Ranking Narrative Query Graphs for Biomedical Document Retrieval

Hermann Kroll
krollh@acm.org
Institute for Information Systems, TU Braunschweig
Germany

Pascal Sackhoff
p.sackhoff@tu-bs.de
Institute for Information Systems, TU Braunschweig
Germany

Timo Breuer
timo.breuer@th-koeln.de
TH Köln (University of Applied Sciences), Germany
Germany

Ralf Schenkel
schenkel@uni-trier.de
Universität Trier
Germany

Wolf-Tilo Balke
balke@ifis.cs.tu-bs.de
Institute for Information Systems, TU Braunschweig
Germany

Abstract

Keyword-based searches are today's standard in digital libraries. Yet, complex retrieval scenarios like in scientific knowledge bases, need more sophisticated access paths. Although each document somewhat contributes to a domain's body of knowledge, the exact structure between keywords, i.e., their possible relationships, and the contexts spanned within each single document will be crucial for effective retrieval. Following this logic, individual documents can be seen as small-scale knowledge graphs on which graph queries can provide focused document retrieval. We implemented a full-fledged graph-based discovery system for the biomedical domain and demonstrated its benefits in the past. Unfortunately, graph-based retrieval methods generally follow an 'exact match' paradigm, which severely hampers search efficiency, since exact match results are hard to rank by relevance. This paper extends our existing discovery system and contributes effective graph-based unsupervised ranking methods, a new query relaxation paradigm, and ontological rewriting. These extensions improve the system further so that users can retrieve results with higher precision and higher recall due to partial matching and ontological rewriting.

Keywords

Graph-based Ranking, Document Retrieval, Digital Libraries

1 Introduction

This article is extended by our technical report [7] that describes and discusses related work, our method and evaluation in more detail. Digital libraries usually implement document retrieval through simple-to-use keyword-based access paths. However, in complex retrieval scenarios like for scientific documents, the use of learning architectures to learn the relevance between a user's query and individual textual documents can severely boost retrieval performance [1, 2, 9, 12]. In a nutshell, such systems use a first stage for initial retrieval and then apply strategies like *neural re-ranking* or *learning-to-rank* to estimate the relevance of documents, e.g., [2, 9, 12–14, 21]. Although these approaches proved to be very effective on different benchmarks, they come with two major limitations: First, a large quantity of training data needs to be provided to learn how the documents' relevance relates to individual queries. Second, applying deep learning in large-scale scenarios is quite costly: acquiring training data, training time, hardware, etc.

Building on the success of large knowledge graphs, a viable alternative is to adapt the graph-based retrieval paradigm to IR-style document retrieval. In the past, we proposed so-called narrative query graphs, see [5, 6] and www.narrative.pubpharm.de for an implementation in the field of bio-medicine. Here, users can represent information needs as directed *edge-labeled graphs*. This intuitive kind of querying means simply stating relevant concepts and their interactions and can be supported by suitable user interfaces [5]. The resulting query graph representation is then matched against a large set of focused document graphs, each individually extracted from some document in the digital library. In contrast to traditional knowledge graph querying, where all extracted information is integrated into one big knowledge graph, document-centered graph query processing ensures the validity of results through context-compatible information fusion [3]. That means that narrative graph queries are only answered in strict document contexts, i.e., by fusing statements mentioned within the scope of a single document. However, the graph-based retrieval approach also suffers from a severe limitation: queries are isomorphically matched against document graphs, i.e., all correct answers show the same level of relevance.

So, how can graph-based document representations be effectively ranked for document retrieval purposes? This paper extends our existing system by introducing novel ranking strategies that 1) intelligently exploit the structure of document graph representations and 2) effectively increase the retrieval recall through a relaxed query matching paradigm (Partial Matches) and ontological query rewriting. Moreover, our methods do not rely on supervision and can thus be deployed without requiring costly training data. In addition to our graph-based discovery system's benefits like structured literature overviews, see [5] for a comprehensive overview, this paper proposes graph-based ranking methods and query relaxation strategies to improve such a system significantly, in terms of precision and recall. We share our code, produced results and detailed topic-wise evaluation figures at GitHub[1] and Software Hertiage[2].

[1] https://github.com/HermannKroll/RankingNarrativeQueryGraphs
[2] Software Heritage ID:swh:1:dir:56036430260e2759be3ac9b72f6160fed361f503

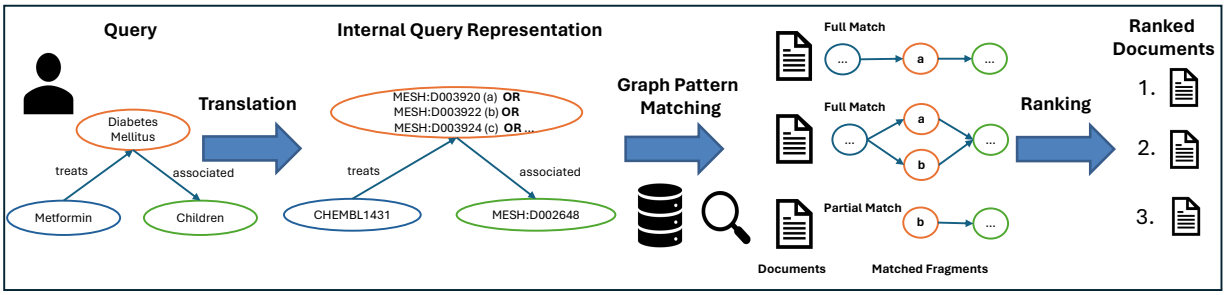

Figure 1: Systematic overview: Users formulate their information needs as graph patterns between concepts. Queries are translated and matched against document graphs. Matches are documents that match the query completely (full match) or partially (partial match). The matched documents are then ranked based on their graphs.

2 Graph-based Discovery System

Our and PubPharm's (German specialized information service for pharmacy) graph-based retrieval service, called the Narrative Service [5] (https://narrative.pubpharm.de), currently features approx. 37 million publications from the National Library of Medicine's Medline collection and 70k COVID-19 pre-prints from ZB MED's preVIEW service [8]. Users are enabled to intuitively formulate their information needs as graph patterns, i.e., conjunctions of triple-like statements (concept, interaction, concept). Correct answers to the query are all documents that contain all search statements. For the query processing, document texts are transformed into a graph representation by linking terms against biomedical concepts and extracting their interactions through the PathIE method [4]. *association* statements were extracted if two concepts were mentioned within the same sentence. Concepts were identified by deriving annotations from the PubTator service [22, 23] and performing a dictionary-based linking through vocabularies derived from ChEBML [11], Wikidata [20] and the Medical Subject Headings.

Formally, C is the set of known **concepts** (e.g., Metformin, Diabetes), and Σ is the set of known **interactions** (e.g., associated, treats, inhibits). A **statement** is an triple (c_1, p_1, c_2) with $c1, c2 \in C$ and $p_1 \in \Sigma$. Each **document** is represented by its so-called document graph, which is harvested from its title and abstract. A **document graph** $graph(d)$ is a directed, edge-labeled graph extracted from the corresponding document d. Please note that an edge could be *extracted* from several sentences of d. Each of these extractions comes with a confidence score, e.g., the applied extraction method PathIE defines **confidence** over the distance between two concepts in the grammatical structure of a sentence. The Narrative Service allows users to formulate their information needs as narrative query graphs [5]. A *narrative query* consists of a set of fact patterns. A **fact pattern** is a triple (s, p, o). The subject s and object o are either concepts from C or variables from a set $\mathcal{V}$. The fact patterns are understood as being logically connected via an AND expression. If the narrative query graph does not ask for variables, matches to the query are documents that contain all searched (s, p, o) triples in their document graph. If a narrative query graph contains variables, a document d matches the query if 1) the function $\mu_d : \mathcal{V} \rightarrow C$ substitutes the query's variables with concrete concepts from C and 2) the resulting, substituted graph with concrete concepts is supported by the document graph of d.

3 Graph-based Retrieval and Ranking

This article contains the central ideas of our ranking methods. Core steps are 1) query translation, 2) result ranking and 3) query relaxation. For a detailed description, we refer the reader to [7].

3.1 Query Translation

In this paper, we improve the query translation process as follows: First, all concepts containing the searched term in one of the synonyms are now considered relevant. The advantage is that it does not matter whether the user enters *diabetes mellitus* or *mellitus diabetes*. We implement the strategy by using a relational database table that maps terms to concepts. In the case of a *diabetes* search, the table is queried by a SQL WHERE expression: *term LIKE %diabetes%*. Suppose the user's entered term contains multiple terms. In that case, the terms are split by space and concatenated by AND operations, e.g., *diabetes mellitus* is translated into a SQL WHERE expression like *term LIKE %diabetes% AND term LIKE %mellitus%*. This strategy ensures that all entered terms are contained, but the order is not essential, making the translation easier to use, and in some cases, more robust, e.g., it does not matter whether the user searches for *diabetes mellitus* or *mellitus diabetes*. We created a trigram-based index to accelerate LIKE searches in the database. Second, we introduce a so-called *translation score* to represent how well a translated concept might represent the user's intended concept search. If the user input directly matches a synonym, it is considered a perfect concept translation. Otherwise we measured the string similarity of the user's input and the concept term.

3.2 GraphRank

Next, we introduce our graph-based ranking method **GraphRank**. Due to alternative concepts in the query expansion, a document graph might thus match with different graph parts (different subgraphs), e.g., one match might include *Diabetes Mellitus type 1* and another *type 2*. The function $matches(q, d)$ computes all distinct subgraph isomorphisms between the query q and the document graph of d. Each subgraph isomorphism maps a part of the document graph to the query. We call that matching part **fragment**, i.e., given a query q, a document d, and a fragment $f \in matches(q, d)$. The function $edges(f)$ returns edges of the fragment f, and $nodes(f)$ returns the nodes of f. In other words: A fragment is a subgraph

of the document graph that matches the query. Please note that a document can have multiple fragments because different document graph concepts can match the same query.

Features. GraphRank uses four features for ranking. In brief: (**confidence**) Each statement comes with a certain extraction confidence, i.e., how sure the system is about the statement's extraction. (**tf-idf**) Each statement has a certain tf-idf value, i.e., how often the statement is mentioned within an abstract versus how frequent the statement is across the whole document collection. The more frequent a statement appears and the more special it is with regard to the whole collection the better is its relevance. (**coverage**) Statements are interactions between concepts. Concepts might be mentioned as a side note in some abstract or are used across the whole abstract. Coverage measures the ratio of text that involves the corresponding concept, i.e., its last text position minus its first text position. (**relational similarity)** Confidence, tf-idf and coverage now allow us to score each document's edge. The neighborhood of some edge, i.e., all edges that are incoming or outgoing to the edge's subject and object could have an influence of the overall relevancy. We used the neighborhood to determine a relational similarity.

Fragment translation score. Concepts have a translation score, i.e., how well they represent the user's input (remember, users enter terms, not concept identifiers). Next, we define how well a fragment represents the user's intended information need. A translated fragment close to the user's input is ranked higher. The translation score for a fragment f is defined as:

$$translation(f) = \min(\{translation\_score(c) \mid c \in nodes(f)\}) \quad (1)$$

Weighting. Some strategies come with scores between 0 and 1, while others, e.g., tf-idf, may yield scores above 1.0. Given a document set D_r to rank, we normalize all scored fragments by their maximum score for each of our previous strategies. Let D_r be the set of all documents to rank, f be the matching fragment of document d and d the actual document to rank. We combine our four similarities $sim = [confidence, min\_tfidf, coverage, relational\_similarity]$ by weighting each one through a vector $W = [w_1, w_2, w_3, w_4]$ with $w_i \in [0, 1]$ and $w_1 + w_2 + w_3 + w_4 = 1$.

$$fscore(f, d) = translation(f, d) \cdot \sum_{w_i \in W} w_i \cdot sim_i(f, d) \quad (2)$$

Document scoring. Each document graph might have multiple fragments that match the initial query. We compute each fragment's score. Here, we multiply the fragment's *score* with its translation score. We then select the best-scored fragment to represent the overall document score for a document d:

$$\textbf{GraphRank}(q, d) = \max(\{fscore(f, d) \mid f \in matches(q, d)\}) \quad (3)$$

3.3 Query Relaxation

Partial Matches. The retrieval system enforces that relevant documents must match the full graph query (**Full Match**). We implement a **Partial Match** strategy as an extension for our discovery system, i.e., documents that matches the query partially are added to the result list. The Partial Match strategy enforces that documents that match the query fully are always placed before partial matches.

Ontological Expansion. Concepts in queries are by default expanded by their subclasses, e.g., if users search for general *diabetes*, queries will also search for particular subtypes like *diabetes type 2*. This decision was made when implementing our system [5] because it reflected the users' needs. However, going upwards in an ontology might also be helpful; for example, consider more general forms of metabolic disease. A concept c can be generalized by the $superclass(c)$ relation that retrieves all direct and transitive superclasses of c. However, each step in the ontology we make might lead to more irrelevant results. That is why we introduce a similarity score for expanded concepts: The more steps we take to generalize a concept within an ontology, the less *well-translated* is the concept in reflecting the query.

4 Evaluation

Our method GraphRank is designed to rank concept-centric narrative query graphs in the biomedical domain. As far as we know, graph-based biomedical document retrieval benchmarks do not exist. That is why we decided to focus on existing biomedical benchmarks that could be used for our purposes. For instance, the TREC Precision Medicine Series 2017-2020 [16–19] were designed as concept-centric document retrieval benchmarks. The central problem when utilizing these benchmarks is that they ask for keyword queries instead of graph queries. Consider, for example, the benchmark query *melanoma BRAF Binimetinib* that asks for three biomedical components. We assumed the predicate was not given in the benchmark and allowed any predicate between the searched concepts. With that assumption, we could generate a graph pattern like $(C_1, ?p_1, C_2) \wedge (C_2, ?p_2, C_3)$ which asks for some interaction between C_1 (all translated concepts for the first component) and C_2, and some interaction between C_2 and C_3. A document then matches the query if it contains both interactions. If a query asks for three components, we have three alternatives to connect the different components. Finally, we can generate the following graph query by using a logical disjunction over all three combinations: $[(C_1, ?p_1, C_2) \wedge (C_2, ?p_2, C_3)] \vee [(C_1, ?p_1, C_2) \wedge (C_1, ?p_2, C_3)] \vee [(C_1, ?p_1, C_3) \wedge (C_2, ?p_2, C_3)]$. Please note that we did not adjust the system's concept vocabulary for this paper. Our subsequent evaluation reveals some major limitations here, i.e., queries or terms in queries that are not reflected in our system's concept vocabulary.

In this paper, we focus on an evaluation based on TREC Precision Medicine 2020 [16]. Results for TREC PM 2017-2019 [17–19] and for TREC COVID [15] are reported in [7]. Each TREC Precision Medicine Series topic asks for a specific type of cancer. The benchmark instructions state that the more precise a document's included cancer type corresponds to the searched one, the more relevant it is. We used the Medical Subject Heading cancer ontology (see https://meshb.nlm.nih.gov/record/ui?ui=D009369) to rewrite our queries, i.e., we expand specific cancer types to general ones.

4.1 Results

Parameters. For our evaluation, we used equal weights for our GraphRank method, i.e., $w_i = 0.25$. We set predicate specificity score (see tf-idf score) based on each predicate's hierachical level in our three-level predicate taxonomy (most-specific predicates received a score of 1.0, one level higher 0.5, and the highest level

Table 1: Evaluation results of TREC-PM2020 [16] (based on 31 out of 31 topics): Recall, nDCG@k and P@k are reported at different ranks. We show the results of the Old System, GraphRank in different combinations and BM25.

Ranking Method	Recall@1000	nDCG@10	nDCG@20	nDCG@100	P@10	P@20	P@100
Old System [5]	0.31	0.37	0.37	0.36	0.42	0.38	0.21
Full Match	0.31	0.37	0.37	0.36	0.42	0.38	0.22
+ GraphRank	0.31	0.42	0.41	0.38	0.45	0.40	0.22
+ BM25	0.31	0.46	0.43	0.40	0.45	0.39	0.22
+ Ontology	**0.45**	0.33	0.33	0.37	0.41	0.36	0.24
+ Ontology + GraphRank	**0.45**	0.44	0.43	0.43	**0.48**	**0.42**	**0.25**
+ Ontology + BM25	**0.45**	**0.47**	**0.46**	**0.45**	0.47	**0.42**	**0.25**
Partial Match	0.78	0.40	0.42	0.48	0.46	0.42	0.29
+ GraphRank	0.78	0.50	0.49	0.50	**0.55**	**0.47**	0.28
+ BM25	0.78	**0.53**	**0.52**	**0.55**	0.53	**0.47**	**0.30**
+ Ontology	**0.86**	0.33	0.34	0.44	0.41	0.36	0.28
+ Ontology + GraphRank	**0.86**	0.47	0.48	0.51	0.52	**0.47**	0.29
+ Ontology + BM25	**0.86**	0.50	0.51	**0.55**	0.51	**0.47**	**0.30**
Native BM25 (Baseline)	0.79	0.48	0.46	0.49	0.48	0.42	0.28

(only associated) 0.25); see taxonomy at https://narrative.pubpharm.de/help/. The idea is that the deeper a predicate is placed in our taxonomy, the more information a predicate carries.

Baselines. We compare GraphRank to the well-known ranking strategy BM25: First, we used BM25 to rerank the documents retrieved by the graph matching paradigm (BM25 Reranking). This setup compares GraphRank to BM25 on the same set of retrieved documents. Second, we used BM25 to retrieve and rank documents without graph-based retrieval. This setup demonstrates how graph-based retrieval plus ranking performs compared to pure BM25 retrieval (BM25 Native). We used PyTerrier [10] to implement BM25.

PM2020. We decided to ignore unjudged documents because our goal is to extend the discovery system and not to outperform other strategies. Many retrieved documents were not judged in the benchmarks; see [7] for a discussion why. For PM2020, 31 out of 31 queries had a translation score above 0.9. The results for PM2020 are depicted in Table 1 which has four parts: 1) the old system [5] without the improved query translation and by sorting documents by their IDs (date) in descending order (old system), 2) using Full Match as a matching paradigm without ranking (just Full Match), plus ranking strategies (+GraphRank and +BM25) and ontological query expansion (+Ontology), 3) Partial Match without ranking, plus ranking and query expansion, and 4) the results for native BM25 retrieval. We report the Recall@1000, the normalized discounted cumulative gain (nDCG), and precision at different ranks k (@10, @20, @100). In summary, the recall of the Full Match paradigm is always below the Partial Match paradigm, which we expected. Partial Match plus ontological expansion achieved a recall of 0.86. In comparison, BM25 achieved a recall of 0.79. The type of queries can explain the high recall of BM25: Queries in PM2020 ask for a specific cancer subtype. Each subtype contained a term like *cancer* (e.g., ovarian cancer). If the exact form cannot be found in documents, BM25 will also rank documents that *just* contain the term *cancer*. In other words, BM25 performed some form of beneficial expansion here (compare it to our ontological rewriting). Partial Match + Ontology + GraphRank achieved a recall ≥ 0.9 in 19 out of 31 topics, whereas BM25 achieved nine times a recall ≥ 0.9. Concerning precision, Partial Match plus BM25 or GraphRank achieved higher scores than native BM25 retrieval (up to 7% points in P@10). Partial Match did not decrease the precision in comparison to Full Match. In contrast, it increased the precision because: First, Partial Match puts partially matched documents behind full matches in a result list (by definition). Second, for nine topics, Full Match yielded less than 20 results, decreasing the precision at rank 20. For instance, if five correct matches were found but nothing more, the precision at 20 is 0.25 by definition. Compared to performing a BM25-reranking of documents retrieved by Partial Match and Ontology, GraphRank achieved a comparable performance (slightly better/comparable for precision, but slightly worse for nDCG).

5 Conclusion

Benefits of graph-based retrieval, like entity-centric structured overviews of the literature have already been discussed in the literature [5, 6]. However, practical ranking methods for such retrieval systems, solely based on the graph-based document representation, were yet missing. In this work, we filled that gap by proposing methods for such retrieval workflows, which opens up a space for future research. Moreover, we proposed effective query relaxation and ontological rewriting that can improve recall and thus help users explore a document collection. Our methods work on an extensive digital library collection with 37M documents, do not require training data or supervision, and can be directly integrated into an existing digital library system. If queries were concept-centric and the system knew those concepts, our methods outperformed BM25. However, not all information needs could be translated successfully because concepts were missing (school closing) or lack of expressiveness (gene modifications). Future work could tackle a fallback mode for switching between graph-based and traditional text-based ranking, depending on a certain information need.

Acknowledgments

Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): PubPharm – the Specialized Information Service for Pharmacy (Gepris 267140244).

References

[1] Lucas Albarede, Philippe Mulhem, Lorraine Goeuriot, Claude Le Pape-Gardeux, Sylvain Marie, and Trinidad Chardin-Segui. 2022. Passage Retrieval on Structured Documents Using Graph Attention Networks. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022 (Lecture Notes in Computer Science, Vol. 13186)*. Springer, 13–21. https://doi.org/10.1007/978-3-030-99739-7_2

[2] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A Position-Aware Neural IR Model for Relevance Matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*. Association for Computational Linguistics, 1049–1058. https://doi.org/10.18653/v1/d17-1110

[3] Hermann Kroll, Jan-Christoph Kalo, Denis Nagel, Stephan Mennicke, and Wolf-Tilo Balke. 2020. Context-Compatible Information Fusion for Scientific Knowledge Graphs. In *Digital Libraries for Open Knowledge - 24th International Conference on Theory and Practice of Digital Libraries, TPDL 2020 (Lecture Notes in Computer Science, Vol. 12246)*. Springer, 33–47. https://doi.org/10.1007/978-3-030-54956-5_3

[4] Hermann Kroll, Jan Pirklbauer, and Wolf-Tilo Balke. 2021. A Toolbox for the Nearly-Unsupervised Construction of Digital Library Knowledge Graphs. In *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2021*. IEEE, 21–30. https://doi.org/10.1109/JCDL52503.2021.00014

[5] Hermann Kroll, Jan Pirklbauer, Jan-Christoph Kalo, Morris Kunz, Johannes Ruthmann, and Wolf-Tilo Balke. 2024. A discovery system for narrative query graphs: entity-interaction-aware document retrieval. *Int. J. Digit. Libr.* 25, 1 (2024), 3–24. https://doi.org/10.1007/S00799-023-00356-3

[6] Hermann Kroll, Florian Plötzky, Jan Pirklbauer, and Wolf-Tilo Balke. 2022. What a Publication Tells You—Benefits of Narrative Information Access in Digital Libraries. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries* (Cologne, Germany) *(JCDL '22)*. Association for Computing Machinery, New York, NY, USA, Article 9, 8 pages. https://doi.org/10.1145/3529372.3530928

[7] Hermann Kroll, Pascal Sackhoff, Timo Breuer, Ralf Schenkel, and Wolf-Tilo Balke. 2024. Ranking Narrative Query Graphs for Biomedical Document Retrieval (Technical Report). arXiv:identifier tba, PDF available at [cs.DL] https://github.com/HermannKroll/RankingNarrativeQueryGraphs

[8] Lisa Langnickel, Roman Baum, Johannes Darms, Sumit Madan, and Juliane Fluck. 2021. COVID-19 preVIEW: Semantic Search to Explore COVID-19 Research Preprints. In *Public Health and Informatics*. IOS Press, Amsterdam, the Netherlands, 78–82. https://doi.org/10.3233/SHTI210124

[9] Zhengdong Lu and Hang Li. 2013. A Deep Architecture for Matching Short Texts. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*. 1367–1375. https://proceedings.neurips.cc/paper/2013/hash/8a0e1141fd37fa5b98d5bb769ba1a7cc-Abstract.html

[10] Craig Macdonald and Nicola Tonellotto. 2020. Declarative Experimentation in Information Retrieval using PyTerrier. In *ICTIR '20: The 2020 ACM SIGIR International Conference on the Theory of Information Retrieval, 2020*. ACM, 161–168. https://doi.org/10.1145/3409256.3409829

[11] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, María Gordillo-Marañón, Fiona Hunter, Laura Junco, Grace Mugumbate, Milagros Rodriguez-Lopez, Francis Atkinson, Nicolas Bosc, Chris J Radoux, Aldo Segura-Cabrera, Anne Hersey, and Andrew R Leach. 2018. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research* 47, D1 (11 2018), D930–D940. https://doi.org/10.1093/nar/gky1075

[12] Sunil Mohan, Nicolas Fiorini, Sun Kim, and Zhiyong Lu. 2018. A Fast Deep Learning Model for Textual Relevance in Biomedical Information Retrieval. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018*. ACM, 77–86. https://doi.org/10.1145/3178876.3186049

[13] Rodrigo Frassetto Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *CoRR* abs/1904.08375 (2019). arXiv:1904.08375 http://arxiv.org/abs/1904.08375

[14] Ronak Pradeep, Rodrigo Frassetto Nogueira, and Jimmy Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *CoRR* abs/2101.05667 (2021). arXiv:2101.05667 https://arxiv.org/abs/2101.05667

[15] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen M. Voorhees, Lucy Lu Wang, and William R. Hersh. 2021. Searching for scientific evidence in a pandemic: An overview of TREC-COVID. *J. Biomed. Informatics* 121 (2021), 103865. https://doi.org/10.1016/J.JBI.2021.103865

[16] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, Steven Bedrick, and William R. Hersh. 2020. Overview of the TREC 2020 Precision Medicine Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020 (NIST Special Publication, Vol. 1266)*. National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.PM.pdf

[17] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, and Alexander J. Lazar. 2018. Overview of the TREC 2018 Precision Medicine Track. In *Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018 (NIST Special Publication, Vol. 500-331)*. National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec27/papers/Overview-PM.pdf

[18] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, Alexander J. Lazar, and Shubham Pant. 2017. Overview of the TREC 2017 Precision Medicine Track. In *Proceedings of The Twenty-Sixth Text REtrieval Conference, TREC 2017 (NIST Special Publication, Vol. 500-324)*. National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec26/papers/Overview-PM.pdf

[19] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, Alexander J. Lazar, Shubham Pant, and Funda Meric-Bernstam. 2019. Overview of the TREC 2019 Precision Medicine Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019 (NIST Special Publication, Vol. 1250)*. National Institute of Standards and Technology (NIST). https://trec.nist.gov/pubs/trec28/papers/OVERVIEW.PM.pdf

[20] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. https://doi.org/10.1145/2629489

[21] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *CoRR* abs/2212.03533 (2022). https://doi.org/10.48550/arXiv.2212.03533 arXiv:2212.03533

[22] Chih-Hsuan Wei, Alexis Allot, Robert Leaman, and Zhiyong Lu. 2019. PubTator central: automated concept annotation for biomedical full text articles. *Nucleic Acids Research* 47, W1 (05 2019), W587–W593. https://doi.org/10.1093/nar/gkz389

[23] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research* 41, W1 (05 2013), W518–W522. https://doi.org/10.1093/nar/gkt441