RnnOIE tool. Evaluation was conducted using the bespoke ORRCA data sets and the benchmark ClauseID; 400 records from each F-scores of 51% and 37% respectively were obtained. This was essentially a proof-of-concept paper indicating the viability of automatically generating literature knowledge graphs using OIE. This paper presented the first of the above contributions and has formed the foundation of the material discussed in Chapter 4.
2. Muhammad, I., Coenen, F., Gamble, C., Kearney, A. and Williamson, P. (2020). Maintaining Curated Document Databases Using a Learning to Rank Model: The ORRCA Experience. Accepted for publication AI-2020, the 40th Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence (BCS - SGAI).

This paper described the first of the two mechanisms considered in this thesis whereby literature knowledge graph represented CDDs can be updated. The motivation for the paper was the observation that the updating of CDDs is a labour intensive and time consuming task and that ML techniques can help to automate the update process and reduce the workload involved. More specifically the paper introduced a technique for the maintenance and updating of CDDs using a learning to rank model. The approach was evaluated using the ORRCA CDD. Data from the ORRCA original systematic review was used to train the learning to rank model, which was then

tested using the 2015 and 2017 ORRCA updates. The evaluation demonstrated that significant time resource savings could be made using the proposed approach. The work included in the paper has provided the foundation for the material presented in Chapter 5 of this thesis.

3. Muhammad, I., Bollegala, D., Coenen, F., Gamble, C., Kearney, A. and Williamson, P. (2021). Document Ranking for Curated Document Databases using BERT and Knowledge Graph Embeddings: Introducing GRAB-Rank. In: Golfarelli M., Wrembel R., Kotsis G., Tjoa A.M. and Khalil I. (Eds), *Big Data Analytics and Knowledge Discovery, Proc. DaWaK 2021, LNCS 12925*, Springer, pp116-127.

This paper described the Knowledge Graph and BERT Ranking (GRAB-Rank) approach for the updating of Curated Document Databases (CDDs). The novel feature of GRAB-Rank was that it uses a hybrid embedding composed of BERT word embeddings and knowledge graph concept embedding. Evaluation was presented in the context of the ORRCA CDD. The work presented in this paper has provided the foundation for the material presented in Chapter 5 of this thesis.

4. Kearney, A., Roberts, A., Muhammad, I., Gillies, K., Coenen, F., Gamble, C. and Williamson, P. (2022) Using machine learning to maintain and improve the ORRCA resource: Lessons learnt and future considerations Submitted In: *International Clinical Trials Methodology Conference (ICTMC 2022)*

This paper highlights the use of machine learning from the perspective of project management including a brief non-technical overview of the algorithm adopted and its impact on ORRCA Curated Document Database. The paper also highlights the challenges and opportunities identified in terms of timing, and resources used along with the scope of the algorithm.

1.4 Thesis Outline

The rest of this PhD thesis is divided into the following chapters. Chapter 2 provides an overview on the relevant literature. Chapter 3 then provides an overview of the ORRCA CDD used throughout this thesis both as a focus and as an evaluation CDD. The generation of literature knowledge graphs is then considered in Chapter 4, and the OIE4KGC approach proposed. The work in Chapter 4 is directed at providing an answer to SRQ1. Chapters

5 then considers the two proposed mechanisms for updating literature knowledge graphs with a particular focus on document ranking. The work in Chapter 5 is specifically directed at SRQ2 but also, in part, is directed at SRQ4, SRQ5 and SRQ6. Chapter 6 considers query based document retrieval in the context of literature knowledge graph represented CDDs, with a focus on knowledge graph embeddings. The work presented in Chapter 6 is specifically directed at SRQ3, but also covers further aspects of SRQ6. The thesis is completed with some concluding remarks, a summary of the main finding in the context of the overriding research question and the related SRQs, and some suggestion for future work, in Chapter 7.

1.5 Summary

This opening chapter has presented the foundations and motivation for the work presented in the thesis. The chapter included the research question that the thesis seeks the address, the adopted research methodology, the main contributions and publication of the research, and the structure of the remainder of the thesis. In summary, the central idea of the work presented in this thesis is to research and investigate techniques and methods whereby CDD literature knowledge graphs can be generated, maintained and queried. The following chapter provides a literature review covering the relevant previous work related to the thesis.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents a review of existing work relevant to the research presented in this thesis regarding the use of literature knowledge graphs in the context of Curated Document Databases (CDD). Literature knowledge graphs are used for the management of scientific literature. In this context the use of literature knowledge graphs provides two advantages:

1. **Efficiency:** The use of knowledge graphs is a more efficient and effective mechanism, with respect to data organisation and consequent data retrieval, than that associated with more traditional relational database systems.
2. **Deep Learning Compatibility:** Various deep learning algorithms can be applied to knowledge graphs for generating knowledge graph embeddings for document retrieval and document ranking; algorithms that are not well suited to the relational database context.

Before considering the background work relevant to this thesis it is appropriate to first conduct some “scene setting”. It was observed in the introduction to this thesis, that researchers continually strive to expand our body of knowledge, to continuously push out the “knowledge envelope”, through a process of scientific research. Scientific research is an incremental process whereby researchers seek to build on existing knowledge. To do this, researchers must continuously monitor, and be aware of, existing work in their field. This awareness is largely obtained from scientific literature; a body of literature that is

continuously being added to. The quantity of scientific literature has increased at a rapid rate over the last decade; as evidenced in, for example, [56, 81]. Hence the management of such scientific literature is critical to the scientific community. This management is typically conducted using Curated Document Databases (CDDs), literature repositories dedicated to a particular field of study such as the ORRCA CDD considered in this thesis. The management of CDDs when done manually can prove to be very cumbersome and resource intensive [58]. One solution, and that advocated in this thesis, is the use of literature knowledge graphs for the management of CDDs [3, 116]. This literature review chapter thus commences with a review of knowledge graphs, and literature knowledge graphs in particular, in Section 2.2.

Three elements to the use of literature knowledge graphs can be identified, each with related existing research work:

1. **Construction:** The initial construction of a literature knowledge graph given a text corpus such as a collection of scientific paper abstracts as in the case of the ORRCA CDD.
2. **Maintenance:** The maintenance of the literature knowledge graph as the body of knowledge continues to expand (in the form of further publication of scientific papers).
3. **Utilisation:** The utilisation of the literature knowledge graph given that the aim is to make it easier for researchers to identify previous work in the form of scientific papers relevant to their domain of study.

Each of these elements is considered in further detail in Sections 2.3 to 2.5. Section 2.3 considers knowledge graph generation with a particular focus on Open Information Extraction (OIE) techniques. Section 2.4 considers knowledge graph maintenance with particular consideration of document ranking techniques to identify documents to be included in an existing literature knowledge graph. Section 2.5 then gives an overview of document retrieval, concentrating on how knowledge graphs can be queried for the purpose of document retrieval, especially using knowledge graph embeddings. The chapter is completed with some conclusions and a summary in Section 2.5.3.

2.2 Knowledge graphs

This section presents a brief review of Knowledge graphs. Well established examples include Google knowledge graph [122], DBpedia [116], Freebase [9] and YAGO [105]. The idea of knowledge graphs has revolutionised the domain of information retrieval [73, 129, 147]. A knowledge graph is a graph where the vertices represent concepts and the edges relations between those concepts as presented in Figure 2.1, where we have eight concepts $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ and eight relations $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}$. A pair of concepts and a relation connecting them, is referred to as a *relational triple* $\langle c_1, r, c_2 \rangle$ where c_1 and c_2 are two concepts belonging to some set of concepts C ($c_1 \neq c_2$), and r is a relation belonging to some set of relations R . A knowledge graph can therefore be thought of as a structure for a better organization of data in that it provides for a more effective way of accessing this data than in the case of more traditional relational database approaches [3, 48]. There are two kinds of knowledge graphs:

1. Domain Specific Knowledge Graphs
2. Literature Knowledge Graphs

The following two subsections will present a detail overview on the related work relevant to domain specific and literature knowledge graphs.

2.2.1 Domain Specific Knowledge Graphs

The term domain-specific implies that the data used in the generation of the knowledge graph is limited to a specialised domain like biology or computer science. The biomedical domain has a few examples of existing knowledge graphs. One of the earliest works in the biomedical domain was on the use of rdf-extraction for the generation of domain-specific knowledge graphs [41]. More recently, the work presented in [107] focused on the construction of a knowledge graph for the domain of biomedical sciences. Besides the examples in the biomedical domain, there are also domain-specific knowledge graphs dedicated to the field of computer science. A few examples of such domain-specific knowledge graphs directed at the topic of computer science can be found in [36, 70]. An example fragment of a domain-specific knowledge graph on the topic of computer science is shown in Figure 2.2. The figure shows the concepts and relations between the concepts. This fragment of a knowledge graph, contains two concepts, “Tim Berners Lee” and “www”.

Both these concepts are related to each other by a directed edge, representing the relation “has invented”. As mentioned earlier, concepts are real-life objects, also known as entities and are usually nouns. Relations (shown by edges in the graph) are verbs describing the relationship between any two concepts (entities).

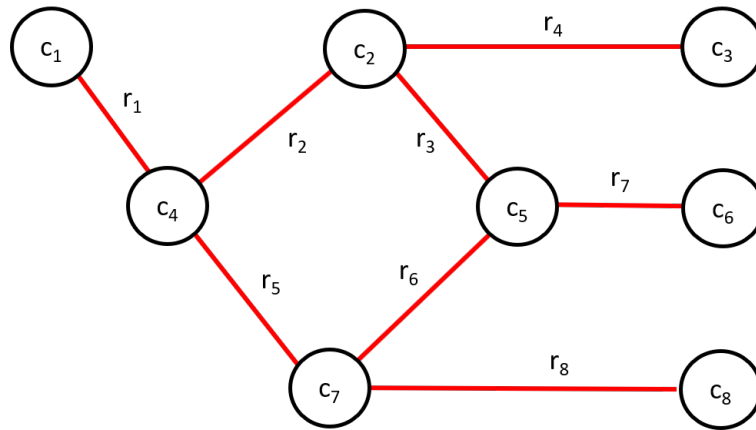


Figure 2.1: Schematic of a simple Knowledge Graph, $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\} \in C$ and $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\} \in R$

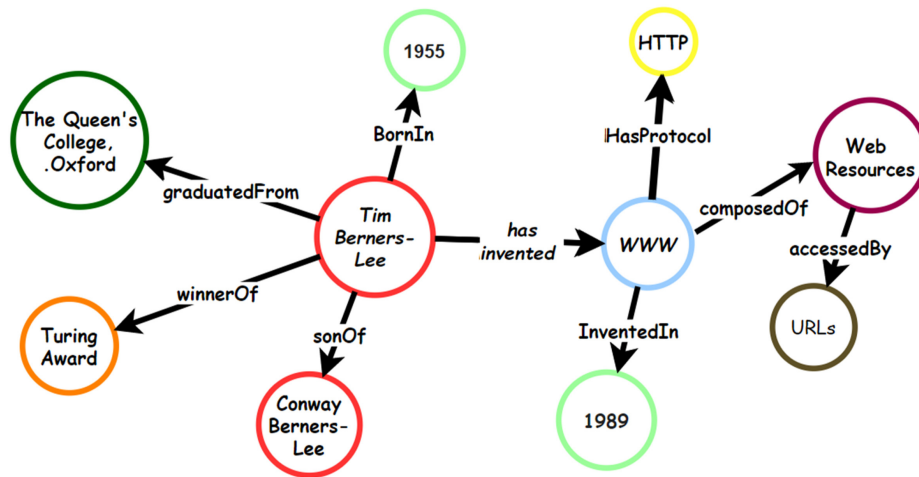


Figure 2.2: A schematic diagram for a domain-specific knowledge graph

2.2.2 Literature knowledge graphs

A literature knowledge graph is a specialised form of knowledge graph. An example fragment of a generic literature knowledge graph is presented in Figure 2.3, whereas a knowledge graph is designed to capture a generic body of information about some real world domain, a literature knowledge graph is designed to capture information related to a specialized domain represented by a document collection (document corpus). Another way of distinguishing between the two is that in a literature knowledge graph some concepts will reference documents (typically scientific research papers) whilst a general knowledge graph will only reference concepts of various kinds. A literature knowledge graph will thus also feature relations linking document with concepts. The fragment of a literature knowledge graph shown in Figure 2.3 features four documents $\{d_1, d_2, d_3, d_4\}$, four concepts $\{c_1, c_2, c_3, c_4\}$ and eight relations $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}$.

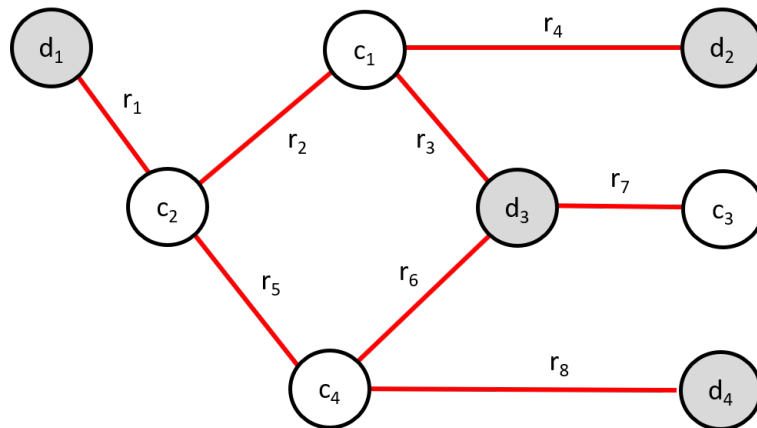


Figure 2.3: Schematic of a simple Literature Knowledge Graph, $\{d_1, d_2, d_3, d_4\} \in D$, $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\} \in C$ and $\{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\} \in R$

The first example of a literature knowledge graph is that used within Semantic Scholar as presented in [3]. Another widely-used literature knowledge graph was created by Microsoft; the literature knowledge graph in this case comprised author vertices, concept vertices, paper vertices and edges connecting them [35]. Semantic scholar and Microsoft's literature knowledge graphs contain documents from all kinds of specialized domains. Some other examples of literature knowledge graphs include Bio2RDF [5] and MeSH [72]. Bio2RF is some of the widely used literature knowledge graphs in life sciences in the context of Human Immunodeficiency Virus (HIV) [90]. MeSH is a literature knowledge graph which

contains knowledge on subject indexing and a search facility for books or journals in life sciences [72]. It was produced by National Library of Medicine (NLM) and is used by many applications. Other examples can be found within the biomedical and life sciences domain where literature knowledge graphs have been used to combine multiple kinds of life sciences data [115]. In [55] the generation of an Ebola literature knowledge graph was described.

2.3 Literature Knowledge Graph Construction

The construction of literature knowledge graphs entails two main challenges:

1. The process for identifying concept entities and relations within a given corpus of scientific texts.
2. The nature of the specific domains that CDDs are directed at, where the vocabulary can be extensive and feature semantic variations and interpretations of concepts used in common parlance.

One of the technologies used to address the above two challenges, and that adopted with respect to the work described in this thesis, is Open Information Extraction (OIE). Broadly, OIE is the process of generating a machine-readable structure for the information contained in a body text and (typically) representing this information as a set of triples expressed using the Resource Description Framework (RDF). Note that RDF is a standard model for data interchange, particularly in the context of the semantic web. OIE, in the context of knowledge graph construction, is typically used to extract the required concepts and relations from sentences in a given a corpus of documents [119]. Two concepts linked by a relation are called the *subject* and *object* arguments, and the relation is the *predicate*. Thus we have triples of the form:

$$\langle \textit{subject}, \textit{predicate}, \textit{object} \rangle \tag{2.1}$$

The OIE techniques used for the generation of knowledge graphs, can be categorised as being either rule-based or supervised [31]. The categories for open information extraction have been adopted from the work in [86]. Both are considered in further detail in the following two sub-sections.

2.3.1 Rule-based OIE for knowledge graph construction

Rule-based OIE techniques use a set of rules for the extraction of concepts and relations, given a corpus of documents [128]. These rules are hand-crafted according to a pre-defined set of target relations along with associated extraction patterns. Well-known examples of rules-based OIE tools include Predpatt [128] and REVERB [32]. PredPratt made use of a set of non-lexicalised rules, defined over universal dependency parses for generating predicate-argument structures. REVERB is what is described as a “shallow extractor” and was particularly focused on avoiding uninformative and incoherent extractions. The advantage offered by rules-based OIE is that no training data is required. The disadvantages are: (i) the resource required to hand-craft the rules which, (ii) usually requires domain experts, and (iii) that the rules will only work with the identified relations. The rule-based approaches for OIE was therefore deemed inappropriate for the clinical trials knowledge graph generation application considered in this thesis.

2.3.2 Supervised OIE for Knowledge graph construction

The second type of OIE technique is supervised OIE, also referred to as “learning-based” techniques, where a training dataset, with labelled entities and relations, is used to train (learn) a model that can be used to automatically identify relations and concepts in previously unseen text [119]. Examples where supervised OIE has been used for entity and relation extraction can be found in [137, 141, 126]. One of the first examples of a supervised OIE model used for extracting relational triples was TextRunner [141]. The process of using TextRunner, for the generation of triples, starts with a small sample of sentences first parsed using Penn TreeBank and then a dependency tree parser is used for the identification of a set of positive and negative labelled “extractions” (training examples). In [126] an OIE technique that relied on a bootstrapping process based on a wikipedia dataset was described. In [137] the idea of a supervised OIE process was discussed that starts with a very small set of seed facts and then learns more relations with the help of distant supervision. The disadvantage of supervised OIE over unsupervised OIE is the requirement for training data which, typically, has to be hand-crafted by domain experts and thus presents a significant resource overhead. The advantage of Supervised OIE over Unsupervised OIE is that the relations and entities of interest do not need to be predefined hence it is often more effective than unsupervised OIE [119]. A further advantage is that pre-trained OIE models are available. One such model is the RnnOIE model [119].

RnnOIE¹ was generated using deep-learning applied to the OIE2016 dataset. The reported evaluation of RnnOIE [119], demonstrated that it was able to outperform many information extraction benchmarks. Thus, given the foregoing, for the literature knowledge graph construction process proposed later in this thesis (see Chapter 4), supervised OIE was adopted using the RnnOIE pre-trained OIE model.

2.4 Literature Knowledge Graph Maintenance

As noted in the introduction to this literature review chapter, an essential element of CDDs realised using literature knowledge graphs is the requirement that they are maintained so that they remain useful to the particular communities which they were intended to serve. Given a collection of candidate documents D we wish to select a subset $U \subset D$ to be included in our CDD. We can conceive of a number of ways that this might be achieved including manual review. Manual review of documents is a labour intensive and time consuming task. The idea advocated in this thesis is to rank the documents in D and then select the top k to form U . The questions are then:

1. What is an appropriate document ranking mechanism to be adopted?
2. What is an appropriate value for k ?

Document ranking is frequently referred to as a score and sort problem. As the name implies, it means listing documents in decreasing order of relevance. Clearly, we wish to automate the ranking process. One way of doing this is to learn a document ranking model, a process frequently referred to as Learning To Rank (LETOR). LETOR models, as the name applies, “learn” a document ranking model given a training set of documents and relevance labels. The discussion on maintaining CDDs presented here is therefore focused on an in-depth review of existing work directed at various kinds of techniques for document ranking, highlighting techniques that increase the effectiveness and efficiency of document ranking. With respect to the work presented in this thesis it should be noted that document ranking also has a role to play with respect to search query resolution whereby potential query responses are listed in order of relevance and the top k returned [91, 4].

Document ranking models can best be categorised according to their adopted document representation. The simplest, “traditional”, approach to document representation

¹<https://github.com/gabrielStanovsky/supervised-oie>

for document ranking is to use statistical measures, such as the frequency of occurrence of selected terms in D [82]. A review of such statistical-based approaches to document representation for document ranking is given in Sub-section 2.4.1 below. The disadvantage of these statistical-based approaches is that they tend to fail to capture the different semantic meanings that can be attached to the selected terms. An alternative document representation method for document ranking, considered more effective than statistical-based document representation, is semantic representation [16, 95]. Semantic document ranking models are discussed in further detail in Sub-section 2.4.2. A further alternative, and that of particular relevance with respect to the work presented in this thesis, is knowledge graph embedding for document ranking. This third alternative is discussed in further detail in Sub-section 2.4.3.

2.4.1 Statistical Document Representation for Ranking Models for Updating Literature Knowledge Graphs

Statistical (traditional) document ranking techniques rely on statistical features such as the frequency of words in documents. There are various kinds of document representations that have been used with respect to statistical approaches to ranking, but the frequently used is the Vector Space Model (VSM) [63]. In a VSM, each dimension represents an attribute associated with the document corpus of interest. Typically, each attribute represents a word or phrase that exists within the document collection. The associated value may be a frequency count, or some kind of weighting, for the word or phrase in question. Term Frequency Inverse Document Frequency (TF/IDF) and Okapi BM25 are popular options here [51]. Each document is thus described by a set of attribute values which serve to locate the document within a multidimensional space. In other words, each document is described by a vector. Using a VSM, a document collection is thus represented by a set of vectors. A disadvantage of the use of VSMs is that they can get very large, given a significant number of attributes and/or a large number of documents, in which case they become computationally cumbersome.

An alternative to a VSM is a Statistical Language Model (SLM). SLMs are defined in terms of the probabilistic distribution of words in a given document corpus. SLMs can therefore be used to give estimates about the relevance of a document in terms of document ranking [74]. Various criteria can be used for the ranking such as Kullback-Leibler (KL) divergence or Maximum Mutual Information.

Regardless of whether a VSM or a SLM is used, traditional, statistically-based, document representations for document ranking have shown good performance in the context of document ranking as evidenced by the encouraging results that have been reported using the datasets provided for text retrieval competitions run as part of the annual Text Retrieval Conference (TREC) [96]. However, the principal disadvantage of these traditional approaches is that they take no account of the context of words or phrases. Semantic document representation for ranking models, as discussed in the following sub-section, have attempted to address this disadvantage.

2.4.2 Semantic Document Representation for Ranking Models for Updating Literature Knowledge Graphs

As the name implies, the representations used by semantic document ranking models capture the meaning behind the relevant terms in a document. Statistical and document representations, notably VSM and SLM, assume each term is independent of its neighbouring terms, whereas semantic document representations for ranking models take into account the context of a term with respect to its surrounding terms, in other words the “semantic” context associated with each term. The distinction can be demonstrated by considering the word “bank”; using a semantic context representation this would comprise a number of vectors depending on the context of the word “bank”, either as:

1. An organisation for investing and borrowing money
2. The side of a river or lake
3. A long heap of some substance
4. The process of heaping up some substance
5. The process of causing a vehicle to tilt to negotiate a corner

Using a statistical non-contextualised representation the word “bank” would be represented using a single vector regardless of context, whereas a contextualized representation of the word “bank” would represent multiple contexts in a single vector. Such a vector is referred to as a *word embedding*. Prior work [16, 88, 144] suggests that contextualized representations, as opposed to statistical representations, are of great importance for effective and efficient document ranking. A frequently cited example of a document ranking model

that used word embeddings to represent documents is ConvKNRM [16]. ConvKNRM is a convolution neural network together with a contextual word representation. Similar work described in [78] used a recurrent neural network. Both of these approaches, and similar approaches, were limited by the requirement for training data. It can be difficult to obtain large amounts of high-quality training data [144], hence recent work has been directed at using pre-trained contextualized term representations for document ranking. The idea is to reuse an existing pre-trained contextual model to generate a word embedding for a given corpus. A popular choice of pre-trained contextual model is the Bidirectional Encoder Representations from Transformer (BERT) model. BERT has been widely adopted for generating word and sentence embeddings. Note that BERT was generated using neural networks that considers the context of a target word using the neighbouring words in a large corpora. BERT has been used with respect to many downstream natural language processing tasks including document ranking [88, 136]. BERT is also used extensively with respect to Google translation, to give another example of its application [21]. An alternative pre-trained contextual model that can be used to generate word embeddings is the Embeddings Language Model (ELMo) [100]. This model is based on deeply contextualized word embeddings generated from Language Models (LMs). The difference between BERT and ELMo is that BERT features a transformer-based architecture whereas ELMo uses a Long Short Term Memory (LSTM) Language model. BERT is Bi-directional, whilst ELMo is semi-bidirectional. BERT relies on a “self-attention” mechanism which gives it the advantage of producing superior word embeddings compared to other models. Self-attention in this context refers to the quantification of the influence that neighbouring terms have on a term under consideration. Language models, such as BERT, that use self-attention mechanisms, are referred to as transformer models. A further advantage of transformer models, such as BERT, with respect to LSTM models such as ELMo, is that transformer models can make use of parallel processing in that attention calculations can be conducted in parallel before calculating the output. This offers an efficiency advantage. For the work presented later in this thesis BERT was adopted with respect to one of the proposed literature knowledge graph updating approaches (see Chapter 4).

2.4.3 Knowledge Graph Embedding Representation for Ranking Models for Updating Literature Knowledge Graphs

The third category of representation for document ranking models considered in this section is the knowledge graph embedding representation [135]. The idea is to utilize the entities from a knowledge graph to form an embedding representation [18, 66, 133, 134, 136]. It has been shown that document ranking models can be improved significantly by using knowledge graph entities and their attributes [71, 133]. Examples of recent work directed at using knowledge graphs for document ranking include the entity-based language models described in [67][73][136]. Another example of existing work that has demonstrated the effectiveness of knowledge graph based document ranking can be found in [4].

Literature knowledge graphs can be utilized in various forms to provide a knowledge source for improving the effectiveness and efficiency of document ranking over alternative statistical and contextual approaches. Many well-known knowledge graphs are publicly available and their entities can be used to annotate documents. Given a document corpus this can be represented using the entities in a given knowledge graph by identifying similar entities in the documents. The knowledge graph entities have links to other vertices in the knowledge graph and these can thus be used to provide context with respect to the similar entities identified in the document corpus. A process referred to as *entity feature expansion*. Entity feature expansion has been used for learning to rank applications as in [18].

Examples from the literature where knowledge graph embedding have been used to represent documents can be found in [26, 71, 136]. In [71] a “latent space model” was proposed for unsupervised document retrieval where rankings of documents was based on their textual similarities with entities in a knowledge graph. A similar model, called EsdRank, is described in [132] that uses connections between knowledge graph entities as features from which to learn a ranking model. In [104] a knowledge graph embedding representations was also used in the context of unsupervised document retrieval. In [26] an entity-based model called the Semantics-Enabled Language Model (SELM) was proposed founded on a knowledge graph entity-based document representation. A unique example of combining word embeddings and entity embeddings for document representation can be found in [73] where the authors used a hybrid embedding model to represent documents for document ranking.

2.5 Literature knowledge graph query resolution

The motivation for using literature knowledge graphs, as already noted, is the desire to provide an efficient mechanism to support researchers in a particular domain. The ability to query a literature knowledge graph is therefore of paramount importance. The advantage offered by knowledge graphs is that their usage speeds up the process of query resolution and consequently information retrieval [129, 18]. Therefore, in the context of the literature review presented in this chapter, the third research area of interest is the querying of literature knowledge graphs for document retrieval.

Knowledge graphs can be queried in a variety of ways. One way is to use a query language of some kind such as SPARQL. However, the use of such query languages requires, on behalf of the user, a comprehensive understanding of the nature of the adopted data model. This drawback has led to the development of query-resolution (QR) systems, also known as Query-Answer systems that enable end-users to express their information needs in natural language [17, 22].

In the context of literature knowledge graphs the focus is on document retrieval, rather than more generic forms of information retrieval. Given a query, the aim is to return a set of documents that match the query. The idea of document ranking therefore also seems to be applicable here. Given a query, documents can be ranked according to their similarity with the content of the query. Document ranking was discussed in the context of literature knowledge graph updating in the previous section where document ranking techniques were categorised as being either:

1. Statistical-based (traditional)
2. Semantic-based
3. Knowledge graph based

In the foregoing the disadvantages of statistical-based approaches was made clear; their inability to take context into account which in turn tended to limit their accuracy. Statistical-based approaches are therefore not discussed further here. The remainder of this section considers semantic-based and knowledge graph-based approaches to literature knowledge graph query resolution in the following two sub-sections.

2.5.1 Semantic Document Ranking Models for Literature Knowledge Graph Query Resolution

The first category of query-resolution system that might be viable in the context of literature knowledge graph is semantic document ranking models. Semantic document ranking models for query-resolution system are based on semantic word embeddings [12, 29]. A general criticism of semantic query resolution models is that they tend to feature lower precision and recall in comparison to knowledge graph embeddings based query resolution methods for document retrieval [110, 18, 73] as considered in the following sub-section. Hence the idea of semantic QA models was not adopted in the context of the query resolution mechanism for literature knowledge graphs presented later in this thesis.

2.5.2 Knowledge Graph Document Ranking Models for Literature Knowledge Graph Query Resolution

The second category query resolution approach whereby literature knowledge graphs can be queried is the knowledge graph embedding approach. Knowledge graph embeddings have already been discussed earlier in this chapter, examples from the literature that promote the ideas of representing queries and documents using knowledge graph embeddings, in the context of QA systems, can be found in [17, 28, 43, 71, 136, 138]. In [138] one of the earliest QA systems that used knowledge graph embeddings was proposed. Natural language questions were mapped to knowledge graph entities using a structured query construction process. Low-dimensional embeddings of n-grams, entity types, and predicates were simultaneously learned from an existing knowledge graph with weak supervision. These generated embeddings were used to measure the semantic associations between lexical phrases, and entity types and logical predicate. The reported evaluation demonstrated that the proposed model outperformed three Knowledge Base QA (KB-QA) baseline systems. This shows the effectiveness of using knowledge graph embeddings in representing questions/queries in a QA system. In [71] a knowledge graph embedding for query based document retrieval and document ranking was proposed directed at Semantic Scholar, an AI supported search engine for academic papers developed by the Allen Institute for Artificial Intelligence [135]. In [136] the Entity-Duet Neural Ranking Model (EDRM) was proposed founded on a neural entity embedding-based search technique. The reported evaluation indicated the effectiveness of using entity embeddings generated using neural networks. In [28] a similar approach was described, using a knowledge graph and a corpus

of documents, to produce ranking scores for the top-k relevant textual passages for a given set of entities. Again, the reported evaluation demonstrated the effectiveness of entity embeddings. Models based on entities were also explored in [43] where entities represent knowledge graph concept mentions, either within queries or documents. Similar significant work directed at the use knowledge graph embeddings for document retrieval can be found in [28, 71, 136]. The main advantage of utilizing the entities held in a knowledge graph is that they provide a relatively simple means for matching queries to documents [40]. A knowledge graph embedding representation was therefore adopted with respect to the work presented in this thesis. Partly because of its superior performance to semantic representations as discussed above, and partly because it seemed a natural choice given the scope of the thesis where the predetermined focus was CDDs represented as literature knowledge graphs.

2.5.3 Summary

The central idea of the work presented in this thesis is to research and investigate techniques and methods whereby CDD literature knowledge graphs can be generated, maintained and queried. This chapter has presented a literature review of the relevant work that underpins the work presented in this thesis. The literature review was divided into three research areas:

1. Knowledge Graph Construction
2. Knowledge Graph Maintenance
3. Knowledge Graph Utilisation

The chapter thus included review of the relevant literature on generating knowledge graphs, updating CDDs using document ranking and the querying of knowledge graphs. In the next chapter, evaluation dataset used with respect to this evaluations reported in Chapter 4, 5 and 6 is discussed.

Chapter 3

Evaluation Dataset

3.1 Introduction

A fundamental requirement for any machine learning technique, is the availability of data sets from which a specific model can be trained and then tested; so called training and test datasets. The work presented in this PhD is directed at Curated Document Databases (CDDs) represented as literature knowledge graphs. More particularly, as noted earlier, the reported research is directed at three research areas:

1. Literature knowledge graph generation
2. Literature knowledge graph updating using learning to rank models
3. Literature knowledge graph querying

For many application domains involving knowledge graphs in general, and literature knowledge graphs in particular, many benchmark datasets are readily available, but this is not the case (at least at time of writing) with respect to CDD literature knowledge graphs of concern with respect to this thesis. Therefore, to act as a focus for the work presented, a specific CDD was considered; namely the Online Resource for Research in Clinical trials (ORRCA) CDD which was produced as a result of the ORRCA research project directed at recruitment strategies clinical trials [58]. This provided the additional benefit that the team of experts, from the Department of Bio-statistics at the University of Liverpool in the UK, who established the dataset, were readily available to assist and advise the author of this thesis. The ORRCA dataset is used throughout this thesis for both illustrative and

evaluation purposes. This short chapter therefore provides an overview of the ORRCA project and the three ORRCA datasets considered in this thesis. The chapter commences, Section 3.2, with a presentation of the background to ORRCA. The chapter then goes on, Section 3.3, to provide a comprehensive overview of three ORRCA datasets collected to support the work presented in this thesis. The chapter is concluded, Section 3.4, with a short summary of the contents of the chapter.

3.2 Background to ORRCA

As noted on the introduction to this thesis, the amount of available scientific literature has increased at a rapid rate over the past decade. There is therefore a need to manage this literature in an efficient and effective manner. As also noted in the introduction to this thesis, for the management of scientific literature one solution is the use of CDDs. ORRCA is an example of such a CDD. The ORRCA CDD was created as part of the ORRCA project [58] whose stated aim was to “to bring together published and ongoing work in the field of recruitment and retention research into a searchable database”¹. The result was the ORRCA CDD; a collection of abstracts concerned with the highly specialised domain of recruitment and retention strategies for clinical trials. The ORRCA CDD was designed to help clinical trialists, and clinical trials scientific researchers, to identify interventions relevant to specific recruitment and retention challenges. Currently there exists a significant ORRCA international community. The ORRCA project was initially funded by the Trials Methodological Research Partnership (TMRP). The ongoing maintenance of the ORRCA CDD is supported by the Trial Conduct Working group within TMRP.

Most of the manual work of identifying relevant documents for inclusion in the ORRCA CDD is currently done by a team of experts. Each person in this team can also be referred to as an annotator, annotating each article for inclusion or exclusion in the ORRCA CDD. The specialized domain experts need to have inter-annotator agreement when reading and flagging each document for inclusion or exclusion in a CDD. A two-stage process, referred to as the *structured review* process, is employed to screen records and identify relevant articles. In Stage 1 the title and abstract of each scientific document to be considered are analysed, by the team of experts, and a “long list” constructed of eligibility documents for inclusion in the CDD. In Stage 2 the team of experts process the long list by reading the

¹<https://www.orrca.org.uk/>

full texts of each scientific document in the long list, and deciding whether each text should be included in the CDD or not. In the case of the ORRCA CDD the main criterion for inclusion was that the documents to be included should reference an applied methodology for determining which outcome domains should be measured in a clinical trial or other forms of health research. The inclusion and exclusion criteria are described in more detail in [42]. The ORRCA database was first set up in 2014. It has been subsequently updated in 2015 and 2017 using the structure review process described above. However, the structured review process is very resource intensive. The automation, or a least partial automation, of the process would therefore be of great benefit. There is no such automation technique available at present that can make the curation and maintenance of CDDs more efficient. The desire to address this overhead was the main motivation for the work presented in this thesis.

Facilitating researchers with reduction in workload and time can be done using machine learning as discussed earlier. A huge number of datasets were publicly available for this PhD; but none of these datasets were annotated as per the requirements of recruitment research for clinical trials. Hence, it was necessary to address the challenge of curating a dataset and asking a team of experts to label a dataset of clinical trial documents.

3.3 Review of ORRCA Datasets

This section presents an overview of the three ORRCA datasets used for training and testing with respect to the research presented in this thesis. This dataset was collated by a team of experts within the Department of Bio-statistics, at The University of Liverpool, who collaborated with the author of this thesis. The raw dataset consisted of multiple features including abstract, title, ISBN, journal name and keywords. In the context of the work presented in this thesis, the generation of CDD literature knowledge graphs, and the maintenance and querying of such knowledge graphs, only the abstract and title of each scientific document was required.

The first of the three datasets consisted of the original curated database collected in 2014 and is referred to as the *Pre-2015 ORRCA* dataset. For machine learning training purpose all three datasets needed to have both negative and positive examples. A positive example in this case was defined as a “relevant” abstracts and a negative example as a “not relevant” abstracts. CDDs do not, by definition, contain negative examples. Hence, the pre-2015 ORRCA dataset had to be augmented with negative examples. The negative

Table 3.1: Statistical overview of the ORRCA evaluation data sets

Database Name	Positive Examples		Negative Examples		Total
	Num.	%	Num.	%	
Pre-2015 Dataset	4570	8.2	51460	91.8	56030
ORRCA 2015 Update Dataset	1302	11.7	9797	88.3	11099
ORRCA 2017 Update Dataset	1027	7.1	13458	92.9	14485

examples were obtained by a member in the team at the Bio-statistic’s Department at the University of Liverpool. Once the ORRCA CDD was established, two updates were undertaken by the Bio-statistics team in 2015 and 2017. The 2015 and 2017 update datasets comprised both positive and negative examples. These update datasets were collected from a combination of searches over public databases like Medline and Scopus [11, 13]. The datasets were all initially stored using EndNote², a commercial reference management software system, designed for the management of bibliographies and references when writing reports, papers and articles. EndNote allows data to be exported in Comma Separated Variable (CSV) format, this was the format used with respect to the work presented in this thesis. Some statistics concerning the three ORRCA evaluation datasets are given in Table 3.1. From the table it can be seen that the number of negative examples was significantly larger than the number of positive examples by an approximate ratio of 1 : 10. The evaluation datasets were clearly highly unbalanced. To mitigate against this, certain weighting techniques were used and implemented from the sklearn python library³ with respect to the machine learning techniques presented later in this thesis (see Chapter 4). The 2015 update dataset was used for training and 2017 was used for testing on the suggestion of the domain experts from the department of bio-statistics university of Liverpool. All of the datasets were pre-processed according the knowledge graph generation and updating approaches proposed in Chapter 4 and Chapter 5.

3.4 Summary

This short chapter has provided an overview of the three ORRCA datasets used with respect to the work presented in this thesis: (i) the Pre-2015 Dataset, (ii) the 2015 Update

²<https://endnote.com>

³Sklearn is a python based machine learning library.

Dataset and (iii) the 2017 Update Dataset. The chapter commenced by giving an overview of the ORRCA project, and then went to consider the three datasets in more detail. In the following chapter, the generation of literature knowledge graphs will be covered.

Chapter 4

Knowledge Graph Generation

4.1 Introduction

In Chapter 1 it was noted that document knowledge graphs provide for a better organisation of data compared to more traditional relational data storage approaches; a better organisation which consequently provides for more effective query resolution than was previously obtainable. Recall also that a literature knowledge graph is a graph where the vertices represent concepts and documents. Concepts are phrases representing real-life ideas. These concepts are linked to documents and other concepts in the knowledge graph by edges. Edges linking concepts to one another represent directional relations, concepts linked to documents are also directional relations.

Literature knowledge graphs can be generated manually from scratch by human experts, however this requires a considerable amount of human resource, as evidenced by the experience gained with respect to the ORRCA CDD [91, 58]. In many cases, and the ORRCA CDD is a good example, the documents that we wish to include in a literature knowledge graph are already available in a computer readable form. Thus it would be of great benefit if a CDD held in a relational format could be automatically translated into a Knowledge graph format; this is the central theme of this chapter. To this end the Open Information Extraction for Knowledge Graph Construction *OIE4KGC* approach, designed to automate the document knowledge graph generation process (given a corpus of documents), is presented in this chapter. The work presented is also designed to address Subsidiary Research Question 1 from Chapter 1:

SQ 1: Given a collection of documents within a CDD, represented using traditional

relational database technology, how can these best be processed so that they form a literature knowledge graph?

The challenge here is: (i) how best to identify the concepts contained in the documents so that these can be encapsulated as concept vertices within the knowledge graph, and (ii) how to identify the linkages between concepts. Note that with respect to the first it is assumed that we do not have a predefined set of concepts; this is certainly the case with respect to the ORRCA exemplar domain considered in this thesis. The above challenges are compounded by the fact that the documents, however they are stored, will be unstructured. Some structure therefore needs to be imposed on the document collection. The idea presented in this chapter is therefore to use an Open Information Extraction (OIE) model [20, 77] whereby content can be expressed as a set of machine-readable Subject Predicate Object (SPO) triples of the form:

$$\langle a_s, r, a_o \rangle$$

where a_s , r and a_o are phrases, and a_s is the subject argument, a_o is the object argument and r is a predicate (relation) between them. In the context of literature knowledge graph generation this is useful because the subject and object can be considered to represent a pair of concept vertices, and the predicate as a directional edge describing the relation linking the subject and object. This makes OIE models ideal for generating knowledge graphs from free text documents.

OIE models can be generated from scratch, but it is more convenient to use a pre-trained, “off-the-shelf”, model provided that the application domain under consideration is not too specific. There are a number of pre-trained OIE models available [34, 119]. Two are considered with respect to the work presented in this chapter:

1. The Recurrent Neural Network OIE (RnnOIE) model [119]
2. LeoLani Triple Extraction Tool [124]

The remainder of this chapter is organised as follows. Section 4.2 presents a formalism of the knowledge graph generation problem considered in this chapter. Section 4.3 presents the proposed *OIE4KGC* approach. This is followed by an evaluation of the results obtained, from experiments conducted using the proposed approach, in Section 4.4. The chapter is completed with a set of concluding remarks in Section 4.5. To aid understanding of the material presented in this chapter the symbols used are presented in Table 4.1.

Symbol	Symbol Definition
G	A literature knowledge graph
V	A set of Vertices in a knowledge graph
E	A set of Edges in a knowledge graph
D	A set of n documents where $D = \{D_1, \dots, D_n\}$
S	A set of sentences where $S = \{S_1, \dots, S_m\}$
T	A set of triples
a_s	Subject argument in a triple
a_o	Object argument in a triple
r	A predicate (relation) between a_s and a_o in a triple

Table 4.1: Symbol table for Chapter 4

4.2 Problem Definition

The objective of the work presented in this chapter is, given a document corpus D , to construct a literature knowledge graph $G = \{V, E\}$ where V is a set of vertices and E is a set of Edges. The vertices in the set V represent either documents (abstracts) or concepts. The edges in the set E represent relationships between documents and concepts, or concepts and concepts. The assumption is that the corpus D comprises n documents such that $\mathbf{D} = \{D_1, \dots, D_n\}$. Each document in the corpus consists of m sentences. The set of sentences for a document $D_i \in D$ is given by $S = \{S_1, \dots, S_m\}$. In order to generate the desired literature knowledge graph the start point is to iteratively extract triples from the sentences in each document $D_i \in \mathbf{D}$, using an OIE tool of some kind, and store these as a set of triples $\mathbf{T} = \{T_1, T_2, \dots\}$. As noted above, the triples to be extracted were of the form $\langle a_s, r, a_o \rangle$, where a_s is the subject argument, a_o is the object argument and r is a predicate; each is represented by a string. The set T is then pruned so that it only comprises those triples with most frequently occurring arguments. It is to be noted that in order for a triple to be complete, it should have both subject and object arguments. Note that a list is also maintained of which triples occur in which document. The triples in the pruned set T thus define pairs of concept vertices in the desired literature knowledge graph. Note that, given the above, every concept vertex will be linked to at least one other concept vertex using a directed edge. It should also be noted that each vertex in the literature knowledge graph will be unique, be it a document or a concept. There cannot be two documents with the same title, nor can there be two concepts described by the same phrase.

4.3 The Open Information Extraction For Knowledge Graph Generation (OIE4KGC) Approach

This section provides an overview of the proposed *OIE4KGC approach*. The approach operates by processing each document $\mathbf{D} = \{D_1, D_2, \dots\}$ in turn. A schematic of the workflow whereby a single document $D_i \in \mathbf{D}$ is processed is given in Figure 4.1. For illustrative purposes a sentence, chosen at random from the ORRCA corpus, is included in the figure; in practice this would be an abstract or a document. From the figure it can be seen that there are four stages that are sequentially applied to \mathbf{D} :

- Triple Extraction: The extraction of a set of triples $\mathbf{T} = \{T_1, T_2, \dots\}$, using an OIE technique. Triples are extracted from each of the sentences in each of the documents in \mathbf{D} .
- Triple Filtering: The filtering of the set \mathbf{T} to give an updated set \mathbf{T} .
- Concept Linking: The identification of subject and/or object arguments that define a concept.
- Knowledge Graph Population: The population of Neo4j database with concepts from the previous stages and documents

The more detailed content of Figure 4.1 will be made clear later in this section. The top-level *OIE4KGC* algorithm is given by the psuedo code shown in Algorithm 1. The input to the algorithm is a document corpus \mathbf{D} and the output is a literature knowledge graph G . The algorithm commences, line 2, by creating a lexicon of the k most frequently occurring nouns in \mathbf{D} . For the evaluation presented later in this chapter $k = 1000$ was used as it was deemed appropriate to selected a value of k , that was suited to the vocabulary of ORRCA domain. This value of k should be neither too small or too large. A small value of k would have meant that only a very few concepts were included in the literature knowledge graph. A large value with too many concepts would be included many of whom would not be sufficiently distinctive. A document corpus \mathbf{D} is then processed document by document. For each document $D_i \in \mathbf{D}$ a vertex in G is created (line 4). Next a set S is created (line 5) comprised of the sentences in D_i . The Spacy sentence extraction tool¹ was used for this purpose, but other appropriate tools could have been used instead. Spacy

¹<https://spacy.io/>

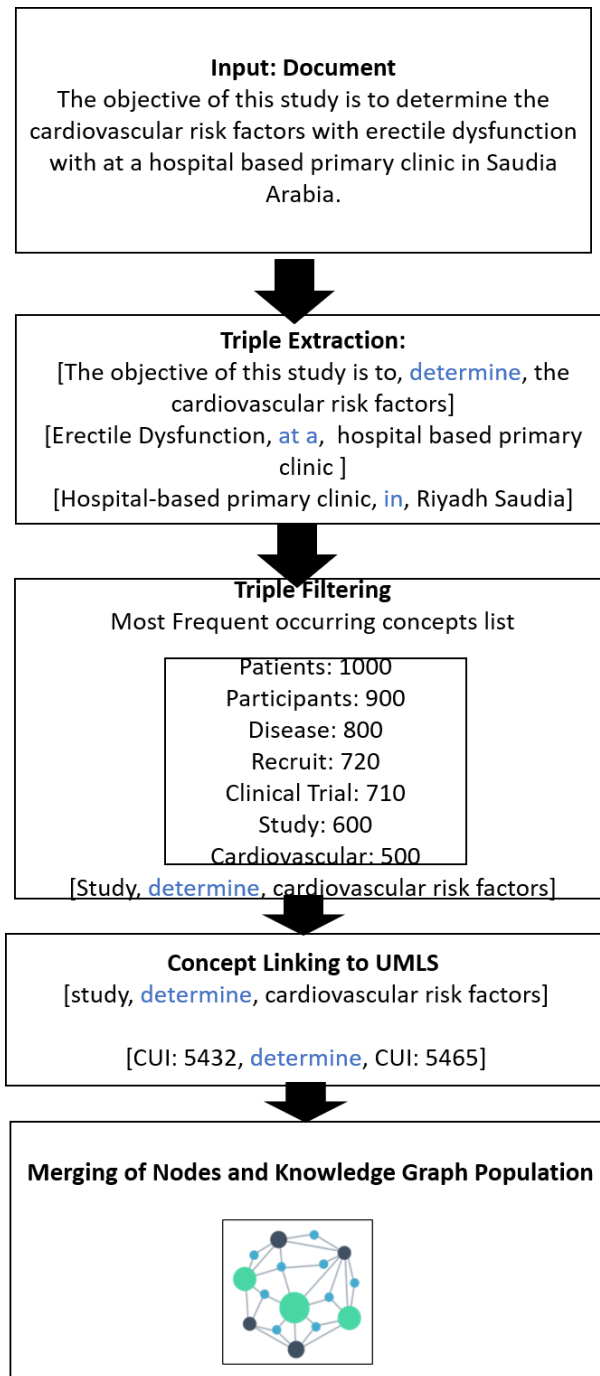


Figure 4.1: A Schematic showing the document processing stages involved in the construction of a literature knowledge graph using the OIE4KGC approach

is a free open-source library for Natural Language Processing. It includes features like Named Entity Recognition (NER), POS tagging and dependency parsing. The set S is then processed sentence by sentence and the triples extracted (OIE4KGC Stage 1). The triples for the current sentence are then processed (lines 9 to 13). First noun chunking is applied (line 10) so as to convert longer noun phrases into shorter ones and also to remove anything that is not a noun because subject and object arguments in each triple should have atleast one noun in them.

Filtering (OIE4KGC Stage 2) is then applied (line 11) using the nouns held in the lexicon L created earlier. Concept linking is then applied (OIE4KGC Stage 3). For the ORRCA domain this involved assigning Concept Unique Identifiers (CUIs) taken from the Unified Medical Language System (UMLS)[8]; but for other domains alternative resources would need to be employed. The set of triples T is then incorporated into G (OIE4KGC Stage 4). This includes adding edges to relevant documents. Each of the four component stages are described in further detail in the following four sub-sections, Sub-sections 4.3.1 to 4.3.4.

Algorithm 1 OIE4KGC Pseudocode

```

1: Input  $\mathbf{D}$ , Output  $G$ 
2:  $L =$  Lexicon of most frequently occurring words in  $\mathbf{D}$ 
3: for  $\forall D_1 \in \mathbf{D}$  do
4:    $G = G$  plus vertex representing  $D_i$ 
5:    $S =$  Set of Sentences in  $D_i$ 
6:    $T = \emptyset$  (Set to hold triples)
7:   for  $\forall S_j \in S$  do
8:      $T =$  Set of triples in  $S_i$  ▷ Stage 1
9:     for  $\forall t_i \in T$  do where  $t_i = \langle a_s, r, a_o \rangle$ 
10:       $t_i = t_i$  with noun chunking applied
11:       $t_i = t_i$  filtered using  $L$  ▷ Stage 2
12:       $t_i = t_i$  annotated with relevant concept links ▷ Stage 3
13:     end for
14:   end for
15:    $G = G$  incorporating content of  $T$  ▷ Stage 4
16: end for
17: Exit with  $G$ 

```

4.3.1 Triple Extraction (OIE4KGC Stage 1)

From Figure 4.1 the first stage in the proposed OIE4KGC process is triple extraction; line 8 of Algorithm 1. Figure 4.1 includes a set of three triples extracted from the example input (taken from the ORRCA domain). The first triple is $\langle \textit{the objective of this study}, \textit{determine}, \textit{cardiovascular risk factors among men} \rangle$, where *determine* is the relation (predicate), and *the objective of this study* and *cardiovascular risk factors among men* are its arguments.

There are many learning-based and rule-based information extraction systems that can be used for the extraction of triples. However such tools are not suitable for domain specific cases [7, 33]. For the *OIE4KGC* approach presented here an OIE approach was therefore adopted. OIE models are designed to address sequence labelling problems [119]. Sequence labelling is a supervised machine learning pattern recognition application domain that involves assigning a categorical label to each member of a sequence of values.

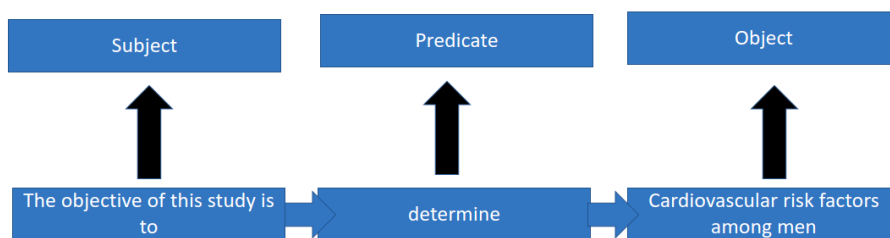


Figure 4.2: A schematic diagram for a predicate-argument sequence labelling problem

The particular sequence labelling problem of interest with respect to the proposed OIE4KGC approach is argument/predicate tagging, where the goal is to label the phrases in a sentence as either arguments, or predicates linking the arguments. This is illustrated in Figure 4.2. In the figure the sentence “The objective of this study is to determine cardiovascular risk factors among men” is considered. The sentence is conceptualised as a sequence. The phrase “determine” is identified as a predicate (relationship) and the pre-fix and post-fix phrases to the predicate identified as arguments, namely the subject and object of the relationship. In the context of the *OIE4KGC* approach the arguments are considered to be concepts to be potentially included as vertices in the final literature knowledge graph to be generated. It is recommended here that OIE tools founded on a pre-trained model should be adopted, because of the resource that would otherwise be requested. For the evaluation presented later in this chapter two OIE tools were considered:

1. **RnnOIE**. The RnnOIE tool was introduced in [119] and is a triple extraction tool based on a Bi-LSTM transducer originally designed to address sequence labelling problems. The Bi-LSTM transducer used has 3 layers. Each LSTM cell comprises 128 hidden units and a linear rectifier (ReLU) activation function.
2. **Leolani**. The Leolani Triple Extraction tool was introduced in [124] to generate triples, given an input sentence, and is based on the idea of context free grammar parsing.

The RnnOIE pre-trained triple extraction tool was selected because:

1. Triple extraction conducted using rule-based OIE tools and context-free grammars based tools, for example as described in [7, 33], are incomplete. A triple is incomplete when it doesn't contain both the subject and object arguments.
2. RnnOIE was one of the first tools that was trained on a hand-labelled dataset (for triple extraction) and can be adapted to domain-specific settings using a “transfer-learning” process [119]. An in-built feature in Pytorch was used for optimization of RnnOIE.

The Leolani triple extraction tool was used for comparison because it was a recently (2018) proposed approach (see [124]). The evaluation of this stage is presented in Section 4.4.

4.3.2 Triple Filtering (OIE4KGC Stage 2)

This section gives an overview of Stage 2 of the OIE4KGC approach, the triple filtering stage (see Figure 4.1 and Algorithm 1). The goal of Stage 2 was to filter the SPO triples generated in Stage 1 so as to only retain the most relevant triples. Prior to filtering, noun chunking was applied to the subject and object arguments. Noun chunking is the process of dividing text into short phrases. For the implementation used with respect to the evaluation presented later in this chapter, the Spacy's Noun Chunker² was used. Once noun chunking was complete, the triple filtering could commence. This involved the removal of “redundant” words found within the triple arguments, and the retainment of “informative” words. The words to be retained, the informative words, were those held in

²<https://spacy.io/usage/linguistic-features>

the lexicon L generated in line 2 of Algorithm 1. Recall, these are the nouns that appear most frequently in the corpus. Thus L holds the most frequent occurring concepts. It is to be noted that the “triple filtering stage” can be seen as a pre-processing stage for cleaning triples.

The approach to filtering SPO triples according to frequency seems a suitable and straight forward approach. It might be possible to identify more sophisticated alternatives filtering techniques, but this was considered to be outside the scope of the thesis (because of the human resource required).

4.3.3 Linking of Clinical Concepts to UMLS (OIE4KGC Stage 3)

Stage 3 of the proposed OIE4KGC approach is the linking of the concepts in the filtered triples; line 12 of Algorithm 1. The objective is to filter the triple vocabulary even further by identifying arguments retained in the triples that reflect the same concept; arguments that express the same concept but in a different manner. The objective of Stage 3 is thus to resolve such ambiguities within the identified arguments, as otherwise the utility of the resulting literature knowledge graph would not be as effective as it might be otherwise have been because concepts that should be linked will not be linked. There are a number of ways whereby the presence of such ambiguity can be addressed, but the idea presented here is to use an existing concept vocabulary. Using an existing concept vocabulary arguments can be annotated with their synonyms so as to allow the desired disambiguation. Concept vocabularies are available for many domains, sometimes in the form of published ontologies. In the case of the ORRCA application domain the arguments were annotated using the relevant Concept Unique Identifiers (CUIs) held in the Unified Medical Language System (UMLS) Metathesaurus [111]. For other application domains, other kinds of appropriate ontologies can be used as well. With respect to the illustrative example included in Figure 4.1, the arguments have been annotated with a unique CUIs. The word “study” is related to the CUI 5432, while the phrase “cardiovascular risk factors” to the CUI 5465. It is to be noted that since the application domain of focus was “recruitment strategies for clinical trials”, specialized medical ontologies like ICD9 [117], SNOMED CT [23] and MeSH [69] could not be used for comparison.

4.3.4 Knowledge Graph Population (OIE4KGC Stage 4)

The arguments in the set triples T associated with a document $D_i \in \mathbf{D}$, were disambiguated in Stage 3 as described above (Sub-section 4.3.3), using Concept Unique Identifier(CUI). This section presents detail concerning the final stage in the OIE4KGC approach, knowledge graph population. Knowledge graph population is the process of populating a knowledge graph database with the identified concepts(vertices) and edges (relationships) between them. There are a range of graph database managements systems available. For the implementation of the proposed OIE4KGC approach, used with respect to the evaluation presented later in this chapter, the Neo4j NoSQL graph database management system³ was adopted because of its current popularity. The data structures provided within Neo4j were used for the storage of concept vertices, document vertices and edges (relations between concepts).

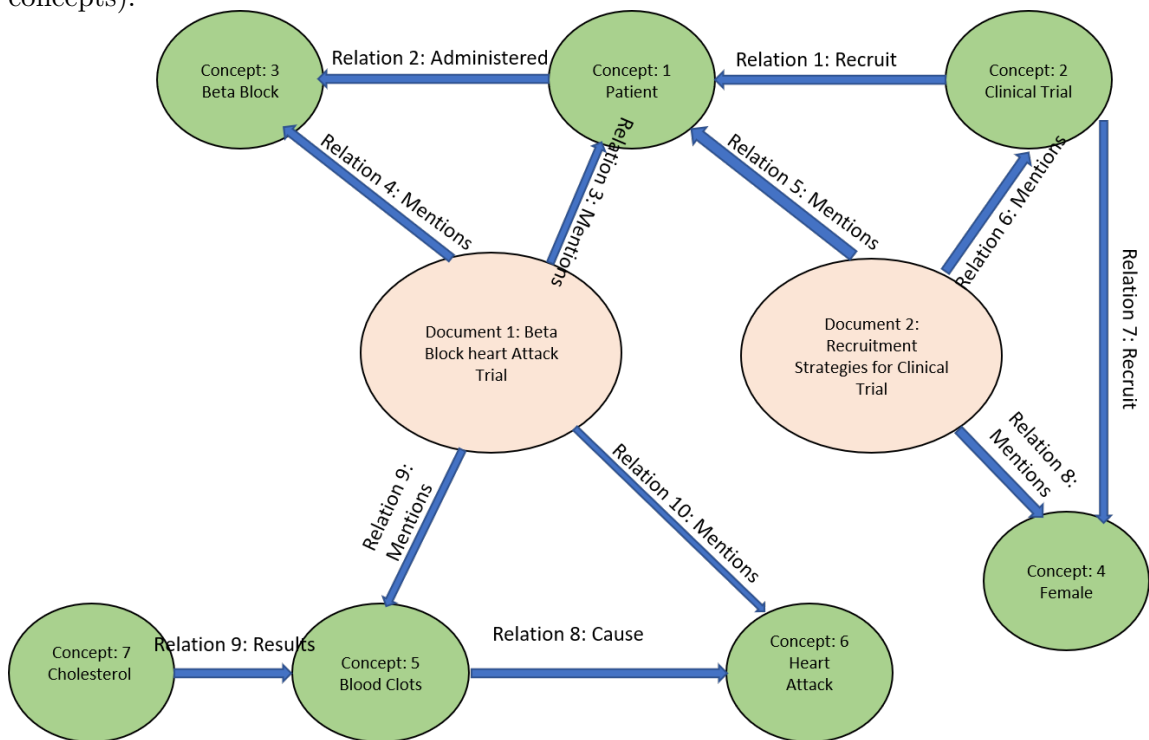


Figure 4.3: An example of a literature knowledge graph generated using OIE4KGC

The literature knowledge graph end-goal was generated by processing each of the $t_i \in T$, for a given document $D_i \in \mathbf{D}$, in turn. For each triple two kinds of vertices were created

³<https://neo4j.com/>

in the knowledge graph, v_s and v_o , connected by the given relation r , and each connected to the document vertex created for D_i . These were then compared to the knowledge graph G so far. There are four options explained below:

1. If v_s and v_o match two vertices v_1 and v_2 in G : merge v_s and v_o with v_1 and v_2 adding the relation r if not already in existence.
2. If v_s matches a vertex v_1 in G , but v_o does not match any vertex in G : merge v_s with v_1 .
3. If v_o matches a vertex v_2 in G , but v_s does not match any vertex in G : merge v_o with v_2 .
4. Otherwise (v_s and v_o do not match any vertices in G): No merging.

For literature knowledge population and merging, Neo4j has a merge utility. The merging of vertices is done to ensure that each concept vertex in the literature knowledge graph is identified by a unique ID. Figure 4.3 shows an illustrative example of a literature knowledge graph generated using the *OIE4KGC* approaches. Note that the literature knowledge graph includes two types of vertices:

1. Concept vertices (Green color)
2. Document vertices (Peach color)

In the case of the ORRCA application the concept vertices possessed two properties:

1. An argument string (the concept name)
2. A CUI based ID that connects the argument string to the UMLS Metathesaurus.

It is to be noted that none of the concepts can have the same CUI and the stage of concept linking using the UMLS Metathesaurus ensures that each concept has a unique ID. The stage of “knowledge graph population” stage could possibly be evaluated by comparison of various graph databases. Given that in the initial stages of the PhD, Neo4j was the only open source choice available hence it was adopted for knowledge graph population. The evaluation of “knowledge graph population” can be carried out by comparing multiple graph databases as future work.

4.4 Evaluation

This section presents the evaluation of the proposed *OIE4KGC* approach for knowledge graph construction. The focus of the evaluation was Stage 1 of the proposed OIE4KGC process, the triple extraction stage. This was because it was considered to be significant with respect to the effectiveness of the proposed process. The second and third stages, triple filtering and linking, are essentially data processing stages where the number of identified triples is reduced. Both stages adopted simple straightforward approaches that therefore were considered not to merit any significant evaluation. The fourth stage comprised the creation of the desired literature knowledge graph by populating a graph database using a Graph Database Management System (GDBMS). Neo4j was used for this purpose although it could equally well have been conducted using some other form GDBMS. The Graph Database population process was not considered to require particular consideration in terms of comparative evaluation with some alternative. The objectives of the evaluation presented here were this:

1. To compare the RnnOIE and Leolani triple extraction Tools.
2. To conduct a qualitative comparison between the triple extractions of RnnOIE and Leolani tools.

Two data sets were used for the evaluation of the proposed approach.

1. The ORRCA data set [58]
2. The ReVerb ClauseIE dataset⁴ [19]

The ORRCA data set mentioned above was presented and explained in Chapter 3 in Section 3.3 so as to provide the reader with a background knowledge on the dataset application domain. For the scope of this chapter and with respect to the ORRCA dataset, a set of 100 sentences was randomly chosen from the ORRCA dataset and a set of triples were generated, using RnnOIE and Leolani, from these 100 sentences. The manual inspection of triples was done by the author so as to determine whether a triple is correct or not; therefore there was a human resource consideration. Each sentence had on average 5 triples which resulted in around 500 triples to be manually inspected for evaluation. The

⁴<https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/software/clusie/>

choice of 100 sentences was therefore an amalgamation of expediency versus effectiveness. The same argument was implied in [1] where the authors hand annotated a sample of 100 sentences to compare a number of methods for extracting subject-verb-object triples from news texts. A further example can be found in [118] where 100 sentences were also used.

The ReVerb dataset is a a triples dataset extracted as a result of extraction facilitated by the ReVerb OIE tool [19]. A subset of the ReVerb dataset was taken comprised of 100 sentences randomly selected from this dataset because a similar number had been selected for the ORRCA dataset.

The evaluation metrics used was F-score, the harmonic mean of precision and recall. Precision was defined as the number of correct triples extracted divided by the total number of triples extracted. Recall was defined as the number of correct triples extracted divided by the overall number of correct triples that should have been extracted, for the selected 100 sentences [1]. The RnnOIE and Leolani triple extraction tool were evaluated at the sentence level as triples are extracted at the sentence level.

4.4.1 Comparison of OIE Tools

The evaluation of the RnnOIE and Leolani tools with respect to the two selected datasets is presented in Tables 4.2 and 4.3. Table 4.2 shows that the RnnOIE was able to achieve an F-Score of 51% when applied on the ORRCA data set and 37% using the ReVerb dataset. The results presented in the Table 4.2 indicate that the precision was better when using the ORRCA dataset when compared with the ReVerb dataset. The reason for this difference in precision is possibly due to the nature of the sentences in the ORRCA dataset. Sentences in the ReVerb dataset are shorter compared to those in ORRCA dataset.

Table 4.3 shows that the Leolani triple extraction tool was able to achieve an F-score of 41% on the ORRCA dataset and 29% using the ReVerb dataset. Table 4.3 also indicates that for the Leolani triple extraction tool, the precision was better using the ORRCA dataset. Both the ORRCA and ReVerb datasets have structural differences at the sentence level and this could be the reason for such difference in precision. It was conjectured that triples extracted from the ReVerb dataset had numerical values in the arguments; which, using the proposed approach, results in a triple being discarded. It was also observed that the sentences in the ORRCA dataset on average had 30 words, whereas the average number of words per sentence in the ReVerb dataset was ten. Comparing the obtained results using the RnnOIE tool and the Leolani triple extraction tool from Table 4.2 and Table 4.3, it can

be seen that the RnnOIE tool coupled with the OIE4KGC approach performs better on both datasets when compared to the Leolani triple extraction tool. This could be due to the fact that RnnOIE is pre-trained on a large triples dataset whereas the Leolani tool is based on context Free Grammar parsing. It can therefore be observed that triple extraction tools which use the concept of “transfer learning” are more effective at generating triples than other existing OIE tools. This observation is supported by previous work, for example that reported in [119]. The results in Table 4.2 also show that the RnnOIE tool is appropriate with respect to the clinical document collections in the ORRCA dataset.

4.4.2 Qualitative Analysis of Open Information Extraction Tools

This section gives an overview of the qualitative analysis of the triple extractions conducted using the two triple extraction OIE tools. The two tools used were the RnnOIE tool and the Leolani triple extractor. There was a need to analyze the output of OIE tools qualitatively as qualitative analysis had been conducted in similar work, such as that reported in [119]. The reason behind this analysis was to determine the nature and correctness of the triples extracted by the considered OIE tools. For longer sentences in a textual dataset, an OIE tool gives multiple extractions, whereas for shorter sentences a single extraction results. The triple extractions were judged by considering the following:

1. **Clarity and correctness of extractions:** In Table 4.4 and 4.5, example extraction outputs using the Leolani and RnnOIE triple extraction methods are presented. A triple extracted by either of the tools is considered to be clear and correct, if it doesn't have any missing subject or object arguments. Secondly, for a triple generated by an OIE tool, it should also make logical sense when read from the subject argument to the object argument. We can see in Table 4.4 and 4.5 that both the triple extraction tools have similar outputs on the sentence, “The two groups had similar levels of functional impairment and similar ages at onset of symptom” and that this triple is correct and clear in meaning. Both Leolani and RnnOIE make mistakes on the Sentence “Conducting clinical research involving critically ill patients is challenging” in terms of the triple generated. RnnOIE was not able to identify a second argument for the sentence “Conducting clinical research involving critically ill patients is challenging” as shown in Table 4.4. Triples with missing arguments or which clearly do not make logical sense (when read from left to right) are regarded as unclear and not correct. The sentence “The two groups had similar levels of functional impairment

and similar ages at onset of symptoms” in both Tables has the only correct triple identified by both tools with tangible arguments and predicate; the triple also makes logical sense.

2. Distinctiveness of triples extracted: The distinctiveness of triples generated by an OIE tool is defined as the uniqueness of the triple, or that triples are not duplicated if multiple triples have been generated. In Tables 4.4 and 4.5, we can see that RnnOIE extracts only single triples whereas Leolani might extract multiple triples for longer sentences. Sentences like “The nations health maintenance organisations were required to tell the Federal government by midnight Monday” resulted in duplicate extractions using the Leolani triple extraction tool whereas RnnOIE only produced a single triple as the output. Duplication of triples as a result of using an OIE tool is a disadvantage because uniqueness of triples is important in triple generation.

Dataset	Precision	Recall	F-score
ReVerb dataset	0.473	0.311	0.375
ORRCA dataset	0.783	0.391	0.512

Table 4.2: Performance of RnnOIE using the ORRCA and ReVerb dataset

Dataset	Precision	Recall	F-score
ReVerb dataset	0.30	0.28	0.293
ORRCA dataset	0.540	0.333	0.412

Table 4.3: Performance of Leolani using the ORRCA and ReVerb datasets

Sentence	Triple Extracted
The two groups had similar levels of functional impairment and similar ages at onset of symptoms	“[ARG0: The two groups] [V: had] [ARG1: similar levels of functional impairment and similar ages at onset of symptoms]”
Older adults receive most of their end-of-life care in the community	“[ARG0: Older adults] [V: receive] [ARG1: most of their end-of-life care]”
Conducting clinical research involving critically ill patients is challenging.	“[V: Conducting] [ARG1: clinical research involving critically ill patients] is challenging.”
The nations health maintenance organisations were required to tell the federal government by midnight Monday	“[ARG0:][V:Conducting][ARG1:clinical research involving critically ill patients]”

Table 4.4: Example triples extracted using the RnnOIE triple extraction tool applied to sentences in the ORRCA dataset

4.5 Summary

This chapter has presented an overview of a proposed literature knowledge graph construction technique called *OIE4KGC* that can be used to generate from a corpus of documents. The work presented was divided into Four major stages:

1. Triple Extraction
2. Triple Filtering
3. Linking of clinical concepts to UMLS
4. Merging of vertices and knowledge graph population

The chapter started off with an introduction and background to the proposed *OIE4KGC* approach followed by the proposed technique and the results obtained from the evaluation. Two open information extraction tools were adopted for comparison, the RnnOIE tool and the Leolani tool. A comparison of the tools in terms of precision, recall and F-score was presented in Section 4.4 using the ORRCA and ReVerb datasets. A qualitative analysis of the triples produced from each of the tools was also presented considering two factors:

Sentence	Triple Extracted
The two groups had similar levels of functional impairment and similar ages at onset of symptoms	“[ARG0: two groups] [V: have] [ARG1: similar-levels-of-functional-impairment]”
Older adults receive most of their end-of-life care in the community	“[ARG0:older-adults] [V:receive] [ARG1:end-of-life-care]”
Conducting clinical research involving critically ill patients is challenging	“[ARG0:research][V:conduct][ARG1:is challenging]”
The nations health maintenance organisations were required to tell the federal government by midnight Monday	“[ARG0:The-nations-health-maintenance-organisations][V:require-to][ARG1:tell-the-federal-government-by-midnight-Monday]”, “[ARG0:The-nations-health-maintenance-organisations][V:require][ARG1:federal-government-by-midnight-Monday]”

Table 4.5: Example triples extracted using the Leolani triple extraction tool applied to sentences in the ORRCA dataset

(i) clearness and completeness of the triples extracted. and (ii) the distinctiveness of the triples generated. The results indicated that the triples extracted by RnnOIE were more clear and distinctive than those extracted using the Leolani triple extraction tool. In summary, the central idea of this chapter was to discuss how triples can be extracted using open information extraction and the generated triples evaluated in terms of effectiveness, completeness and distinctiveness. The generated triples are then used to create a literature knowledge graph. In the following chapter, the updating of literature knowledge graphs will be discussed.

Chapter 5

Maintenance of literature knowledge graph using document ranking

5.1 Introduction

The previous chapter explained how a Curated Document Database (CDD) can be represented as a literature knowledge graph so that end users can take full advantage of the benefits offered by knowledge graphs. However, as already noted earlier in this thesis, the amount of scientific literature published each year is increasing rapidly. Hence, so that our knowledge graph represented CDD can remain useful, it is essential that it is maintained (further relevant documents added as and when they come available).

The maintenance of a CDD, regardless of whether it is stored in the form of a literature knowledge graph or not, requires a review of a set of candidate publications that might potentially be included, U , to identify a subset Q of publications to be included in the CDD ($Q \subset U$). In the case of the ORRCA CDD, used as a focus for the work presented in this thesis, a manual *systematic review* process was used previously to maintain the CDD. Other examples where such systematic reviews have been adopted can be found in [58, 91]. Systematic review, as in the case of any other manual approach to maintaining CDDs, requires substantial human resource. The central theme of this chapter is how best to automate the process of systematic review with respect to the maintenance of knowledge

graph represented CDDs. The work presented in this chapter seeks to address subsidiary research questions SQ2 and SQ5 from Chapter 1:

[SQ2] *Given an existing CDD, represented as a literature knowledge graph, how can this knowledge graph be maintained to ensure that it is up to date.*

[SQ5] *In the context of document ranking can the concepts within a literature knowledge graph be utilized to improve a document ranking mechanism and how would this operate?*

The central idea promoted in this chapter is that of a document ranking mechanism whereby the candidate documents in U are ranked in decreasing order of relevance, so that the top k most relevant documents, the set Q , can be selected for inclusion in the CDD. The question is how can this ranking best be achieved. It is suggested here that some form of machine learning is adopted whereby a model is trained to rank documents, a process known as a *Learning-To-Rank* (LETOR) [82, 85]. The motivation here is that LETOR offers advantages of efficiency and effectiveness compared with the hand-crafting of a ranking model, provided that appropriate training data is available [58]. The LETOR process takes as input a pre-ranked set of documents D , which is then used to train a ranking model. The quality of the model can be assessed using a second pre-ranked set of documents (a test dataset). The LETOR model once generated can be applied to U to generate Q . LETOR models can be categorised as being either:

1. Pointwise approaches [47]
2. Pairwise approaches [127]
3. Listwise approaches [102]

Pointwise approaches consider a single document at a time to produce a ranking; each document is considered to be a “point” in a “document space” and assigned a ranking value. Each document is assigned a ranking (a relevance score) using, a pre-trained regression model. On completion the documents can then be sorted into rank order [47]. Pointwise approaches offer the advantage of simplicity and have been shown to work well [47, 91]. The following two approaches are based on pointwise ranking model.

1. The CN approach

2. The Knowledge Graph And BERT Ranking (GRAB-Rank) approach.

The first is founded on work presented in [91] where a pointwise LETOR approach was considered that used a feature vector representation of U . The significance of the work presented in [91] is that it was directed at a clinical outcomes CDD not dissimilar from the ORRCA CDD (although not represented in the form of a knowledge graph). This approach is referred to as the CN approach after the initials of the author of [91]. The CN approach uses a feature vector representation that represents a set of n-grams selected using Term Frequency - Inverse Document Frequency (TF-IDF) scores. The advantage of feature vector representations based on n-grams and TF-IDF, as used with respect to the CN approach, is that they are simple to implement. The disadvantage is that they fail to capture the semantic meaning present between words in the input set of documents.

The alternative is to use a word embedding of some kind. A word embedding is a learnt numeric vector representation of a word that captures its “usage” by taking into account preceding and proceeding words. The representation for a word embedding takes the form of a numeric vector comprised of many elements (> 300 for even the simplest word embedding). The embedding for a sentence can be defined as the average of the component word vectors in a sentence. The evidence from recent literature [27, 85] suggests that LETOR approaches founded on word embedding are more effective compared to traditional approaches, such as CN approaches, founded on n-grams and TF-IDF. However to learn a set of word embeddings requires a large training dataset. This presents a particular challenge, especially in the context of the CDD application domain considered in this thesis where such a training dataset is not readily available. A popular solution is to use an existing embedding that has been pre-learnt. Arguably the state-of-the-art in pre-learnt embedding models is the Bidirectional Encoder Representations from Transformers (BERT) model [88] previously referenced in Chapter 2.

In the context of knowledge graph represented CDDs, pre-trained embeddings, feature two disadvantages:

1. By their nature, pre-trained embeddings tend to be generic, whilst CDDs, by definition, tend to be domain specific. Therefore a pre-trained embedding may not be the most appropriate.
2. Pre-trained embeddings make no use of the available structure of the knowledge graph represented CDDs as advocated in this thesis.

One solution to the first of the above disadvantages is to use a more domain-specific pre-trained embedding. For example, there are many variations of BERT embedding that have been trained for particular domains of interest such as SciBert, a pre-trained BERT-based language model for performing scientific NLP applications. Another option is to “fine tune” an existing embedding model; many embedding environments support this. However, as in the case of generating a dedicated embedding from scratch, this requires training data, thus not a realistic option in the case of the CDD application considered in this thesis because of the resource required. The solution presented in this chapter is to use some form of graph embedding [38, 125], a *Knowledge Graph Embedding*. This offers three advantages with respect to knowledge graph represented CDDs:

1. It does not require dedicated training data.
2. It makes full use of the information captured in the knowledge graph represented CDD (the second of the two disadvantages of generic pre-trained embedding models listed above).
3. It is not necessary to first create a knowledge graph as this will already exist, although if this was a requirement the techniques explored in Chapter 4 could be adopted.

A Knowledge Graph Embedding could be used on its own for the purpose of generating a LETOR model to support the maintenance of knowledge graph represented CDDs. However, the idea presented in this chapter is that a better LETOR model can be used if a “general purpose embedding” is combined with a “domain specific embedding”. This is the philosophy underpinning the second approach considered in this chapter, the GRAB-Rank approach. For the general purpose embedding, as indicated by the GRAB-Rank acronym (Knowledge Graph And BERT Ranking), BERT was used because of it being state-of-the-art in word embeddings [21].

The rest of this chapter is structured as follows. Section 5.2 gives a problem definition for the Knowledge Graph represented CDD maintenance problem. The relevant notations and symbols used with respect to this chapter are defined in Table 5.1 below. Section 5.3 presents the generic LETOR framework adopted with respect to the work presented in this thesis. Further detail of the CN and GRAB-Rank approaches are given in Sections 5.4 and 5.5. The comparative evaluation of the two approaches is then presented in Section 5.6. The chapter ends with a set of concluding remarks in Section 5.7.

Symbol	Symbol Definition
D	A database of documents for a CDD of the form $D = \{d_1, d_2, \dots\}$
U	A set of candidate publications for inclusion in a CDD
Q	The set Q , generated using a systematic review process applied to a larger data set U ($Q \subset U$)
k	The top k most relevant documents in Q selected for inclusion in CDD d_i a document in D
σ	A threshold defined as the percentage of documents in U to be included in Q
θ	A frequency threshold value θ used to decide which n-grams, using the CN approach, should be included in a feature vector representation
p	The probability that a previously unseen document in U belongs to the positive (to be included) class, the set Q
w	An n-gram, defined as a contiguous sequence of words found in a piece of text (a sentence in a document)
n	Number of words in a n-gram
$tfidf(w)$	The TF-IDF value for an n-gram w
G	A knowledge graph
R	A set of random walks over a knowledge graph G
rw	Random walk length over a knowledge graph
v_j	A concept vertex in G (as opposed to a document vertex)

Table 5.1: Symbol table for Chapter 5

5.2 Problem Definition

This section provides a formal definition of the problem addressed in this chapter. The reader may find it useful to refer to Table 5.1 with respect to the following. A CDD is defined as a set $D = \{d_1, d_2, \dots\}$ where each $d_i \in D$ is a document, for example an abstract. To maintain D it is necessary to periodically add a set $Q = \{q_1, q_2, \dots\}$ of recent relevant publications, $D_{new} = D \cup Q$. The set Q , as noted in the introduction to this chapter, is traditionally generated using a systematic review process applied to a larger data set U ($Q \subset U$). A systematic review is performed by a team of domain experts, using a search strategy with the goal of identifying Q in U [58, 91]. Systematic reviews are resource intensive. Hence, it is suggested in this chapter that a learning-to-rank (LETOR) approach is adopted. The idea is that the resulting ranking model will be able to order

U according to relevance. The top k most relevant documents can then be selected for inclusion in D .

5.3 The Proposed LETOR Framework

This section presents an overview of the end-to-end LETOR process as adopted with respect to the work presented in this chapter. A schematic of the proposed framework is presented in Figure 5.1. The process commences, top left of the figure, with an existing Curated Document Database D and ends with an updated Curated Document Database, bottom middle of the figure. By definition D only contains “positive” examples. To train a ranking model we need both positive and negative examples. The first step is thus to augment the set D with negative examples. The set of positive and negative examples then needs to be pre-processed so as to generate some kind of vector representation of the documents (feature vectors and/or embeddings). This is then the input to the selected learning to rank algorithm which then generate a ranking model (the grey box with rounded corners to the right of the figure). This model can then be used to rank documents to be potentially included in D .

The set U of documents to potentially be included needs to be pre-processed and feature vectors generated for them so that they are represented in the same manner as the set D . The set U of candidate documents is typically generated from a bibliographic database of some kind, that list titles, authors and publication details. The generated LETOR model (the grey box with rounded corners to the right of the figure) was used to assign a probability value p to each document in U . This probability value is treated like a score. The probability p that a previously unseen document in U belongs to the positive (to be included) class. The probability that the document belongs to the negative class (the not to be included class) is then $p - 1$. The result, bottom right of the figure, is the set of documents U ranked according to p . A “cut-off” threshold value σ is then applied and the top k selected (the set Q). A final “human screening” is then undertaken and the final selection added to the CDD. The human screening is undertaken to ensure no anomalies are included in Q (the top k documents). Only the top k documents are selected as these will be the most relevant. This strategy is analogous with that adopted with respect to web search engines where the end-user is assumed to be only interested in the most relevant hits (to avoid “information overload”) since a domain expert will potentially be only interested in seeking the a limited number of relevant clinical trial abstracts at the top of documents

list.

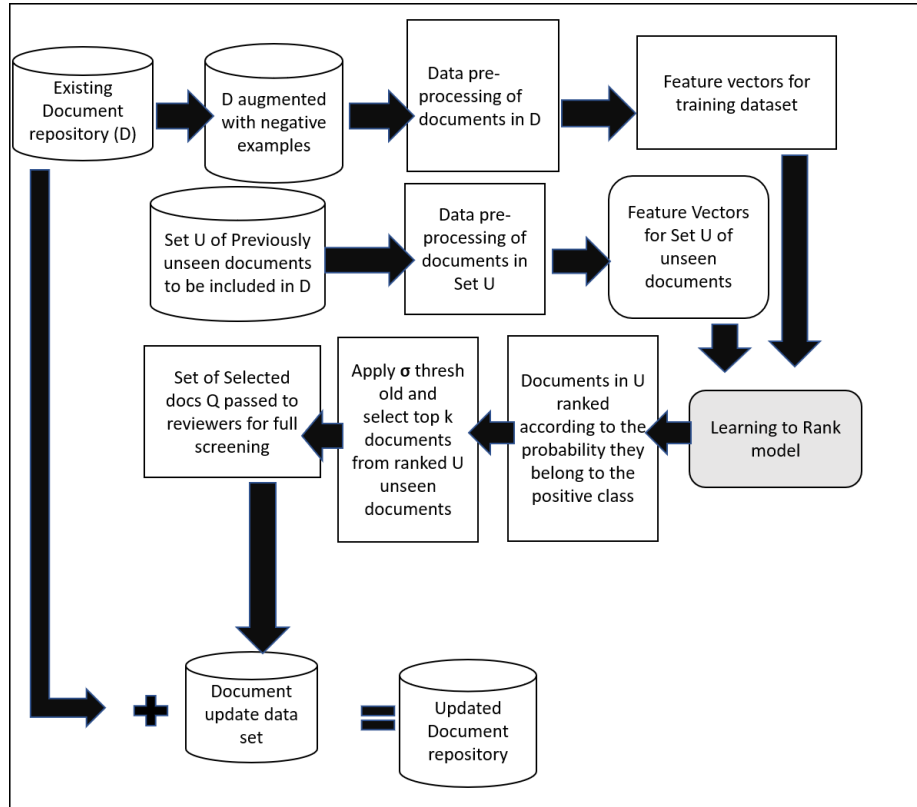


Figure 5.1: The Proposed LETOR Framework

Note that the proposed LETOR framework generates a probability score which is then used to rank documents and select documents for inclusion in the CDD. The proposed LETOR framework is basically a static pointwise document ranking approach and similar literature [91] supports the idea of such static pointwise document ranking for selection of documents for CDDs. As noted above, two are considered in this chapter; the CN approach and the GRAB-Rank approach. Further detail concerning these two approaches is presented in the following two sections, Section 5.5 and 5.4 respectively.

5.4 The CN Approach

This section presents the CN approach, the first of the two approaches for maintaining CDD represented as knowledge graphs, considered in this chapter. As noted earlier, the CN approach is founded on the work presented in [91] where a LETOR algorithm was proposed directed at clinical outcomes articles² which has similarities with the ORRCA data set used here for evaluation purposes in this thesis. The distinctions between the proposed CN approach and that presented in [91] are:

1. The LETOR presented in [91] used the SGD Classifier from the Sklearn³ Python library whilst the proposed CN approach uses the SVR algorithm (also from Sklearn).
2. CN approach used two kinds of n-grams ($n \leq 3$) over words in each document, whilst the LETOR presented in [91] used five kinds of n-grams ($n \leq 5$) over words in each document.

Figure 5.2 presents a schematic of the CN approach. It is useful to compare this schematic with the general schematic presented previously in Figure 5.1. The main differences are in the pre-processing stages where TF-IDF values, for n-grams included in the document set (D or U) are used to generate feature vectors. For the pre-processing, the python Natural Language Processing Toolkit (NLTK)¹ library was used along with stop word removal. The generated feature vectors were then fed into the SVR learning to rank model for training. The rest of the stages in the LETOR process are the same as for the general LETOR framework presented in Figure 5.1.

For the ORRCA application considered here, the ranking model was trained using log loss². Using the CN approach the candidate set of documents U was represented using feature vectors, as seen in the top right corner of Figure 5.2. In natural language processing the features used in a feature vector representation are typically keywords. These can be defined in a variety of ways, for example using a lexicon. As noted earlier, the approach adopted with respect to the CN approach was to select frequently occurring n-grams. An n-gram is defined as a contiguous sequence of n words found in a piece of text (a sentence

²<http://www.comet-initiative.org/>

³Sklearn Python machine learning library

¹<https://www.nltk.org/>

²Logarithmic loss (related to cross-entropy) measures the performance of a classification model where the prediction output is a probability value between 0 and 1

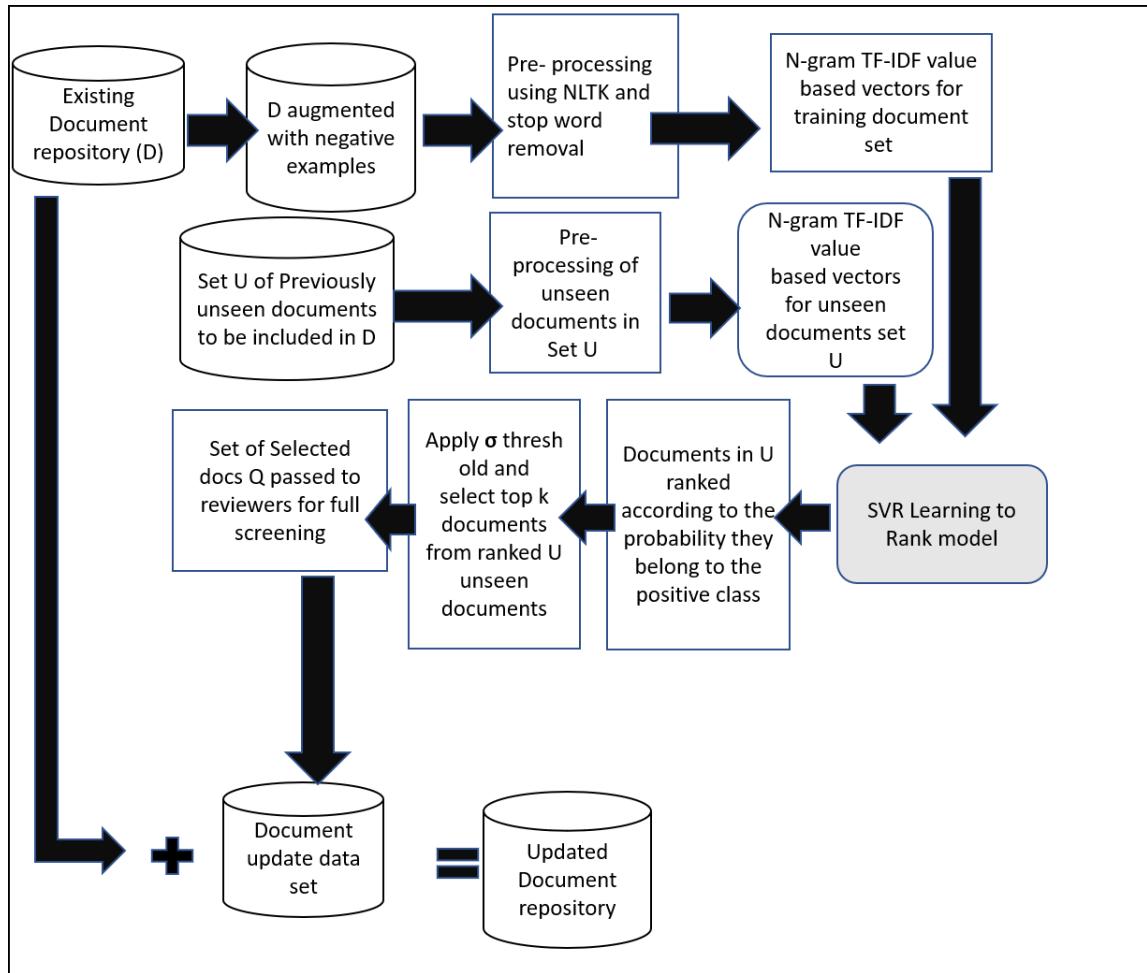


Figure 5.2: The Proposed CN Approach

in a document). Uni-grams, bi-grams and tri-grams were used with respect to the work presented here. The frequency for each identified n-gram was then determined using the well known Term-Frequency Inverse-Document Frequency (TF-IDF) metric [45]. The TF-IDF value for a n-gram w , $TFIDF(w)$, is calculated as shown in Equations 5.1, 5.2 and 5.3 where:

- TF is the term frequency.
- u is a document in U .
- w is a n-gram.
- $|u|$ is the size of the document $u \in U$ in terms of words.
- $|U|$ is the size of document collection U in terms of documents.

A frequency threshold value θ was then used to decided which n-grams should be included in the feature vector representation. A default value for θ was used for this purpose as calculated using Sklearn³ Python library.

$$TFIDF(w) = TF(w) \times IDF(w) \quad (5.1)$$

$$TF(w) = \frac{\text{frequency count of } w \in u}{|u|} \quad (5.2)$$

$$IDF(w) = \frac{|U|}{\text{total number of } u \in U \text{ in which } w \text{ appears}} \quad (5.3)$$

The next stage was to use the feature vector representations to generate the desired ranking model (the stage shown in the right side of Figure 5.2). Support Vector Regression (SVR) was adopted for this purpose. SVR is the regression equivalent of classification Support Vector Machines [50]. The reasons for selecting SVR were as follows:

1. For large datasets in natural language processing, with respect to applications in document ranking, SVR has proven to give improved results similar to neural networks and in some cases better [24, 120].
2. SVR provides a better implementation compared to other machine learning algorithms when deployed at scale commercially, as seen in similar systems as reported in [52, 91].

³https://scikit-learn.org/stable/modules/generated/Sklearn.linear_model.SGDClassifier.html

Once the ranking model has been generated, as shown in lower part of Figure 5.2, it can be used to process set U and identify Q in order to update an existing CDD.

5.5 The GRAB-Rank Approach

This section presents the GRAB-Rank approach. As noted above, the novelty of the GRAB-Rank approach is that it uses a combination of two embeddings:

1. The well established “off-the-shelf” general purpose BERT word-embedding.
2. A domain specific knowledge graph embedding that captures the knowledge held in the knowledge graph.

The intuition underpinning the proposed GRAB-Rank approach was that if two document embeddings, generated in different ways, were concatenated together it would produce a better document embedding than if the embeddings were used in isolation. It should be noted here that, in the context of literature knowledge graphs, GRAB-rank is not the first to use document embeddings. However, in the initial stages of PhD (2019) there was no reference found in the literature in the context of “to the use of hybrid embeddings. Figure 5.3 gives a schematic of the GRAB-Rank approach, note that it differs slightly from the generic LETOR framework presented in Figure 5.1. The main two differences between the generic LETOR framework and the GRAB-Rank LETOR framework are in (i) the pre-processing of the training set of documents D , and the candidate unseen test documents U ; and (ii) the BERT and knowledge graph embeddings generated for U used as an input to the SVR learning to rank model. The rest of the stages in the GRAB-Rank LETOR process are the same as in generic LETOR process presented in Figure 5.1.

For the pre-processing stages, the default BERT tokenizer was used. Knowledge graph embeddings were then added to each of the BERT embeddings. To differentiate between the BERT embedding and knowledge graph embedding, we refer to a “left hand embedding” and a “right hand embedding” respectively. The remainder of this subsection is organised as follows. Further detail concerning the BERT embedding process is given in Sub-section 5.5.1, whilst further detail concerning the Knowledge Graph embedding process is given in Sub-section 5.5.2. In the context of medical application domains, it should also be noted here that Grab-Rank is not the first system to consider knowledge graph random walk embeddings [143, 146].

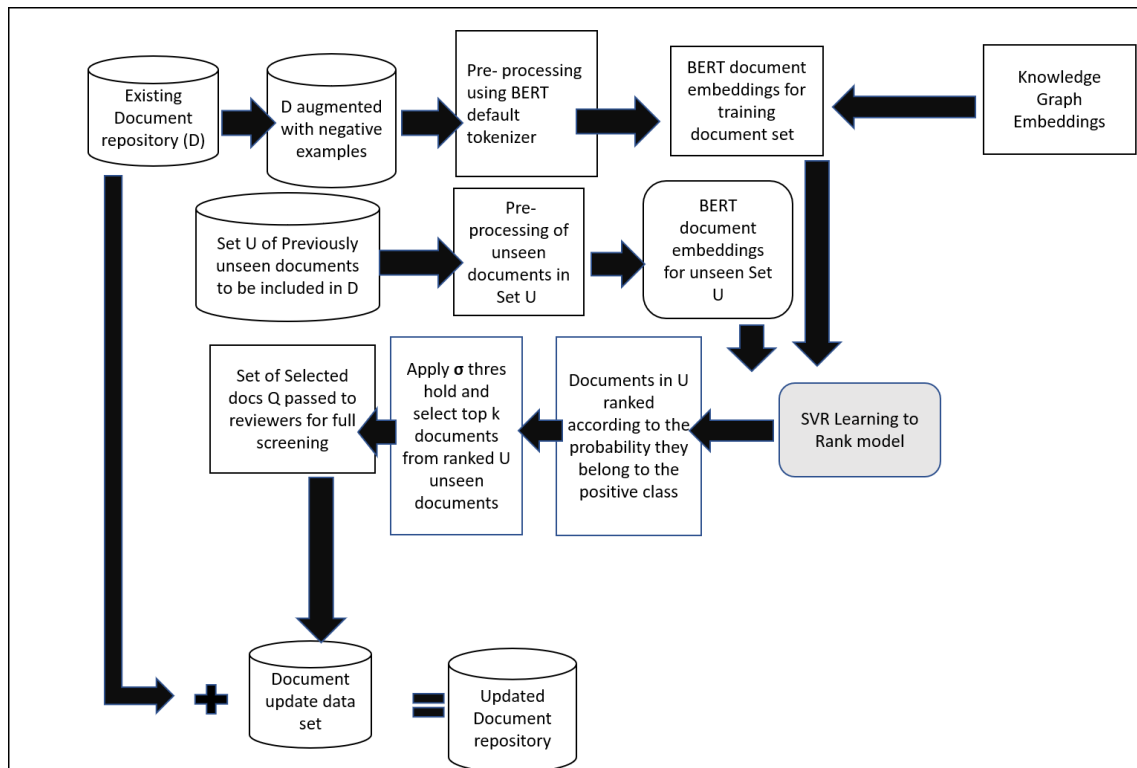


Figure 5.3: The GRAB-Rank LETOR process

5.5.1 BERT Embedding

The general purpose (left hand) document embedding incorporated into the proposed GRAB-Rank approach was a BERT embedding. BERT language models are generated using a transformer-based deep learning architecture first developed by the Google Brain Team [123] to address the shortcomings of Recurrent Neural Networks (RNNs). Transformer deep network learners dispense with the need for recurrence by replacing the recurrent layers used in RNNs with a multi-headed self-attention mechanism. The advantage is that the self-attention mechanism processes items in a sequence in parallel. The transformer has an encoder which reads an input sequence, and a decoder which produces an output sequence. BERT makes use of the encoder part of the transformer to generate word embedding models. As a result the language models produced by BERT are context-aware because they take into consideration words the precede and proceed each current word; unlike models such as GLOVE [99] where each word is represented using a single vector

regardless of context. Examples of studies that have utilized BERT word embeddings for document ranking to realise an improved performance when compared to traditional LETOR models can be found in [35, 88]. Using the GRAB-Rank approach the pre-trained embeddings from BERT were used to generate the left-hand general purpose embeddings that were then concatenated with the right-hand knowledge graph embedding (discussed in the following sub-section).

5.5.2 Knowledge Graph Embedding

The domain specific (right-hand) embedding in Figure 5.3 incorporated into the proposed GRAB-Rank approach was a literature Knowledge graph embedding generated using a random walk technique applied to a graph G . There has been some previous work where the idea of knowledge graph embeddings have been used for document ranking [73, 125]. The idea of a random walk was presented in [108]. From this previous work there is evidence that knowledge graph embeddings can achieve improved results over more traditional document ranking models such as statistical document ranking models [67, 73, 136]. The proposed knowledge graph embedding was generated by first identifying a set of random walks (paths) $\mathbf{R} = \{R_1, R_2, \dots\}$, where each $R_i \in \mathbf{R}$ is of the form $[v_1, v_2, \dots, v_k]$ where: v_j is a concept vertex in G , rw is the length of the walk and no two values for v_j are the same. In other words, each $R_i \in \mathbf{R}$ links a sequence of concept vertices in a given literature knowledge graph. Each random walk across G can be conceptualised as a “sentence” to which natural language processing (NLP) techniques can be applied. Each sentence can be represented using (say) a “bag of words” model or a “skip gram” model [60].

With respect to the work presented in this thesis, the Node2vec framework was used to generate random walk embeddings. This was chosen because it has proven to be effective for document ranking in recent works [94, 108, 131]. Using the Node2vec framework two strategies can be adopted for generating random walks:

1. Breadth-First Sampling (BFS)
2. Depth-First Sampling (DFS)

The BFS strategy generates random walks in parallel from a given start vertex v_j and considers all immediate neighbours of v_j before moving on to the immediate neighbours plus one, and so on until a pre-specified maximum random walk length rw is reached. Using the DFS strategy random walks from a given start vertex v_j are generated in sequence rather

than in parallel. The BFS strategy was adopted with respect to the propose GRAB-Rank approach for generating knowledge graph embeddings because it had been proven to generate effective embeddings as reported in [37, 112].

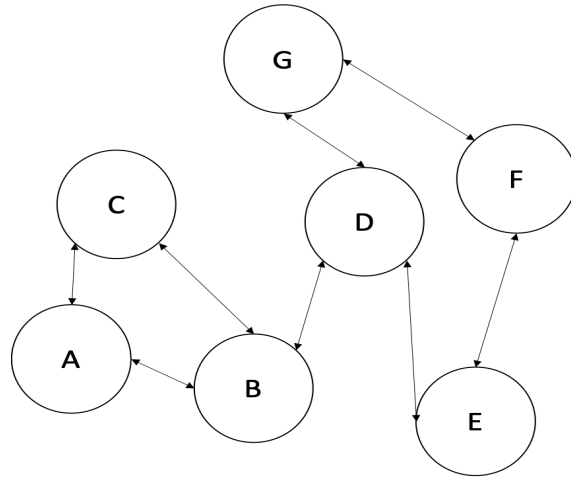


Figure 5.4: Example graph for explaining the concept of random walk

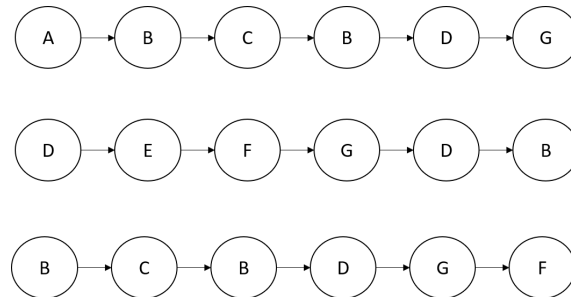


Figure 5.5: Some example random walks generated from the graph given in Figure 5.4

As noted earlier, a random walk can be conceptualised as a sentences. This can be illustrated using the example graph given in Figure 5.4. The figure shows seven vertices identified by the upper case letters $\{A, B, C, D, E, F, G\}$, and eight bi-direction edges $\{AB, AC, BC, BD, DE, DG, EF, FG\}$. A vertex is then randomly selected from which the random walk will commence. Let us assume vertex B is selected and that $rw = 6$. Then, using DFS, a neighbour to B is selected randomly, possible candidates are A, C and D . Then the next neighbourhood node is selected and so on. This process is repeated until a length rw is arrived at. Cycles are permitted. The process is then repeated. Some example

random walks generated from the graph given in Figure 5.4 are presented in Figure 5.5. Thus we have the “sentences”: *ABCBDG*, *DEFGDB*, *BCBDGF* and so on. Note that the same set of sentences can be generated using BFS.

To summarize, the Node2vec algorithm uses random walks to generate sentences from nodes in a knowledge graph. Once the sentences from the graph are generated using random walks, the Node2vec algorithm inputs the sentences into the Word2vec skip-gram model and retrieves the node embeddings. Word2vec skip-gram is a neural network based language model that is capable of generating embeddings and in this case vertex (node) embeddings. More details about the Word2vec skip-gram model for generating embeddings can be found in [39]; a variant of Word2vec is considered in Section 6.3.1 of Chapter 6. It is to be noted that node embeddings from a knowledge graph, have external knowledge in them, which can prove to be helpful in increasing the effectiveness of a document ranking technique [87].

5.6 Evaluation

The previous two sections have presented two approaches for maintaining knowledge graph represented CDDs: the CN approach and the GRAB-Rank approach. This section presents the outcomes of the comparative evaluation of the two approaches. For the evaluation both were implemented using the Python Programming Language. All experiments were run using a NVidia K80 GPUs kaggle kernel. The objectives of the evaluation were as follows:

1. To determine the most appropriate value for σ , the “cut-off” threshold for separating Q from U , both for the CN and the Grab-Rank algorithm.
2. In the case of the GRAB-Rank approach, to determine most appropriate value for rw , the random walk length.
3. Also in the case of the GRAB-Rank approach, to determine whether the hypothesis that using a combined BERT and Knowledge Graph embedding was more effective than when using a single embedding was correct.
4. To determine the comparative effectiveness of the proposed approaches.
5. To estimate the time saving gained using the proposed approaches in comparison with conducting a manual systematic review.

Recall that for the CN approach frequency threshold θ was used to determine, whether a n-gram should be included in the feature vector representation or not. For the selection of the CN frequency threshold θ , a parameter called *maxdf* in the Sklearn Python library, was also used ⁴. This parameter automatically selects the default value for θ . Thus, experiments were not conducted to consider alternative values for θ .

The rest of this section is organised as follows. Some detail concerning the evaluation data sets used are presented in Sub-section 5.6.1. The evaluation metrics used are discussed in Sub-section 5.6.2. The evaluation results with respect to the individual evaluation objectives listed above are then discussed in Sub-sections 5.6.3 to 5.6.7.

5.6.1 Evaluation Data Set

For the evaluation presented in this section, the following data sets were used:

ORRCA-400: A small dataset which could be easily inspected. It comprised 400 abstracts (hence the name), 200 positive examples (examples to be included in Q) and 200 negative examples (examples not to be included in Q).

ORRCA-Update 2015: The 2015 ORRCA update collection comprised of 11,099 abstracts, 1302 positive examples and 9797 negative examples.

ORRCA-Update 2017: The 2017 ORRCA update collections, comprised of 14,485 abstracts, 1027 positive examples and 13458 negative examples.

Details concerning these data sets were presented previously in Chapter 3. Recall that in each case each document consisted of a title and an abstract. The titles and abstracts were pre-processed so that punctuation and stop words were removed. A statistical overview of these data sets was presented in Table 3.1 in Chapter 3.

5.6.2 Evaluation Metrics

To compare the effectiveness of the two proposed approaches precision and recall were used. LETOR algorithms are usually analysed using metrics such as Mean Average Precision (MAP) or Mean Reciprocal Rank (MRR). However, for the work presented in this chapter we are interested in whether a given document (abstract) $u \in U$ should be included in

⁴https://scikit-learn.org/stable/modules/generated/Sklearn.feature_extraction.text.TfidfVectorizer.html

Q or not. In other words we have a binary classification problem. Hence, precision and recall were adopted with respect to the evaluation presented here, metrics normally used in the context of classification. This approach has been adopted with respect to earlier work where LETOR has been expressed in terms of a binary classification problem [25, 91]

Precision and recall are calculated as shown in Equations 5.4 and 5.5 where:

1. TP is the number of true positives
2. FP is the number of false positives
3. FN is the number of false negatives.
4. TN is the number of true negatives.

A true positive (TP) is an outcome where the model correctly predicts the positive class (u_i should and was included in Q). A true negative (TN) is an outcome where the model correctly predicts the negative class (u_i should not and was not included in Q). A false negative (FN) is an outcome where the model incorrectly predicts the negative class (u_i should have been included in Q , but was not included).

$$Precision = TP / (TP + FP) \quad (5.4)$$

$$Recall = TP / (TP + FN) \quad (5.5)$$

5.6.3 Determination of The Most appropriate Value for sigma

As noted earlier, a threshold σ was used to define a “cut-off decision threshold” for documents to be included in Q (from which the top k will be selected); thus a value of between 0 and 1 ($0 \leq \sigma \leq 1$). If $\sigma = 1$ all $u \in U$ will be included in Q . If $\sigma = 0$ no documents will be included ($Q = \emptyset$). The value for σ therefore needs to be chosen appropriately so that false positives are minimised so as to limit the resource required for the human intervention. The intuition here is that for a LETOR algorithm to effectively rank documents, the maximum number of relevant documents should be present at the top of a ranked document list. For 100% of relevant documents to be within the top k of documents within the ranked document list requires a recall of 1, with a compromise of values for precision. In other words there is a trade off between the two. An optimum point for the value of σ can

thus be identified using a precision-recall curve. Note that in any typical precision-recall curves for document ranking, if the recall increases the precision decreases.

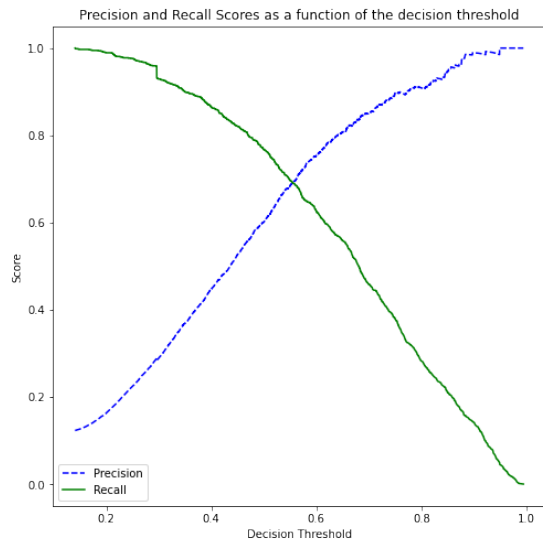


Figure 5.6: Precision-recall curve for ORRCA 2015 dataset using CN approach, decision thresholds σ given on the x-axes, and precision and recall on the y-axes

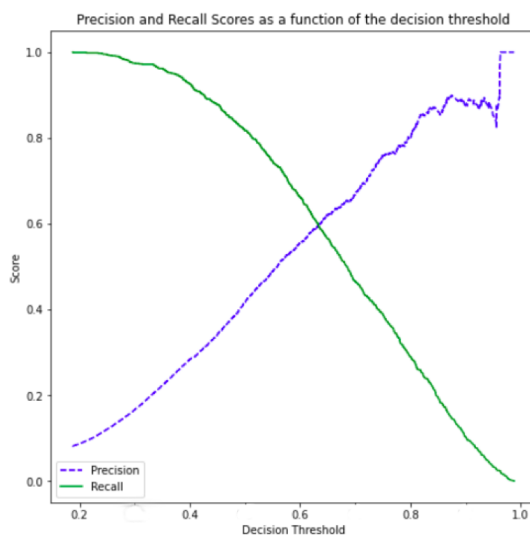


Figure 5.7: Precision-recall curve for ORRCA 2017 dataset using CN approach, decision thresholds σ given on the x-axes, and precision and recall on the y-axes

For the evaluation reported here a sequence of experiments was conducted using a range of values for σ from 0 to 1 increasing in steps of 0.2. The precision and recall results obtained were plotted using a precision-recall curve. Experiments were conducted using the ORRCA-Update 2015 and ORRCA-Update 2017 data sets. The resulting precision-recall curves, using both the CN and GRAB-Rank approaches, are given in Figures 5.6, 5.7, 5.8 and 5.9. Potential values for σ are plotted on the x-axis and the precision-recall score on the y-axis. Selecting the value of σ is a compromise between the number of relevant documents and the number of irrelevant documents in Q with respect to individual values of σ . Considering the CN approach first, Figures 5.6 and 5.7 show the relevant precision and recall curves. Inspection of Figure 5.6 indicates that if $\sigma = 0.4$ is chosen, this would result in identifying 97% of the relevant abstracts in terms of recall. From Figure 5.7 it can be seen that if $\sigma = 0.4$ was chosen this would result in identifying 97% of the relevant abstracts in terms of recall. It was considered that a “safety margin” should be included, and that a loss of 3% to 5% was acceptable trade-off when ranking clinical trial documents

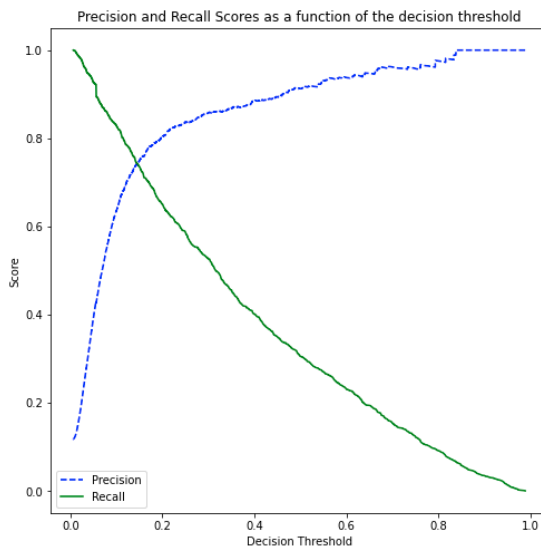


Figure 5.8: Precision-recall curve for ORRCA 2015 dataset using GRAB-Rank approach, decision thresholds σ given on the x-axes, and precision and recall on the y-axes

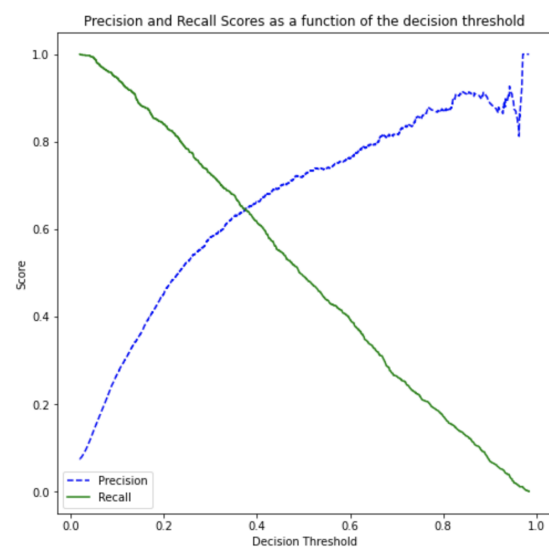


Figure 5.9: Precision-recall curve for ORRCA 2017 dataset using GRAB-Rank approach, decision thresholds σ given on the x-axes, and precision and recall on the y-axes

(a view supported by the individuals responsible for updating the ORRCA CDD). Hence it was concluded that $\sigma = 0.4$ was the most appropriate value in the case of CN algorithm (for getting 97% of the relevant abstracts). Figures 5.8 and 5.9 present the precision-recall curves obtained using the GRAB-Rank approach, for the ORRCA 2015 and 2017 update data sets respectively. As before, the potential values for σ are plotted on the x-axis and the precision-recall score on the y-axis. From Figure 5.8 it can be seen that $\sigma = 0.1$ should be selected (from the x-axis) if the goal is to achieve a recall of 97%. From Figure 5.9 it can be seen that a value of $\sigma = 0.2$ results in a recall of 97%. Again a loss of 3% to 5% in recall was considered an acceptable trade-off when ranking clinical trial documents. Hence it was concluded that in the case of GRAB-Rank algorithm $\sigma = 0.1 - 0.2$ would be the best possible value in order to achieve a recall of almost 97%.

5.6.4 Determination of The Most appropriate Value for the random walk length

The investigation of the use of the most appropriate value for the parameter rw , the random walk length, was conducted by considering the knowledge graph embedding in isolation. Recall that it was noted in Section 5.5.2 that the higher the value for rw the more concepts will be included in the knowledge graph embedding. Experiments were conducted using a range of values for rw from $rw = 1$ to $rw = 5$ incremented in steps of 1. The experiments were conducted using the following:

1. The ORRCA-400 dataset used as a prototype literature knowledge graph (400 examples)
2. The concatenation of the ORRCA-Update 2015 and ORRCA-Update 2017 datasets, referred to simply as the *ORRCA-Update* data set. The concatenated dataset represents the whole literature knowledge graph ($11099 + 14485 = 25584$ example).

A training testing split of 80:20 was used as adopted with respect to other similar work [57]. The knowledge graph embeddings were generated using the random walk technique presented in Section 5.5.2. A value of $\sigma = 0.2$ was used for these experiments because it had been found to be the most appropriate value for the GRAB-Rank algorithm according to the experiments and observations reported on in Sub-section 5.6.3 above. Recall that $\sigma = 0.2$ is the percentage of documents in U to be included in Q . So in the case of the ORRCA-400 dataset this will be 80, and in the case of the ORRCA-Update dataset this will be 5116. The experiments follow the same method of cross-validation as mentioned in [101]. Table 5.2 presents the precision and recall values obtained in each case (best results in bold font). From the table, the first thing that can be observed is that there were differences between the two sets of results. It was conjectured that these differences were due to the small number of examples in the ORRCA-400 dataset (400 examples), compared to number of examples in the ORRCA-Update data set (25584 examples), which meant that there were significant differences in the overall size of the two knowledge graphs. The ORRCA-400 dataset had 1600 vertices and the ORRCA-update knowledge graph had 102,336 vertices. Using 100 random walks, a greater coverage would be obtained for a small knowledge graph (as in the case of the ORRCA-400 knowledge graph), than for a large knowledge graph (as in the case of the ORRCA-Update knowledge graph). Closer

Table 5.2: The performance of GRAB-Rank using a range of values for rw , the random walk length (best results in bold font).

rw	ORRCA-400		ORRCA-Update	
	Precision	Recall	Precision	Recall
1	0.68	0.42	0.17	0.59
2	0.75	0.46	0.24	0.83
3	0.74	0.46	0.26	0.87
4	0.73	0.45	0.26	0.86
5	0.74	0.46	0.26	0.86

inspection of Table 5.2 indicates that better recall results were obtained using the ORRCA-Update knowledge graph than ORRCA-400 knowledge graph; and that better precision results were obtained using the ORRCA-400 knowledge graph than the ORRCA-Update knowledge graph. However, as noted above, we wish to maximise recall. From the table the larger knowledge graph supports this.

Further inspection of Table 5.2 indicates that $rw = 1$ did not perform well in both cases. Again this is likely to have been because of the poor coverage in both cases. For the smaller ORRCA-400 knowledge graph best results were obtained using $rw = 2$. As the value of rw increases, in the context of the ORRCA-400 knowledge graph, there is little improvement. It is conjectured that this is because that coverage (quality) of embedding has reached a “plateau” beyond which no further improvement is gained. The same phenomena can be observed with respect to the ORRCA-Update knowledge graph except with $rw = 3$. An argument can therefore be made that $rw = 3$ is the most appropriate value.

5.6.5 Combined BERT and Knowledge Graph Embedding Versus Single Embedding

To determine whether the hypothesis that a combined BERT and Knowledge Graph embedding would produce a more effective ranking, an ablation study was conducted by comparing the effectiveness of using the combined GRAB-Rank embedding with that of using BERT and Knowledge Graph embeddings on their own. The comparison was conducted using the ORRCA 400 data set (prototype knowledge graph) and the ORRCA update data sets (the larger ORRCA literature knowledge graph). For the experiments a value of $\sigma = 0.2$ was used because of the conclusions made in Section 5.6.3. A value of $rw = 3$ to was used as it was concluded to be the most suitable value according to the

Table 5.3: Comparison, in terms of precision and recall, using the GRAB-Rank approach, and using BERT and knowledge graph embeddings in isolation (best results in bold font).

LETOR technique	ORRCA-400		ORRCA-Update	
	Precision	Recall	Precision	Recall
GRAB-Rank with SVR	0.81	0.50	0.26	0.88
BERT embeddings only with SVR	0.76	0.47	0.23	0.80
Knowledge graph embeddings only with SVR	0.75	0.46	0.26	0.87

experiments discussed in Section 5.6.4. A training test split of 80 : 20 was again used. The experiments here follow the same method of cross-validation as mentioned in [101].

The metrics used were again precision and recall. Support Vector Regression (SVR) was again used to generate probabilities as to whether each $u_i \in U$ should be included in Q . The results are presented in Table 5.3, best results are indicated in bold font. From the table it can clearly be seen that the combined GRAB-Rank approach outperforms the usage of BERT and knowledge graph embeddings when used in isolation. It was conjectured that the differences in magnitude between the results obtained using the ORRCA 400 data set and the ORRCA update data sets was due to the significant difference in size between the two data sets; 400 examples versus 25584 document examples.

5.6.6 Comparative effectiveness

To analyse the comparative effectiveness of the two proposed approaches, the CN Approach and the GRAB-Rank approach, their operation was compared with a “classical document ranking system”. Namely an approach founded on the Okapi BM25 ranking function presented in [51, 121] and presented in Sub-section 2.4.1 of the literature review (Chapter 2). Influenced by earlier results $\sigma = 0.2$ and $k = 3$ were again adopted with respect to the evaluation reported on in this sub-section. Table 5.4 presents the results obtained in terms of precision and recall. From the table it can be seen that the proposed GRAB-Rank hybrid approach produced the best performance for both the ORRCA-400 and ORRCA update datasets.

5.6.7 Time Savings Gained

To evaluate the time savings gained (in comparison to a manual systematic review) using the proposed approaches effort recall curves were constructed (Figures 5.10 to 5.13). In

Table 5.4: The performance of GRAB-Rank and CN approaches in comparison with the Okapi BM25 approach (best results in bold font).

LETOR technique	ORRCA-400		ORRCA-Update	
	Precision	Recall	Precision	Recall
GRAB-Rank with SVR	0.81	0.50	0.26	0.88
CN algorithm	0.79	0.49	0.07	0.49
Okapi BM25 ranking	0.53	0.33	0.16	0.54

these graphs “effort” is presented on the x-axis and “recall” on the y-axis. Effort is defined as the number of candidate abstracts (thus both relevant and irrelevant abstracts) to be screened as a percentage of the total number of abstracts. Recall is defined as the percentage of relevant documents identified in the ranked document list. For the evaluation the ORRCA 2015 and 2017 update datasets were used as they represent the real manual systematic review updates. For calculating the *hours saved* after automating the updating of ORRCA CDD (if compared to the manual systematic review), we must consider the number of *total candidate articles* for that systematic review dataset and the *relevant articles identified*. Hence the following equation was used for calculating the hours saved:

$$hours\ saved = \frac{total\ candidate\ articles - relevant\ articles\ identified}{60} \quad (5.6)$$

The 60 used in the equation earlier was to convert minutes to hours as the assumption made was that the screening rate of a domain expert is one abstract per minute.

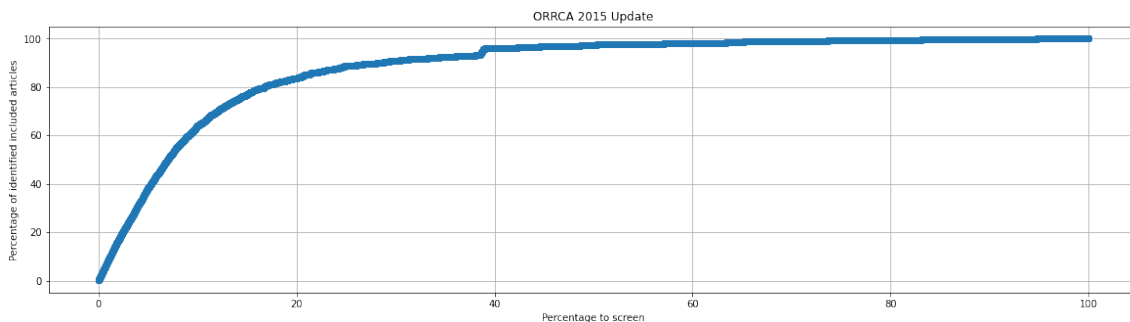


Figure 5.10: The 2015 update effort-recall curve using CN algorithm

Figures 5.10 and 5.11 present the effort-recall curves for the CN approach in the context

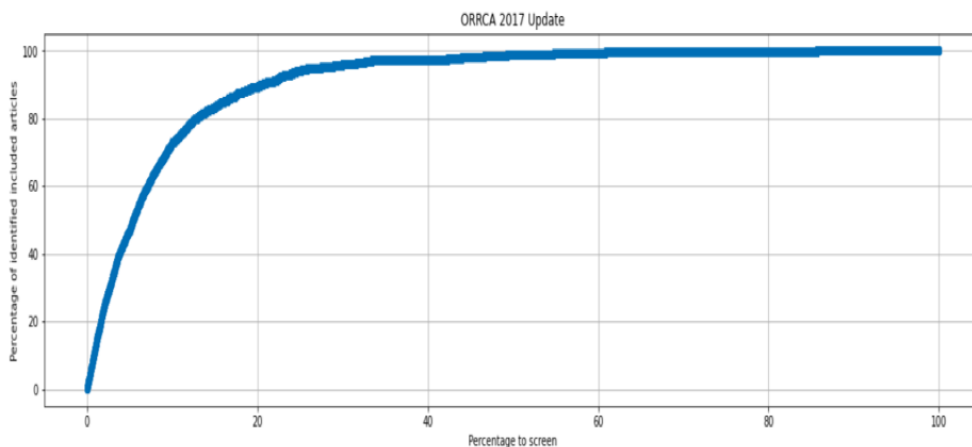


Figure 5.11: The 2017 update effort-recall curve using CN algorithm

of the ORRCA 2015 and ORRCA 2017 update data sets. From Figure 5.10, it can be seen that in order to get 97% of relevant abstracts (on the y-axis), we would need to screen the top 40% of the candidate abstracts (on the x-axis). Estimating the time saved by automating the manual screening process, an assumption was made that the screening rate of a domain expert is one abstract per minute. Assuming a best value for σ of 0.4 for the CN algorithm (see discussion in Sub-section 5.6.4), and considering the 2015 ORRCA update data set, this will result in 6660 ($11099 - 4439 = 6660$) abstracts being excluded, equating to a time saving of $6660 \div 60 = 111.0$ hours (assuming an experienced screener for the abstract screening process). Similarly, from Figure 5.11 and with respect to the 2017 ORRCA Update, and selecting best value for σ of 0.4 for the CN algorithm, ($14485 - 5794 = 8691$) 8691 abstracts would be excluded equating to a time saving of $8691 \div 60 = 144.0$ hours. Hence it can be concluded that when using the CN algorithm, savings of between 111 and 144 persons hours could be made (when compared to manual systematic review process) respectively.

Figure 5.12 and 5.13 present the corresponding effort-recall curves for the GRAB-rank algorithm with respect to the ORRCA 2015 and 2017 update datasets. From earlier work (see Sub-section 5.6.3), reported above, $\sigma = 0.2$ was considered to be the most appropriate value when using the GRAB-Rank approach. A σ value of 0.2 means that the top twenty percent of the documents should be screened in a ranked document list. From Figure 5.12

(the ORRCA 2017 update dataset), we would need to screen 20% of the candidate abstracts using $\sigma = 0.2$, in order to get 97% of relevant abstracts (on the y-axis). This equates to almost 2897 abstracts. Similarly, for Figure 5.13 (the ORRCA 2015 Update dataset), using the $\sigma = 0.2$, we would need to screen almost 20% of the candidate abstracts in order to get 97% of relevant abstracts (on the y-axis). This equates to 2219 documents. Again assuming a screening rate of a domain expert as one abstract per minute, and considering the 2017 ORRCA update, for the GRAB-Rank algorithm, referring to Figure 5.12, this will result in 11588 ($14485 - 2897 = 11588$) abstracts being excluded, equating to a time saving of $11588 \div 60 = 193$ hours (assuming an experienced screener for the abstract screening process). Similarly for the GRAB-Rank algorithm referring to Figure 5.13 and with respect to the 2015 ORRCA Update and using $\sigma = 0.2$ value, 8880 abstracts would be excluded ($11099 - 2219 = 8880$) equating to a time saving of $8880 \div 60 = 148$ hours. Hence it can be concluded for the Grab-Rank algorithm, that savings of 148 and 193 persons hours could be made (when compared to manual systematic review process) respectively.

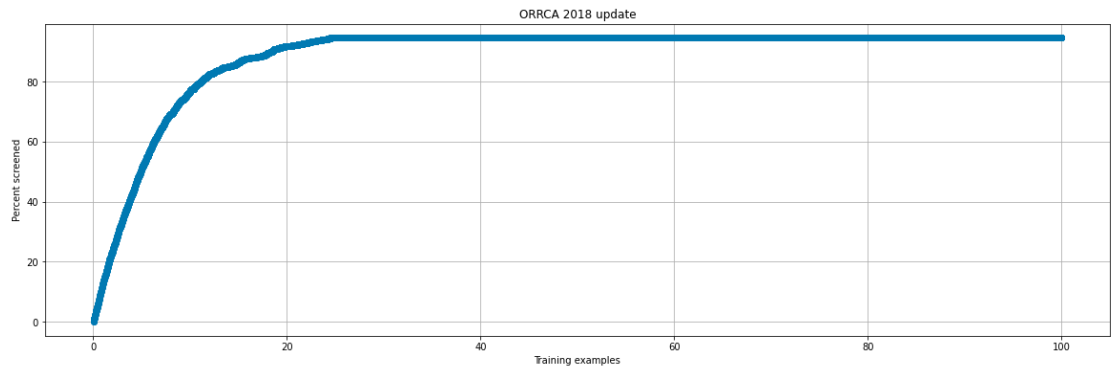


Figure 5.12: The 2017 update effort-recall curve using GRAB-Rank algorithm

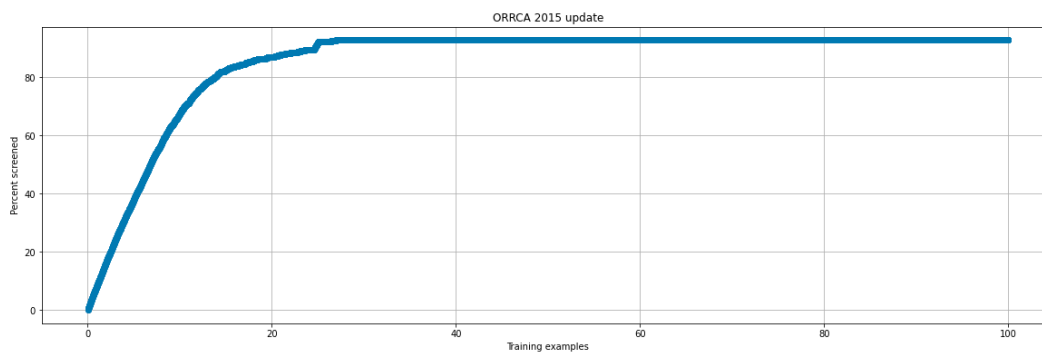


Figure 5.13: The 2015 update effort-recall curve using GRAB-Rank algorithm

5.7 Summary

This chapter has presented an overview of two proposed approaches to maintaining knowledge graph represented CDDs: the CN approach and the GRAB-Rank approach. The first used a feature vector representation and the second an embedding representation. The novel element of the GRAB-Rank approach was that it combined two embeddings, a general purpose embedding and a domain specific embedding. For the first BERT was adopted and for the second a bespoke knowledge graph embedding was adopted. The knowledge graph embedding was generated using a graph random walk of length rw . The fundamental idea was to rank a set of potential documents to be considered and select the top rw . A range of experiments were conducted from which it was established that the most appropriate values for σ and rw were $\sigma = 0.2 - 0.4$ and $rw = 3$. Out of the two approaches, and comparing with BM25 as a baseline, and BERT and knowledge graph embeddings used in isolation, the proposed GRAB-Rank approach was found to be the most effective. It was estimated that by using the GRAB-Rank approach a time saving of 148 to 193 persons hours could be obtained over the manual systematic review process currently often used to update CDDs. In the next chapter, techniques to query literature knowledge graph will be considered.

Chapter 6

Literature Knowledge Graph Query Resolution

6.1 Introduction

In the previous two chapters, Chapters 4 and 5, the knowledge graph construction and updating approaches were presented, and evaluated using the Online Resource for Recruitment research in Clinical trials (ORRCA) data set. The central theme to this chapter is the utilisation (querying) of literature knowledge graphs constructed and maintained as described in the foregoing two chapters. Recall that the central motivation for the work presented in this thesis is to make it convenient for researchers to identify previous work, in the form of scientific papers, relevant to their domain of study. At the same time, the work presented in this chapter is directed at deriving an answer to Subsidiary Questions three and six from Chapter 1:

[SQ 3]: *Given an existing CDD, represented as a literature knowledge graph, how can this knowledge graph be queried so as to retrieve relevant documents?*

[SQ 6] *Can the embeddings implicit within a literature knowledge graph be used to provide an answer to a query in the context of document retrieval?*

This chapter provides answers to SQ3 and SQ6 using a proposed approach to literature knowledge graph querying.

The fundamental idea, as with the majority of document query resolution mechanisms, is to represent a given query and the documents in the knowledge graph in a manner that will allow the documents to be ranked, according to some relevance measure, with respect to the query. From the literature [3, 76, 136], there are a number of ways in which the query and documents in the literature knowledge graph can be represented. The current trend is to use a word embedding approach of some kind. An embedding is a learned text representation whereby each term in a given corpus is represented using a numerical vector, also referred to as an embedding [2, 85, 98]. Once a word embedding has been generated, a document embedding can be obtained by averaging the individual word embeddings. An intuitive way of generating document embeddings is by averaging the word embeddings of all the words in a document. Given a single document, each word in the document would be represented by a single word embedding. A simple arithmetic operation of averaging can be performed on these word embeddings to obtain a single document embedding. Recently, many methods involving deep learning, have been used to generate embedding models, which have been used to effectively score query-document pairs [6, 21, 78]. Three different embedding models were considered with respect to the work presented in this chapter:

1. Continuous Bag of Words (CBOW) [80].
2. Bidirectional Encoder Representations from Transformers (BERT) embedding [21].
3. SciBERT embedding [6].

Each of these word embeddings, once generated, were combined with the random walk knowledge graph embedding presented earlier in Chapter 5. The intuition here was that by using knowledge graph embeddings additional semantic knowledge would be added which might help to increase the effectiveness of query resolution when applied to CDDs represented as literature knowledge graphs [115, 114, 129].

The remainder of this chapter is organised as follows. The relevant notations and symbols used are given in Table 6.1 below.

6.2 The Query Resolution Process

The proposed high level query resolution process is illustrated in Figure 6.1. The input, top of the Figure, is a query Q and the document collection $\mathbf{D} = \{D_1, D_2, \dots\}$ referenced by

Symbol	Symbol Definition
Q	A set of Queries, $\mathbf{Q} = \{Q_1, Q_2, \dots, Q_k\}$
D	A set of Documents, $\mathbf{D} = \{D_1, D_2, \dots\}$ referenced by the knowledge graph
d	A set of terms in a document D_i where $\mathbf{d} = \{d_1, d_2, \dots, d_n\}$
Q'	A cleaned, pre-processed query Q_i
D'	A cleaned and pre-processed version of s D , $\mathbf{D}' = \{D'_1, D'_2, \dots\}$
t_i	Term t_i (in a query or a document)
el_i	The Left hand Embedding with respect to the Schematic given in Diagram 6.1
er_i	The Right hand Embedding with respect to the Schematic given in Diagram 6.1
EQ	Query Embedding
ED	A Set of Document Embeddings, $\mathbf{ED} = \{ED_1, ED_2, \dots\}$
k	The number of documents to be selected from the top of a given ranked list.
c	Context words in the CBOW language model
t	Target words in the CBOW language model
t	Number of Transformer Blocks in the BERT language model
h	Number of hidden layer size in the BERT language model
a	Number of attention heads in the BERT language model
S_{cos}	The cosine similarity between two given vectors
ap_{jk}	The average precision metric used for evaluation purposes

Table 6.1: Symbol table for Chapter 6, Utilisation of Literature Knowledge Graphs using Query Based Document Ranking

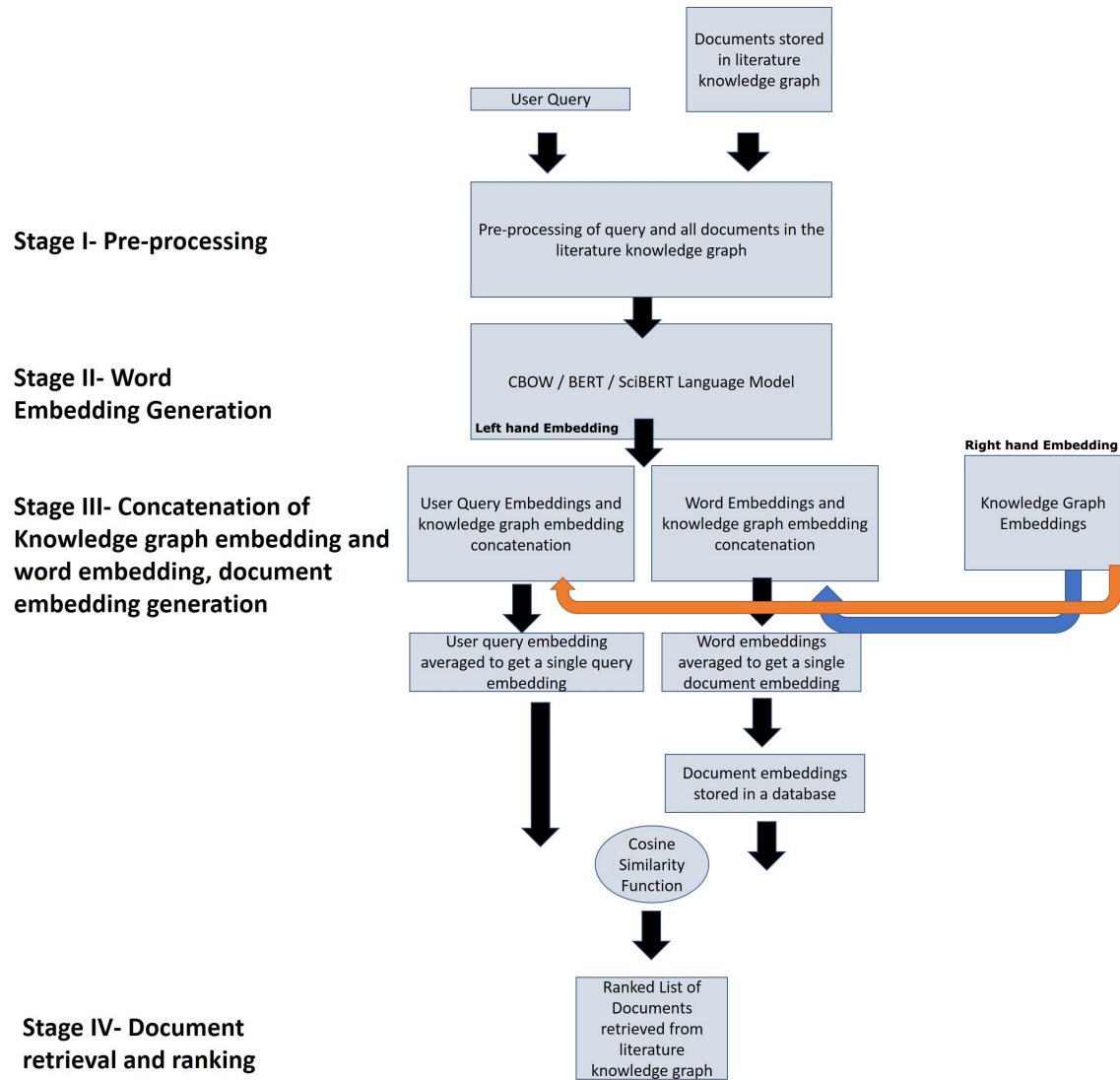


Figure 6.1: Schematic of the adopted literature knowledge graph query resolution process.

the knowledge graph. The query shown in this query-resolution process is expressed in the form of a natural language sentence. A bespoke query language, such as SPARKQL, was not used. The reason for this was that how queries were expressed using the pre-existing ORRCA system used as a test bed throughout thesis. Some examples of biomedical queries used in the form of a natural language sentence can be found in [53, 54]. Note also that the adopted query format does not provide for logical operators such as “OR”, again since this was because of a design decision made with the pre-existing ORRCA interface. It should be noted that the whole document collection need only to be processed once for document embedding generation, after which the generated document embedding is stored (provided the document collection remains unchanged). Each document $D_i \in \mathbf{D}$ comprises n terms. $D_i = \{d_1, d_2, \dots, d_n\}$.

From Figure 6.1 it can be seen that the query resolution process comprises four stages:

Stage I: Pre-processing of the documents

Stage II: Word embedding generation for both the query and each of the document in the knowledge graph

Stage III: Concatenation of knowledge graph embeddings generated from the knowledge graph and pre-trained word embeddings.

Stage IV: Similarity Measurement between query embedding and document embedding, generated above from Stage III, using the cosine similarity measure, and the ranking of documents using cosine similarity score

Although query-resolution can be done using various strategies, the above “query-resolution process” is founded on the concept of text-matching between a query embedding and a document embedding. Similar work on text matching can be found in [46, 49, 103]. In this context, text matching using cosine similarity measures results in a ranked list of abstracts which is a different concept from the pointwise learning-to-rank model used for knowledge graph maintenance as described in the foregoing chapter. The first stage in the above “query-resolution process”, Stage I, comprises the pre-processing of the inputs Q and \mathbf{D} . The nature of the pre-processing will be dependent on the nature of the language model used. Using the CBOW model all documents and queries were pre-processed by removing punctuation and stop words to give Q' and \mathbf{D}' . For stop word removal the Python Natural Language Tool Kit (NLTK) was used². Both BERT and SciBERT required

²<https://www.nltk.org/>

that the input is tokenized. The BERT default tokenizer was used for this purpose which applied pre-processing prior to tokenization. Note that the BERT tokenizer requires special “classification” (CLS) and “seperating” sentence (SEP) tokens to be added to the start and end of each sentence. The result was a cleaned version of Q and \mathbf{D} , Q' and $\mathbf{D}' = \{D'_1, D'_2, \dots\}$.

The next stage, Stage II, was to generate the desired word embeddings for Q' and each document $D'_i \in \mathbf{D}'$. Note that a word embedding is expressed as a numeric vector of a constant length. As indicated in Figure 6.1, three alternatives were considered, CBOW, BERT and SciBERT, for word embedding generation. Details concerning CBOW, BERT and SciBERT embedding generation are given in the following section, Section 6.3.

In Stage III, the word embeddings from Stage II are concatenated with Random Walk Knowledge Graph embeddings, generated as described in Chapter 5, and then averaged to generate document embeddings. The idea encapsulated in this stage is that a better word embedding can be produced if two embeddings are concatenated together, as opposed to using the embeddings individually. To distinguish between pairs of embeddings, we refer to a “left hand embedding” and a “right hand embedding” as indicated in Figure 6.1. The result is two embeddings for each term (word) t_i contained in a query or a document, el_i (left) and er_i (right). For the work presented here the right hand embedding was the random walk knowledge graph embedding presented in Chapter 5. For the left hand embeddings, as stated above, and as indicated in Figure 6.1, three alternatives were considered (CBOW, BERT and SciBERT) for query and document embedding. It is to be noted that for each term t_i (in a query or a document), we get an embedding (numerical vector) e_i such that $e_i = el_i + er_i$ (the $+$ infix operator used here should be interpreted as an append operator). The desired document and query embeddings are then generated by averaging the content of the constituent word embeddings. The result is a query embedding EQ and a set of document embeddings $\mathbf{ED} = \{ED_1, ED_2, \dots\}$. Once the document embeddings have been generated they are stored.

The final stage, Stage IV, involves determining the similarity between EQ and each document in \mathbf{ED} to produce a ranked list as indicated in Figure 6.1. Embedding similarity measurement is discussed in Section 6.4, however, for the evaluation presented later in this chapter cosine similarity was used. The generated cosine similarity values were then used to create a ranked list from which the top k could be selected. It is to be noted that the query-resolution defined here, is the similarity matching between a query embedding and a document embedding. Similar work on text matching can be found in [14, 103, 109]. The

document ranking defined in Chapter 5 was a static pointwise document ranking technique, which doesn't involve any similarity matching between documents.

6.3 Contextual and Non-contextual Embedding Systems

This section gives some background concerning the three “left hand” embeddings considered:

1. Continuous bag of word embedding (CBOW)
2. Bi-directional transformer embedding (BERT)
3. Scientific Bi-directional transformer embedding (SciBERT)

Recall that an embedding is essentially a learnt numerical vector used to represent a word in a document. Document embeddings can be generated by averaging or concatenating word embeddings. The ability to represent documents using some form of embedding has a wide range of applications, including query based document retrieval. In [60, 79], the authors mention the use of embeddings for semantic relatedness, paraphrase detection, document retrieval and ranking.

There are various techniques for generating document embeddings. We can categorise these techniques as being either non-contextual or contextual techniques. CBOW is an example of the first, and BERT and SciBert of the second. Both non-contextual and contextual embeddings can be used in the context of transfer learning. The concept of transfer learning is the idea of learning a model for one domain with the intention that it be applied in related domains. The CBOW, BERT and SciBERT embeddings adopted with respect to the work presented in this thesis were used in this manner. The embeddings were generated using one domain with the intention of using them in the context of a curated document database domain (particularly the ORRCA domain). The remainder of this section is structured as follows. Non-contextual document embedding is considered in Sub-Section 6.3.1 with a focus on CBOW. Contextual document embedding is then considered in Sub-Section 6.3.2 with a focus on BERT and SciBERT.

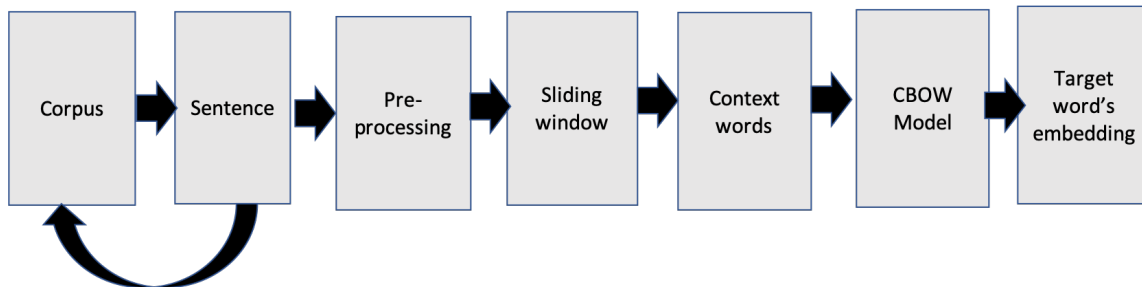


Figure 6.2: A schematic diagram for CBOW model word embedding generation

6.3.1 Non-Contextual Embedding Systems (CBOW)

Non-contextual embeddings do not take into account the context of individual words within a document. The advantage offered by non-contextual techniques is that they are easier to implement than contextual techniques; especially when deployed as part of a large scale document retrieval system. This section gives an overview of the Continuous Bag-Of-Words (CBOW) non-contextual embedding approach. A popular class of non-contextual embedding is what is known as Word2Vec embeddings. As the name suggests, the input to a Word2Vec model is a word and the output is a vector (an embedding). There are two common Word2Vec models, skip-gram and CBOW. With respect to the work presented in this thesis only the CBOW model was considered, because of its effectiveness [80, 149]. CBOW operates in an iterative manner, on each iteration producing an embedding of an input word. Once we have the embeddings CBOW is no longer required. This makes CBOW embeddings easy to use.

A high level overview of CBOW is presented by the Schematic presented in Figure 6.2. The figure should be read from left to right. The left most side in the figure shows a corpus of documents. The CBOW model takes one sentence from the corpus at a time, iteratively. Given each sentence in the corpus, a sliding window is used to take a set of words as the input from each sentence in the document. The sliding window will slide over a sequence of words from each sentence in an iterative manner (for the whole document). The goal of CBOW model is to generate word embeddings for each target word t . For each iteration in this word embedding generation process, a sequence of input words (using the sliding window) is taken from the corpus. These words from the sliding window are then fed into the CBOW model as seen in Figure 6.2. The right side of the figure shows the output of

the CBOW model, an embedding for a target word.

6.3.2 Contextual Embedding Systems (BERT and SciBERT)

Contextual embedding techniques take the context of surrounding words into account. The distinction between contextual and non-contextual embedding can best be illustrated 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-contextual embedding system would create the same word embeddings for the word “bank” regardless of its usage, whereas a contextual embedding system would generate different word embeddings depending on the context of the word “bank”. The coexistence of many possible meanings for a word is referred to as *polysemy*. From the foregoing it can be seen that context-informed word embeddings can be argued to produce better representations than non-context-informed word embeddings. Contextual models have shown great promise with respect to various NLP tasks [21, 75]. However, contextualized language models require significantly more computing resources than non-contextual models (such as CBOW). Contextual models are typically generated using some form of machine learning, usually either a LSTM or a transformer deep learner. An overview of the two contextual embedding systems considered in this thesis, BERT and SciBERT, is presented below, commencing with BERT.

BERT

BERT stands for Bidirectional Encoder Representations from Transformers [21]. It is bidirectional because it takes into account the context of words both before and after a given target word. As the acronym suggests, BERT is founded on the use of transformers, a deep learning model learnt using a process of attenuation [123]. As noted above, the primary distinction between non-contextual embedding systems, such as CBOW, and contextual embedding systems, such as BERT, is that non-contextual systems generate single unique embeddings (one for each word in the vocabulary), while contextual systems can generate more than one embedding for each word. In the case of BERT this is done using the location (index) of each word. Therefore, instead of individual words as in the case of

CBOW, BERT requires whole sentences as input. The “knock-on effect” of this is that the model needs to be retrained so that specific embeddings can be generated given a particular document corpus; not the case with respect to CBOW.

BERT expects input data in a specific format, with special tokens to mark the beginning (CLS) and separation/end of sentences (SEP). The given query Q and documents D in our knowledge graph thus had to be translated into this format. A tokenizer is available to achieve this; the BERT tokenizer. For the items in each tokenized sentence, BERT requires input IDs. BERT input IDs are a sequence of integers identifying each input token to its index number in the BERT tokenizer vocabulary. BERT models are usually defined by the number of transformer blocks (t), the number of hidden layers (h) and the number of attention heads (a). Two primary models were created by the BERT developers, BASE and LARGE, defined as follows:

- BASE: $t = 12$, $h = 768$ and $a = 12$.
- LARGE: $t = 24$, $h = 1024$ and $a = 16$.

Both were trained using words extracted from the BooksCorpus [148] and from the entirety of English Wikipedia. The BooksCorpus is a collection of 11036 books written by unpublished authors (around 74M sentences and 1G words). With respect to the work presented in this thesis the BERT BASE model was selected, because of the limited computational resource available, which meant that BERT LARGE could not be deployed at scale. A BERT model is essentially a deep neural network comprised of a number of layers. The authors in [21] identified that it helped to sum the last 4 layers of BERT. This was therefore also adopted with respect to the work presented here to generate EQ and **ED**.

SciBERT

As noted above, BERT was trained using words extracted from Wikipedia and BookCorpus. In other words BERT is a general, all-purpose, language model. It might thus be the case that it is not well suited to specialised domains such as query resolution with respect to the ORRCA domain-specific dataset. We have two options here:

1. Continuing to train BERT with domain-specific examples so as to create a more specialised, domain specific, language model.
2. Use an existing domain specific language model that fits the domain of interest.

The latter are created by using the BERT architecture to train a dedicated BERT model using a domain-specific corpus. There are many domain specific BERT models that are readily available. Well known examples include: FinBERT for financial services applications [139], BioBERT for biomedical applications [64], and SciBERT for biomedical and computer science applications. Inspection of the many domain specific BERT models that are available indicated that SciBERT might be well matched to the ORRCA clinical trials research domain. SciBERT was trained on a random sample of 1.14M scientific publications [3]. SciBERT word embeddings are therefore considered to be well-suited to representing scientific document content [6].

Implementation Details

For evaluating the proposed knowledge graph query resolution approach, all three variations using the above three document embeddings, were implemented using Pytorch [97], with the Hugging Face transformer library [130] and Scikit-learn library [61]. All the above models mentioned earlier in this section were specialised to the clinical trials domain using the Adam optimizer [59].

6.4 Similarity Measurement and Ranking

From the methodology presented in Section 6.2, and shown in Figure 6.1, the fourth stage in the proposed process was measuring the similarity between the query embedding EQ and each document embedding ED_i in ED . There are various mechanisms whereby the similarity between two embeddings can be calculated (recall that an embedding is simply a vector of numbers). Popular choices are dot product, Euclidean distance and Cosine similarity. Dot product similarity favours long vectors, which may skew the outcome and thus not be ideal for query resolution. Euclidean distance similarity measures the distance between the two points defined by two vectors. Cosine similarity measures the angle between the two vectors. These are not the same thing, the points defined by two vectors may be a long way apart, but the angle between the vectors may be small. Euclidean distance similarity is only significant where we wish to take the length of the vectors into consideration whereas cosine similarity is important when the angle between two vectors is important. Therefore, for the evaluation presented later in this chapter cosine similarity was adopted.

Cosine similarity, as the name suggests, is the cosine of the angle between two vectors x and y . This is the same as the inner product of the two vectors (normalized so that they both have a length of 1). Cosine Similarity (S_{cos}) is calculated as shown in Equation 6.1, where $x.y$ is the dot product between the two vectors. The notation $\|x\|$ ($\|y\|$) denotes the “norm” of x within the normalised vector space. A cosine similarity of 1 indicates an exact match.

$$S_{cos}(x, y) = \frac{x.y}{\|x\| \times \|y\|} \quad (6.1)$$

The documents in **ED** were ranked according to their cosine similarity scores. The top k could then be selected to be returned to the user. The value for k depends on the number of documents that the end user would like to be returned. For the evaluation presented later in this chapter $k = 5$ and $k = 10$ were considered.

6.5 Evaluation

This section presents the evaluation of the proposed query-based document retrieval from a literature knowledge graph approach. The challenge here was the lack of an appropriate test data set. A bespoke data set had thus to be created. To this end funding was obtained, from the UK Medical Research Council - National Institute for Health Research (MRC-NIHR) Trials Methodology Research Partnership¹, in a collaboration between the Universities of Liverpool, Leeds and Aberdeen (all in the UK).

The objective of the evaluation was to compare the operation of the considered embeddings (CBOW, BERT and SciBERT) coupled with the random walk knowledge graph embedding, and when used in isolation. The evaluation metrics used were Average Precision and Mean Average Precision (MAP). The experiments were conducted using NVidia K80 GPUs kaggle kernel GPUs. Each method was limited to accessing only one GPU for fair comparisons. The remainder of this section is organised as follows. Further detail concerning the evaluation dataset is presented in Sub-section 6.5.1. The dataset collection process is explained in Sub-section 6.5.2. The evaluation metrics used are considered in

¹Anna Kearney (PI and University of Liverpool), Frans Coenen (Co-I and University of Liverpool), William Cragg (Co-I and University of Leeds), Katie Gillies (university of Aberdeen, Co-I), Iqra Muhammad (Co-I and University of Liverpool), Amanda Roberts (Co-I and University of Liverpool), Paula Williamson (Co-I and University of Liverpool) January 2021. Using Machine learning with user feedback to improve ORRCA. Medical Research Council - National Institute for Health Research (MRC-NIHR) Trials Methodology Research Partnership. £20,000

further detail in Sub-section 6.5.3. The results obtained are then presented and discussed in Sub-section 6.5.4. The section is concluded with a discussion of the results obtained from an empirical study in Sub-section 6.5.5.

6.5.1 Evaluation Dataset

For the evaluation, a dataset of query-document pairs was collected along with relevance judgements using the funding obtained from MRC-NIHR. The dataset consisted of a set of 45 clinical trial queries. Each query was paired with the relevant and irrelevant documents. The relevance judgements were binary, ‘relevant’ or ‘not-relevant’ for each document corresponding to a query. A similar approach was adopted in [68, 65, 89]. This dataset was collected by a team of experts from:

1. The Health Services Research Unit at the University of Aberdeen
2. The Department of Health Data Science at the University of Liverpool
3. The Department of Public Health, Policy and Systems at the University of Liverpool

The querying was conducted using the existing ORRCA CDD, not the knowledge graph variation proposed in this thesis (because at this stage the new version of the CDD was not yet generally available), which was queried using a bespoke graphical interface which allowed the user to enter keywords or phrases. A fragment of the dataset is given in in Table 6.2. The left hand column gives the query (one or more keywords or phrases). The middle column gives a document title returned as a consequence of the query. The right hand column gives the reviewer’s view as to whether the document was relevant or not.

6.5.2 Evaluation Dataset Collection Process

The ORRCA dataset collection process is described in this section. The dataset was collected using human labelling done by a group of experts. This was because the author had no knowledge of the specialized domain of clinical trials. As noted above participants from various institutions took part in the data set collection process. The following process were adopted for the collection and relevance labelling process:

1. Experts from various institutes, were selected based on the criteria that they should be members of the TMRP Group. The TMRP group is a specialized group of clinical trials recruitment strategy experts.

Query	Document Title	Document Relevance
labour	Use of a cancer registry.	Not Relevant
labour	Parental preferences for neonatal resuscitation	Relevant
Postpartum haemorrhage	Marma therapy for stroke rehabilitation	Not-Relevant
Postpartum haemorrhage	VERA	Relevant
D2 and cancer does Participant Information Sheet and Consent Form in cancer trials affect recruitment	Response rates in a case-control study: effect of disclosure of biologic sample collection in the initial contact letter	Not Relevant
D2 and cancer does Participant Information Sheet and Consent Form in cancer trials affect recruitment	Randomised comparison of procedures for obtaining informed consent in clinical trials of treatment for cancer	Relevant
situational incapacity Recruitment research methods	Why do breast cancer patients decline entry into randomised trials and how do they feel about their decision later: a prospective, longitudinal, in-depth interview study	Relevant
situational incapacity Recruitment research methods	The role of therapeutic optimism in recruitment to a clinical trial in a peripartum setting: balancing hope and uncertainty	Not Relevant


Table 6.2: Fragment of ORRCA Query-document Evaluation Data set

The screenshot displays the ORRCA website's main search interface. At the top, a navigation bar includes links for Home, About, Search, Join the team, Ongoing Research, and Contact. Below this is the ORRCA logo and a 'Welcome to ORRCA' heading. A paragraph explains the project's goal: 'The ORRCA project (Online Resource for Research in Clinical trials) aims to bring together published and ongoing work in the field of recruitment and retention research into searchable databases.' A subsequent paragraph states: 'We are still updating the recruitment database with publications from 2018 and 2019 due to the large volume of literature in this field. Articles will be added periodically throughout the review process. An update of the retention database for 2020 and 2021 publications will be starting shortly. Authors are welcome to submit papers as soon as they are published. Retention database last update: 09/02/2022. Recruitment database last update 25/02/2021'.

The 'Search' section features two statistics: '4,393 Articles in recruitment database' and '1,338 Articles in retention database'. Below these are search buttons for 'Search Recruitment', 'Search Retention', and 'Search Both', along with an 'Advanced Search' link. A note at the bottom right reads: 'if you use the ORRCA databases in your research, please reference either of the following papers in any reports or publications:'.

On the right side of the page, there is an 'ORRCA Newsletter' sign-up button, a 'Social' section with a 'Follow @ORRCA_rct' button, and a 'Tweets' section showing a tweet from @ORRCA_rct dated Oct 25, 2021, which mentions a paper and a list of reviewers and collaborators.

Figure 6.3: A screenshot of the Online Resource for Research in Clinical trials (ORRCA) main search interface



Home About Search Join the team Ongoing Research Contact

Search

Search Filters
Please select the areas you wish to search with:

Article details

Author Title Abstract Journal Year Type of funding

Recruitment research

Domain(s) Evidence type Intervention Summary of findings Research methods Research outcomes Timing within host study

Retention research

Domain(s)

ORRCA now includes literature on participant recruitment and retention. Please select which database(s) you would like to search.

Recruitment Database **Retention Database**

Include Conference Abstracts

The database automatically uses an 'AND' search operator for multiple terms in the search filters below.

Use an 'OR' Search

Recruitment domain(s)

[View/Select Recruitment Domains](#) [View Recruitment Domains \(PDF popup\)](#) [Help](#)

Retention domain(s)

[View/Select Retention Domains](#) [View Retention Domains \(PDF popup\)](#) [Help](#)

[Search](#)

Figure 6.4: A screenshot of the Online Resource for Research in Clinical trials (ORRCA) advance search interface

2. The ORRCA main search interface presented in Figure 6.3 and ORRCA advance search interface presented in Figure 6.4 were modified for downloading CSV files of documents returned for each search query.
3. Each expert was then given a specific number of search queries and given a certain allocated time for relevance labelling.

The author collaborated with a number of experts so as to seek insight as to whether the relevance labels and number of documents retrieved were meaningful given the clinical trials domain.

6.5.3 Evaluation Metrics

The generated evaluation data set did not have a ground truth ranking, although the advocated approach presented in this chapter, given a query, produced a document ranking. Therefore metrics usually used with respect to “learning to rank” algorithms, such as Normalized Discounted Cumulative Gain (NDCG), could not be used. Thus Average Precision (AP) and Mean Average Precision (MAP) was considered to be the most appropriate evaluation metrics. The MAP at a rank k was calculated as follows (there are alternative formulations):

$$MAP(k) = \frac{1}{|Q|} \sum_{j=1}^{j=|Q|} AP_{jk} \quad (6.2)$$

Where: (i) k is a desired rank threshold, (ii) Q is the evaluation query dataset and (iii) ap_{jk} is the “Average Precision” at the rank threshold k (up to the rank threshold k) for a query j . The average precision, AP , at rank k for a query j is calculated as follows:

$$AP_{j,k} = \frac{1}{m} \sum_{i=1}^{i=k} p_{ji} \text{ if document at } i \text{ is relevant} \quad (6.3)$$

Where: (i) m is the number of relevant documents returned, and (ii) p_{ji} is the ranked precision for query j at rank i . Ranked precision is defined as the fraction of relevant documents for a query q_j retrieved from the the total number of documents retrieved at (up to) rank i . Ranked precision is calculated as shown in the following equations, 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 .

$$p_{ji} = \frac{tp_{ji}}{tp_{ji} + fp_{ji}} = \left(\frac{\text{relevant}}{\text{retrieved}} \right) \quad (6.4)$$

Note that p_{ji} will have a value between 0 and 1, the nearer to 1 the better. For the evaluation presented later in this chapter $k = 5$ and $k = 10$ were used; thus MAP values for the top 5 and top 10 documents were generated. This was because the average number of documents returned per query was never greater than 20, given the evaluation data set used. For the results presented in the following sub-section the “average” precision and “mean average” precision at rank k were used, indicated by the notations $AP@k$ and $MAP@k$. Note that an Average Precision of 0.0 occurs when no relevant documents are identified, and an Average Precision of 1.0 occurs when all relevant documents are identified, with respect to a given test query.

6.5.4 Results and Discussion

This section presents the Average Precision (AP) at $k = 5$ and $k = 10$ results with respect to the evaluation of the proposed querying mechanisms. For each of the 45 clinical trial queries in the ORRCA query-document dataset the results are presented as follows.

1. The results obtained using CBOW embeddings combined with the KG random walk embedding approach, and CBOW embeddings when used in isolation, are presented in Table 6.3
2. The results obtained using BERT embeddings combined with KG random walk embedding approach, and BERT embeddings when used in isolation, are presented in Table 6.4
3. The results obtained using Sci-BERT embeddings combined with KG random walk embedding approach, and Sci-BERT embeddings when used in isolation, are presented in Table 6.5.
4. The results obtained using Knowledge Graph Random Walk embeddings when used in isolation are presented in Table 6.6.

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 6.3: $AP@k$ results for combined CBOW and random walk embeddings, in comparison with CBOW 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 6.4: $AP@k$ results for combined BERT and random walk embeddings, in comparison with 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

Table 6.5: $AP@k$ results for combined SciBERT and random walk embeddings, in comparison with Sci-BERT used in isolation

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

Table 6.6: $AP@k$ results for Random Walk embeddings 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

Table 6.7: $MAP@k$ Table for BERT, SciBERT and CBOW when combined with Random Walk embeddings, and when in isolation

In each of the tables, Table 6.3 to 6.6, results in bold show the queries that performed significantly better than other queries. Query numbers 31, 32, 33 and 34 produced the best results from all the query searches considered. This could be attributed to the fact that the number of keywords in these queries, on average, was greater compared to the rest of the queries. It is conjectured that queries with a greater number of keywords tend to achieve better results compared to those with fewer keywords as seen in similar works recorded in [62, 113]; this makes logical sense.

Table 6.7 presents a summary of the results obtained using the Mean Average Precision at k ($MAP@k$) for all the embedding techniques. In the table best results are highlighted in bold. Inspection of the table indicates that combined CBOW and random walk embeddings produced the best results and random walk embeddings on their own also produced good results. It was conjectured that this was because the CBOW embedding training vocabulary, although “general” in nature, was more suitable to the ORRCA application domain than in the case of BERT and SciBERT. Among the experiments where CBOW, BERT, SciBERT and Knowledge Graph Random Walk embeddings were used in isolation for query resolution, the Random Walk embeddings produced better results (in terms of $MAP@k$). Knowledge Graph Random Walk embeddings used in isolation were also found to performed better than BERT and SciBERT embeddings when combined with random walk embeddings. It was conjectured that for datasets such as ORRCA external knowledge from a knowledge graph is helpful in getting a high number of relevant documents hence Knowledge Graph Random Walk embeddings performed better overall. However, when coupled both BERT and SciBERT embeddings this produced a negative affect. From Table 6.7 it can be seen that the recorded $MAP@k$ values were relatively lower for the BERT and SciBERT embeddings when used in isolation. It was thus concluded that this was because the BERT embeddings were more suitable for shorter queries; with respect to

the ORRCA dataset, the majority of the queries are relatively longer in length [93].

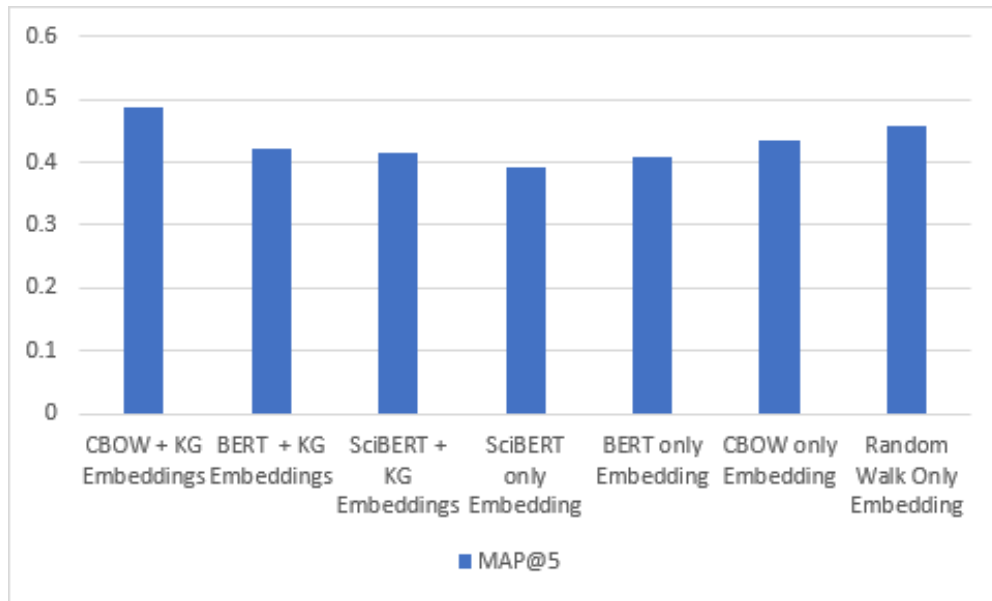


Figure 6.5: A bar chart showing the recorded $MAP@5$ values for the document embedding techniques considered

The results from Table 6.7 are presented in bar graph in Figures 6.5 and 6.6. The figures indicate the distribution of the recorded $MAP@5$ and $MAP@10$ values across all embedding techniques. In the figures the document embedding techniques for query resolution are plotted on the x-axis, whereas the values for $MAP@5$ and $MAP@10$, as appropriate, are plotted on the y-axis from 0 upwards incrementing in steps of 0.1. From Figure 6.5, which gives the $MAP@5$ results, it can again be seen that best results were obtained using CBOW embeddings coupled with Random Walk embeddings. The overall distribution for the MAP values, shows that the MAP values were usually in the range of 0.4 to 0.5. This range of values indicates that most of the document embedding techniques considered were partially successful in retrieving documents.

Returning to Tables 6.3 to 6.6 it is useful to investigate the spread of Average Precision values so as to get an alternative indicator of the operation of the different approaches considered. The distribution of the Average Precision values, $AP@5$ and $AP@10$, for the different approaches, are shown in bar graph form in Figures 6.7 to 6.20 (using the information given in Tables 6.3 to Table 6.6). Bar graphs are presented for: (i) CBOW

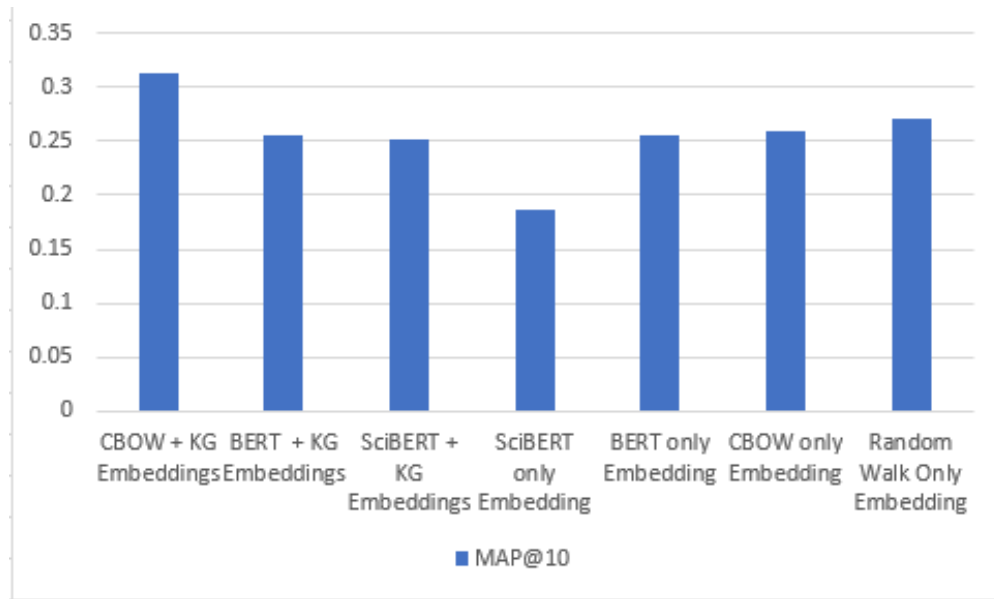


Figure 6.6: A bar chart showing the recorded $MAP@10$ values for the document embedding techniques considered

and Random walk embeddings, (ii) CBOW only embeddings, (iii) BERT and Random walk embeddings, (iv) BERT only embeddings, (v) SciBERT and Random walk embeddings, (vi) SciBERT only embeddings and (vii) Knowledge Graph Random walk only embeddings. For each bar graph Average Precision values are listed on the x-axis (incremented in steps 0.1) and the number of associated queries on the y-axis. A summary of the information contained in the bar charts is given below:

1. CBOW and random walk embedding combined (Figures 6.7 and 6.8: From Figure 6.7 (bar graph for $AP@5$), it can be seen that the value of $AP = 0.4$ has the highest number of search queries. The same figure also shows that the second highest number of search queries is for $AP = 0.6$. In addition to this, from Figure 6.8 (bar graph for $AP@10$), it can be seen that the value of $AP = 0$ has the highest number of search queries and the second highest search queries is for $AP = 0.4$. Recall that a value of $AP = 0$ means that such search queries failed to retrieve any relevant documents.
2. CBOW in isolation (Figures 6.9 and 6.10: From Figure 6.9, $AP@5$, it can be noted that the highest number of search queries is for $AP = 0$ and the second highest number of search queries is for $AP = 1$. From Figure 6.10, $AP@10$, it can be seen

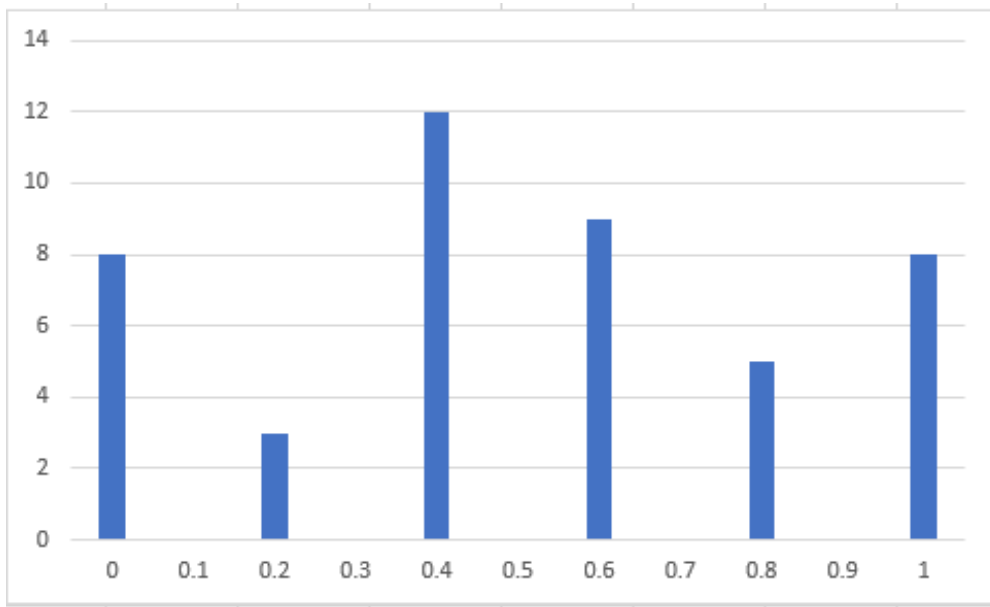


Figure 6.7: Bar graph showing number of queries against $AP@5$ values when using CBOW+RandomWalk document embeddings

that the highest number of search queries was associated with $AP = 0$ and the second highest with $AP = 0.3$.

3. BERT and random walk embedding combined (Figures 6.11 and 6.12): From Figure 6.11, $AP@5$, it can be seen that the $AP = 0$ has the highest number of search queries associated with it, whereas the second highest number of search queries in this case was for $AP = 0.9$. In the case of Figure 6.12, $AP@10$, the highest number of search queries was also associated with $AP = 0$ and second highest number of search queries for $AP = 0.9$.
4. BERT in isolation (Figures 6.13 and 6.14): From Figure 6.13, $AP@5$, it can be seen that $AP = 0$ had the highest number of search queries associated with it, and $AP = 0.6$ the second highest. From Figure 6.14, $AP@10$, it can be seen that the highest number of search queries was associated with $AP = 0$ and the second highest with $AP = 0.7$.
5. SciBERT and random walk embedding combined (Figures 6.15 and 6.16: From Figure 6.15 (bar graph for $AP@5$), it can be observed that the highest number of search

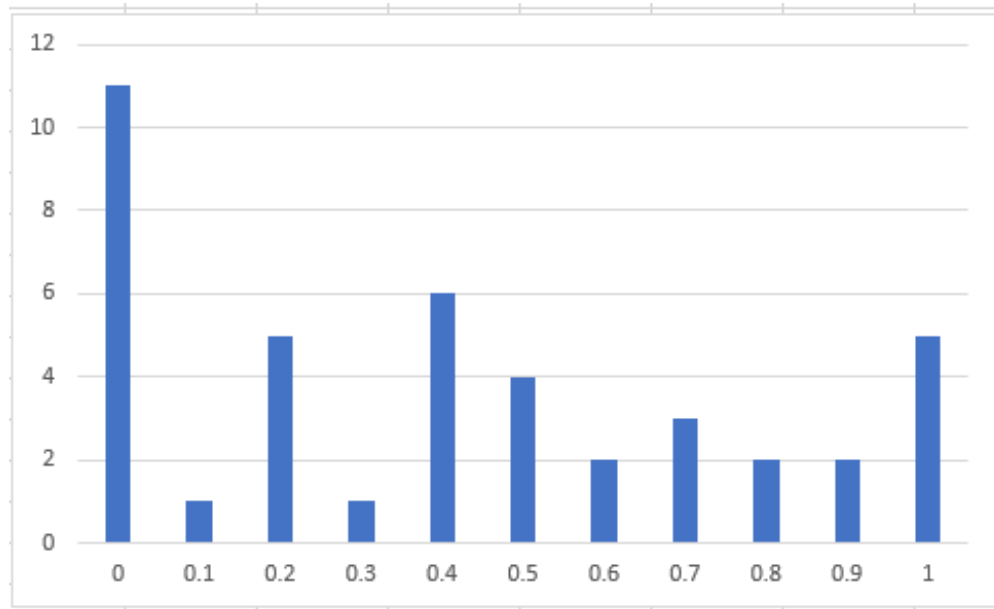


Figure 6.8: Bar graph showing number of queries against $AP@10$ values when using CBOW+RandomWalk document embeddings

queries was associated with $AP = 0$, meaning that in these cases no relevant documents were returned. Whereas for the same figure, the second highest number of search queries was associated with an $AP = 0.4$. For Figure 6.16 (bar graph for $AP@10$), the highest number of search queries had a value of $AP = 0$, whereas the second highest number of search queries is for $AP = 0.1$.

6. SciBERT in isolation (Figures 6.17 and 6.18): From Figure 6.17, $AP@5$, the highest number of search queries are associated with $AP = 0$ and the second highest number of search queries with $AP = 1$. Similarly, Figure 6.18, $AP@10$, demonstrates that the highest number of search queries was associated with $AP = 0$ and the second highest with $AP = 0.1$.
7. Random walk in isolation (Figures 6.19 and 6.20): Figure 6.19, $AP@5$, shows that the highest number of search queries was associate with $AP = 0$ and the second highest with $AP = 0.2$. Figure 6.20, $AP@10$, shows that the highest number of search queries was associated with $AP = 0$ and the second highest with $AP = 0.3$

From the foregoing analysis, it can be concluded that for the majority of the embedding

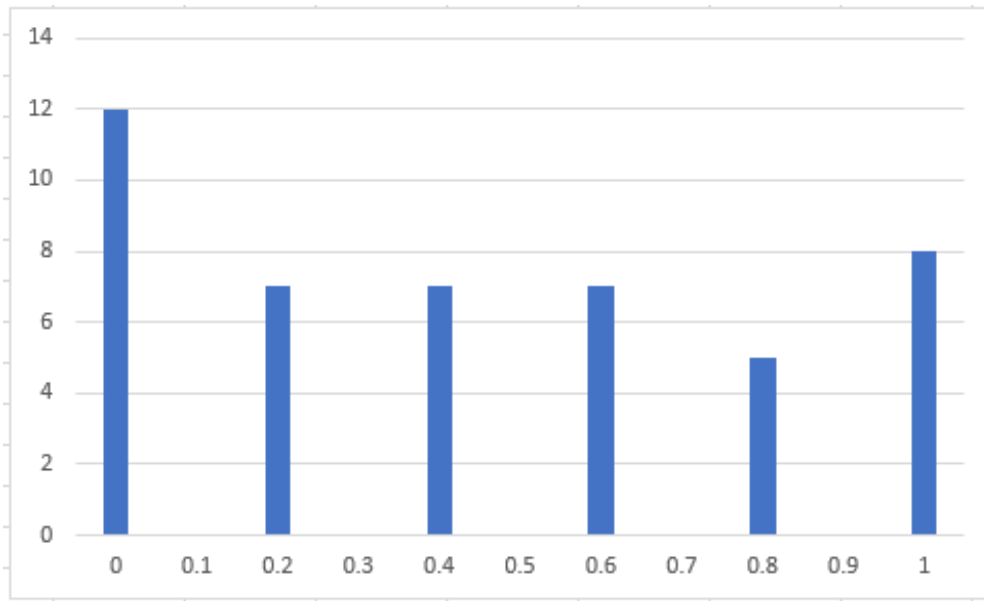


Figure 6.9: Bar graph showing number of queries against $AP@5$ values when using CBOW only document embeddings

techniques used above for query resolution, on average 12/45 search queries (both for $AP@5$ and $AP@10$) failed at retrieving relevant documents. Secondly, the second highest number of search queries (both for $AP@5$ and $AP@10$ bar graphs) varied in terms of average precision for each document embedding technique. The overall trend in the bar graphs is that for most search queries, the average precision was between 0.4 and 0.9 meaning that they at least retrieved some relevant articles.

6.5.5 Empirical study for knowledge graph query resolution

To obtain a better understanding of the operation of the proposed knowledge graph query resolution mechanism, an empirical study was conducted by gathering information regarding experience of using the proposed approach. The idea was to select a number of queries and present the results to selected ORRCA end users, domain experts selected from the Department of Health Data Science at the University of Liverpool. The study was motivated by the observation that query resolution would result in the following categories of document:

1. Documents identified by the proposed system and also identified using the ORRCA

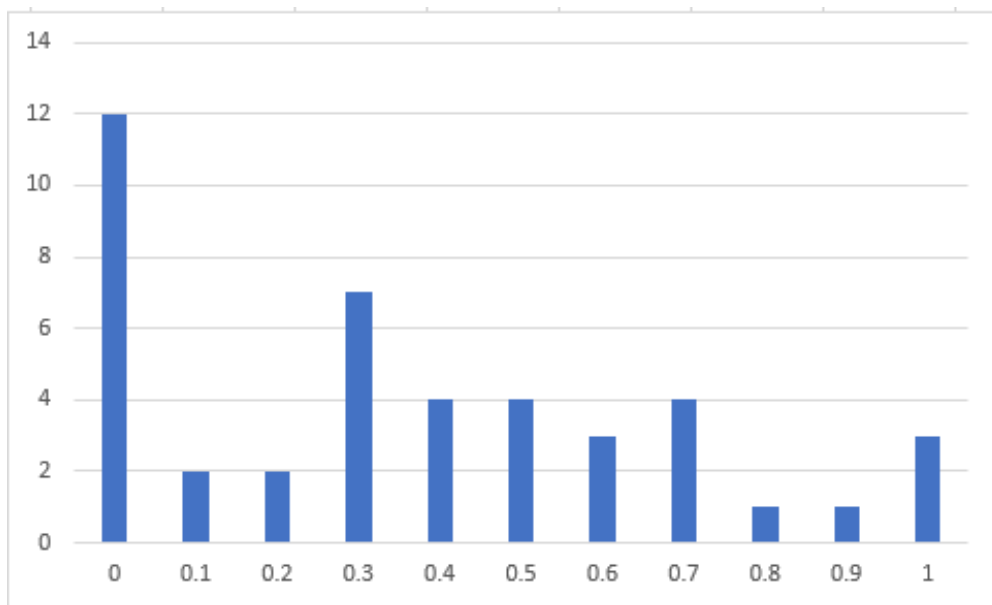


Figure 6.10: Bar graph showing number of queries against $AP@10$ values when using CBOW only document embeddings

search function, the set D_{po} . The po stands for relevant documents returned by both proposed search system (p) and ORRCA keyword matching system (o).

2. Documents identified by the proposed system and not identified using the ORRCA search function, but which should have been identified by the ORRCA search function, the set D_{pr} . The pr stands for documents returned only by the proposed search system (p) that are relevant (r).
3. Documents identified by the proposed system and not identified using the ORRCA search function, which should not have been identified by the proposed system, the set D_{pn} . The pn stands for the documents that are returned only by the proposed search system (p) that are not relevant (n).

The sets of interest here are the sets D_{pr} and D_{pn} ; the content of D_{po} can be readily established from the ORRCA query-document pairs data set. The sets D_{pr} and D_{pn} collectively form the set of false positives, the set D_{fp} ($D_{fp} = D_{pr} \cup D_{pn}$). To demonstrate the advantages of the literature knowledge graph approach, the focus of the work presented

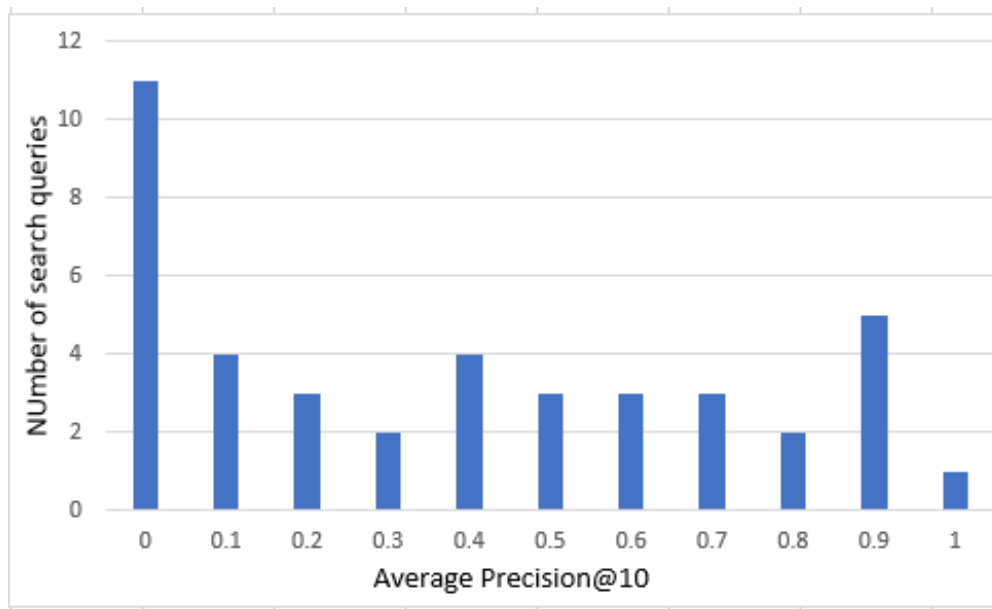


Figure 6.11: Bar graph showing number of queries against $AP@5$ values when using BERT+RandomWalk document embeddings

in this thesis, the ideal result would be a substantial number of items in D_{pr} and for D_{pn} to be empty.

For the experiment the “CBOW and Random walk embeddings” approach was used with k set to 10. A set of four queries were selected randomly from the ORRCA query-document pairs dataset identified by the key words: (i) Facebook, (ii) Bereaved, (iii) Palliative and (iv) Obesity. Recall that each query in the query-document pairs dataset has multiple documents associated with it, with their relevance manually labelled by domain experts in each case (as described in Sub-section 6.5.2). For each of the selected queries the sets D_{po} and D_{fp} were generated. The set D_{fp} was then presented to the selected end user domain experts for allocation to D_{pr} and D_{pn} . The results obtained are shown in Table 6.8. From the table it can be seen that the proposed search system with CBOW+ Knowledge Graph random walk embedding was able to identify relevant documents not identified by ORRCA search system for the search query “facebook” and “Palliative”, whereas for the search query “obesity” and “bereaved” it was not able to identify relevant documents.

From table 6.8 it can thus be concluded for two queries, the proposed knowledge graph query resolution system successfully identified relevant documents, not identified on the

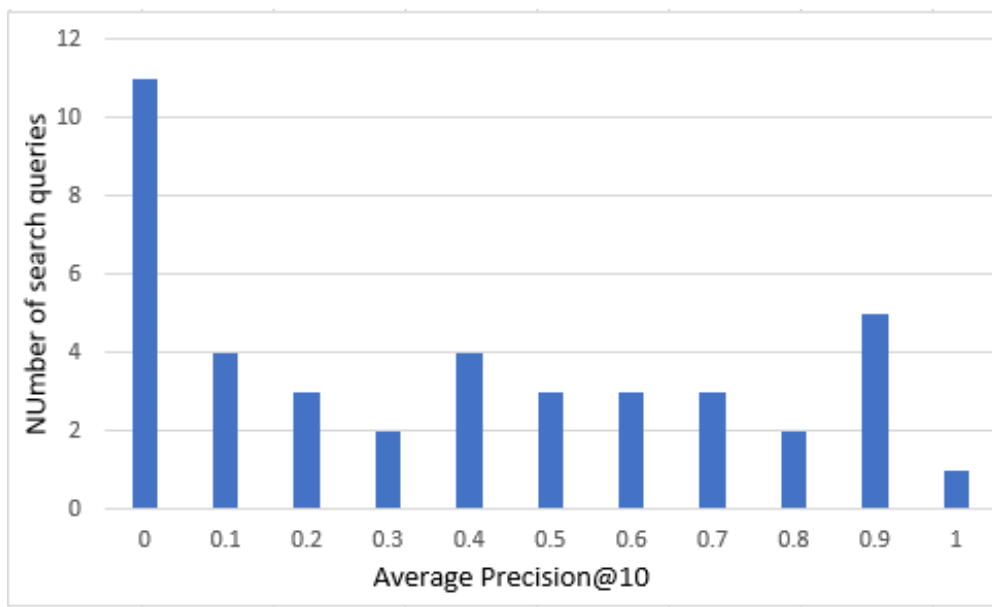


Figure 6.12: Bar graph showing number of queries against $AP@10$ values when using BERT+RandomWalk document embeddings

Query	Number of documents		
	D_{po}	D_{pr}	D_{pn}
Facebook	5	3	2
Bereaved	1	0	9
Palliative	7	3	0
Obesity	10	0	0
Total	23	6	11

Table 6.8: Results from empirical study

existing system. The possible reason for this is that semantic matching signal play an important role in the success of query resolution. The proposed *CBOW + Knowledge Graph random walk* document embeddings contain semantic knowledge sourced from the knowledge graph in the form of random walk knowledge graph embeddings and that resulted in more successful query resolution.

Figure 6.21 shows the overall performance of knowledge graph query resolution mechanism in terms of relevance score with respect to the empirical study. In this figure, the relevance scores returned by the proposed query resolution mechanism are plotted on the

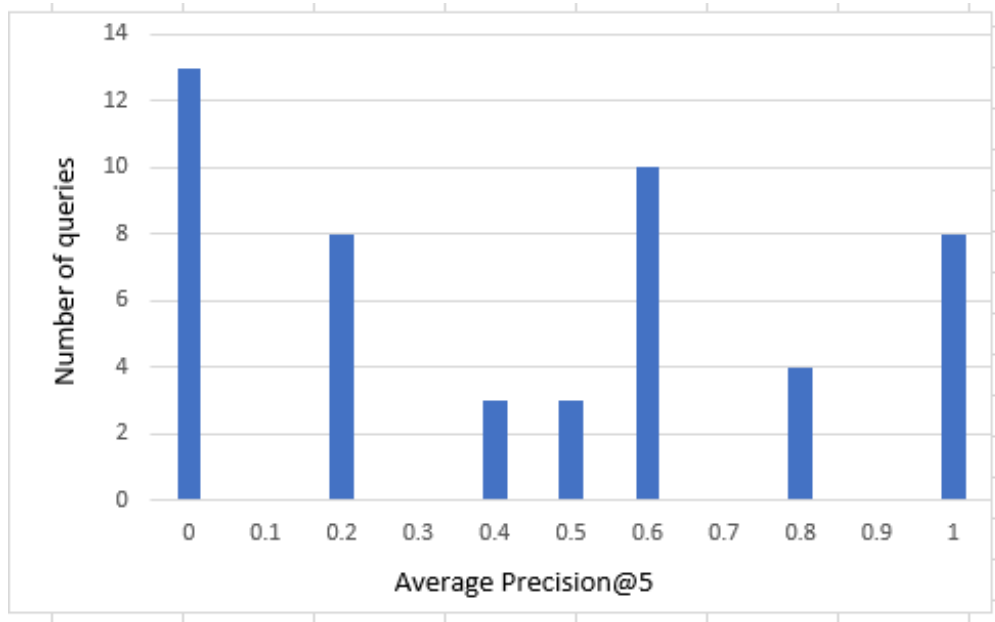


Figure 6.13: Bar graph showing number of queries against $AP@5$ values when using BERT only document embeddings

x-axis, and the number of relevant and irrelevant documents identified by the proposed system according to the relevance score on the y-axis. It can be seen from Figure 6.21 that the highest number of relevant documents with a relevance score are in the range of 0.3 – 0.4. In addition to this, the highest number of irrelevant documents with a relevance score were in the range of 0.5 – 0.6.

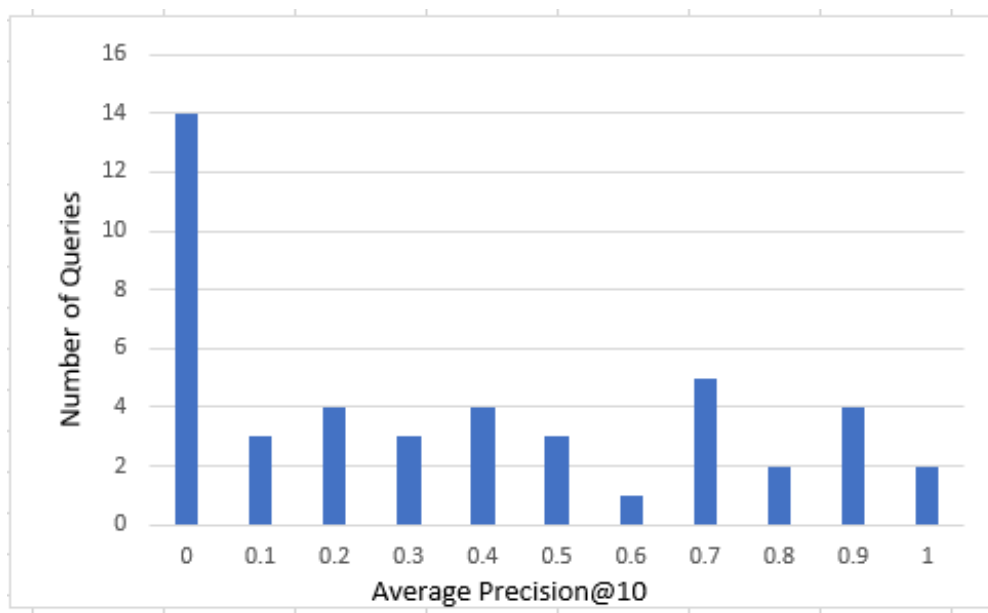


Figure 6.14: Bar graph showing number of queries against $AP@10$ values when using BERT only document embeddings

6.6 Conclusion

This chapter has proposed document embedding techniques for knowledge graph query resolution. The objective of the proposed document embedding techniques was to represent a query and the documents in the knowledge graph in a manner that will allow the documents to be ranked, according to some relevance measure, with respect to the query. The work presented in this chapter was divided into two main proposed document embedding techniques:

1. Non-Contextual Embedding Systems(CBOW embeddings)
2. Contextual Embeddings Systems (BERT and Sci-BERT embeddings)

The first, non-contextual embedding systems do not consider the context of individual words in a document. CBOW embeddings is a type of non-contextual embedding technique and was used in this chapter for query and knowledge graph document representation. The second, contextual embedding systems considered the context of individual words in a document. BERT and Sci-BERT were the two types of contextual embedding systems

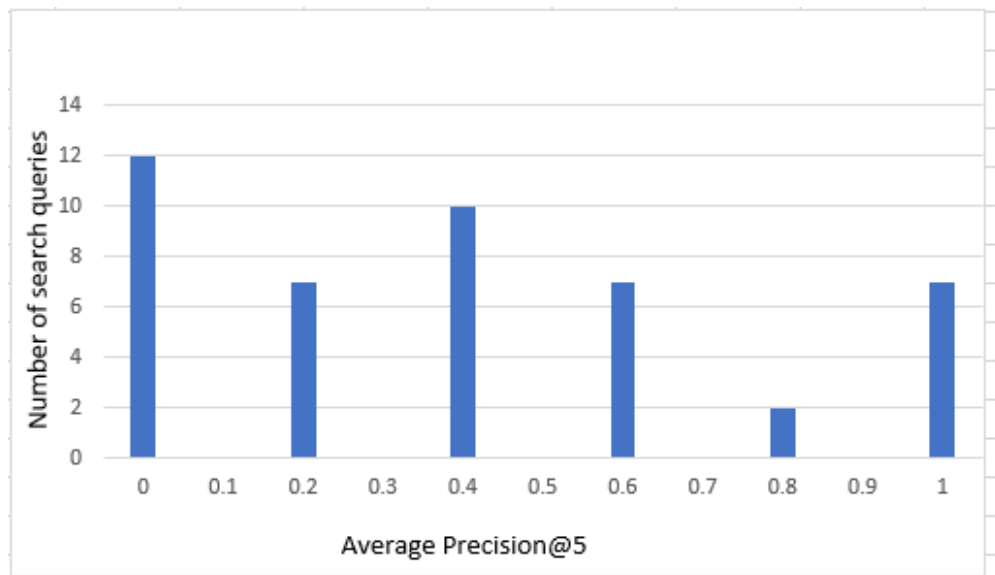


Figure 6.15: Bar graph showing number of queries against $AP@5$ values when using SciBERT+Random Walk document embeddings

used for query and knowledge graph document representation. The chapter started off, for completeness, with an introduction and background to document embedding techniques followed by the proposed methodology for query resolution. The evaluation considered precision and recall for the top five and ten documents returned by the system. The results show that CBOW combined random walk embeddings were the most suitable for query-resolution system. An empirical study was also conducted with experts from the University of Liverpool, Bio-statistics department. The empirical study demonstrated that the proposed knowledge graph query resolution system was able to identify relevant documents that the existing system was not able to identify. In some cases, there was a vocabulary mismatch between the words in a query and words in a document. In summary, the central idea in this chapter was to research and investigate techniques and document embedding methods whereby CDD literature knowledge graphs can be queried to retrieve and rank relevant documents. The following Chapter 7 concludes this thesis.

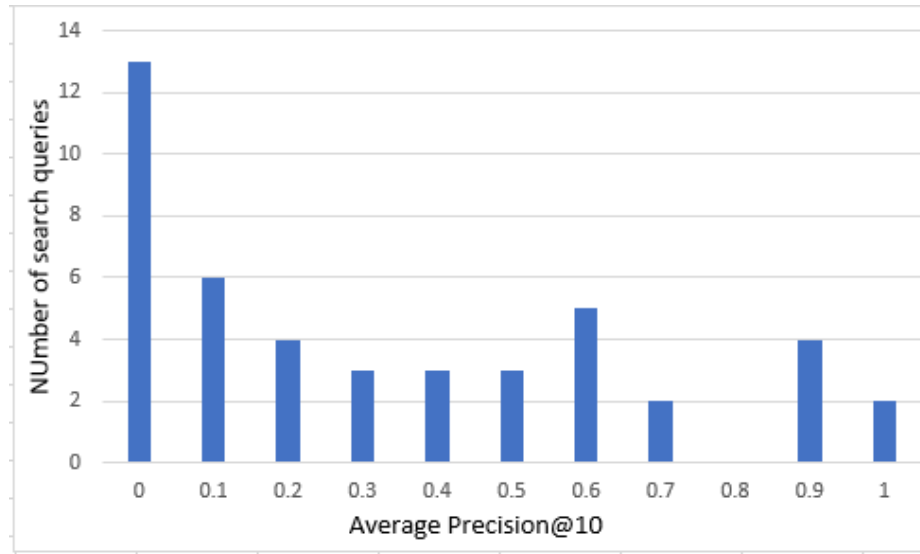


Figure 6.16: Bar graph showing number of queries against $AP@10$ values when using SciBERT+Random Walk document embeddings

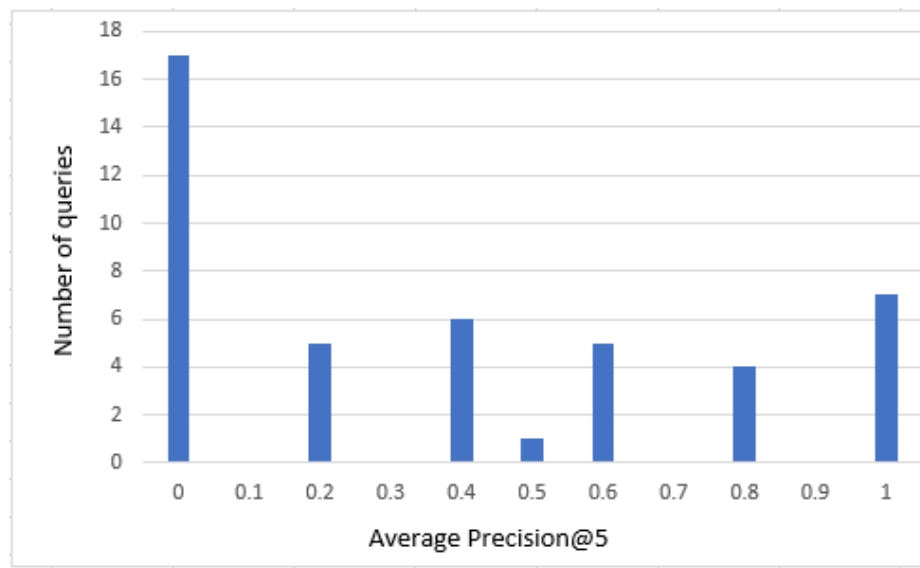


Figure 6.17: Bar graph showing number of queries against $AP@5$ values when using SciBERT only document embeddings

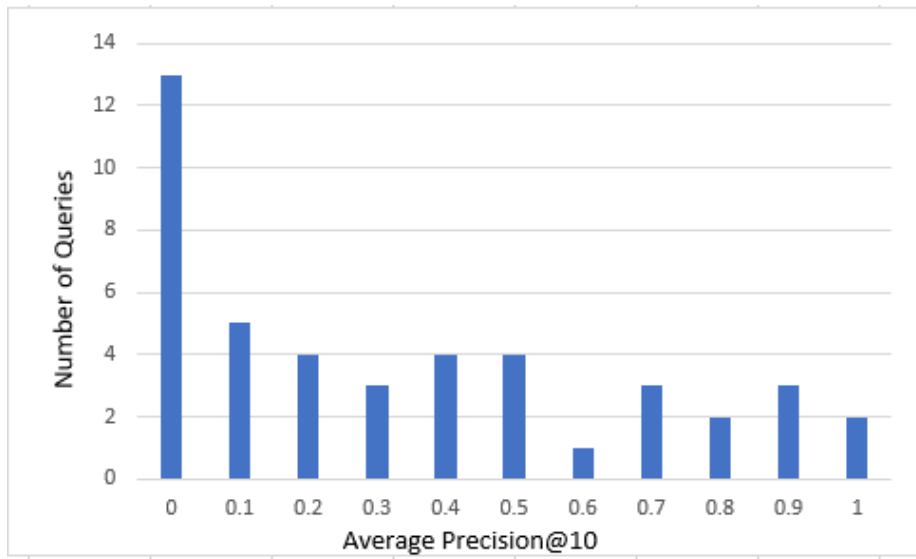


Figure 6.18: Bar graph showing number of queries against $AP@10$ values when using SciBERT only document embeddings

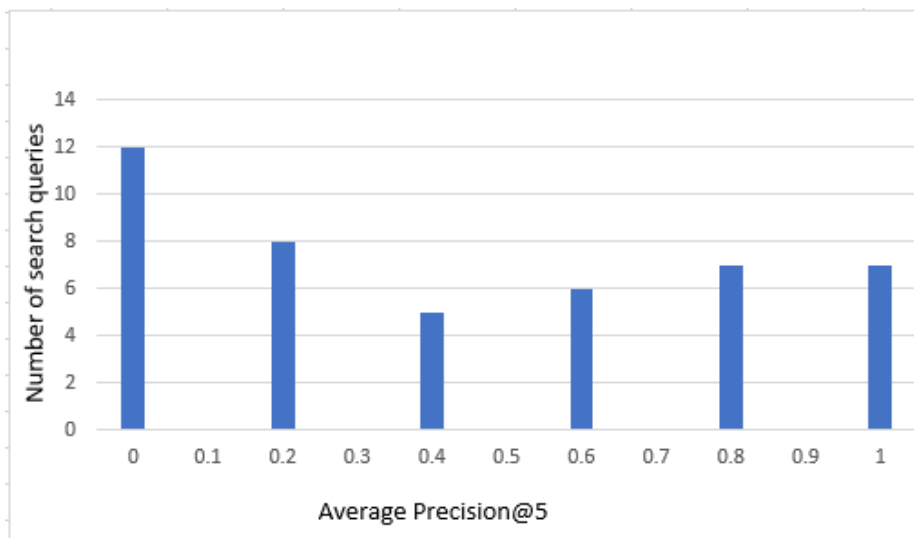


Figure 6.19: Bar graph showing number of queries against $AP@5$ values when using Random Walk document only embeddings

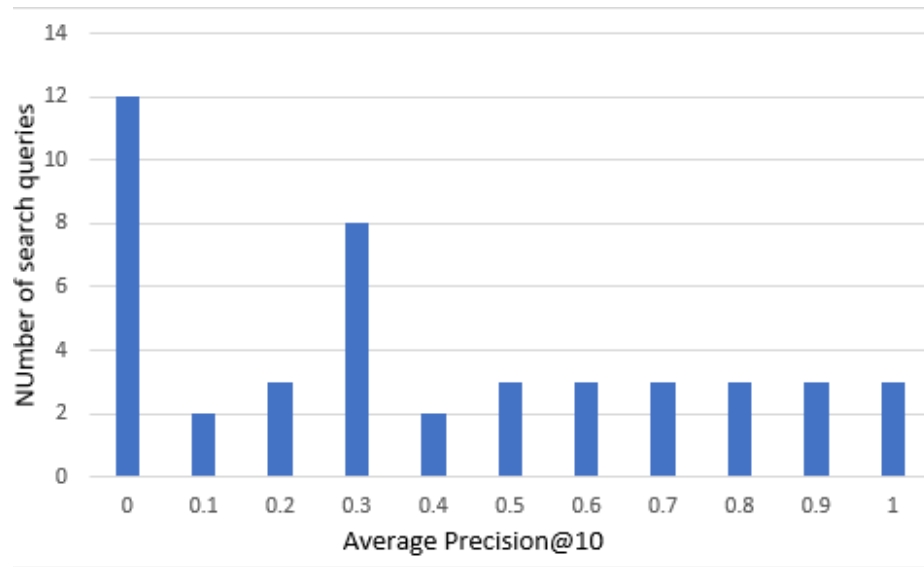


Figure 6.20: Bar graph showing number of queries against $AP@10$ values when using Random Walk only document embeddings

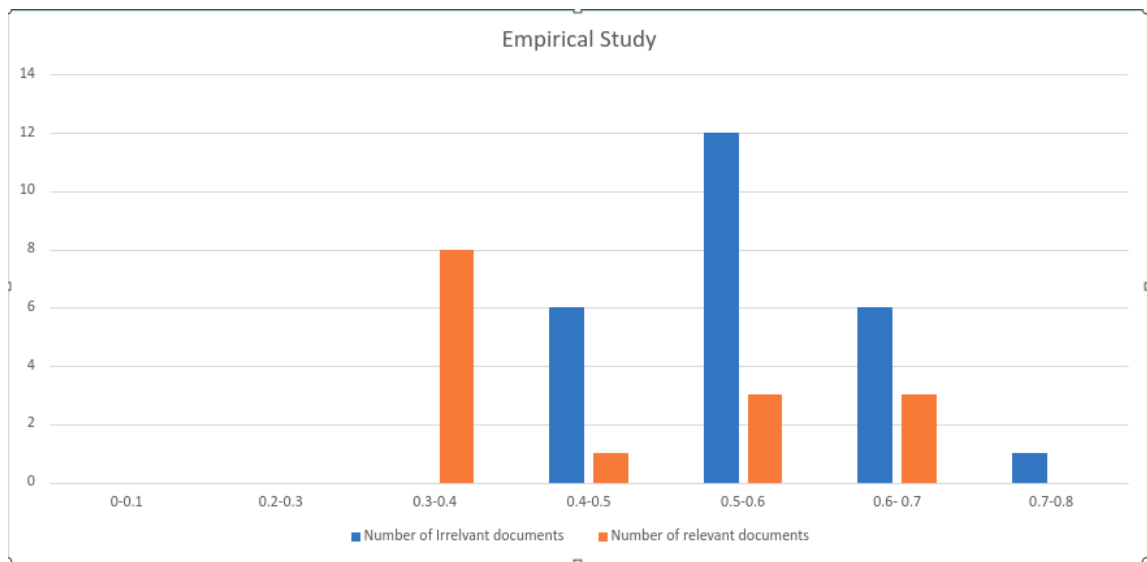


Figure 6.21: A bar graph representing the relevance scores from the KG query resolution empirical study on x-axis and number of irrelevant/relevant documents on y-axis

Chapter 7

Conclusion and Future Work

7.1 Introduction

This chapter concludes the work presented throughout this PhD thesis with a summary of the content, the main findings and some directions for future work. The chapter starts, Section 7.2, with a summary of the work. Section 7.3 then presents the main findings and contributions of the work presented in the context of the research questions and subsidiary research questions presented in Chapter 1. The chapter ends with Section 7.4, which lists a number of potential areas for future work that build upon the work presented in the thesis.

7.2 Summary of Thesis

This section gives a summary of the work presented in each chapter of this thesis. The thesis started with an introductory chapter, Chapter 1.

7.2.1 Chapter 1 key findings

1. The motivation for the research presented.
2. The research questions to be addressed, together with a set of subsidiary questions.
3. The research methodology adopted so as to provide answers to the subsidiary questions and consequently the main research question.

The central idea underpinning the thesis is the construction, maintenance and utilization of Literature Knowledge graphs in the context of Curated Document Databases (CDDs) so that scientific literature can be stored, managed and queried efficiently and effectively. The main motivation behind the idea of Literature Knowledge Graphs was the rapid increase in the volume of scientific literature being published, together with the difficulty researchers face in maintaining and querying it. Researchers, in any domain of discourse, must continuously analyse, and be aware of, existing work in their field; CDDs provide a solution. One well-known example of a CDD that has been used throughout the thesis, as both a focus for the work and as a evaluation application, is the Online Resource for Recruitment research in Clinical trials (ORRCA) CDD [58]. The ORRCA CDD¹ is a CDD of scientific publications directed at the highly specialised domain of recruitment strategies for clinical trials.

Various types of AI and machine learning techniques could have been potentially used to address the problem of creating, maintaining and using literature knowledge graph represent CDDs; however, the research focus in this thesis has been on the following:

1. Generation of literature knowledge graphs using open information extraction techniques.
2. Maintenance of literature knowledge graphs using document ranking algorithms.
3. Query-resolution using various embedding techniques including CBOW, BERT and knowledge graph embeddings.

7.2.2 Chapter 2 key findings

Chapter 2 then provided a review of the previous relevant work in context of Literature knowledge graphs. The chapter commenced with an overview of the defining literature on knowledge graphs. The chapter was divided into three research areas matching the focus for the work presented in the thesis :

1. Knowledge graph construction
2. Knowledge graph maintenance
3. Knowledge graph utilisation

¹<https://www.orrca.org.uk/>

The relevant literature was reviewed with respect to each of these research areas. With respect to the first the relevant literature on how to generate knowledge graphs using various forms of information extraction techniques was discussed. The chapter then discussed existing techniques that were adapted to update and maintaining knowledge graphs using Learning To Rank (LETOR) techniques. The chapter concluded with a review of the word embedding techniques presented in the literature; techniques that can be potentially be used to support the querying of literature knowledge graphs.

7.2.3 Chapter 3 key findings

The following chapter, Chapter 3, gave a description of the ORRCA CDD and the evaluation data sets. The chapter started with a review of the ORRCA application domain. The chapter then went on to consider the ORRCA CDD itself and the motivation behind its usage in this thesis. The chapter was concluded with a comprehensive review of the various ORRCA evaluation data sets used to support the work presented in this thesis.

The following three chapters, Chapters 4 to 6, presented a sequence of proposed approaches for the generation, maintenance and querying of CDDs represented as literature knowledge graphs. All these chapters were structured in a similar manner comprising an introduction, a description of the proposed approach, the evaluation of the approach and discussion.

7.2.4 Chapter 4 key findings

Chapter 4 introduced the proposed *Open Information Extraction for Knowledge Graph Construction* (OIE4KGC) approach for literature knowledge graph generation. The principle idea behind this approach was to use an Open Information Extraction (OIE) mechanism for the extraction of triples from a document collection that would be representative of vector-edge-vector constructs. The pre-trained RnnOIE [119] OIE extraction tool was embedded into the *OIE4KGC* approach. Two OIE tools, RnnOIE and Leolani [124] were compared and evaluated using two data sets, the ORRCA and Reverb data sets. Precision, recall and F-score were used as evaluation metrics. The results indicated that RnnOIE, turned out to be the better than Leolani in terms of precision, recall and F-score, when generating triples from a clinical trials dataset.

7.2.5 Chapter 5 key findings

Chapter 5 presented two proposed approaches to maintaining and updating literature knowledge graphs:

1. The CN approach founded on work presented in [91] and named after the initials of the author in [91].
2. The Knowledge Graph And BERT Ranking (GRAB-Rank) approach.

The chapter commenced with a review of the CN approach which used a TF-IDF n-gram feature vector representation. The GRAB-Rank approach was described next, based on a hybrid document embedding technique. The unique aspect of the GRAB-rank approach was that it combined two types of word embedding, BERT word embedding and knowledge graph embedding. The knowledge graph embeddings were generated using a random walk applied over the knowledge graph followed by the application of the node2vec framework for generating embeddings. For both the approaches the idea was to rank a set of potential documents for inclusion in a CDD, and then to select the top k for inclusion. The chapter was concluded with the evaluation of the proposed approaches. A sequence of experiments were conducted aimed at identifying the best possible values for the parameters used, and to determine the comparative effectiveness of the CN and Grab-rank approaches and the time savings gained. The evaluation suggested that the GRAB-rank approach was the most effective approach; significant time savings were identified.

7.2.6 Chapter 6 key findings

Chapter 6 commenced with a discussion of hybrid document embedding techniques for literature knowledge graph query-resolution. The work presented in this chapter comprised three proposed techniques, categorised according to the nature of the adopted embedding:

1. Continuous Bag Of Word (CBOW) embedding.
2. BERT embedding.
3. SciBERT embedding.

Each of the above embedding was combined with a graph embedding and utilized for query-resolution. The best approach identified in Chapter 6 was the combined CBOW and graph embeddings approach for query-resolution according to the evaluation performed.

7.3 Main Findings and Contributions

This section revisits the primary and subsidiary research questions that the work presented in this thesis sought to provide answers to as listed in Chapter 1. The motivation for this thesis was to investigate a set of machine learning techniques that can best support the generation, maintenance and querying of CDDs represented as literature knowledge graphs. The challenge was how this could best be achieved; the primary research question to be answered was thus:

What are some suitable techniques that can be used for generating, maintaining and utilizing literature knowledge graphs to support the concept of CDDs?

The resolution of this research question involved the resolution of a set of five subsidiary research questions. These Subsidiary Research Questions (SRQs) will therefore be considered first. Note that some of the solutions presented below impact more than one of the subsidiary research questions.

[SRQ 1] *Given a collection of documents within a CDD, represented using traditional relational database technology, how can these be processed so that they form a literature knowledge graph.*

Knowledge graphs can, of course, be created manually by using human experts but this would require extensive resource. Hence the proposed solution was to harness the tools and techniques of machine learning techniques. The challenge was to identify the nature of the vertices and edges to be included in the knowledge graph. Hence it was proposed that an existing pre-trained machine learning model should be used to extract triples from a given document collection. More specifically the *OIE4KGC* approach was proposed (Chapter 4). It was found that the proposed approach can support the effective extraction of triples from the document collection which could then be used to represent the document collection as a literature knowledge graph. The RnnOIE tool, incorporated into the *OIE4KGC* approach, was found to be one of only a very few pre-trained models that could successfully be used to extract coherent triples from document collections. Other OIE tools, such as Leolani tool [124] failed to extract meaningful and coherent triples from scientific text like ORRCA because of the long and complex sentence structure found in scientific documents. The most appropriate mechanism for processing a CDD represented as a relational database was thus found to be the proposed *OIE4KGC* approach.

[SRQ 2] *Given an existing CDD, represented as a literature knowledge graph, how can this knowledge graph be maintained to ensure that it is up to date.*

The answer to this subsidiary research question is that a knowledge graph represented CDD can best be updated by applying some form of document ranking to a collection of candidate documents and then selecting the top k .

The challenge was how to represent the set of candidate documents in a manner compatible with a literature knowledge graph. Two Learning-to-rank (LETOR) techniques were considered, the CN approach and the GRAB-Rank approach, the second featured the novel element that it combined two embeddings, a general purpose BERT embedding and a domain specific knowledge graph embedding. This representation was found to be best suited to the task of maintaining CDD literature knowledge graphs.

[SRQ 3] *Given an existing CDD, represented as a literature knowledge graph, how can this knowledge graph be queried so as to retrieve relevant documents.*

To address the challenge of query resolution with respect to CDD literature knowledge graphs a four stage process was proposed:

Stage 1: Pre-processing

Stage 2: Word embedding generation.

Stage 3: Concatenation of knowledge graph embedding and word embedding.

Stage 4: Measuring similarity between query embedding and document embeddings, and ranking.

For the word embedding generation stage three alternatives were considered:

- (a) Continuous bag of word (CBOW) embeddings
- (b) BERT
- (c) SciBERT

Each of these embeddings was used in combination with the proposed knowledge graph embedding. Again, the intuition was that combining two forms of word embedding would yield better results in terms of the effectiveness of queries than using

a single embedding on its own. The final stage, Stage 4, involved determining the similarity between queries and documents to produce a ranked list. The similarity values were then used to create a ranked list from which the top k could be selected as a response to a query. For the evaluation, a data set of query-document pairs was collected along with relevance judgements. The results indicate that the hybrid approach of CBOW and Knowledge graph Random Walk embeddings for query-resolution gave the best results. This is therefore presented as the answer to SRQ 3.

[SRQ 4] *Assuming that the maintenance and querying of literature knowledge graphs will entail some kind of document ranking, what is some suitable mechanism for deriving a ranked list of documents and what would this mechanism entail?*

The answer to SRQ 4 was considered within the context of the resolution of SRQs 2 and 3 as discussed above. The most suitable mechanism for deriving a ranked list of documents was to use some form of hybrid embedding that included a proposed domain-specific knowledge graph embedding (a graph walk embedding was advocated), and a more general embedding of some kind. For maintaining knowledge graphs a BERT embedding was found to be the most appropriate general embedding (the GRAB-Rank approach), while for querying knowledge graphs CBOW was found to be the most appropriate general embedding.

[SRQ 5] *In the context of document ranking can the concepts within a literature knowledge graph be utilized to improve a document ranking mechanism and how would this operate?*

With respect to SRQ 5 the answer was obtained as a consequence of the work directed at identifying a solution to SRQs 2 and 3. As noted above, it was established that the concepts within a literature knowledge graph could be usefully employed to improve document ranking by capturing information held within a knowledge graph using a knowledge graph embedding. The idea was incorporated into the GRAB-Rank approach where the proposed knowledge graph embedding was combined with a BERT embedding to form a hybrid embedding. The intuition in the case of the proposed Grab-Rank approach was that if two document embeddings, generated in different ways, were concatenated together it would produce a better document embedding than if the embeddings were used in isolation. Various experiments were carried

out, reported in Chapter 5, with embeddings used in combination with other embedding methods, and when used in isolation. The proposed GRAB-Rank approach was found to be the most effective. It was estimated that by using the GRAB-Rank approach a time saving of from 148 to 193 persons hours could be obtained over the manual systematic review process often used to update CDDs.

[SRQ 6] *Can the embeddings implicit within a literature knowledge graph be used to provide an answer to a query in the context of document retrieval?*

The resolution of SRQ 6 was investigated using the knowledge graph embedding model used in the context of query resolution. This embedding model was compared with three other word embedding models including:

- (a) Continuous bag of word (CBOW) embeddings
- (b) BERT document embeddings
- (c) SciBERT embedding

Various experiments were performed to investigate if knowledge graph embeddings can yield better results in the context of query-resolution. The reported evaluation indicated that when CBOW embeddings were combined with knowledge graph random walk embeddings effective query-resolution could be undertaken. The reason for this could be due to the fact that CBOW embeddings were trained on a more “general” data set and its vocabulary more suited to the ORRCA application domain when combined with a knowledge graph embedding. The pairing therefore provided for a more effective embedding than when the individual embeddings were used in isolation.

7.4 Future Work

This section presents some suggested directions for future work whereby the work presented in this thesis can be extended. Eight potential future research directions are identified as follows:

1. **Further experimentation with knowledge graph embeddings:** In Chapters 5 and 6 knowledge graph embeddings were used for the purpose of updating and

querying knowledge graph represented CDDs. Because of the computational overhead involved, the knowledge graph embeddings generated were limited to 100 random walks. It was conjectured that it might be possible, given an appropriate value for rw , the length of the random walk, to reduce this number. There is clearly a correlation between the number of random walks and the value of rw , that merits further investigation in the context of future work.

2. **Use of a Generative Pre-Trained Transformer (GPT) language model:** In Chapters 5 and 6, the BERT [21] deep learning based language model was used for generating hybrid document embeddings. The BERT language model is based on a transformer architecture. BERT only uses the encoder part of the architecture but not the decoder part. One possible fruitful avenue for further research is to investigate the use of a transformer architecture based on blocks of decoder. A Generative Pre-Trained Transformer (GPT) language model [10] can be used for this purpose. In recent literature, GPT Version 2 (GPT-2) has proven to outperform BERT with respect to many NLP tasks including document ranking. GPT-2 was trained on 175 billion parameters (ten times more than previous models, including BERT). Using GPT for word and document embeddings may lead to increasing effectiveness with respect to CDD literature knowledge graph maintenance and CDD knowledge graph query resolution. The most recent versions of GPT is GPT-3.
3. **Query expansion:** The work described in Chapter 6 proposed a query resolution mechanism. Recent literature [15, 142] has proposed the idea of query expansion whereby a given query is reformulated so as to improve retrieval performance. This recent work has shown that the use of query expansion techniques results in an increase in recall. In the context of the ORRCA CDD, the queries used by end-users are clinical in nature and domain-specific. Another topic for future research is therefore to investigate the potential of a machine learning model that can suggest additional clinical keywords when a user queries a Literature knowledge graph thereby expanding the query. The idea is that these additional keywords added to a query will improve recall and retrieve more relevant documents. It is suggested that language models like BERT and GPT-2 can be used for the query expansion.
4. **Alternative domains:** The application domain for the work presented in this thesis is the clinical trials domain; however, it is conjectured that the proposed approaches

have much wider generic applicability. It is anticipated that the proposed approaches, such as GRAB-Rank, can equally well be applied to other domains. This will provide another fruitful area for future work.

5. **Knowledge Graph Completion:** In Chapter 4 the *OIE4KGC* knowledge graph generation technique was proposed. The proposed technique generated a literature knowledge graph from a documents corpus. However, a literature knowledge graph generated in this way may have missing entities or concepts. To address the issue, a technique called Knowledge Graph Completion [145, 140] can be applied to improve the coverage of the Literature Knowledge Graph by “filling in” missing vertices and edges. This is thus the fifth proposed area for future research.
6. **Triple Extraction using alternative neural network models:** The work described in Chapter 4, used an RNN based open information extraction tool, the RnnOIE tool, for literature knowledge graph generation. With respect to future work, instead of using the RnnOIE tool it may be worthwhile to investigate the use of a BERT based architecture. It is suggested here that this may result in increased effectiveness in terms of the nature of the triples extracted. The intuition here is that BERT has been trained on a million parameters and hence might be argued to be more effective than the RnnOIE tool adopted with respect to the work presented in Chapter 4.
7. **Query resolution using a Pseudo Relevance Feedback Framework:** The work described in Chapter 6 proposed a query resolution mechanism using knowledge graphs. This proposed mechanism can be improved in real-time by receiving feedback from the user. A new concept in the field of information retrieval, called Pseudo-Relevance Feedback can be used to boost the performance of traditional Information Retrieval (IR) models by using top-ranked documents to identify new query terms, thereby reducing the effect of query-document vocabulary mismatches. It might be useful, in this context, to investigate the use of an end-to-end neural network based framework for the generation of query-terms for pseudo-relevance feedback.
8. **Knowledge Graph updating using a few-shot based document ranking approach:** Few-Shot Learning (FSL), is a kind of machine learning used in scenarios where the training data set is very limited. The general practice in machine learning

methods is to feed as much data to a machine learning model as possible to get better results. However, in domain specific cases, the data might be limited. In this case it may be fruitful to adopt FSL based methods. For the clinical trials application domain, the focus of the work presented in this thesis, the amount of training data can be argued to be limited. Hence it is suggested that it would be worthwhile to investigate the use of FSL LETOR models in the context of updating (maintaining) CDDs represented as literature knowledge graphs.

Bibliography

- [1] Yeasmin Ara Akter and Md Ataur Rahman. Extracting rdf triples from raw text. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–4. IEEE, 2019.
- [2] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*, 2019.
- [3] 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. *arXiv preprint arXiv:1805.02262*, 2018.
- [4] Ebrahim Bagheri, Faezeh Ensan, and Feras Al-Obeidat. Neural word and entity embeddings for ad hoc retrieval. *Information Processing & Management*, 54(4):657–673, 2018.
- [5] François Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics*, 41(5):706–716, 2008.
- [6] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.
- [7] Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, 2011.
- [8] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl.1):D267–D270, 2004.

- [9] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Free-base: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.
- [10] Pawel Budzianowski and Ivan Vulić. Hello, it’s gpt-2—how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*, 2019.
- [11] Judy F Burnham. Scopus database: a review. *Biomedical digital libraries*, 3(1):1–8, 2006.
- [12] Happy Buzaaba and Toshiyuki Amagasa. Question answering over knowledge base: a scheme for integrating subject and the identified relation to answer simple questions. *SN Computer Science*, 2(1):1–13, 2021.
- [13] Kathi Canese and Sarah Weis. Pubmed: the bibliographic database. *The NCBI Handbook*, 2:1, 2013.
- [14] Dhivya Chandrasekaran and Vijay Mago. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37, 2021.
- [15] Vincent Claveau. Query expansion with artificially generated texts. *arXiv preprint arXiv:2012.08787*, 2020.
- [16] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 126–134, 2018.
- [17] Zihang Dai, Lei Li, and Wei Xu. Cfo: Conditional focused neural question answering with large-scale knowledge bases. *arXiv preprint arXiv:1606.01994*, 2016.
- [18] Jeffrey Dalton, Laura Dietz, and James Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374, 2014.
- [19] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.

-
- [20] Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *Future Generation Computer Systems*, 116:253–264, 2021.
- [21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [22] Laura Dietz, Alexander Kotov, and Edgar Meij. Utilizing knowledge graphs for text-centric information retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1387–1390, 2018.
- [23] Kevin Donnelly et al. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279, 2006.
- [24] Jennifer D’Souza, Isaiah Onando Mulang, and Sören Auer. Team svmrank: Leveraging feature-rich support vector machines for ranking explanations to elementary science questions. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 90–100, 2019.
- [25] Mohamed El Mohadab, Belaid Bouikhalene, and Said Safi. Predicting rank for scientific research papers using supervised learning. *Applied Computing and Informatics*, 15(2):182–190, 2019.
- [26] Faezeh Ensan and Ebrahim Bagheri. Document retrieval model through semantic linking. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 181–190, 2017.
- [27] Faezeh Ensan, Ebrahim Bagheri, Amal Zouaq, and Alexandre Kouznetsov. An empirical study of embedding features in learning to rank. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2059–2062, 2017.
- [28] Gonenc Ercan, Shady Elbassuoni, and Katja Hose. Retrieving textual evidence for knowledge graph facts. In *European Semantic Web Conference*, pages 52–67. Springer, 2019.

- [29] Massimo Esposito, Emanuele Damiano, Aniello Minutolo, Giuseppe De Pietro, and Hamido Fujita. Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. *Information Sciences*, 514:88–105, 2020.
- [30] Andy Extance. How ai technology can tame the scientific literature. *Nature*, 561(7722):273–275, 2018.
- [31] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545, 2011.
- [32] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, 2013.
- [33] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165, 2014.
- [34] Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *arXiv preprint arXiv:1910.08435*, 2019.
- [35] Michael Färber. The microsoft academic knowledge graph: a linked data source with 8 billion triples of scholarly data. In *International Semantic Web Conference*, pages 113–129. Springer, 2019.
- [36] Said Fathalla and Christoph Lange. Eventskg: a knowledge graph representation for top-prestigious computer science events metadata. In *International conference on computational collective intelligence*, pages 53–63. Springer, 2018.
- [37] Valeria Fionda and Giuseppe Pirrò. Learning triple embeddings from knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3874–3881, 2020.
- [38] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

-
- [39] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
- [40] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. Semantic matching by non-linear word transportation for information retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 701–710, 2016.
- [41] Lushan Han, Tim Finin, Cynthia Parr, Joel Sachs, and Anupam Joshi. Rdf123: from spreadsheets to rdf. In *International Semantic Web Conference*, pages 451–466. Springer, 2008.
- [42] Nicola Harman, Shaun Treweek, Mike Clarke, Paula Williamson, Pete Bower, and Carrol Gamble. Development of an online resource for recruitment research in clinical trials (orrca). *Trials*, 16(2):1–1, 2015.
- [43] Faegheh Hasibi, Krisztian Balog, Darío Garigliotti, and Shuo Zhang. Nordlys: A toolkit for entity-oriented and semantic search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1289–1292, 2017.
- [44] Saeed-Ul Hassan, Mubashir Imran, Sehrish Iqbal, Naif Radi Aljohani, and Raheel Nawaz. Deep context of citations using machine-learning models in scholarly full-text articles. *Scientometrics*, 117(3):1645–1662, 2018.
- [45] Lukáš Havrlant and Vladik Kreinovich. A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). *International Journal of General Systems*, 46(1):27–36, 2017.
- [46] Weiwei Hu, Anhong Dang, and Ying Tan. A survey of state-of-the-art short text matching algorithms. In *International Conference on Data Mining and Big Data*, pages 211–219. Springer, 2019.
- [47] Muhammad Ibrahim and Mark Carman. Comparing pointwise and listwise objective functions for random-forest-based learning-to-rank. *ACM Transactions on Information Systems (TOIS)*, 34(4):1–38, 2016.

- [48] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture*, pages 243–246, 2019.
- [49] Jyun-Yu Jiang, Mingyang Zhang, Cheng Li, Michael Bendersky, Nadav Golbandi, and Marc Najork. Semantic text matching for long-form documents. In *The world wide web conference*, pages 795–806, 2019.
- [50] Thorsten Joachims. Svmlight: Support vector machine. *SVM-Light Support Vector Machine* <http://svmlight.joachims.org/>, University of Dortmund, 19(4), 1999.
- [51] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.
- [52] Xuchan Ju, Zhenghao Yan, and Tianhe Wang. Overview of optimization algorithms for large-scale support vector machines. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 909–916. IEEE, 2021.
- [53] Zakaria Kaddari, Jamal Berrich, Noureddine Rahmoun, Saida Belouali, and Toumi Bouchentouf. Inkad covid-19 intellisearch: a multilingual search engine for answering questions about covid-19 in real-time from the scientific literature. In *2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS)*, pages 1–6. IEEE, 2021.
- [54] Zakaria Kaddari, Youssef Mellah, Jamal Berrich, Toumi Bouchentouf, and Mohammed G Belkasmi. Biomedical question answering: A survey of methods and datasets. In *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, pages 1–8. IEEE, 2020.
- [55] Maulik R Kamdar and Michel Dumontier. An ebola virus-centered knowledge base. *Database*, 2015, 2015.
- [56] Joseph Kamtchum-Tatuene and Joseline Guetsop Zafack. Keeping up with the medical literature: Why, how, and when? *Stroke*, 52(11):e746–e748, 2021.

- [57] P Karthik, M Saurabh, and U Chandrasekhar. Classification of text documents using association rule mining with critical relative support based pruning. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1219–1223. IEEE, 2016.
- [58] 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.
- [59] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [60] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors, 2015.
- [61] Oliver Kramer. Scikit-learn. In *Machine learning for evolution strategies*, pages 45–53. Springer, 2016.
- [62] Mayura Kulkarni and Shubhangi Kale. Information retrieval based improvising search using automatic query expansion. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pages 1226–1230. IEEE, 2021.
- [63] Dik L Lee, Huei Chuang, and Kent Seamons. Document ranking and the vector-space model. *IEEE software*, 14(2):67–75, 1997.
- [64] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.
- [65] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. Parade: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093*, 2020.
- [66] Hang Li and Jun Xu. Semantic matching in search. *Foundations and Trends in Information retrieval*, 7(5):343–469, 2014.

- [67] Zhenyang Li, Guangluan Xu, Xiao Liang, Feng Li, Lei Wang, and Daobing Zhang. Exploring the importance of entities in semantic ranking. *Information*, 10(2):39, 2019.
- [68] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325, 2021.
- [69] Carolyn E Lipscomb. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265, 2000.
- [70] Jiaying Liu, Jing Ren, Wenqing Zheng, Lianhua Chi, Ivan Lee, and Feng Xia. Web of scholars: A scholar knowledge graph. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2153–2156, 2020.
- [71] Xitong Liu and Hui Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, 2015.
- [72] Ying-Hsang Liu and Nina Wacholder. Evaluating the impact of mesh (medical subject headings) terms on different types of searchers. *Information Processing & Management*, 53(4):851–870, 2017.
- [73] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. *arXiv preprint arXiv:1805.07591*, 2018.
- [74] Xuan Lv and Nora M El-Gohary. Enhanced context-based document relevance assessment and ranking for improved information retrieval to support environmental decision making. *Advanced Engineering Informatics*, 30(4):737–750, 2016.
- [75] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104, 2019.
- [76] 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.
- [77] Jose L Martinez-Rodriguez, Ivan López-Arévalo, and Ana B Rios-Alvarado. Openie-based approach for knowledge graph construction from text. *Expert Systems with Applications*, 113:339–355, 2018.
- [78] Ryan McDonald, Georgios-Ioannis Brokos, and Ion Androutsopoulos. Deep relevance ranking using enhanced document-query interactions. *arXiv preprint arXiv:1809.01682*, 2018.
- [79] HJ Meijer, J Truong, and R Karimi. Document embedding for scientific articles: Efficacy of word embeddings vs tfidf. *arXiv preprint arXiv:2107.05151*, 2021.
- [80] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [81] Beverley C Millar and Michelle Lim. The role of visual abstracts in the dissemination of medical research. *Ulster Med J*, 91(2):67–78, 2022.
- [82] Mandar Mitra and BB Chaudhuri. Information retrieval from documents: A survey. *Information retrieval*, 2(2-3):141–163, 2000.
- [83] 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.
- [84] Iqra Muhammad, Anna Kearney, Carrol Gamble, Frans Coenen, and Paula Williamson. Open information extraction for knowledge graph construction. In *International Conference on Database and Expert Systems Applications*, pages 103–113. Springer, 2020.
- [85] Eric Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 83–84, 2016.

- [86] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. A survey on open information extraction. *arXiv preprint arXiv:1806.05599*, 2018.
- [87] Fedor Nikolaev and Alexander Kotov. Joint word and entity embeddings for entity retrieval from a knowledge graph. In *European Conference on Information Retrieval*, pages 141–155. Springer, 2020.
- [88] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.
- [89] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Document ranking with a pre-trained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*, 2020.
- [90] Marc-Alexandre Nolin, Michel Dumontier, François Belleau, and Jacques Corbeil. Building an hiv data mashup using bio2rdf. *Briefings in bioinformatics*, 13(1):98–106, 2012.
- [91] Christopher R Norman, Elizabeth Gargon, Mariska MG Leeflang, Aurélie Névéol, and Paula R Williamson. Evaluation of an automatic article selection method for timelier updates of the comet core outcome set database. *Database*, 2019, 2019.
- [92] Alison O’Mara-Eves, James Thomas, John McNaught, Makoto Miwa, and Sophia Ananiadou. Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic reviews*, 4(1):1–22, 2015.
- [93] Harshith Padigela, Hamed Zamani, and W Bruce Croft. Investigating the successes and failures of bert for passage re-ranking. *arXiv preprint arXiv:1905.01758*, 2019.
- [94] Enrico Palumbo, Giuseppe Rizzo, Raphaël Troncy, Elena Baralis, Michele Osella, and Enrico Ferro. Knowledge graph embeddings with node2vec for item recommendation. In *European semantic web conference*, pages 117–120. Springer, 2018.
- [95] Gaurav Pandey. Utilization of efficient features, vectors and machine learning for ranking techniques. *JYU dissertations*, 2019.
- [96] Beomjoo Park, Muhammad Afzal, Jamil Hussain, Asim Abbas, and Sungyoung Lee. Automatic identification of high impact relevant articles to support clinical decision making using attention-based deep learning. *Electronics*, 9(9):1364, 2020.

- [97] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [98] Maria Pelevina, Nikolay Arefyev, Chris Biemann, and Alexander Panchenko. Making sense of word embeddings. *arXiv preprint arXiv:1708.03390*, 2017.
- [99] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [100] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [101] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*, 2019.
- [102] Razieh Rahimi, Ali MontazerAlghaem, and James Allan. Listwise neural ranking models. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 101–104, 2019.
- [103] Jinfeng Rao, Linqing Liu, Yi Tay, Wei Yang, Peng Shi, and Jimmy Lin. Bridging the gap between relevance matching and semantic matching for short text similarity modeling. 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 5370–5381, 2019.
- [104] Hadas Raviv, Oren Kurland, and David Carmel. Document retrieval using entity-based language models. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 65–74, 2016.
- [105] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*, pages 177–185. Springer, 2016.

- [106] Rasmus Ros, Elizabeth Bjarnason, and Per Runeson. A machine learning approach for semi-automated search and selection in literature studies. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pages 118–127, 2017.
- [107] Anderson Rossanez, Julio Cesar Dos Reis, Ricardo da Silva Torres, and Hélène de Ribaupierre. Kgen: a knowledge graph generator from biomedical scientific literature. *BMC medical informatics and decision making*, 20(4):1–24, 2020.
- [108] 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.
- [109] Mourad Sarrouiti and Said Ouatic El Alaoui. Sembionlqa: A semantic biomedical question answering system for retrieving exact and ideal answers to natural language questions. *Artificial intelligence in medicine*, 102:101767, 2020.
- [110] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, 2020.
- [111] Peri L Schuyler, William T Hole, Mark S Tuttle, and David D Sherertz. The umls metathesaurus: representing different views of biomedical concepts. *Bulletin of the Medical Library Association*, 81(2):217, 1993.
- [112] Xiao Sha, Zhu Sun, and Jie Zhang. Hierarchical attentive knowledge graph embedding for personalized recommendation. *Electronic Commerce Research and Applications*, 48:101071, Jul 2021.
- [113] Dilip Kumar Sharma, Rajendra Pamula, and DS Chauhan. Semantic approaches for query expansion. *Evolutionary Intelligence*, pages 1–16, 2021.
- [114] Sonam Sharma. Fact-finding knowledge-aware search engine. In *Data Management, Analytics and Innovation*, pages 225–235. Springer, 2022.
- [115] 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.

- [116] Kuldeep Singh, Ioanna Lytra, Arun Sethupat Radhakrishna, Saeedeh Shekarpour, Maria-Esther Vidal, and Jens Lehmann. No one is perfect: Analysing the performance of question answering components over the dbpedia knowledge graph. *J. Web Semant.*, 65:100594, 2020.
- [117] Vergil N Slee. The international classification of diseases: ninth revision (icd-9), 1978.
- [118] Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, 2016.
- [119] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, 2018.
- [120] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers to non-factoid questions from web collections. *Computational linguistics*, 37(2):351–383, 2011.
- [121] Gesare Asnath Tinega, Waweru Mwangi, and M Rimiru Richard. Text mining in digital libraries using okapi bm25 model. *International Journal of Computer Applications Technology and Research*, 7(10):398–406, 2019.
- [122] Ahmet Uyar and Farouk Musa Aliyu. Evaluating search features of google knowledge graph and bing satori: entity types, list searches and query interfaces. *Online Information Review*, 2015.
- [123] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [124] Piek Vossen, Selene Baez, Lenka Bajcetić, and Bram Kraaijeveld. Leolani: a reference machine with a theory of mind for social communication. In *International conference on text, speech, and dialogue*, pages 15–25. Springer, 2018.

- [125] 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.
- [126] Daniel S Weld, Raphael Hoffmann, and Fei Wu. Using wikipedia to bootstrap open information extraction. *Acm Sigmod Record*, 37(4):62–68, 2009.
- [127] Tino Werner. A review on instance ranking problems in statistical learning. *Machine Learning*, pages 1–49, 2021.
- [128] Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, 2016.
- [129] Colby Wise, Vassilis N Ioannidis, Miguel Romero Calvo, Xiang Song, George Price, Ninad Kulkarni, Ryan Brand, Parminder Bhatia, and George Karypis. Covid-19 knowledge graph: accelerating information retrieval and discovery for scientific literature. *arXiv preprint arXiv:2007.12731*, 2020.
- [130] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.
- [131] Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. 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 219–228, 2019.
- [132] Chenyan Xiong and Jamie Callan. Esdrank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 951–960, 2015.
- [133] Chenyan Xiong and Jamie Callan. Query expansion with freebase. In *Proceedings of the 2015 international conference on the theory of information retrieval*, pages 111–120, 2015.

-
- [134] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. Bag-of-entities representation for ranking. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pages 181–184, 2016.
- [135] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. Word-entity duet representations for document ranking. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 763–772, 2017.
- [136] Chenyan Xiong, Russell Power, and Jamie Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pages 1271–1279, 2017.
- [137] Mohamed Yahya, Steven Whang, Rahul Gupta, and Alon Halevy. Renoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 325–335, 2014.
- [138] Min-Chul Yang, Nan Duan, Ming Zhou, and Hae Chang Rim. Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 645–650, 2014.
- [139] Yi Yang, Mark Christopher Siy Uy, and Allen Huang. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*, 2020.
- [140] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.
- [141] Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, 2007.
- [142] Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. Improved neural relation detection for knowledge base question answering. *arXiv preprint arXiv:1704.06194*, 2017.
- [143] Xiang Yue, Zhen Wang, Jingong Huang, Srinivasan Parthasarathy, Soheil Moosavinasab, Yungui Huang, Simon M Lin, Wen Zhang, Ping Zhang, and Huan Sun.

Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, 36(4):1241–1251, 2020.

- [144] Hamed Zamani, Mostafa Dehghani, Fernando Diaz, Hang Li, and Nick Craswell. Sigir 2018 workshop on learning from limited or noisy data for information retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1439–1440, 2018.
- [145] Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V Chawla. Few-shot knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3041–3048, 2020.
- [146] Guiyang Zhang, Pan Wang, You Li, and Guohua Huang. Random walks on biomedical networks. *Current Proteomics*, 18(5):608–619, 2021.
- [147] Sendong Zhao, Chang Su, Andrea Sboner, and Fei Wang. Graphene: A precise biomedical literature retrieval engine with graph augmented deep learning and external knowledge empowerment. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 149–158, 2019.
- [148] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.
- [149] 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.

Appendix A

Appendix 1

Table A.1: Table showing the search queries in the ORRCA query-document dataset mentioned in Chapter 6.

File Name	Search Terms in Query	Records
ORRCA 06 04 16 10 52	E5	84
ORRCA 06-17	Blinding	37
ORRCA 06 18 10 40 25	B3 and health area=Cancer	32
ORRCA 06 21 13 38 30BD searchone	Qualitative inter- views	10
ORRCA 06 23 11 40 33 jh SEARCH 2	randomisation	156
ORRCA 06 28 19 02 19 JH search 3	recruiter equipoise	82
ORRCA 06 29 11 31 52 JH search 4	D3 and Health area=Cancer	86
ORRCA 06 29 16 43 42 AK3	Methods=focus groups AND Out- comes+ reasons for participation or refusal	66
ORRCA 06 3013 51 07 JH search 5	B8	21

Continued on next page

Table A.1 – continued from previous page

File name	Search Terms in Query	Records
ORRCA 07 06 11 28 59 JH search 6	D2 AND cancer	64
ORRCA 07 1513 50 08 AK4	Abstract= 'ethnic' AND Health de- scription=Diabetes	11
ORRCA 07 1513 59 04 AK5	Evidence type = randomised evalua- tion AND research outcomes =recruit- ment Cost	20
ORRCA 07 15 14 48 51 AK6	Newsletter	6
ORRCA 07 15 14 56 28 AK7	Video	39
ORRCA 07 15 1020 17BDSearch 2b	Abstract = 'so- cial media' AND Timing= during feasibility	61
ORRCA 07 15 10 58 02 BD search three	labour	31
ORRCA 07 15 10 20 17 BD Search 5	Postpartum haemor- rhage	31
BD Search 6	Abstract= 'situa- tional incapacity' AND Recruitment	research meth- ods= Qual- ita- tive inter- views
31		
ORRCA Search2021-07-28 ₁ 1 – 17 – 10 _A K8	Recruitment setting =intensive care	80
ORRCA Search2021 08 09 14 16 34 AK	Pharmacy	20

Continued on next page

Table A.1 – continued from previous page

File name	Search Terms in Query	Records
ORRCA Search2021 08 09 14 40 33 Ak10	F3	49
<i>ORRCA</i> Search2021 08 09 ₁₅ 20 03AK11	rural	90
ORRCA Search2021 08 19 13 13 30 Ak12	C10 AND Health area=Cardiovascular	10
ORRCA Search2021 08 19 13 33 36 Ak13	Research methods= systematic review and reviews AND Health area= Mental Health	34
ORRCA Search2021 08 20 10 13 58 BD search 7	Abstract=Research without prior con- sent AND A4	64
ORRCA Search2021 08 20 11 55 25 BD 8	Abstract= accept- ability AND Host design =Cluster	15
ORRCA Search2021 08 26 11 41 12 BD search 9	Migraine	6
ORRCA Search2021 08 26 13 48 27 search 10	abstract= obesity AND research meth- ods= qualitative interviews	8
ORRCA Search2021 08 26 14 43 18 bd search 11 v4	abstract =paediatric AND recruitment research methods =qualitative inter- views AND host design =RCT	14
ORRCA Search2021 08 26 15 14 43 AK14	Facebook	47
ORRCA Search2021 08 26 15 43 23 AK15	palliative	63
ORRCA Search2021 08 27 11 18 48 jh SEARCH 7	C10	98
ORRCA Search2021 08 27 12 20 18 JH SEARCH 8	B5	41

Continued on next page

Table A.1 – continued from previous page

File name	Search Terms in Query	Records
ORRCA Search2021 08 27 12 47 16 JH SEARCH 9	D4 AND health area= Cardiovascular	9
ORRCA Search2021 08 27 12 55 28 Jh Search 10	D3 AND health area= Infection	35
ORRCA Search2021 08 27 13 57 00 JH Search 11	E2 AND health area= Mental Health	25
ORRCA Search2021 08 27 14 25 34 JH search 12	C8 AND host design =RCT AND health intervention= surgery	25
ORRCA Search2021 08 27 14 38 46 JH Search 13	F2 AND host design =RCT	74
ORRCA Search2021 08 27 17 01 22 JH Seearch 14	F3 AND Health area= Cancer	4
ORRCA Search2021 08 27 17 06 38 JH Search 15	c7 AND Host design = RCT and Health intervention= surgery	16
ORRCA Search2021 08 27 13 57 00 JH Search 11	E2 AND health area= Mental Health	25
ORRCA Search2021 08 26 15 04 21 bd search 12	bereaved	11
ORRCA Search2021 08 31 20 42 03 BD search 13	abstract=obstetrics	9
ORRCA Search2021 09 04 20 36 28 BD search 14	Situational incapacity	4
ORRCA Search2021 08 26 15 04 21 bd search 12	bereaved	11

Continued on next page

Table A.1 – continued from previous page

File name	Search Terms in Query	Records
ORRCA Search2021 09 03 14 03 43search15	abstract=outcome measures setting AND research meth- ods= Focus groups	28

Document title	Document Relevance label
Why is recruitment to trials difficult? An investigation	n
What are the barriers and facilitators to patient and carer recruitment	n
Recruitment strategies for caregivers of children with mental health problems	n
Participation rates in epidemiologic studies	n
Surrogate and patient discrepancy regarding consent for critical care research	y
Challenges of a community based pragmatic, randomised controlled trial of weight loss maintenance	n
Cancer clinical trials: reasons for poor patient accrual	n
The effect of depression on the decision to join a clinical trial	n
Bayesian modeling and prediction of accrual in multi-regional clinical trials	n
Recruitment and retention of homeless mentally	n

Table A.2: Table showing documents returned by the proposed system as part of the Empirical Study for the query “bereaved” and their relevance label as discussed in Chapter 6

Document title	Document Relevance label
I will do it if it will help others: motivations among patients taking part in qualitative studies in palliative care	y
Predictive Hierarchic Modeling of Operational Characteristics in Clinical Trials	n
Strategies for assessment and recruitment of subjects for nursing research	n
PLANNING A CLINICAL-TRIAL WITH ALLOWANCE FOR COST AND PATIENT RECRUITMENT RATE	n
Recruiting for research in hospice: Feasibility of a research screening protocol	y
Equipoise: a case study of the views of clinicians involved in two neonatal trials	n
Conceptual framework and systematic review of the effects of participants' and professionals' preferences in randomised controlled trials	n
Recruitment in multicentre trials: Prediction and adjustment	n
Preparation, information and liaison: conducting successful research in palliative care	y
Surgical management of subfoveal choroidal neovascular membranes in age-related macular degeneration by macular relocation: experiences of an early-stopped randomised clinical trial	n
The impact of patient involvement in the work of the Dementias	n

Table A.3: Table showing documents returned by the proposed system in Empirical Study for the query “palliative” and their relevance label as discussed in Chapter 6

Document title	Document Relevance label
Conducting online focus groups on Facebook to inform health behavior change interventions	y
The effect of exposure to social annotation on online informed consent beliefs and behavior	n
Recruiting young adults to child maltreatment research through Facebook: A feasibility study	y
Clinical Trial Recruitment with Social Media - What to Expect	y
Impact of Baseline Assessment Modality on Enrollment and Retention in a Facebook Smoking Cessation Study	y
Internet versus mailed questionnaires: a randomized comparison	n
The Use of Facebook in Recruiting Participants for Health Research Purposes: A Systematic Review	y
Exploring the Viability of Using Online Social Media Advertising as a Recruitment Method for Smoking Cessation Clinical Trials	y
Outcomes in Child Health: Exploring the Use of Social Media to Engage Parents in Patient-Centered Outcomes Research	y
Social Media-Delivered Sexual Health Intervention A Cluster Randomized Controlled Trial	y
Recruitment of adolescents for a smoking study: use of traditional strategies	y

Table A.4: Table showing documents returned by the proposed system in Empirical Study for the query “facebook” and their relevance label as discussed in Chapter 6

Document title	Document Relevance label
Small Changes and Lasting Effects (SCALE) Trial	n
Organizational and employee level recruitment into a worksite-based weight loss study	n
Evaluation of active and passive recruitment methods used in randomized controlled trials targeting pediatric obesity	n
Recruitment Evaluation of a Preschooler Obesity-Prevention Intervention	n
A feasibility randomised controlled trial of a motivational interviewing-based intervention for weight loss maintenance in adults	n
Sample size in obesity trials: Patient perspective versus current practice	y
Reach, engagement, and retention in an internet-based weight loss program in a multi-site randomized controlled trial	n
Evaluation of recruitment methods for a trial targeting childhood obesity: Families for Health randomised controlled trial	y
Recruiting young adults into a weight loss trial: Report of protocol development and recruitment results	y
Barriers to Recruitment in Pediatric Obesity Trials: Comparing Opt-in and Opt-out Recruitment Approaches	n
Racial and ethnic minority enrollment in randomized clinical trials of behavioural weight loss utilizing technology: a systematic review	n

Table A.5: Table showing documents returned by the proposed system in Empirical Study for the query “obesity” and their relevance label as discussed in Chapter 6