BIP! Finder: Facilitating Scientific Literature Search by Exploiting Impact-Based Ranking

Thanasis Vergoulis
vergoulis@imis.athena-innovation.gr
IMSI, "Athena" RC

Serafeim Chatzopoulos
schatz@imis.athena-innovation.gr
Univ. of the Peloponnese & IMSI, "Athena" RC

Ilias Kanellos
ilias.kanellos@imis.athena-innovation.gr
IMSI, "Athena" RC & NTU of Athens

Panagiotis Deligiannis
cst11017@uop.gr
Univ. of the Peloponnese

Christos Tryfonopoulos
trifon@uop.gr
Univ. of the Peloponnese

Theodore Dalamagas
dalamag@imis.athena-innovation.gr
IMSI, "Athena" RC

ABSTRACT

Due to the rapidly increasing number of scientific articles, finding valuable work for further research has become tedious and time consuming. To alleviate this issue, search engines have used citation-based article impact ranking. However, most engines rely on very simplistic impact measures (usually the citation count) and make the problematic assumption that there is a one-size-fits-all impact measure. To address these problems, we present BIP! Finder, a search engine that facilitates the identification of valuable articles by exploiting two different impact measures, each capturing a different aspect of the article impact. In addition, BIP! Finder provides many useful features (article comparison, intuitive visualisations, article bookmarking mechanism, etc.) making it a powerful addition to the researcher's toolbox.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Retrieval models and ranking**.

KEYWORDS

scientific impact, citation networks, ranking, search engines

ACM Reference Format:
Thanasis Vergoulis, Serafeim Chatzopoulos, Ilias Kanellos, Panagiotis Deligiannis, Christos Tryfonopoulos, and Theodore Dalamagas. 2019. BIP! Finder: Facilitating Scientific Literature Search by Exploiting Impact-Based Ranking. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3357850

1 INTRODUCTION

In the last decades, the growth rate of scientific articles has been increasing, a trend that is expected to continue [7]. This is not only due to the increase in the number of researchers worldwide, but also to the growing competition that pressures them to continuously produce publishable results, a trend widely known as "publish or perish" [3]. This trend has also been correlated with a significant drop in the average quality of scientific articles [8]. As a result, identifying valuable articles relevant to a particular research topic, a task that dominates the researchers' daily routine, has become extremely tedious and time consuming.

Quantifying and measuring the impact of scientific articles could facilitate the above task. The impact, combined with keyword-based relevance, can be used to implement ranking schemes beyond the traditional content-based ranking (i.e., ranking articles based on the similarity to the user-provided query). Most contemporary search engines for scientific articles (e.g., Google Scholar, CiteSeer^x) follow this approach to help researchers prioritise their reading.

However, providing a valid measure of impact is not a trivial task. For example, citation counts, on which most search engines rely, have serious drawbacks, such as not differentiating citations based on the importance of the articles making them. Therefore citation counts are vulnerable to malpractices, e.g., excessive self-citation, or may present articles of predatory journals, which are heavily cited by other trivial works, as invaluable[1].

Another important issue of existing search engines, is their assumption that there is a single, one-size-fits-all article impact measure. This is an oversimplification, since there are at least two different aspects in the impact of an article: its *influence*, i.e., its general, long-term importance for a discipline and its *popularity*, i.e., its impact in the short term (its hype right now). These impact aspects are not completely correlated and different researchers may prefer the one over the other based on their needs. For instance, consider Penny, an experienced researcher who needs to revisit a topic of interest to learn about its latest developments. Ranking articles based on their popularity would be preferable for her. On the other hand, Corto, a new researcher wanting to delve into the same topic to prepare a survey, would prefer to rank the relevant articles based on their influence. Although established impact measures, employed by current search engines, would satisfy Corto's needs to an extent, they would fail to help Penny, since they are biased against recent articles [2]. This is because any recent article (irrespective of its current attention in the research community) usually requires months or even years to receive its first citations [5].

We introduce BIP! Finder[2] (BIbliograPhy Finder), a scientific article search engine that addresses the aforementioned issues. It

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11.
https://doi.org/10.1145/3357384.3357850

[1] The reverse phenomenon can be also observed: Sometimes, the impact of sparsely cited articles that have influenced breakthrough research is disregarded.
[2] https://bip.imsi.athenarc.gr

is built on top of a very large, interdisciplinary dataset containing more than 45M articles and more than 447M citations. Our main contributions follow:

- We support ranking based on combinations of keyword relevance and influence/popularity, additionally providing multiple search filters. We use PageRank [6] (which differentiates between citations based on the paper making them), and TAR-RAM [4] (which alleviates bias against recent papers) as impact measures for influence and popularity, respectively.
- We provide scalable, open source implementations of PageRank and TAR-RAM, as well as access to all calculated paper impact scores through an API, to encourage third party development of additional services adding value for the market of research analytics.
- We provide a number of visualisations that provide insights into each article's characteristics (impact, latent topics etc), as well as functionalities such as bookmarking mechanisms.

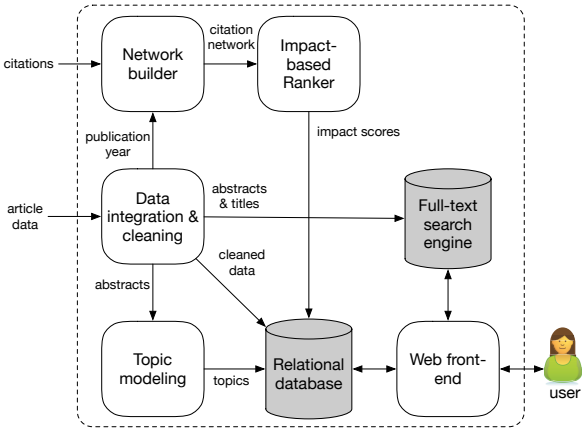

Figure 1: BIP! Finder's Architecture.

2 SYSTEM OVERVIEW

2.1 BIP! Finder's Architecture

To support all BIP! Finder's functionalities, a set of software components have been developed. Figure 1 summarises BIP! Finder's architecture, illustrating these software components and the data flow between them. The following paragraphs discuss the functionality and the implementation details of each software component.

Network builder. This component is responsible to build the underlying citation network. Its input is the latest version of the OpenCitations COCI dataset[3], which contains almost 450M citations for more than 45M articles. This dataset has been created by the I4OC[4] initiative and it is updated on a regular basis. Each time a new version of the dataset is released, BIP! Finder's database is also updated. It should be noted that COCI contains only DOI-to-DOI relations, thus, extra data required by some impact-based ranking algorithms (e.g., the article's publication year needed by TAR-RAM [4]) have to be fetched from the data integration & cleaning component.

Data integration & cleaning component. This component collects and integrates research article data (e.g., titles, abstracts, author lists, venues, publication dates) from multiple sources. Currently, BIP! Finder collects data from the Crossref REST API[5] and the Open Academic Graph[6] [9, 10]. Due to the use of multiple sources, the collected data may contain inconsistencies or redundancies (e.g., different names for the same venue). This is why extensive data integration and cleaning must take place. E.g., the cleaning of venue names includes, among others, the following: removing redundant white space characters, enforcing particular capitalisation rules, handling common variations (e.g., removing the numbering from the conference name or replacing special characters with equivalent words), etc. Similar tasks are also used for the cleaning of other article data, such as author names. The output of this component is stored in BIP! Finder's relational database and is used by the Web front-end to produce most of the dynamic content displayed to the user. Moreover, parts of the output are propagated to other software components for further processing.

Impact-based ranker. This component implements the impact-based article ranking algorithms on which many of BIP! Finder's features rely. In particular, PageRank [6] was selected to capture the influence of the articles, since it differentiates citations based on the importance of articles making them. However, since PageRank relies on the current centrality of each article in the citation network, it is inappropriate to estimate popularity (see also Section 1). For this case, we selected TAR-RAM [4], which is suitable, since it is based on the idea that recent citations are more important than older ones and promotes articles gaining citations recently, alleviating the bias against recent publications. We have implemented both algorithms as MapReduce scripts. Our implementations[7] are scalable and open source (under a GNU/GPL license). For each update of the underlying citation network, we recalculate PageRank and TAR-RAM scores for all articles by executing our implementations on a Hadoop cluster of 10 VMs, each with 4 cores and 8GB RAM.

Topic modeling component. This component takes as input the article abstracts and trains an LDA [1] model. The gensim[8] topic modelling library was used to train a model for 500 topics. Then, for each article, the 3 most relevant topics to its abstract were identified and stored in BIP! Finder's relational database.

Web front-end, data storage & indexing. BIP! Finder's Web UI was implemented using PHP under the MVC architecture. All visualisations were implemented using a combination of CSS and JavaScript, also exploiting third-party libraries (e.g., the *D3.js* library). All data are stored and indexed in a relational database. In addition, titles and abstracts are indexed in an Apache Solr[9] full-text search engine running on a 3 VM cluster (8 cores & 16GB RAM per node).

2.2 BIP! Finder's features

2.2.1 Searching for articles. A powerful search engine, based on user-provided keywords, lies at the heart of BIP! Finder. The user

[3] http://opencitations.net/download
[4] https://i4oc.org/
[5] https://www.crossref.org/services/metadata-delivery/rest-api/
[6] https://www.openacademic.ai/oag/
[7] https://github.com/diwis/PaperRanking
[8] https://radimrehurek.com/gensim/
[9] http://lucene.apache.org/solr/

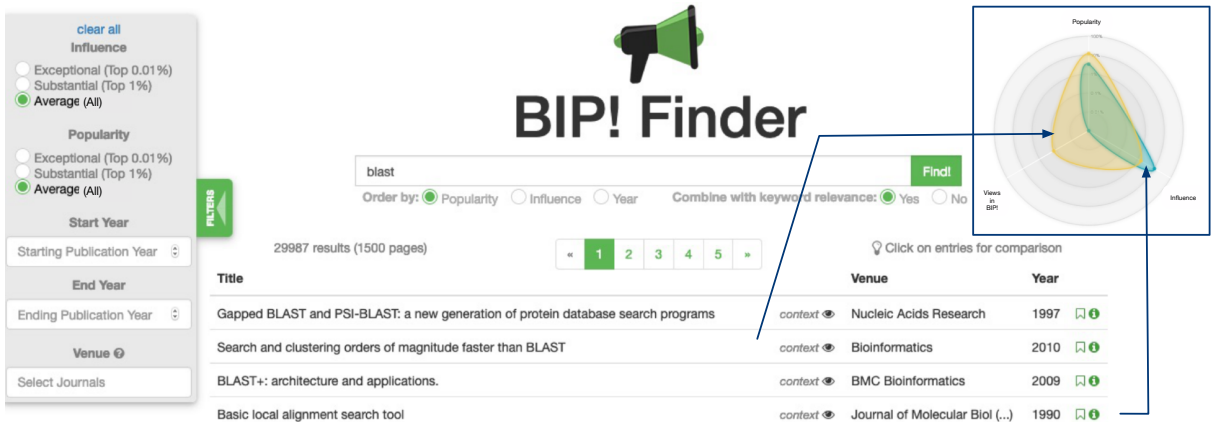

Figure 2: Screenshot from BIP! Finder's user interface.

provides the desired keywords in BIP! Finder's search box and, after clicking the "Find" button, all relevant articles appear at the bottom of the page (see Figure 2). The great power of this engine is that it takes advantage of each article's impact in two ways: (a) by supporting ranking based on a combination of popularity/influence with traditional keyword relevance, and (b) by supporting filtering the results based on predefined levels of popularity/influence. The search engine also provides a traditional ranking option based on the publication year of the articles.

As regards ranking options, users can select (a) their preferred ranking criterion and (b) whether they want to combine the desired ranking criterion with keyword relevance scores, by clicking on the corresponding radio buttons below the search box. Regarding the impact-based filtering, users can exclude low popularity/influence search results by configuring the corresponding options in the sidebar at the left of the interface. It should be noted that, apart from the impact-based filters, BIP! Finder also provides some other filtering options (based on publication year ranges and venues).

2.2.2 Article comparison. BIP! Finder users can select a group of articles for comparison. To add a particular article to the comparison list the user has to click on the corresponding row of the search results. After selecting at least two articles, the user clicks on the "Compare" button rendering a new page in a separate browser tab. The page contains the list of articles to be compared and displays a radar chart that summarises their popularity, influence, and number of views in BIP! Finder (see Figure 2). Finally, instead of displaying the radar chart, the user can also select to see an alternative infographic comparing the citations per year for each article.

2.2.3 Article infographics. Users that want to learn more about any particular article, can click on the information button located at the right of the article's entry in the search results (Figure 2). A new page, containing useful article metadata will appear in a new tab. It also contains some useful infographics (see Figure 3):

Impact pyramids. This infographic provides an intuition on the article's popularity and influence in comparison to the popularity and influence of (a) the rest of the articles in BIP! Finder's database (the first two pyramids) and (b) the rest of the articles published in the same venue (the last two pyramids). In each pyramid the article is classified as having exceptional (top 0.01%), substantial (top 1%), or average impact (the rest).

Citation history plot. This plot shows the number of citations the article received per year. To provide further insight into its yearly citation history, the plot also displays the citation history of two artificially created articles, one of exceptional and another of substantial influence.

Topics visualisation. This infographic shows the article's 3 most relevant topics. Article topics were defined by applying LDA [1] on their abstracts (see also Section 2.1). The extent to which each topic participates in the article's abstract is visualised with a pie chart, while each topic is represented by a tag cloud containing a stemmed version of its most typical terms.

2.2.4 Article bookmarks. A logged-in user can bookmark an interesting article by clicking on the bookmark-shaped icon that appears at the right of each search result (see Figure 2). The user can browse her created bookmarks by clicking on the corresponding BIP! Finder menu item. To facilitate the management of bookmarks, BIP! Finder supports organising them into user-defined folders.

2.2.5 API for impact scores. We have developed an API that provides access to all calculated impact scores for the scientific articles that are included in BIP! Finder's database. This interface is freely available[10] and was developed based on the Microservices Architecture as an independent Node.js application running in a docker container. Making, for the first time, article impact scores easily available for programmatic access, we encourage third-party developers to extend the researcher's toolbox by building useful services on top of these data. This way, an added value for the market of research analytics platforms will also be created.

3 DEMONSTRATION

At the conference, we will explain the concepts of article popularity and influence to the audience and we will demonstrate BIP! Finder's functionality showing its benefits in comparison to other academic search engines (e.g., Google Scholar). We will let the members of the audience interact with BIP! Finder's interface, to give them the

[10] http://bip.imsi.athenarc.gr:4000/documentation

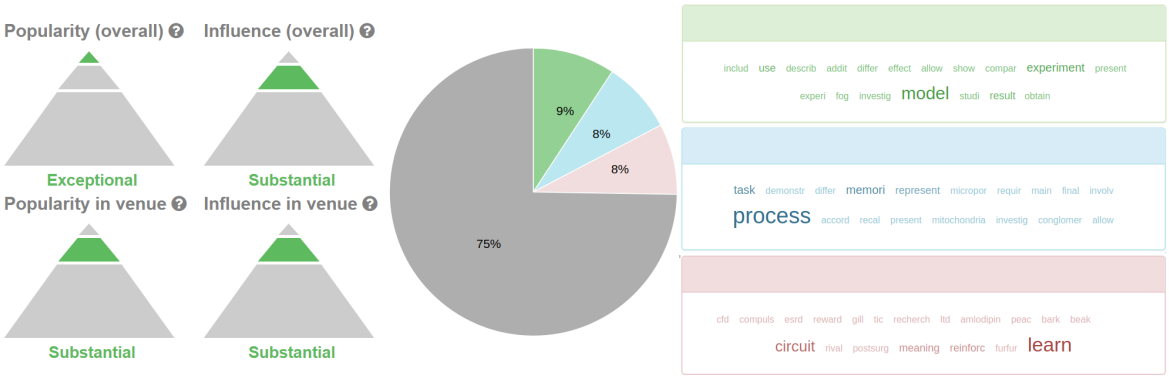

Figure 3: Examples of article infographics (left: impact pyramids, right: topics visualisation)

opportunity to examine its capabilities in real-world scenarios that are relevant to their personal research interests. However, we will also demonstrate some interesting scenarios that we have identified. Short descriptions for two such indicative scenarios follow.

Scenario 1 (search): A member of the audience inserts "string matching" (with quotes) to the search box and selects to rank the results based solely on influence (i.e., keeping keyword relevance disabled). She also applies a filter to avoid very old papers (e.g., Start Year = 1998). Her intention is to search for works that summarise exact and approximate string matching algorithms. After identifying some important works in the results (e.g., the survey titled "A guided tour to approximate string matching"), she selects, instead, to rank results by popularity (without changing anything else). To her surprise, the first page of the displayed results now includes an extra, recent survey titled "The exact online string matching problem: A review of the most recent results" that is very relevant. Toggling to the "ranking by year" option, this work does not appear in the first page of results in this case, either. The audience member will be further prompted to contrast the results with those of another well-known academic search engine, applying the same year-based filter. The highly related, recent survey article, only appears in the results after the fourth page.

Scenario 2 (comparison): A member of the audience searches for articles relevant to the query "text mining" (with quotes). Initially, she selects to rank results by combining popularity with keyword relevance. In the first page of results, she identifies an article titled "Mining Text Data" and she adds it to the comparison list. Then, she toggles the ordering option, choosing influence over popularity (without changing anything else). In the new list she identifies "The text mining handbook", an interesting article, which was not among the top results of the previous search. She includes it to the comparison list and clicks on the "Compare" button. On the radar chart of the comparison page the user can examine the differences of the two articles based on different impact aspects. The relatively small popularity of the second paper clarifies why the initial search excluded it from the first page of results. Finally, toggling to the citation history plot, the user reveals that while the older article has had a longer citation history, the more recent one has received more citations recently.

4 CONCLUSION

We demonstrated BIP! Finder, a system that facilitates identifying scientific articles with notable impact. Its powerful ranking mechanism, exploiting two different impact-based algorithms, alleviates important problems of ranking mechanisms employed by other current academic search engines. Furthermore, it provides many other useful features like multidimensional comparison of articles (based on impact scores and other data), intuitive infographics that provide useful insights about an article's characteristics, and support for the creation and management of user-defined bookmarks.

ACKNOWLEDGMENTS

We acknowledge support of this work by the project "Moving from Big Data Management to Data Science" (MIS 5002437/3) which is implemented under the Action "Reinforcement of the Research and Innovation Infrastructure", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
[2] Peng Chen, Huafeng Xie, Sergei Maslov, and Sidney Redner. 2007. Finding Scientific Gems with Google's PageRank Algorithm. *Journal of Informetrics* 1, 1 (2007), 8–15.
[3] Daniele Fanelli. 2010. Do Pressures to Publish Increase Scientists' Bias? An Empirical Support from US States Data. *PLOS ONE* 5, 4 (2010), 1–7. https://doi.org/10.1371/journal.pone.0010271
[4] Rumi Ghosh, Tsung-Ting Kuo, Chun-Nan Hsu, Shou-De Lin, and Kristina Lerman. 2011. Time-Aware Ranking in Dynamic Citation Networks. In *Data Mining Workshops (ICDMW)*. 373–380.
[5] Paul Groth and Thomas Gurney. 2010. Studying Scientific Discourse on the Web Using Bibliometrics: A Chemistry Blogging Case Study. In *WebSci*.
[6] R. Motwani L. Page, S. Brin and T. Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab.
[7] Peder Olesen Larsen and Markus von Ins. 2010. The Rate of Growth in Scientific Publication and the Decline in Coverage Provided by Science Citation Index. *Scientometrics* 84, 3 (2010), 575–603. https://doi.org/10.1007/s11192-010-0202-z
[8] D. Sarewitz. 2016. The Pressure to Publish Pushes Down Quality. *Nature* 533, 7602 (2016), 147. https://doi.org/10.1038/533147a
[9] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In *WWW (Companion Volume)*. 243–246.
[10] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *SIGKDD*. 990–998.