

# Hybrid Scalable Researcher Recommendation System Using Azure Data Lake Analytics

Dinesh Kalla<sup>1</sup>, Nathan Smith<sup>1</sup>, Fnu Samaah<sup>2</sup>, Kiran Polimetla<sup>3</sup>

<sup>1</sup>Department of Doctoral Studies, Colorado Technical University, Colorado Springs, CO, USA

<sup>2</sup>Department of Computer Science, Harrisburg University of Science and Technology, Harrisburg, PA, USA

<sup>3</sup>Adobe Technology Service (ATS) Department, Adobe, San Jose, CA, USA

Email: dinesh.kalla@student.ctuonline.edu, Nathan.smith246@student.ctuonline.edu, fsamaah@alumni.harrisburgu.edu, pri20074@adobe.com

**How to cite this paper:** Kalla, D., Smith, N., Samaah, F. and Polimetla, K. (2024) Hybrid Scalable Researcher Recommendation System Using Azure Data Lake Analytics. *Journal of Data Analysis and Information Processing*, 12, 76-88.

<https://doi.org/10.4236/jdaip.2024.121005>

**Received:** January 8, 2024

**Accepted:** February 5, 2024

**Published:** February 8, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

This research paper has provided the methodology and design for implementing the hybrid author recommender system using Azure Data Lake Analytics and Power BI. It offers a recommendation for the top 1000 Authors of computer science in different fields of study. The technique used in this paper is handling the inadequate Information for citation; it removes the problem of cold start, which is encountered by very many other recommender systems. In this paper, abstracts, the titles, and the Microsoft academic graphs have been used in coming up with the recommendation list for every document, which is used to combine the content-based approaches and the co-citations. Prioritization and the blending of every technique have been allowed by the tuning system parameters, allowing for the authority in results of recommendation versus the paper novelty. In the end, we do observe that there is a direct correlation between the similarity rankings that have been produced by the system and the scores of the participant. The results coming from the associated scrips of analysis and the user survey have been made available through the recommendation system. Managers must gain the required expertise to fully utilize the benefits that come with business intelligence systems [1]. Data mining has become an important tool for managers that provides insights about their daily operations and leverage the information provided by decision support systems to improve customer relationships [2]. Additionally, managers require business intelligence systems that can rank the output in the order of priority. Ranking algorithm can replace the traditional data mining algorithms that will be discussed in-depth in the literature review [3].

## Keywords

Azure Data Lake, U-SQL, Author Recommendation System, Power BI, Microsoft Academic, Big Data, Word Embedding

## 1. Introduction

Academic entities can be found by using Microsoft Academic as the search engine. Some of the entities include the venues, journals, and conferences, affiliations such as organizations and institutions, the authors, study fields or fields of study, patents, and papers are some of the top-level entities of Microsoft Academic Graph. This study aims to present the paper similarity computation and the recommender for the patents and the English papers in the Microsoft Academic Graph unless otherwise stated. The word paper in this context means patents and English papers in the Microsoft Academic Graph. The algorithms can generate several relationships between data and another variable [4]. Most BI systems utilize statistical techniques to analyse and create reports used by managers to make decisions [5]. Companies such as Oracle, Microsoft dynamic, and SAP ERP software have business intelligence functionality [6].

SAS, a statistical tool, currently has business intelligence functionalities. Additionally, SQL Server 2005/2008 also contains business intelligence functionality to help users sieve specific information from the database server [7]. Business intelligence systems offer more advanced reporting features than traditional statistical software; therefore, managers find it more intriguing to use data mining tools that can provide reports in sequential order to support their decision-making process [8]. As managers grapple with huge data that they use to make a decision, there is a need to rank the reports generated by BI systems based on their significant impact on business processes [9].

To have a competitive advantage and effectively run a business, there is a need to rely on automated data mining tools that can support their daily activities [10]. Page rank algorithm is used by Google search engine to rank websites based on their importance. Google scholar is the commonly used recommender system for research papers. The increasing number of research papers makes it difficult for novice academic researchers to obtain relevant scholarly articles with ease. The current recommender system does not have scalable functionality to cater to the increasing number of journals and articles published daily in the global arena. Other recommender systems such as Microsoft Academic, Web of Science, and PubMed databases use the collaborative approach, hindered by data protection legislation. Additionally, the said academic search engines domicile in one particular field of study. My research paper will focus on resolving this problem by recommending a hybrid scalable search engine that will utilize a page rank algorithm to determine the relevance of academic documents and display them in the order of importance. It will combine content-based and co-citation recommendations to produce related documents, including the recently published ones, without compromising the quality of recommendations.

The paper recommender systems in existence currently have a lot of limitations. Apart from Science Web, Semantic Scholar, Microsoft Academic, Google Scholar, and other key players, most of the paper search engines are meant only for specific domains like Biology and Medicine, the IEEE Xplore for engineering

fields. As a result, the recommender systems on the specific search engines can't come up with a suggestion of the cross-domain recommendations.

The platform of the Microsoft Academic paper recommender has an objective of removing some of the limitations that have been mentioned through various ways, which include deployment of the whole Microsoft academic graph citation network and the interdisciplinary corpus of close to two hundred million papers, utilization of the content embedding and the co-citation in optimizing the coverage for a recommendation over the whole corpus as you maneuver the cold start problem and finally coming up with the recommendations that are computer-based to the engineering fraternity and the broader research, so that be thoroughly improved through analysis by other research peers.

Microsoft Academic Recruiting Tool is Powerful Tool for Recruiting purpose at Microsoft AI + R teams and it makes recruiting job easy and simple. It is sub version of Microsoft Academic portal which contains the top 1000 authors for every Computer Science Field of Study.

As of now it is limited to Computer Science and Information technologies related field of studies and the primary objective is to find out Top Authors based on filters such as Field of Study and Career start year of Authors.

This tool contains the Author Page which is Main page, and it contains the top 1000 authors and their papers, and citations for every computer science field of study selected. Drill down page is Author details page which provides details of each author for particular field of study selected including Top 3 papers, Affiliations, Top Conferences, Top Journals, Co-Authors and other related field of studies. It also contains total papers and citations count of Author and counts per Field of Study Selected. This tool has provided several URL icons which will help to navigate it to Microsoft Academic portal directly to get details of particular Author, Paper, Journal, Conference and Affiliation respectively.

## 2. Literature Review

The complex issues that stem from coming up with a recommendation system for the Microsoft Academic have requirements as follows; one is coverage which means you need to optimize the coverage of recommendation compared to Microsoft Academic Graph corpus. The second one is scalability because the recommender generation should be done with storage and time requirements because the Microsoft academic graph crops hundreds of thousands of new papers almost weekly. The third requirement is freshness. Any new papers brought about by the Microsoft academic data pipeline have to be designated papers related to them. They can be termed as the related papers in the already existing corpus. The fourth and final one is user satisfaction. This means that a balance ought to be struck between the novel recommendations and the authoritative recommendations. This will be good because it means that the new papers can be discovered with zero compromises on recommendation quality.

In handling these requirements, the paper has come up with a recommender system that applies the co-citation-based recommendations to produce the rec-

ommendations list for each paper in the Microsoft academic graph and a hybrid recommender system that applies a mapping function that can be tuned. As asserted in this particular technique or approach deploys the weighted mixed hybridization approach. This content-based approach is the same as the previous work which was done on the content embedding-based citation recommendation as put in. It, however, majorly differs in the fact that the paper applies the clustering approaches for more speedup. Using the pure dual information embedding common for the closest information search is not sustainable because it is an complex problem compared to the whole paper collections. In this particular scenario, it can be  $2.56 \times 10^{16}$  computations that are similar.

Recommender systems have been applied in a variety of applications such as Twitter and Netflix. For instance, Twitter uses PageRank to suggest new followers to its users. To generate this suggestion, the application is designed in such a way it performs personalized PageRank by establishing two copies of each active user; a content producer and a content consumer. In this sense, a user can post a tweet or, in other cases, read a tweet by other users. A content produced by the user can be retweeted, further-reaching a new audience. This developing network can leverage on PageRank algorithm to suggest new followers.

Another application scenario is in movie rental sites such as Netflix, where a user considers a movie to be relevant if other users like the movie. The second approach to like a movie is when a mutual friend likes movies that we both have interests. With these two approaches, Netflix can create a personalized ranking for users on the film they want using the PageRank algorithm. The “behavior” of other mutual users on the platform enables the system to determine what other users might like.

Data mining systems also utilize a similar approach where user behaviors can predict what products they might like [11]. In Business intelligence systems can track user preferences and suggest new products to the user in priority. Data mining is categorized into three groups, namely, web content mining, Web usage mining, and web structure mining. Web structure mining uses the PageRank concept to analyze and rank web pages in an internet search.

Also, a page ranking algorithm is applied to academic services such as e-libraries, e-learning, online purchasing systems, and news synthesis. In this way, users are provided a personalized recommendation based on their liking behavior of a certain service or product. In this paper, I will focus on offering an alternative solution to the problematic collaborative filtering (CF) by taking into account the freshness of contents, scalability feature, improved user satisfaction, and turnaround time (computational time ) to generate a search list.

### **3. Architecture of Author Recommendation System**

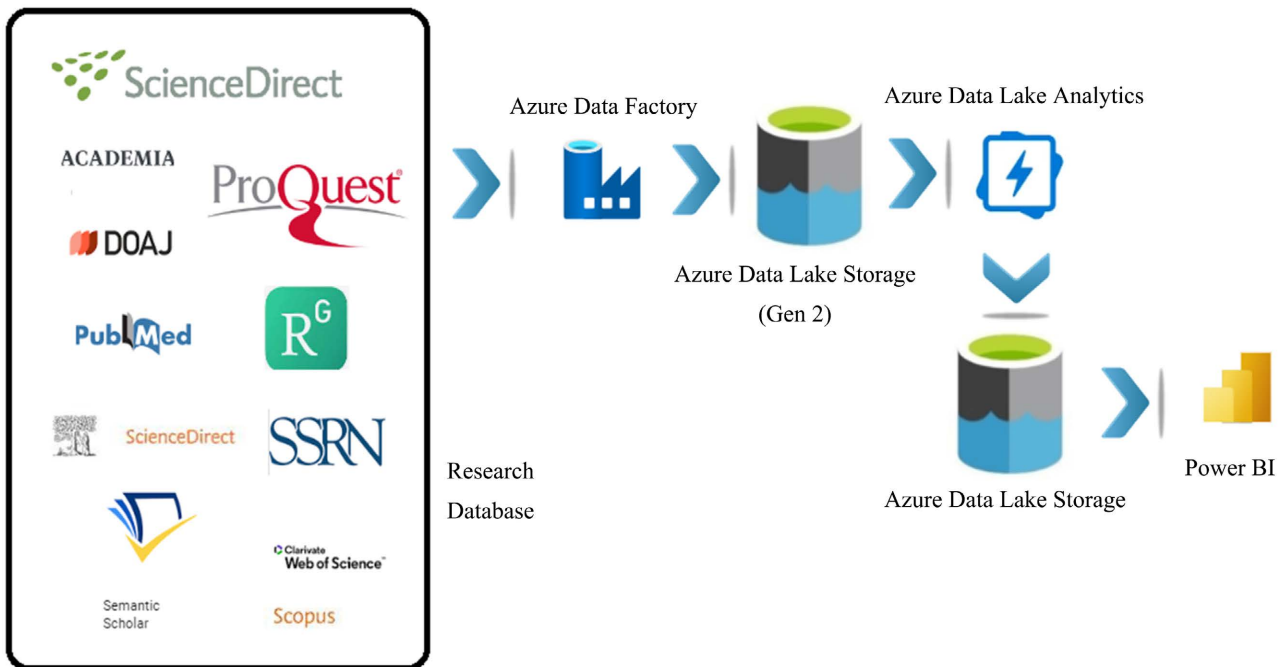
Author recommendation system collects published papers from major journals and conferences which is in the form of unstructured data and semi structured data. The data is further moved to Microsoft Big Data Platform Azure Data Lake

using Azure Data Factory. The data is further extracted from storage account using Azure Databricks or Azure Data Lake Analytics where data pre-processing, data transformation and ranking algorithm implemented. Microsoft Azure data factory is further used for scheduling pipelines and transferring data from one storage location to different big data storage location for keeping data up to date. The transformed data is further pushed into Power BI to create dashboards for the Recruiters. **Figure 1** shows architecture where all the stages of author recommendation system implemented includes monitoring, ingesting, storing and analyzing.

Azure Data Lake analytics extract data from the big data storage platform with the help of U SQL script which is combination of SQL and C#. Entire transformations and implementation of ranking algorithm will be written with the help of U-SQL scripts on Azure Data Lake Analytics platform. The structured and transformed data will be further stored in Big data platforms like Azure data lake storage, SQL and data warehouse. The pipeline scheduling and data movement to maintain data up to date will be performed using Azure Data Factory. **Figure 2** shows the U-SQL code for Author ranking based on number of paper and citation.

#### 4. The Recommendation Generator

The recommender system applied the hybrid technique of CcB and the CB recommendations. The CcB recommendations are considered to be of high quality because they show a large +ve correlation with generated scores of the user. However, the paper citation information is at times hard to acquire, and thus the



**Figure 1.** Architecture/flow diagram of author recommendation system.

```

// Get author pagerank
authorPaperRanks =
SELECT authorsInFos.PaperSeqId AS PaperSeqId, authorsInFos.AuthorSeqId AS AuthorSeqId,
authorsInFos.FieldOfStudySeqId AS FieldOfStudySeqId, Math.Exp(papers.HeteroRank / -1000) AS
PaperProb FROM authorsInFos INNER JOIN papers ON papers.PaperSeqId == authorsInFos.PaperSeqId}

authorPaperCount = SELECT AuthorSeqId, FieldOfStudySeqId, COUNT() AS PaperCount,
Math.Log(SUM(PaperProb)) AS LogProb FROM authorPaperRanks GROUP BY FieldOfStudySeqId,
AuthorSeqId HAVING PaperCount > 2;

authorNameRankData =select authors.AuthorSeqId AS AuthorSeqId, authorPaperCount.LogProb AS
LogProb, authors.AuthorName AS AuthorName, authorPaperCount.PaperCount AS PaperCount,
authorPaperCount.FieldOfStudySeqId AS FieldOfStudySeqId, authors.AuthorSeqId.ToString() + "+" +
authorPaperCount.FieldOfStudySeqId.ToString() AS JoinKey FROM author INNER JOIN authorPaperCount
ON authorPaperCount.AuthorSeqId == authors.AuthorSeqId;

authorData = SELECT authorNameRankData.FieldOfStudySeqId AS FieldOfStudySeqId,
authorNameRankData.AuthorSeqId AS AuthorSeqId, authorNameRankData.AuthorName AS
AuthorName, authorNameRankData.LogProb AS LogProb, authorNameRankData.PaperCount
AS PaperCount, authorCitCountJoinKey.AuthorCitationCount AS CitationCount,
authorFirstPaperYearJoinKey.FirstPaperYear AS FirstPaperYear FROM authorNameRankData
INNER JOIN authorCitCountJoinKey ON authorCitCountJoinKey.JoinKey ==
authorNameRankData.JoinKey INNER JOIN authorFirstPaperYearJoinKey ON
authorFirstPaperYearJoinKey.JoinKey == authorNameRankData.JoinKey;

authorsRanked = SELECT authorData.FieldOfStudySeqId AS FieldOfStudySeqId, fos.DisplayName
AS FieldOfStudyName, ROW_NUMBER () OVER (PARTITION BY authorData.FieldOfStudySeqId ORDER BY
authorData.LogProb DESC, authorData.CitationCount DESC, authorData.PaperCount DESC ) AS Rank,
authorData.AuthorSeqId AS AuthorSeqId, authorData.AuthorName AS AuthorName,
authorData.PaperCount AS PaperCount, authorData.CitationCount AS CitationCount,
authorData.FirstPaperYear AS FirstPaperYear, authorData.LogProb AS LogProb FROM authorData
INNER JOIN fos ON fos.FieldOfStudySeqId == authorData.FieldOfStudySeqId
HAVING Rank <= @Param_AuthorTakeCount;

```

**Figure 2.** U-SQL code for ranking authors based on number of papers and citations.

CB technique suffers as a result of low coverage. This issue can be combated by the use of information embedding similarity-based recommendations. The CB recommendation system are obviously costly and of less quality as compared to the CB recommendations, but they come with advantages of coverage and freshness. It is just the metadata of the paper that includes keywords, abstracts that are available, and titles that are required in the generation of such recommendations. Because every paper in Microsoft academic graph must have a headline, the research can find the content embedding for each pdf document in the research paper collection as you depend on the heading if required.

#### 4.1. Citation Based Recommendation

Citation based algorithm leverages the relation between authors based on citation patterns. It refers to the occurrence of two papers being cited together by third research paper. Just factor a collection of papers,  $P = \{p_1, p_2, p_3, \dots, p_n\}$ . The paper used  $C_{i,j} = 1$  to show that  $p_i$  is citing  $p_j$ , or 0 otherwise. The co-citation count between  $P_j$  and  $P_i$  can be deduced by using the formula.

$$cc_{i,j} = \sum_{k=1}^n c_{i,k} c_{j,k} \quad (1)$$

When  $c_{i,j} \geq 1$  we call that  $P_i$  is co citation of  $P_j$  vice versa. Co-citation captures implicit relationship between paper and can lead to serendipitous recommendations by suggesting paper that are not related to researcher current preferences.

This method is often more robust in the face of sparse research data compared to recommendation methods. Co-citation patterns can be utilized to identify communities and focus areas of related papers. This pattern can reveal contextual insights into the relationships between papers. Based on the paper citation count authors are recommended as per specific field of study.

## 4.2. Heterogeneous Rank Based Recommendation

The Heterogeneous network comprises of three network nodes and the rank operates based on the whole network retrieving information back and forth in iterative pattern. Entire process combines the outcome from paper citation network and two inter class network which is between article and author and also article and journal bipartite graph. Let  $R^p$  be the vector of author rank by  $R^A$  and vector of journal rank by  $R^J$ . Eventually  $R^p$   $R^A$   $R^J$  corresponding topic vector of  $n \times T$ ,  $m \times T$  and  $q \times T$  matrix. Here  $q$  is number of journals,  $m$  is number of authors,  $n$  is number of papers and  $T$  is number of topics. The author score will be calculated based on the below algorithm

$$R^A = M^{A^T} * R^p \quad (2)$$

Here  $M^{A^T}$  is the transpose of paper author adjacency matrix and  $R^p$  is topic probability of individual paper for a particular field of study.

Above equation shows paper transfer author score to author and author receives authority scores from publication. Authors rank dependent on paper and also journal rank dependent on published paper which makes paper is most deciding factor or entity in the Author rank algorithm. Author rank is based on paper citation network and bipartite author network.

## 4.3. Combined Recommendation

Eventually, both the Citation and the heterogenous rank sets for a Author are joined up in the creation of a uniform last set of recommendations for the authors. The Co-citation candidate sets have counts of co-citation that are related to every paper recommended pair. The Co-citation counts are mapped to scores of between the zero or one to enable it to relate them with Combined similarity. The function is an altered log function as seen in the equation below;

$$\sigma(cc_{i,j}) = \frac{1}{1 + e^{\theta(\tau - cc_{i,j})}} \quad (3)$$

$\tau$  and  $\theta$  are tunable parameters used in controlling the offset and slope of the logistic sigmoid. In this research  $\tau$  and  $\theta$  values to 5 and 0.4 is set based on the factor of standard deviation and mean co-occurrence count of research papers [12] [13] [14]. In Equation (3) changing the tunable parameter permit one to weigh content based versus co-citation based recommendations. Citation and heterogeneous rank based methods can complement each other's strengths and compensate for each other's weaknesses, leading to more robust and accurate recommendations. While combining these approaches offers various advantages, the implementation details and the success of the hybrid model depend on the

specific characteristics of the dataset, the nature of the items being recommended, and the user interaction patterns. Experimentation and fine-tuning are often required to achieve the best performance for a given recommendation scenario. In this method author with more papers and author with more citations together will be checked along with the journal quality. Based on journal quality, number of papers, citations count and the paper which is citing authors paper Authors rank is derived.

## 5. Results and Discussion

Recruiting Tool helps us get Top 1000 Research Authors based on the Filters of Research area and Career Start year filter. Research area or Field of Study field help us to find out authors related to that particular Focus area and Career Start year Filter helps us to know the most recent active Authors on that particular Focus Area. We have also provided additional Filter which contain List of Author Names which help us search author by using Author Name.

We have provided a profile link beside author's name which will help to Navigate to Microsoft Academic portal. Ranking of Author has done using page rank algorithm by considering parameters like Citations, Number of Papers, Field of Study Level and Hetero Rank of Microsoft Academic.

**Figure 3** shows the main dashboard page of the author recommendation dashboard containing filters like field of study, author name and year filters. The field of study filter will contain all the major fields related to Major computer science since it is portable system. Career Year filter divided into four parts papers published for 2016 year above, from 2013-2015, 2010-2012 and 2009 below which will help hiring authors based on the years of Active. As recruiters try to hire active authors who are recently publishing papers in the trending fields of computer science and in those cases 2016 and above year button in career filter will help in hiring authors who are currently active. Based on the filters final results of Top 100 authors will be displayed along with information included number of papers published, total citations received. Field of study and career start date.

We can also further Drill through to Author Details Page by right click on Author Name → Drill through → Author details page. **Figure 4** shows how the select the drill through options on the main dashboard. The Author Details page contains details of Authors like Total Citations, Total Papers, Papers per Field of Study and Citations per Field of Study. It also provides other features like authors top 3 papers per Field of Study selected, Affiliations, Top Conferences, Top Journals, Other Focus Area and also Top Co-Author Names. All details provided have been tagged with URL icon in front of them which will help to Navigate to Microsoft Academic Web Portal. From drill down page, we can go back to Main page with the help of Arrow Sign displayed from the top of the page. This drill down information will help to check further about the authors like their top 3 papers, most cited papers, current affiliation, top journal, top conferences, top focus areas and co-authors of that particular author. The back button will help to



navigate to the main page which is actual author recommendation main page. The result of this page is displayed with the help of page rank algorithm and citation algorithm whereas paper and citation counts are calculated using the U SQL scripts in Azure Data Lake Analytics.

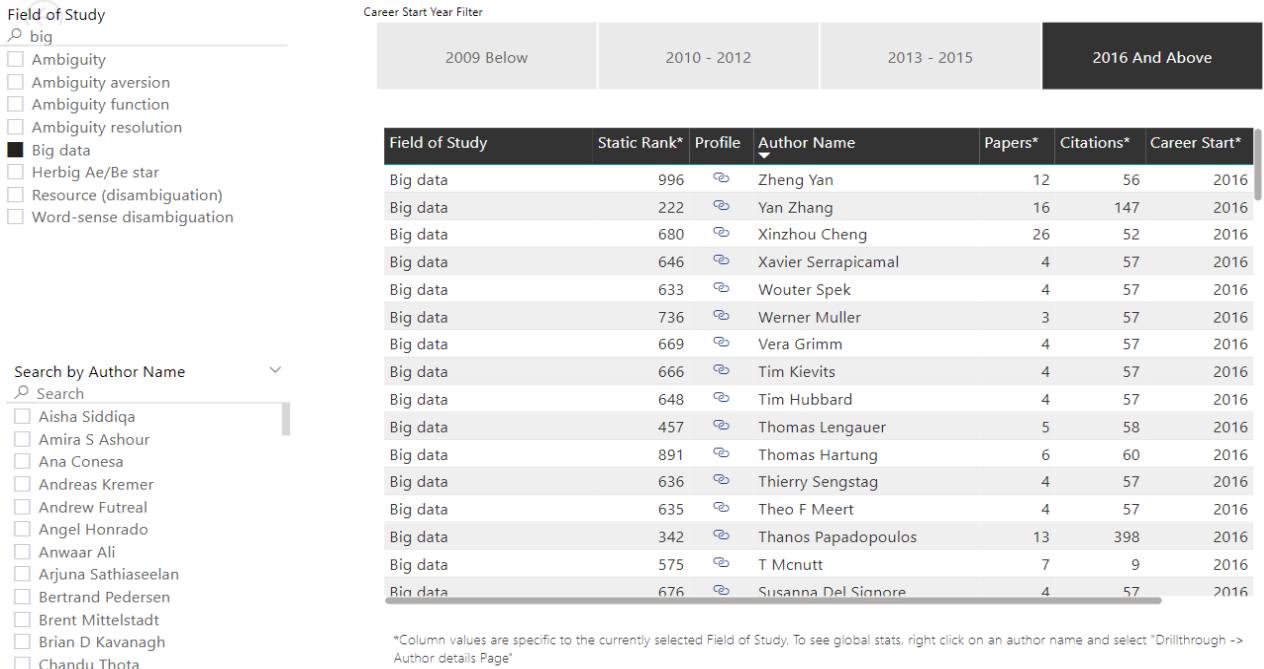


Figure 3. Main page of author recommendation system tool.

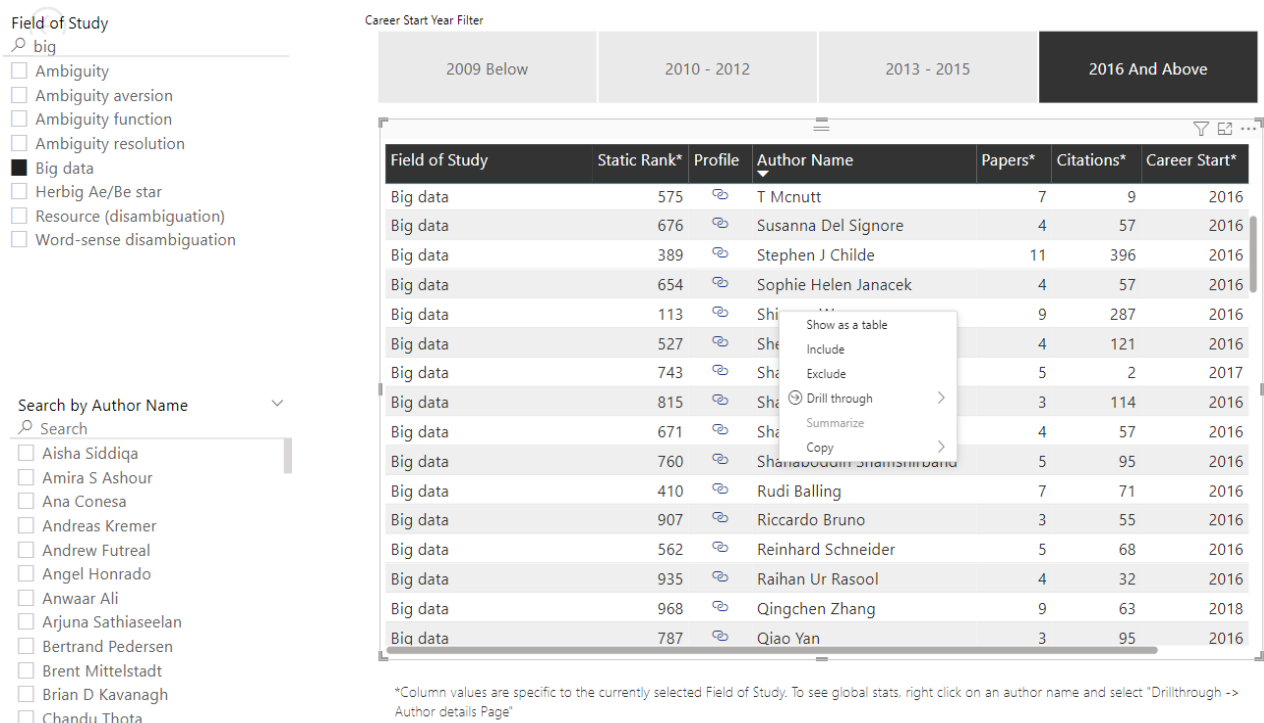
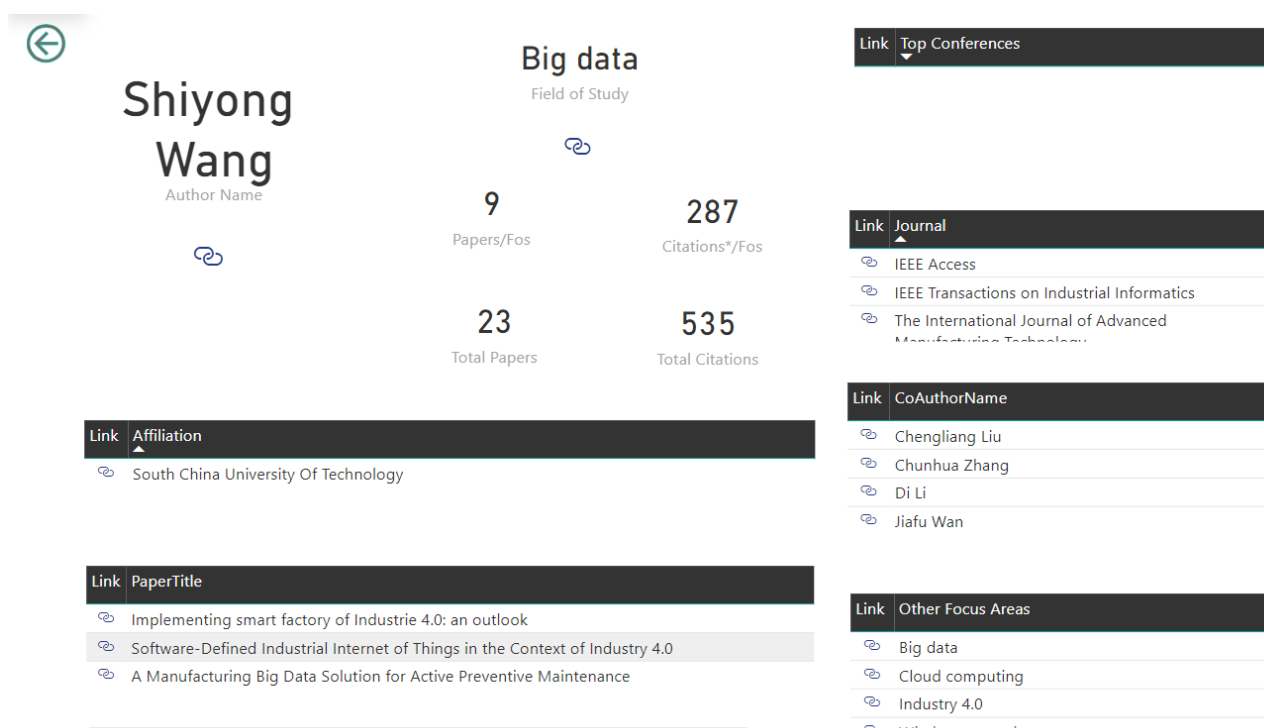


Figure 4. Drill through functionality of author recommendation system tool.



**Figure 5.** Drill down page of author recommendation system tool.

**Figure 5** shows Author drill down page which will help to obtain further information about the author like number of papers he published and number of citations he received in general and also in specific to field of study selected. For example, if we have to hire someone in robotics, we have to make sure Authors has more papers and citation specific to that particular field of study. Affiliation details will help to know which affiliation author belongs to. Sometimes recruiters want to hire someone from a reputed university and organization. Other focus area information will help recruiters to check all other interested focus areas of the author in the field of computer science. Co Authors details will also help recruiters because at this place recruiters may get information about Authors related in that field of study. Top journal and conferences details will help to check whether the specific author selected has published articles in major journals and conference or low impact factor journals. Best author will have more citation, better affiliations and papers in high impact factor journals and conferences.

## 6. Conclusions

To conclude, this research paper has provided a portable hybrid author recommender platform utilized by my application, a user content-based and co-citation recommendations in optimizing freshness, user satisfaction, scalability, and coverage. I have assessed the perfection of the result by the system through the user study and demonstrated the correlation between the scores and the system computed similarities for couple of research paper recommendations. Eventually, I

have collated the user study results together with the real recommendation lists used by Microsoft Academic made available to the full-time researchers to conduct analysis and assist future research in research author and paper recommender systems. In this research, I have used quantitative measures to establish which algorithms are used in various business intelligence systems and how effective they are in supporting managers' decision-making. Face-to-face interviews will be conducted and questionnaires to managers of different organizations to ascertain their satisfaction and the general performance of the BI systems.

This research studied, compared, and analyzed different algorithms and their effectiveness in the decision-making process. I propose to use a hybrid recommender system that utilizes both content-based (CB)s and co-citation-based (CcB) to generate a search of items. CcB is computationally time-consuming and generates low-quality recommendations, whereas CB uses metadata, keywords, articles, and abstracts to create fresh content and huge coverage. CcB uses reference information in a paper to make a recommendation. This empirically resembles human behavior when it's searching for related documents. The reference feature of CcB is a major drawback considering most old papers were published in hard copies, and their reference is not available digitally.

On the other hand, CB resolves some of the inefficiencies of CcB, *i.e.*, complexity issues, privacy concerns, and cold start problems. CB uses information available on the web pages, such as metadata, keywords, title, and abstract. Cluster paper embedding and generating paper embedding shall be used to determine the relevance of a paper in the search domains in larger data sets [15]. This augments well with my research that seeks to provide a scalable recommender system that can utilize content-based and co-citation-based to improve on quality and hierarchical clustering based on relevance. Combining both CB and CcB generates unified search items by ordering related papers from both lists based on congruency to the document in question.

### **Acknowledgements**

We researchers would like to thank Microsoft for providing big data tools to conduct extensive research related to the implementation of the Author Recommendation System. We want to express our sincere appreciation and our deepest gratitude to Respective Authors University faculty members for providing guidance in research and writing papers. We also thank the anonymous referee, reviewers, and editors for reviewing our paper. Finally, we sincerely thank the Journal of Data Analysis and Information Processing for allowing us to publish the paper.

### **Conflicts of Interest**

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Olszak, C.M. (2016) Toward Better Understanding and Use of Business Intelligence in Organizations. *Information Systems Management*, **33**, 105-123. <https://doi.org/10.1080/10580530.2016.1155946>
- [2] Visinescu, L.L., Jones, M.C. and Sidorova, A. (2017) Improving Decision Quality: The Role of Business Intelligence. *Journal of Computer Information Systems*, **57**, 58-66. <https://doi.org/10.1080/08874417.2016.1181494>
- [3] Florescu, C. and Caragea, C. (2017, February) A Position-Biased PageRank Algorithm for Keyphrase Extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, **31**. <https://doi.org/10.1609/aaai.v31i1.11082>
- [4] Portugal, I., Alencar, P. and Cowan, D. (2018) The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review. *Expert Systems with Applications*, **97**, 205-227. <https://doi.org/10.1016/j.eswa.2017.12.020>
- [5] Arnott, D., Lizama, F. and Song, Y. (2017) Patterns of Business Intelligence Systems Use in Organizations. *Decision Support Systems*, **97**, 58-68. <https://doi.org/10.1016/j.dss.2017.03.005>
- [6] Gounder, M.S., Iyer, V.V. and Al Mazyad, A. (2016, March) A Survey on Business Intelligence Tools for University Dashboard Development. 2016 *3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, Muscat, 15-16 March 2016, 1-7. <https://doi.org/10.1109/ICBDSC.2016.7460347>
- [7] Thalhammer, A., Lasier, N. and Rettinger, A. (2016, June) LinkSUM: Using Link Analysis to Summarize Entity Data. In: Bozzon, A., Cudre-Maroux, P. and Pautasso, C., Eds., *ICWE 2016: Web Engineering*, Springer, Cham, 244-261. [https://doi.org/10.1007/978-3-319-38791-8\\_14](https://doi.org/10.1007/978-3-319-38791-8_14)
- [8] Tvrdikova, M. (2007) Support of Decision-Making by Business Intelligence Tools. *6th International Conference on Computer Information Systems and Industrial Management Applications (CISIM07)*, Elk, 28-30 June 2007, 364-368.
- [9] Kasemsap, K. (2016) The Fundamentals of Business Intelligence. *International Journal of Organizational and Collective Intelligence (IJOICI)*, **6**, 12-25. <https://doi.org/10.4018/IJOICI.2016040102>
- [10] Oussous, A., Benjelloun, F.Z., Lahcen, A.A. and Belfkih, S. (2018) Big Data Technologies: A Survey. *Journal of King Saud University-Computer and Information Sciences*, **30**, 431-448. <https://doi.org/10.1016/j.jksuci.2017.06.001>
- [11] Bathrinath (2019) PageRank Algorithm-Based Recommender System Using Uniformly Average Rating Matrix. <https://doi.org/10.4018/978-1-5225-5445-5.ch006>  
<https://www.igi-global.com/chapter/pagerank-algorithm-based-recommender-system-using-uniformly-average-rating-matrix/216694>
- [12] Kanakia, A., Shen, Z., Eide, D. and Wang, K. (2019) A Scalable Hybrid Research Paper Recommender System for Microsoft Academic. *WWW'19: The World Wide Web Conference*, 2893-2899. <https://doi.org/10.1145/3308558.3313700>
- [13] Arnab, S., Zhihong, S., Yang, S., Hao, M., Darrin, E., Bo-June (Paul), H. and Kuan-san, W. (2015) An Overview of Microsoft Academic Service (MAS) and Applications. *Proceedings of the 24th International Conference on World Wide Web (WWW'15 Companion)*. ACM, Florence, 18 May 2015, 243-246.
- [14] Wang, K., Shen, Z., Huang, C., Wu, C.H., Eide, D., Dong, Y., Qian, J., Kanakia, A., Chen, A. and Rogahn, R. (2019) A Review of Microsoft Academic Services for Science of Science Studies. *Front Big Data*, **2**, 45. <https://doi.org/10.3389/fdata.2019.00045>
- [15] Xing, W. and Ghorbani, A. (2018, May) It Weighted the PageRank Algorithm.

*Second Annual Conference on Communication Networks and Services Research*,  
2004, Fredericton, NB, 21-21 May 2004, 305-314.  
<https://doi.org/10.1109/DNSR.2004.1344743>