SI 650: Information Retrieval
Learning-to-Rank Search Engine for Querying Academic Research Articles and Authors

Team Members :
Nawat Swatthong *(nawatsw@umich.edu)*
Louise Zhu *(louiszyf@umich.edu)*
Jenny Ye *(jennyye@umich.edu)*

Introduction

Attracted by the prestige and outstanding academic performance of higher education in the United States, the number of applications to American graduate schools is increasing year by year. As of now, there are no good matching mechanisms to pair specific research labs and the applicants.

Our project aims at creating a more sophisticated searching mechanism for finding research advisors and labs, according to the specific research interests. Ideally, our search engine will cover a variety of factors, ranging from topics, institute reputation, citations, etc, providing a more universal and personalized ranking. Our main scope of discipline of interest is computer science, where we succeeded in constructing a combination of features from both Google scholar statistics and DBLP citation statistics.

Via the project, we successfully construct a system that can provide rankings of academic authors and research papers according to the queries. Specifically, at the level of authors ranking, three different models are proposed. In analyzing the results, both MAP and NCDG metrics are applied to all the models, showing their respective characteristics. Compared to the baseline at paper-level ranking, our model shows excellent results.

Dataset

For our authors and papers dataset, our team decided to use the search results from the Google Scholar api for the research papers by using Serpapi Google Scholar API that returns the list of papers that show up in the google scholar search based on the specific query. In order to streamline the process, we wrote an api fetching script in Python, which accepts the user input from the command line arguments and returns the specific search results using the Serpapi's API. We did some pre-processing by removing unnecessary columns (e.g. serpapi specific cite links like serpapi_cite_link) and applied some regular expressions to parse the author email column to retrieve the actual university name. Here are some examples of the sourced dataset.

The following are incorporated in our training data:

I. Google Scholar API[1]

As this project needed a good amount of baseline query results for relevance score measures, we use Serpapi's Google Scholar API developer account that allows up to 5000 query searches per month. This dataset will provide the publication information, with proper author and institute information. The API takes citation values and source as input parameters, returning the matched papers in json format. The API supports precise searching by publication years and pagination. The content from this API will be split into training dataset and testing dataset. Our team use the fetching script we use to store the multiple pages of results (10 query results per page) for each query and store the results in .csv format which can be loaded to a dataframe.

1) Research Topic query - Research Paper documents data

- Data Source : https://serpapi.com/google-scholar-organic-results
- Example dataset link : paper_data
- Columns : **query** (search query), **position** (order of the search results within Google Scholar), **name** (name of the author), **affiliations** (affiliations written in the author's bio), **email** (email written in the author's bio in Google Scholar page), **university** (University info retrieved by the affiliations/email columns) **cited_by** (The total number of cited papers for a given author), link (Google Scholar page link for the author), **author_id** (Unique author ID), **interests** (Interests area that the author specified in the Google Scholar page), **thumbnail** (Thumbnail picture for the Google)
- Statistics : Since this dataset is based on the search results that can be retrieved from Serpapi API, we have the flexibility to retrieve the total number of search results per queryset. For example, each page contains 10 authors, and we can retrieve multiple pages of search results for a certain query. The example dataset contains 10 authors list for each query. We're going to use the ground level relevance score by how the Google Scholar profile search returns the authors list, which is the "position" column in the dataset.

2) Research Topic query - Author lists data

- Data Source : https://serpapi.com/google-scholar-profiles-api
- Example dataset link: authors_data
- Columns : **query** (search query), **position** (order of the search results within Google Scholar), **title** (name of the paper), **result_id** (the unique idea of the paper result), **type** (type of the paper), **link** (link to the specific paper), **snippet** (Snippet of the actual research paper document), **publication_info** (Contains a dictionary of summary of the

[1] https://serpapi.com/google-scholar-api

paper, and the list of authors), **authors** (The first author's name that's retrieved from the search)

- Statistics : Since this dataset is based on the search results that can be retrieved from Serpapi API, we have the flexibility to retrieve the total number of search results per queryset. For example, each google scholar page contains 10 papers, and we can retrieve multiple pages of search results for a certain query. The example dataset contains 10 papers for each query. We're going to use the ground level relevance score by how the Google Scholar paper search returns the research papers list, which is the "position" column in the dataset.

II. **Citation Network Dataset**[2]

A publication collection dataset that focuses on information of publications extracted from DBLP, ACM and MAG (Microsoft Academic Graph). Among the collections, version 12 (DBLP V12), having the timestamp in 2020, is chosen as our reference network.

Related Work

[2] Donthu et al. offers a comprehensive guide to bibliometric analysis, providing valuable methodologies that can be incorporated into the ranking algorithm of the project. Ding, Foo & Chowdhury [3] focus on the bibliometric analysis of collaboration in the field of information retrieval. The research likely focuses on understanding the landscape of collaborations rather than ranking individual researchers or institutions based on their expertise in a particular area. Ellegaard & Wallin [4] delve into quantifying scholarly impact through bibliometric analyses. The paper seems to focus on understanding the impact of scholarly work but does not aim to build a retrieval system to rank researchers or institutions. However, the scholarly impact quantification approach can be another dimension that could be integrated into the system to give a various view of a researcher's contribution. Hou & Jacob [5] compare various metrics and methodologies used in ranking higher education institutions, offering additional factors that could be considered in the project's ranking algorithm. Lastly, Moreira et al. [6] utilizes machine learning algorithms to rank academic experts. However, their research's approach could be limited to the DBLP dataset that needs some adjustment to apply with more expertise fields. Together, these articles provide a range of methodologies and metrics that could be complementarity combined to build a robust ranking system for academic expertise in specific fields.

Methodology

Our work is divided into two parts: paper ranking, and author ranking. For paper level, we conducted document preprocessing on both the paper's body text and title by removing stop

[2] http://www.arnetminer.org/citation

words and using RegexTokenizer to remove punctuation, followed by indexing and ranking. We then added publication-related and citation-related metrics into account such as number of total citations of each author, or actual abstract document of each research paper etc.

After obtaining all scores and rankings for each paper-query pair, we aggregate them at the author level to rank authors relevant to the queries. Google scholar profile information (bio) was used to return initial author query results.

I. Create an updated paper ranking

A total of 4,894,081 papers are considered in our network. The following metrics of the network are extracted as features:

a. The pagerank score
b. The Hub and Authority score
c. Paris hierarchy: a hierarchical analysis of the nodes in the citation network
d. Principal Component Analysis (PCA): apply PCA to the (directed) adjacency matrix of the citation network.

Apart from the network features, the below variables are taken into account, for each paper:

a. The length of the abstract
b. The length of the title
c. The length of the query
d. Document body text TF
e. Document body text TF-IDF
f. Document title TF
g. Document title TF-IDF
h. Pivot Normalization
i. Year of release: the year in which the paper is published
j. Citation number: equivalent to the the original ranking in Google's search results
k. Cross-Encoder score
l. Field of study: the tags of the disciplines to which the paper belongs. The tags we considered are software-development, human-computer interaction, engineering, concurrent computing, and marketing.

An additional explanation to be added to these features is that the "text" of the document refers to the abstract of the paper. In making this decision, we aim at reducing the computing resource that this system will consume.

The above features will be used in our L2R ranking system. What distinguishes our ranking from the original ranking in the google scholar is that we only consider the citation from the papers which are also within the DBLP citation network, whereas the Google ranking will accumulate all possible citations of a paper, regardless of the disciplines.

II. Create an updated author ranking

1) Ideally, the updated ranking will depend on the relevance scores from each of the papers associated with this author. To simplify the scoring system, we accumulate the relevance score of each paper obtained in step (I). In order to better capture the contribution of each author in the papers, we designed a weighting factor for every author listed in the paper, as:

$$r_{j,q} = \sum_{i \in N_q^{10000} \cap C_j} r_{i,q} \big/ o_{i,j}$$

- $r_{j,q}$ is the relevant score of author j to query q
- $r_{i,q}$ is the relevant score of paper i to query q
- $o_{i,j}$ is the contribution order of author j to paper i (e.g. first author = 1, second author = 2)
- N_q^{10000} is the set of top-10k ranking of papers from query q
- C_j is the set of papers belong to author j

By this equation, the system ensures that the first author (and co-authors) take the largest share of the relevance score of the paper.

2) We edit author ranking system from method 1) by replacing the contribution order into the citation number of each paper as:

$$r_{j,q} = \sum_{i \in N_q^{10000} \cap C_j} r_{i,q} ln(c_i + 1)$$

- c_i is the citation number of paper i

3) This method focuses on the author representation constructed by the paper collection and paper-level relevant scores. We encode all titles of papers into dense vectors using the Sentence Transformer. Then we combine all title vector belong to each author and weight by relevant score in order to get a vector representing the author as:

$$v_j = \sum_{i \in N_q^{10000} \cap C_j} r_{i,q} v_i$$

- v_j is the encoded vector for author j
- v_i is the encoded vector for title of paper i

We also encode query words into vectors and then calculate relevant scores of authors to a query by using cosine similarity between author vector and query vector.

Evaluation and Results

I. Paper-level results:

Two methods are used in this section, namely BM25 and L2R. The corresponding scores at position 10 are shown in Figure 1. It is noteworthy that with our reference baseline, the MAP score is very low, whereas the NDCG metric performs well.

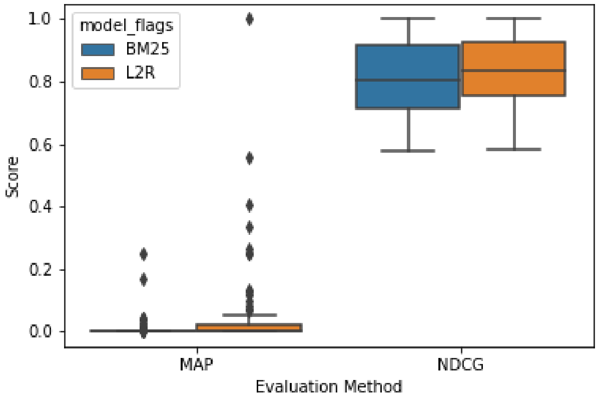

Figure 1: Paper-level Model Performance

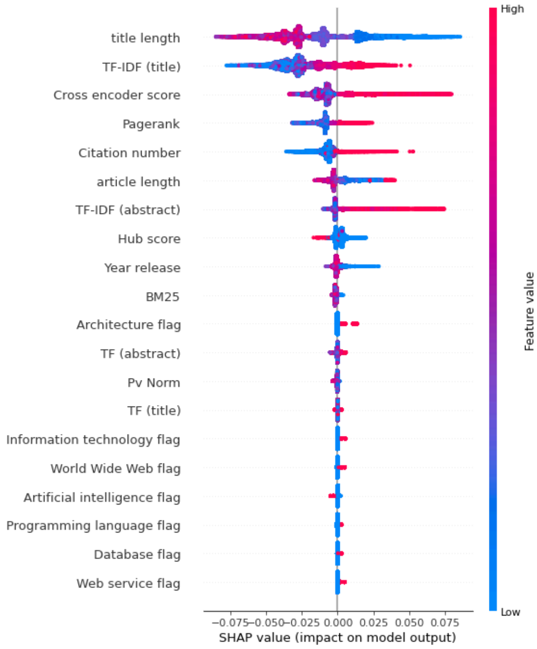

Figure 2: SHAP value of L2R model

Among the features used in producing the L2R model results, we use the SHAP metric to illustrate the importance of them as shown as Figure 2. The length of the paper title is proven to be the most important feature. Other positively influential parameters include number of citations, cross encoder score, and TF-IDF title. One explanation is that the title length, by involving in TF-IDF calculation, contributes to its high influence. Notice we have a negative relationship between the title length and the positive/negative sign its influence: a shorter title length represents a more positive influence. This can be attributed to the fact that the most influential academic works are generally shorter in title length, because they will deliver a more fundamental and comprehensive analysis on the subject(s). Without doubt, the number of citations is positively related to the rankings; and the cross-encoder score is positively capturing the relation between queries and documents. It is noteworthy that only high TF-IDF have an influence on the results, and low values don’t make a difference.

On the other hand, the simple TF of paper title contributes negatively to the model. One possibility is that, in academic research papers, the words used in titles will be discipline-specific, not easily associated with simple query keywords such as discipline names. The situation will be exacerbated when the title is not long. This explanation also holds for our reference baseline, where the ranking is determined by Google API.

II. Author-level results

The results for author ranking from three methods described in the methodology section are shown in FIgure 3. The system performs slightly worse given the scores from the evaluation.

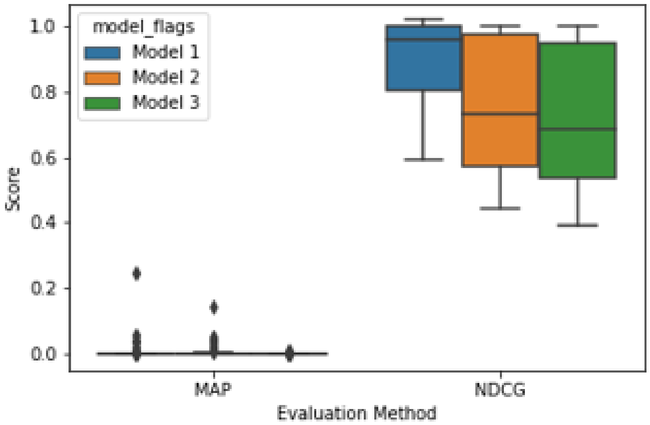

Figure 3: Author-level Model Performance

Model (1) which partitions the contribution of each author shows smaller variance, While model (2) utilizing citation number as a factor and model (3) using vector encoder provide higher variances. It is possible because that one paper typically involves multiple authors, and in certain disciplines, this number is extremely huge. A simple collection of papers which a researcher has, regardless of contribution, will have a larger variation in collaborators, and finally larger variance in author rankings.

Discussion

I. Applicability in Real Settings

- The affiliations of an author may change, yet we fail to capture that in profile accordingly. Neither do we have accessibility on the timestamp at which the profile is updated.
- Admittedly, our deliverable shows deviance with the initial design. Currently, our project satisfies the needs of paper and researcher ranking, yet without the support of a stable website (and approachable user interface), additional time is needed if our system is to face end-users.
- However, we still hold the confidence that our system is appropriate for providing innovative methodologies in ranking the researchers (higher institution employees).

II. System performance compared to Baseline

- Our baseline is BM25, for paper-level ranking. From the brief analysis in the methodologies section, it can be seen that our model is slightly better. This can be attributed to our broad variety of selected features.
- The possible reason why the vector encoder in the model (3) is not well-performed is that the title vector combination may not fit with the ranking score system from Google API. Google provides ranking positions for each author by using only profile biography, citation number and field of work, without information about the author's academic paper. Therefore, the ranking for Google will not be able to capture the depth of information gathered from papers, thereby paper title vector method doesn't perform well.

Conclusions

I. Other Things We Tried

One of the important tasks for building our IR system was to get enough dataset to build the baseline results. While the fetching script that we wrote using the Google Scholar API provided approximately 10K papers list for 50 different queries, and 5K authors list, since the Google Scholar API would charge additional fee for 5000+ search/month, we faced a challenge to retrieve more datasets to train and test our features applied to the L2R models. Also, we faced some challenges with computing power where it took us about 21 hrs to encode the paper title, which has relatively shorter length, hence requiring less computing power and storage requirements than encoding the actual paper document. Given this, we were not really able to encode the entire abstract or the actual paper document text data.

II. What You Would Have Done Differently or Next

The dataset that needs pre-processing is too large and messy, which doesn't leave us much space to construct the final level of ranking according to our original plan, the institution-level ranking. Initially, our team was planning to incorporate university ranking into paper-level as well as author-level models so that we can apply more relevant metrics for ranking the most influential research labs that prospective users (our target users are graduate school applicants). However, the institution ranking dataset we found (Integrated Postsecondary Education Data System (IPEDS) was pretty large and needed a lot of pre-processing, neither did we come up with very plausible and applicable system of how to integrate the results from the author-level rankings to institution-level rankings.

Another dissatisfaction comes from the scope we are investigating. Originally, the expected disciplines should cover multiple disciplines, yet we failed to find a proper dataset that can capture the network relationships between the paper citations. Although this can be done via web scraping, yet we didn't come up with such a pipeline.

Team Work Distribution
paperwork – Nawat, Louise, Jenny
DBLP Database preprocessing – Nawat
Google API data preprocessing – Jenny
Network feature generation – Louise
IPEDS dataset processing (abandoned) - Louise
Model development – Nawat, Louise, Jenny
Model implementation – Nawat

References

[1] Zhou, E. (2022). Graduate Enrollment and Degrees: 2011 to 2021. Washington, DC: Council of Graduate Schools

[2] Naveen Donthu, Satish Kumar, Debmalya Mukherjee, Nitesh Pandey, Weng Marc Lim, How to conduct a bibliometric analysis: An overview and guidelines, Journal of Business Research, Volume 133, 2021, Pages 285-296, ISSN 0148-2963, https://doi.org/10.1016/j.jbusres.2021.04.070.

[3] Ying Ding, Schubert Foo, Gobinda Chowdhury, A Bibliometric Analysis of Collaboration in the Field of Information Retrieval, The International Information & Library Review, Volume 30, Issue 4, 1998, Pages 367-376, ISSN 1057-2317, https://doi.org/10.1006/iilr.1999.0103.

[4] Ellegaard, O., Wallin, J.A. The bibliometric analysis of scholarly production: How great is the impact?. *Scientometrics* **105**, 1809–1831 (2015). https://doi.org/10.1007/s11192-015-1645-z

[5] Hou, Ya-Wen & Jacob, W.J.. (2017). What contributes more to the ranking of higher education institutions? A comparison of three world university rankings. International Education Journal.16.29-46.https://www.researchgate.net/publication/322366231_What_contributes_more_to_the_ranking_of_higher_education_institutions_A_comparison_of_three_world_university_rankings

[6] Moreira, C., Calado, P., & Martins, B. (2015). Learning to rank academic experts in the DBLP dataset. *Expert Systems*, *32*(4), 477-493.https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.12062