**DataRank: A Framework for Ranking Biomedical Datasets**

**Arya Iranmehr1, Xiaoqian Jiang2, Hannah Chen3**

1Department of Electrical and Computer Engineering,UC San Diego,   
2Department of Biomedical Informatics, UC San Diego  
3Department of Computer Science, UC San Diego

**Abstract**

*Since the advent of high-performance technologies for generating, storing and precessing data, research in biomedical sciences is increasingly moved toward manipulating and analyzing biomedical datasets, which even sometimes it is referred to “data-driven research”. Unfortunately, the vast majority of biomedical datasets are underutilized after their initial publications due to the increased rate of dataset generation and difficulty in searching for relevant datasets. In this paper, we develop a framework for ranking biomedical datasets by incorporating three different criteria, query-relevance, research-importance and user-preference into our model. We implement query-relevance by multi-label classification approach using binary relevance scheme. We also incorporate research-importance of datasets into ranking by taking into account of network of citations. Preference ranking is implemented in an online setting which DataRank corrects the ranking results using user feedback. To evaluate DataRank's effectiveness for query searches, we compare Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) measures of DataRank search results with those of Gene Expression Omnibus (GEO) Search results on five different validation set of queries and show that DataRank performs better. Finally, we assess the effect of incorporating user feedback into ranking by measuring Regret for five different subjects and show that the regret is non-increasing,*

**1. Introduction**

New advancements in technologies have resulted in the proliferation of data production at an unprecedented scale, leading scientists to the problem that the capacity to organize, let alone find or understand the data, has lagged behind our ability to generate data. Back in 2006, shortly after the popularization of next-generation sequencing, it was found that data volumes were doubling every year and becoming increasingly complex and correlated over many dimensions [1]. Using nucleotide sequencing as an example, it was found in 2009 that submission rates to a global database were growing at 200% per annum; conservation of this data is guarded by three global repositories which exchange new data on a daily basis [2].

Alongside this proliferation, where invested financial resources may be provided by the public sector, follows an expectation of freely accessible data for global biomedical research. As such, much of the scientific community has been concerned with the scalability and collaborative issues associated with data management like necessary standardization and infrastructure needed to support useful processing and work-flow methods. Increased commitments to data sharing is reinforced by efforts dedicated to building databases, repositories and biobanks [3]. Multidisciplinary databases also led to the collection and combined analysis of data found from multiple repositories, which eventually may be deposited elsewhere for shared use. In a generation of data-intensive and data-driven computing, scientists are engaging in records as a corpus of text and collections of interlinked data resources that may identify papers of interest, suggest hypotheses to explore, and even the production of new data [4]. Existing entities have been addressing issues to facilitate data use: the European Bioinformatics Institute has developed open standards and search systems indexing data, and the National Center for Biotechnology Information (NCBI) has invested greatly in the PubMed [5] search engine for accessing biomedical literature citations of the MEDLINE database. Despite these efforts, the growing size and number of partially coordinated and incompatible dataset formats makes data discovery and integration a notable challenge.

In the past, people may have cited dataset identification numbers explicitly, cited the publication connecting to the dataset’s first use, merely mentioned it within a paper, or not cited the dataset at all. Authors and publishers are now encouraged to include data availability information in publications. While much work has focused on PubMed article querying, there still lacks of a unified, systematic way to retrieve datasets.

A direct search requiring background knowledge on the part of the enquirer can be done by searching within a repository for the data or authors directly connected to the dataset. The National Institute of Health (NIH) currently manages 65 biomedical data sharing repositories [6]. In the best case, NIH may have invested resources to provide user-friendly mechanisms for query and analysis, such as with GEO. GEO records can be accessed directly using an accession number or queries can be made to study-level or gene-level databases. Queries can be “effectively performed by simply entering appropriate keywords and phrases into the search box. However, given the large volumes of data stored in these databases, it is often useful to perform more refined queries” [7]. In other words, efficiently querying requires already knowing what a dataset encompasses.

Citation analysis and bibliometrics establish precedence of ideas and hint at the quality or significance of a scientific work [27,28]. We exploit articles bibliometrics information to 1) infer features (information) about dataset using related articles 2) learn about datasets popularity. The end goal of DataRank is developing a biomedical dataset search engine which takes into account of research-context of the dataset that is actually being reflected by the articles, as well as the significance of dataset in the biomedical research. The model is adaptive in the sense that the dataset context evolves with the active biomedical research publications.

This paper provides a framework for for searching biomedical datasets using the relationship between datasets and articles as well as relation between articles themselves. More precisely, we use a bipartite network between datasets and papers to augment datasets with the information of the related papers such as Medical Subject Heading (MeSH), topic, etc. and then apply machine learning techniques to provide an ranking model which accounts for different aspect of datasets such as *query-relevance*, *research-importance* and *user-preference*.

The organization of this paper is as follows. In section 2, we review related and in Section 3, we describe DataRank model and its extensions. In section 4, we evaluate our approach on five different validation set of queues and compare the results to existing GEO search results. Finally, in Section 5, we conclude and discuss possible extensions of interest.

**2. Background**

Ranking is the issue problem in many information retrieval, machine learning, natural language processing, data mining and advertisement problems. Given a query *q*, the task of ranking involves computing the ranking score for all the items in the collection . The function *f* is called the *ranking model* which takes two arguments, the searching query and an item from the collection, and computes the relevance of the examining item to the query based on the implementing algorithm.

**Query-Relevance Ranking**. In general, relevance ranking models can be divided into two major groups: *traditional* models and *learning-to-rank* models. Traditional models compute the relevance of each item directly from their argument without a training process. For example of in document retrieval, popular methods such as BM25 [12], LMIR [13], VSM [14,15], BM25F [16] compute their ranking score for each document only base on Term-Frequency (TF) for each term of the query in the examining document and Inverse Document Frequency (IDF) for the query terms. On the other hand learning-to-rank models operate by learning *f* using a training data, i.e., a set of queries with known relevant items. For example, RankSVM [11, 17], RankBoost [18,21], RankNet [19,20] GBRank [26] adapt Support Vector Machines, AdaBoost, Neural Network and Regression models for learning-to-rank, respectively. Although learning-to-rank approach generally outperform traditional methods [22], they are only feasible when a representative training corpus of queries with corresponding relevant target available. Collecting training data however is a costly and time consuming task which can be done either by human judgment labeling a set of queries using crowd-sourcing [23,24] or existing real-world query logs [25], i.e. click-through data. In the context of biomedical informatics there has been a lot of efforts to improve ranking. For example ATM [9] is proposed to map the free-text query to a set of MeSH terms. In [8] authors modeled relevance ranking using TF-IDF features for PubMed documents.

**Item-Importance Ranking**. PubFocus [27] takes into account of item popularity, i.e., publication quality, by incorporating bibliometrics. Similarly, [33,34] prioritized important articles in the search results by PageRank [29] algorithm on the citation network of articles. In addition, CoRank [31] and MultiRank [32] introduces the use of collaboration networks between authors jointly with citation networks to estimate popularity of items in the corpus, i.e. importance ranking of items without any query.

**User-Preference Ranking.** In an effort to incorporate user feedback into ranking, RefMed [10] uses RankSVM to learn a model for each query on PubMed articles. Unfortunately, RankSVM only works when enough labeled training samples (user feedback) are provided to the learning algorithm, which is very unlikely in real world situations, to expect a user to rate many the search results. Other authors [30,35] has also exploited online learning techniques to improve the ranking results for the user

In the next section we first describe our indexing procedure, and then we explain different aspects of DataRank algorithm including, relevance ranking, importance ranking, preference ranking.

**3. Methods**

Before outlining DataRank algorithm we need to first specify the training corpus that we used in this paper. Creating a training corpus amounts to indexing datasets and articles, collect a set of queries and their associated relevant targets.

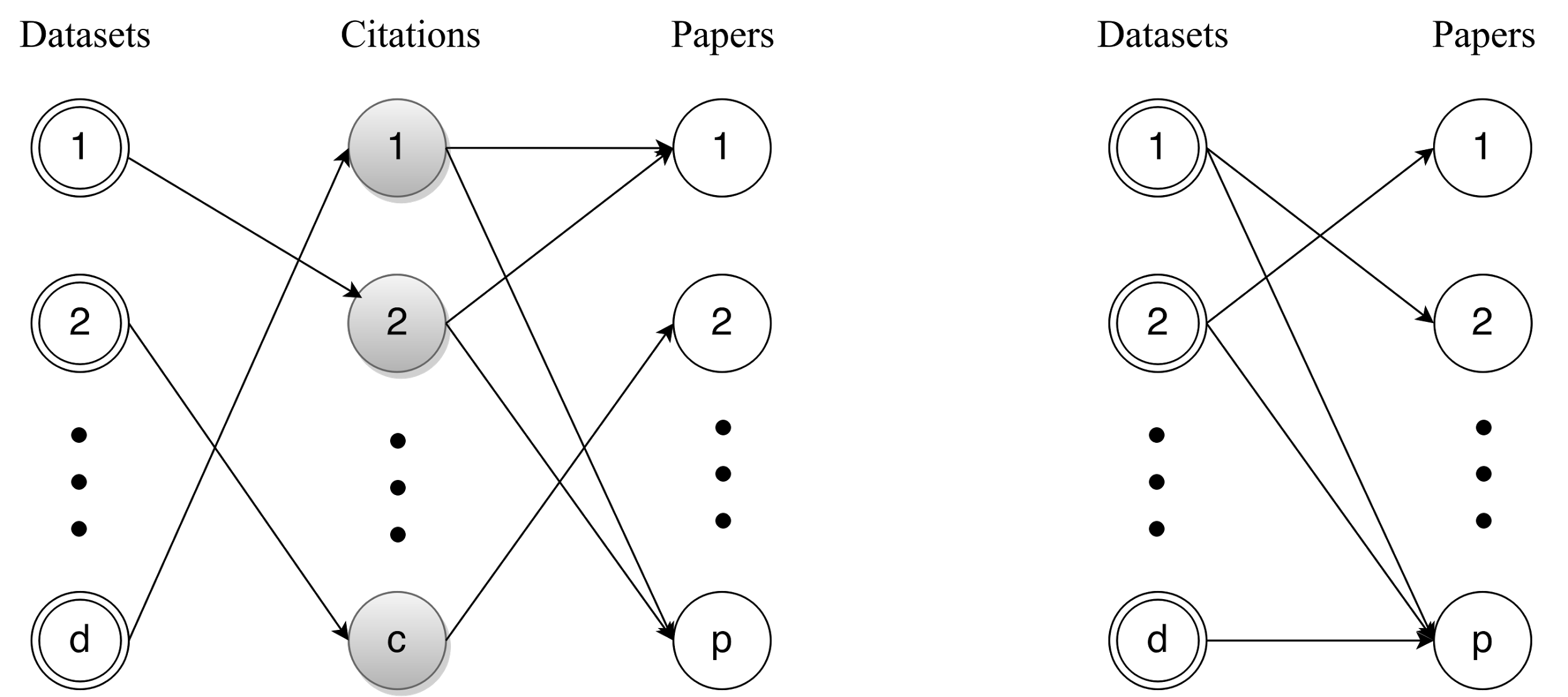
**3.1 Collection Training Corpus**

The first step to create any search engine is to index items of interest. In case of DataRank we need to index the biomedical datasets that are used in the scientific publications and create a bipartite graph between datasets and papers. The many different use cases and dataset origins proves to be a complication for dataset indexing. For example, a repository may have strict guidelines to only accept primary datasets, not reanalysis, meta analysis or comparisons. Datasets may also be collected over a range of time, published at various times, or be a subset or derivative of larger, dynamic databases. There are three ways to perform indexing: 1) searching full text documents and matching regular expression rules for accession numbers of each repository. To our experience this method is restrictive not only because of many different ways that authors may cite a dataset but also small fraction of PubMed articles are publicly available through PubMed Central webserver for mass download. 2) Finding particular datasets can also be done via MEDLINE citations via the secondary source identifier (SI). Beginning in 2005, many data types discussed within MEDLINE articles, such as sequencing data, gene expression data, and clinical trial data, could be included in the SI element. Unfortunately, this MEDLINE meta-data does not consistently reflect paper-dataset relationship. For example for all the 58,950 GEO repository datasets[[1]](#footnote-0) it can only find 14,700 which is much less than number of datasets. 3) We can use the each NIH Data repository to find all the datasets and the corresponding original citation and find all the papers that are cited the original citations. After examining all three alternative, we find out that the third approach give a larger set of dataset-papers.

For this paper, we focus only on the popular GEO repository but the same method can be applied to all the other 64 NIH data repositories to create an integrated searching system. After removing datasets without any citations in GEO we ended up with 35,497 datasets connecting to 26,685 original citations in a many-to-one relation, i.e. each dataset has exactly one original citation but each citation can be referred by multiple datasets. Also, to make computations tractable for experimental study, we omitted the original citations (and the corresponding datasets) which has been cited 60 times or less in the Web of Science database, which left us with 3,700 datasets, 2,549 original papers. We further, extended this network to find all the publications that cited any of 2,549 original citations of the datasets using Web of Science database which resulted in finding 216,350 papers that cited original dataset citations 367,623 times in total, see Figure 1 left for illustration. After constructing dataset-citation-paper network we removed original citations and connected papers directly to the related datasets.

Having datasets and the related papers index, we can now create training corpus. We consider each paper as a query and all the related dataset as the relevant datasets for that query. To keep computations simple, instead of full article, abstract or title, we only consider set of MeSH terms associated with each paper as the search query for that paper. For instance, a query-target pair could be the article with PMID 25525874:  
**Query**={ Animals, Apoptosis, genetics, metabolism, classification, metabolism, Crosses, Genetic DNA, Mitochondrial, Female, Hematopoietic Stem Cells, Interferon Type I, Interferon Type I, Membrane Proteins, Mice, Inbred C57BL, Signal Transduction}  
**Targets**={GSE57934, GSE59972}

We can use this corpus for training and validation of relevance ranking and importance ranking tasks. In addition, we can run these queries on GEO search engine provide a baseline performance for DataRank. In the next parts we outline DataRank methods for ranking that operate on this corpus.



**Figure 1.** Left) Illustration of network between datasets, original citations of dataset (citations in the figure) and papers that used datasets (papers) in the figure. After cleaning and ignoring datasets with low number of citations we reached to a network with d=3700, c=2549 and p=216350. The number of edge in the left network is 3700 and in the right network is 367,623. Right) the same network after removing original citations and connecting papers to datasets.

**3.2 Relevance Ranking**

Since the our training corpus does not contain all the rankings of all items for each query but only contain relevant items for each query, we model the problem as a *multi-label classification* problem [36,37]. As the name suggests, Multi-label classification algorithms applies to supervised learning tasks in which each sample corresponds to one or more targets (labels). As shown in Figure 1 right, each query is connected to one or more datasets, and the connected datasets are equally important, i.e., for each paper the connected datasets are equally relevant or disconnected datasets are equally irrelevant. Thus, we can cast the ranking problem into multi-label classification we can consider each query as a sample and its related datasets as labels.

To convert queries into features for learning problem, we need to transform the set of MeSH terms into a fixed length feature vector. We used the widely used Bag-of-Word model to represent set of mesh terms, i.e., represent each query by just number of occurrences of each MeSH term. Since there are 27,455 possible MeSH terms, the feature vector is a 27,455 dimensional sparse binary vector, which can be efficiently represented by sparse feature vectors. For example, for the following query we can write feature vector which can be efficiently represented by a sparse feature vector:

**Query** = {genetics, DNA, Female, }

**Feature** = [0, . . . , 0, 1, 0, . . . , 1, 0, . . . , 1, 0, . . . , 0]**T**

**Sparse Feature** = {1023:1, 9861:1, 22397:1}

A prevalent paradigm for multi-label classification is called *problem transformation*, in which the multi-label classification problem is transformed into a set of classical single-label classification problems [36]. In particular, binary relevance methods [37] transforms the multi-label classification problem with *L* labels to *L* different binary classification problems and takes the one-versus-the-rest approach in the training process. In the test phase, binary relevance evaluates all the *L* decision function values for each test sample to predict a set of labels for that sample.

In case of DataRank, the training phase is identical to the binary relevance learning procedure, but in the test phase we are not interested in predicting targets for each test case, instead the raw real-valued decision valued are to be used in the relevance-ranking. More precisely, for each test query DataRank sorts all the decision values and corresponding labels (datasets). It should be noted that the training process is time consuming and is performed once and offline. For each test case DataRank needs to compute *L*=3700 decision function, which 1) can be done in parallel 2) in case of linear classifiers, evaluating each decision function amounts to computing a dot product between the feature vector *x* and a weight vector *w*:

which scalar *b* is offset in a linear transformation. Note that the 27,455 dimensional feature vector is extremely sparse and only a few components are non-zero, so the relevance-ranking can rank new test queries in real-time without latency.

For binary classification model we use standard soft-margin Linear Support Vector Machines (SVM) [41] which is efficiently implemented in the sklearn package [38]. Linear SVM can effectively learn all the weight vectors and offset terms for all the *L* decision functions. In this setting, each component of the weight vectors is the weight determining how much each MeSH term is predictive of each dataset. In other words, in this model each MeSH term in the query can be predictor of different datasets with different weights. Moreover, these weights are learned on a separate training corpus and can be evaluated on a held-out test corpus. In Section 4 we perform these experiments and compare it to GEO search engine on the same held-out queries.

**3.3 Importance Ranking**

“Importance” itself is a broad term and may encode a wide range of objectives, such as recentness, research impact, burstiness etc. of item in the ranking, For sake of simplicity, we only consider research impact as a factor for importance of a dataset.

A natural way to evaluate the importance of a dataset in the research community is to compute the number of times that dataset is used and cited. A more comprehensive way is to include the significance of each publication into this process. In other words, the importance of each dataset depends on *how many times* and *by which* publications is cited. Ideally, given a full citation network of papers one can apply PageRank algorithm to the graph to end up with the ranking of articles. Unfortunately, PageRank only works well if the full citation network is available, i.e. it requires to find all the papers that cited the 216,350 papers in Figure 1 right, and recursively find all the other papers that cited this group which is entirely cumbersome. Only for the first-level descendants of the 216,350 papers in Figure 1 right we found more than 1.5 millions of papers and it expected to grow exponentially with the number of hops. Therefore, instead of building a possibly full citation network of PubMed articles, we simply considered the number of times that each paper is cited as a index of significance of a scientific publication. We then aggregate the number of citations of papers to the corresponding datasets. For example, according to this model top most datasets is shown in Table 1.

**Table 1.** Top 5 important dataset based on total number of Original Paper Citation Count, OPCC, (left) and based on Total Dataset Citation Count, TDCC, (right).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Rank | Accession | OPCC | TDCC | | 1 | GSE5259 | 7559 | 204379 | | 2 | GSE9164 | 4363 | 134104 | | 3 | GSE29611 | 2104 | 17684 | | 4 | GSE1133 | 1984 | 75243 | | 5 | GSE20846 | 1961 | 32257 | | |  |  |  |  | | --- | --- | --- | --- | | Rank | Accession | OPCC | TDCC | | 1 | GSE5259 | 7559 | 204379 | | 2 | GSE9164 | 4363 | 134104 | | 3 | GSE3425 | 1900 | 121390 | | 4 | GSE58 | 1273 | 114180 | | 5 | GSE55 | 1273 | 114180 | |

aaa

**3.4 preference ranking**

a

**4. Results**

Mean Reciprocal Rank (MRR) [39]

Mean Average Precision (MAP) [40]

**5. Conclusion**

Although a conclusion may review the main points of the paper, it must not replicate the abstract.

**Acknowledgments**

The authors would like to thank those that have supported you and your work.

**References**

1. A. Szalay and J. Gray. 2020 computing: Science in an exponential world. Nature, 440(7083):413–414, 2006.
2. C. Southan and G. Cameron. Beyond the tsunami: developing the infrastructure to deal with life sciences data., 2009.
3. S. Leonelli. Introduction: Making sense of data-driven research in the biological and biomedical sciences.Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences, 43(1):1–3, 2012.
4. C. A. Lynch. Jim gray’s fourth paradigm and the construction of the scientific record., 2009.
5. N. C. for Biotechnology Information. Pubmed. http://www.ncbi.nlm.nih.gov/pubmed. [Online; queried 1-July-2015].
6. N.I. of Health. NIH Data Sharing Repositories.  
   http://www.nlm.nih.gov/NIHbmic/nih\_data\_sharing\_repositories.html, [Online; accessed 20-May-2015].
7. N. NCBI. Geo overview: Query and analysis. http://www.ncbi.nlm.nih.gov/geo/info/overview.html#query. [Online; queried 27-May-2015].
8. Lu Z, Kim W, Wilbur WJ. Evaluating Relevance Ranking Strategies for MEDLINE Retrieval. JAm Med Informatics Assoc 2009;16:32–6. doi:10.1197/jamia.M2935
9. Lu Z, Kim W, Wilbur WJ. Evaluation of Query Expansion Using MeSH in PubMed. Inf RetrBoston 2009;12:69–80. doi:10.1007/s10791-008-9074-8
10. Yu H, Kim T, Oh J, et al. RefMed: relevance feedback retrieval system of PubMed. In: Proceedings of the 18th ACM conference on Information and knowledge Management. 2009. 2099–100.
11. Joachims T. Optimizing search engines using clickthrough data. In: KDD. 2002.
12. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at trec-3. In: Proceedings of the 3rd Text REtrieval Conference (1994).
13. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (2004).
14. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (1975)
15. Lee, Dik L., Huei Chuang, and Kent Seamons. "Document ranking and the vector-space model." Software, IEEE 14.2 (1997): 67-75.
16. Zaragoza, Hugo, et al. "Microsoft Cambridge at TREC 13: Web and Hard Tracks." TREC. Vol. 4. 2004.
17. Herbrich, R., Graepel, T., & Obermayer, K. (1999). Support vector learning for ordinal regression. Proceedings of ICANN 1999 (pp. 97–102).
18. Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (1998). An efficient boosting algorithm for combining preferences. Proceedings of ICML 1998 (pp. 170–178).
19. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. Proceedings of ICML 2005 (pp. 89–96).
20. Quoc, C., and Viet Le. "Learning to rank with nonsmooth cost functions." Proceedings of the Advances in Neural Information Processing Systems 19 (2007): 193-200.
21. Zheng, Zhaohui, et al. "A general boosting method and its application to learning ranking functions for web search." Advances in neural information processing systems. 2008.
22. Chapelle, Olivier, and Yi Chang. "Yahoo! Learning to Rank Challenge Overview." Yahoo! Learning to Rank Challenge. 2011.
23. Hang, L. I. "A short introduction to learning to rank." IEICE TRANSACTIONS on Information and Systems 94.10 (2011): 1854-1862.
24. Hsueh, Pei-Yun, Prem Melville, and Vikas Sindhwani. "Data quality from crowdsourcing: a study of annotation selection criteria." Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing. Association for Computational Linguistics, 2009.
25. Li, Hang. "Learning to rank for information retrieval and natural language processing." Synthesis Lectures on Human Language Technologies 7.3 (2014): 1-121.
26. Zheng, Zhaohui, et al. "A regression framework for learning ranking functions using relative relevance judgments." Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007.
27. Plikus, Maksim V., Zina Zhang, and Cheng-Ming Chuong. "PubFocus: semantic MEDLINE/PubMed citations analytics through integration of controlled biomedical dictionaries and ranking algorithm." BMC bioinformatics 7.1 (2006): 424.
28. Dong, Peng, Marie Loh, and Adrian Mondry. "The" impact factor" revisited." Biomedical digital libraries 2.7 (2005): 1-8.
29. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.
30. Moon, Taesup, et al. "Online learning for recency search ranking using real-time user feedback." Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 2010.
31. Zhou, Ding, et al. "Co-ranking authors and documents in a heterogeneous network." Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. IEEE, 2007.
32. M. K.-P. Ng, X. Li, and Y. Ye. Multirank: co-ranking for objects and relations in multi-relational data. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1217–1225. ACM, 2011.
33. Lin, Yongjing, et al. "A document clustering and ranking system for exploring MEDLINE citations." Journal of the American Medical Informatics Association 14.5 (2007): 651-661.
34. Bernstam, Elmer V., et al. "Using citation data to improve retrieval from MEDLINE." Journal of the American Medical Informatics Association 13.1 (2006): 96-105.
35. Ciaramita, Massimiliano, Vanessa Murdock, and Vassilis Plachouras. "Online learning from click data for sponsored search." Proceedings of the 17th international conference on World Wide Web. ACM, 2008.
36. Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview." Dept. of Informatics, Aristotle University of Thessaloniki, Greece (2006).
37. Elisseeff, André, and Jason Weston. "A kernel method for multi-labelled classification." Advances in neural information processing systems. 2001.
38. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." The Journal of Machine Learning Research 12 (2011): 2825-2830.
39. McFee, Brian, and Gert R. Lanckriet. "Metric learning to rank." Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010.
40. Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern information retrieval. Addison Wesley.
41. Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.

1. In fact, GEO repository consist of 58,950 series and 3,848 dataset which each dataset consist of one or more series. Each series is itself is a collection of samples that is submitted to GEO by its creator. In this paper, by GEO dataset we refer to GEO series because the series are referred and cited in the PubMed articles much more often than GEO datasets. [↑](#footnote-ref-0)