Improving Scientific Search Engine Interfaces Using Scientific Networks

Joep Schyns

Abstract

Scientific networks make it possible to visualise the structure of research fields, how (and how closely) different topics are connected, and enable spotting of emergent research. These networks have the potential to improve scientific search. However, although large amounts of literature focus on prescribing how scientific search engines should be used, there is little research performed on how they are used in practice. This makes it challenging to assess the added value of scientific networks to that of scientific search engines, even though interviews with expert researchers indicate their potential usefulness. This work aimed to address this question with a user study, using a prototype search engine. Where is manipulated how search results are displayed.

Keywords: Scientific Networks, Scientific Search Engine Interfaces, Prototype

1. Introduction

As more literature becomes better available, the importance of scientific search engines increases. That is, search engines help researchers to find existing studies that are relevant for their research. However, due to the ever growing amount of available literature, even with search engines, it can become difficult to find and identify research that is relevant. As a consequence, the methods used to search for literature continuously have to be improved.

This research explores how the interfaces of existing scientific search engines can be enhanced. The study is divided into three parts. First of all, scientific networks were studied in literature for their commonalities with scientific search engine interfaces. This parallel included the creation of user interfaces to analyse and explore scientific data. In the literature study the different types, benefits and disadvantages of scientific networks, and analytic toolsets were investigated.

The second part of the study moves on from most available literature. This is done by drafting an envisioned solution to improve scientific search engine interfaces based on the first part of this study. Hereupon, an initial exploration of issues related to the envisioned solution was performed. This exploration included a theoretical framework and interviews with experts.

In the third and last part, a prototype was built based on the envisioned solution discussed in the second part of the study. Using an iterative development process, a recording was made of deciding moments of the build. As no earlier examples were found of similar initiatives, the prototype is build from the ground up, while using as many open-source software libraries wherever applicable. Finally, the prototype was evaluated in a small user test.

2. Scientific Networks

Scientific Networks are networks made up of scientific matter and interconnected by relations. In the literature review of Schyns [1], it was found that these networks are praised for their ability to learn how research is build up [2], to show how different topics are connected and how close they are connected [3], and spot clusters of emergent research [4].

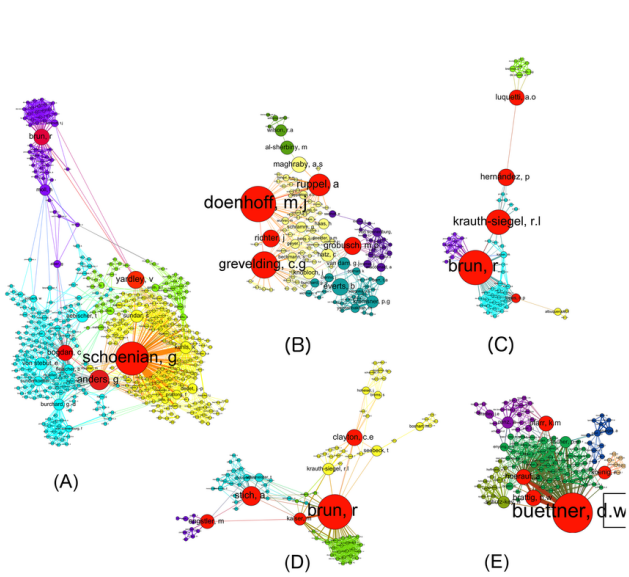
Three main types of scientific networks were found to be explored the most, namely co-author, citation and topic networks, see examples in figure 1. First of all, The nodes of a co-author network represent authors and the edges represent acquaintance or co-authorship. These networks are found to be useful to study clustered and close relations between authors within the same research field, especially in highly specialised fields [5][6][7]. Topic networks are primarily used to investigate scientific relations without being influenced by the social structure of the scientific community [8][3][9]. In these networks nodes represent topics of study and edges give the degree to which topics occur simultaneously. Finally, citation networks offer a directional structure and are often used in combination with clustering algorithms to extract topic fields [2]. In these citation networks papers are represented as nodes and citations as edges. However, also hybrid variants of the three main types where found [10]

Additionally, the main factor restraining scientific network research was established to be the lack of open access bibliometric datasets. These datasets have great economic value [11]. Therefore, full public access to complete datasets is rare [12]. However, change was found to be on the horizon. That is, both governmental [12], non-profit [13][14][15] and commercial [16][17] initiatives start to open-up bibliometric resources.

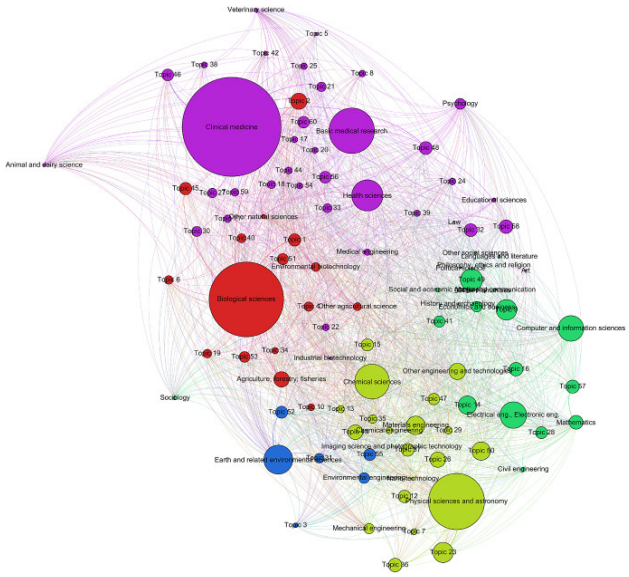
Moreover, various analytic toolsets have been created by the scientific community [18] in the past years to make use of the benefits of scientific networks. The tools implement among other methods for data processing [19][20], relation extraction [21], similarity measures [22][23][24][25] and filtering [4][26][27][28][20].

However, research of Schyns [1] discovered that no definitive answer on how

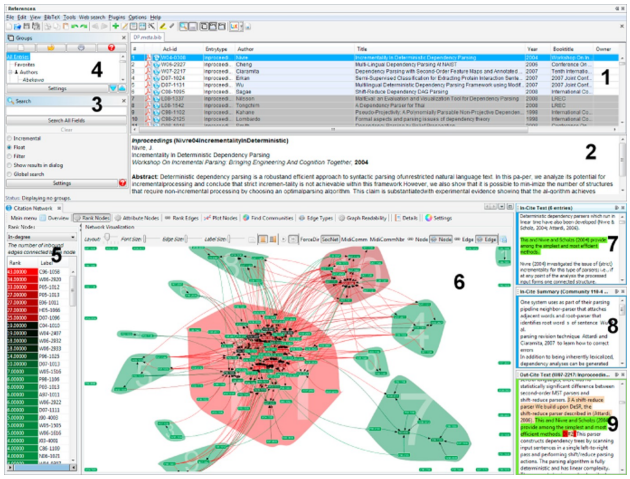
to analyse scientific networks exists. Not only are various types of (sub-) networks used, also the techniques to process these networks are not agreed upon. The only consensus was noted to be the requirement of a reduction of information in network visualisations. Due to the diversity, analytic tools where found to include many options for users to choose from. Thereby complicating user interfaces, making them not user-friendly and not usable by researchers outside of the field itself. Attempts to overcome this complexity where also encountered [13][29]. In order to reduce complexity the approaches conform in using web-based solutions with build-in datasets and cut down user interfaces. Hereby is tried to reduce the steps users need to take in order to use the applications. However, these attempts were found to be in early stages. The tools are not yet available and/or are undergoing development.



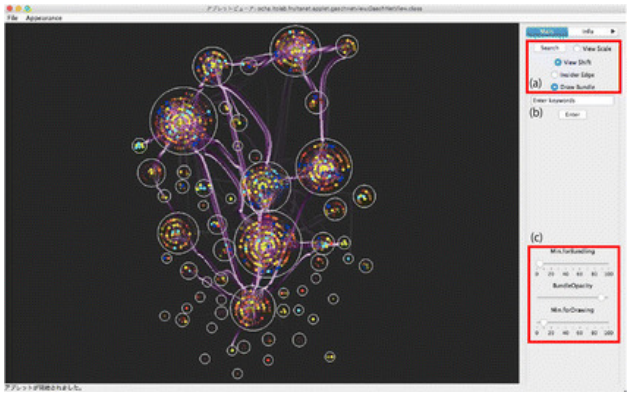
(a) Example of a co-author network



(b) Example of a topic network



(c) Example of a citation network



(d) Example of a hybrid network

Figure 1: Main types of scientific networks

3. Preliminary Study

In the preliminary study part of this research, scientific network literature, discussed in the previous section, is applied to scientific search engine interfaces. Hereby an vision is created that attempts to improve within two established areas of improvement. Thereupon, an initial exploration of issues investigates feasibility and applicability of the envisioned solution by interviewing expert researchers.

3.1. Areas For Improvement

By defining two areas of improvement in current scientific search engines a starting point is created for this study to improve scientific search engines.

3.1.1. AFI.1

To begin with, current scientific search engines show results in ranked lists. Thereby engines indicate the relevance of search results to a search query. However, researchers are presumably not after the relevance of a search query. More likely, researchers seek the relevance of search results to a research question. The title and a small text fragment of each search result provide insight into the sought after relevance, see figure 2. These two texts are the only methods researchers have to their disposal to asses whether it is worth reading said papers. As the texts are minimal a wrong relevance estimation is easily made. Therefore, this research defines the first area for improvement as:

AFI.1 Search results should indicate their relevance in more ways than text fragments only.

Figure 2: Example of methods to determine relevance to a research question

3.1.2. AFI.2

Literature research using scientific search engines is currently an iterative process of (1) defining a query, (2) reading search results and (3) altering the query based on results [30][31][32]. This process is repeated until literature is found that meets expectations. Whenever this process does not deliver results research is stranded. No other methods that can help explore literature are currently available in search engines. Therefore, this research defines the second area for improvement as:

AFI.2 A variety of search strategies, possibly a combination of, should come in place for the limited keyword based iterative strategy progress.

3.2. Research questions

Following the identified areas for improvement, the following research questions are drafted.

RQ.1 How to improve search for scientific publications by combining existing search engines with methods developed for scientific networks?

And the following sub-questions.

RQ.1.1 How to use scientific network visualisations to indicate the relevance of search results?

RQ.1.2 How to use visualisations to support the process of exploration for new relevant literature?

3.3. Envisioned Solution

Based on the scientific networks studied in a previous study [1], a vision to improve scientific search engines within the two areas of improvement, as defined in the previous paragraph, is drafted. This vision consist of four main parts, as can be seen in the list below. The following paragraphs will elaborate on the list.

- Simple to use user interface

- Clear, easy-to-read visualisations
- Allowing in-depth research while keeping a clear overview
- Enable recommendation of papers based on previous activity

3.3.1. Simple to use user-interface

As scientific search engines provide a portal for all types of researchers, scientific search engines should not be difficult to use. In a previous study [1], current scientific networks tools were found to be difficult to use and to interpret. Mostly due to the immense amount of options and lack of included data. Hence, the networks were considers useless by researchers from other fields of study than scientific networks itself. In order to be able to use scientific networks in scientific search engines, the way scientific networks are implemented should be changed. This paper envisions a scientific network search engine that reduces complicity by pre-loading data and without many options for different (sub)types of networks. A simple query interface should be the only required user input. An example can be seen in figure 3

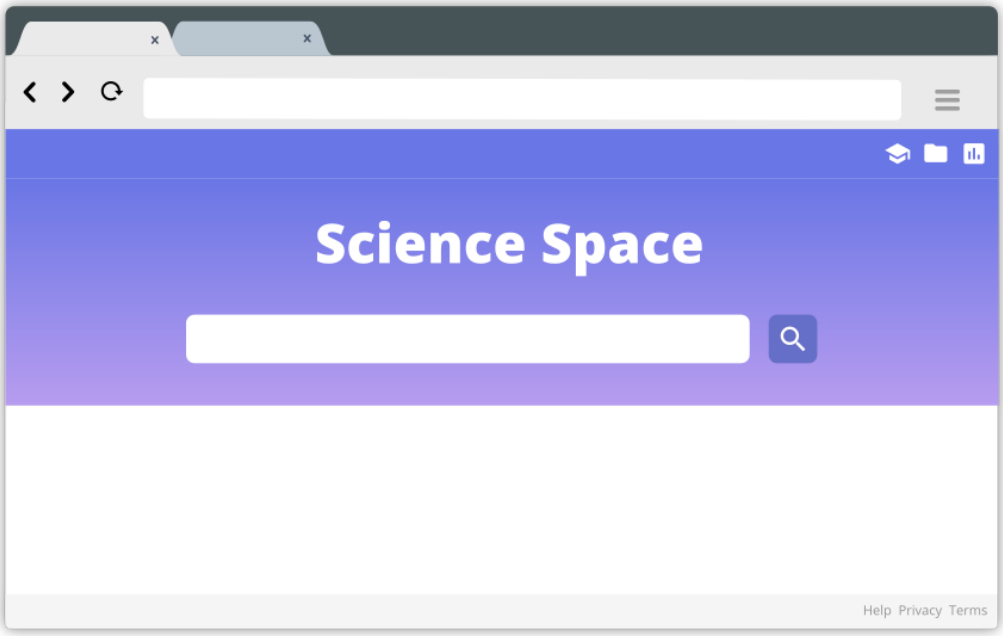


Figure 3: Mock-up of a simple search interface

3.3.2. Clear, easy-to-read visualisation

Besides result lists, scientific search engines should use scientific networks, as they provide more insights in trends and relations, see figure 4. In scientific network literature networks are created where links and nodes display relations between various matters, such as topics, authors and papers. With these visualisations, analysts are able to spot trends and relations in scientific knowledge. These trends and relations could also be used in scientific search engines. For example, by showing how the results of a query relate to each other, or by showing how results are connected to other topics and authors.

3.3.2.1. Clustering. Moreover, various clustering techniques should be applied to search result visualisations as in literature it quickly became apparent that it is impossible to show all relevant matter. For example, a citation network

scales exponentially as more citations are added and thereby becomes unreadable fast. Therefore, scientific networks implement various methods to reduce the number of links and nodes in the networks. Similar techniques should also be used in networks created by the results of search engines. This research envisions network visualisations that extensively merges networks into sub-networks. Preferably with a short explanation defining the clustered together content. By displaying a limited amount of nodes and relations, the interface of the search engine should be kept clear, see figure 4.

3.3.2.2. Filtering. Besides clustering, also the filtering techniques applied in scientific networks should be adopted to promote network clarity. That is, filtering removes content from networks that have had minor impact on the scientific world while bringing out others that have done the contrary. By using filtering techniques the density of relevant information within networks can be increased. Various types of filtering techniques exist, each with their own set of rules determining what is scientifically important and what is not. In order to make the use of filtering techniques as transparent as possible, this paper envisions that results adjusted by a certain filter should be annotated by this filter. Through these annotations the reason for alteration can be transferred transparently to the user, see figure 4, 5 and 6.

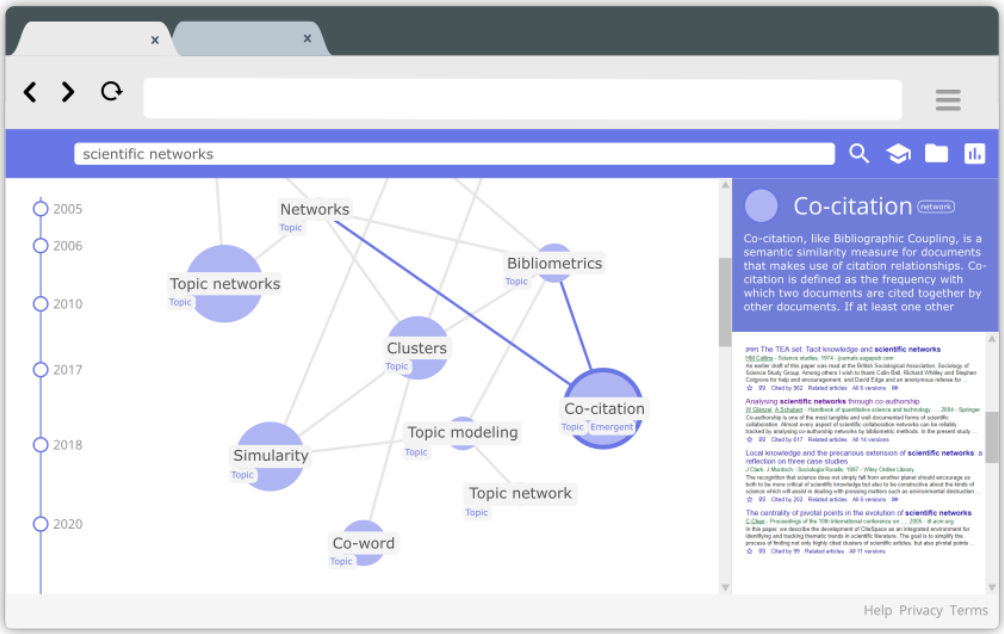


Figure 4: Mock-up of a search interface with merged networks

3.3.3. Allowing in-depth research while keeping a clear overview

Additionally, a middle ground between simplicity and detail should be found in the interface of a scientific search engine. As literature research involves thorough in-depth analysis of knowledge, a scientific search engine should support thorough search and enable the user to exhaust the available information. Additionally, as mentioned in the previous paragraphs, scientific search engines should be simple to understand. This simplicity can stand in contrast with the in-depth analysis. In-depth analysis benefits from information in as great detail as possible, while simple to use interfaces benefit from displaying the least amount of detail. This research proposes forming as many clusters of similar matter as possible and enable researchers to expand these clusters. The networks remain clear due to the small set of initial information. At the same time

by allowing expanding it also supports in-depth analysis. An example of this expand-ability can be found in figure 5. In figure 5 the node selected in figure 4 is expanded. Thereby figure 5 shows the network wherefrom the selected node is built up.

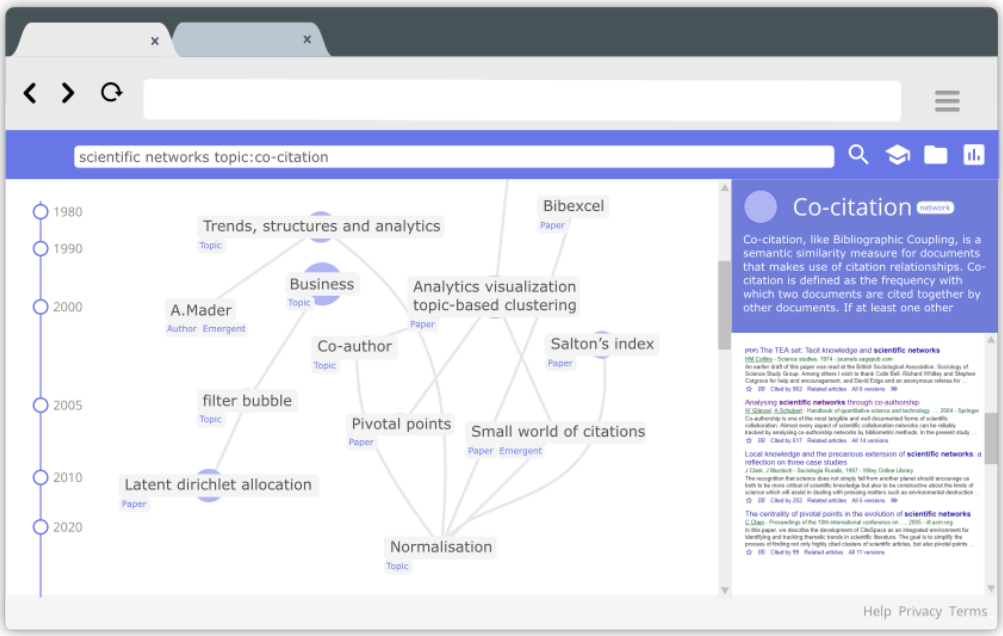


Figure 5: Mock-up of an expanded sub-network from a query

3.3.4. Enable recommendation of papers based on previous activity

In addition, in order to give researchers a better overview of literature during their studies, the prototype should include recommendation networks. Saved references of a literature study can act as nodes in a network and subsequently, similar topics, authors and papers can connect. Thereby, a researcher should be able to see if he or she is missing something and why this is relevant. For example, topics that discuss similar matter, or authors that are important in the field of research. An example of such a recommendation network can be

seen in figure 6.

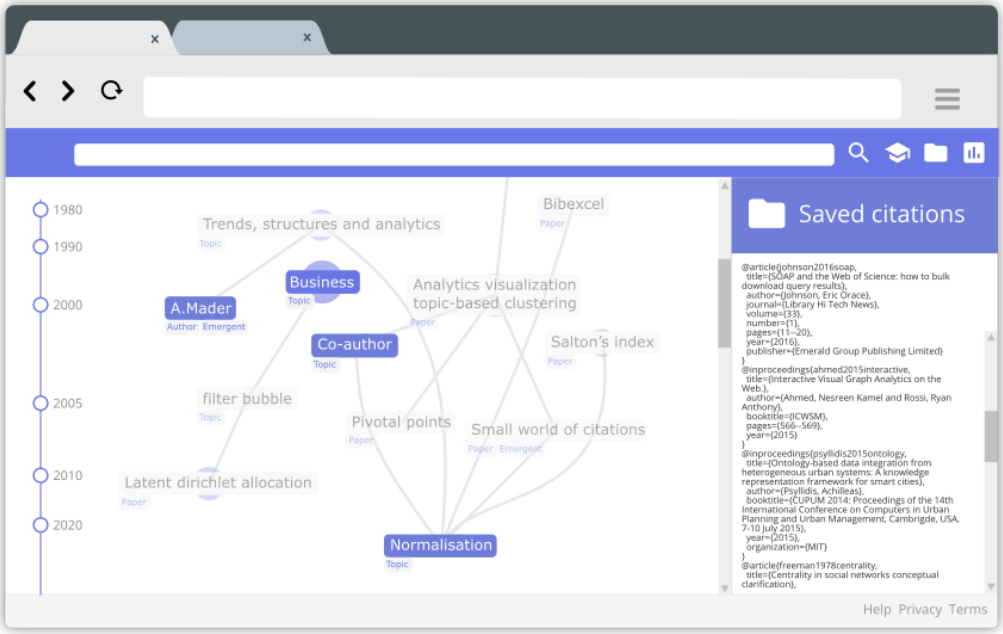


Figure 6: Mock-up of a recommendation network

3.4. Initial Exploration of Issues

Before a prototype based on the envisioned solution can be build, an initial exploration of issues was performed in a preliminary study. The study can be split into two parts.

Preliminary Study Question(PSQ) 1, see below, investigates whether the concepts envisioned meet the needs of users. In the interviews the experienced advantages, disadvantages and methods to overcome problems are studied.

Furthermore, PSQ.2 looks into data visualisations. PSQ.2 tries to answer how data visualisations, such as envisioned, can be created and what issues have to be taken into consideration. Each PSQ and sub-questions are studied both by interviewing expert researchers from various backgrounds and in literature.

PSQ.1 What kind of processes use expert researchers to find relevant works?

(a) Do researcher use search engines? What kinds?

(b) Do researchers use other means of search than search engines?

(c) How do researchers determine which papers are relevant?

(d) How do researchers come up with search queries?

(e) Do researchers think they overlook papers? How can researchers be enabled to find these overlooked works?

PSQ.2 Can scientific networks be used to add context in scientific search engines?

(a) How to display large networks in a clear way?

(b) How to reduce load times?

(c) How to interact with large networks?

(d) How to enable interactive in-dept analysis?

3.4.1. ***PSQ.1****What kind of processes use expert researchers to find relevant works?*

First of all, few literature studies examine the literature research processes and strategies used by expert researchers. Numerous papers are available that teach novice researchers how to conduct a literature review [33][30][31][32]. Likewise, plentiful, often cited, literature reviews are available. However, few literature describes how researchers adopted the methods explained to them as novices. Therefore, the adaptations experts practice for their own uses and preferences are unknown.

Literature on how experts use scientific search engines is absent as well. With insight into the use of search engines by experts, insight into the advantages and disadvantages of current scientific search engines could be gained. One of the few papers describing expert search processes dedicated to the field of health care [34]. In this research is found that researchers do not completely feel supported by current scientific search engines. Experts spend undesired large amounts of

time creating and re-evaluating search queries. The amount of obtained results is also found to be larger than ideal. Therefore, the time to evaluate each result gets reduced, as it is impossible to do an in-depth analysis of all results.

3.4.1.1. Interviews. In order to better understand the usage of scientific search engines by experts in practice, five researchers at the University of Twente were interviewed. These five researchers come from various backgrounds, including physics, chemistry, computer science, and electrical engineering. Moreover, they are employed as either PhD candidate or as professor. In the interviews, participants were asked to run through their current literature search process and elaborate on how this process changed during their career.

From the interviews struck that all participants described similar trends. Namely, as their career developed, their use of scientific search engines decreased. Participants described a growing memory of the important players in their field and an awareness of the new leading edge in literature. The importance of conferences was noted often. Through conferences, participants discovered new developments in their area of research. Thereby new knowledge gets added to their overview of the field. However, conferences were not the only method to discover new literature. Participants indicated that the discussion of research with colleagues and students also played a significant role in the discovery process. All participants pointed out that they use scientific search engines in some form, though not as often as at the beginning of their career. Participants described that, when they felt like they where missing something or felt like some subject was likely studied, they used specific keywords in combination with names of known researchers in the field to look whether new content was published. However, this process was often experienced as a time-consuming endeavour. Some participants indicated that they sometimes doubt search engines. Search engines were found to not always return the papers they thought where relevant, such as their own published papers.

3.4.1.2. Conclusions. From the interviews can be concluded that expert researchers are not the main target audience for the scientific search engine in-

terface developed in this research. Nonetheless, the methods expert researchers deploy for scientific search could be useful for developing this interface. All interviewed expert researchers have some type of network of relevant papers and authors in mind when performing their search. This network is used to position research, determine the relevance of new papers and explore their research field. The network is not only for personal use, non-experts or not yet experts make use of this network by being supervised by these experts too. The question arises, whether the envisioned scientific networks can be created to be similar to these "expert networks". When this similarity can be reached the improved scientific search could provide the benefits of expert networks to novice researchers.

3.4.2. ***PSQ.2*** *Can scientific networks be used to add context in scientific search engines?*

Data visualisations have been proved to be good tools to explain large amounts of data in various contexts [35]. Although scientific search engines are based on large amounts of data and try to give insights into this data, currently scientific search engines do not make use of data visualisations. In order to investigate how data visualisations could be used in scientific search engines, experts of the field of data science, data visualisations and publishers of scientific information were interviewed regarding the in section 3.3 proposed visualisation based scientific search engine interface.

3.4.2.1. Added value. The interviews disclosed that insights that can be attained from visualisations should focus on presenting more than what is already existing in search engines without visualisations. For example, a visualisation based on the results from a network-based search algorithm should not solely provide insights into its operation. Instead, the visualisation should add to a researchers ability to find relevant papers. In addition, the resulting networks are likely to be very complex as both the pure quantity of data and relations have proven to be expansive. Therefore, various techniques should be applied

to shrink the networks to an information-dense humanly interpretable visualisation.

3.4.2.2. Data acquisition. Scientific networks rely on bibliometric data, however, this data is not easily obtained and used. As examined in previous research [1] bibliometric data can be freely acquired from open access journals. Also, some commercial parties, such as Scopus and web of science, start to open up, enabling downloading of large sets of bibliometric data. However, by talking to researchers that tried to use these kinds of data, it was found that the issue with these (semi) open access journals is that they currently do not possess datasets that are as extensive as their commercial counterparts. Complete research fields are missing from the datasets. Therefore, in practice, their utility is limited. Moreover, non-open access sources do also provide limited access to their datasets. For example, Google Scholar shows search results through queries but does not allow access to the raw data. These limitations make it difficult to acquire enough data to build data visualisations.

Another problem with bibliometric data is that, even though the datasets may not be complete, these datasets consist of enormous amounts of data. A large amount of resources is required for searching through, and handling storage on, a local machine.

3.4.2.3. Data acquisition from non-open access sources. In discussion with an expert for a solution to the bibliometric data problem described in the previous paragraph, was suggested to access data from non-open sources "on the fly" instead of ahead of time. Although state of the art search engines do not allow to examine their complete datasets, small fragments can be requested through queries. Therefore, when bibliometric data is required, for example a network visualisation, it can be obtained from excising search engines using multiple targeted queries. For instance, snowballing sampling [36] could be used as a method to dynamically create targeted queries. Utilizing snowballing sampling, results from an initial search query can act as input for the snowballing. The result papers act as starting samples, from these samples parameters defined

in scientific networks, such as topics, authors and citations can be extracted. Consecutively, a chain of papers can be queried based on these parameters. Together these chains could form a bibliometric dataset wherefrom visualisation can be generated. The main disadvantage that was put forward of such an "on the fly" system is that no pre-processing can be done on the data. No clustering, network, or other algorithms can be ran beforehand. Also, the system will require more time to request data for a user since acquiring fragmented data through third-parties will involve more overhead.

3.4.3. Conclusions

In the preliminary study, a solution to improve scientific search engines using scientific networks methods is devised and an initial exploration of issues using expert interviews and literature is done. The envisioned search engine displays results both in ranked lists and scientific networks visualisations. These visualisations strive to offer more options to researchers to explore literature and more ways to discover the relevance of search results.

In the initial exploration of issues was found that few literature is available on the adaptation of taught literature research methods and use of scientific search engines in practice. This lack of literature makes it complicated to gain insight into the advantages and disadvantages of current search engines. Thus, from literature it is difficult to conclude whether the vision takes up the experienced issues in search. Only one paper [34] was found which studies scientific search engine usage and adaptation. This paper concludes that expert researchers deem the literature search process with current methods as a time consuming and not always a productive endeavour. Analogous conclusions can also be drawn from expert researcher interviews done in this research. From these conclusions can be presumed that change in scientific search would be appreciated. However, more research into the exacts is welcome.

In addition to the reported experience was found that expert researchers, over the years, create a form of scientific network in their heads. This network is used to position research, determine the relevance of new papers and explore

their research field. Often this network is used instead of search engines. The reported network shows similarities to the vision of this research to improve upon scientific search engine. Therefore, this similarity shows potential for the vision.

One problem was found for implementing the envisioned solution. Namely, the required bibliometric datasets are not easily obtained. Datasets are either closed source, or whenever datasets are available they are lacking large amounts of data. In both cases, the datasets are so large that the handling of resources is challenging. To circumnavigate the data issue a on the fly, data acquisition approach could be used. Instead of relying on beforehand obtained data, multiple targeted queries could request the required data from state of the art search engines. The disadvantage is that the data cannot be pre-processed and more overhead must be taken into account.

4. Prototype Development

Proceeding from the vision explored in the preliminary study, a prototype is built. Building a prototype contains various advantages. By building a prototype, can not only be evaluated whether scientific networks can convey relevance of search results (AFI.1) and aid in exploring literature (AFI.2), also feasibility can be incorporated in the research. In all probability, the concept will not be fully implemented as envisioned, practical limitations will determine the final outcome. The evaluation stages can be used to test whether practical adjustments have compromised the concept.

4.1. Iterations

In this section, the iterative development process of the prototype scientific search engine interface is described. Each of the sections describes a main deciding moment in the build as a small sub-study. For each iteration step, a problem description is given, thereafter a solution is proposed, and a conclusion is drawn on the success of the proposal. The iterations work towards a system as

can be seen in figure 7. Consisting of three main elements, back-end, front-end, and external services.

4.1.1. Iteration 1. Back-end Framework

4.1.1.1. Problem description. The software of the prototype can roughly be divided into two parts, the front-end and the back-end. Within the back-end tasks such as routing, data processing, storage, are handled, see figure 7. In order to be able to focus on the unique challenges of the prototype and to not reinvent the wheel for already well-studied systems, a framework should be selected were on which this back-end can be based.

4.1.1.2. Proposed solution. Node.js [37] could be used as a framework for the back-end of the prototype. Node.js is known for its high-performance, large ecosystem of libraries, and ability to do asynchronous jobs, as well as the ability to use one programming language in both the front as back-end. Combining Node.js with Express.js [38] allows to make the service accessible as a web application.

4.1.1.3. Conclusion. The Node.js back-end works well at this development stage. However, the prototype is still an empty shell and thus little can be tested.

4.1.2. Iteration 2. Front-end Framework

4.1.2.1. Problem description. As with the back-end, a framework for the front-end should be chosen. The front-end of the prototype supplies users with a graphical user interface. Data from the back-end is collected in the front-end and then transformed in a user understandable format, see figure 7.

4.1.2.2. Proposed solution. React.js [39] could be used as front-end framework for the prototype. React.js is a JavaScript framework that is built around interactive UIs and reusable components. Moreover, most important for the prototype, React is developed to update components as data changes. This updating will allow the prototype to reduce user waiting times, as it can dynamically update the application as data comes in.

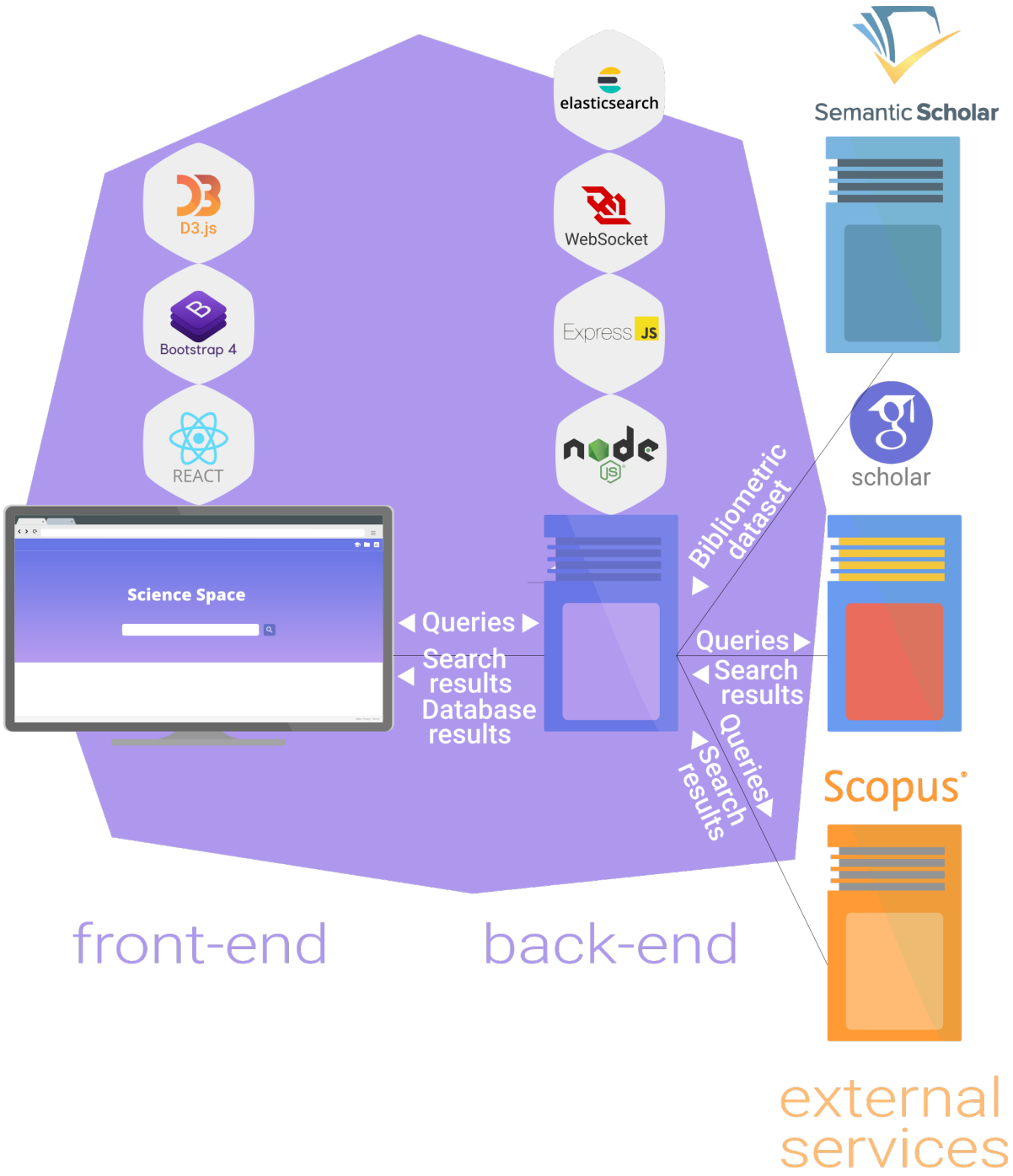


Figure 7: System architecture

4.1.2.3. Conclusion. Development of the prototype's front-end using React showed that React has a steep learning curve. Much time was spent to understand React, which will continue in other front-end development stages. However, Reacts data handling features show promise for handling incoming data in the prototype as efficient as possible.

4.1.3. Iteration 3. Migrating design to code

4.1.3.1. Problem description. Using React as front-end framework, the design drafted in the preliminary study can be converted into code. First of all, since React is component-based, the design should be split into components. These components will implement logic and graphics for its parts of the interface.

Secondly, in order to comply with Reacts philosophy, all components should be designed to be self-sufficient. For example, the visualisation component should be able to receive data from any source without it influencing its graphical result. This containment makes it simple to develop each component individually and reuse in different parts of the application.

4.1.3.2. Proposed solution. The design of the prototype could be implemented in React by splitting into the following components with the structure as can be seen in figure 8.

The *router component* could handle front-end routing, and is also the place were cookies and browser history can be implemented.

The *search component* can become the main component of the prototype, its main purpose would be to receive and distribute data generated by other components.

The *navigation bar component* can take the role of containing all interfaces were through users navigate the prototype. As proposed, the bar changes shape after the initial query. Due to the modular approach, the input element can be reused in both forms.

The *query input component* could accept user input and send the input to the back-end. The results can then also be received by the input component when the input is processed.

Finally, the *visualisation component* would receive data to create scientific networks.

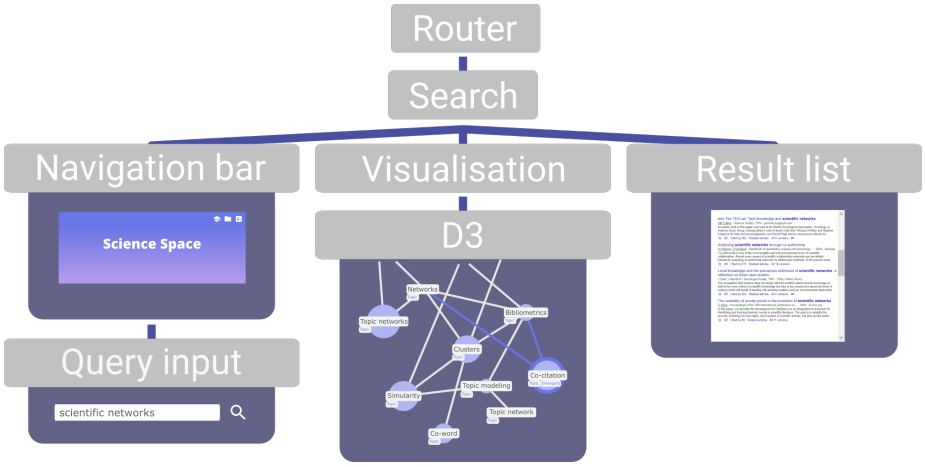


Figure 8: React component structure

4.1.3.3. Conclusion. For now, the React components, created by splitting the proposed design, work as expected. However, the Search component is already cluttered with data wrangling functions. This cluttering is due to that all components communicate through this main component. The implemented model is feasible for applications with few features, however as the application grows, a more streamlined model needs to be designed.

4.1.4. Iteration 4. Database Platform

4.1.4.1. Problem description. A database should be implemented to store acquired data and generated networks for later use. The back-end will bring in search results from scientific search engines and will acquire other data such as topics, authors and papers. From this data, scientific networks will be generated. By implementing a database the prototype does not have to request these resources over and over again. A database platform needs to be picked that allows for storing the generated and acquired data.

4.1.4.2. Proposed solution. The MongoDB [40] database platform for storing generated and acquired data of the prototype should be ideal. MongoDB is a non-SQL scale-able general-purpose database, and thus should be able to store the inconsistency of structure and continuously changing data of the prototype.

4.1.4.3. Conclusion. The MongoDB database for storing generated and acquired data of the prototype is still empty at this stage, thus no conclusions on applicability can be drawn. The database will be tested after data is acquired and networks have been generated.

4.1.5. Iteration 5. Data acquisition

4.1.5.1. Problem description. As proposed in the preliminary study, bibliometric data required to generate scientific networks will be retrieved on the fly from external platforms. However, one problem with this method is that not all bibliometric data sources have API's. These non-API sources have bibliometric data available via their websites but are not created to be accessible by servers. Therefore, in order to use the services without API's, a workaround has to be found.

4.1.5.2. Proposed solution. In order to receive bibliometric data for creating scientific networks from non-API sources, Request [41] could be used. The Request Node library allows to execute HTTP calls and follow redirects to retrieve data from online sources. Combined with Cheerio [42], the retrieved HTML can be interpreted and the required data could be stripped.

4.1.5.3. Conclusion. Using Request in the prototype is able to request bibliometric data from non-API sources as expected. However, for now, only the ability to search for papers with a query using scientific search engines is implemented. Using this method a user is able to type in a query in the prototype and receive search results from an external service. Looking up relating papers for building scientific networks is left for a future iteration.

4.1.6. Iteration 6. search results from multiple sources

4.1.6.1. Problem description. Currently, only one scientific search source is implemented. By adding another, more participants would be enabled to compare the prototype with their preferred search engine of choice. The single difference between the prototype and a "normal" scientific search engine should be the way search results are being displayed. Therefore, among other things, the prototype imports search results from existing search engines. Hereby, participants can compare their usual experience with the prototype's. In order to increase the amount of participants that can compare experiences, another search engine wherefrom search results are brought in should be added.

4.1.6.2. Proposed solution. To be able to import search results from multiple sources and process them equally, an abstract base class for acquiring search results could be created. This class would implement all generic features necessary to acquire bibliometric data from external services. All concrete implementations can then be based on this base class and enjoy all current and yet to come features. The code to switch between sources also becomes simple, as the same methods are being used.

4.1.6.3. Conclusion. The only found difficulty during the implementation of an abstract base class for acquiring data from different sources, was the difference in the formatting of search results. The response from the different services had to be converted to one single format. Hereby the rest of the program is able to use the data without implementing fixes in various places.

4.1.7. Iteration 7. Meet the rate limits

4.1.7.1. Problem description. During testing, strict rate limits were found to protect external bibliometric data services. Whenever a user or a service exceeds a set amount of request the service stops responding or asks for a Captcha. This is done to prevent misuse. In order to stop the prototype from being blocked, a way to restrict the prototype from exceeding limits needs to be found.

4.1.7.2. Proposed solution. A restriction method to not exceed rate limits of bibliometric data services could be implementing a task scheduler and rate limiter using Bottleneck [43]. Bottleneck allows to create a first in first out waiting queue for tasks. Hereby, the prototype is able to spread out the requesting tasks in the queue so that set rate limit is not exceeded.

4.1.7.3. Conclusion. Using Bottleneck rate limits of bibliometric sources were hit less often. Since limits were found to be not always documented, setting up Bottleneck demanded trial and error. In its final form, the setup is as follows: only one request is allowed to be executed at once, each request must be at at least 300ms apart and a maximum of 100 requests are allowed to be executed in a time frame of ten minutes. These rules are set up to be fairly conservative and try to mimic a human user as much as possible. In the future, the rules could be set up less conservative. However, this means that the risk of running into the rate limiter becomes higher as well.

4.1.8. Iteration 8. cache requests

4.1.8.1. Problem description. In the previous iteration was attempted to hit rate limits of bibliometric data sources less often. However, these limits turned out to be stricter than expected. The limits are clearly designed to prevent non-humans to use the service. Repeated identical calls to the service for testing were found to trigger the rate limits frequently. Thus, a system needs to be implemented that prevents repeated identical calls from triggering the rate limits.

4.1.8.2. Proposed solution. Cached-request [44] could help repeated requests to external services to not trigger rate limits. Cached-request allows to cache request done with the Request [41] library. By using Cached-request calls local results stored for a set time are checked before requesting new data from a service.

4.1.8.3. Conclusion. By implementing Cached-request, rate limits are almost never hit. Not only does Cached-request not trigger the rate limits as often,

also a speed increase was found whenever data was cached. The round trip time for a cached request is far lower than a new request to a service, therefore the user sees results more rapidly.

4.1.9. Iteration 9. Increase lookup speed

4.1.9.1. Problem description. More bibliometric data needs to be brought in to generate scientific networks. However, the current method to request data is unable to handle more load. As proposed in the preliminary study, bibliometric data can be acquired on the fly from external services. However, until this iteration, only search results were obtained from these external services. As it is the goal to display these search results within a scientific network, more bibliometric data relating to the search results needs to be acquired. In previous iterations rate-limiting of external services was found to be strict. By adding on more requests to the same services the limiting could become even more problematic. A solution needs to be found that allows to look up bibliometric data without overwhelming rate limits.

4.1.9.2. Proposed solution. In order to look up bibliometric data without running into rate limits, the Semantic Scholar [45] database could be used. Semantic Scholar is a large open-source database of bibliometric data with over 45 million published research papers in Computer Science, Neuroscience, and Biomedical fields. Due to the open-source nature of the database, the usage limits of the database are far less strict as commercial counterparts.

4.1.9.3. Conclusion. Unfortunately, the Semantic Scholar dataset will not always be able to provide additional bibliometric data relating to search results. In tests whether bibliometric data relating to search results can be found in the database, titles of search result papers were manually looked up. In these tests was found that only about one-third of search results matched with entries in the database. Although the Semantic Scholar database seems large with 45 million records, it can be seen as a tip of the scientific paper iceberg. However, by

adding the database to the prototype the amount of request to external services can be decreased and thus rate limits can be stressed less.

4.1.10. Iteration 10. loading data

4.1.10.1. Problem description. This iteration considers how to make use of the Semantic Scholar bibliometric dataset for looking up bibliometric data to create scientific networks programmatically. The simplest approach would be to make use of the Semantic Scholar API. However, the API is only meant to serve one-off requests and does not support repeated lookup of papers as required to generate scientific networks. For tasks that are not on a one-off basis, the Semantic Scholar dataset can be downloaded and used locally. Hence, the dataset is provided in 40 zip files of 1GB each providing JSON files of papers one per line. In order to be able to use the dataset in the prototype, a script needs to be designed that obtains and prepares the set to be used in the prototype.

4.1.10.2. Proposed solution. In order to load and prepare Semantic Scholar data for use in the prototype locally, a script should be written. The script should be able to automatically download, extract, import and finally index the Semantic Scholar documents. A custom bash script could execute the above tasks using various GNU packages [46] combined with the MongoDb's mongoimport [40] and Mongo.js for indexing.

4.1.10.3. Conclusion. The bash script for loading and preparing Semantic Scholar data script turned out to be less simple than was envisioned. One of the first problems that was encountered were disc space deficiencies. Although enough disc space was available for the dataset alone, not enough space was available to both store the zips, extracted zips, work files, and newly created MongoDB database. The script from the previous iteration needed to be altered to delete all intermediate files as they served their purpose while keeping some key files for redundancy. This redundancy turned out to be important as the process was not without its teething troubles and repetition of some of the steps would take a long time.

4.1.11. Iteration 11. slow querying

4.1.11.1. Problem description. MongoDB data platform used in the prototype turns out not to be optimised for text queries on particularly large datasets. In the previous iteration, a bibliometric dataset was imported in a local MongoDb database of the prototype. Using MongoDB's Mongo.js library looking up papers in the dataset should have been simple. Papers in the database are indexed based on their title and by using the title of a search result the two can be cross-referenced. However, the Mongo service takes tens of minutes to return a single request. Although a fast system is not required for this research, the experienced speeds do not allow for reasonable testing and debugging. Therefore, a solution needs to be devised to increase lookup speed.

4.1.11.2. Proposed solution. A solution to the slow querying speeds of MongoDB could be to switch the database to the Elasticsearch [47] platform. Elasticsearch is an open-source search engine optimised for full-text search and thus seems more fitted to the task. However, interchanging Elasticsearch for MongoDB means that the data from Semantic Scholar needs to be re-downloaded, unzipped, adapted to meet Elasticsearch standards, uploaded to the local Elasticsearch cluster, and indexed.

4.1.11.3. Conclusion. The query performance has been increased by switching database platforms from MongoDB to Elasticsearch and altering the data import script. The data upload script written in a previous iteration had to be altered slightly because Elasticsearch uses more performance intense data indexing algorithms than MongoDB. In the new script, the Semantic Search files are first split into smaller chunks and altered to conform with the bulk uploading format of Elasticsearch.

In the new system, cross-referencing search results is significantly faster. Querying takes with the new system less than a minute and can handle more queries in parallel without significant performance decline. While a query of a minute is still long compared to the waiting times of state of the art search

engines, this could be improved by using faster hardware and or more clusters.

4.1.12. Iteration 12. Faster server communication

4.1.12.1. Problem description. In the previous iterations, an HTTP RESTful API is built by which the front-end communicates with the back-end, however this turns out to be inefficient. Due to the stateless request-response pattern of HTTP, the back-end needs to fully process all (sub)tasks before it can send any of it back to a client. In the back-end of the prototype, many of these consecutive (sub)tasks are carried out. For example, (1) looking up search results, (2) cross-referencing the search results in the database, and (3) getting cited papers for each cross-referenced search from the database (4-) and so on. Therefore, back-end could either wait until all tasks are executed and then send results back, or send the first set of (sub)results and proceed the consecutive process when a request is received for the next (sub)tasks.

The first option would require the user to wait for a long time before any result is returned, without receiving any intermediate indication of progress. The second option will return intermediate results. However, a large number of overhead is required as the statelessness of HTTP requires to resend the intermediate results. Consequently, the complete process, using option two, will take even longer than option one. Therefore, this iteration studies a method that can return intermediate results without requiring considerable overhead.

4.1.12.2. Proposed solution. A method for back- and front-end communication that can return intermediate results without requiring lots of overhead could be to incorporate WebSockets [48]. WebSockets establish and maintain a connection between server and client. Additionally, Bi-directional communication is supported and almost real-time can be achieved due to moderate overhead. By using WebSockets, the back-end will be able to return multiple sets of data for each result as soon as the result becomes available.

4.1.12.3. Conclusion. Implementing WebSockets for back- and front-end communication turns out to be more cumbersome than RESTful HTTP. This is mostly due to WebSockets being a more low-level protocol. However, the user experience seems to benefit greatly, because data is directly presented as it is available.

4.1.13. Iteration 13. Visualisation framework

4.1.13.1. Problem description. As with the back-end and the front-end, a framework should be selected for creating the visualisation part of the prototype as well. This visualisation part should house the scientific networks and handle everything from raw data to rendering graphics.

4.1.13.2. Proposed solution. In pursuance of creating the visualisations part of the prototype, the Data-driven documents framework or D3.js [49] should be considered. D3.js is a JavaScript library created for creating visualisations based on data and has a wide supporting community. An example whereon the networks of the prototype can be based is already created by Heybignick [50]. In the example, a network visualisation is created that consist of nodes and links which spread apart by applying repulsive and attractive forces.

4.1.13.3. Conclusion. Using D3.js as the framework for creating network data-visualisations offers much flexibility. Various types of visualisations can be created with D3.js and much customisation is possible. However, this flexibility is also the downside of D3.js. Since there are so many possibilities for choosing, finding the right path to reach a premeditated goal can be difficult. Thereby the development process is lengthy. Nonetheless, after some trail and error, Heybignick's [50] example code for networks seems to work with some static bibliometric data samples as expected.

4.1.14. Iteration 14. D3.js compliance with react

4.1.14.1. Problem description. Intermediate interfaces need to be created to convert in- and outgoing data-flows for communications between the various

frameworks used in the prototype. In this case, the front-end of the prototype is based on React and needs to communicate with D3.js deployed to create network visualisations. This communication emerges to be challenging as both use a different approach to handle data. D3.js loads data, attaches it to DOM, and transforms those elements transitioning among states if necessary. React creates components to keep track of their state and passes in properties to re-render themselves. A method to combine the two approaches within one application needs to be found.

4.1.14.2. Proposed solution. Many approaches have been developed by the community in order to create communication between React and D3.js. These approaches vary from full-fledged libraries to communication philosophies with some examples. Therefore, to examine which approach fits best, various approaches need to be looked over and tried out.

4.1.14.3. Conclusion. To be able to communicate between React and D3.js various implementations from the community were analysed. From the libraries that were available, no library offered an implementation of network graphs. Therefore, no libraries are suitable to be used in the prototype. From the approaches that were looked into, Nicolas Hery's approach [51] suited best. The approach offers relatively simple methods to create a communication channel between React and D3. On top of that, the approach does not require any large adjustments to the implementation of both of the two frameworks. Thereby, example code, such as the force directed graphs from Heybignick [50], should be able to be implemented largely as is.

4.1.15. Iteration 15. Updating force graphs

4.1.15.1. Problem description. As the front-end is built upon React to update specific elements as data changes, the visualisations in the front-end should be able to do the same. However, D3.js out of the box does not implement features to dynamically update visualisations. In the D3.js force-directed graph example code from Heybignick [50], networks have to be re-rendered when its

data is updated. Re-rendering the complete visualisation is sub-optimal as it takes up time and causes inconsistencies in the visualisations. Therefore, this iteration looked at how to dynamically update individual elements of force-directed graphs.

4.1.15.2. Proposed solution. In order to dynamically update force-directed graphs in D3.js, various approaches were investigated. From these approaches, Bostock's [52] approach seems to be most promising as it shows simple features for adding, removing, and modifying individual elements within D3.js visualisations.

4.1.15.3. Conclusion. Enabling the prototype to update D3.js visualisations similarly to React elements have been proved to be successful. However, as with the initial setup of the D3 library, it can be concluded that D3.js can do much. Nonetheless, it can also be complicated to understand. Although the implementation is successful, grafting in an algorithm that appears to be simple took far longer than expected. On top of the lengthy process, many improvements can still be made both functionally as in the cleanliness of the code.

4.1.16. Iteration 16. Create citation networks

4.1.16.1. Problem description. For this iteration the goal is to implement the most basic type of network, the citation network. The citation network is chosen to keep complexity down, while still being able to show the potential of scientific networks. Implementing the scientific network consists of converting a user query to bibliometric data and thereafter divide the data into nodes and links.

4.1.16.2. Proposed solution. A citation network based on a user query could be created as follows: (1) retrieve search results from a state of the art scientific search engine based on a query, (2) cross-reference the search results based on their title in the local Semantic scholar database, (3) return metadata for each search result which includes a list of citations , and (4) pass citations as links to the network visualisation algorithm and papers as nodes.

4.1.16.3. Conclusion. The proposed four-step process to create citation networks cannot be fully established due to missing data. During development was found that within entries of Semantic scholar data, the set of inbound and outbound citations were often incomplete. Not only can more citations be found by manually looking into papers, continuity is not present in inbound and outbound papers as well. For example, paper X is cited by paper Y. Therefore, paper X should be in the outbound list of paper Y. Subsequently, Paper X should be present in the inbound list of papers Y. However, the latter is often not true. Papers that are expected to be in certain lists are often not there. Due to the missing citations, only very limited citation networks could be created.

4.1.17. Iteration 17. Pre-process data for citation networks

4.1.17.1. Problem description. This iteration revolved around supplementing Semantic Scholar citation lists by pre-processing. Pre-processing is necessary since in previous iterations only limited citation networks could be created due to incomplete citations lists in the Semantic scholar database.

4.1.17.2. Proposed solution. Pre-processing of Semantic Scholar data to supplement in- and outbound citation lists could be done as follows.

First of all, a method should be selected to alter data in the prototype's local Semantic Scholar database. This method should be created to use computer resources as efficient as possible, as the number of documents in the database is large, namely over 45 million research papers). Based on Elastic.js [53] - the Node.js client for Elasticsearch data platform used in the prototype - an abstract database altering script could be created. This script should scroll through the local Semantic scholar dataset using so-called pagination. Thereby, a set amount of documents is loaded into memory, which then can be processed parallel of each other and afterwards re-uploaded to the database. By that, altering should be as resource efficient as possible. The script on its own does not implement any algorithms to alter data. Specific altering functions can be created that accept a document with paper data and returns an altered/updated document.

Secondly, an altering function should be written to update the incomplete in and outbound citation lists of Semantic Scholar. A solution could be to write a function that takes a paper document (X) and looks up documents that are in X's inbound citation list. Each of the looked-up papers should have X in the outbound list. If this is not the case the script updates the looked-up paper. The same updating and checking could also be done for the inbound lists.

4.1.17.3. Conclusion. The proposed solution to pre-process data for supplementing the database of the prototype has, although created to be resource-efficient, many issues with memory management. The issues arose as many sub-tasks were being executed for each document. Therefore, the number of documents processed in parallel had to be turned down significantly. However, by turning down the amount of parallel processed documents, the run time of the script increased to hundreds of days. In consequence, the proposed script is not suitable to alter large sets of bibliometric data. During investigation of the run time, the problem was found that similar issues with Elasticsearch are known at the department of data science of the university of Twente. From the department was learned that the solution is to use Hadoop clusters [54] for altering data instead of Elasticsearch. Hadoop is optimised for processing large datasets while Elasticsearch is optimised for search. These two services can work consecutively to provide the pursued results. However, implementing such a system was found to be out of scope for this research. Therefore, the prototype will not be able to display any large citation networks.

4.1.18. Iteration 18. Proof of concept author and topic networks

4.1.18.1. Problem description. In order to increase the utility of visualisations, author and topic networks could be added. Due to database limitations the prototype is not able to display large citation networks and thus utility of the visualisations is limited. However, the already used data does include the necessary information for author and topic networks. Initially, the goal was to only implement citation networks as this would be sufficient to give an initial look at the concept. However, due to the data problem, the latter was not true.

4.1.18.2. Proposed solution. A solution to add author and topic networks to already existing citation networks needed to be devised. Topic networks could be created by looking up papers in the database that are annotated with similar combinations of topics as search result papers. Hereby the topic networks should become an interconnected web of papers that discuss similar topics as search results. Similarly, author networks could be created. These author networks can be created by looking up papers in the database with similar combinations of authors as in search results. Hereby a network of papers, written by researchers occurring in, or authors closely relating to, search results should be created. By all means, there are more ways to create topic and author networks, however, the above methods should be able to give an idea of the added value of these scientific networks.

4.1.18.3. Conclusion. Topic and author networks created to increase the value of scientific networks have been implemented successfully. Moreover, clusters can be seen of authors, topics and citations. However, sometimes the lookup speed leaves something to be desired.

4.2. Results

The result of the iterative development process is a rudimentary web-based scientific search engine that includes visualisations to display search results. Although some envisioned features are missing, the search engine should be able to give an initial look at the proposed concept. The requirements drafted in the vision, *"Simple to use user interface"* and *"Clear, easy-to-read visualisations"* are implemented partially, and due to the complexity found during development the requirements *"Allowing in-depth research while keeping a clear overview"* and *"Enable recommendation of papers based on previous activity"* did not get implemented.

The resulting user experience looks like the following. In figure 9, the welcome page is showed as displayed after navigating to the web page. As proposed, the interface is kept straightforward and the only required user input is

the search query. Figure 10 shows how a user can choose between sources of scientific search results. With this option, users can pick the source of search results they are most accustomed to. Thereby test participants should be able to compare experiences with their favoured state of the art search engine.

After entering a search query, on the left of the screen an initial visualisation gets created from search results, as can be seen in figure 11. In addition, the search results are displayed in a more common listed format on the right as well. The visualisation gets updated as the back-end attains additional data.

In figure 12 additional information from the back-end is added to the visualisation. In this step the search results have been cross-referenced with the underlying database. Herefrom, metadata describing the search results is acquired and scientific networks are created. The last three figures show how a final visualisation could look. A birds eye perspective of the scientific networks can be seen in figure 13. Furthermore, in figure 14 a zoomed in view is given of a topic network that links multiple search results. Lastly, in figure 15 shows how individual search results are linked.

The process to create a visualisation, as described above, can take 1 to 3 minutes. This is due to the used hardware and lack of additional optimisation. Furthermore, besides dragging around the network no other user interaction is built in.

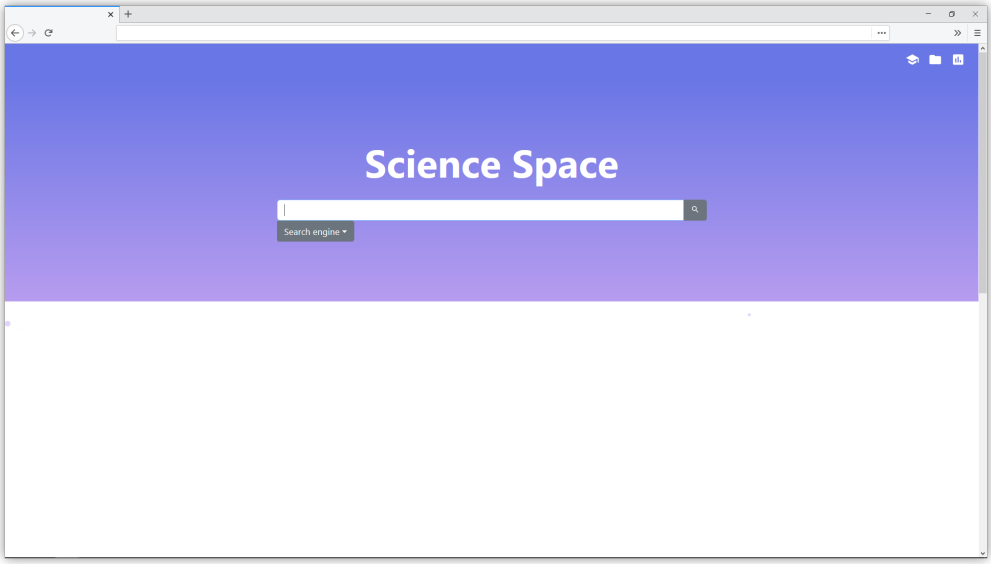


Figure 9: Welcome screen

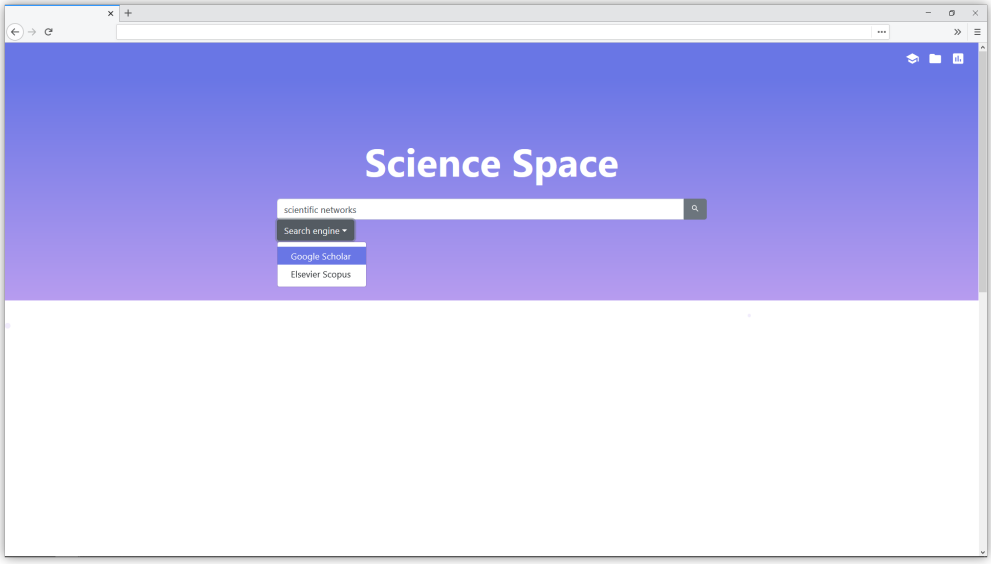


Figure 10: Choose search engine

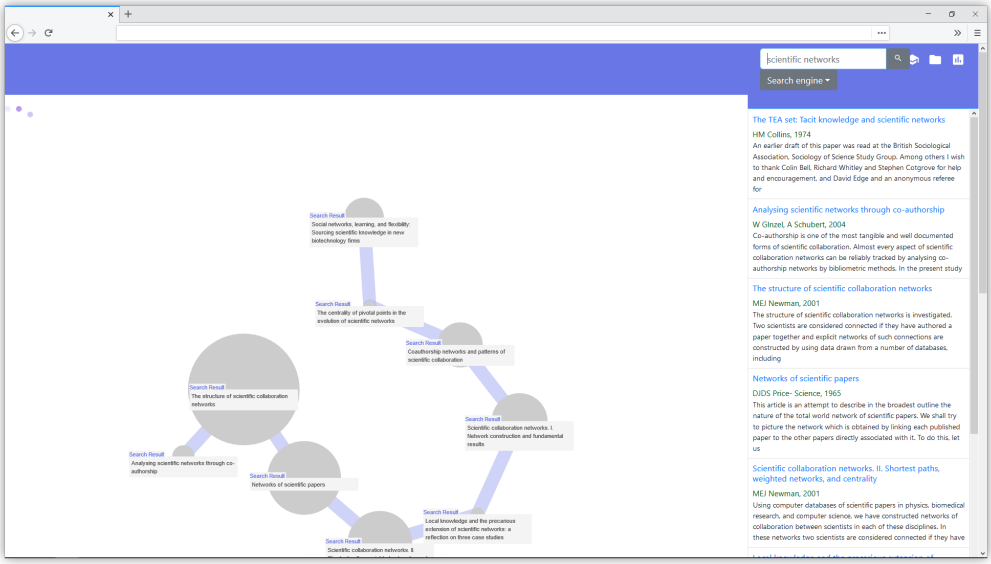


Figure 11: Search results

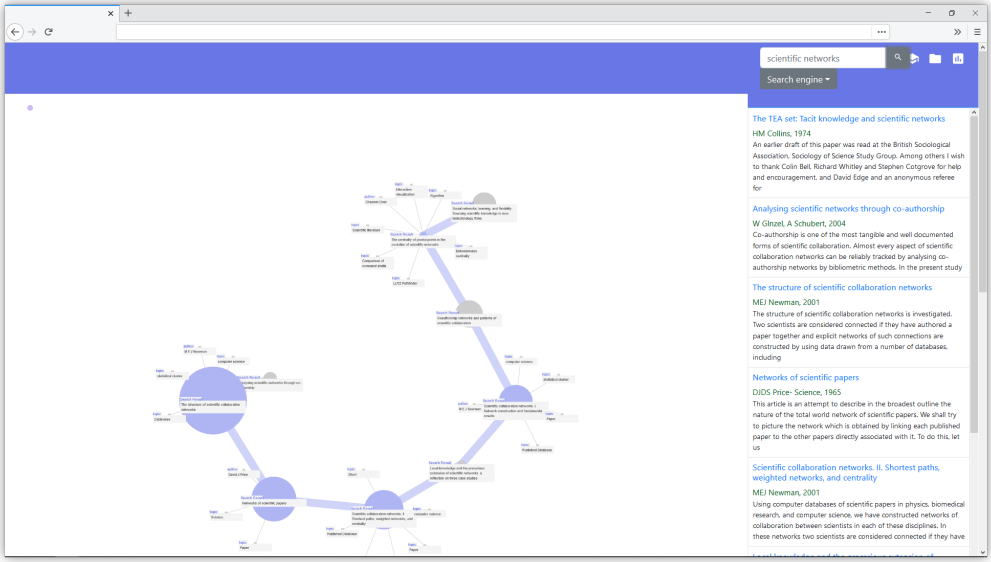


Figure 12: Lookup Search results

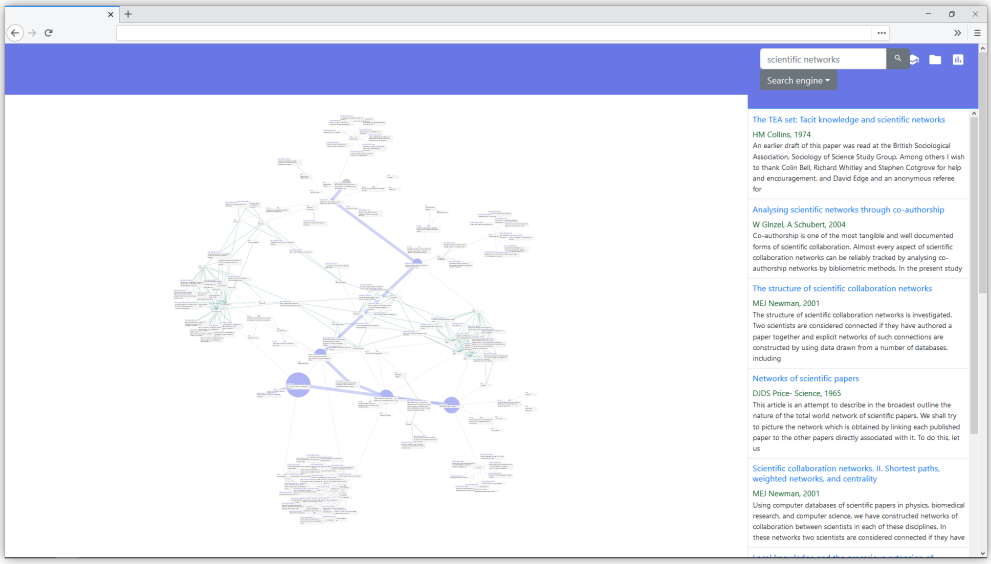


Figure 13: Network generation

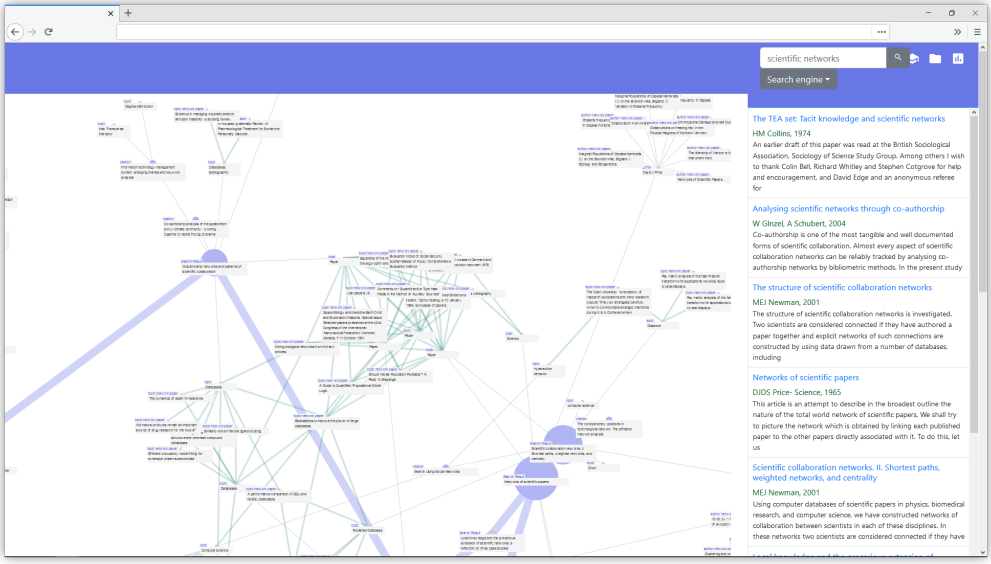


Figure 14: Network zoomed in

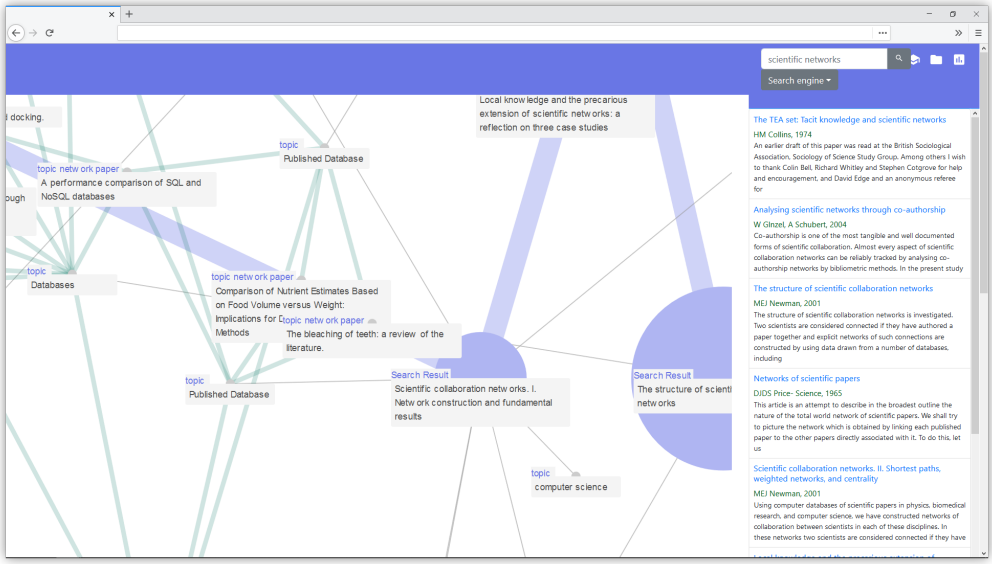


Figure 15: Node

5. Evaluation

Using a small evaluation the prototype's ability to help researchers explore literature and convey the relevance of search results is assessed. The prototype incorporates search algorithms of state of the art scientific search engines and makes use of scientific networks to display search results. Using scientific networks, it was attempted to improve on state of the art search engine interfaces. Particularly the ability to help researchers explore literature and methods to convey the relevance of search results.

The prototype creates scientific networks by cross-referencing search results of a state of the art search engine to a local database. From the database metadata of the search results is acquired. Thereupon data for topic, author, and citation networks is accumulated. Additionally, the back-end directly streams the acquired data to the frond-end, thereby informing the user about the contemporary progress. Hereby the front-end renders visualisations and dynamically updates as data changes.

Given these points, the prototype practical limitations of the proposed con-

cept are included in the evaluation. The evaluation was performed by interviewing future users and experts. The participants were walked through the prototype and design decisions were explained where necessary. Hereby insights were gathered whilst taking in consideration the experimental stage of refinement.

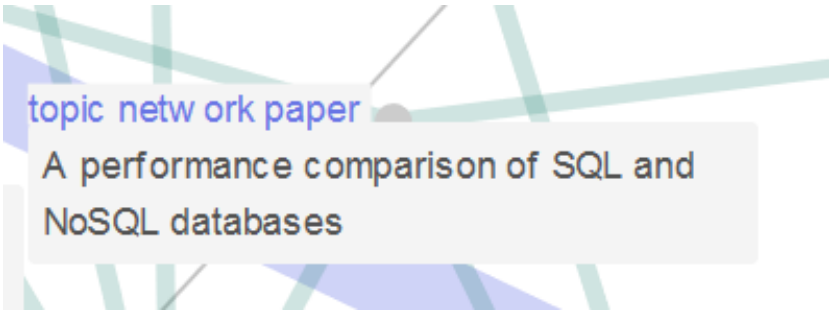
5.1. Students

To accumulate first impressions of future users, about fifteen students were asked to navigate on their own computer to a website running the prototype. These students were gathered from the DesignLab at the University of Twente and from a multidisciplinary group ranging from social sciences, industrial design to computer science. Before interacting with the prototype, the nature of the research and the current state of progress was put forth. Thereafter, participants were invited to use the application, as far as it was allowed at this stage, as if they were executing a literature study. In order to take out the learning curve involved with using a new application, participants were instructed to search for familiar topics. During interaction with the prototype participants reported on their experience.

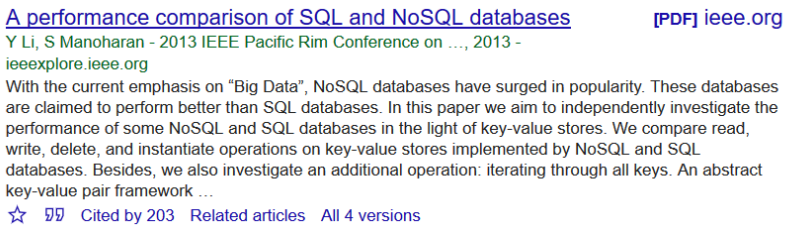
Participants noted that the current prototype cannot be used as a replacement for a state of the art scientific search engine. The prototype takes to long to load and key features that participants expected from web-applications are missing, such as the user history, stability and various unimplemented clickable items.

As proof of concept, participants found the prototype interesting. Participants could see themselves using an alike application in the future. Especially the unimplemented recommendation networks were noted to be a valuable addition to search engines.

The networks as currently presented by the prototype were found to be too chaotic. Participants had trouble distinguishing search results from papers added by the scientific network algorithms. Moreover, the number of links and nodes of aspects relating to these search results were found to be too great



(a)



(b)

Figure 16: Methods to convey context: (a) context provided by a network node in the prototype, (b) context provided by a result list in a state of the art search engine

and/or unordered.

When asked to look for papers they were not yet familiar with that related to their own studies, participants were able to find multiple examples within 5 to 10 seconds. However, participants had trouble to determine how relevant these papers were for their research. The context provided in a listed results page was reported missing, such as the text segment under the title of a paper in a more traditional interface, see figure 16.

5.2. Postgraduates

Two postgraduates from the Human Media Interaction group were asked for their opinion of the prototype. The postgraduates were generally more positive than the students. Although neither the networks were perfect, nor all premeditated features were implemented, postgraduates generally thought that with some tweaks the developed search engine interface could increase the ability

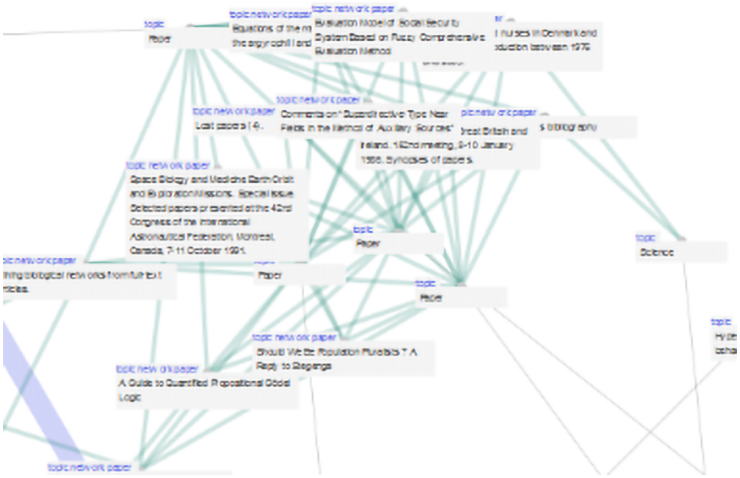
of search engine users to perform literature research. This was especially the case after showing how in the mock-up images the networks were uncluttered, as can be seen in figure 17.

Concerning the user interface to create networks, postgraduates noted that it is difficult to strike a good balance between giving options and simplicity. Allowing users to tailor to specific needs by giving them options can be crucial. However, ease of use and a gradual learning curve were also found to be important. A middle ground between these two facets can be hard to attain. The interviewees remarked that the prototype leans towards restrained simplicity by fixing how scientific networks are created and displayed. In future work could be sorted whether the interface should offer more versatility or not.

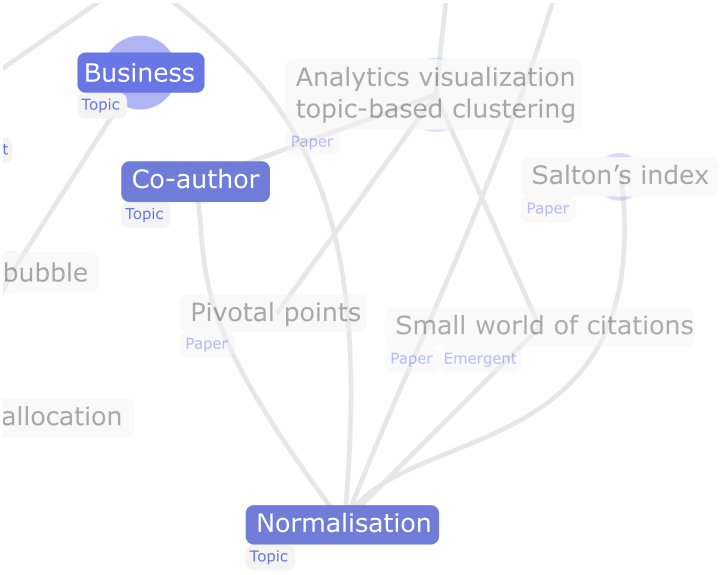
5.3. Experts

Two researchers in related topics (data science and from the scientific search engine industry) were included in the evaluation as well. The experts were as positive as the postgraduates. Nevertheless, they noted that uncluttering a network as shown in the prototype could prove to be impossible. Herefore the application must have exhaustive understanding of the data. The metadata required to build this understanding is not always available in bibliometric datasets. The absence of this metadata is something state of the art search engines are also struggling with. An international standard for paper metadata does not yet exist, but would be very helpful when building these metadatasets.

Finally, further research is necessary to make the application usable for actual users. Experts thought it was relevant to further improve scientific search engine interfaces. They recognised this as an existing problem, and would therefore be interested in further investigation. A next step to improve the prototype could be to incorporate data pre-processing as proposed in section 4.1.17 using Hadoop. Hereby the process to create scientific networks should become faster. This would be beneficial to develop and test various networks forms.



(a)



(b)

Figure 17: Difference between prototype and envisioned networks: (a) cluttered network of the prototype, (b) clear network of the mock-up

6. Conclusion and Discussion

Improving search for scientific publications by combining existing search engines with methods developed for scientific networks, turns out to be a challenging endeavour. Nevertheless, this research was able to build a first working example. This study explored how to use scientific network visualisations to indicate the relevance of search results and how to use visualisations to support the process of exploration for new relevant literature. A definitive answer on how exactly scientific could help indicate relevance and aid in exploring was not found. As the prototype included limited functionality for both of the examined subjects. However, in interviews with experts, postgraduates and students almost all participants expressed to see potential in the concept of using scientific networks for scientific search, taken into account that the prototype needs to be developed further. By building the prototype was shown that the vision drafted in the start of the research could only exists in a "perfect world". Nonetheless, by trying to build the vision, a first glance was cast on the obstacles that need to be overcome.

6.1. Scientific Networks

In preparation of the research, scientific networks have been explored in literature. This exploration was done for the parallel of scientific networks and search engine interfaces of explaining bibliometric data. In literature was found that scientific networks are priced for showing how research is built up, to show how different topics are connected and how close they are connected, and spot clusters of emergent research. However, the most common limiting factor was named to be the lack of open access data and not being practical for researchers outside the field. Much progress is to be made before scientific networks yield for the average researcher.

6.2. Vision

A vision was drafted in order to gain insight into whether scientific networks could be incorporated in scientific search engine interfaces and if some of the

issues found in literature could be addressed.

First of all, it became emergent that scientific networks tend to become cluttered and hence hard to interpret. Therefore, a collection of clustering and filtering techniques found in scientific network literature were included in the vision.

Secondly, it was noted to keep the interface usable. Due to an immense amount of options, previous scientific network toolsets demonstrated to become unusable for unacquainted searchers. Therefore, a focus on keeping the number of options for the end-user to choose from as low as possible was proposed.

Thirdly, the vision makes a point of focusing on being more than a pretty visualisation, since it was established that the versatility of multiple previous scientific networks visualisations was low. The visualisations should focus on allowing in-depth research while keeping a clear overview.

Lastly, recommendation networks were envisioned. These recommendation networks couple relevant bibliometric data to bookmarked references in order to extend the abilities of researchers to explore literature.

6.3. Exploration of initial issues

As an initial exploration of issues of the vision, expert researchers were interviewed. This research mostly relied on interviews since few literature was found regarding adaptation of taught literature research methods and use of scientific search engines in practice. The latter makes it difficult to conclude from literature whether the vision takes up any experienced issues in search.

From the interviews emerged that current scientific search engines can be a time consuming and not always a productive undertaking. In the available literature similar conclusion were drawn. Furthermore, it was found that expert researchers, over the years, create a form of scientific network in their heads. This network is used to position research, determine the relevance of new papers and explore their research field. Often this network is consulted instead of using search engines. The reported networks show similarities to the vision of this research to improve upon scientific search engine interfaces.

However, the availability of the required data for creating the envisioned visualisation was noted by experts to be in its infancy. Much data is present, yet not available in large quantities for the general public. However, in the coming years, this data should become more accessible. Nonetheless, whenever the data is available, to handle the vast amounts of data was noted to require large advanced infrastructure.

6.4. Prototype Development

Instead of finishing the research by evaluating the vision, a proceeding has been made by building a prototype scientific search engine. Hereby, practical limitations, such as the data deficiencies noted in scientific network literature and expert interviews, could be taken into account.

In developing a scientific network search engine interface, a route full of alternatives has to be navigated. Furthermore, the absence of available literature and various dependent subsystems complicated the development process. Due to complexities, not all parts of the vision were implemented in the final prototype. As conceptualised, the interface is kept as simple as possible, whilst communicating with the user using real-time updates without requiring full re-rendering. Furthermore, the visualisations were build from nodes and links and separated by repulsive and attractive forces. However, the visualisations are not as clear and refined as envisioned. Likewise, the methods envisioned for allowing in-depth research and recommendation networks were not implemented. Additionally, the back-end turned out more extensive than imagined. The back-end supplies the interface with data and is able to gather search results from multiple sources by user preference. Due to rate limiting conflicts, instead of an on-the-fly bibliometric data collection method, a local database was implemented. Text search is used to cross-reference search results and bibliometric data entries. Using the latter, a collection of metadata can be accumulated wherefrom scientific networks can be created.

The result is a rudimentary web-based scientific search engine that includes visualisations to display search results. Although various envisioned features are

missing, the prototype is able to give an initial look at the proposed concept.

6.5. Evaluation

In a small evaluation students and experts were interviewed. At first glance, students expressed themselves doubtful as the prototype is not ready to be used as a replacement for a state of the art search engine. The speed wherein the visualisations are created was found not fast enough and the clarity of the networks not yet adequate. Although students noted to see potential in the concept. They were able to find relevant papers to known to them topic quickly by using the visualisations. Thus, in the future the prototype could add to their ability to explore literature and understand the relevance of search results. Experts also recognised that the prototype is not ready for prime time and that a few of the encountered problems, such as data deficiencies, are not easily solved. However, the preeminent consent was that the prototype is a good first attempt to improving search engine interfaces and worth investigating further.

7. Future Work

The prototype shows potential to aid literature research in the future, however, to do so future work is necessary.

Speed improvements A first hurdle is to increase the speed by which networks are created. By increasing this speed, elements such as network algorithms, user interfaces, communication protocols, can be tested and refined more rapidly. As it stands, this process takes too long to effectively do development work.

Clarity and versatility After looking into speed improvements, a start can be made to improve user experience. Since in improving clarity and versatility of the scientific networks much progress is to be made. The networks did not provide enough context, were cluttered and in-depth research is not yet possible. To build upon the current state, for example, multiple types of networks can

be proposed and shown to prospective users. Hereby a better understanding of how exactly to improve clarity and a better sense of the needs of users could be gained. Perhaps this examination can be done using a lower fidelity user interface to further increase the rate of development.

Bibliometric data Subsequent to investigating the best form or type of network a challenging next step embarks. Namely, as mentioned by experts, creating the perfect networks requires much understanding of data. As the metadata of papers is not always publicly available, creating the most desirable type of network will be challenging. Herewith, a collaboration with a supplier of bibliometric data could be the key to success.

References

[1] J. Schyns, Literature review; scientific networks.
URL `http://joep.space#scientific_networks`

[2] C. Dunne, B. Shneiderman, R. Gove, J. Klavans, B. Dorr, Rapid understanding of scientific paper collections: Integrating statistics, text analytics, and visualization, Journal of the American Society for Information Science and Technology 63 (12) (2012) 2351–2369.

[3] A. Suominen, H. Toivanen, Map of science with topic modeling: Comparison of unsupervised learning and human-assigned subject classification, Journal of the Association for Information Science and Technology 67 (10) (2016) 2464–2476.

[4] G. Naik, The quiet rise of the nih's hot new metric, Nature 539 (7628) (2016) 150.

[5] M. L. Wallace, V. Larivière, Y. Gingras, A small world of citations? the influence of collaboration networks on citation practices, PloS one 7 (3) (2012) e33339.

[6] M. E. Bender, S. Edwards, P. von Philipsborn, F. Steinbeis, T. Keil, P. Tinnemann, Using co-authorship networks to map and analyse global neglected tropical disease research with an affiliation to germany, PLoS neglected tropical diseases 9 (12) (2015) e0004182.

[7] M. E. Newman, The structure of scientific collaboration networks, Proceedings of the national academy of sciences 98 (2) (2001) 404–409.

[8] J. D. Dworkin, R. T. Shinohara, D. S. Bassett, The emergent integrated network structure of scientific research, arXiv preprint arXiv:1804.06434.

[9] C. Wang, D. M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 448–456.

[10] R. Nakazawa, T. Itoh, T. Saito, Analytics and visualization of citation network applying topic-based clustering, Journal of Visualization 21 (4) (2018) 681–693.

[11] D. Nicholas, I. Rowlands, P. Huntington, D. Clark, H. Jamali, E-journals: their use, value and impact, Research Information Network.

[12] B. Mons, C. Neylon, J. Velterop, M. Dumontier, L. O. B. da Silva Santos, M. D. Wilkinson, Cloudy, increasingly fair; revisiting the fair data guiding principles for the european open science cloud, Information Services & Use 37 (1) (2017) 49–56.

[13] N. K. Ahmed, R. A. Rossi, Interactive visual graph analytics on the web., in: ICWSM, 2015, pp. 566–569.

[14] Infrastructure services for open access.
URL `https://is4oa.org/`

[15] Eric - education resources information center.
URL `https://eric.ed.gov/`

[16] Webofscience.
URL https://www.webofknowledge.com/

[17] Scopus preview.
URL https://www.scopus.com/

[18] P. Pradhan, Science mapping and visualization tools used in bibliometric & scientometric studies: An overview.

[19] O. Persson, R. Danell, J. W. Schneider, How to use bibexcel for various types of bibliometric analysis, Celebrating scholarly communication studies: A Festschrift for Olle Persson at his 60th Birthday 5 (2009) 9–24.

[20] C. Chen, The citespace manual, Google Scholar.

[21] R. Feldman, J. Sanger, Information extraction, The text mining handbook: Advanced approaches in analyzing unstructured data (2006) 94–130.

[22] G. Salton, A. Wong, C.-S. Yang, A vector space model for automatic indexing, Communications of the ACM 18 (11) (1975) 613–620.

[23] L. Leydesdorff, On the normalization and visualization of author co-citation data: Salton's cosine versus the jaccard index, Journal of the American Society for Information Science and Technology 59 (1) (2008) 77–85.

[24] P. Ahlgren, B. Jarneving, R. Rousseau, Requirements for a cocitation similarity measure, with special reference to pearson's correlation coefficient, Journal of the American Society for Information Science and Technology 54 (6) (2003) 550–560.

[25] B. Jarneving, A comparison of two bibliometric methods for mapping of the research front, Scientometrics 65 (2) (2005) 245–263.

[26] L. Egghe, Theory and practise of the g-index, Scientometrics 69 (1) (2006) 131–152.

[27] S. Lapinski, H. Piwowar, J. Priem, Riding the crest of the altmetrics wave: How librarians can help prepare faculty for the next generation of research impact metrics, arXiv preprint arXiv:1305.3328.

[28] M. Fenner, J. Song, Z. Dennis, M. Whitwell, J. Osowski, R. Ivimey-Cook, R. Cave, J. Lin, J. Chodacki, Lagotto 4.2.1doi:10.5281/zenodo.20046.

[29] R. A. Rossi, N. K. Ahmed, The network data repository with interactive graph analytics and visualization, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
URL http://networkrepository.com

[30] J. Webster, R. T. Watson, Analyzing the past to prepare for the future: Writing a literature review, MIS quarterly (2002) xiii–xxiii.

[31] D. Budgen, P. Brereton, Performing systematic literature reviews in software engineering, in: Proceedings of the 28th international conference on Software engineering, ACM, 2006, pp. 1051–1052.

[32] C. Hart, Doing a Literature Review: Releasing the Research Imagination, Sage, 2018.

[33] D. Papaioannou, A. Sutton, C. Carroll, A. Booth, R. Wong, Literature searching for social science systematic reviews: consideration of a range of search techniques, Health Information & Libraries Journal 27 (2) (2009) 114–122. doi:10.1111/j.1471-1842.2009.00863.x.

[34] T. Russell-Rose, J. Chamberlain, Expert search strategies: The information retrieval practices of healthcare information professionals, JMIR Medical Informatics 5 (4). doi:10.2196/medinform.7680.

[35] B. Shneiderman, The eyes have it: A task by data type taxonomy for information visualizations, in: Visual Languages, 1996. Proceedings., IEEE Symposium on, IEEE, 1996, pp. 336–343.

[36] P. Biernacki, D. Waldorf, Snowball sampling: Problems and techniques of chain referral sampling, Sociological methods & research 10 (2) (1981) 141–163.

[37] S. Tilkov, S. Vinoski, Node.js: Using javascript to build high-performance network programs, IEEE Internet Computing 14 (6) (2010) 80–83. doi:10.1109/mic.2010.145.

[38] A. Mardan, Express. js Guide: The Comprehensive Book on Express. js, Azat Mardan, 2014.

[39] A. Fedosejev, React. js Essentials, Packt Publishing Ltd, 2015.

[40] K. Chodorow, MongoDB: the definitive guide: powerful and scalable data storage, " O'Reilly Media, Inc.", 2013.

[41] request/request, [Online; accessed 14. May 2019] (May 2019).
URL https://github.com/request/request

[42] cheerio, [Online; accessed 14. May 2019] (May 2019).
URL https://cheerio.js.org

[43] bottleneck, [Online; accessed 15. May 2019] (May 2019).
URL https://github.com/SGrondin/bottleneck

[44] cached-request, [Online; accessed 15. May 2019] (May 2019).
URL https://www.npmjs.com/package/cached-request

[45] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H.-H. Ooi, M. Peters, J. Power, S. Skjonsberg, L. L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen, O. Etzioni, Construction of the literature graph in semantic scholar, in: NAACL, 2018.
URL https://www.semanticscholar.org/paper/09e3cf5704bcb16e6657f6ceed70e93373a54618

[46] gnu.org, [Online; accessed 16. May 2019] (May 2019).
URL https://www.gnu.org/#fsf-links

[47] Open Source Search & Analytics · Elasticsearch | Elastic, [Online; accessed 17. May 2019] (May 2019).
URL https://www.elastic.co

[48] A. Wessels, M. Purvis, J. Jackson, S. Rahman, Remote data visualization through websockets, in: 2011 Eighth International Conference on Information Technology: New Generations, IEEE, 2011, pp. 1050–1051.

[49] M. Bostock, D3.js - Data-Driven Documents, [Online; accessed 27. May 2019] (May 2019).
URL https://d3js.org

[50] D3.js v4 Force Directed Graph with Labels, [Online; accessed 27. May 2019] (Apr 2019).
URL https://bl.ocks.org/heybignick/3faf257bbbbc7743bb72310d03b86ee8

[51] Integrating D3.js visualizations in a React app - Nicolas Hery, [Online; accessed 27. May 2019] (Apr 2018).
URL http://nicolashery.com/integrating-d3js-visualizations-in-a-react-app

[52] M. Bostock, Modifying a Force Layout, [Online; accessed 27. May 2019] (Apr 2019).
URL https://bl.ocks.org/mbostock/1095795

[53] @elastic/elasticsearch [7.x] | Elastic, [Online; accessed 3. Jun. 2019] (May 2019).
URL https://www.elastic.co/guide/en/elasticsearch/client/javascript-api/current/index.html

[54] Apache Hadoop, [Online; accessed 5. Jun. 2019] (May 2019).
URL https://hadoop.apache.org