UNIVERSITY OF CALIFORNIA, SAN DIEGO

Information Retrieval in Biomedical Research: From Articles to Datasets

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Bioinformatics and Systems Biology
with a Specialization in
Biomedical Informatics

by

Wei Wei

Committee in charge:

Lucila Ohno-Machado, Chair
Xiaoqian Jiang, Co-Chair
Dina Demner-Fushman
Chun-Nan Hsu
Hyeoneui Kim
Lawrence Saul

2017

The Dissertation of Wei Wei is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I want to thank Dr. Lucila Ohno-Machado, my advisor, my role model, and my friend. In my difficult times, she gave me the most needed help, and maintained her confidence in me. To my knowledge, it is extremely rare for a PhD advisor to revise a student's manuscripts and proposals so thoroughly, but she did. Also, one thing that will never fade from my memory is the moment, in Lucila's office at "the hut," when she informed me that she had already obtained Dina's signature for my proposal. I could not have completed the PhD training and the dissertation without her advice, support and encouragement.

Next, I would like to thank my committee members, Dr. Dina Demner-Fushman, Dr. Chun-Nan Hsu, Dr. Xiaoqian Jiang, Dr. Hyeonui Kim, and Dr. Lawrence Saul, for their long-term support and guidance.

In addition, my colleagues and friends have helped throughout my studies: Dr. Shuang Wang helped me find my first project, offering me advice and discussions about research and non-research topics; Dr. Rui Zhang, Dr. Qi Li, Dr. Ya Hua, Dr. Wei Wang, Dr. Tsung-Ting Kuo, and Dr. Ko-wei Lin always gave me endless support and guidance, for my research and my career plans.

I would also like to extend much thanks to my friends, Yupeng He, Zhanglong Ji, Joanne Liu, Eric Levy, and Tyler Bath. I cannot image life without you. I miss the times when we completed a homework assignment submission just a few minutes before the deadline, and the dim sum hours we had together. Thank Chenxiao Ling, for her help on my dissertation and the happiness she brought into my life. Thank B.J. (Byungji) Kim, my writing center tutor and friend, literally the co-

author of this dissertation. It has been more effective to improve my writing skills by working with you than taking a writing class.

Finally, thank you, Mom and Dad, for always being my strongest support. This is my dissertation, but it is also yours.

Chapter 4, in part, has been submitted for publication of the material as it may appear in Finding Relevant Biomedical Datasets: the UC San Diego Solution for the 2016 bioCADDIE Retrieval Challenge. Wei, Wei; Ji, Zhanglong; He, Yupeng; Zhang, Kai; Ha, Yuanchi; Li, Qi; Ohno-Machado, Lucila, Database(Oxford), 2017. The dissertation author was the primary investigator and the author of this paper.

Chapter 6, in part, is a reprint of the material as it appears in Finding Related Publications: Extending the Set of Terms Used to Assess Article Similarity. Wei, Wei; Marmor, Rebecca; Singh, Siddharth; Wang, Shuang; Demner-Fushman, Dina; Kuo, Tsung-Ting; Hsu, Chun-Nan; Ohno-Machado, Lucila, AMIA Summits on Translational Science Proceedings, 2016. The dissertation author was the primary investigator and author of this paper.

VITA

2009 Bachelor of Science, Zhejiang University, Hangzhou, China

2011 Master of Science, University of Pittsburgh

2017 Doctor of Philosophy, University of California, San Diego

Publications

Wei Wei, Zhanglong Ji, Yupeng He, Kai Zhang, Yuanchi Ha, Qi Li, Lucila Ohno-Machado. Finding Relevant Biomedical Datasets: the UC San Diego Solution for the 2016 bioCADDIE Retrieval Challenge. *Database(Oxford)*, in submission.

Wei Wei, Rebecca Marmor, Siddharth Singh, Shuang Wang, Dina Demner-Fushman, Tsung-Ting Kuo, Chun-Nan Hsu, Lucila Ohno-Machado. Finding Related Publications: Extending the Set of Terms Used to Assess Article Similarity. *AMIA Summits on Translational Science Proceedings.* 2016

Wei Wei, Dina Demner-Fushman, Shuang Wang, Xiaoqian Jiang, Lucila Ohno-Machado. Ranking Medical Subject Headings using a Factor Graph Model. *AMIA Summits on Translational Science Proceedings.* 2015

Mindy Ross*, **Wei Wei***, Lucila Ohno-Machado. Big Data and the Electronic Health Record. *IMIA Yearbook of Medical Informatics* 2014 (*co-first authors)

Wei Wei, Shyam Visweswaran, and Gregory F. Cooper. The Application of Naive Bayes Model Averaging to Predict Alzheimer's disease from Genome-Wide Data. *J Am Med Inform Asso* 2011;18:370-375

Fields of Study

Major field: Bioinformatics and Systems Biology with a Specialization in Biomedical Informatics

ABSTRACT OF THE DISSERTATION

Information Retrieval in Biomedical Research:
From Articles to Datasets

By

Wei Wei

Doctor of Philosophy in Bioinformatics and Systems Biology
with a Specialization in
Biomedical Informatics

University of California, San Diego, 2017

Professor Lucila Ohno-Machado, Chair
Professor Xiaoqian Jiang, Co-Chair

Information retrieval techniques have been applied to biomedical research for a variety of purposes, such as textual document retrieval and molecular data retrieval. As biomedical research evolves over time, information retrieval is also constantly facing new challenges, including the growing number of available data, the emerging new data types, the demand for interoperability between data

resources, and the change of users' search behaviors. To help solve the challenges, I studied three solutions in my dissertation: (a) using information collected from online resources to enrich the representation models for biomedical datasets; (b) exploring rule-based and deep learning-based methods to help users formulate effective queries for both dataset retrieval and publication retrieval; and (c) developing a "retrieval plus re-ranking" strategy to identify relevant datasets, and rank them using customized ranking models.

In a biomedical dataset retrieval study, we developed a pipeline to automatically analyze users' free-text requests, and rank relevant datasets using a "retrieval plus re-ranking" strategy. To improve the representation model of biomedical datasets, we explored online resources and collected information to enrich the metadata of datasets. The rule-based query formulation module extracted keywords from users' free-text requests, expanded the keywords using NCBI resources, and finally formulated Boolean queries using pre-designed templates. The novel "retrieval plus re-ranking" strategy captured relevant datasets in the retrieval step, and ranked datasets using the customized relevance scoring functions that model unique properties of the metadata of biomedical datasets. The solutions proved to be successful for biomedical dataset retrieval, and the pipeline achieved the highest inferred Normalized Discounted Cumulative Gain (infNDCG) score in the 2016 bioCADDIE Biomedical Dataset Retrieval Challenge.

In a biomedical publication retrieval study, we developed the eXtended PubMed Related Citation (XPRC) algorithm to find similar articles in PubMed. Currently, similar articles in PubMed are determined by the PubMed Related

Citation (PRC) algorithm. However, when the distributions of term counts are similar between articles, the PRC algorithm may conclude that the articles are similar, even though they may be about different topics. On the other hand, when two articles discuss the same topic but use different terms, the PRC algorithm may miss the similarity. For the above problem, we implemented a term expansion method to help capture the similarity. Unlike popular ontology-based expansion methods, we used a deep learning method to learn distributed representations of terms over one million articles from PubMed Central, and identified similar terms using the Euclidean distance between distributed representation vectors. We showed that, under certain conditions, using XPRC can improve precision, and helps find similar articles from PubMed.

In conclusion, information retrieval techniques in biomedical research have helped researchers find desired publications, datasets, and other information. Further research on developing robust representation models, intelligent query formulation systems, and effective ranking models will lead to smarter and more friendly information retrieval systems that will further promote the transformation from data to knowledge in biomedicine.

1 Introduction

1.1 Problem statement

Information retrieval techniques have been applied to biomedical research for decades[1–3]. As biomedical research evolves over time, information retrieval is also constantly facing new challenges. Here, I highlight four major challenges: the growing amount of available data; new emerging data types; the need for interoperability between data resources; and the constant change of users' search behaviors.

1.1.1 Growing number of literature and datasets

The number of publicly available biomedical publications and datasets has grown exponentially in the last decade (see Table 1). From 2007 to 2017, biomedical publications indexed in PubMed[1] have increased more than 1.5 times, gene expression samples in the Gene Expression Omnibus database (GEO)[2] have grown 15 times, registered clinical trial studies in ClnicalTrials.gov[3] four times, and macromolecular structures in the Protein Data Bank (PDB)[4] database twofold.

[1] https:/ /www.ncbi.nlm.nih.gov/pubmed
[2] https://www.ncbi.nlm.nih.gov/geo
[3] https://www.clinicaltrials.gov
[4] http://www.rcsb.org/pdb/home/home.do

Table 1. The increasing number of biomedical publications and datasets.

Database	Data type	2007	2017	Release date
PubMed	literature	16,785,314	25,378,350	1997
GEO	gene expression samples	131,416	2,052,685	2002
ClinicalTrials.gov	registered studies	49,241	234,336	2000
PDB	macromolecular structures	47,616	125,795	1971

1.1.2 Emerging data types

Over the last two decades, there were significant increases in the diversity of available biomedical data types for two reasons: (a) new technologies resulted in new types of data, such as the next generation sequencing (NGS) data[4]; (b) information technologies made biomedical data easier to access, such as medical images[5,6] and documents[7,8].

From late 1990s to early 2010s, microarray technology progressed rapidly and was applied for various purposes such as gene expression analysis, genotyping, and medical diagnosis[9]. However, upon the arrival of NGS technologies, DNA microarray has gradually faded away. For gene expression studies, a microarray can only inspect regions for which probes have been designed, while NGS can sequence the entire target genome. For genotyping studies, a microarray can only inspect SNPs they contain (mostly common variants), while NGS can inspect both common and rare variants. For diagnosis, given the ability to identify traces of cell-free DNA from blood samples, NGS is already clinically used for fetal trisomy screening during pregnancy[10], while the microarrays have never been efficient for such tasks.

Since the 2000s, medical images have become available for various purposes. For example, the Cancer Imaging Archive (TCIA)[5] provides access to computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET) images for common cancer types; the Open Access Series of Imaging Studies (OASIS)[6] provides access to neuroimaging MRI data for Alzheimer's disease studies; the Open sharing of Functional Magnetic Resonance Imaging (OpenfMRI)[7] provides access to raw functional MRI (fMRI) data.

Medical documents have also become more accessible in the last decade. In 2000, the National Library of Medicine (NLM) established ClinicalTrials.gov, a clinical trial database and online registry. Since then, more than 200,000 trials have been registered and the information has been made available to the public. PhysioBank[8] is another effort to collect and share clinical data. It contains over 90,000 recordings of digitized physiologic signals and time series, such as cardiopulmonary and neural signals, from healthy subjects and patients with a variety of conditions[8].

Beyond the above examples, a variety of new data types have emerged in the biomedical domain, such as epigenetic data[11], 3D genome structure (Hi-C data)[12], and gene regulatory network[13] data.

[5] http://www.cancerimagingarchive.net
[6] http://www.oasis-brains.org
[7] https://openfmri.org
[8] https://physionet.org/physiobank/about.shtml

1.1.3 Interoperability between resources

Biomedical datasets are stored in various data resources that fulfill different functions. The diversity makes it difficult for users to find these datasets. For example, a user who studies the role of p53 in breast cancer may have to search through various literature databases, genome databases, pathway databases, and pharmaceutical databases to collect the desired information because each database contains only a subset of the relevant information. To retrieve desired information from a data resource, users need to formulate effective queries, which require knowledge of the research domain and the retrieval system; the presence of more resources implies more work for users. Therefore, users sometimes avoid the use of multiple resources.

1.1.4 Change of users' search behaviors

Formulating queries used to be the work of professional librarians, but the traditional service is no longer as popular, since current users are often self-sufficient[14]. In an NIH-wide survey[15], 95% of the respondents agreed that the most common way they obtained information was through independent search; among them, 91% of respondents preferred to make self-guided queries. However, an early study showed that an average third-year medical student needs 14 separate queries to get the desired information[16], while a recent querying behavior study showed that experienced health librarians need an average of three queries to answer a question[17].

1.2 Solutions

The growing amount of heterogeneous data requires information retrieval systems to be effective in identifying relevant objects from a large set of candidates, and to be accurate in estimating the relevance of matched objects. Emerging new data types need compatible representation models to characterize the objects, and this may be achieved through expansion of existing models or development of new models. The interoperability needs and the constant change of users' behavior require information retrieval systems to be more intelligent in understanding users' requests and in helping users formulate effective queries. Together, these challenges call for a more robust and more intelligent information retrieval system for biomedical data than what is available today.

This dissertation provides a comprehensive review of current biomedical research applications of information retrieval techniques, and introduces some solutions to the above challenges by:

1. Using information collected from external resources to enrich the representation of biomedical datasets,
2. Exploring rule-based and deep learning-based methods to help users formulate effective queries for both dataset retrieval and publication retrieval, and
3. Developing a "retrieval plus re-ranking" strategy to identify relevant datasets and rank them using customized relevance metrics.

1.3 Dissertation organization

This dissertation is organized as follows. Chapter 2 covers the background of general information retrieval. Chapter 3 introduces the biomedical dataset.

Chapter 4 presents a retrieval strategy and a retrieval pipeline for biomedical datasets. Chapter 5 discusses the indexing of biomedical articles. Chapter 6 presents a PubMed similar article retrieval method. Lastly, Chapter 7 summarizes the dissertation, proposes methods to solve two problems encountered in biomedical dataset retrieval, and concludes the dissertation.

2 Background of general information retrieval

2.1 Overview

In the 1960s, Gerard Salton defined information retrieval as[18]: "a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information." Half a century later, this definition remains valid for modern information retrieval, as it covers all essential functions of an information retrieval system, despite expansion of retrievable information, such as genome sequences and 3D structures. Also in the 1960s, Cyril Cleverdon proposed a new evaluation methodology that did not require exhaustive manual judgments of relevance for all resources for each information request[19]. *Relevance*, the core concept of the evaluation methodology, measures how retrieved documents satisfy users' information needs.

The emergence of the SMART retrieval system[1] and the development of Cranfield evaluation methodology[19] indicated the beginning of the rapid development of automatic information retrieval methods[20]. Of course, the types of documents in information retrieval have vastly expanded to include web pages, emails, research articles, books, and news reports, as well as images, videos, gene sequences, etc.[21]. In particular, non-text objects have attracted plenty of attention, such as biomedical data[22–24], image and video[25–27], audio and music[28,29], and geographical data[30]. While previous methods for retrieval of non-text objects relied on textual metadata, new approaches have turned to the contents

themselves for content-based retrieval. In this chapter, the retrieval objects should be considered textual documents, unless otherwise specified.

2.2 Text transformation

Depending on the type, textual documents may be structured or unstructured. Structured textual documents such as electronic medical records (EMR) are composed of multiple fields and each field may contain controlled vocabulary terms. Unstructured text documents are independent of such semantically overt organization.

The unstructured nature of textual document contents makes text transformation an essential step in document retrieval. Text transformation parses documents to recognize structural elements, decide the document unit, and decode character sequences. Frequently used text transformation techniques include tokenization, stop word removal, and stemming.

2.2.1 Tokenization

A term is a word or phrase used to describe a thing or to express a concept. People create documents using terms as the building blocks. However, in information retrieval, token, rather than term is the atomic unit of documents and queries. Manning et al. provide a strict definition: “a token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing”[31].

Tokenization splits a character sequence into tokens. A trivial option is to split character sequences at the white space. There are many tricks in tokenization, and there is no universal solution for all applications. For example, in news report

texts, a period may indicate a split; in medical documents, the period may frequently appear inside tokens such as “Dr. Joe”. Various tokenization tools have been developed for different purposes, such as the Penn Treebank tokenizer[32] for general English text and BioTokenizer[33] for biomedical text. The same tokenization method must be applied to both the documents and the queries to enable the identification of matched documents.

2.2.2 Stop word removal

Stop words are extremely common words in a document collection and “appear to be of little value in helping selecting documents matching a user need”[31]. The extremely common words are typically function words that help form sentence structure, but contribute little to the description of the topics of the text[21]. For example, “the”, “a”, and “of” are usually stop words in a collection of English news reports. While sorting terms by their frequencies can lead to the generation of a list of stop word candidates, manual effort is often required for an accurate list. Although certain high frequency words are commonly used in different fields, the complete list of stop words are different from domain to domain. For instance, a biomedical stop word list provided by the National Library of Medicine (NLM)[9] may be very different from a list of stop words in legal documents[10]. The general trend of the size of stop words has been from 200-300 terms, to a very small list (7-12 terms), or no stop list[31]. In fact, web search engines often do not use stop lists[31].

[9] https://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T.stopwords/
[10] http://www.leginfo.ca.gov/help/stopwords.html

2.2.3 Stemming

Stemming reduces inflectional and derivationally related forms of a word to a common base form[31]. For example: operating, operation, operational all become "operat"; democracy, democratic, democratization all become "democra". Generally, a stemming algorithm (e.g. Potter's algorithm[34]) uses heuristic methods to remove the end of words and the derivational suffixes. Stemming increases the likelihood that terms in queries and documents are matched, and generally produces small improvements in ranking effectiveness of matched documents[21]. However, aggressive stemming may result in ambiguity and mismatch.

2.3 Information need and query

Information need and query have been defined from multiple perspectives. Robert Taylor[35] provided a definition in the scenario of "at the library reference desk". The definition split an information need into four levels: (a) the visceral need, which is actual but vague, probably inexpressible; (b) the conscious need, which is ambiguous and rambling; (c) the formalized need, which is a qualified and rational statement of the question; and (d) the compromised need, which is the query presented to the information retrieval system.

Nicholas Belkin[36,37] defined information need using the Anomalous State of Knowledge (ASK) based on Taylor's work[35]. An ASK is a situation where "the user realizes that there is an anomaly in that state of knowledge with respect to the problem faced"[36]. An information need is triggered off when a user has an ASK, i.e. "from a recognized anomaly in the user's state of knowledge concerning some

topic and situation and that, in general, the user is unable to specify precisely what is needed to resolve that anomaly"[37].

Brenda Dervin[38] explained information need using a "sense-making metaphor": humans move along through time and space; when they encounter a gap, an information need arises.

A query is the compromised need in Taylor's[35] definition. A query may not perfectly convey the actual information need; in fact, a query often must be refined repeatedly to help retrieve documents that satisfy a user's information need.

2.4 Manual indexing process

It is extremely inefficient to scan all the raw documents and then retrieve relevant answers to queries when provided with a large collection of text documents. Therefore, an efficient data structure is required to represent documents, and to enable fast retrieval. The indexing process is designed for the purpose of: representing documents using index terms, and enabling retrieval using inverted index, which is currently accepted as the most efficient structure for supporting ad hoc retrieval[11] [31]. Figure 1 provides an overview of the indexing process.

[11] In an ad hoc retrieval task, a system aims to provide documents from within the collection that are relevant to an arbitrary user information need, communicated to the system by means of a one-off, user-initiated query[31].

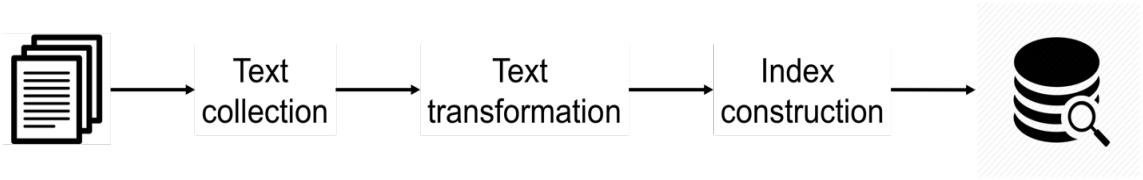


Figure 1. Indexing process. This is an overview of the indexing process for text documents. Text documents may be research articles, clinical notes, email, news reports, etc. The text collection module collects textual contents, the text transformation module applies various actions, such as stemming on the raw textual contents, to produce tokens. The index construction module builds the actual indices as the output of this pipeline.

The indexing process transforms documents using index terms, which are generated by the text transformation operations. The simplest index terms are single words in documents, and may be complicated objects rather than text strings. For example, an index term may come with its frequency in each document, its positions in each document, and the total number of documents that contain it. Such information is required for calculating term weights and ranking documents in the retrieval process. After the text transformation, the index terms are often in a set of stemmed, lower case, and non-stop-word tokens, called the index vocabulary. An index vocabulary can also include named entities identified from document contents and the metadata, such as dates, authors, and keywords.

When index terms are ready, the indexing process can build an inverted index by converting document-index term (doc-term) pairs into term-doc pairs. Figure 2 provides an example to illustrate this conversion. Therefore, given a term from a user's query, the retrieval system can efficiently locate documents containing this term and avoid scanning all the documents. Various algorithms are available for constructing inverted indices, such as blocked sort-based indexing

(BSBI)[31], single-pass in-memory indexing (SPIMI)[39], and sort-based multiway merge[40].

Doc 1: We know what we are, but know not what we may be.
Doc 2: A fool thinks himself to be wise, but a wise man knows himself to be a fool.

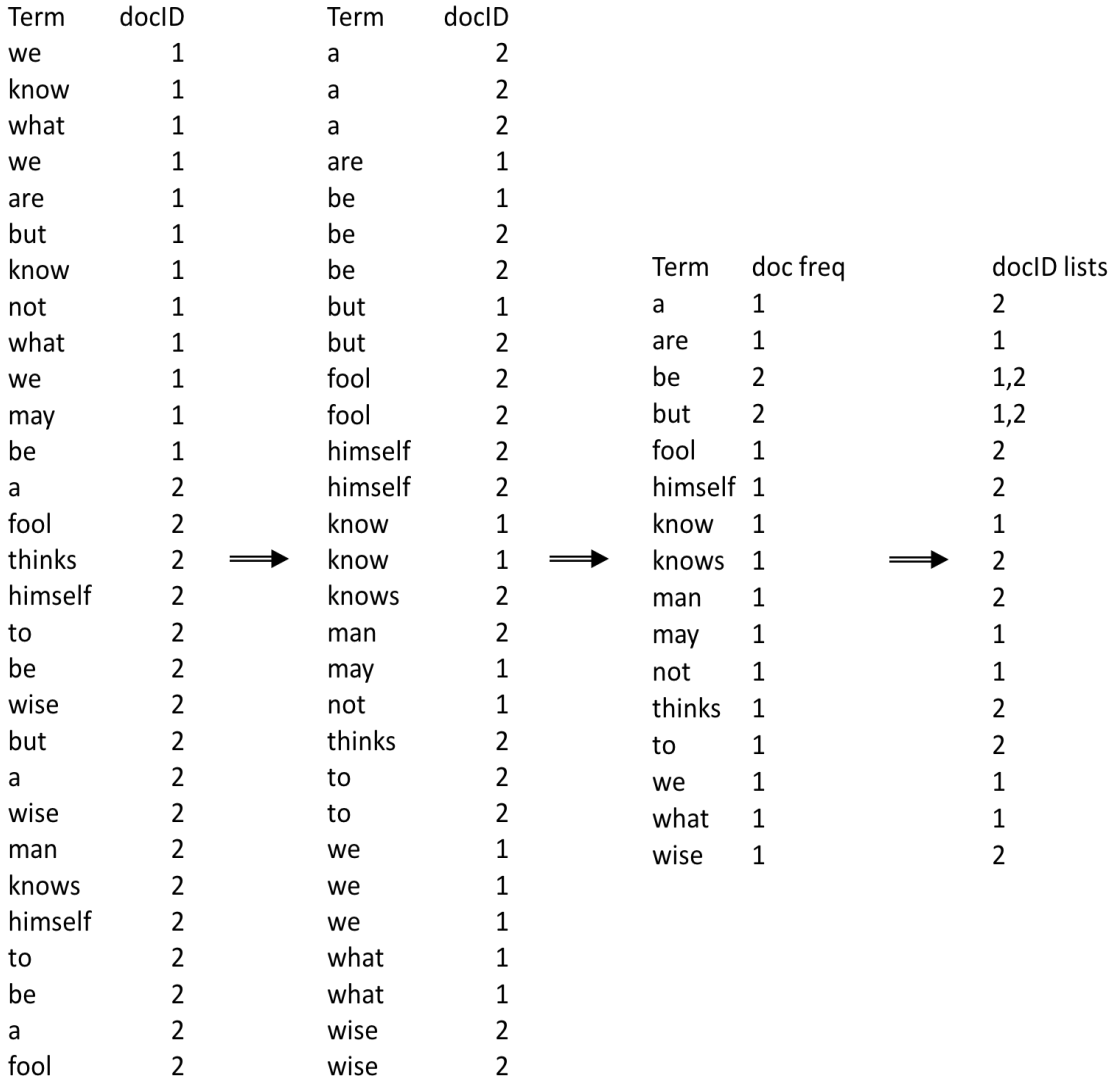


Figure 2. An example of building an inverted index. Modified from Figure 1.4 in "Introduction to information retrieval"[31]. In the first step, doc-term pairs are transformed into term-doc pairs. After that, the document frequency of each term is counted.

2.5 Retrieval models

In an ad hoc retrieval task, which is the most standard information retrieval task, users generate queries from arbitrary information needs; a retrieval model takes the queries and returns relevant documents from the collection.

A retrieval model is a representation of the process of matching a query and a document[41]. Its basic function is to identify relevant documents for given queries. Moreover, most retrieval models also rank matched documents according to their relevance to the given queries.

The Boolean retrieval model is the first information retrieval model[42] based on Boolean logic and classic set theory. This retrieval model treats both a document and a query as the bag of words model, which ignores the ordering of the terms. Terms in queries are combined using Boolean logic operations AND, OR, and NOT. The Boolean retrieval model finds well-matched documents that contain terms in the queries, but it cannot rank matched documents. Despite the simplicity, the Boolean model was the main option for large collections of documents from the 1960s to the early 1990s[31].

2.5.1 The vector space model

The vector space model[43] identifies documents that share terms with a query, then ranks the documents according to the cosine distance between the document and the query. In this model, documents and queries are represented using vectors in a Euclidean space, $\vec{d} = (t_1, t_2, \dots, t_{\mathrm{m}})$, where each value t_i measures the relative importance of the associated index term, and m is the size

of the index vocabulary. Both the documents and the queries are represented in the same space (i.e., a document $\vec{d} = (d_1, d_2, \ldots, d_m)$ and a query $\vec{q} = (q_1, q_2 \ldots, q_m)$), therefore, a dot product between two vectors $\vec{d} \cdot \vec{q} = \sum_{i=1}^{m} d_i \cdot q_i$ produces the cosine distance between $\vec{d}$ and $\vec{q}$, which is an intuitive measure of similarity.

The major challenge for the vector space model is in determining the relative importance of an index term in a document or a query, i.e., term weighting. Finding a solution is non-trivial, and a significant number of term weighting methods have been studied since the 1970s[42]. Term frequency-inversed document frequency (TF-IDF)[44,45] was the most successful term weighting method before 1990s. TF-IDF and its derived methods consider the frequency of a term t, in a document d (i.e., term frequency, TF), and the number of documents in the collection that contains term t (i.e., document frequency, DF). A typical TF-IDF formula is,

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \quad [1]$$

$$idf_t = \log \frac{N}{df_t} \quad [2]$$

where $tf_{t,d}$ is the frequency of t in d; idf_t is the inversed document frequency of t in the collection; N is the number of documents in the collection; and df_t is the number of documents containing t. The formula favors terms that are observed in a small number of documents, and that occur frequently in the given document.

Therefore, if a term appears in every document in the collection, it lowers the TF-IDF score; if a term occurs many times in a document, it increases the score.

2.5.2 Probabilistic retrieval models

The probabilistic retrieval model is another genre of retrieval models. The models are built on probability theory, and use different methods to calculate the probability of relevance between queries and documents.

2.5.2.1 Okapi BM25[46,47]

Okapi BM25 is one of the most widely used probabilistic retrieval models. By the time BM25 had been developed, there were two main problems to be solved for probabilistic retrieval models. The first problem was the difficulty in including various variables that may affect the retrieval performance when dealing with models that specify exact formulas. The second problem was the lack of guidance in variable selection for the empirical ad hoc models[46].

Okapi BM25 was designed to provide an exact but intractable formula, and to use it to suggest a simpler formula[46,47]. The model was built on the linked dependence assumption[48], the eliteness assumption[46], the Robertson/Sparck Jones term weighting formula[49], and the 2-Poisson model[50]. Robertson and Zaragoza provide a comprehensive review on the development of BM25[51]. The standard BM25 formula[47] is

$$score(\vec{d},\vec{q}) = \sum_{t\in q}\left[w^{RSJ}\times\frac{(k_1+1)tf_{td}}{k_1\left((1-b)+b\left(\frac{L_d}{L_{ave}}\right)\right)+tf_{td}}\times\frac{(k_3+1)\mathrm{t}f_{tq}}{k_3+tf_{tq}}\right] \qquad [3]$$

$$w^{RSJ} = log\frac{(r+0.5)(N-n-R+r+0.5)}{(R-r+0.5)(n-r+0.5)} \quad [4]$$

where $\frac{(k_1+1)tf_{td}}{k_1\left((1-b)+b\left(\frac{L_d}{L_{ave}}\right)\right)+tf_{td}}$ is the document term weight; $\frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}$ is the query term weight; $k_1\left((1-b)+b\left(\frac{L_d}{L_{ave}}\right)\right)$ is the document length normalization factor, where L_d is the document length, and L_{ave} is the average document length in the collection. Variable k_1 is a positive tuning parameter for the document term frequency scaling, variable k_3 is a positive tuning parameter for the term frequency scaling of the query, and variable b $(0 \leq b \leq 1)$ determines the scale of document length. w^{RSJ} is the Robertson/Sparck Jones term weight, where r is the number of relevant documents that contain the term, R is the number of relevant documents, n is the number of documents that contain the term, N is the number of documents, $R-r$ is the number of relevant documents that do not contain the term, $N-n-R+r$ is the number of non-relevant documents that do not contain the term, $n-r$ is the number of non-relevant documents that contain the term.

In summary, BM25 is an effective probabilistic term weighting model, and it has been widely used by the scientific community, e.g. in the Text Retrieval Conference (TREC)[12]. However, BM25 lacks theoretical support and is difficult to modify for further applications.

[12] http://trec.nist.gov

2.5.2.2 PubMed related citation algorithm (PRC)[52]

The PubMed related citation (PRC) algorithm is a probabilistic retrieval model designed for the MEDLINE records of biomedical articles indexed by PubMed. Given a MEDLINE record that a user has indicated interest in, the PRC algorithm retrieves other MEDLINE records that are similar in terms of the topics or concepts, and ranks them based on the probability of the user's interest [52].

The PRC algorithm predicts $P(d_2|d_1)$, which is the conditional probability that a reader will be interested in an unseen article d_2, given that the reader shows an interest in article d_1. In the following scenario, d_1 is a query, and d_2 is a document that is potentially relevant. According to the authors, "for computational tractability, we make the simplifying assumption that each term in a document represents a topic (that is, each term conveys an idea or concept)". Also, the authors assumed "single-word terms, as opposed to potentially complex multi-word concepts", to satisfy the requirement that the set of topics be exhaustive and mutually-exclusive.

The weight of a term t in either article d is,

$$w_{t,d} = \frac{\sqrt{idf_t}}{1 + \left(\frac{\mu}{\lambda}\right)^{k-1} e^{-(\mu-\lambda)l}} \quad [5]$$

where idf_t is the inverse document frequency of t; k is the term frequency of t in d; l is the word count of d; λ is the expected occurrence of t when t is the topic of d; μ is the expected occurrence of t when t is not the topic of article d. λ and μ were calculated using an extensive tuning approach. Based on the exactly

matched terms (i.e., the same terms appear in two articles) and term weights, the similarity score of two articles d_1 and d_2 is,

$$P(d_2|d_1) = \sum_{t=1}^{N} w_{t,d_1} * w_{t,d_2} \quad [6]$$

where N is the number of matched terms in d_1 and d_2.

In summary, the PRC algorithm is a successful retrieval model for MEDLINE records, as it effectively captures the relatedness between articles.

2.5.2.3 The term dependence model (TDM)[53,54]

Term dependencies are frequently observed in text. For example, in biomedical informatics literature, if “electronic” occurs in a sentence, it is likely that “medical records” will occur nearby. However, many retrieval models assume some form of independence among terms because it is not trivial to model term dependencies. The term dependence model provides a framework to model the dependencies among multiple terms in proximity.

TDM models term dependencies via Markov random fields[53]. It considers the unigrams, ordered bigrams, and unordered bigrams in text, and uses the bigram component to embed the context information of matched terms. Accordingly, the TDM score is composed of three parts, one for the match of unigrams, one for the match of ordered bigrams, and one for the match of unordered bigrams within a window of size 8 (details in Formula 7-10). The pre-determined weights ($\lambda_T = 0.8,\ \lambda_O = 0.1, \lambda_U = 0.1$) of the three parts are robust and near-optimal across a wide range of retrieval tasks [54].

$$P(d|q) =$$

$$\lambda_T \sum_{q_i \in q} f_T(q_i, d) + \lambda_O \sum_{q_i, q_{i+1} \in q} f_O(q_i, q_{i+1}, d) + \lambda_U \sum_{q_i, q_{i+1} \in q} f_U(q_i, q_{i+1}, d) \quad [7]$$

$$f_T(q_i, d) = \log \left[\frac{tf_{q_i,d} + \mu \frac{cf_{q_i}}{|C|}}{|d| + \mu} \right] \quad [8]$$

$$f_O(q_i, q_{i+1}, d) = \log \left[\frac{tf_{\#1(q_i,q_i+1),d} + \mu \frac{cf_{\#1(q_i,q_{i+1})}}{|C|}}{|d| + \mu} \right] \quad [9]$$

$$f_U(q_i, q_{i+1}, d) = \log \left[\frac{tf_{\#\text{uw8}(q_i,q_i+1),d} + \mu \frac{cf_{\#uw8(q_i,q_{i+1})}}{|C|}}{|d| + \mu} \right] \quad [10]$$

In Formula 7, d is a document; q is a query; q_i, q_{i+1} are two consecutive terms in the query; $f_T(q_i, d)$ is the weight of term q_i in d; $f_O(q_i, q_{i+1}, d)$ is the weight of the exact phrase "$q_i\ q_{i+1}$" in d; $f_U(q_i, q_{i+1}, d)$ is the weight of unordered terms q_i and q_{i+1} in a window of size eight in d. In Formulas 8, 9, and 10, $tf_{q_i,d}$ is the number of times q_i has an exact match in d; cf_{q_i} is the number of times q_i matches in the entire document collection; $|d|$ is the length of d; $|C|$ is the total length of the collection; μ is a hyper-parameter that is set to 2500; subscript $tf_{\#1(q_i,q_i+1),d}$ in Formula 9 represents the number of times phrase "$q_i\ q_{i+1}$" appears in d; $tf_{\#\text{uw8}(q_i,q_i+1),d}$ in Formula 10 represents the number of times two terms q_i and q_{i+1} in a window of 8 terms in d, no matter which term comes first, appear in a

document; similarly, $cf_{\#1(q_i,q_{i+1})}$ and $cf_{\#uw8(q_i,q_{i+1})}$ are the number of times "$q_i\ q_{i+1}$" matches in the entire collection.

In summary, the TDM provides a framework to model the dependencies over terms in proximity, resulting in more precise matches of relevant documents.

2.6 Re-ranking methods

2.6.1 Overview

Traditional retrieval models (e.g. TF-IDF, Okapi BM25) may not be always satisfying because they do not fully discover available information, from the text that they rely on to external knowledge bases.

Re-ranking is a widely-used strategy to solve the problem[55–58]. Re-ranking is the reordering of the results from retrieval systems, based on the initial search results or an external knowledge base[55]. In practice, a re-ranking method evaluates the relevance of each object in the initial search result that is typically achieved by a text-based retrieval system. Compared with term-weight based retrieval methods, re-ranking methods may employ more features, such as patterns identified from the initial list, but they also face the challenges from the low quality of initial retrieval results.

Re-ranking methods can be classified into four categories[55]:

1. Self re-ranking uses the initial results from a retrieval system to refine the ranks of retrieved objects in next search. All of the information for re-ranking is provided by the initial results.
2. Example-based re-ranking uses query examples, rather than keywords.
3. Crowd re-ranking uses results from additional retrieval systems.

4. Interactive re-ranking involves user interaction.

Table 2 provides highly cited examples of the re-ranking algorithms.

Table 2. A summary of re-ranking algorithms by category. This content is mostly reproduced from Mei et al.[55]

Category	Examples	Description
Self re-ranking	Cluster-based methods[59–61]	Group the initially retrieved objects into different clusters and then re-rank objects inside each group.
	Pseudo relevance feedback[62,63]	Label the top ranked objects as “positive” examples and other as "negative". Learn a ranking model using labeled examples.
	Graph-based methods[64–68]	Build a graph from the initial top-ranked documents from the entire document collection.
Example-based re-ranking	Query-by-example (QBE)[25,56,69–71]	Understand the query provided by users using accompany examples.
	Concept-based re-ranking[72–75]	Utilize the results from concept detection to aid search, thereby leveraging human annotation on a finite concept lexicon to help answer infinite search queries.
	Query expansion[58,76]	Generated a new query by using the highly ranked documents.
Crowd re-ranking	Translingual Information Retrieval (TIR)[77]	Provide a query in one language and search objects in one or more different languages.
	Metasearch[78–81]	Find the representative patterns and relations in results of multiple search engines, and then combine the search result lists.
	Query expansion[82]	Mine a knowledge base to select the most relevant and informative keywords for querying a different search engine.
Interactive re-ranking	Rerank-by-example[83]	A user is enabled to edit a part of the search results (i.e. delete and emphasis operations)
	Re-ranking using collaborative filtering[84]	Learn the profiles of the users using machine learning techniques by making use of past browsing histories and then re-rank the results based on collaborative filtering techniques
	Relevance feedback[85,86]	Users are required to annotate whether a subset of initial search results is relevant or not at each iteration

2.6.2 Learning-to-rank (LTR) algorithms

LTR algorithms are a family of ranking algorithms using similar algorithmic frameworks[87]. When compared with the traditional retrieval models (e.g. Okapi BM25, language models) that are created without training, LTR algorithms employ machine learning techniques to automatically learn the ranking models[87]. As a supervised learning task, the LTR algorithms are provided with a set of queries and associated documents in the learning step. For each pair of query and document, ranking features are created as a function of the query-document pair. While the features are designed according to specific task requirements, there exist widely-used features that can perform in multiple tasks, such as BM25 and PageRank[88]. When the ranking models are learned, they are used in the same way as traditional retrieval models.

Let $\mathcal{X}$ be the input feature space, and $\mathcal{Y}$ be the output space of ranking results. Further denote x as an instance from $\mathcal{X}$, and y as an instance from $\mathcal{Y}$. Therefore, x is a list of feature vectors, and every vector represents an instance to be ranked. y is a list of outcomes from the ranking model. The goal of the LTR algorithm is to learn a ranking model F that maps features to correct likelihoods. Given a training data set $X = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\}$, where x_i $(i \in [1, n])$ is a list of feature vectors and y_i is a list of ranks, a ranking function is applied to all x_i, and generates predicted ranking lists $F(x) = \{F(x_1), F(x_2), \dots, F(x_n)\}$. A loss function $L(F(x), y)$ is then defined on $F(x)$ and $y = \{y_1, y_2, \dots, y_n\}$. LTR minimizes the empirical loss

$$\widehat{R(F)} = \frac{1}{m}\sum_{i=1}^{m} L(F(x_i), y_i) \quad [11]$$

where m is the number of training instances in this study.

According to the loss function, LTR algorithms are classified into three categories: point-wise, pair-wise and list-wise. In the point-wise LTR algorithms, the loss function is defined on individual objects to be ranked. For example,

$$L_{pointwise}(F(x), y) = \sum_{i=1}^{m}\sum_{j=1}^{r}\left(F\left(x_{ij}\right) - y_{ij}\right)^2 \quad [12]$$

where r is the number of objects to be ranked in a training instance[89].

The pair-wise LTR algorithms evaluate loss based on the difference between objects[87]. For example,

$$L_{pairwise}(F(x), y) = \sum_{i=1}^{m}\sum_{j=1}^{r-1}\sum_{k=j+1}^{r} \phi\left(sign\left(y_{ij} - y_{ik}\right), F\left(x_{ij}\right) - F(x_{ik})\right) \quad [13]$$

where ϕ may be the hinge loss, exponential loss, or logistic loss functions.

The list-wise LTR algorithms evaluate the loss on lists of objects[87]. For example, ListNet[90] uses the function

$$L_{listwise} = \sum_{i=1}^{m} crossentropy(F(x_i), y_i) \quad [14]$$

Cao et al.[90] showed improved performance of list-wise LTR when compared to three major pair-wise LTR algorithms. Both list-wise LTR and pair-wise LTR algorithms capture more information than point-wise LTR algorithms due to the design of the loss function[91].

2.7 Biomedical dataset retrieval systems

2.7.1 Existing systems

There exists a great number of biomedical dataset resources. Some resources simply use database management systems to retrieve datasets[24], while some employ advanced retrieval systems dedicated to their datasets[92,93]. Such advanced retrieval systems often serve multiple biomedical databases and enable cross-resource and cross-domain search; the well-known systems are Entrez[92] and EMBL-EBI search[93].

Entrez is a search engine for biomedical databases built by the National Center for Biotechnology Information (NCBI, NLM). In 2012, Entrez provided access to 37 databases that contained 690 million records[94]. This system supports text search using simple Boolean queries, and can efficiently retrieve datasets in various formats such as sequences, structures, and references. Similarly, EMBL-EBI search, developed by the European Bioinformatics Institute (EBI) enables users to retrieve data across all disciplines in biology and biomedicine, including sequences, genes, gene products, proteins, protein domains, protein families, enzymes and macromolecular structures, and life science literature.

2.7.2 Data Discovery Index (DDI), bioCADDIE and DataMed

In 2014, the NIH requested to develop a Data Discovery Index (DDI) to promote biomedical research[95]. As part of the NIH Big Data to Knowledge (BD2K) initiative, the Biomedical and Healthcare Data Discovery Index Ecosystem (bioCADDIE) project was launched with the aim of helping users find datasets from data repositories that they would be unlikely to encounter[96]. bioCADDIE has

developed a DDI prototype DataMed to support the findability and accessibility of datasets[96]. As of May 7, 2017, DataMed has covered 1,375,728 datasets from 66 repositories, and has accommodated dozens of features to help users explore datasets. DataMed includes three components[96,97]: (a) a repository ingestion and indexing pipeline, which harmonizes disparate metadata from each repository into the Data Tag Suite (DATS) model[97], a dataset representation model; (b) a terminology server to maintain the semantic consistency of the metadata; and (c) a Lucene-based dataset search engine to find datasets in the DDI.

2.7.3 Data Tag Suite (DATS) model

The Data Tag Suite (DATS)[97] is a metadata schema for characterizing the metadata elements and the structure of biomedical datasets, as well as supporting the indexing and retrieving functionalities of DataMed. In DataMed, the metadata of datasets from various repositories are harmonized into the DATS model.

DATS metadata elements could be divided into two parts: a core set of elements that are “generic and applicable to any type of datasets”, and cover the basic information of datasets[97], and an extended set containing additional elements for specific data types, such as datasets from biomedical science and environmental science domains. The objective of DATS is to find and access datasets via key metadata descriptors. Based on this objective, the DAT model is designed around the Dataset element, and the key information about the Dataset element is its accessibility.

2.8 Evaluation

Precision and recall are still important measures in current information retrieval research, and both require exhaustive judgments. Precision measures the percentage of relevant documents in the all the retrieved documents,

$$Precision = \frac{number\ of\ relevant\ documents}{number\ of\ retrieved\ documents} \quad [15]$$

and recall measures the percentage of relevant documents among all relevant ones,

$$Recall = \frac{number\ of\ retrieved\ relevant\ documents}{number\ of\ all\ relevant\ documents} \quad [16]$$

Derived from precision, $P@k$ measures the precision of the top k retrieved documents,

$$P@k = \frac{number\ of\ relevant\ documents\ in\ top\ k\ results}{k} \quad [17]$$

Another commonly used measure is average precision (AP), which considers both precision and recall,

$$AP = \frac{\sum_{i=1}^{n}(P(i)\times rel(i))}{number\ of\ relevant\ documents} \quad [18]$$

where $P(i)$ is the precision at the i^{th} document, and $rel(i)$ is the relevance of the i^{th} document (0 for irrelevant, 1 for relevant), n is the number of retrieved documents.

When the relevance judgments are incomplete, $P(i)$ is estimated using sampling methods[98]. Therefore, AP becomes inferred AP ($infAP$),

$$infAP = \frac{\sum_{i=1}^{n} \left(\widehat{P(i)} \times rel(i) \right)}{number\ of\ relevant\ documents} \quad [19]$$

where $\widehat{P(i)}$ is the expected precision at the i^{th} document. More information is available in Yilmaz & Aslam's work[98].

Mean average precision (MAP) is the mean of average precision scores for every query,

$$MAP = \frac{\sum_N AP}{N} \quad [20]$$

where N is the number of queries.

Macro-average precision is the average of precisions for every query,

$$Macro - average\ precision = \frac{\sum_N P}{N} \quad [21]$$

where N is the number of queries.

Normalized Discounted Cumulative Gain (NDCG)[99] is a family of measures. NDCG is based on Discounted Cumulative Gain (DCG)[99], a weighted sum of the relevance of the ranked objects. The weight is a decreasing function of the rank of the object, therefore called discount[100]. The definition of DCG is,

$$DCG_n = \sum_{i=1}^{n} \frac{rel_i}{\log_2(i + const)} \quad [22]$$

where n is the number of retrieved objects, rel_{i} is the graded relevance (0 for not relevant, 1 for partially relevant and 2 for relevant) of the i^{th} retrieved object, and $const$ is a smoothing constant. Both the base of the logarithm and $const$ are adjustable according to different needs.

NDCG normalizes DCG by the ideal DCG, which is the DCG of the best ranking result (i.e. retrieved documents ordered by the relevance values)[100],

$$NDCG_n = \frac{DCG_n}{IDCG_n} \quad [23]$$

$$IDCG_n = \sum_{i=1}^{|REL|} \frac{rel_i}{\log_2(i + const)} \quad [24]$$

where REL is the best ranking list, and $|REL|$ is the size of the best ranking list.

When only the top i documents are considered, NDCG is referred to as NDCG@i[100]. Therefore,

$$NDCG@i = \frac{DCG_i}{IDCG_i} \quad [25]$$

When judgments are incomplete, inferred NDCG ($infNDCG$) is an estimation of NDCG, and both DCG_n and $IDCG_n$ are estimated using sampling methods[101],

$$NDCG_n = \frac{\widehat{DCG_n}}{\widehat{IDCG_n}} \quad [26]$$

where $\widehat{DCG_n}$ is the expected DCG_n, and $\widehat{IDCG_n}$ is the expected $IDCG_n$. More information is available from Yilmaz et al.[101]

P@10, infAP, NDCG@10, and infNDCG are used in Chapter 4 to measure the performance of a biomedical dataset retrieval pipeline. Macro-average precision, AP and MAP are used in Chapter 6 to measure the performance of a PubMed similar article retrieval method.

3 The retrieval of biomedical datasets

Biomedical researchers and clinical professionals often need datasets to validate hypotheses. The datasets may be created in their work, obtained from collaborators, or downloaded from public resources. Creating a dataset is usually expensive. Therefore, sharing datasets is highly encouraged for promoting research efficiency, reducing costs and saving time. However, this is not a trivial task. Major technical challenges in a data discovery index include a common representation and indexing of datasets, interpretation of users' requests, retrieval of the most relevant results, and proper display of results to users.

Biomedical datasets are generated from a wide variety of sub-domains in biomedicine. Datasets often have distinct attributes with different formats. Such distinctions make the dataset representation and the following indexing work difficult. Similarly, users of biomedical datasets may come from any sub-domain or even outside biomedicine and may have different background knowledge and search skills. Therefore, to achieve satisfying retrieval performance with such diverse datasets and users, helping users formulate effective queries is crucial. Also, like in any other information retrieval application, optimization of the ranking algorithm is a must for biomedical dataset retrieval. Finally, a good design for result display can help users efficiently identify the desired datasets.

This chapter introduces the background of biomedical dataset representation, the challenges and solutions in the indexing processes, and the users' information-seeking behavior in biomedical information retrieval tasks.

3.1 Biomedical dataset heterogeneity

Biomedical datasets are naturally heterogeneous because they are serving different purposes, composed of diverse data types, and organized in multiple formats. For example, clinical trial records, nuclear magnetic resonance (NMR) images, genomic sequences, and proteomic spectra are all within the scope of biomedical datasets. However, they have little in common regarding the purpose, composition, and format. The diversity results in significant difficulty in extracting a set of universal features to represent biomedical datasets.

Metadata, which typically consists of text describing the datasets, have been widely used in biomedical dataset indexing and retrieval. However, there are two problems to be solved. First, when metadata are manually created, there exists the potential for inconsistent wording. For example, two creators may use different terms to refer to the same phenomenon. Second, there is not much space in metadata for elaboration. To deal with the space problem, controlled vocabularies are adopted to make sure the information is precisely and clearly conveyed. Section 3.1.1 introduces the definition of metadata, compositions and attributes. Section 3.1.2 introduces the motivation of developing controlled vocabularies, the definition, and a few examples.

3.1.1 Metadata

Metadata is “the sum total of what one can say at a given moment about any information object at any level of aggregation”[102], where an information object is “anything that can be addressed and manipulated as a discrete entity by a

human being or an information system". In this chapter, an information object is a biomedical dataset, unless otherwise specified.

According to the function, metadata can be classified into the description, administration, and preservation types[102,103]. The description metadata are the most frequently discussed in this chapter because they are used to identify, authenticate, and describe collections of datasets[102].

Typical metadata consist of three parts[103]: schema, language and instance. Metadata schema is a set of attributes with precise semantic definitions[104], such as title, author, and subject. The meanings of attributes define the semantics of a schema. Metadata languages, such as XML Schema[13], Web Ontology Language (OWL)[105], and Resource Description Framework Schema (RDFS)[106], define metadata schemas. Metadata instance is a set of metadata attributes and associated content values.

Of note, the metadata are not part of the dataset content; they are provided by either the dataset creators or database curators based on their understanding of the dataset. Therefore, the quality of metadata of biomedical datasets depends on the design of the schema, the curators who provide the content values of the schema attributes, the controlled vocabularies selected for the content values, and the quality the dataset content.

[13] https://www.w3.org/TR/xmlschema-1/

3.1.2 Controlled vocabulary

Synonyms, polysemy, and acronyms are frequently observed in the metadata of biomedical datasets and they are caused by multiple factors. For example, a drug may have multiple commercial brand names, e.g. acetaminophen is commercialized under various names including Tylenol, Paracetamol, and Mapap. Diseases and syndromes can also have multiple names. For instance, Addisonian syndrome and primary adrenal deficiency refer to the same problem. Researchers who work on different topics may name independent discoveries using the same name or the same acronym. For example, NF2 is a protein as well as a gene encoding for the protein. Even if a concept has a standardized name, the name may also evolve over time. For example, HIV-1/HIV-2 was originally called Human T-Cell Leukemia Virus/HTLV/LAV in the 1980s.

These linguistic phenomena make the representation of biomedical datasets inconsistent, and thereby hamper efficient retrieval of users' desired datasets. To deal with the phenomena, researchers have developed a variety of terminologies[107,108], each of which is a set of organized terms that represent concepts and their relationships for a specific domain. According to the organization and complexity, terminologies could include subject headings list, thesaurus, ontology, etc[109]. These sub-types share common attributes but also have distinctions. For example, an ontology emphasizes strict semantic relationships among concepts and attributes with the goal of knowledge representation in machine-readable form, while a subject headings list is typically

arranged in alphabetical order and lacks complicated relationships that the ontology has[109]. However, in this chapter, these sub-types are interchangeable.

Widely used biomedical terminologies include Unified Medical Language System (UMLS)[110], Medical Subject Headings (MeSH)[111], International Classification of Diseases (ICD)[112], SNOMED-CT[113], and Gene Ontology[114].

For example, MeSH is a hierarchical thesaurus that was designed by the NLM for indexing MEDLINE[115] records (i.e. the metadata of biomedical articles). MeSH terms specify the topics of biomedical articles to easily find the articles. MeSH has constantly been updated, and has influenced the development of many other terminologies[111]. The number of MeSH terms was gradually increased: while the 1960 edition contains 4,400 terms, the 2016 edition contains 27,883 terms are hierarchically arranged in thirteen levels.

3.2 Indexing process

Once the metadata of biomedical datasets are prepared, the datasets are ready for indexing. The indexing process of biomedical datasets resemble the process for textual documents, which has been studied for decades.

Like textual documents, a large portion of biomedical datasets are retrieved using text-based information retrieval systems because they are also represented with text. A major difference between the textual documents and the text-based biomedical datasets is the source of text: the textual documents primarily rely on the content, while the datasets typically depend on textual metadata for indexing and retrieval. From the creator's perspective, the textual content must be created by the authors of textual documents, while text in metadata may be created by

others, such as database curators. Even if created by the authors of datasets, metadata do not necessarily comprehensively characterize the datasets from all perspectives because metadata are merely the creators' perception of the dataset. From the data perspective, metadata are typically more brief than textual documents, but more complicated in the structure that may include more attributes and attribute types.

Even though there are many differences, the textual document is a good starting point for the indexing of biomedical datasets. The experience and lessons from the domain are likely to be useful for biomedical dataset. In this section, we will compare the indexing processes for textual documents and biomedical datasets.

3.2.1 Textual documents

In textual document retrieval, indexing means finding good representations for the document contents[116]. The indexing methods are mostly based on a term-document matrix or an approximation of the matrix. Figure 2 provides an example to illustrate a basic indexing method, in which the elements of the term-document matrix represent the occurrence number of terms in documents. Other advanced methods have been developed to estimate the relationships between each pair of term and document, i.e. term weight, such as TF-IDF[43,44] and Okapi BM25[47]. The term weight estimation may be empirical (e.g. TF-IDF[43,44]) or theory-based (e.g. Okapi BM25[47], continuous language model[117]).

When the vocabulary of a document collection contains millions of terms, the size of the term-document matrix may be too large to be implemented

efficiently. Therefore, various methods were developed to reduce the dimension of vocabulary, thereby approximating the term-document matrix. A successful example is Latent Semantic Indexing (LSI)[118] that uses Singular Vector Decomposition (SVD) to reduce the dimension of vocabulary before indexing documents. According to Manning et al.[31], Dumais[119,120] conducted experiments with LSI on TREC tasks and achieved precision at or above that of the median TREC participant. Later, Latent Dirichlet Allocation (LDA)[121], which reduces the dimension by representing documents with topics, was incorporated into a language modeling framework for textual document indexing and retrieval[122].

3.2.2 Biomedical datasets

The metadata of biomedical datasets consist of various attributes, such as title and release date. Within each attribute, the indexing process includes text transformation, term-document matrix construction, and term weight computation following the selected weighting scheme. For details of text transformation, please refer to section 2.2. For an example of term-document matrix, please see section 2.4. For weighting schemes, please see section 2.5.

Compared with metadata, linked information is an underused information source in biomedical dataset indexing. Linked publications, especially the primarily associated articles (i.e. articles that announce the existence of the datasets), contain abundant information of datasets. Another advantage is that they are often well-formatted and ready for analysis. MacMullen et al.[123] reviewed early studies on the integration of linked publications and biomedical datasets.

3.3 User behaviors

A good retrieval result is the joint work of the retrieval system and the user. The work includes representation, indexing, formulating effective queries, retrieving satisfactory results, and properly displaying results, etc.

This section introduces the information needs of different user groups, users' information-seeking behaviors, and the interface design for retrieval systems. We discuss users' behaviors for both dataset and general biomedical information retrieval because these behaviors are similar.

3.3.1 User groups and information needs

The users of biomedical information and datasets have a wide range of information needs. While molecular biology researchers may seek in-depth information[14,123], clinicians may require quick, concise information, frequently related to diagnosis and treatment[14,15]. Thus, understanding user needs would benefit the design of information retrieval systems and users' retrieval experience. However, information need is case-dependent, and thus mapping a spectrum of all information needs in the biomedical research domain is non-trivial. Geer et al.[124] provided a new approach to study users' information needs: by categorizing users according to their requested aids from libraries and bioinformatics centers based on 11 years of data collected from the NCBI User Service, and then inferring their information needs [see Table 3 and 4].

Table 3. User groups in biomedical researchers and their needs related to bioinformatics, based on Geer[124]. The table format and contents are adapted. bioCADDIE/DataMed users may have similar needs. Trainees of all levels and faculty share many common needs, e.g. identifying best-fit resources, efficiently retrieving the data needed, and identifying programming techniques for large-scale data retrieval and analysis. Genome labs and computational biologists have highly technical needs that may involve molecular biology resource developers, bioinformatics centers, and libraries.

		User Groups					
		Undergrad	Grad	Postdoc	Faculty PI	Genome Labs	Computational Biologists
Needs for assistance with bioinformatics resources	Resource Awareness	Understand the range of resources available for a given information need or research problem beyond those commonly used. Select the resource(s) that best meet the need.				Large-scale genome sequencing and computational biology labs generally include staff who are expert users of bioinformatics resources and are unlikely to need assistance in these areas. Their questions and needs are often highly technical and often require the involvement of molecular biology resource developers.	
	Data Structure & Organization	Understand (a) organization of various data types; (b) scope and nature of primary research databases and derivative, value-added databases, and the relationships between them; (c) database record content and format for each data type.					
	Text Search Systems	Apply knowledge of data organization and advanced features of search systems such as Entrez to efficiently retrieve the data needed. Use advanced tools such as Entrez Utilities for automated search updates and batch searching.					
	Sequence Similarity Search Systems	Apply knowledge of various analysis programs to select appropriate program(s), adjust search parameters, interpret results, etc.					
	Other Sequence Analysis Programs	Select and use programs for sequence analysis such as primer design, restriction mapping, multiple sequence alignment, phylogenetic analysis, identification of potential regulatory regions, analysis of mass spectra, structure prediction, and more.					

Table 3. Continued. User groups in biomedical researchers and their needs related to bioinformatics, based on Geer[124]. The table format and contents are adapted. bioCADDIE/DataMed users may have similar needs. Trainees of all levels and faculty share many common needs, e.g. identifying best-fit resources, efficiently retrieving the data needed, and identifying programming techniques for large-scale data retrieval and analysis. Genome labs and computational biologists have highly technical needs that may involve molecular biology resource developers, bioinformatics centers, and libraries.

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="6">User Groups</td></tr>
<tr><td>Undergrad</td><td colspan="2">Grad</td><td>Postdoc</td><td>Faculty PI</td><td>Genome Labs</td><td>Computational Biologists</td></tr>
<tr><td rowspan="4">Needs for assistance with bioinformatics resources</td><td>Commercial Databases & Software</td><td colspan="4">Evaluate, select, and purchase licenses for the use of commercial products. Provide educational and end-user support, as needed.</td><td colspan="2" rowspan="4">Depending on the resources available with the genome and computational biology labs, their institution's organizational structure, and benefits of research collaborations, some might work together with a bioinformatics center on data mining, lab data management, and programming, and with a library or bioinformatics center on leasing commercial products.</td></tr>
<tr><td>Data Mining</td><td rowspan="3">Many undergrads, except those in majors such as bioinformatics or computer science, are unlikely to be involved in these areas</td><td colspan="3">Large scale data retrieval and analysis. Requires a thorough understanding of data organization, scope and nature of user's input, and desired output in order to identify appropriate data sources, extract desired data elements, and identify relationships among them.</td></tr>
<tr><td>Lab Data Management</td><td colspan="3">Develop, purchase, or customize hardware and software for management of data generated by specific laboratories or by a university's overall research program.</td></tr>
<tr><td>Programming</td><td colspan="3">Write scripts for customized bioinformatics needs, such as data mining and lab data management. Development of new bioinformatics tools.</td></tr>
</table>

Table 4. User groups from the clinical community and their needs for genetic information, based on the work by Geer[124]. The table format is modified. biocaddie/DataMed users may have similar needs. Users from different groups have multiple common needs for different purposes. For example, these users may attempt to obtain concise information about genetic conditions to diagnose and manage those conditions or to identify relevant resources for clinical genetics research.

<table>
<tr><td></td><td></td><td colspan="4">User Groups</td></tr>
<tr><td></td><td></td><td>General Public</td><td>Genetic Counselors</td><td>Physicians</td><td>Medical Faculty and Students</td></tr>
<tr><td rowspan="7">Needs for assistance with genetics information resources</td><td>Resource Awareness</td><td colspan="4">Understand the range of resources available for a given information need and audience. Select the resource(s) that best meet the need.</td></tr>
<tr><td>Patient Information</td><td colspan="4">Obtain clear and understandable background information on the molecular basis of genetic conditions, what to expect, how to cope, treatment options, advocacy groups, etc. Identify and use the appropriate resource(s) for a given information need, based on an understanding of the scope and nature of consumer health resources such as Genes and Diseases, Genetics Home Reference, Genetic Alliance.</td></tr>
<tr><td>Diagnosis and Management</td><td colspan="4">Identify resources such as GeneTests and Gene Reviews to obtain salient clinical information about genetic conditions, including symptoms, differential diagnosis, genetic testing, management, and more. Use additional resources such as the clinical synopsis and allelic variants portions of Online Mendelian Inheritance in Man (OMIM) and drug response information in PharmGKB.</td></tr>
<tr><td>Clinical Trials</td><td colspan="4">Identify ongoing clinical trials through the use of resources such as ClinicalTrials.gov to investigate the possibility of novel therapies.</td></tr>
<tr><td>Basic Research Information</td><td colspan="3">Search for up-to-date information about basic research on genetic conditions, including journal literature in systems such as PubMed, literature summaries on human genes and genetic conditions in resources such as OMIM.</td><td>Also retrieve, if appropriate, associated primary research data in molecular biology databases.</td></tr>
<tr><td>Clinical Research Data Management</td><td>Not Applicable</td><td colspan="3">Develop, purchase, or customized hardware and software for management and analysis of data from clinical research on genetic conditions. Integrate with data from basic research resources, if appropriate. For example, integrate data from clinical research on allelic variants and drug response with data from genetic variation and population study resources such as dbSNP and HapMap project, respectively.</td></tr>
</table>

3.3.2 User information-seeking behaviors

Users' behaviors are complex. This section covers the awareness of resources and the search skills.

3.3.2.1 Awareness of resources

Until the early 2000s, many researchers were not yet familiar with online biomedical information resources such as NCBI databases. By 2010, online information resources had been widely accepted and strongly preferred in the biomedical community[14,15].Today, researchers have a wide array of options, including periodical websites, databases (e.g. NCBI GEO), literature databases (e.g. MEDLINE/PubMed), search engines (e.g. Google Scholar), publisher websites (e.g. the Highwire Press archive), table-of-contents alerts, and other miscellaneous resources[14,15,125].

3.3.2.2 User search skills

Modern biomedical information retrieval systems are complicated and require users to have some knowledge of the systems to maximize the potential of the retrieval system. Since users are not equally knowledgeable of these systems, we expect to observe great disparities in the ability to discover, understand, and proper use of information resources[124].

The disparity starts with query formulation. Typical text-based biomedical information retrieval systems accept three types of queries[126]: keyword query, Boolean query, and long-text query. Keyword queries can be named entities such as protein name, disease name, and protein database identifier. Boolean queries use Boolean operators (e.g. AND, OR, NOT) to combine keywords. Long-text

queries may be composed of any text, such as an article abstract or even a set of documents. Based on the NCBI user log data, Grefsheim et al.[15] reported that 70% of the searches are keyword queries, 21% use Boolean operators, 13% use attribute specifiers, and 1% use the wildcard, range searching, and the History function. As for the dataset retrieval, keyword queries are even more frequent, such as 75% for the nucleotide and 89% for the protein[15].

Clearly, users prefer keyword queries that do not require much knowledge about the information retrieval systems. However, keyword queries are likely to result in massive, irrelevant and redundant results. For example, searching Entrez with the simple query “p53” will bring datasets including expressed sequence tags, genes, proteins, SNP dataset, clinical phenotypes, etc[15]. According to the NIH survey[15], although more advanced queries can often improve retrieval performance, users either do not like building complicated queries, or are not aware of how to build such queries. Fortunately, users often reformulate simple queries once they have gained a better understanding of the requests after reviewing the initial retrieval results. In fact, Jay et al.[127] found that browsing is the key step in query reformulation and biomedical dataset discovery.

3.3.3 Interface design of retrieval systems

A user-friendly design of the interface may improve user satisfaction. Yet there is no standard template for the design. For instance, the designer of a literature database dedicated to Alzheimer’s disease may tune the system toward users with prior knowledge by providing a complex query interface with optional fields to support the expert users. However, the cross-disciplinary nature of

biomedical research does not guarantee a user's expertise on the subject domain[127], as users outside of the subject domain may also be interested.

The search interfaces of existing biomedical dataset retrieval systems include the traditional design and single-field design. The traditional design (e.g. PubMed advanced search interface) provides multiple options, thus expects accurate queries[127,128]. In particular, it requires a precise conceptualization of the information need, and a good understanding of the information retrieval system to capture more relevant results[127]. In contrast, the single-field design (e.g. Google) is suitable for vague queries and users working under time constraints[127,129]. The single-field design does not require user expertise on the subject of query or on the information retrieval system, and may be advantageous in general data discovery: it is reported that biomedical researchers and clinical professionals favor the single-field interface for such purposes in two recent studies[14,127].

The output interfaces of existing biomedical dataset retrieval systems are similar to those of document retrieval systems: a set of relevant datasets is retrieved because their metadata match the queries, and they are displayed according to the relevance. In a recent study, Jay et al.[127] found that summarization, analytics, and visual presentation can help users better digest the results, and that descriptions of each result are the primary focus of users. The same study encouraged the use of faceted search and navigation to allow users to organize retrieved datasets hierarchically to help users reformulate queries.

3.4 Summary

The heterogeneity of biomedical datasets is a barrier for the extraction of representative features from the dataset contents. Therefore, researchers primarily use metadata to represent, index and retrieve the data. To convey precise information in the limited space of metadata, metadata creators intensively use controlled vocabularies to standardize the wording. The indexing process of biomedical datasets largely relies on the metadata, and indexing techniques for textual documents may be helpful. Lastly, the users of biomedical dataset retrieval systems have a wide variety of information needs. Therefore, it is essential to categorize users, understand their needs, and study their search strategies to build a more efficient dataset retrieval system.

4 Retrieval and re-ranking for biomedical datasets

The rapidly growing DataMed system faces important challenges, including dataset representation, query formulation, and ranking algorithm. First, the metadata do not always provide sufficient descriptions of corresponding datasets, such as the organism. Since the metadata have been harmonized into the DATS[97] model, detailed information that is specific to a particular data type may not be easily be transformed into DATS (e.g., free-text annotations and related biomedical articles). Second, DataMed is expected to take users' free-text requests as inputs and reformulate them to comply with retrieval system syntax. The free-text search requires a two-step process of first capturing keywords from requests and then building queries from the keywords. Finally, identifying and appropriately ranking relevant datasets depend on the specific questions a researcher is trying to answer. Many optimization approaches remain to be explored.

We developed a pipeline and explored different approaches to overcome the abovementioned obstacles. Our pipeline consists of five main modules:

1. Automatic collection of additional information beyond metadata for existing datasets,
2. Dataset indexing using metadata and the additional information,
3. Query formulation by analyzing users' free-text requests,
4. Dataset retrieval using Elasticsearch[14],
5. Elasticsearch result re-ranking using multiple re-ranking algorithms.

[14] https://www.elastic.co/

The 2016 bioCADDIE Dataset Retrieval Challenge[130] provided us with an opportunity to evaluate our pipeline, and we achieved the highest infNDCG in the Challenge.

This chapter introduces the design of pipeline, the mechanism of each module, and its performance in the 2016 bioCADDIE Dataset Retrieval Challenge.

4.1 Pipeline

To achieve real-time retrieval on the extensive collection of datasets, we employed a "retrieval plus re-ranking" strategy to improve the retrieval performance while maintaining efficiency. Our pipeline collects additional information for datasets to supplement the metadata, builds indices, automatically interprets free-text requests and generates Boolean queries, retrieves datasets using Elasticsearch, re-ranks top datasets from Elasticsearch, and evaluates the performance of datasets (Figure 3).

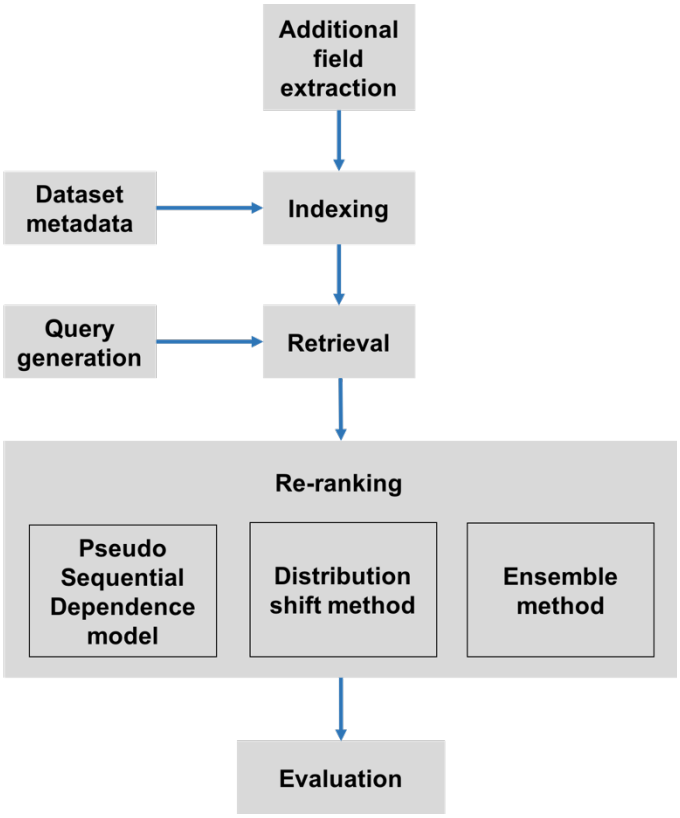


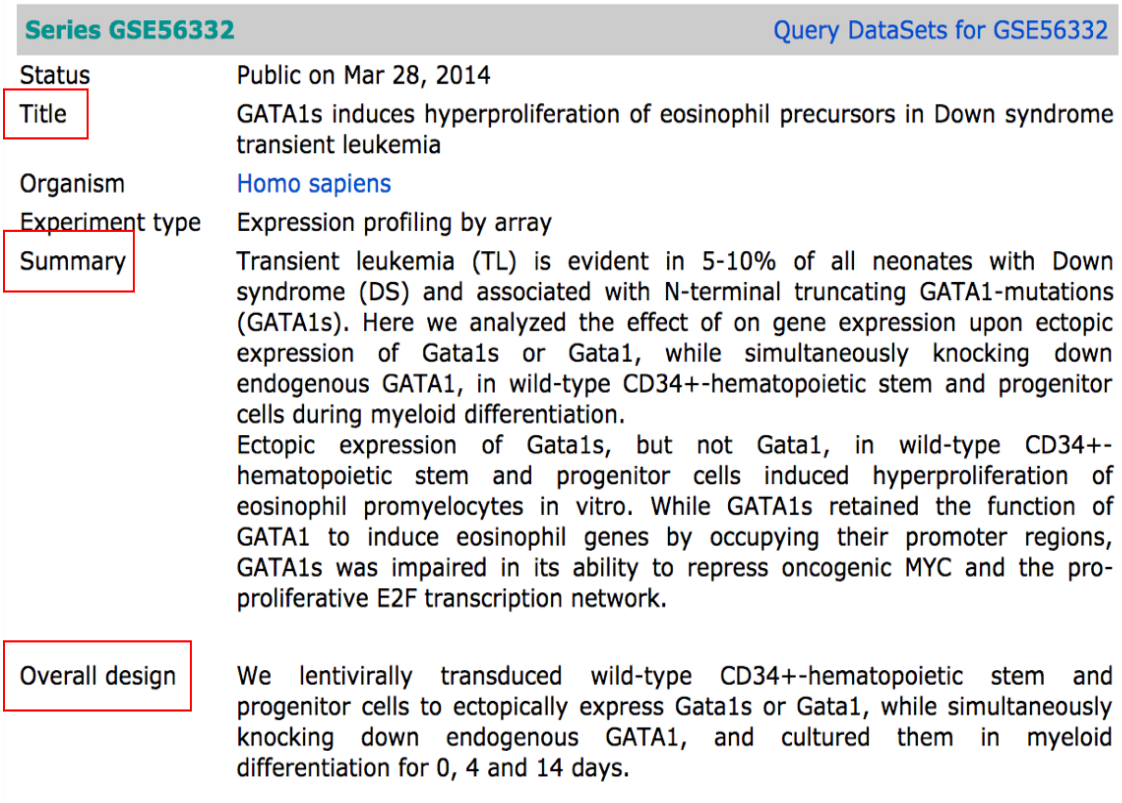
Figure 3. The pipeline for biomedical dataset retrieval. The pipeline consisted of five modules: additional field extraction, indexing, query generation, retrieval, and re-ranking. Additional information was collected as a supplement to the dataset metadata. Indices were built on the combination of metadata and additional information. Once a query was automatically generated from a user's free-text request, the Elasticsearch system retrieved relevant datasets using the query. Next, these datasets were re-ranked using two different algorithms, the pseudo sequential dependence model and the distribution shift method. The re-ranked results could be further merged using the Ensemble method. Finally, re-ranked datasets were evaluated on the test set provided by the Challenge.

4.1.1 Additional data collection

Retrieval systems depend on comprehensive metadata to obtain user-desired datasets. However, metadata often contain limited information. For example, the metadata of a typical ArrayExpress[15] dataset use a "description" field to summarize the study in a few sentences.

[15] https://www.ebi.ac.uk/arrayexpress/

However, at the same time, rich information embedded in related sources, such as related publications, is not fully explored. Therefore, we extended the original metadata of the datasets. We identified 158,963 datasets that have connections with GEO Series records, and collected the fields “Summary”, “Title”, and “Overall design” for these datasets from GEO. An example of these fields is provided in Figure 4. We named this collection of new fields and values “additional information”.



Series GSE56332	Query DataSets for GSE56332
Status	Public on Mar 28, 2014
Title	GATA1s induces hyperproliferation of eosinophil precursors in Down syndrome transient leukemia
Organism	Homo sapiens
Experiment type	Expression profiling by array
Summary	Transient leukemia (TL) is evident in 5-10% of all neonates with Down syndrome (DS) and associated with N-terminal truncating GATA1-mutations (GATA1s). Here we analyzed the effect of on gene expression upon ectopic expression of Gata1s or Gata1, while simultaneously knocking down endogenous GATA1, in wild-type CD34+-hematopoietic stem and progenitor cells during myeloid differentiation. Ectopic expression of Gata1s, but not Gata1, in wild-type CD34+-hematopoietic stem and progenitor cells induced hyperproliferation of eosinophil promyelocytes in vitro. While GATA1s retained the function of GATA1 to induce eosinophil genes by occupying their promoter regions, GATA1s was impaired in its ability to repress oncogenic MYC and the pro-proliferative E2F transcription network.
Overall design	We lentivirally transduced wild-type CD34+-hematopoietic stem and progenitor cells to ectopically express Gata1s or Gata1, while simultaneously knocking down endogenous GATA1, and cultured them in myeloid differentiation for 0, 4 and 14 days.

Figure 4. An example of additional fields, including study title, study summary, and overall design.

4.1.2 Indexing

The metadata and the additional information were indexed using Elasticsearch. We developed customized mapping schemas for Elasticsearch

based on the DATS model. In particular, we selected fields in the metadata as "standard fields", which contains the most valuable information about the datasets from each database. The standard fields for each data repository are provided in the Appendix A. During the construction of indices, fields in the DATS model were classified into three groups: exact matching (e.g., MeSH terms), regular string matching (e.g., description) and others (e.g., release date). The text contents of metadata were analyzed using the standard tokenizer, English possessive stemmer, lower case filter, non-ASCII character filter, stopword[16] filter, and the Elasticsearch light English stemmer. However, all MeSH terms and their associated entry terms (i.e. synonyms) were protected against the stemmer.

4.1.3 Query generation

To enable automatic query generation, we built a module to analyze users' free-text requests, extract keywords, and generate Boolean queries. One example of the free-text request is "find data of all types related to TGF-beta signaling pathway across all databases". In the module, a rule-based filter removed less informative words from questions and kept key concepts. The less informative words include the English stopwords from Natural Language Toolkit (NLTK[17] (module detail: nltk.corpus.stopwords.words("english")) and self-defined stopwords: "database", "databases", "datasets", "dataset", "data", "related", "relate", "relation", "type", "types", "studies", "study", "search", "find", "across", "mention", "mentions", "mentioning", "i", and "a".

[16] https://www.ncbi.nlm.nih.gov/books/NBK3827/table/pubmedhelp.T.stopwords/
[17] http://www.nltk.org/

Next, the remaining words (e.g. TGF-beta signaling pathway) were passed to PubMed for concept expansion using NCBI E-utilities[18]. In this step, key concepts were identified and then expanded. In the example, two concepts, “TGF-beta” and “signaling pathway”, were identified in the above request. Then, the “TGF beta” was expanded as two representations, “TGF beta” and “transforming growth factor beta”, while the concept “signaling pathway” was expanded to “signal transduction” and “signaling pathway”. Queries generated based on expanded concepts enabled ElasticSearch to search all fields and retrieve relevant datasets that would be likely missed by the search based on queries without expansion. See Figure 5 for an illustrative example.

Finally, the key concepts and the expanded associations were formulated into nested Boolean queries based on their relationships. Specifically, the representations of the same concept were connected by “OR” operator and the different concepts were also linked by “OR” operator. A concept is recognized as present if the original concept or the expanded associations are observed in the metadata of a dataset, and a dataset is retrieved if at least one concept is present. We performed the search by first retrieving datasets with all concepts present, then obtaining datasets with one less concepts matching, etc. Datasets with more concepts matched were ranked higher. Lastly, we only kept (at most) the top 5,000 documents for each query.

[18] https://www.ncbi.nlm.nih.gov/books/NBK25497/

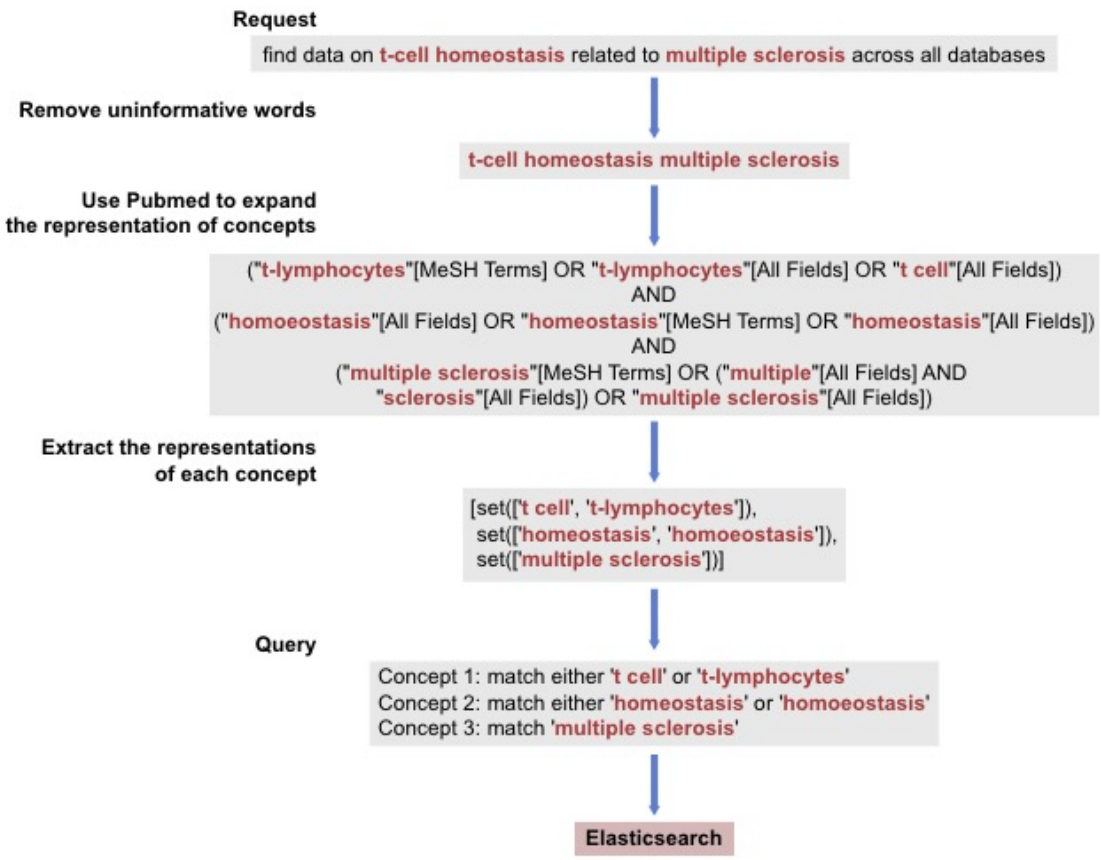


Figure 5. Query interpretation: from a free-text request to a query. The query expansion used the same method as PubMed does, relying on NCBI E-utilities.

4.1.4 Retrieval and re-ranking

We implemented a two-step "retrieval plus re-ranking" strategy. In step 1, Elasticsearch retrieved datasets from the entire collection. In this step, we attempted to capture as many relevant or partially relevant datasets as possible in the top 5,000 retrieval results, with less focus on the ranking performance within the top 5,000. In step 2, we applied re-ranking algorithms to the top 5,000 results and aimed at higher infNDCG. We explored multiple re-ranking algorithms, and finally adopted a pseudo sequential dependence (PSD) model, a distribution shift method, and an ensemble method.

4.1.4.1 Pseudo sequential dependence (PSD) model

The PSD model was derived from the sequential dependence (SD) model developed by Metzler et al.[53] and Bendersky et al.[54] The original SD models rank documents in consideration of the unigrams (i.e. single words), ordered bigrams (i.e. two consecutive words), and unordered bigrams (i.e. two words not necessarily consecutive) in the documents. In our scenario, the documents refer to the metadata of datasets to be re-ranked. In the experiments, we found that neither ordered bigrams or unordered bigrams provided contributions to the performance improvement. One possible explanation is that most keywords were independent of each other. For example, “chromatin modification” contains more specific information than “chromatin” and “modification”, while “flu car” is as informative as “flu” and “car”. Bigrams may benefit the former example, but not the latter case. Therefore, we removed the bigram components from the original formula, and modified the unigram component to make it compatible for dataset retrieval tasks, i.e., making ‘whether a word appears in the metadata’ more important than ‘how many times a word appears’.

Provided with a query and a list of candidate datasets from Elasticsearch, PSD scores every candidate dataset and re-ranks them accordingly. The PSD score is defined in Formulas 26 and 27 based on Metzler and Croft’s work[53,131].

$$P = \sum_{q_i \in Q} f(q_i, D) \qquad [27]$$

$$f(q_i, D) = log\left(\frac{I(tf_{q_i,D} > 0)(tf_{q_i,D} + \delta) + \mu\frac{cf_{q_i}}{|C|}}{|D| + \mu}\right) \quad [28]$$

In Formula 26, P is a sortable quantifier of relevance. D is a dataset, Q is an input (e.g. a free-text request), q_i are words in the input, $f(q_i, D)$ is the weight of word q_i in the metadata of dataset D.

In Formula 27, $tf_{q_i,D}$ is the number of times word q_i matches in the metadata of dataset D, cf_{q_i} is the number of times word q_i has matches in the metadata of the entire collection of datasets, $|D|$ is the word number of the metadata of dataset D, $|C|$ is the total word number of the collection, and μ is a hyper-parameter that is set to 2500, following Bendersky et al.[54] Different from the original algorithm, we added a constant $\delta = 5$, an empirical parameter to $tf_{q_i,D}$ if it was greater than 0, where $I(tf_{q_i,D} > 0)$ is an indicator function. This modification puts more weight on the existence of a word in the metadata than on the times the word appears.

The default version of the PSD model took as input the original Q, i.e., the free-text request. Therefore, we named this version "PSD-allwords". We further developed a "PSD-keywords" version that analyzed only keywords extracted from Q. To identify valuable keywords from free-text requests, PSD-keywords firstly calls MetaMap[132], a biomedical named entity recognizer, to identify the UMLS concepts from Q, and then uses the UMLS concept set Q' as input to PSD, with the aim to exclude the impact of less informative words in requests. In the

experiments, we used the default setting of MetaMap, collected all recognized UMLS concepts, and removed the duplicated concepts.

4.1.4.2 Distribution shift method

Hiemstra stated that in order to search a document collection, the user should first prepare a document that is similar to the needed documents[42]. The idea has been widely accepted and implemented, such as in relevance feedback methods[1].

If documents are represented by words, the closer the word distribution of a document is to that of the user's document, the more likely the document will be relevant to the user's query. However, neither Elasticsearch or PSD consider the difference of word distributions in users' requests and the dataset metadata.

Based on this idea, we developed a method to find Google returned documents according to users' requests, and then transformed these documents into queries for relevant datasets. This way, we shifted the words distribution of requests with the aim of approaching the word distribution of the metadata of relevant datasets. Requests were sent to Google using an in-house script, and then the top 10 retrieved text documents (not limited to datasets) were concatenated into a document as a query for the re-ranking algorithm. Next, the Elasticsearch retrieved datasets were re-ranked based on the concatenated documents using the PSD-allwords model.

4.1.4.3 Ensemble method

This method was developed on an assumption that no single method works for all tasks. Our ensemble method averaged the reciprocal of ranks from different

methods, and re-ranked datasets according to the mean of rank reciprocals. We experimented with combinations of different re-ranking algorithms, and finally chose the combination of PSD-allwords and PSD-keywords. The performance of different combinations is provided in Section 4.3.5, Table 7.

4.1.5 Evaluation

The only required metrics in the Challenge announcement was infNDCG, but infAP[98], NDCG@10[101], and two different precisions (one considered partially relevant as relevant, while the other considered partially relevant as irrelevant) were also evaluated. The definitions of these metrics are available in section 2.8. The infNDCG was computed using a tool[19] from the National Institute of Standards and Technology (NIST), NDCG@10 was evaluated using TREC_EVAL[20] from NIST, and the two types of precision were evaluated using a tool provided by the Challenge.

4.2 Data and information from the Challenge

The Challenge provided a collection of metadata[21] from 794,992 biomedical datasets collected from 20 repositories. The metadata followed the DATS[97] model.

The Challenge also provided relevance criteria, six sample requests with annotated judgments, 30 sample requests without judgments, and 15 requests for evaluation[133]. The requests were artifacts fashioned after TREC topics[22] that

[19] http://www-nlpir.nist.gov/projects/t01v/trecvid.tools/sample_eval/sample_eval.pl
[20] http://trec.nist.gov/trec_eval/
[21] The Challenge data: https://biocaddie.org/benchmark-data
[22] TREC topics: http://trec.nist.gov/data/topics_eng

emulated the tasks to professional librarians, such as “search for gene expression data that mention E2F cell line in cellular differentiation across all databases”.

The judgments followed TREC evaluation procedures for ad hoc retrieval tasks post-hoc assessment, but without pooling[134]. A dataset was judged “relevant” if it met all the constraints in the request, or “partially relevant” if it met part of the constraints. Judgements were pre-determined but released after the submission deadline.

Participants could submit up to five automatic or manual runs, although automatic runs were preferred.

4.3 Results

4.3.1 Implementation

The pipeline was coded in Python, Java, and Perl[23]. The metadata of datasets were indexed using Elasticsearch. Third-party libraries were also used in the implementation, including MetaMap for biomedical concept recognition.

4.3.2 Computation performance

The experiments were completed on an iDASH[135] cloud virtual machine with 16 processors (Intel(R) Xeon(R) CPU E7-4870 v2) and 32 GB RAM. Indexing all datasets took up to 3 hours. PSD-allwords and PSD-keywords each required 4 minutes to re-rank 5,000 dataset candidates on 45 requests.

[23] All scripts are available from https://github.com/w2wei/dataset_retrieval_pipeline

4.3.3 Annotated requests

To facilitate the pipeline development, we annotated 943 datasets for the provided 30 unannotated requests described below. Also, the annotation is available from our github repository[24]. The self-made gold standard was used for optimizing configurations, tuning parameters, and selecting models before we submitted our results.

We created rules to implement the provided relevance criteria. First, we defined key concepts as biomedical concepts that are included in UMLS. Given a request, if fewer than 50% key concepts appear in the metadata of a dataset, the dataset is labeled irrelevant. Among the remaining datasets, if the components of key concepts scatter in one or more sentences in the metadata, and individual components have lost the meaning of the original key concepts, the dataset is labeled partially relevant. If exact key concepts are found in the metadata, but have different meanings (i.e., polysemy), the dataset is labeled partially relevant. If all the exact key concepts appear in the metadata, and the metadata answer the request, this dataset is labeled relevant. If all the key concepts appear in the metadata, but some concepts are partially matched, this dataset is labeled relevant as long as the metadata answer the request.

4.3.4 Performance in the Challenge

We submitted results from five methods (see Table 5), Elasticsearch, PSD-allwords, PSD-keywords, the distribution shift method, and the ensemble method.

[24] Annotations are available from https://github.com/w2wei/dataset_retrieval_pipeline

These methods were evaluated on the test set of 15 requests and all 794,992 datasets.

Table 5. Results of five methods. The indices were built on the provided metadata and the additional information. All methods used automatically generated queries. Method Elasticsearch did not use any re-ranking methods. The other four methods used re-ranking algorithms accordingly. infAP is inferred average precision, infNDCG is inferred NDCG, NDCG@10 is the NDCG score on top 10 results, P@10(+partial) is the precision on top 10 results and counts partially relevant as relevant, P@10(-partial) is the precision on top 10 results and counts partially relevant as irrelevant.

Category	Method	infAP	infNDCG	NDCG@10	P@10 (+partial)	P@10 (-partial)
No re-ranking	Elasticsearch	0.2446	0.4333	0.4228	0.5200	0.2733
Re-ranking	PSD-allwords	0.2792	0.4980	0.6152	0.7600	0.3267
	PSD-keywords	0.2391	0.4490	0.4088	0.5200	0.1667
	Distribution Shift	0.3309	0.4783	0.6504	0.7467	0.3600
	Ensemble	0.2801	0.4847	0.5398	0.6800	0.2400

Among the five methods, PSD-allwords achieved the highest infNDCG and the highest P@10 (+partial) (precision on top 10 results and counting partially relevant as relevant), and Distribution Shift was the best method for infAP, NDCG@10, and P@10 (-partial) (precision on top 10 results and counting partially relevant as irrelevant).

When compared with methods from other teams in the Challenge, PSD-allwords achieved the top infNDCG score among 45 submissions from 10 teams. The Ensemble method and Distribution Shift placed second and third infNDCG in the Challenge. PSD-allwords also tied for third place in P@10 (+partial). The full results of the Challenge are available from Appendix B.

4.3.5 Breakdown analysis

The retrieval step of the pipeline includes three key features: additional information collected from external resources, standard fields in the mapping schema, and query expansion using NCBI E-utilities. To understand the contribution of each feature, we evaluated the infNDCG scores of the pipeline with settings of different combinations of three features (Table 6) on the 15 test request and the associated judgements. When all of the three features were included, the retrieval step achieved the highest infNDCG score. Removing query expansion (row 4) resulted in a larger decrease when compared with either additional fields (row 2) or standard fields (row 3). This observation indicates that the contribution from query expansion is more significant than the other two features. When looked into individual features, we noticed that additional fields alone (row 5) or standard fields (row 6) alone did not improve infNDCG when compared with using no feature (row 8). Combined this observation with row 1, row 2 and row 3, we inferred that there exist interactions between the features and the interactions also help improve the infNDCG performance.

Table 6. Comparison of the pipeline with settings of combinations of four different features (Additional Fields, Standard Fields, and Query Expansion). Y indicates the feature is included, and N indicates the feature is not included. infNDCG measurements are scored in the rightmost column.

	Additional Fields	Standard Fields	Query Expansion	infNDCG
1	Y	Y	Y	0.4333
2	N	Y	Y	0.4164
3	Y	N	Y	0.4159
4	Y	Y	N	0.3961
5	Y	N	N	0.3868
6	N	Y	N	0.4015
7	N	N	Y	0.4084
8	N	N	N	0.4019

For the ensemble method, we explored all combinations of PSD methods and the Distribution Shift method (Table 7) using reciprocal vote (see Section 4.1.4.3 for the voting method details), and evaluated their performance on the 15 test requests. The combination of PSD-allwords and PSD-keywords methods achieved the highest infNDCG.

Table 7. The performance of the Ensemble methods. Y indicates the feature is included, and N indicates the feature is not included.

PSD-allwords	PSD-keywords	Distribution Shift	infAP	infNDCG	NDCG@10	P@10 (+partial)	P@10 (-partial)
Y	Y	Y	0.3120	0.4560	0.6089	0.7267	0.3067
N	Y	Y	0.3120	0.4442	0.5649	0.6800	0.2800
Y	N	Y	0.3216	0.4735	0.6439	0.7733	0.3333
Y	Y	N	0.2801	0.4847	0.5398	0.6800	0.2400

4.4 Discussion

In the study, we collected additional fields “Summary”, “Title”, and “Overall design” for 158,963 datasets from Arrayexpress, Gemma, and GEO databases to

enrich the metadata of the datasets. In a breakdown analysis, we found that contribution from the additional information was positive, but not significant. There were at least two underlying causes: (a) only approximately 20% of all datasets had additional information fields, and (b) the additional information was too noisy, containing information that was not immediately related to the dataset, such as the study background. Identifying additional information for more datasets and filtering out irrelevant information may make linked evidence more valuable for dataset retrieval.

Another aim of the Challenge was to automatically generate queries from users' requests to narrow the gap between users' requests and queries. In our pipeline, we defined rules to extract keywords from requests, selected keywords using MetaMap, and expanded keywords using the NCBI E-utilities. Finally, we built Boolean queries on the expanded keywords. Since the rules are pre-defined, it is inevitable that information is lost when requests are converted into queries. Machine learning methods will provide new solutions for this problem. For example, using deep learning methods, requests may be translated into sentence embeddings to preserve all key information, and the sentence embeddings could act as queries for more effective dataset retrieval.

The distribution shift method used the commercial search engine Google to collect relevant documents, and then identified relevant datasets using the top retrieved results. The rationale is that commercial search engines have been well optimized, and we may use their results as features in our ranking methods. We used only unigrams as features in this study. Therefore, the performance of this

re-ranking method may be further improved if better features are extracted and noisy information is removed.

There are also several limitations in this project. Before indexing, concepts in both the metadata and additional information were not normalized. For example, transforming growth factor beta could be written as tgf-beta, tgf beta, or tgf-β. A query containing only tgf-beta will miss datasets that only have tgf-β in the metadata. In addition, the re-ranking algorithms did not consider complicated features such as named entities, which can also help filter out ambiguous results. Finally, disambiguation methods could also have been applied to the query expansion to avoid retrieval of irrelevant datasets.

Chapter 4, in part, has been submitted for publication of the material as it may appear in Finding Relevant Biomedical Datasets: the UC San Diego Solution for the 2016 bioCADDIE Retrieval Challenge. Wei, Wei; Ji, Zhanglong; He, Yupeng; Zhang, Kai; Ha, Yuanchi; Li, Qi; Ohno-Machado, Lucila, Database(Oxford), 2017. The dissertation author was the primary investigator and the author of this paper.

5 Biomedical articles

There are many ways to index datasets. Even though not all datasets are associated with articles, many of them are (e.g., articles describing studies in which the data set was used). For this reason, indexing of articles may indirectly contribute to indexing of data. Lessons learned from article indexing can also help us pave the way towards better data indexing.

Biomedical publication retrieval is a traditional domain of information retrieval research. The NLM hosts the largest collection of citations and abstracts for biomedical literature (i.e. MEDLINE), and provides retrieval service via PubMed. To help users identify desired articles from such a massive collection, the NLM assigns MeSH terms to specify the topics of the articles.

Historically, the NLM indexers manually assigned MeSH terms following a set of rules[25]. As the number of biomedical articles started growing rapidly in the 1990s, the indexers reached the limit of their processing capability. A statistical analysis from the NLM customer service showed that 25% of the articles were indexed within 30 days of receipt, 50% within 60 days, and 75% within 90 days[136]. To handle the ever-growing biomedical literature, NLM developed an automatic system, Medical Text Indexer (MTI)[137,138], to help the indexers select MeSH terms. In the meantime, researchers from across the world also contributed to the automation of MeSH term assignment[136,139–147]. Like biomedical dataset retrieval,

[25] https://www.nlm.nih.gov/bsd/disted/pubmedtutorial/015_010.html

the external information provides useful information to help improve the assignment performance[136].

This chapter provides a comprehensive review on the MeSH term assignment methods (string matching, machine learning, hybrid methods), and a NLM-performed benchmark study[136] that proved the significance of external information.

5.1 Introduction to MeSH term assignment methods

Widely used MeSH term assignment methods include string matching, machine learning-based methods, and hybrid methods.

5.1.1 String matching

Given an article and a set of MeSH terms, string matching counts common words or phrases in the article and MeSH terms, and chooses the MeSH term containing the largest overlap as most relevant[148]. This strategy was popular before the 1990s, but the success rate was low (typically 15%-20%)[149] due to two problems: (a) the method cannot capture synonyms; and (b) the method recognizes irrelevant words and phrases that partially share components with the desired MeSH terms[148]. To handle these problems, linguistic knowledge was applied to modify the common words or phrases. An example is MetaMap[132], an NLM developed tool for named entity recognition. MetaMap parses the article into sentences (utterances), utterances into phrases, and finds UMLS concepts in the phrases.

5.1.2 Machine learning

The machine learning-based approaches generally formulate MeSH term assignment as a classification question, such as a binary classification task on each MeSH term or a multi-label classification task on a list of MeSH terms. Many well-recognized models have been applied to classify MeSH terms, including support vector machine (SVM), Bayesian networks, k-nearest-neighbor (KNN), neural networks, naïve Bayes, logistic regression, decision tree, etc.

SVM is popular for binary classification of MeSH terms. It can work alone on individual MeSH terms[150,151], but more often it provides relevance probabilities for individual MeSH terms in a pipeline, such as the Meta-Labeler[152] framework. For example, Mao et al.[153], Tsoumakas et al.[154], Papanikolaou et al.[155,156] employed different SVM models to estimate the probabilities of MeSH term candidates and then used these estimations as features in downstream models.

Bayesian networks are a natural choice for modeling the hierarchical structure of the MeSH thesaurus. For example, Ribadas et al.[157] built a Bayesian network based on a top-down hierarchical classification scheme.

Deep learning methods have been proved to capture features previously undiscovered. Rios et al.[158] and Xu et al.[159] both applied convolutional neural networks to classify a set of MeSH terms that are difficult to assign. Peng et al.[160] utilized Word2Vec[161] and Document2Vec[162] packages to transform documents into embeddings, on which they trained classifiers. Jimeno-Yepes[142] explored the deep belief network that stacks restricted Boltzmann machine models.

K nearest neighbors (KNN) methods are widely accepted for identifying MeSH term candidates. For example, Mao et al.[153], Kamineni et al.[163], Ribadas et al.[164], Trieschnigg et al.[141], Liu et al.[165] and Yu et al.[166] all adopted KNN methods to find the most similar articles and then employed different strategies to identify MeSH term candidates from these articles.

Ensemble methods are another attractive option. Jimeno-Yepes et al.[144,145,167] compared multiple models (e.g. naïve Bayes, logistic regression, SVM, decision tree, Rocchio, and AdaBoostM1), and concluded that the models complemented each other. The group developed a voting algorithm to select the most appropriate methods for individual MeSH terms, which improved the performance.

Machine learning-based methods have shown outstanding performance, and dominated the field for two decades. They also face challenges, including the sparseness of the training data and the individual variations in tem assignment. The entire MEDLINE corpus or its subset are typical training data for MeSH term assignment. However, the number of articles indexed with a MeSH term (positive examples) is always much smaller than the number of articles not indexed with it (negative examples), i.e., an imbalanced data problem. Sohn et al.[139] explored an optimal training set strategy to solve this problem. On the evaluation side, the consistency of the gold standard is a problem. First, MeSH terms evolve over time: new MeSH terms become popular, while some fade away. Second, an early study suggests the inconsistency of indexers' selection of MeSH terms[168]. Therefore, if a model were trained on a set of MEDLINE records from the 1990s, it would

probably not perform well on a more recent test set. This is an argument for training in the recent years.

5.1.3 Hybrid

The hybrid strategy combines UMLS-based term recognition and machine learning-based methods[137,138,164,169]. This strategy can form complex systems such as MTI, which includes named entity recognition tool MetaMap[170], a machine learning module[52], KNN as in PubMed related citations, and a series of post-processing modules, which includes learning to rank and SUM-based ranking of candidate terms[171].

Given the title and abstract of a new article, MetaMap recognizes UMLS concepts and maps them to MeSH terms using ontology-based restrict-to-MeSH[110]. In parallel, the machine learning module uses the PubMed Related Citations (PRC) algorithm[52] to identify similar articles from the entire MEDLINE corpus, then collects MeSH terms assigned to the similar articles. A third machine learning module is used to improve performance on some of the most frequently used MeSH terms, such as “Human”[172]. The post-processing module merges MeSH term from the three modules, creates a candidate set, and determines their order based on the scores and rules[146]. Finally, the top-ranked candidates are recommended to the human indexers for review. Figure 6 illustrates the workflow of MTI.

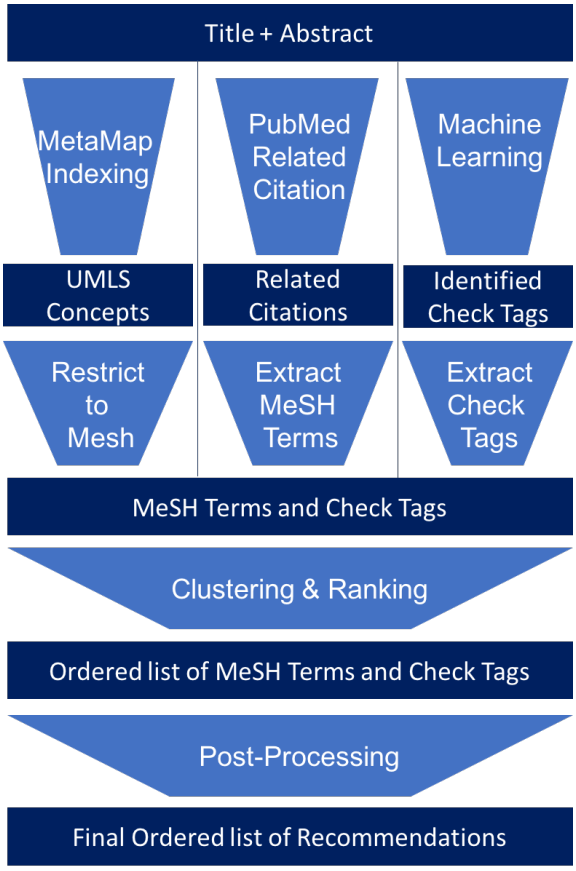


Figure 6. The workflow of MTI[137,138,146,172], modified from Mork et al.[146,172]. The MetaMap indexing captures MeSH terms from the target articles, PubMed Related Citation (PRC) module collects MeSH terms from k Nearest Neighbor articles, and a supervised machine learning method recommends candidates for most frequently used MeSH terms. Candidates from the three sources are pooled and ranked following pre-designed rules.

MTI proved to be a success for MeSH term assignment. In 2014, over 1.5 million articles were indexed at NLM; MTI provided MeSH term suggestions to NLM indexers for every incoming article, resulting in approximately 4,000 new MeSH assignments per day.

5.2 A benchmark study

In a retrieval task, a query is submitted and a ranking model computes a relevance score for each candidate document, and then all candidate documents are ranked based on the scores. This strategy also applies to MeSH term

assignment. When an article is provided, a set of MeSH term candidates are prepared (e.g., collected from MetaMap and the PRC module of MTI), then a ranking model computes a relevance score for every MeSH term candidate. Similarly, all MeSH term candidates are ranked based on the scores, and the top ranks are recommended to the NLM indexers.

In recent years, learning-to-rank (LTR) algorithms have been a popular choice for building relevance score functions in MeSH term assignment tasks[136,160,165,173]. Huang et al.[136] made one of the earliest attempts to approach MeSH assignment using a learning-to-rank algorithm. The method outperformed the MTI system, and was further experimented with more learning-to-rank algorithms[147] (e.g. RankNet[174], AdaRank[175], and LambdaMART[176]) to be developed into a new system MeSH Now[153], which won the 2014 BioASQ Challenge[177], and served as the baseline model in the 2015 BioASQ Challenge[169].

5.2.1 List-wise learning-to-rank

Like many other models at that time, Huang et al.[136] collected MeSH term candidates from similar articles for each target article. In particular, they modified the PRC algorithm[52] to retrieve KNN articles from the MEDLINE database for each target article. However, instead of simply summing the affinity scores between the target article and its neighbors, the group adopted a list-wise learning-to-rank algorithm ListNet[90] to estimate the relevance of candidates.

Given a target article D in a training set, a list of r MeSH term candidates $\{x_1, x_2, \dots, x_r\}$ is prepared, where every candidate x_i is represented by a feature vector $x_i = (x_1^i, x_2^i, \dots, x_k^i)$, where k is the number of features. Every candidate x_i

has a label 1 or 0; 1 means x_i is assigned by the human indexer while 0 means not. Therefore, labels form a list $Y = \{y_1, y_2, \dots, y_r\}$. The ranking function $f(x)$ assigns a score list $F = \{f_1, f_2, \dots, f_r\}$ based on $\{x_i\}$. Every element in Y and F is a relevance score of a MeSH term candidate estimated by the human indexer and the ranking function. Based on the relevance scores, the authors proposed two empirical distributions

$$\Pr(y_i) \propto \frac{\exp(y_i)}{\sum_{j=1}^{r} \exp(y_j)} \quad [29]$$

and

$$\Pr(f_i) \propto \frac{\exp(f_i)}{\sum_{j=1}^{r} \exp(f_j)} \quad [30]$$

The distance between $\Pr(y_i)$ and $\Pr(f_i)$ was minimized to also minimize the cross-entropy between the two distributions,

$$L(Y, F) = -\sum_{i=1}^{m} \Pr(y_i) * \log \Pr(f_i)) \quad [31]$$

$$f_i = f(x_i) = \sum_{l=1}^{k} w_l * x_l^i \quad [32]$$

where m is the number of training instances, and w_l is the weight of every feature learned from the training data.

5.2.2 Similar articles for MeSH term assignment

Features for MeSH term assignment have two important sources that have been widely used in MeSH term assignment studies[136,160,165]: the target article (i.e., the article needs MeSH terms), and external sources including articles similar to

the target article, and knowledge bases. Huang et al.[136] adopted five categories of 11 features that form a wide feature spectrum (Table 8). Among the five categories, neighborhood features are from similar articles, synonym features are from external knowledge bases, and the rest are from target articles.

Table 8. Five categories of 11 features. The content of the table is cited from Huang et al.[136] Among the five categories, neighborhood features are from similar articles, synonym features are from external knowledge bases, and the rest are from target articles.

Category	Feature
Neighborhood features	Number of neighbor documents in which a candidate MeSH term appears
	Summed document similarity scores
Overlap features	Number of unigrams overlapping between the MeSH term and the title or the abstract
	Number of bigrams overlapping between the MeSH term and the title or the abstract
Translation probability features	Probability of translating the title into a set of MeSH terms
	Probability of translating the abstract into a set of MeSH terms The assumption behind the features is that title and abstract is written in the authors' language, while MeSH terms are indexers' language. A translation model evaluates the translation probability, $$\Pr(MH\|text) = \frac{1}{n^m}\prod_{t_i \in MH; i=1}^{m}\sum_{s_j \in text; j=1}^{n}\Pr(t_i\|s_j) \quad [33]$$ where MH is a MeSH term, text is either a title or an abstract, m is the number of words in the MeSH term, n is the number of words in the text, t_i and s_j are single words in the MeSH term and in the text respectively.
Query-likelihood features	Two translation-based likelihood scores between a MeSH term and the title and abstract of an article when using the MeSH term as a query
	The Okapi scores between a MeSH term and the title and abstract of an article when using the MeSH term as a query
Synonym features	Whether one of the entry terms can be exactly matched to the title and abstract
	Whether there exists an entry term whose unigram words have all been observed in the document text

Although most features come from the target article, Huang et al.'s analysis showed that the two features from similar articles dominated the results (Table 9). This result suggested that the algorithm for selecting similar articles has substantial impact on the performance of MeSH term assignment.

Table 9. Feature ablation study cited from Huang et al[136]. The asterisks indicate significant differences from 'All features'. After removing the neighborhood features, all scores dropped significantly.

Feature set	Precision	Recall	F score	MAP
All features	0.39	0.712	0.504	0.626
Neighborhood features	0.315*	0.575*	0.407*	0.435*
Unigram/bigram features	0.389	0.711	0.503	0.626
Translation probability features	0.389	0.711	0.503	0.626
Query likelihood features	0.385	0.704	0.498	0.626
Synonym features	0.385	0.703	0.497	0.618
Only neighborhood features	0.370*	0.677*	0.478*	0.602*

6 Retrieval and similarity determination for biomedical articles

6.1 The importance of neighborhood features

6.1.1 Overview

Inspired by the results from Huang et al.[136], we employed Convolutional Neural Networks[178,179] (CNN) to extract features (CNN features) from target articles, and aimed at capturing information that was not covered by the manually created features in Huang et al.[136]. In our study, we analyzed the performance of CNN features in a point-wise learning-to-rank[91] framework for all of the MeSH terms. Also, we combined CNN features with features from similar articles, and studied the performance in the same framework. For each target article, the 20 most similar articles were identified from the entire MEDLINE collection using the PubMed Related Citation (PRC) algorithm[52], which was also employed by Huang et al.[136]. Since PRC is a k-Nearest-Neighbor (KNN) algorithm, we called similar articles from PRC KNN articles. Accordingly we called the features from KNN articles, KNN features (e.g. neighborhood features in Huang et al.[136]).

6.1.2 Methods

This section describes the methods used in CNN-based MeSH term assignment, including document model, CNN model, and point-wise learning-to-rank framework.

6.1.2.1 Document model

A document, such as an abstract, consists of a sequence of words. Using a distributed representation[180], a word is represented by a unique k-dimensional vector $x \in \mathbb{R}^k$ of real numbers. All words from the training, validation, and test sets form a vocabulary $V \in \mathbb{R}^{N_V \times k}$, where N_V is the size of the vocabulary. Given a document D of N_D words, it is represented as a $N_D \times k$ matrix, where row i is a k dimensional vector for the word x_i in i-th position of D. In our work, each MeSH term candidate was appended by its entry terms, which were given in the MeSH thesaurus, and the list of terms was treated as a document.

6.1.2.2 CNN model

We adopted the CNN model (Figure 7) from Kim[178] to extract features from target articles. In the model, a convolution layer extracted distributed representation features from the input article. The layer consisted of three components: the convolution operation, the non-linearity operation and the pooling operation.

In the convolution operation, we used wide convolution filters to recognize patterns (wide means padding the input matrix if it does not match the shape of a filter[91]). For each filter $w \in \mathbb{R}^{hk}$, where h is a window size, i.e. number of consecutive words this filter covers, a new matrix is generated after the convolution operation. For example, given a document fragment $x_{i:i+h-1}$ (the consecutive h words from the i-th position in a document), the convolution $w \cdot x_{i:i+h-1}$ generated a real number output.

After the convolution operation, a non-linearity operation (a.k.a. activation function) is applied to the output of the convolution filters. We chose the hyperbolic tangent as the non-linear function. Therefore, an outcome c_i is generated as $c_i = \tanh(w \cdot x_{i:i+h-1} + b)$, where b is a bias term for every filter. When a filter slides through all of the available windows in an input text matrix, the filter generates a list of features $c = [c_1, c_2, \dots, c_{N_D}]$. The size of the feature list is N_D, because we chose the wide convolution.

When filter outcomes were ready, we applied a pooling operation over the outcomes to aggregate the information, and generated features for ranking algorithms. We chose the max pooling method, which is frequently used to capture the most important feature. $\hat{c} = \max\{c\}$ was taken as the feature corresponding to the filter, and it was the input to the learning-to-ranking algorithm.

6.1.2.3 Point-wise learning-to-rank framework

We adopted a point-wise learning-to-rank framework from Severyn et al.[91] to integrate CNN features and KNN features for MeSH term assignment (Figure 8). When CNN feature vectors and KNN feature vectors were ready, we concatenated the vectors and fed them into the point-wise learning-to-rank algorithm to learn a ranking model. The ranking model was learned using stochastic gradient descent with a Python package Theano[181]. When the learning process was completed, we applied the ranking model to compute probabilities of MeSH term candidates for given target articles.

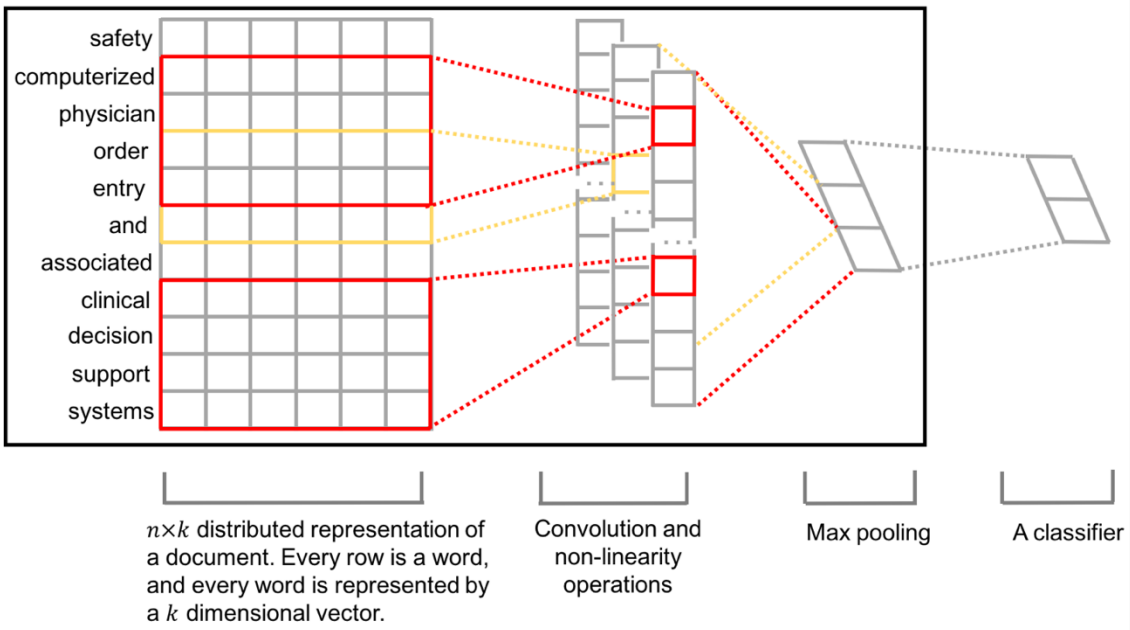


Figure 7. An illustration of Kim's CNN model[178] for sentence classification. We adopt the feature generation part in the black box. This figure is a modification of Figure 1 in Kim[178]. An example document "safety computerized … support systems" consists of 11 words. Every word is represented by a six-dimensional vector of random real numbers. The convolution layer extracts information from input texts using the multiple convolution filters and the non-linearity operation. The extracted information is aggregated by the max pooling operation.

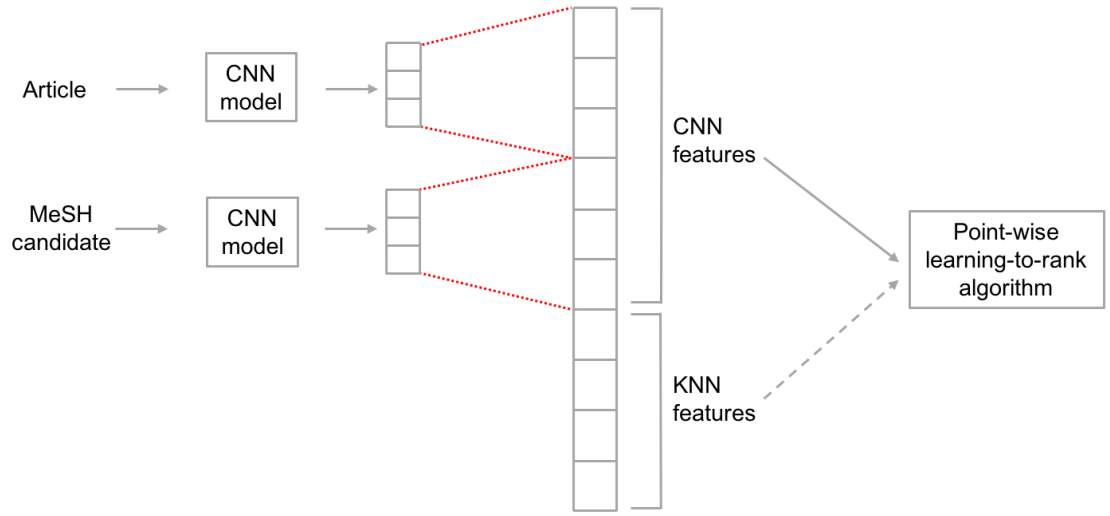


Figure 8. CNN features and KNN features are applied to learn a ranking model in a point-wise learning-to-rank framework. When KNN features are included, they are concatenated with the CNN features.

6.1.3 Data and evaluation

We made Huang et al.[136] the baseline model, and trained and tested our models on the same article collection that Huang et al. used. Every article came with a title, abstract, MeSH terms, and the 20 most similar articles.

The article collection includes three subsets: SMALL200, NLM2007, and L1000. SMALL200 is a set of randomly selected 200 article that received MeSH terms between 2002 and 2009. NLM2007 is a test set obtained from the NLM indexing initiative[137] created in 1997. L1000 includes randomly selected 1000 articles that received MeSH terms between 1961 and 2009.

The performance was measured using mean average precision (MAP), which is detailed in section 2.8. For each target article, the gold standard was the set of associated MeSH terms assigned by NLM, and the predicted MeSH terms were the top 25 MeSH terms from the point-wise learning-to-rank algorithm trained ranking model. The predictions were compared with the gold standard to compute MAP scores.

6.1.4 Implementation

Python package Theano[181] is employed for implementing CNN and the point-wise learning-to-rank algorithm. The scripts are available from https://github.com/w2wei/deep-qa. A virtual machine on the iDASH[135] cloud with 32 CPU (Intel(R) Xeon(R) 2.30GHz) and 32 GB RAM was used.

6.1.5 Results

In the baseline method[136], the ranking model was trained on SMALL200, and was evaluated on NLM2007 and L1000 separately. All articles in these

datasets were represented by 11 manually generated features. KNN features included the number of neighbor documents, in which a candidate MeSH term appears, and summed document similarity scores.

We also trained ranking models on SMALL200 and tested them on NLM2007 and L1000. However, articles are represented by CNN features alone or together with a KNN feature, i.e. the number of neighbor documents in which a candidate MeSH term appears. The results (Table 10) show that the performance of the combination of CNN features and one KNN feature approaches all manually generated features. However, CNN features alone cannot compete with all manual features. In fact, the two feature types demonstrate approximately equal contributions to the performance. The results are consistent with the observation in Huang et al.[136] that the KNN feature is critical for improving MeSH term assignment performance.

Table 10. MAP scores from different features. Ranking models were trained on SMALL200, and tested on NLM2007 and L1000 separately. The baseline method uses all eleven features in Huang et al.[136] The decimals are the mean average precision scores. The percentages correspond to the differences from the baseline values.

Feature set	NLM2007	L1000
All features in Huang et al.[136] (baseline)	0.626	0.615
CNN	0.335 (-46.4%)	0.307 (-50.1%)
CNN + KNN	0.602 (-3.8%)	0.584 (-5.0%)

6.2 Finding similar PubMed articles

6.2.1 Introduction

Based on the results from the above CNN study and the work by Huang et al[136]., we realized it is critical to find the most appropriate KNN articles from

PubMed for MeSH term assignment. Therefore, I studied the PubMed Related Citations (PRC) algorithm[52] developed by the NLM, which recommends related articles that may be of interest to users. A brief introduction to the PRC algorithm is available in section 2.5.2.2. For more comprehensive information, please refer to Lin and Wilbur[52] and PubMed Help[182].

Even though it is widely used, the PRC algorithm may not accurately recommend desired articles to the reader. In particular, two articles of different topics may have similar distributions of term counts; in such case, the PRC algorithm may conclude that the two articles are related and recommendable. For example, if two articles share descriptions of experimental techniques and related genes, but differ in the topic of disease mechanisms, the articles may have a large number of terms in common. On the other hand, if two articles discuss the same topic, but use different terms, the PRC algorithm is likely to miss this recommendation.

Our objective was to improve the PRC algorithm and to promote the selection of articles related to the same research topic. This was not the first attempt to do so, and much effort has been spent to improve the retrieval performance of related MEDLINE citations. For example, Fontaine et al.[183] developed MedlineRanker, which is a system that identifies the most discriminative words in query articles, and uses the words as query terms to retrieve related citations. Poulter et al.[184] developed a system named MScanner that trains a naïve Bayes classifier on MeSH terms and on journal identifiers extracted from a set of user-provided articles, and uses the classifier to select and rank related citations.

Both performed well when compared on nine topics, in terms of the area under the ROC curve[183]. However, these approaches were not very practical because both systems required users to provide a set of articles related to a query topic, rather than a few keywords or a short description. eTBLAST[185] is a method similar to PRC, but it determines similarity based on word alignment. Therefore, the length of the query text has significant impact on the retrieval performance. Boyack et al.[186] investigated the accuracy of five similarity metrics (PRC, BM25, topic modeling, latent semantic analysis, and term frequency-inversed document frequency) for clustering two million biomedical articles. The group concluded that PRC generated the most coherent and the most concentrated cluster solution. Aside from suggesting related articles to PubMed users, the PRC algorithm is used for other purposes as well. For example, Huang et al.[136] collected MeSH terms from articles recommended by the PRC algorithm for assignment of MeSH terms to a new article.

6.2.2 Methods

6.2.2.1 An extension of the PRC algorithm

Our approach extends the PRC algorithm by considering similar terms. In the PRC algorithm, a topic is associated with a single unique term. We relaxed the assumption in the modified algorithm, and allowed a topic to be associated with multiple similar terms. Similar terms were considered as important as the original term. We prepared similar terms for the vocabulary of TREC data using

Word2Vec[26], a package based on the Skip-gram model[161]. We trained distributed vector representations of terms with Word2Vec (vector size 100, minimum word count 40, window size 10) on three million MEDLINE citations that are available from the 2014 BioASQ Challenge[177], and derived similar terms by comparing cosine distances between associated vectors. Training takes a few hours on a computer with four 2.67GHz processors and 16 GB of RAM. The trained model and derived similar terms can be reused for other PubMed article retrieval tasks.

We expanded terms in the query article to a set that includes the original term and the five most similar terms according to the trained Skip-gram model. The expansion allows approximate term matching: for a particular term in the article, if one of its similar terms occurs in a candidate related article, then the similar term is treated as the original one, and the contribution of this pair of terms is included in the similarity score. Therefore, articles that focused on the same topic but used different terms had a higher chance of being connected.

Given an article c for a particular query article d, in the term weight function $w_{\mathrm{t,c}}$ we changed the term frequency k to $p \sum_i k_i$, where $\sum_i k_i$ is count of term t and its similar terms in article c, and p is the ratio of the count of term t in article d over the count of all terms in article d.

$$w_{\mathrm{t,c}} = \frac{\sqrt{idf_t}}{1 + \left(\frac{\mu}{\lambda}\right)^{p\sum_i(k_i)-1} \mathrm{e}^{-(\mu-\lambda)l}} \qquad [34]$$

[26] https://code.google.com/p/word2vec/

The term weights in article d are not changed. Therefore, the similarity score $P(c|d) = \sum_{t=1}^{N} w_{t,d} * w_{t,c}$ is asymmetric, and depends on the set of terms in query article d.

6.2.2.2 Experimental design

We evaluated the performance of our eXtended PRC algorithm (XPRC) on two datasets separately: (1) 4,584 articles (utilizing only title, abstract and MeSH terms) from the TREC 2005 Genomics Track evaluation dataset (Genomics data)[27], and (2) 3034 articles (utilizing only title and abstract) from the TREC 2014 Clinical Decision Support Track evaluation dataset (CDS data)[28]. Among the 4,584 Genomics articles, we identified 4,234 valid ones for evaluation[29]. The valid articles were assigned to 50 TREC official topics (i.e. information needs); the 3,034 CDS articles were assigned to 30 topics; one article could be assigned to multiple topics. If an article was labeled as "possibly relevant" or "definitely relevant" to a topic, we assigned the article to the topic. If two articles had topics in common, we considered them to be "similar" in our evaluation.

In the evaluation step, within each dataset, each article served as a query article, and the remaining articles were ranked according to the PRC or XPRC similarity[30]. For each query article, the PRC algorithm recommended articles that

[27] http://skynet.ohsu.edu/trec-gen/data/2005/genomics.qrels.large.txt
[28] http://trec-cds.appspot.com/qrels2014.txt
[29] Among the 4584 PMID in the TREC evaluation dataset, 92 PMIDs appear multiple times, 1 PMID no longer exists, 248 PMIDs have no abstract, 6 PMIDs have problems with PRC (i.e., the most similar article is not itself), 2 PMIDs have problems with Lucene, the indexing software. After removing all these articles, 4234 articles were used for the experiments.
[30] Code is available from https://github.com/w2wei/XPRC.git

were assigned into the true positive (TP) group or the false positive (FP) group according to the TREC gold standard. If the recommended article and the query article shared the same topic, this was considered a TP result. If the recommended article and the query article had no topics in common, this was considered a FP result.

We processed all of the text following NLM Help[182]. For example, we split the abstract and title of each article into terms following the PRC tokenization rules. In addition, terms were stemmed using the Porter stemmer. To understand why the PRC algorithm had false positives, we compared the number of matched terms in the sets of TP and FP articles under multiple conditions, and measured the Kullback–Leibler (K-L) divergence of normalized weight distributions between articles and corresponding recommended articles obtained using the PRC algorithm. The term matching was based on string comparison, and the weights were calculated using the formula described in the NLM fact sheet[187]. When comparing two articles, matched terms were kept and the associated weights were normalized.

6.2.2.3 Evaluation measures

We used precision at the threshold of five articles in the same way as described for the development of the PRC algorithm. In addition, we also measured average precision (AP) and mean average precision (MAP) at the same threshold. The definitions of precision, AP and MAP are available in section 2.8.

6.2.3 Results

6.2.3.1 Evaluation of the PRC algorithm

We recorded some characteristics of TP articles and FP articles. First, as expected, the average number of matched terms in TP articles is different from the number in FP articles. The average number of matched terms in TP was 29, and the average number in FP was 24. We used an independent two sample t-test on the 4,234 Genomics articles to test the null hypothesis that the average number of matched terms in TP was equal to the number in FP. The p value was 9e-139, hence the hypothesis was rejected, as expected.

Next, we considered the normalized weight distributions of matched terms in TP and FP articles. The average K-L divergence between a query article and the TP articles from the PRC algorithm recommendations was 0.18, while the divergence between a query article and its FP recommendations was 0.21. Using an independent two sample t-test and the Genomics dataset, we tested the null hypothesis that the average K-L divergence between query articles and their TP articles was equal to the average K-L divergence between query articles and the FP articles. The p value was 3e-93, hence the hypothesis was rejected, as expected.

Finally, we analyzed PRC's capability to match terms at various PRC weight thresholds, for TP and FP articles. We used a series of independent two sample t-tests to test the null hypothesis that the count of matched high-weight terms in the set of TP articles was equal to the count of high-weight terms in the set of FP articles at different weight thresholds on the Genomics dataset. As we increased

the threshold from 0 to 1.8, there was a significant change in the counts of matched high-weight terms in the two groups (Figure 9), except for a small region in which the null hypothesis of equal counts could not be rejected. When the weight threshold was lower than 0.75, TP articles matched significantly more high-weight terms than FP articles. However, when the threshold was over 0.8 (i.e., only terms with weight over 0.8 were considered in computing the similarity score), FP articles matched significantly more terms than TP articles. This result conflicts with our intuition that TP articles should always share more meaningful and important terms with the query article than FP articles. In the experiments, we observed that PRC high-weight terms were not necessarily the critical terms in an article (i.e., terms directly related to the focus of the article, such as disease names, gene names). High-weight terms were often general terms, such as “gene”, “protein” and “disease”. When critical terms are missing from matched terms, terms that are less relevant to the focus of the article make major contributions to the similarity score. If there are large numbers of such high-weight matched terms, a FP article is recommended. For example, the PRC algorithm recommends article PMID11480035 as related to article PMID10226605, although the two articles are not of the same topic according to the TREC evaluation dataset. The articles match in high-weight terms, such as “mucosa”, and “mRNA”, but PMID11480035 lacks critical terms, such as “APC”, “colon” and “colorectal”.

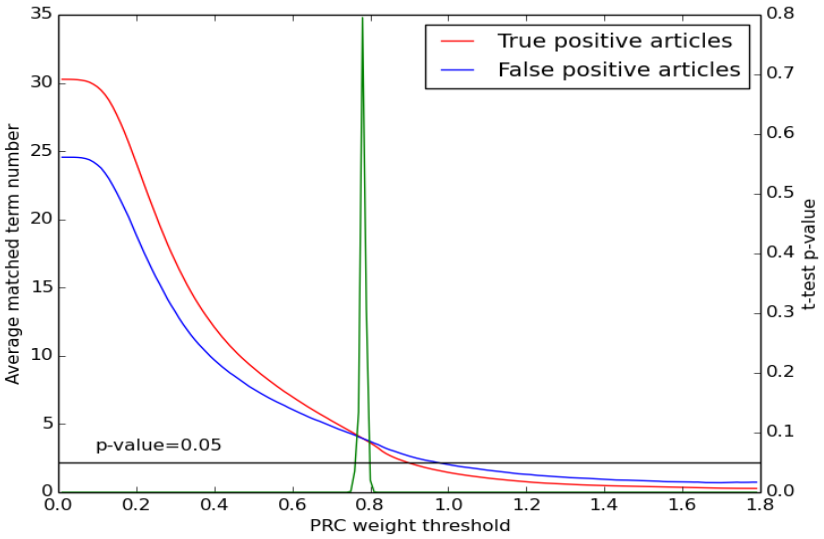


Figure 9. A comparison of the number of matched term counts at different PRC weight thresholds. The red curve is the smoothed trend of matched terms in TP articles. The blue curve is the smoothed trend of matched terms in FP articles. The two curves cross between X= 0.76 and X=0.8. The green curve illustrates the p-value of the difference between TP and FP for the null hypothesis that the count of matched terms in TP is equal to the count of terms in FP above different weight thresholds. When 0.76<X<0.8, we cannot reject the null hypothesis of equal counts. Only 0.14% of all term occurrences have weights over 1.8. Therefore, we do not show these special cases in this figure.

6.2.3.2 XPRC: eXtended PRC algorithm

Term expansion is an effective approach to improve the performance of the PRC algorithm. The expansion helps the PRC algorithm recognize articles on related topics, even though they do not have matched critical terms. We wanted to understand in which situations XPRC could potentially enhance the results of PRC. First, we stratified the articles according to precision and AP of the PRC algorithm. After that, we ran XPRC on every stratum of data and compared its performance with PRC. The results of XPRC and the comparisons stratified by precision and AP are shown in Figures 10 and 11, and in Tables 11, 12, 13 and 14. The results

show that the XPRC algorithm achieves better performance than the PRC algorithm for certain categories of cases.

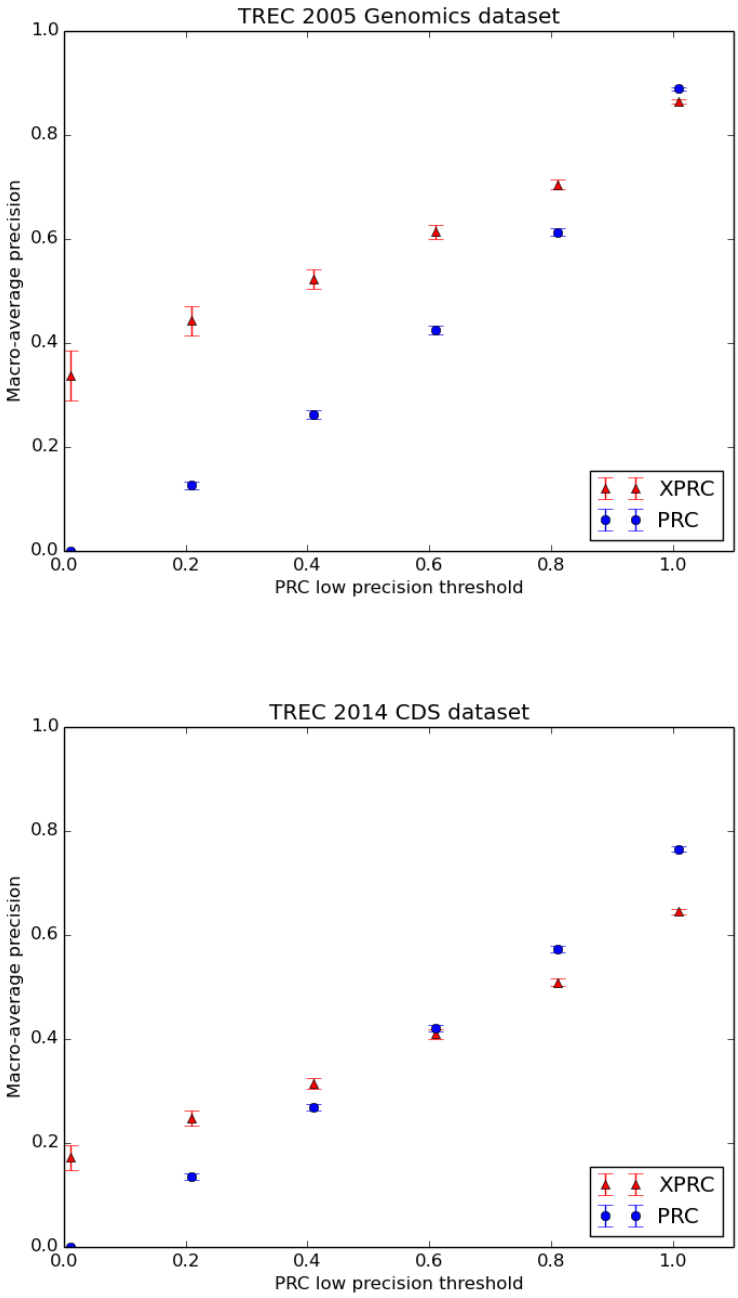


Figure 10. A comparison of PRC and XPRC at five precision levels determined by the PRC algorithm on the Genomics dataset and CDS datasets. For the Genomics articles in which PRC does not achieve perfect precision, XPRC has better overall performance in all but one group. For the CDS articles, XPRC achieved better performance in PRC's low precision articles. Values associated with every data point are available in Tables 11 and 12. The error bars indicate standard errors.

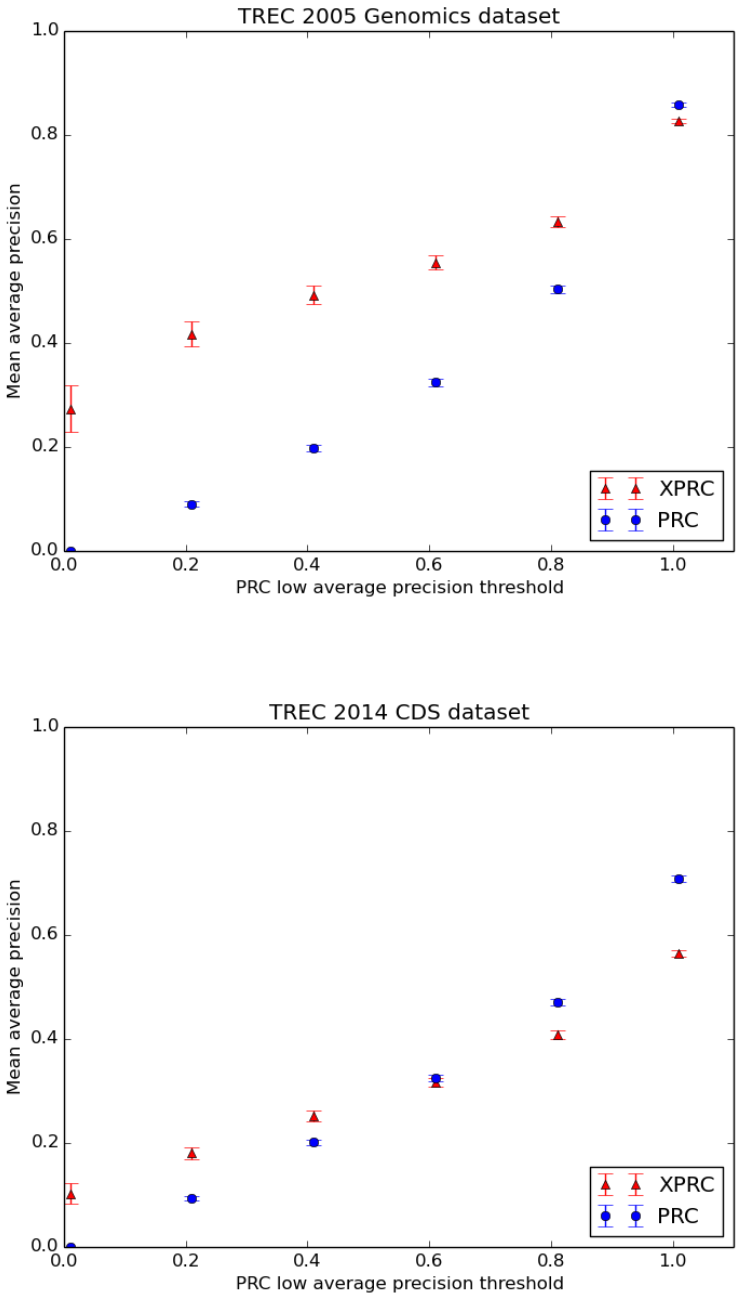


Figure 11. A comparison of PRC and XPRC at five average precision (AP) levels determined by the PRC algorithm on the Genomics dataset and CDS dataset. For the Genomics articles in which PRC does not achieve perfect AP, XPRC has better performance in all but one group. For the CDS articles, XPRC achieved better performance in PRC's low precision articles. Values of every data point are available in Tables 13 and 14. The error bars indicate standard errors.

Table 11. A comparison of PRC and XPRC at different precision levels determined by the PRC algorithm on the Genomics dataset. The cumulative article count is the number of articles with PRC precision below a given precision level. For example, there are 158 articles that result in PRC precisions lower than 0.2. PRC has precision 0.0 on all the 58 articles in the 0.0 group, so its macro-average precision and standard error are also 0. XPRC has better performance on these articles. p-value shows the significance of the difference between PRC and XPRC at different precision levels.

Precision levels		0.0	0.2	0.4	0.6	0.8	1.0
Cumulative article counts		58	158	314	603	1215	4234
PRC	macro-average precision	0	0.127	0.262	0.424	0.613	0.889
	standard error	0	0.008	0.009	0.008	0.007	0.003
XPRC	macro-average precision	0.338	0.443	0.523	0.614	0.705	0.864
	standard error	0.048	0.028	0.019	0.013	0.009	0.004
p-value		3e-09	3e-21	3e-30	2e-31	2e-16	4e-07

Table 12. A comparison of PRC and XPRC at different precision levels determined by the PRC algorithm on the CDS dataset. The format of this table is the same as that of Table 11.

Precision levels		0.0	0.2	0.4	0.6	0.8	1.0
Cumulative article counts		87	268	539	1000	1670	3034
PRC	macro-average precision	0	0.135	0.268	0.421	0.573	0.765
	standard error	0	0.006	0.006	0.006	0.006	0.005
XPRC	macro-average precision	0.172	0.248	0.315	0.410	0.509	0.645
	standard error	0.024	0.015	0.011	0.009	0.007	0.005
p-value		2e-10	2e-11	3e-4	0.294	4e-12	6e-57

Table 13. A comparison of PRC and XPRC at five average precision (AP) levels determined by the PRC algorithm on the Genomics dataset. The cumulative article count is the number of articles with PRC AP below the given AP level. XPRC has better performance at all levels except for 1.0. p-value shows the significance of difference between PRC and XPRC at different AP levels.

Average precision levels		0.0	0.2	0.4	0.6	0.8	1.0
Cumulative article counts		58	231	420	687	1215	4234
PRC	MAP	0	0.09	0.198	0.324	0.504	0.858
	standard error	0	0.005	0.007	0.007	0.007	0.004
XPRC	MAP	0.274	0.417	0.493	0.555	0.633	0.827
	standard error	0.045	0.024	0.018	0.014	0.01	0.004
p-value		1e-07	8e-31	2e-45	2e-45	2e-25	1e-07

Table 14. A comparison of PRC and XPRC at five average precision (AP) levels determined by the PRC algorithm on the CDS dataset. The format of this table is the same as that of Table 13.

Average precision levels		0.0	0.2	0.4	0.6	0.8	1.0
Cumulative article counts		87	363	663	1079	1670	3034
PRC	MAP	0	0.094	0.201	0.325	0.471	0.709
	standard error	0	0.004	0.005	0.006	0.006	0.006
XPRC	MAP	0.103	0.181	0.253	0.317	0.408	0.565
	standard error	0.020	0.011	0.010	0.009	0.008	0.006
p-value		2e-6	3e-12	8e-6	0.446	2e-10	6e-62

6.2.3.3 The scalability of XPRC

We compared the time and memory usage for running queries using PRC and XPRC (Table 15). We ran queries on different sizes of corpora and recorded the time and maximum memory usage. The algorithm and its implementation can still be further optimized.

Table 15. Time and memory usage of PRC and XPRC. The corpora were randomly selected from the Genomics dataset. For each algorithm, we ran 10 queries on every corpus. The time shown in this table is the average value of all queries on every corpus. The memory in this table is the maximum value for all queries on every corpus.

	PRC		XPRC	
Corpus Size	Time (s)	Maximum Memory (MB)	Time (s)	Maximum Memory (MB)
10	0.08	65	2.1	590
100	0.15	68	2.2	591
1000	0.7	144	2.6	601

6.2.4 Discussion

The gold standard is critical in the measurement of model performance. In this study, the gold standards were provided in the 4584 annotated articles in the TREC 2005 Genomics Track data and the 3034 TREC 2014 Clinical Decision Support Track data. One issue with the gold standard is that there were a large number of articles under every topic. The average number of articles per topic in the Genomics dataset was 815 and this number was 1264 in the CDS dataset. This issue sometimes makes PRC and XPRC indistinguishable in terms of precision and AP: PRC and XPRC make different recommendations for the same query, but all of their recommendations are true positives.

Our data-driven approach provided similar terms that could not be found in traditional synonym dictionaries. One limitation of the XPRC algorithm is that the expansion was applied to every term in the query article. This may introduce undesired expansion to non-critical terms. In addition, the parameters were not optimized for our experimental setting. We used the μ and λ proposed by Lin and Wilbur[52]. To further improve the performance of the XPRC algorithm, we could

develop targeted term expansion and optimize the parameters on the TREC evaluation dataset. However, the algorithm needs to be applied to more annotated corpora so we can confirm our results and evaluate its scalability. From the analysis of the PRC algorithm, we confirm that TP articles and FP articles have different distributions of term weights, and that the majority of articles achieve perfect precision, but a significant number of them still result in low precision, leaving some room for improvement.

We could explore heuristic methods to select when to use the PRC or XPRC results, but there is no simple solution. An empirical method[188] achieved good performance on the Genomics dataset (i.e., better performance than PRC in all conditions) but it did not perform as well on the CDS dataset.

The principle of extending a set of terms to assess similarity can be utilized in other problems in which the goal is to find related objects. For example, XPRC can be used to find a set of articles that report on analyses on a particular data set of interest (i.e., articles that are related to the one that first described or utilized the data set). These articles may point to derived data or new and related data sets of interest. Term expansions could also be used for meta-data in the same way we used them for embedded terms in titles and abstracts.

Chapter 6, in part, is a reprint of the material as it appears in Finding Related Publications: Extending the Set of Terms Used to Assess Article Similarity. Wei, Wei; Marmor, Rebecca; Singh, Siddharth; Wang, Shuang; Demner-Fushman, Dina; Kuo, Tsung-Ting; Hsu, Chun-Nan; Ohno-Machado, Lucila, AMIA Summits

on Translational Science Proceedings, 2016. The dissertation author was the primary investigator and author of this paper.

7 Conclusions

7.1 Dissertation summary

Biomedical dataset indexing is an emerging, fertile area for information retrieval research. Making use of external resources (e.g. additional linked information) to enrich the characterization of the data objects has the potential to improve the effectiveness of biomedical dataset retrieval.

In the biomedical dataset retrieval study (Chapter 4), I developed a pipeline to collect additional linked information for datasets, transform users' free-text requests to queries, and rank relevant datasets using a "retrieval plus re-ranking" strategy. To improve the representation of biomedical datasets, we explored online resources and collected information to enrich the metadata of datasets. In particular, we experimented with three key fields (title, description, and overall design) of the studies associated with datasets. The results showed that removing noise is critical to make use of online resources. The rule-based query formulation module extracted keywords from users' free-text requests, expanded the keywords using NCBI resources, and finally formulated Boolean queries using pre-designed templates. The module is not yet robust to handle all types of free-text requests, but it has laid a foundation for intelligent query generation mechanisms when user query logs are available. The novel "retrieval plus re-ranking" strategy captured relevant datasets in the retrieval step, and ranked datasets using the customized relevance scoring functions that models unique properties of the metadata of

biomedical datasets. The “retrieval plus re-ranking” strategy is open to various retrieval and ranking models, thus there remains much to be explored.

Linked information can exist in the form of articles that cite datasets. Therefore, understanding and improving upon methods to index articles may indirectly help with the difficult task of indexing datasets. We thus extended our work to explore improvements to automated MeSH assignment and article similarity determination. During a MeSH term assignment study, we realized that similar articles determination is a key external resource for MeSH term candidates. Therefore, we studied the retrieval of biomedical publications, and developed an algorithm to find similar articles in PubMed (Chapter 6). Currently, similar articles in PubMed are determined by the PRC algorithm[52]. However, the PRC algorithm may not always find correct articles. In particular, when the distributions of term counts are similar, the PRC algorithm is likely to conclude that the articles are similar, even though they may be about different topics. For example, if two articles detailing the mechanisms of different diseases describe similar techniques and mention related genes, the articles may have a large number of terms in common. On the other hand, when two articles discuss the same topic but use different terms, the PRC algorithm is likely to miss the similarity. For the above problem, we implemented a term expansion method to improve the PRC algorithm, dubbed the eXtended PRC (XPRC). Unlike popular ontology-based expansion methods, we used the Word2Vec model[161] to learn distributed representations of terms over one

million articles from PubMed Central[31], and identified similar terms using the Euclidean distance between distributed representation vectors. We showed that, under certain conditions, using the query expansion based on a distribution representation can improve precision, and helps find similar PubMed articles.

7.2 Future work

A natural extension of the dissertation is to develop advanced representation models for biomedical objects. Token normalization and information integration are two fundamental tasks in representation model design. Token normalization helps match character sequences when they are not identical but are expected to be matched, and information integration helps enrich the representation of objects. In this section, I propose future work on improving token normalization and information integration.

7.2.1 Token normalization

When an information retrieval system receives a query, the system breaks the query into tokens, and identifies documents containing the tokens for potential relevance to the query.

Manning et al.[31] discussed two widely-used strategies to implement token normalization: creating equivalence classes and maintaining relations between unnormalized tokens; and "the most standard way to normalize is to implicitly create equivalence classes, which are normally name after one member of the set". For example, 'anti-discriminatory' and 'antidiscriminatory' are mapped to the latter.

[31] https://www.ncbi.nlm.nih.gov/pmc

However, the equivalence class strategy excels at removing characters from tokens, but is mediocre at adding characters (e.g. mapping ‘antidiscriminatory’ to ‘anti-discriminatory’).

A deep learning-based solution may enable character addition in the equivalence classing strategy. The deep learning methods can transform tokens into distributed representation vectors (i.e. embeddings, embedding vectors) of high dimensions, such as Word2Vec[161]. Based on the distributed representation vectors, it is possible to identify related tokens according to the Euclidean distances between vectors. Once related tokens are identified, rule-based algorithms (e.g., editing distance based methods) may be applied to find morphologically similar tokens among the related tokens, and find tokens with more characters (i.e. character addition). Moreover, a recent deep learning model, the transE model[189] and its derivative models improve upon the Word2Vec by further encoding the manually defined hierarchical relations among terminologies into distributed representation vectors. The equivalent tokens inferred from the transE model family will naturally include the hierarchical relations, and thus potentially provide more benefits to token matching in the retrieval.

7.2.2 Data-literature integration

In this dissertation, we studied retrieval tasks for datasets and articles in an independent fashion. In the future, it will be important to develop methods in which the tasks are combined in a systematic fashion. Citing articles are one way to approach this. Cited articles may be another. Associated articles, especially primarily associated articles (i.e. articles that announce the existence of the

datasets), are expected to help enrich metadata, and to make datasets more discoverable. However, it is not yet clear how to make efficient use of the associated articles.

Articles are complex objects, rather than mere texts. At the text level, articles are split into multiple sections including title, abstract, main text, references, appendix, etc. The sections may be further split into sub-sections, such as introduction, background, methods, results, and conclusions. Beyond the text, latent features, such as topics, can be extracted from text using machine learning methods or manually labeled according to certain controlled vocabularies.

To maximize the contribution of associated articles, I propose the integration of associated articles with metadata in two ways.

First, one can expand the metadata schema. In this approach, dedicated metadata attributes are designed for selected parts of articles, such as the title, the conclusion in abstract, and the topic distribution of the main text. Moreover, the approach does not change the retrieval process. In Chapter 4, we discussed the contribution of additional information to the retrieval of biomedical datasets. The results showed that the additional information itself slightly improved the infNDCG, potentially due to the noise. This issue may also apply to the associated articles, which may contain more irrelevant information, such as the background. Therefore, it is worthwhile to study the contribution of each sub-section of the abstract.

Second, one can build a separate index for selected parts of articles. In this approach, a separate index can be built for each section (e.g. title) or their combination (e.g. title and abstract), and the results from each index will be merged

with the results from the metadata index. The approach is flexible in weighting the contribution from each section of article. For example, if the title and the conclusions in abstract were the most informative part of articles, we would assign or learn larger weights for results from the corresponding indices. Accordingly, the final ranking list of relevant datasets receive more impact from the highly-weighted parts of articles. Compared with the first approach, it is also easier to interpret the final results when using multiple indices for each section of the associated articles.

7.3 Final remarks

Information retrieval techniques have advanced across the entire field, in the form of representation models fulfilling the requirements of different applications, indexing strategies of higher efficiency in space or time, retrieval models with improved performance in identifying the most relevant objects, and evaluation metrics designed for a variety of purposes. In biomedical research, information retrieval techniques have helped researchers find desired publications, datasets, and other information with ease. Further research on developing more robust representation models, more intelligent query formulation systems, and more scalable, efficient, and accurate ranking models will lead to smarter and more friendly information retrieval systems for biomedical research, and further advance the transformation from data to knowledge in biomedicine.

Appendix A. The standard fields for 20 data repositories

YPED
"dataset.title","dataset.description", "organism.name"

ProteomeXchange
"dataset.title", "keyword","organism.name"

PhysioNet
"dataset.title","dataset.description","organism.name"

Phenodisco
"topic", "MESHterm", "phenCUI"(UMLS concepts), "title", "Demographics",
"demographics","inexclude", "desc","disease", "phenDesc", "gender",
"organism.name"

Peptideatlas
"dataset.title","dataset.description","treatment.description",
"organism.name","organism.strain"

PDB
"materialEntity.name", "dataItem.keywords", "dataItem.title",
"dataItem.description","citation.title", "gene.name",
"organism.source.scientificName", "organism.source.strain",
"organism.host.scientificName", "organism.host.strain"

Openfmri
"dataset.title","dataset.description", "organism.name"

Nursadatasets
"publication.description", "dataset.keywords", "dataset.title","dataset.description",
"organism.name"

Neuromorpho
"dataset.title", "dataset.note", "treatment.title",
"organism.strain","organism.scientificName","organism.name",
"organism.gender", "anatomicalPart.name"

MPD
"dataset.title","dataset.description",
"organism.strain","organism.scientificName","organism.name", "dataset.gender"

GEO
"dataItem.description","dataItem.title", "organism", "source_name"

Gemma
"dataItem.title","dataItem.description", "organism.source.commonName"

Dryad Data Repository
"dataset.title", "dataset.keywords"

Dataverse Network Project
"publication.description", "dataset.title","dataset.description"

CVRG
"dataset.title","dataset.description"

CTN
"dataset.title","dataset.description", "organism.scientificName", "organism.name"

Clinicaltrials
"Study.recruits.criteria" (inclusion and exclusion criteria), "Treatment.description",
"Dataset.briefTitle", "Dataset.keyword", "Dataset.title","Dataset.description"

CIA
"disease.name", "dataset.title", "anatomicalPart.name",
"organism.scientificName", "organism.name"

Bioproject
"dataItem.description","dataItem.title","dataItem.keywords",
"organism.target.species"

Arrayexpress
"dataItem.description","dataItem.title"

Appendix B. The full results of 2016 bioCADDIE Challenge.

Group	Submission	infAP	infNDCG	NDCG@10	P@10 (+partial)	P@10 (-partial)
University of California, San Diego	Elasticsearch	0.2446	0.4333	0.4228	0.5200	0.2733
	PSD-allwords	0.2792	0.4980	0.6152	0.7600	0.3267
	PSD-keywords	0.2391	0.4490	0.4088	0.5200	0.1667
	Distribution Shift	0.3309	0.4783	0.6504	0.7467	0.3600
	Ensemble	0.2801	0.4847	0.5398	0.6800	0.2400
University of Melbourne	1	0.202	0.3657	0.5200	0.6733	0.2067
	2	0.1985	0.3664	0.5129	0.6867	0.2000
	3	0.1815	0.3843	0.5298	0.7067	0.2000
	4	0.2568	0.4017	0.5366	0.7000	0.2733
	5	0.2436	0.3838	0.6325	0.7733	0.3333
Elsevier	1	0.2789	0.4292	0.5271	0.7000	0.2667
	2	0.2963	0.3925	0.5242	0.7067	0.2667
	3	0.2810	0.4219	0.5514	0.7133	0.3667
	4	0.3049	0.4368	0.6861	0.8267	0.4267
	5	0.3283	0.4235	0.6011	0.7133	0.3400
Emory University	1	0.2314	0.3985	0.4761	0.6267	0.2067
	2	0.2278	0.4011	0.4891	0.6000	0.2333
	3	0.2471	0.4241	0.5296	0.6933	0.2200
	4	0.2818	0.4173	0.5538	0.7200	0.2667
Harbin Institute of Technology	1	0.0998	0.3000	0.2043	0.2933	0.0533
	2	0.0683	0.2539	0.0991	0.1467	0.0333
	3	0.1957	0.3710	0.5265	0.6467	0.2800
	4	0.1185	0.2810	0.1512	0.2333	0.0867
	5	0.2576	0.3850	0.5472	0.7000	0.2800
Poznan University of Technology	1	0.0876	0.3580	0.4265	0.5333	0.1600

Group	Submission	infAP	infNDCG	NDCG@10	P@10 (+partial)	P@10 (-partial)
Mayo Clinic	1	0.1393	0.3485	0.5735	0.7267	0.2600
	2	0.1424	0.3516	0.5726	0.7467	0.2533
	3	0.1077	0.3006	0.4406	0.5333	0.2267
	4	0.1423	0.3253	0.4453	0.5400	0.2333
	5	0.1628	0.3933	0.5243	0.6667	0.2600
Oregon Health & Science University	1	0.3193	0.3965	0.6006	0.7467	0.3333
	2	0.1396	0.4024	0.3953	0.4800	0.1933
	3	0.1921	0.4405	0.5345	0.6533	0.2800
	4	0.2862	0.4454	0.6122	0.7600	0.3333
	5	0.083	0.3156	0.2531	0.3400	0.1133
Swiss Institute of Bioinformatics	1	0.3006	0.3898	0.5736	0.7067	0.3200
	2	0.2997	0.3864	0.5726	0.7067	0.3267
	3	0.3008	0.3875	0.5718	0.7067	0.3267
	4	0.3458	0.4258	0.5237	0.6600	0.3267
	5	0.3664	0.4188	0.6271	0.7533	0.3467
University of Illinois Urbana-Champaign	1	0.3054	0.4207	0.4877	0.6400	0.2667
	2	0.2246	0.3877	0.4724	0.5733	0.2333
	3	0.304	0.4185	0.4659	0.6133	0.2600
	4	0.2569	0.3959	0.4675	0.5800	0.2333
	5	0.3228	0.4502	0.5569	0.7133	0.2867

References

1. Salton G. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall, Inc.; 1971.

2. Canese K. PubMed celebrates its 20th anniversary. *NLM Tech Bull*. 2016;(410):e12.

3. Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW. GenBank. *Nucleic Acids Res*. 2013;41(Database issue):D36-42. doi:10.1093/nar/gks1195.

4. Metzker ML. Sequencing technologies - the next generation. *Nat Rev Genet*. 2010;11(1):31-46. doi:10.1038/nrg2626.

5. Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, Moore S, Phillips S, Maffitt D, Pringle M, Tarbox L, Prior F. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. *J Digit Imaging*. 2013;26(6):1045-1057. doi:10.1007/s10278-013-9622-7.

6. Marcus DS, Fotenos AF, Csernansky JG, Morris JC, Buckner RL. Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults. *J Cogn Neurosci*. 2010;22(12):2677-2684. doi:10.1162/jocn.2009.21407.

7. Zarin DA, Tse T, Williams RJ, Califf RM, Ide NC. The ClinicalTrials.gov results database - update and key issues. *N Engl J Med*. 2011;364(9):852-860. doi:10.1056/NEJMsa1012065.

8. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*. 2000;101(23):e215-e220. doi:https://doi.org/10.1161/01.CIR.101.23.e215.

9. Bumgarner R. Overview of DNA microarrays: types, applications, and their future. *Curr Protoc Mol Biol*. 2013;Chapter 22:Unit 22.1. doi:10.1002/0471142727.mb2201s101.

10. Lo YMD, Corbetta N, Chamberlain PF, Rai V, Sargent IL, Redman CW, Wainscoat JS. Presence of fetal DNA in maternal plasma and serum. *Lancet*. 1997;350(9076):485-487. doi:10.1016/S0140-6736(97)02174-0.

11. Berger SL, Kouzarides T, Shiekhattar R, Shilatifard A. An operational definition of epigenetics. *Genes Dev*. 2009;23(7):781-783.

doi:10.1101/gad.1787609.

12. Lieberman-Aiden E, van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie BR, Sabo PJ, Dorschner MO, Sandstrom R, Bernstein B, Bender MA, Groudine M, Gnirke A, Stamatoyannopoulos J, Mirny LA, Lander ES, Dekker J. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*. 2009;326(5950):289-293. doi:10.1126/science.1181369.

13. Karlebach G, Shamir R. Modelling and analysis of gene regulatory networks. *Nat Rev Mol Cell Biol*. 2008;9(10):770-780. doi:10.1038/nrm2503.

14. Haines LL, Light J, O'Malley D, Delwiche FA. Information-seeking behavior of basic science researchers: implications for library services. *J Med Libr Assoc*. 2010;98(1):73-81. doi:10.3163/1536-5050.98.1.019.

15. Grefsheim SF, Rankin JA. Information needs and information seeking in a biomedical research setting: a study of scientists and science administrators. *J Med Libr Assoc*. 2007;95(4):426-434. doi:10.3163/1536-5050.95.4.426.

16. Wildemuth BM, Moore ME. End-user search behaviors and their relationship to search effectiveness. *Bull Med Libr Assoc*. 1995;83(3):294-304.

17. Kharazmi S, Karimi S, Scholer F, Clark A. A study of querying behaviour of expert and non-expert users of biomedical search systems. In: *Proceedings of the 2014 Australasian Document Computing Symposium*. New York, USA: ACM Press; 2014:10-17. doi:10.1145/2682862.2682871.

18. Salton G. *Automatic Information Organization and Retrieval.* McGraw-Hill; 1968.

19. Cleverdon C. The Cranfield tests on index language devices. *Aslib Proc*. 1967;19(6):173-194. doi:10.1108/eb050097.

20. Singhal A. Modern information retrieval: a brief overview. *Bull IEEE Comput Soc Tech Comm Data Eng*. 2001;24(4):35-42.

21. Croft WB, Metzler D, Strohmann T. *Search Engines*. Pearson Education; 2010.

22. Altschul S, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997;25(17):3389-3402. doi:10.1093/nar/25.17.3389.

23. Barrett T, Suzek TO, Troup DB, Wilhite SE, Ngau W-C, Ledoux P, Rudnev D, Lash AE, Fujibuchi W, Edgar R. NCBI GEO: mining millions of expression profiles - database and tools. *Nucleic Acids Res*. 2005;33(Database issue):D562-6. doi:10.1093/nar/gki022.

24. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The protein data bank. *Nucleic Acids Res*. 2000;28(1):235-242. doi:10.1093/nar/28.1.235.

25. Datta R, Joshi D, Li J, Wang JZ. Image retrieval. *ACM Comput Surv*. 2008;40(2):1-60. doi:10.1145/1348246.1348248.

26. Müller H, Michoux N, Bandon D, Geissbuhler A. A review of content-based image retrieval systems in medical applications-clinical benefits and future directions. *Int J Med Inform*. 2004;73(1):1-23. doi:10.1016/j.ijmedinf.2003.11.024.

27. Hu W, Xie N, Li L, Zeng X, Maybank S. A survey on visual content-based video indexing and retrieval. *IEEE Trans Syst Man, Cybern Part C*. 2011;41(6):797-819. doi:10.1109/TSMCC.2011.2109710.

28. Cano P, Batlle E, Kalker T, Haitsma J. A Review of audio fingerprinting. *J VLSI signal Process Syst Signal, Image Video Technol*. 2005;41(3):271-284. doi:10.1007/s11265-005-4151-3.

29. Orio N. Music retrieval: a tutorial and review. *Found Trends Inf Retr*. 2006;1(1):1-90. doi:10.1561/1500000002.

30. Jones CB, Purves RS. Geographical information retrieval. *Int J Geogr Inf Sci*. 2008;22(3):219-228. doi:10.1080/13658810701626343.

31. Manning CD, Raghavan P, Schütze H. *Introduction to Information Retrieval*. Vol 1. Cambridge University Press; 2008.

32. Marcus MP, Marcinkiewicz MA, Santorini B. Building a large annotated corpus of English: the Penn Treebank. *Comput Linguist*. 1993;19(2):313-330.

33. Jiang J, Zhai C. An empirical study of tokenization strategies for biomedical information retrieval. *Inf Retr Boston*. 2007;10(4-5):341-363. doi:10.1007/s10791-007-9027-7.

34. Porter MF. An algorithm for suffix stripping. *Program*. 1980;14(3):130-137. doi:10.1108/eb046814.

35. Taylor RS. Question-negotiation and information seeking in libraries. *Coll*

Res Libr. 1968;29(3):178-194.

36. Belkin NJ. Anomalous states of knowledge as a basis for information retrieval. *Can J Inf Sci*. 1980;5:133-143.

37. Belkin NJ, Oddy RN, Brooks HM. Ask for information retrieval: part I. background and theory. *J Doc*. 1982;38(2):61-71. doi:10.1108/eb026722.

38. Dervin B. From the mind's eye of the user: the sense-making qualitative-quantitative methodology. *Qual Res Inf Manag*. 1992;9:61-84.

39. Heinz S, Zobel J. Efficient single-pass index construction for text databases. *J Am Soc Inf Sci Technol*. 2003;54(8):713-729. doi:10.1002/asi.10268.

40. Witten IH, Moffat A, Bell TC. *Managing Gigabytes: Compressing and Indexing Documents and Images*. (Fox E, ed.). Morgan Kaufmann Publishers; 1999.

41. Hiemstra D. Information retrieval models. In: *Information Retrieval: Searching in the 21st Century*. Wiley London; 2009:1-17.

42. Hiemstra D. *Using Language Models for Information Retrieval*. Taaluitgeverij Neslia Paniculata; 2001.

43. Salton G, Wong A, Yang C-S. A vector space model for automatic indexing. *Commun ACM*. 1975;18(11):613-620.

44. Salton G, Yang C-S. On the specification of team values in automatic indexing. *J Doc*. 1973;29(4):351-372. doi:10.1108/eb026562.

45. Salton G, Yang C-S, Yu CT. A theory of term importance in automatic text analysis. *J Am Soc Inf Sci*. 1975;26(1):33-44. doi:10.1002/asi.4630260106.

46. Robertson SE, Walker S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Dublin, Ireland: Springer-Verlag New York, Inc.; 1994:232-241.

47. Robertson SE, Walker S, Jones S, Hancock-Beaulieu MM, Gatford M. Okapi at TREC-3. In: Harman DK, ed. *Overview of the Third Text REtrieval Conference (TREC-3)*. Gaithersburg, MD: NIST; 1994:109-126.

48. Cooper WS. Some inconsistencies and misnomers in probabilistic information retrieval. In: *Proceedings of the 14th Annual International ACM*

SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM Press; 1991:57-61. doi:10.1145/122860.122866.

49. Robertson SE, Jones KS. Relevance weighting of search terms. *J Am Soc Inf Sci*. 1976;27(3):129-146. doi:10.1002/asi.4630270302.

50. Harter SP. A probabilistic approach to automatic keyword indexing. Part I. On the distribution of specialty words in a technical literature. *J Am Soc Inf Sci*. 1975;26(4):197-206. doi:10.1002/asi.4630260402.

51. Robertson S, Zaragoza H. The probabilistic relevance framework: BM25 and beyond. *Found Trends Inf Retr*. 2010;3(4):333-389. doi:10.1561/1500000019.

52. Lin J, Wilbur WJ. PubMed related articles: a probabilistic topic-based model for content similarity. *BMC Bioinformatics*. 2007;8(1):423. doi:10.1186/1471-2105-8-423.

53. Metzler D, Croft WB. A Markov random field model for term dependencies. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Salvador, Brazil: ACM; 2005:472-479.

54. Bendersky M, Metzler D, Croft WB. Learning concept importance using a weighted dependence model. In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*. New York, New York, USA: ACM; 2010:31-40.

55. Mei T, Rui Y, Li S, Tian Q. Multimedia search reranking: a literature survey. *ACM Comput Surv*. 2014;46(3):38. doi:10.1145/2536798.

56. Philbin J, Chum O, Isard M, Sivic J, Zisserman A. Object retrieval with large vocabularies and fast spatial matching. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, MN, USA: IEEE; 2007:1-8. doi:10.1109/CVPR.2007.383172.

57. Jégou H, Douze M, Schmid C. Improving bag-of-features for large scale image search. *Int J Comput Vis*. 2010;87(3):316-336. doi:10.1007/s11263-009-0285-2.

58. Chum O, Mikulik A, Perdoch M, Matas J. Total recall II: query expansion revisited. In: *2011 IEEE Conference on Computer Vision and Pattern Recognition*. Colorado Springs, CO, USA: IEEE; 2011:889-896.

59. Lee K-S, Park Y-C, Choi K-S. Re-ranking model based on document clusters.

Inf Process Manag. 2001;37(1):1-14. doi:10.1016/S0306-4573(00)00017-0.

60. Hsu WH, Kennedy LS, Chang S-F. Video search reranking via information bottleneck principle. In: *Proceedings of the 14th ACM International Conference on Multimedia*. New York, New York, USA: ACM Press; 2006:35.

61. Jing F, Wang C, Yao Y, Deng K, Zhang L, Ma W-Y. IGroup. In: *Proceedings of the 14th ACM International Conference on Multimedia*. New York, New York, USA: ACM Press; 2006:377.

62. Yu S, Cai D, Wen J-R, Ma W-Y. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In: *Proceedings of the 12th International Conference on World Wide Web*. New York, New York, USA: ACM Press; 2003:11.

63. Yan R, Hauptmann AG, Jin R. Multimedia search with pseudo-relevance feedback. In: *Proceedings of the 2nd International Conference on Image and Video Retrieval*. Springer, Berlin, Heidelberg; 2003:238-247.

64. Page L, Brin S, Motwani R, Winograd T. *The PageRank Citation Ranking: Bringing Order to the Web*.; 1999. http://ilpubs.stanford.edu:8090/422/.

65. Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. *Comput Networks ISDN Syst*. 1998;30(1-7):107-117. doi:10.1016/S0169-7552(98)00110-X.

66. Kurland O, Lee L. PageRank without hyperlinks. *ACM Trans Inf Syst*. 2010;28(4):1-38. doi:10.1145/1852102.1852104.

67. Zitouni H, Sevil S, Ozkan D, Duygulu P. Re-ranking of web image search results using a graph algorithm. In: *19th International Conference on Pattern Recognition*. Tampa, FL, USA: IEEE; 2008:1-4.

68. Jing Y, Baluja S. VisualRank: applying PageRank to large-scale image search. *IEEE Trans Pattern Anal Mach Intell*. 2008;30(11):1877-1890. doi:10.1109/TPAMI.2008.121.

69. Zloof MM. Query-by-example: the invocation and definition of tables and forms. In: *Proceedings of the 1st International cZitounionference on Very Large Data Bases*. New York, New York, USA: ACM Press; 1975:1-24.

70. Bogers T, Bogers T, Bosch A. Authoritative re-ranking in fusing authorship-based subcollection search results. In: *Proceedings of the Sixth BelgianDutch Information Retrieval Workshop*. Enschede: Neslia Paniculata; 2006:49-55.

71. Rui Y, Huang TS, Chang S-F. Image retrieval: current techniques, promising directions, and open issues. *J Vis Commun Image Represent*. 1999;10(1):39-62. doi:10.1006/jvci.1999.0413.

72. Hauptmann AG, Rong Yan, Wei-Hao Lin, Christel MG, Wactlar H. Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news. *IEEE Trans Multimed*. 2007;9(5):958-966. doi:10.1109/TMM.2007.900150.

73. Hauptmann AG, Christel MG, Rong Yan. Video retrieval based on semantic concepts. *Proc IEEE*. 2008;96(4):602-622. doi:10.1109/JPROC.2008.916355.

74. Kennedy LS, Chang S-F. A reranking approach for context-based concept fusion in video indexing and retrieval. In: *Proceedings of the 6th ACM International cZloofonference on Image and Video Retrieval*. New York, New York, USA: ACM Press; 2007:333-340.

75. Li X, Li X, Wang D, Li J, Zhang B. Video search in concept subspace: a text-like paradigm. In: *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*. Amsterdam, The Netherlands: ACM Press; 2007:603-610.

76. Carpineto C, Romano G. A survey of automatic query expansion in information retrieval. *ACM Comput Surv*. 2012;44(1):1-50. doi:10.1145/2071389.2071390.

77. Yang Y, Carbonell JG, Brown RD, Frederking RE. Translingual information retrieval: learning from bilingual corpora. *Artif Intell*. 1998;103(1-2):323-345. doi:10.1016/S0004-3702(98)00063-0.

78. Dwork C, Kumar R, Naor M, Sivakumar D. Rank aggregation methods for the web. In: *Proceedings of the 10th International Conference on World Wide Web*. New York, New York, USA: ACM Press; 2001:613-622.

79. Liu Y-T, Liu T-Y, Qin T, Ma Z-M, Li H. Supervised rank aggregation. In: *Proceedings of the 16th International Conference on World Wide Web*. New York, New York, USA: ACM Press; 2007:481-490.

80. White RW, Richardson M, Bilenko M, Heath AP. Enhancing web search by promoting multiple search engine use. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, New York, USA: ACM Press; 2008:43-50.

81. Liu Y, Mei T, Hua X-S. CrowdReranking. In: *Proceedings of the 32nd*

International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, New York, USA: ACM Press; 2009:500-507.

82. Zha Z-J, Yang L, Mei T, Wang M, Wang Z, Chua T-S, Hua X-S. Visual query suggestion. *ACM Trans Multimed Comput Commun Appl*. 2010;6(3):1-19. doi:10.1145/1823746.1823747.

83. Yamamoto T, Nakamura S, Tanaka K. Rerank-by-example: efficient browsing of web search results. In: *Proceedings of the 18th International Conference on Database and Expert Systems Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007:801-810.

84. Rohini U, Varma V. A novel approach for re-ranking of search results using collaborative filtering. In: *International Conference on Computing: Theory and Applications*. Kolkata, India: IEEE; 2007:491-496.

85. Yong Rui, Huang TS, Ortega M, Mehrotra S. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Trans Circuits Syst Video Technol*. 1998;8(5):644-655. doi:10.1109/76.718510.

86. Salton G, Buckley C. Improving retrieval performance by relevance feedback. *J Am Soc Inf Sci*. 1990;41(4):288-297. doi:10.1002/(SICI)1097-4571(199006)41:4<288::AID-ASI8>3.0.CO;2-H.

87. Li H. A short introduction to learning to rank. *IEICE Trans Inf Syst*. 2011;94(10):1854-1862. doi:10.1587/transinf.E94.D.1854.

88. Li H. Learning to rank for information retrieval and natural language processing, second edition. *Synth Lect Hum Lang Technol*. 2014;7(3):1-121. doi:10.2200/S00607ED2V01Y201410HLT026.

89. Cossock D, Zhang T. Subset ranking using regression. In: *Proceedings of the 19th Annual Conference on Learning Theory*. Pittsburgh, PA: Springer-Verlag Berlin, Heidelberg; 2006:605-619.

90. Cao Z, Qin T, Liu TY, Tsai MF, Li H. Learning to rank. In: *Proceedings of the 24th International Conference on Machine Learning*. Corvalis, Oregon, USA: ACM; 2007:129-136.

91. Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks. In: *Proceedings of the 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Santiago, Chile: ACM; 2015:373-382.

92. Ostell J. The Entrez Search and Retrieval System. In: *The NCBI Handbook [Internet]. 2nd Edition.* Bethesda (MD): National Center for Biotechnology Information (US); 2014.

93. Squizzato S, Park YM, Buso N, Gur T, Cowley A, Li W, Uludag M, Pundir S, Cham JA, McWilliam H, Lopez R. The EBI search engine: providing search and retrieval functionality for biological data from EMBL-EBI. *Nucleic Acids Res*. 2015;43(W1):W585-W588. doi:10.1093/nar/gkv316.

94. Sayers EW, Barrett T, Benson DA, Bolton E, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Federhen S, Feolo M, Fingerman IM, Geer LY, Helmberg W, Kapustin Y, Landsman D, Lipman DJ, Lu Z, Madden TL, Madej T, Maglott DR, Marchler-Bauer A, Miller V, Mizrachi I, Ostell J, Panchenko A, Phan L, Pruitt KD, Schuler GD, Sequeira E, Sherry ST, Shumway M, Sirotkin K, Slotta D, Souvorov A, Starchenko G, Tatusova TA, Wagner L, Wang Y, Wilbur WJ, Yaschenko E, Ye J. Database resources of the national center for biotechnology information. *Nucleic Acids Res*. 2011;39(suppl 1):D38-D51. doi:10.1093/nar/gkq1172.

95. Collins FS, Tabak LA. Policy: NIH plans to enhance reproducibility. *Nature*. 2014;505(7485):612-613. doi:10.1038/505612a.

96. Ohno-Machado L, Sansone S-A, Alter G, Fore I, Grethe J, Xu H, Gonzalez-Beltran A, Rocca-Serra P, Gururaj AE, Bell E, Soysal E, Zong N, Kim H. Finding useful data across multiple biomedical data repositories using DataMed. *Nat Genet*. 2017;49:816–819. doi:10.1038/ng.3864.

97. Sansone S-A, Gonzalez-Beltran A, Rocca-Serra P, Alter G, Grethe JS, Hua X, Ian F, Lyle, J, Gururaj AE, Chen X, Kim H, Zong N, Li Y, Liu R, Ozyurt B, Ohno-Machado L, the bioCADDIE Working Group Members. DATS: the data tag suite to enable discoverability of datasets. *Sci Data*. 2017;4:170059. doi:10.1038/sdata.2017.59.

98. Yilmaz E, Aslam JA. Inferred AP : estimating average precision with incomplete judgments. In: *Fifteenth ACM International Conference on Information and Knowledge Management*. Arlington, Virginia, USA: ACM Press; 2006:102-111.

99. Järvelin K, Kekäläinen J. Cumulated gain-based evaluation of IR techniques. *ACM Trans Inf Syst*. 2002;20(4):422-446. doi:10.1145/582415.582418.

100. Wang Y, Wang L, Li Y, He D, Liu T-Y. A theoretical analysis of NDCG type ranking measures. In: *Proceedings of the 26th Annual Conference on Learning Theory*. Princeton, NJ, USA: JMLR; 2013:25-54.

101. Yilmaz E, Kanoulas E, Aslam JA. A simple and efficient sampling method for estimating AP and NDCG. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Singapore, Singapore; 2008:603-610.

102. Baca M. *Introduction to Metadata*. 2nd ed. Getty Research Institute; 2016.

103. Haslhofer B, Klas W. A survey of techniques for achieving metadata interoperability. *ACM Comput Surv*. 2010;42(2):1-37. doi:10.1145/1667062.1667064.

104. Rahm E, Bernstein PA. A survey of approaches to automatic schema matching. *VLDB J*. 2001;10(4):334-350. doi:10.1007/s007780100057.

105. Bechhofer S. OWL: Web Ontology Language. In: Liu L, Özsu MT, eds. *Encyclopedia of Database Systems*. Boston, MA: Springer US; 2009:2008-2009.

106. McBride B. The Resource Description Framework (RDF) and its vocabulary description language RDFS. In: Staab S, Studer R, eds. *International Handbooks on Information Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004:51-65.

107. Bodenreider O, Stevens R. Bio-ontologies: current trends and future directions. *Brief Bioinform*. 2006;7(3):256-274. doi:10.1093/bib/bbl027.

108. Cimino JJ, Zhu X. The practical impact of ontologies on biomedical informatics. *Yearb Med Inform*. 2006:124-135.

109. Harpring P. *Introduction to Controlled Vocabularies : Terminology for Art, Architecture, and Other Cultural Works*. Getty Research Institute; 2010.

110. Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res*. 2004;32(90001):267D-270. doi:10.1093/nar/gkh061.

111. Lipscomb CE. Medical subject headings (MeSH). *Bull Med Libr Assoc*. 2000;88(3):265.

112. World Health Organization. *International Classification of Diseases : [9th] Ninth Revision, Basic Tabulation List with Alphabetic Index*. Geneva : World Health Organization; 1978.

113. Stearns MQ, Price C, Spackman KA, Wang AY. SNOMED clinical terms: overview of the development process and project status. *Proc AMIA Symp*.

2001:662-666.

114. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. Gene ontology: tool for the unification of biology. *Nat Genet*. 2000;25(1):25-29. doi:10.1038/75556.

115. Jimeno-Yepes A, Aronson AR. Knowledge-based biomedical word sense disambiguation: comparison of approaches. *BMC Bioinformatics*. 2010;11(1):569. doi:10.1186/1471-2105-11-569.

116. Belkin NJ, Croft WB. Information filtering and information retrieval: two sides of the same coin? *Commun ACM*. 1992;35(12):29-38. doi:10.1145/138859.138861.

117. Schwenk H. Continuous space language models. *Comput Speech Lang*. 2007;21(3):492-518. doi:10.1016/j.csl.2006.09.003.

118. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. *J Am Soc Inf Sci*. 1990;41(6):391.

119. Dumais ST. Latent semantic indexing (LSI) and TREC-2. In: *The Second Text REtrieval Conference (TREC 2)*. NIST; 1994:105-115.

120. Dumais ST. Latent semantic indexing (LSI): TREC-3 report. In: *Overview of the Third Text REtrieval Conference (TREC-3)*. NIST; 1995:219-230.

121. Blei DM, Ng AY, Jordan MI. Latent Dirichlet Allocation. *J Mach Learn Res*. 2003;3(Jan):993-1022.

122. Wei X, Croft WB. LDA-based document models for ad-hoc retrieval. In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Seattle, Washington, USA: ACM; 2006:178-185. doi:10.1145/1148170.1148204.

123. MacMullen WJ, Denn SO. Information problems in molecular biology and bioinformatics. *J Am Soc Inf Sci Technol*. 2005;56(5):447-456. doi:10.1002/asi.20134.

124. Geer RC. Broad issues to consider for library involvement in bioinformatics. *J Med Libr Assoc*. 2006;94(3):286-298, E152-5.

125. Kumpulainen S, Järvelin K. Information interaction in molecular medicine. In: *Proceedings of the Third Symposium on Information Interaction in Context*.

New Brunswick, New Jersey, USA: ACM; 2010:95-104.

126. Kim J -j., Rebholz-Schuhmann D. Categorization of services for seeking information in biomedical literature: a typology for improvement of practice. *Brief Bioinform*. 2008;9(6):452-465. doi:10.1093/bib/bbn032.

127. Jay C, Harper S, Dunlop I, Smith S, Sufi S, Goble C, Buchan I. Natural language search interfaces: health data needs single-field variable search. *J Med Internet esearch*. 2016;18(1):e13. doi:10.2196/jmir.4912.

128. Sanderson M, Croft WB. The history of information retrieval research. *Proc IEEE*. 2012;100(Special Centennial Issue):1444-1451. doi:10.1109/JPROC.2012.2189916.

129. Smyth B, Balfe E, Freyne J, Briggs P, Coyle M, Boydell O. Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Model User-adapt Interact*. 2004;14(5):383-423. doi:10.1007/s11257-004-5270-4.

130. Roberts K, Gururaj A, Chen X, Pournejati S, Hersh WR, Demner-Fushman D, Ohno-Machado L, Cohen T, Xu H. Information retrieval for biomedical datasets: the 2016 bioCADDIE dataset retrieval challenge. *Database (Oxford)*. 2017.

131. Metzler D, Croft WB. Linear feature-based models for information retrieval. *Inf Retr Boston*. 2007;10(3):257-274. doi:10.1007/s10791-006-9019-z.

132. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: *AMIA Symposium Proceedings*. ; 2001:17–21.

133. Cohen T, Roberts K, Gururaj A, Chen X, Pournejati S, Hersh WR, Demner-Fushman D, Ohno-Machado L, Xu H. A publicly available benchmark for biomedical dataset retrieval: the reference standard for the 2016 bioCADDIE dataset retrieval challenge. *Database (Oxford)*. 2017.

134. BioCADDIE 2016 dataset retrieval challenge. https://biocaddie.org/biocaddie-2016-dataset-retrieval-challenge. Accessed May 26, 2017.

135. Ohno-Machado L, Bafna V, Boxwala AA, Chapman BE, Chapman, Wendy W. Chaudhuri K, Day ME, Farcas C, Heintzman, Nathaniel D. Jiang X, Kim H, Kim J, Matheny, Michael E. Resnic FS, Vinterbo SA. iDASH: integrating data for analysis, anonymization, and sharing. *J Am Med Informatics Assoc*. 2012;19(2):196-201. doi:10.1136/amiajnl-2011-000538.

136. Huang M, Névéol A, Lu Z. Recommending MeSH terms for annotating biomedical articles. *J Am Med Inform Assoc*. 2011;18(5):660-667. doi:10.1136/amiajnl-2010-000055.

137. Aronson AR, Mork JG, Gay CW, Humphrey SM, Rogers WJ. The NLM indexing initiative's medical text indexer. *Medinfo*. 2004;11(Pt 1):268-272. doi:10.3233/978-1-60750-949-3-268.

138. Mork JG, Demner-Fushman D, Schmidt SC, Aronson AR. Recent enhancements to the NLM Medical Text Indexer. In: Cappellato L, Ferro N, Halvey M, Kraaij W, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2014*. Sheffield, UK: CEUR Workshop Proceedings; 2014:1328-1336.

139. Sohn S, Kim W, Comeau DC, Wilbur WJ. Optimal training sets for Bayesian prediction of MeSH assignment. *J Am Med Inform Assoc*. 2008;15(4):546-553. doi:10.1197/jamia.M2431.

140. Ruch P. Automatic assignment of biomedical categories: toward a generic approach. *Bioinformatics*. 2006;22(6):658-664. doi:10.1093/bioinformatics/bti783.

141. Trieschnigg D, Pezik P, Lee V, de Jong F, Kraaij W, Rebholz-Schuhmann D. MeSH Up: effective MeSH text classification for improved document retrieval. *Bioinformatics*. 2009;25(11):1412-1418. doi:10.1093/bioinformatics/btp249.

142. Jimeno-Yepes A, MacKinlay A, Bedo J, Garnavi R, Chen Q. Deep belief networks and biomedical text categorisation. In: *Proceedings of the Australasian Language Technology Workshop 2014*. Melbourne, Australia: Association for Computational Linguistics; 2014:123-127.

143. Jimeno-Yepes A, Plaza L, Mork JG, Aronson AR, Díaz A. MeSH indexing based on automatically generated summaries. *BMC Bioinformatics*. 2013;14(1):208.

144. Jimeno-Yepes A, Mork J, Demner-Fushman D, Aronson AR. Automatic algorithm selection for MeSH Heading indexing based on meta-learning. In: *The Fourth International Symposium on Languages in Biology and Medicine, a Pre-Conference Workshop of the 25th Pacific Asia Conference on Language, Information and Computation (PACLIC 25)*. Singapore; 2011.

145. Jimeno-Yepes A, Mork JG, Demner-Fushman D, Aronson AR. A one-size-fits-all indexing method does not exist: automatic selection based on meta-learning. *J Comput Sci Eng*. 2012;6(2):151-160.

doi:10.5626/JCSE.2012.6.2.151.

146. Mork JG, Jimeno-Yepes AJ, Aronson AR. The NLM Medical Text Indexer System for indexing biomedical literature. In: *Proceedings of Conference and Labs of the Evaluation Forum 2013*. Valencia, Spain; 2013.

147. Mao Y, Lu Z. *NCBI at the 2013 BioASQ Challenge Task: Learning to Rank for Automatic MeSH Indexing*. Valencia, Spain; 2013.

148. Yang Y, Chute CG. A linear least squares fit mapping method for information retrieval from natural language texts. In: *Proceedings of the 14th Conference on Computational Linguistics*. Nantes, France: Association for Computational Linguistics; 1992:447-453.

149. Blair DC, Maron ME. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Commun ACM*. 1985;28(3):289-299.

150. Aljaber B, Martinez D, Stokes N, Bailey J. Improving MeSH classification of biomedical articles using citation contexts. *J Biomed Inform*. 2011;44(5):881-896. doi:10.1016/j.jbi.2011.05.007.

151. Choi S, Choi J. Classification and retrieval of biomedical literatures: SNUMedinfo at CLEF qa track BioASQ 2014. In: Cappellato L, Ferro N, Halvey M, Kraaij W, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2014*. Sheffield, UK: CEUR Workshop Proceedings; 2014:1283-1295.

152. Tang L, Rajan S, Narayanan VK. Large scale multi-label classification via metalabeler. In: *Proceedings of the 18th International Conference on World Wide Web*. Madrid, Spain; 2009:211-220.

153. Mao Y, Wei CH, Lu Z. NCBI at the 2014 BioASQ challenge task: large-scale biomedical semantic indexing and question answering. In: Cappellato L, Ferro N, Halvey M, Kraaij W, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2014*. Sheffield, UK: CEUR Workshop Proceedings; 2014:1319-1327.

154. Tsoumakas G, Laliotis M, Markantonatos N, Vlahavas IP. Large-scale semantic indexing of biomedical publications. In: Ngomo A-CN, Paliouras G, eds. *Proceedings of the First Workshop on Bio-Medical Semantic Indexing and Question Answering, a Post-Conference Workshop of Conference and Labs of the Evaluation Forum 2013 (CLEF 2013)*. Valencia, Spain: CEUR Workshop Proceedings; 2013.

155. Papanikolaou Y, Dimitriadis D, Tsoumakas G, Laliotis M, Markantonatos N,

Vlahavas IP. Ensemble approaches for large-scale multi-label classification and question answering in biomedicine. In: Cappellato L, Ferro N, Halvey M, Kraaij W, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2014*. Sheffield, UK: CEUR Workshop Proceedings; 2014:1348-1360.

156. Papanikolaou Y, Tsoumakas G, Laliotis M, Markantonatos N, Vlahavas I. AUTH-Atypon at BioASQ 3: large-scale semantic indexing in biomedicine. In: Cappellato L, Ferro N, Jones GJF, Juan ES, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2015*. Toulouse, France: CEUR Workshop Proceedings; 2015.

157. Ribadas FJ, de Campos LM, Darriba VM, Romero AE. CoLe and UTAI participation at the 2014 BioASQ semantic indexing challenge. In: Cappellato L, Ferro N, Halvey M, Kraaij W, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2014*. Sheffield, UK: CEUR Workshop Proceedings; 2014:1361-1374.

158. Rios A, Kavuluru R. Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In: *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*. ; 2015:258-267.

159. Xu H, Dong M, Zhu D, Kotov A, Carcone AI, Naar-King S. Text classification with topic-based word embedding and convolutional neural networks. In: *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. Seattle, WA, USA: ACM; 2016:88-97.

160. Peng S, You R, Wang H, Zhai C, Mamitsuka H, Zhu S. DeepMeSH: deep semantic representation for improving large-scale MeSH indexing. *Bioinformatics*. 2016;32(12):i70-i79. doi:10.1093/bioinformatics/btw294.

161. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: *Neural Information Processing Systems 2013*. Lake Tahoe, CA, USA: Curran Associates; 2013:3111-3119.

162. Le Q V., Mikolov T. Distributed representations of sentences and documents. In: *Proceedings of the 31st International Conference on Machine Learning*. Beijing, China: JMLR; 2014:1188-1196.

163. Kamineni A, Fatma N, Das A, Shrivastava M, Chinnakotla M. IIITH at BioASQ Challenge 2015 Task 3a: extreme classification of PubMed articles using MeSH labels. In: Cappellato L, Ferro N, Jones GJF, Juan ES, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2015*.

Toulouse, France: CEUR Workshop Proceedings; 2015.

164. Ribadas FJ, Campos LM de, Darriba VM, Romero AE. CoLe and UTAI at BioASQ 2015: experiments with similarity based descriptor assignment. In: Cappellato L, Ferro N, Jones GJF, Juan ES, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2015*. Toulouse, France: CEUR Workshop Proceedings; 2015.

165. Liu K, Peng S, Wu J, Zhai C, Mamitsuka H, Zhu S. MeSHLabeler: improving the accuracy of large-scale MeSH indexing by integrating diverse evidence. *Bioinformatics*. 2015;31(12):i339-47. doi:10.1093/bioinformatics/btv237.

166. Yu Z, Bernstam E, Cohen T, Wallace BC, Johnson TR. Improving the utility of MeSH terms using the TopicalMeSH representation. *J Biomed Inform*. 2016;61(C):77-86. doi:10.1016/j.jbi.2016.03.013.

167. Jimeno-Yepes A, Mork JG, Demner-Fushman D, Aronson AR. Comparison and combination of several MeSH indexing approaches. In: *AMIA Annual Symposium Proceedings*. Vol 2013. WashingtonDC: AMIA; 2013:709.

168. Funk ME, Reid CA. Indexing consistency in MEDLINE. *Bull Med Libr Assoc*. 1983;71(2):176-183.

169. Balikas G, Kosmopoulos A, Krithara A, Paliouras G, Kakadiaris I. Results of the BioASQ tasks of the question answering lab at CLEF 2015. In: Cappellato L, Ferro N, Jones GJF, Juan ES, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2015Tsoumakas*. Toulouse, France: CEUR Workshop Proceedings; 2015.

170. Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. *J Am Med Informatics Assoc*. 2010;17(3):229-236. doi:10.1136/jamia.2009.002733.

171. Aronson AR, Bodenreider O, Chang HF, Humphrey SM, Mork JG, Nelson SJ, Rindflesch TC, Wilbur WJ. The NLM Indexing Initiative. In: *AMIA Symposium Proceedings*. Los Angeles, CA: AMIA; 2000:17-21.

172. Mork J, Aronson A, Demner-Fushman D. 12 years on – is the NLM medical text indexer still useful and relevant? *J Biomed Semantics*. 2017;8(8). doi:10.1186/s13326-017-0113-5.

173. Liu K, Wu J, Peng S, Zhai C, Zhu S. The Fudan-UIUC participation in the BioASQ challenge task 2a: The antinomyra system. In: Cappellato L, Ferro N, Halvey M, Kraaij W, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2014*. Sheffield, UK: CEUR Workshop Proceedings;

2014:1311-1318.

174. Burges C, Shaked T, Renshaw E, Lazier A, Deeds M, Hamilton N, Hullender G. Learning to rank using gradient descent. In: *Proceedings of the 22nd International Conference on Machine Learning*. Bonn, Germany: ACM; 2005:89-96.

175. Xu J, Li H. AdaRank. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* . New York, New York, USA: ACM Press; 2007:391.

176. Burges CJC. *From Ranknet to Lambdarank to Lambdamart: An Overview*.; 2010. http://research.microsoft.com/en-us/um/people/cburges/tech%5C_reports/MSR-TR-2010-82.pdf.

177. Balikas G, Partalas I, Ngomo A-CN, Krithara A, Paliouras G. Results of the BioASQ track of the question answering Lab at CLEF 2014. In: Cappellato L, Ferro N, Halvey M, Kraaij W, eds. *Working Notes for Conference and Labs of the Evaluation Forum 2014*. Sheffield, UK: CEUR Workshop Proceedings; 2014:1181–1193.

178. Kim Y. Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar: Association for Computational Linguistics; 2014:1746-1751.

179. Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland: Association for Computational Linguistics; 2014:655–665.

180. Hinton GE, McClelland JL, Rumelhart DE. Distributed representations. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*. MIT Press; 1986:77-109.

181. Theano Development Team. Theano: a Python framework for fast computation of mathematical expressions. *arXiv e-prints*. 2016;abs/1605.0. http://arxiv.org/abs/1605.02688.

182. PubMed Help. http://www.ncbi.nlm.nih.gov/books/NBK3827/#pubmedhelp.Computation_of_Similar_Articl.

183. Fontaine J-F, Barbosa-Silva A, Schaefer M, Huska MR, Muro EM, Andrade-Navarro MA. MedlineRanker: flexible ranking of biomedical literature.

Nucleic Acids Res. 2009;37(suppl 2):W141-W146. doi:10.1093/nar/gkp353.

184. Poulter GL, Rubin DL, Altman RB, Seoighe C. MScanner: a classifier for retrieving Medline citations. *BMC Bioinformatics*. 2008;9(1):108. doi:10.1186/1471-2105-9-108.

185. Errami M, Wren JD, Hicks JM, Garner HR. eTBLAST: a web server to identify expert reviewers, appropriate journals and similar publications. *Nucleic Acids Res*. 2007;35(suppl 2):W12-W15. doi:10.1093/nar/gkm221.

186. Boyack KW, Newman D, Duhon RJ, Klavans R, Patek M, Biberstine JR, Schijvenaars B, Skupin A, Ma N, Börner K. Clustering more than two million biomedical publications: comparing the accuracies of nine text-based similarity approaches. *PLoS One*. 2011;6(3):e18029. doi:10.1371/journal.pone.0018029.

187. MEDLINE Fact Sheet. https://www.nlm.nih.gov/pubs/factsheets/medline.html.

188. Hsu C-N, Chang Y-M, Kuo C-J, Lin Y-S, Huang H-S, Chung I-F. Integrating high dimensional bi-directional parsing models for gene mention tagging. *Bioinformatics*. 2008;24(13):i286-94. doi:10.1093/bioinformatics/btn183.

189. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In: C. J. C. Burges, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, eds. *Advances in Neural Information Processing Systems 26*. Lake Tahoe, CA, USA: Curran Associates; 2013:2787-2795.