ulm university universität
uulm

Ulm University | 89069 Ulm | Germany

Faculty of Engineering,
Computer Science and Psychology
Institute of Media Informatics
Visual Computing Group

Interactive Citation Trend Analysis of TVCG Articles

Master Thesis in Cognitive Systems at Ulm University

Presented by:
Nitai Chandro Roy
nitai.roy@uni-ulm.de

Examiner:
Prof. Dr. Timo Ropinski
Prof. Dr. Enrico Rukzio

Advisor:
Christian van Onzenoodt

2018/04

Last updated April 27, 2018

Typesetting: PDF-LaTeX 2$_\varepsilon$

Abstract

Citation data can be used to measure the popularity of an article. As we know citation data changes over time so if these changes are presented over time with the help of visualization they reveal hidden trends in citation data. However, existing applications do not offer a simple way to extract these citation data, especially how they evolve in time-series. We developed a dynamic web scraper to extract this information from existing platforms and uniformly stored the cleaned data.

Using our acquired data we created an interactive visualization to explore this citation network. Individual article citation trends over time and comparison of the different article are in consideration. With the simple interaction, a user can get the overview of citations, compare citations between the article and find the citation patterns of the article.

Acknowledgement

I am thankful to Prof. Dr. Timo Ropinski and Prof. Dr. Enrico Rukzio for giving me the opportunity to do my master thesis under their supervision.

I would also like to thank my advisor Mr. Christian van Onzenoodt for his continuous support, guidance whenever I ran into trouble throughout my thesis implementation and writing. His supervision played an important role in successful completion of my thesis.

Last but not least, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my study. Many thanks to my friend Imran Mehmood for discussing issues when I needed.

Contents

1 Introduction

Scientific literatures are being frequently published to their related association. Many online scientific portals are publishing research paper so that student or researcher can get the access and foster their knowledge. In computer science community, DBLP, Institute of Electrical and Electronics Engineers (IEEE), and Association for Computing Machinery (ACM) are the largest leading organizations publishing subject related papers. Among many of them, The IEEE is one of the largest rapidly growing technical professional organizations provides access to the literature. In particular, IEEE *Transactions on Visualization and Computer Graphics (TVCG)* journal started publishing papers on the subject related to computer graphics, information visualization, visual analytics, human-computer interaction, etc. since 1995. In this TVCG, journal articles are being published monthly. Besides, DBLP is a computer science bibliography offering free access to quality assured bibliographic metadata including links to the different electronic edition of publications. These online libraries or repositories are containing the metadata of the articles. The metadata is in the textual format. Anyone can visit these online portals and search by keywords, authors, and title to get their target articles. For example, queries by keyword search will show results in a list of webpage link where the keyword matches insight the documents. That is fine for retrieving documents related to that keywords. But when we want to know the impact of an article or its influence on current research field that depends on many aspects such as citation, reference, or abstract. If we consider the citation aspect among them and want to know about citation details of an article, then it is laborious.

A selected article in IEEE (TVCG)[1] journal represents the citation information in text format. Visiting each of cited papers are time-consuming even challenging to traverse insight on it. Therefore, it is a challenging task for the user to explore the citation details. One of the challenges is to find the trends of citing a paper over time.

A visitor can gather citation information when an article has few citations. But, when the citation data becomes more substantial that includes multiple pages, then it is difficult to browse to gain an overview [29]. The traditional data extraction approach from web source is to write specific programs, called wrappers that recognize data of interest and map them to some suitable format as, for example, XML or relational tables [23]. Many data extraction tools such as Northwestern Document Structure

[1] https://ieeexplore.ieee.org

Extractor (NoDoSEa) [2] and Roadrunner [8] are available to extract data from HTML pages. But their approach cannot handle who wants to scrape data from a scientific literature portal such as TVCG journal.

Since the web relies on standardized technologies such as TCP and HTML it can be scraped for data in an automated way. An application for such a data extraction is therefore called a web scraper. It traverses web pages and scrapes the data that is specified in the scraper script. For extracting data from TVCG journal, a dynamic scraper was needed to retrieve the metadata. The IEEE (TVCG) website is built by AngularJS JavaScript-based web application framework. So, we developed a dynamic web scraper that can scrape metadata from dynamic websites which are generated in the client side using JavaScript. The approach of the dynamic scraper is to collect the data automatically, even scrape the updated data that is available in TVCG journal. This scraping process will help novice user without having programming skills but want to extract citation data from TVCG journal.

It is clear that a visual presentation is much easier to explore the overview of information than a textual description or spoken reports [29]. So, information visualization through graphical interfaces is likely to have an expanding role [29]. In the existing system of (TVCG) journal, a user can select an article and can discover insights the citation. For that, a user has to browse each of the citation links. For instance, to know the total citations for each year of an article, a user has to visit each citation links and find the year of citations manually. It can be possible, but it is a time-consuming process. Or a user may want to know the citation frequency over the years after publishing the paper. Or a user may wish to compare citation trends between multiple documents. But the existing system does not have such a functionality where a user can visualize the citation data in visual form. However, our visualization method illustrates the citation details for each year separately in a histogram visualization with minimal user interaction. Consequently, a user can get the total overview of citing articles for a selected article. Besides, citation trends are also visualized using line chart where a user can compare and contrast the citation trends between multiple papers in time-series. As we know the more a paper is getting cited by other papers, the more impact it has in the same research area. That is why the article is being cited for further discussion within other papers. We have developed an interactive visualization system that helps the user, who wants to explore knowledge regarding citation references of the articles published in (TVCG) journal.

In the following sections, background and motivation are briefly discussed. The second chapter is covered with related work those were implemented to collect data and visualize the textual data via different visualization techniques. In the third chapter, the overview of information visualization, citation indexing, database selection, web scraper, CitationVis, and the tools are discussed. In the fourth chapter, database

design for citation data acquisition using dynamic web scraper and processing for the visualization are explained broadly. In chapter five, data visualization techniques such as line chart and histogram are used to visualize citation data, those are illustrated with the explanation. Last but not least, the conclusion chapter summarises scraper and visualization techniques including future work.

1.1 Background

One of the fundamental tasks of studying or researching in a particular field is to know the existing area. How did the previous researcher research in that field? How current research helped to the topic, the researcher wanted to implement. Whether the implementation of the task met, they tried to prove. Following the previous investigation, new scholar seeks to improve either the existing system or to get the new idea from the related work and develop a new system. In this scene, whenever a scientific paper cites by another scientific article, it has the impact on the subsequent research [31]. Citations are being followed for many decades in scientific publications to extend the knowledge of previous researchers. So, the cited paper has a contribution to the topic of research on the citing paper. Therefore, the value of prior researched paper increased for its contribution to that field.

Because of digital libraries or repositories, we can quickly browse scientific papers. Most of the information on the web is available either unstructured HTML pages, semi-structured therefore the unstructured nature of these pages makes it hard to do sophisticated querying over the information present in them [3, 9]. But extracting structured data from the web pages is easy since it enables to pose complex queries over the data [3]. Most of the scientific paper published in online has either unstructured or semi-structured form. So, data extraction is a challenging task. Although, data extraction from the website started when digital libraries became available in order to explore the information in visualization for knowledge discovery.

Keywords based search is the most common information extraction method used in many digital libraries. Searching by keywords in digital libraries or repositories return many articles because the query runs all over the abstracts of metadata and matches the keywords then return the result. If the searcher wants to know which paper has a most recent impact on the current topic, then searcher have to go through all of the documents, read through them and find the most influential studies. Similarly, knowing details about the citation of their articles is challenging. So, extracting data based on topic or citation references would have an accurate result. Web of Science is the most significant citation database contain around 92 million records including more than a billion of cited references. It's primary focus on citation indexing, literature peer viewer. Search option from general to advance level are available. Advanced level of

search has a more specific search, and only expert users are normally able to utilize the advanced level search option. Also, Web of Science tries to find out the co-citation relationship between or within the papers. As linking of different documents increases the connectivity of various articles when focusing on the same concept or topic. So, being cited by other papers, or journals forwarding the theories for further improvement without changing the fundamental concepts.

Text visualization is a rapidly growing important subfield of information visualization thus many challenges arise for researchers to look for related work with a specific task or visual metaphors in mind [22]. Many textual visualizations were developed, and many web-based application and publications are available such as Neoformix, WordNet and TextVis [7, 19, 22]. Also, there are many visualization techniques visualized co-citation network, co-authorship, citation networks namely CiteWiz and CiteVis [10, 31]. But if a researcher wants to look to the trends of citing papers for a particular article in a graphical visual form, then it is a missing part of the research till now. Citation creates a bridge between one document to another, and give a better understanding of the related topic. So, it creates the citation network. One of the most effective citation network visualization tool is CiteVis that used to visualize author network, citation network. But we did not find any application which visualizes citation trends over time for individual articles that published in TVCG journal.

From a large scale of the dataset, information extraction, and exploration the vast volumes of data are becoming increasingly challenging [21]. Simultaneously, citation references are dramatically increasing in the scientific literature. To avoid plagiarism in the scientific research or giving credit to the previous study or work, citation plays a vital role. However, we are not going to prove whether or not the citation followed adequately. We will rather visualize graphically the current citation references of articles, the trends of citing articles in time-series, and comparison among cited papers over time. Also, the fluctuation of citation references is visualized yearly from 1995-2018. Consequently, we have developed CitationVis application that helps the user for observing and seeing the overall trends from the article publication time.

1.2 Motivation

Data visualization via graphical interface provides a glance overview of a large number of texts. There are many visual interfaces or web application such as CiteVis, PubVis, and TextVis [31, 16, 22], developed to explore information from the textual data. Different approaches are being used to extract and explore the data. The goal of data visualization is to make the data clean so that the information can be comprehensive to the users. Data visualization reveal the vital information behind the extracted data. Data is being visualized such a manner so that the viewer can observe the trends of data in time-

series. Therefore, it helps the researcher to come up with a decision for comparison of multiple articles citation [33].

A few decades ago, static histogram or line chart was used for information visualization. These visual methods help to aggregate information and summarise in the graphical view. Almost there were no user interactive visualization methods. Data is rapidly increasing, and it is a challenge to retrieve information by simple keyword searching in browser [11]. Many researchers started developing tools those are used for extracting and exploring data with the minimum interaction by the user. By the growth of data, the trend of data is changing over the time interval. A user might want to know the trend of data and analysis data. That covers how the pattern of data is varying over the time [34]. With the help of modern interactive visualization methods, we found many visualizations successfully developed for trend analysis. Some of them are analyzing the fundamental trend of the data or visualizing the pattern in a complicated way. A novice researcher might confront to find out the exact trends in the data.

Many developers are trying to make data visualization more interactive with a minimum of user interaction. For example, CiteVis [31] was developed to visualize citation relationship within IEEE (VIS) journal. In the related work chapter, CiteVis are discussed broadly including some other related works. Most of the existing visualization tools such as CiteVis, PubVis, and TextVis collect dataset one time. But our motivation is to make a scraper that is easy to use and can collect data automatically while the scraper is running for every scraping period. Trend analysis mostly involved in time-series visualization, for example, a line chart visualization. In our system, we have also focused on analysis trends of any particular topic after published. Therefore, our approach is to build a system that can extract citation data every time the scraper runs, and if the citation information is new, then the data should be scrapped and store to a database. IEEE (TVCG) publishing scientific topics regularly. If the article is informative for further research, different authors are citing that paper. So, every time paper is being cited by other papers means that the citing information of the cited paper is changing over the time.

Time-series data are missing from the existing applications. However, the citation of the papers are on the website, but they are in textual format. That is a challenge to discover how the trends of citation are changing or finding the missing citations in a year. For instance, a user is looking for an article, searching by the article title. A user seeks to citation details of a particular paper to explore some basic information from the citation content. Then the user should read citation details one by one which is a time-consuming task to know their author name, article title, and year of publication. We thought to scrape the citation data which belongs to each article in TVCG journal and later use the information to visualize in graphical form which helps the user to get an overview of the citations.

IEEE Website is a dynamic website build in AngularJs. Data scraping from a dynamic website is a complicated procedure because most of the data are only available when a TCP Get-request is sent to the server using JavaScript or AJAX call. Also, the problem arises when someone wants to know from which journal the citation is coming. TVCG journal is publishing papers containing two different kinds of citation information. Their internal citations within their journal and other citations from the different journal. So, every time a visitor has to visit the pages to get its citation references. Fortunately, there are some libraries namely Nightmare.js.js or Pupeteer.js can help to build a dynamic scraper to scrape contents from a dynamic website. Our data source is IEEE (TVCG) journal, and we are motivated to develop a specific dynamic scraper using Nightmare.js.js in Node.js only for TVCG journal. This scraper helps to collect citation data from TVCG journal with a minimal effort by the user.

Authors may want to know after publishing a paper that how many times their article is being cited. They can recognize it by just searching in Google Scholar. But these are in textual format. When the existing topic is being frequently cited that means this topic is more exciting. Other researcher also like to investigate or extend the ideas by quoting the text. In this case, the existing application cannot demonstrate the citations in graphical visualization. Therefore, we developed CitationVis web application. It helps to trend analysis of citation references in TVCG journal. Analysing the trends of citations of each article helps to know the overview of citations. Also, there is no such application that search by full title to explore the citation data of an article. We have developed CitationVis web-based application that helps to visualize citation trends by simple user interaction with the application.

2 Related Work

There are existing visualization tools to explore metadata of scientific publications. They collected the metadata either manually or alternative way. Their visualization tools tried to find seemingly hidden insights in metadata of textual form. The existing application attempted to visualize about citation details such as citation networks. In this chapter, we are reviewing some of the related applications, which are closely related to our citation trends visualization approach.

CiteVis [31] can be used to visualize the citation data about papers. This web-based application portrays the citation data of the IEEE InfoVis Conference including its articles. They used an attribute-based layout rather than a node-link network visualization. It designed to examine the total citation count of papers and its internal citations. Also, to explore a particular author's papers citation data of specific topics within the IEEE InfoVis Conference. It has search option by a paper, authors, organizations, and concepts. They included another extra feature where it shows the receiving Best Paper Award each year. They tried to visualize the paper networks used the concepts of the title of a paper or by authors network. Although, they visualized citation for paper but not considered citation trends visualization. This is missing research they have had. Also, the legend is ambiguous, so it is even difficult to understand how the citation network is working. Only expert users can use the application for searching an article if they know it otherwise it is not possible to get an overview of an article citation.

Another application Vispubdata.org [18] collected the metadata of all publications from IEEE VIS conference and made easily accessible to all. Their dataset included the metadata information about each paper title, abstract, authors, and citation to other papers that presented at the IEEE VIS conferences during 1990-2015. They used CDs, DVDs, and Memory sticks to collect the data. The data collection was done manually. Since citation data could be changed over time, so the real trend of citing papers are missing. They developed three visualization tools CiteVis2, CiteMatrix, and VISLists to represent the metadata graphically [18, 35]. Their focus was citations between papers across the conferences in IEEE visualization and counting the total citations. CiteMatrix [18, 35] followed matrix based visualization where row showed the cited papers and column showed the citing papers. It merely shows which paper is citing which paper. Therefore, it is also not intuitive when we want to know the overview of citations over time. VISLists visualization [18] followed by Jigasaw [15] List View system

that showed the lists of items namely authors, conferences, and years. This visualization is also not intuitive, and the research paper is limited. They visualized the contribution of the author in different years of the conferences. We can only know the research contribution in publishing paper.

Another citation network visualization was done by Elmqvist and Tsigas [10]. Their application CiteWiz can focus on the taxonomy of citation database usage for researchers. This tool tried to visualize the chronology of bibliographic and the network influences of the scientific articles. Their timeline visualization helped to get the overview and navigate in a full citation database. Influence visualization helped to understand the detailed views of a specific subset of the citation database. Interactive concept map for keywords and co-authorship in the database. Recent work found that node-link diagram has two kinds of significant downsides such as poor scalability for dense networks and an aggregated methods required to reduce the density of the citation network [14, 10]. CiteWiz is an article focused visualization. Overall they only tried to visualized citation network only.

Paperscape [13] is an interactive bubble chart visualization tool to visualize the scientific research papers. This tool visualizes all scientific articles published in *arXiv* (an open, online repository). It automatically extracts and analyses word frequency in the title and abstract of the article, and indicate the subject matter of that paper. It also worked with citation network by selecting an article. When a viewer clicks to the citation link, a network is created for the paper showing all the citing papers, but it overlaps the bubble chart. Therefore, it is hard to understand the citation network which paper has cited and when.

PubVis [16] is another web application helps to search by keywords and abstract based search options to find the relevant articles. This app includes such functions to obtain the abstracts from the PubMed, arXiv API and possible to add content from other sources to PubVis simply writing a custom scraper [16]. Data collection technique especially custom scraper is quite similar to our CitationVis application. Instead of a custom scraper, we developed an automatic scraper that can extract data from a dynamic website as well as the static website. Authors can only cite a paper for the references based on what they know. PubVis compare the actual texts of the published articles one to another for checking their similarities.

TextVis [22] is an interactive text visualization browser that is used for the visual survey of techniques used for getting an overview, finding related work based on various categories. Introduction to the subfield and gaining insight into research trends. This text visualization browser has a main view with a collection of thumbnails (ordered by time) represented the individual visualization technique as well as filter controlling that comprise text search field. A user can search any particular visualization technique by keyword search to look insight it. The searched result shows the corresponding details in

a dialog box, list of assigned category tags, bibliographic reference, and URL. Besides, it allows authors to add a new entry if needed but it should be sent administrators to avoid direct manipulation of the browser content.

Citeology [26] reveals the relationships between research publications through their use of citations. Authors collected the sample dataset papers published at ACM CHI and UIST and its all citation among them in the year of 1982 to 2010. This system organized the papers into vertical columns by year. Twenty-five characters of each title with very tiny font size displayed. The most cited articles of every year sorted in the middle of their respective columns. Therefore, highly cited papers of each year have found along a horizontal band throughout the center of the visualization. The connection between each paper and the referred papers were drawn. Hovering over any individual paper showed a tooltip with details information such as article title. By clicking on a single paper, highlighted the links in two color lines. Descendants of the paper were linked in blue lines and red lines for ancestors. This system overlaps the citations network, so it is hard to get the overview.

Imran Mehmood developed TrendVis for his master thesis, which is a web-based interactive application [27]. TrendVis was divided into two parts: Collecting the scientific publications metadata from well-known online platforms using an automatic scraper. Furthermore, the user can explore the data and reveal insights to hidden data. The interface can overview, search, group, filter results by doing simple interaction in the application. It helps to find the exciting trends and patterns from the scrapped metadata. Although, there are many new features can be added.

However, the different application used different methods for their data collection which were either manually, or an alternative way such as CDs, or web scraper. Later they explored the data via visualization process for their preferred research area. TrendVis [27] developed an automatic scraper for TVCG journals focused on scraping title, author, and keywords of articles except for citation details. Their web scraper works on static websites which are not dynamically created using JavaScript. We improved the existing web scraper and also programmed a new dynamic web scraper that can extract data from dynamic pages automatically, applied to TVCG journal.

Previously, keywords and abstract of the articles from TVCG journal were collected. Now, the dynamic scraper has used to collect citation data and visualize the citation information via interactive visualization. So, the user can interactively find citation trends over time.

After observing all of their visualizations, we have found that none of them have worked for article title based citation visualization. They have not done any trend analysis of citations of individual articles in TVCG journal over time. Neither they have presented citation comparison between multiple articles, nor visualize missing citation data for different years in the graph. Therefore, a user cannot analyze citation trends of

an article, compare articles on behalf of citation growth, and citation overview yearly. As a result, we have made our goal to developed CitationVis web-based application to analyze citation data only for TVCG articles. In the following chapter, we are going to discuss the overview of different aspected that are directly engaged to develop our CitationVis application.

3 Overview

In the first section of this overview chapter, information visualization (InfoVis) are explained briefly to give a better understanding how information visualization works. Other parts of this overview chapter are primarily the work overview that we have implemented.

3.1 Information Visualization

Information visualization is the study of visual images of abstract data that aims to help users in exploring, understanding, and analyzing data through progressive, iterative visual exploration. The origins of information visualization drive from several communities from the seventeenth century, leading the development of information visualization as a discipline [6]. The visualization pipeline is being followed for each visualization application. The Figure 3.1 shows the visualization pipeline. It helps the user to know the several stages of data transformation as well as get the sense of visual data. In the Figure 3.1 illustrates the arrows flow from raw data on the left to the human on the right, indicating a series of data transformations task. The indicators imply multiple chained processing throughout the visualization pipeline. A User-operation moderate these transformations.

At the beginning of the visualization pipeline, the raw data is a collection data. This data can be either structured or unstructured. The data transformations and analysis prepare data for visualization by applying filtering. Data reduction technique is used if the input data set is vast to fit into computer memory. For unstructured data, some data mining techniques such as clustering or categorization can be adapted to extract related structure data for visualization. With the structured data, this module removes noise by applying a smoothing filter, interpolating missing values, or correcting erroneous measurements usually computer centered with a little or no user interaction. The output of this module is sent to the filtering module, which automatically or semi-automatically selects data portions to be visualized (focus data). Given the results produced by the filtering module, the mapping module maps the focus data to geometric primitives (e.g., points, lines) and their attributes (e.g., color, position, size). With the rendering module, geometric data are transformed into image data. Users can then interact with

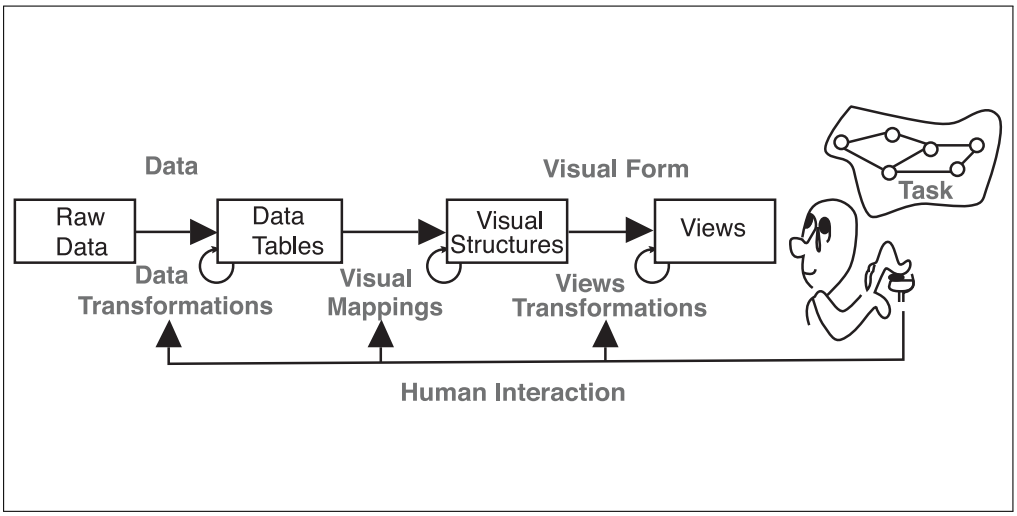

Figure 3.1: The Figure shows the model of visualization pipeline [6]. The pipeline describes from data transformation to visual form. Human interacts through the visualization process.

the generated image data through the various user interface (UI) controls to explore and understand the data from multiple perspectives [24].

3.2 Citation Indexing

Before leading to our developed CitationVis application, it would be worthwhile to have a basic understanding of the concept of citation indexing. Eugene Garfield[1] founded citation indexing in the sciences and scholarly journal literature. In which the cited references in every paper are written and serve as links between the article, making a chain of ideas and concepts that can be navigated backward and forward in time. Almost all the documents, notes, reviews, corrections, and correspondence published in scientific journals contain citations [12]. These citing references include the title, author, where and when released. Citations are the formal, exact connections among documents that have particular circumstances in general. A citation index is created encompassing those specific linkages. It indexes publications that have been cited and identifies the sources of the citations. Anyone conducting a literature search can find from one to many of additional papers on a subject related merely by acknowledging individual those have been quoted. Every article that is attained provides a list of new citations with which to continue the search [12].

[1] https://clarivate.com/dr-garfield/

However, many studies about citation indexing, citation network, co-citations relationship, etc. was done including its visualization [12, 17, 30]. But the visualization of citation trends is missing, particularly if we look to the IEEE (TVCG) journal. So, CitationVis application aims to scrape all IEEE citations from citation columns automatically as new citations appear. Then, represent the citation details in graphical form by using CitationVis with simple user interaction.

3.3 Database Selection

A database selection depends on what type of data should be stored. A database can be divided into a relational database and non-relational database. A database is basically containers of data. The relational database management systems (RDMS) include SQL databases such as MySQL whereas a non-relational database is NoSQL, object-oriented database such as MongoDB. The RDMS (MySQL) database store data in rows and columns, which makes up tables and set of tables makes a schema, and some schemas create a database. On the other hand, NoSQL (MongoDB) database does not follow the RDMS approach. However, Both database approach can store a large amount of data. Since MySQL database is a relational database and our citation data has a lot of relation between them, so the MySQL database fits for our data. Also, normalization helps to break up data into the smallest possible parts. MySQL is free of cost, well suited for the web application to retrieve data from the database, and access complex information [32]. Other software such as PostgreSQL could be used for our data storage. But, We preferred MySQL as a conventional database system for citation data. The content of citation references in IEEE (TVCG) website is unstructured data. Normalization is applied to MySQL database to maintain the redundancy or duplicate data and improve data integrity in our citation data. In chapter four, we explain the selected database in details.

3.4 Web Scraper

Web Scraping is a term that is called web harvesting, or web data extraction method used to collect required data from websites [4]. When the number of data is too big, and therefore an automated web scraping approach is needed to extract data from the website. Theoretically, web scraping is a process of collecting data through any means other than a program interacting with an API (or, apparently, through a human using a web browser) [28]. Whereby the data is extracted and later save to a local computer or in a database. A web scraper aim is to gather information from websites automatically. Web scraper can be divided into two categories such as static web scraper and dynamic

web scraper. If the target contents are in plain HTML website, then a static web scraper can be used created by Node.js (Cheerio) parser or any other framework or technology. There are many dynamic websites which means that data is loaded when a JavaScript AJAX request sends to the server. Or a website where the content is dynamically created on the server is also called a dynamic webpage even though it looks like a static website for the client. For a dynamic website, a dynamic scraper can perform to extract the data. We developed a dynamic scraper built by Node.js (Nightmare.js.js) for scraping data from dynamic websites.

In general, web scraping first downloads the web page. Then web scraper traverses to web pages to extract data which are specified to the scraper. For that, the scraper should understand data structure on the web. Data contains inside HTML document. The scraper has to traverse the Document Object Model (DOM) within the HTML page and selects the nodes or elements then scrape the text within DOM. As we know web scraping has two different approaches for scraping data from websites. It depends how the data are available on the website. If the data are in static, then static web scraper works fine but for the dynamic website a dynamic scraper needs. We extended the existing TrendVis application [27]. TrendVis used static web scraping procedure.

Website functionality has improved to dynamic from static which means every information of a webpage might not be available but only shows once we click on a particular link. Then this click sends a TCP Get-request to the server via JavaScript, AJAX call. CitationVis interactive application has selected IEEE (TVCG) journal pages for citation data as a data source. This webpage is built as a dynamic website by using AngularJs. This AngularJs is a JavaScript-based front-end web application framework. In this case, a dynamic website scraper is programmed using Nightmare.js.js library in Node.js. JavaScript and jQuery libraries can get access, able to manipulate the DOM inside of a web browser. So, writing web scraping using in Node.js is convenience since many methods can be used for DOM manipulation in the client-side code for the web browser [25].

3.5 CitationVis

CitationVis is a complete tool package developed for data collection from websites automatically and visualize them using modern visualization techniques. This application is divided into two parts. The first part, demonstrate a dynamic scraper to collect information from dynamic websites. In our project, DBLP and IEEE (TVCG) are selected as a data source. CitationVis application focuses on cited information in each article of TVCG journals. As our data source is from the dynamic webpage, CitationVis web scraper also developed to scrape dynamically. However, if someone wants to extract data from any other websites, they can quickly do it. The database and scraper are

extendible. In data gathering process, all the citation bibliographic data of articles scrapped, published in TVCG journal which contains the metadata of papers between 1995-2018. One of the key features of our dynamic web scraper is that it scrapes data over time and in an automated way. Therefore, the increment of any trends or patterns of metadata can be observed. The Figure 3.2 demonstrates the technique of CitationVis web scraping metadata of articles, published in TVCG journals.

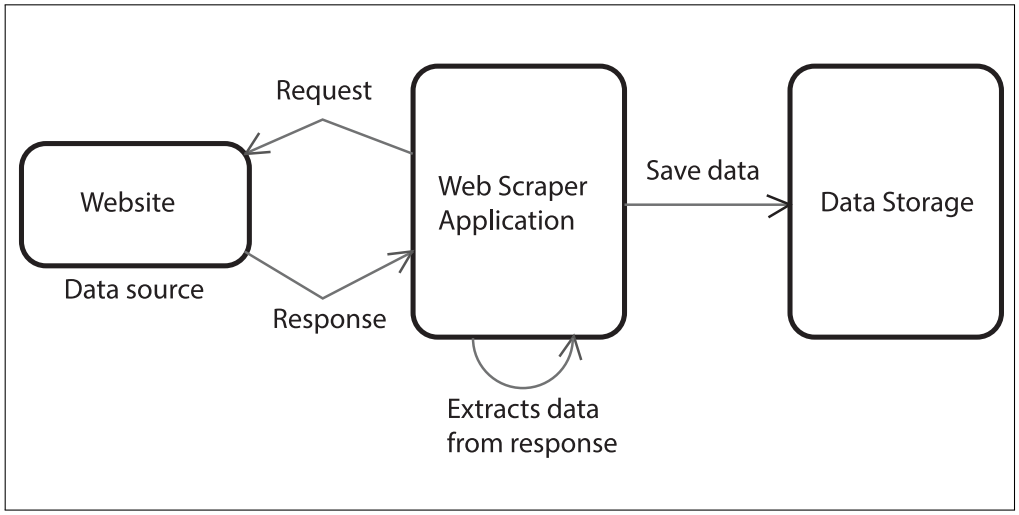

Figure 3.2: Dynamic web scraping request to get URLs and pass info to the scraper. All URLs are selected to scrape each citation details of an article. Finally, scraper forwards the extracted data to MySQL database. After storing citation data for an article, the scraper process the next URL and follow the same steps recursively.

Initially, dynamic web scraper send a request (1) to the data source DBLP to get all the URLs of IEEE (TVCG) journal. All the URLs of articles are selected then return (2) to the scraper. Now each article URL pass to extract the citation data (3). In this stages, scraper visits every citation links and retrieve the specific data. This process continues recursively until unless all the citation links finished scraping. Finally, the scraped data forward to a local database to store in their specified columns (4). In this work purpose, we have selected MySQL database system. After collecting citation data for one article page, The scraping process again starts to scrape the next article URL. This scraping procedure repeatedly runs until all URLs scraped. This is the benefit of a dynamic web scraper that the dynamic scraping procedure can traverse multiple links, going through multiple internal clicks on websites. In Chapter 4, the data collection process with the CitationVis web scraper has explained very explicitly.

In the second part of CitationVis application, the collected data are visualized using various visualization form. The Figure 3.3 shows an overview of CitationVis visualization

procedure. In short, data were inserted into publication database in MySQL database. CitationVis application sends a request to retrieve the particular data. Once the appeal is successful, and data are available then it visualizes in histogram chart or line chart. In this project, only two kinds of visualization technique have applied, but other visualizations can also be implemented as CitationVis a flexible application, but it will require the basic understanding of Data-Driven Documents (D3) libraries as well as JavaScipt knowledge.

We have focused article title search in visualization, but there might be more possibilities to search namely searching by author. The principal purpose of developing this web-based interactive application is to visualize the data changes over time for the published article citations. With the simple interaction in this application, a user can find the trends yearly basis from the published data of any articles or the comparison between two or multiple papers. Therefore, a user can get the overview of citation result. The concept of CitationVis is similar to visual analysis as it also visualizes the trends and possible for the analysis of citation data for articles.

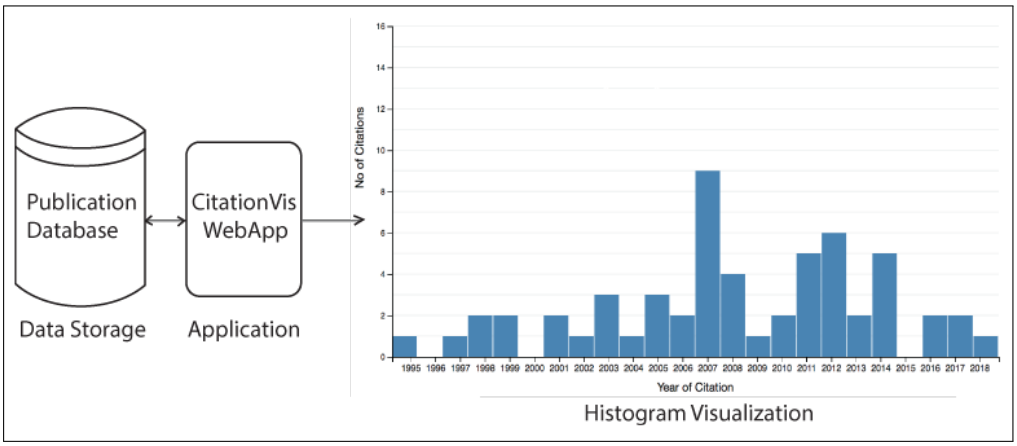

Figure 3.3: CitationVis Visualization Process

If we go through the bullet points below, it would be easier to understand the application user interface on the current trend analysis of citation bibliography of the published papers in TVCG journal.

- How a dynamic web scraping helps:

The challenges arise when scraper need to traverse to different links within a page and data are only available with multiple clicks. If the data are continuously changing or only possible with JavaScript, jQuery, AJAX call. However, our sophisticated dynamic web scraper is built to traverse different links and wait to load the page. On each running time, the scraper goes through each link check for any updated information in citation

data column. If new data appear, then extract the citation data otherwise forward to next URL. Therefore, a user can get the updated data that are available. Otherwise manually data collection would be difficult for exploring their trends and time-consuming.

- The cited paper published in TVCG journal:

If a user wants to know the total citation count of an article then can know it just by visiting an article page. Since they are in text form, it requires clicking every link to understand it user case. A user might want to know who is citing a paper. The more a topic is being referred to another article, indicates the subject is essential or being discussed in that research field. However, A user can find the trends doing a simple interaction with CitationVis application.

- The citation trends of a paper are increasing:

TVCG journal started publishing papers from 1995, and they are regularly releasing. If an article published in the 1995 and want to analysis the citation details, then a user can find the citation trends fluctuation yearly over the time interval in our histogram graph. It is also possible to know who cited and when. Therefore, it is possible to monitor how the research activities in specific fields are changing over time.

3.6 Tools

The Figure 3.4 shows the tools have been used to develop CitationVis application. It provides an overview of CitationVis application workflow. Initially, we planned which data to scrape from where. DBLP and IEEE were selected as our data source. We developed a dynamic web scraper that can scrape data from their website automatically, and store the data into relational database. We used Node.js and Nightmare.js.js to built the scraper and MySQL database stored the extracted data. PHP scripts are executed on the server. Finally, the extracted data is visualized running on the browser. As a frontend HTML, CSS, JS, and D3 are used.

Data Source: We used DBLP and IEEE for our data source.

Node.js and Nightmare.js.js: We developed a dynamic web scraper using both tools.

MySQL: We used MySQL database to store and retrieve data for visualization.

PHP: We used PHP as Server-side scripting.

HTML: We used HTML to annotate text.

CSS: We used CSS to style web application that includes, layout, fonts, and color.

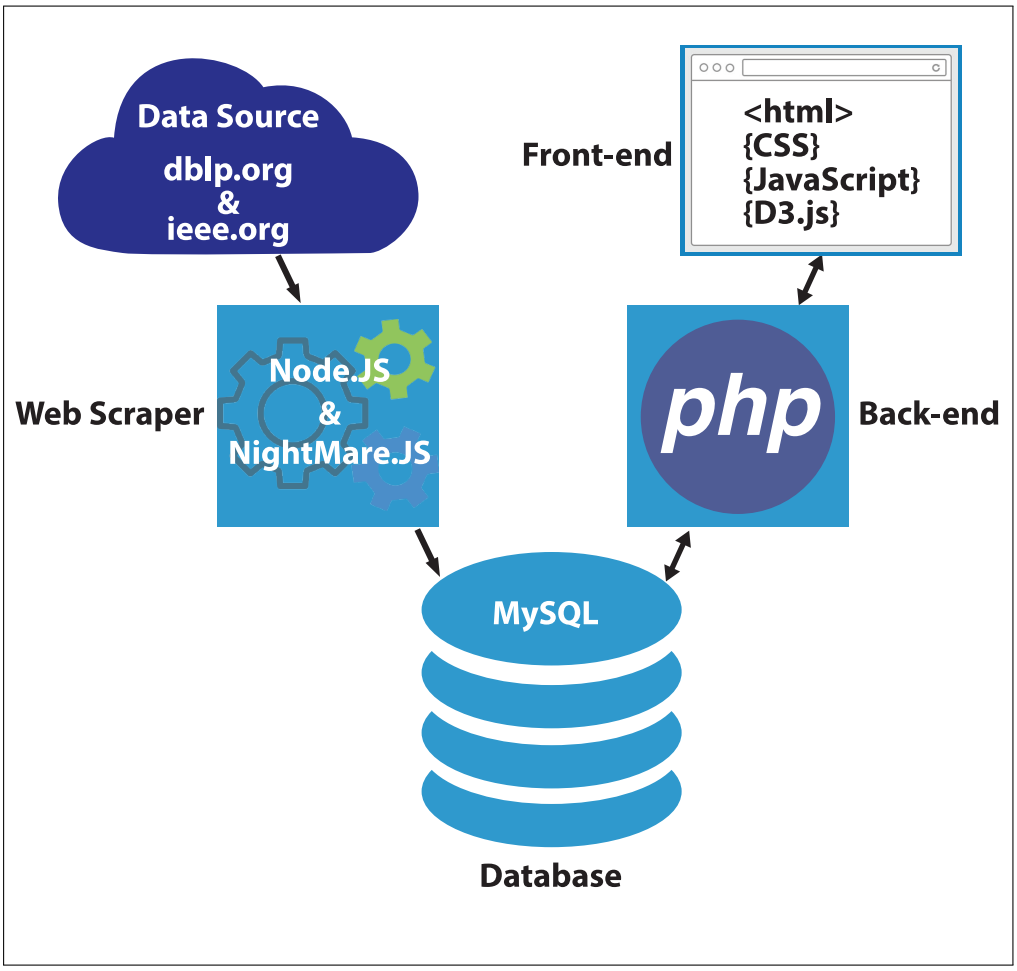

Figure 3.4: The Figure shows tools used for CitationVis application. It shows the data-source and which tools is used for what purpose.

JavaScript: We used JavaScript to make the application interactive.

D3: We used D3 JavaScript library for visualizing data in graphical form.

4 Citation Data Collection

We are going to discuss the data collection process using dynamic web scraper. We have maintained several essential things that are mandatory while the data collection process. We observed data format in the data source and prepared our database MySQL system and the insertion process. We have collected citation data in order to visualize the citation trends in time-series via an interactive visual form. We organized and maintained the database in such a way that we can repeatedly reuse the data for exploration graphically. In next part, we are going to talk about data collection process we followed such as data preprocessing, cleaning, inserting, storing. We have used both static web scraping and dynamic web scraping for extracting data. Citation data is collected from TVCG journal of IEEE association. We have built the dynamic web scraper so that it can traverse HTML DOM for a dynamic website and extract citation data that we require for visualization. We found that the TVCG journal contains unstructured data. Data is not the pre-defined data model, loosely structured data. As a result, we applied data transformation procedure such as filtering, removing unwanted data. Once the data is cleaned and ready to store in the database, we inserted the data into MySQL relational database. Later, this data is explored via visualization using our CitationVis interactive web application.

4.1 Database Design

We extended the existing database structure defined by Mehmood [27] and modified the database design to include citation data. This database designed followed by third normal form (3NF) which satisfied and normalized database constraints.

The citation data of each article was missing in the given database. So, for citation trends analysis we need citation information about an article. Having the table serve many purposes introduces many of the challenges; namely, data duplication, data update issues, and increased effort to query data. These anomalies can be discharged or reduced by correctly separating the data into different tables, to reuse the data in tables which serve a single purpose. For citation data to be in 3NF, we need to remove the transitive dependencies. Therefore, we could not include the citation data into existing table. As a result, we created another table that serves as a child table of the

given article table by making the one-to-many relationship. The Entity-Relationship (ER) Figure 4.1 shows the relationship between article table and citation table.

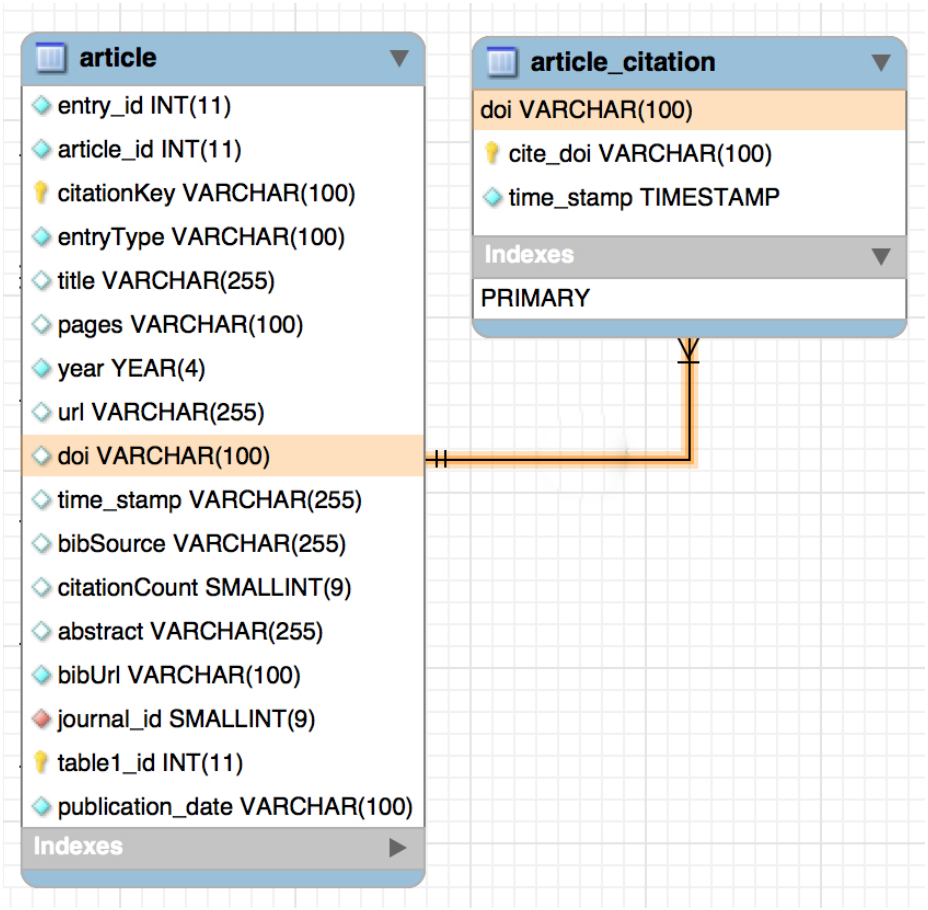

Figure 4.1: The Entity-Relationship (ER) Figure shows the one-to-many relationship between article table and citation table of the database. DOI serves as attribute keys in both tables for connecting the relationship for citation of each article.

Every article has a unique Digital Object Identifier (DOI) and is cited by an arbitrary amount of other articles, also identified using their DOI. Therefore, we created the relationship between article table and citation table as the one-to-many relationship. The line between the two entities in the Figure 4.1 represents the relationship. The line notation style between two tables shows that the endpoint of the article is one, and only one but the citation table line notation shows many, having a set of three lines. We created the relationship with a common attribute between two tables, and DOI is key attributes. This DOI key attributes we created as primary key in citation table and foreign

key in article table. Their relationship is maintained by connecting the two attributes via these key attributes.

Citation data contains the title of article, year, DOI, URL of the article published in TVCG. This data also is in the given database in article table, but they are not in the relationship for citation of an article. This information keeps the record of the article only. It does not have any relation with citation data of another article. This is also another reason to create a separate table where only DOI of the paper insert for identifying individual article and all the citation DOI of articles are inserted against that article.

The given database schema also missed scraping the publication date for each article. We needed to scrape it because we also want to show the exact date when the citation happened for the selected article. So, we modified the given scraper to extract the publication date and store into the article table in the database.

4.1.1 Data Preprocessing and Cleaning

Data insertion is only possible when the database constraints satisfied, and insertion process followed according to database constraints. Due to unstructured data in TVCG journal, citation data has some discrepancies and inconsistencies in the citation data. The discrepancies might come from data for their inconsistent, incomplete or ambiguities. DBLP contains TVCG article data in BibText format. For citation trends visualization, article title and year are selected which collected from DBLP repository. On the other hand, publication date and DOI collected from IEEE websites. DBLP data already in JavaScript Object Notion (JSON) format so we didn't need to convert them. But the data that we have scraped from TVCG converted to JSON format after the collection. Data processing and cleaning are necessary to be able to store it and make the data ready for reusing to the visualization. We have used D3 [1] JavaScript library for citation trends visualization. D3 has built-in functionality that accept different data formats namely XML document fragment, Comma-Separated values (CSV), Tab-Separated Values (TSV) or JavaScript Object Notion (JSON). We have parsed BibTex data to JSON and also TVCG data to JSON format so that we can use them via D3 visualization method. There is a parser called bib2json [2] in Node.js, and it helps to convert to JSON format. Now the data in comma-separated in an array and it is easily accessible to the data object.

After parsing the data, we have to make sure that the data is clean, which means there might have some special characters or symbols from the raw data. So, we implemented a JavaScript function that fixes the special character or symboling issues in the data. This function is performed in all articles. We found some articles has no citation data or missing the links. But the article URL is there, or the data links incorrectly located. In

[1] https://d3js.org
[2] https://www.npmjs.com/package/bib2json

that case, we needed to make sure that articles without any citations are appropriately stored by explicitly marking missing links such as null. When we visualize citation chart, if there is no citation data for an article then it treats as that year has no citation or those are initialized as zero values in visualization. Node.js unique module manages this and successfully passed the array.

4.1.2 Data Insertion and Storage

We want to retrieve data from MySQL database, and we can query SQL efficiently to explore the data. We can also store the parsed data in a JSON file, but there is some limitation. JSON file cannot store dates, undefined data and also it cannot save the data permanently. JSON performs well while exchanging information at a server level. As we want the data to be consistent and updated, so we need to store it into a relational database that makes data insertion and updating fast and easier. Also, we need to operate some join SQL functions to create some relation between the different tables in the database. Later, it is easier to perform several SQL operations to retrieve the data.

For our user case, we created a database in MySQL followed by TrendVis database schema [27]. We named publication of the database and several tables designed according to its database schema. For citation trends visualization, we included another column in existing article table for the publication date of each article. We created a new table called article citation where each citation DOI are recorded which belongs to the article. If a paper has no citation, still we inserted its DOI and the citation DOI column as null so that we can even visualize in charts if anyone wants to compare different articles for their citation. All parsed data stored in a separate table as it required. Our data was parsed and scraped in JSON format, and later we inserted into the database in different tables to avoid inconsistency, duplication of data.

As the data format is unstructured in TVCG, data is preprocessed and cleaned before inserting into the database. While we scraped citation data, at first, we called all the URLs from the database through JavaScript query. We presumed the existing database maintained its constraints with different tables. Article URLs are stored in article table, so we referred article table as a parent table and citation table as child table making the DOI column between two tables as the primary key and foreign key. When we inserted an article citation references, first scraped the main article DOI and then all the citation DOI for that article. Since citation data continuously changes over time, the insert statement always checks for duplicate article citation DOI, and if the DOI value is already inserted, then it will not be inserted again, instead only the new record is inserted. The query checks for duplicate using primary key and if a record is already in the database, then only citation count is updated and otherwise a new record inserted.

First, we selected all the URLs from article table then all URLs are stored into an array set, so it avoids duplicate of URL if there are any. We iterate through each URL and send an HTTP request using that URL of the article. Web page data load on click events, so we traverse through each element and click the required button and wait for the web page to load data then we scrape the article data. This data is stored in the MySQL database. The database constraints ensure on each insertion. This scraping process repeats for each URL of the article. Whenever we run the dynamic web scraper, it checks the article URL, if it is already scraped then pass to the new article URL, but if any citation updates for any article, then this record is also scraped and insert to the database.

4.1.3 Database Normalization

We scraped all citation data from each article in TVCG journal. There is some duplicate data rarely found within citation data. So, normalization is necessary for the database. It requires for mainly three reasons for a database. To reduce redundancy of data, minimize or avoid data modification issues, and improves data integrity. Also, normalization is used to ignore data anomalies such as insertion anomaly, deletion anomaly, update anomaly. Citation data is continuously changing over time, so normalization is essential to avoid data modifcation issues and purify the data integrity in the database. From the existing database schema of TrendVis [27], if we store all of the citation data in article table, then it violates data integrity. So, we created another table called article citation for citation DOI information only. This table has one-to-many relationship with article table to article citation table. Therefore, we could separate all citation DOI for an article DOI. There are many normalization forms used in MySQL such as Unnormalized form, First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd normal form. However, we followed 3NF normalization process in our database design.

4.2 Structure of Data Collected from TVCG

Data can be structured, semi-structured, or unstructured used in web. The different scientific journal publishes books, articles, research paper and manuscript followed by their desired data structured. In this project, we have worked with citation data of each article published in TVCG journal. TVCG journal is a popular journal of IEEE association. TVCG articles metadata are also contained in DBLP online repository. The TVCG data in DBLP web page is in BibTeX format. This TVCG journal is continuously publishing papers regarding computer graphics and visualization from 1995 to the present. For our work, we collected around 3050 articles metadata till January 2018. We have found

that the TVCG articles were cited by each other approximately 27450 times within IEEE association.

Unstructured data is dates, numbers, special character in data. Articles published in TVCG has the unstructured data. That is why a data preprocessing and cleaning has been done. The publication database also has several tables, such as author table, keywords table. These data also stored in the database but ignored while implementing the citation trends visualization. Keywords of the articles are out of our interest as well. The other information such as article title, year, publication data, and DOI we have used because this information is directly related to citation trends visualization.

4.3 Web Scraping

There are a variety of techniques used to scrape information from a website. Automatic web scraping also allows for defining whether the process should run at regular intervals to capture changes in the data [1]. In this section, a general web scraping procedure is discussed, and in next part, a dynamic web scraping is discussed. First, we have to know the website architecture. We can see the architecture of any website by inspecting the elements of the website using a web browser. Most of the web browser such as Google Chrome, Safari, and Firefox has an inspect element option. By investigating the HTML page, we can understand the DOM. We can access DOM elements by selecting div class name, id or tag name. If the contents are in static HTML format, then a static web scraping works fine. For this static scraping Node.js Request and Cheerio, objects can be used. This request object sends HTTP request, and then Cheerio objects download the HTML document. The web content can be collected if one can traverse DOM. Traversing throughout the DOM is done by jQuery elements manipulation. jQuery makes it easy. Now the several classes, ids or tags can be accessed using the DOM traversal functions.

Some specialized libraries are already available. We can modify the libraries and write the script for web scraping that will automate the web scraping process for instance Cheerio for Node.js and Beautiful Soup for Python. For the TrendVis [27], Node.js (Cheerio) framework used since it was static web scraping process. We used Node.js (Nightmare.js.js) framework to scrape dynamic contents from the dynamic website. We have selected Node.js because it helps to insert data into the database. We have installed Node.js package by Node Package Manager (NPM) including the necessary modules for web scraping. Node.js (Cheerio) package installed to extract the publication date of cited papers which was a missing part of the existing scraper. Now, we have all the metadata information such as article title, URL, year, DOI, and publication date.

4.3.1 Web Scraping with Node.js

In Node.js web scraping procedure starts with a base URL, which is selected first scraper function that we want to search. It obtains all the URLs of all volumes of TVCG journal from DBLP website. Then the next function scrapes the URL for every year articles. Finally, the last function extracts the required data from each article URL. Later, data is stored in a MySQL database.

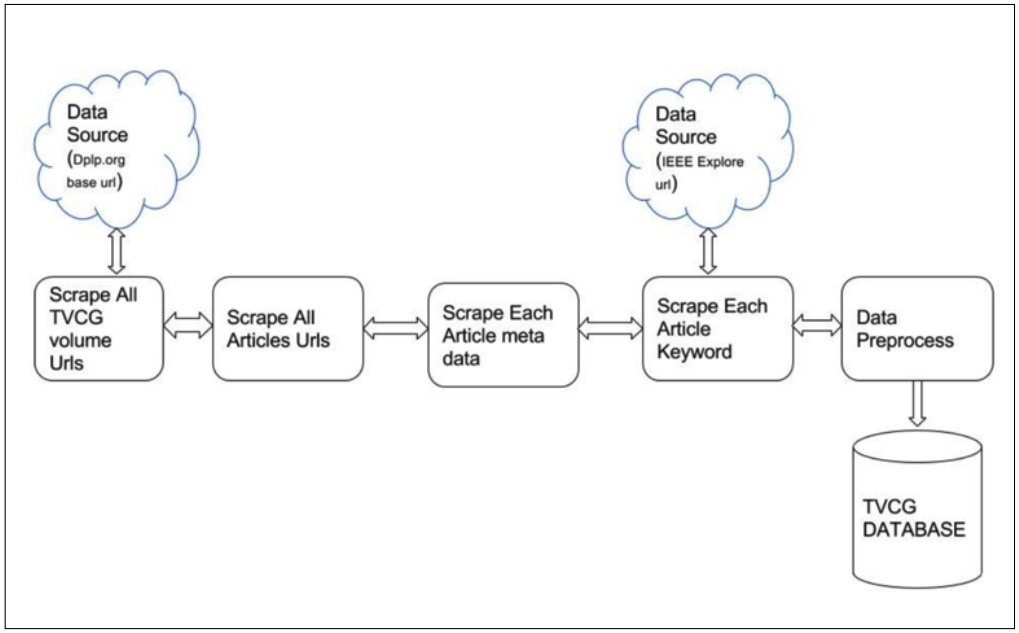

Figure 4.2: The Figure shows web scraping process using Node.js in details. Initially, the web scraper collects data from DBLP.org and then use article URL to extract data from IEEE Explore.org. Lastly data preprocessed and stored in TVCG database [27].

The Figure 4.2 is a web scraping procedure of TrendVis [27]. This web scraping sends a request object to the HTTP request, and then the HTML DOM is parsed by Cheerio. Then jQuery is used to extract data from the parsed HTML. DBLP and IEEE (TVCG) journal are the data source, but the architecture of the website is different. Therefore, different scraping functions applied to get data from both websites. Scraper scraped all URL in DBLP [3] webpage and then Cheerio parsed the HTML DOM. After that jQuery method applied to traverse DOM and collect the URLs of several volumes of TVCG journal. TVCG journal is publishing papers since 1995. There are total 24 years of metadata saved.

[3] http://DBLP.org/db/journals/TVCG/index.html

Static web scraper module/library in Node.js	
List of Module/Library:	Implemented for:
Request Module	Helps to load HTML document of web page
Cheerio Module	Helps to parse HTML DOM of web page
bib2json	Helps to parse BibTex to JSON data within JavaScript
Async	Helps to iterate through a loop asynchronously
Unique	Helps to remove duplicate from array
JSON parser	Helps to parse JSON data
Throttle-request	Helps to add delay between two requests

Table 4.1: Shows the list of modules that were used for static web scraping.

The Table 4.1 shows the modules used to create the static web scraper for specified data extraction. However, we have extended the existing scraper to get more data considering the citation details. We had to go through the same recursive process and insert the data in MySQL database and includes the publication date of each article from TVCG journal. We have been focusing visualization for citation information, so we need the article publication date from IEEE (TVCG) journal. This is our extension work of existing TrendVis application. Thus, we have not created a new database scheme. We simply inserted the scheme and created the database using MySQL SQL command. We have scraped the data including publication date by rewriting the scraper.

4.3.2 Dynamic Web Scraping with Nightmare.js.js

The existing web scraper has the functionality to collect most of the metadata of articles from two different data source. Each scraped article includes title, author, keywords, year, URL, DOI, [4], and citation count. Both of the websites contain TVCG journal articles. DBLP includes the article's data in BibTeX format. However, the keywords were scraped from IEEE website. Later merged the data and inserted into MySQL database.

Our goal is to visualize the trends of citation data for each article and also compare different articles on behalf of the number of citations yearly. Therefore, we have considered the article title, year, and URL from the existing database. We also need the publication date of all articles as we want to show the actual date of citation occurred via graphical visualization. The publication date inserted into publication date column for their specified articles. Almost all information (i.e., title, year, URL, publication date, DOI) of an article is available in the current database. If a paper is selected, then possibly we can know the information about that paper. The paper information can be such as title name, year, URL, publication date, and DOI. But it is impossible to distinguish which articles belong to this paper citation. Because of all information of the articles

[4] http://www.DOI.org

inserted into the database as their individual entity. There is no citation relation between one article to another article for the citation. Therefore, we simply cannot explore the citation information via visualization. That is why URLs of the articles from database selected and pass to scrape its citation data. Article citations in TVCG journal contain author names, paper title, journal name, volume, article link and article in pdf format. We need a unique identifier which cannot be duplicated by any means in the citation data for an article. We have found DOI is the unique key which identifies only one article. Other information such as author name and year of publication can have duplicate value. So, DOI is targeted value for our citation data references to scrape using the dynamic web scraper. This DOI is only available when we click on the view article links. For retrieving DOI of each cited articles, we have to visit each article page and then scrape it. However, this could not be possible by using static web scraping process. As a result, we have developed a dynamic web scraper using Nightmare.js.js (A high-level browser automation library) [5] which get first all the article URLs from local TVCG database.

Dynamic web scraper module/library for Nightmare.js.js	
List of Module/Library:	Used for:
Nightmare.js.js	A sophisticated high-level browser automation library. Each method is written in simple English command for example: goto, wait, click
Vo.js	It supports to control asynchronous flows either pipeline or stacks
Async	This module allow other input/output processing before the first processing has finished
Promise	It is useful for chaining multiple steps together using '.then()' method
Jsdom	JSDOM parse the HTML and useful for scraping
Electronjs	Used in Node.js and as our Chromium browser

Table 4.2: Shows the list of libraries/modules that have been used to create an automatic dynamic web scraper.

The Table 4.2 shows the libraries and modules used to create the dynamic web scraper for specified data extraction.

By using dynamic web scraper, we extracted all article URLs from local database first and pass into an array set. From the array set, every URL is forwarded as argument to next function. Nightmare.goto then send a request via HTTP request to load the article web page and browse the page via Electron Chromium browser which is built in function in Nightmare.js A wait function is applied to fully load the page to avoid blocking from the website. The minimum ideal wait time to load a page sets to 3000 milliseconds. Waiting time function is applied for every JavaScript, AJAX call to load the page. The chromium

[5] http://www.Nightmare.js.js.org

browser is used to load the web page since the citation data is only loaded when a JavaScript, AJAX call. A function is set like if the target link is exist then go to next function otherwise terminate the Nightmare.js.js and callback to the Nightmare.js.goto function again for next URL from stored URLs in array set.

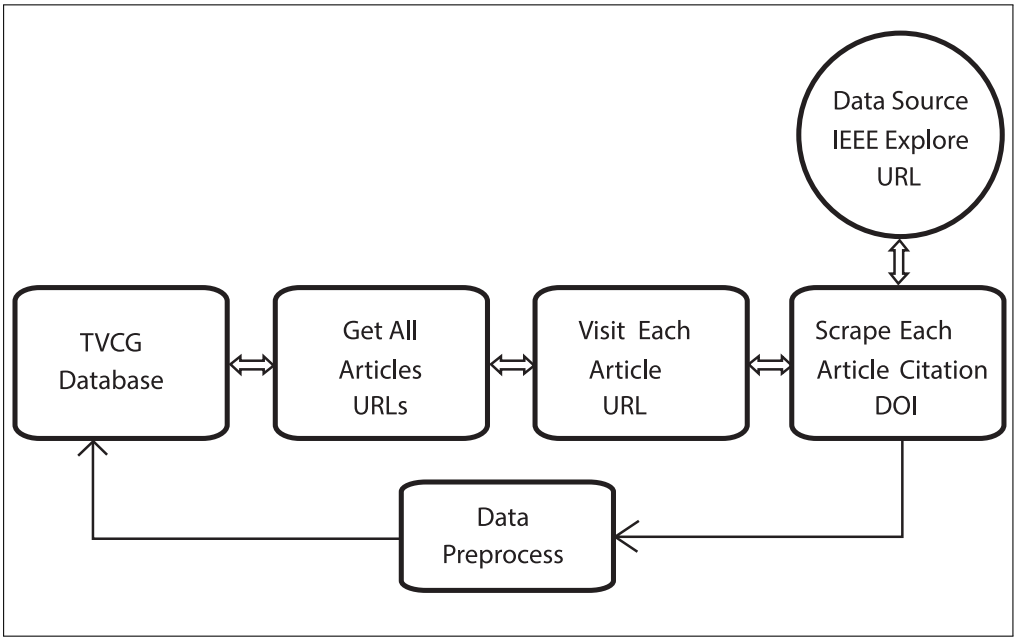

Figure 4.3: A dynamic web scraping process for CitationVis application. It collects all article URLs from existing database and passes into array set. Visit each article and click on citation links. Then scrape DOI for every cited article from their article page. The scraping process continues until all DOIs are extracted from TVCG articles. Finally, proceed the cite DOI to insert into TVCG database in citation table.

Once the scraper gets the target destination of all citation links, it starts to evaluate each citation links. Nightmare.js evaluates and visit every cited URL and scrape the DOI from their article page. This evaluation runs until all DOI is scraped from all citation articles. If no citation then it again callback to next function. After scraping all DOI parsing through HTML DOM, DOI data inserted into the database that belongs to article URL. Nightmare.js iterates through each step and scrape all DOI repeatedly. If no article URL is left, then the scraper stop. This scraping process recursively happens when we run the dynamic web scraper.

IEEE has two type of citation sources. IEEE websites itself citation reference and from other sources. However, we have considered only IEEE citations. We covered in this chapter, the database creation, scraping, storing and making the citation data ready to visualize.

5 Citation Visualization

This chapter, we are going to discuss citation trends visualization with CitationVis application. We will also describe the visualization libraries that we used.

Earlier, no citation visualization method concerned for citation trend analysis based on article title search. So, concerning on citation trend analysis, we have tried to visualize citation trends searched by article title. Searching any article by title helps to autocompletion search, visualize the full overview of citations. Compare citation between multiple article citation, the citation trend changes. Also, citation links with exact citation date are available to browse the cited articles. The citation data is flexible to visualize different perspective such as citation comparison of authors. However, the stored citation data can be used to explore any other hidden insight of citation data later.

Data visualization depends on how the data can be accessed and visualize it with the minimum user effort. For any data visualization, data should be ready so that the user can use the intuitive visualization to explore insight of the data. As we want to visualize citation trends of articles published in TVCG journal, we developed CitationVis interactive web-based application. CitationVis application offers full freedom to the user to interact with the application so that the user can discover hidden information insight the citation data.

We would like to remind the data visualization pipeline again. The Figure 3.1 shows the data visualization process. We can divide the data visualization pipeline into several steps such as data transformation, filtering, mapping, and rendering into visual form. Data extraction, collection, and mapping are done. The visual form is ready to represent data into some visual methods by human interaction. There are many open source libraries for data visualization such as Dygraphs.js, InfoVis, and Polymaps.js but we have used D3 JavaScript library. A human user can interact with the application and visualization technique will proceed the data graphically.

In general, data visualization purpose is to reduce cognitive load, extract data as much as possible but keep the information understandable, gather knowledge from visualization without going through reading a lot of text content. We are endeavoring to do the same thing with our citation data visualization. We know citation data is in textual format in TVCG journal. We collected data only from the article of IEEE, and there are approximately 27,453 times articles cited each other for 3,050 articles from 1995 to 2018. We selected histogram chart to visualize changes in citations over the years.

A histogram is an obvious choice to overview changes in data over time. It helps to compare data of single article over time. We have mainly two visualizations a histogram chart and line chart. Both visualizations are interactive, so a user can see details and can filter results using year filter.

As we know, the size of citation data is increasing, and it is not possible to find trends in a large dataset. So, the visualization help to visualize such trends in data. Knowledge acquisition from the given text information and discovering new insights from these textual form is not useful in practice, especially for the novice users. Therefore, we want to explore the same citation information using interactive visual form, it becomes expressive, interesting and quickly interpreted to the user. We know a picture can render the overview of the data quickly rather than reading textual data. But pictorial representation will only be informative when we can choose the right visualization method for the data. When we want to visualize time-series data, a line graph is the simplest way to represent. As line graph is intuitive, easy to understand and helps the user to get a quick perception of how something has changed over time.

A user can compare citations between different articles. An excellent visual representation helps to avoid vast language processing, also reduces cognitive work to perform certain exploration [20]. So deciding a right visualization method is necessary otherwise a visualization technique may misinterpret information about the data. So, CitationVis developed with an aim to help the user find interesting patterns and hidden trends in citation data over time. The user can overview the changes in citation data, compare results and can explore details of each citation. The details include the citer, year, author, and URL link. This visualization will reduce cognitive loads and quickly explore insight citation details of articles.

We used histogram visualization to show the total citation of an article over the years. A user can interact histogram graph and get more information about its citation changes for different years. Besides, a user can get an overall idea for the different year of citation of the selected article. On the other hand, we used the line chart to display the trends of citations over time-series. Also, citation comparison between papers gives another interesting overview. Next section, we are going to discuss data visualization using D3 JavaScript library.

5.1 Data Visualization with D3

D3 is a novel representation approach that exposes the web standards for the visualization on web [5]. We may get several open source data visualization libraries namely

D3, InfoVis.js, Dygraphs.js [1] [2] [3]. However, we used D3 library since It is most powerful visualization library that provides dynamic and user interactive data visualization. D3 is JavaScript library for the client-side visualization process and works as a wrapper around the API to the DOM and Scaleable Vector Graphics (SVG) on web-based visualization. Mike Bostock [5] developed D3 JavaScript library. It helps for manipulating documents based data into visualization.

D3 JavaScript library uses the functional style and many built-in frameworks that makes it reusable the code for developing the visualization. In D3, data can be specified as an array of values, and every value is forwarded as the first argument to selection functions. Fundamentally, D3 can nest data, group by, manipulate data taking a flat data structure with a minimal amount of code. We have visualized citation data on a webpage with user interaction. D3 allows the user to bind arbitrary data to DOM elements and then the data-driven transformation is done to the document. D3 is used in the front-end of our CitationVis web application, and the back-end offers a way to get the necessary data. We used D3 library to develop our CitationVis application where we have visualized two kinds of visualization procedures, and these are Histogram Chart and Line Chart.

Histogram chart or graph will allow the user to explore the underlying frequency distribution of citation data of an article from 1995 to 2018 year of range. We have selected Histogram Chart visualization that is used by D3 because histogram chart can plot the citation data frequency in a continuous data set over time. The Figure 5.1 shows the hight of the block of the Histogram graph as the total number of citation received in a year for an article over the time interval.

Therefore, we can get the entire overview of all cited articles for the selected article in time-series. As a result, the user can get the whole overview of each year citation details. If no citation occurred in a particular year, then this year block will have zero value. So, the user will not miss any citation details even though there was no citation. On the other hand, The line chart shows the trends of getting the citation for a selected paper over time. The user can also select multiple articles and compare their receiving citations.

5.2 Visual Prototyping

Before visualizing our citation data using D3, we have drawn a visual prototype of the desired web application. The Figure 5.2 shows the plan we made before implementing

[1] https://d3js.org
[2] http://philogb.github.io/jit
[3] http://dygraphs.com

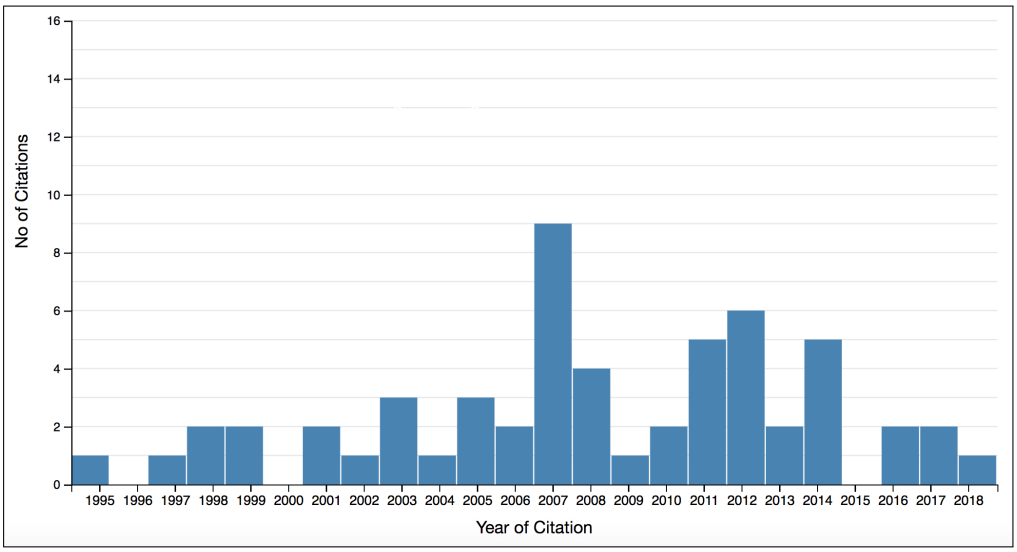

Figure 5.1: The Figure shows a histogram graph of several blocks of citation occurred in the different year. The hight of each block indicates the total number of citation received in a year for a selected article over the time interval.

the final visualization. We did the visual prototype design using Adobe Photoshop. Visual prototyping is nothing but is a model of our concept to visualize the citation data.

With the visual prototype, we tried to initiate how the actual visualization will look like in the final application. In this visual prototype, we decided to show all the papers in bubble chart where every bubble point an article published in TVCG journal. In Figure 5.2 implies the bubble size and color that describes the number of citation received for that article. If we select a bubble point, then another graph will be created using its citation network. Also, we could get a line chart showing the trends of the citation of the chosen article received. However, Due to a large number of articles published in TVCG journal within the year of 1995 to 2018, the idea of bubble chart did not fulfill the concept of changes in citation data over time, and the citation network could not have an impressive concept showing all the papers together. Some article could be missing if the article has not received any citation. But the line chart makes sense indicating the number of citations received for any specific article over time. So, we finally implemented the line chart concept and also histogram chart for illustrating the citation trends. We plan to visualize the actual citation trends of articles in time interval rather than visualizing citation network. However, this pilot experiment with bubble chart visualization helped us to determine with line chart and histogram visualization.

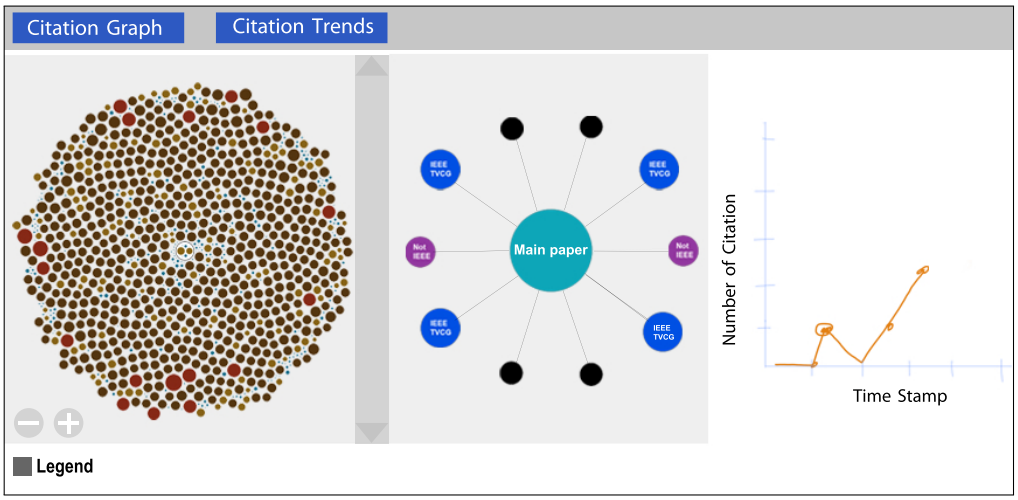

Figure 5.2: The Figure shows a visual prototype of CitationVis application as initial visualization concept. It was planned to visualize every article in the bubble chart as in the first image. The size and color would indicate the number of the citation an article received. Once a user selects any bubble, it would map the citation network as the second image. Or it would visualize the citation trends in line chart as the third image. Also, a color legend would describe the amount of citation an article received.

In the next section, we are continuing to discuss the CitationVis application used for the visualization. We will describe the GUI of CitationVis application in details so it would be easier to interact or use.

5.3 Citation Visualization with CitationVis

CitationVis is an interactive web-based application. We have developed this CitationVis to visualize citation trends of articles in TVCG journal of IEEE organization. CitationVis aim to scrape data from TVCG journal and store into the database automatically. Later, a user can interact with the interface and explore the citation information into visualization retrieving from the database. We have attempted to make the application as interactive as possible. The application interface is intuitive and easy to use.

From the top left of the interface in the Figure 5.3, we can see the application name CitationVis, Home, Trend Chart and Citation Trends. These are the different tab in the main menu bar. The user can switch one item to another item in the menu bar. The user can switch between views by just clicking histogram or citation trends or on click to Trend Chart menu switch to line chart visualization. The main menu bar, there is a search name 'Enter article title' including a search box. This search bar is interactive. It has

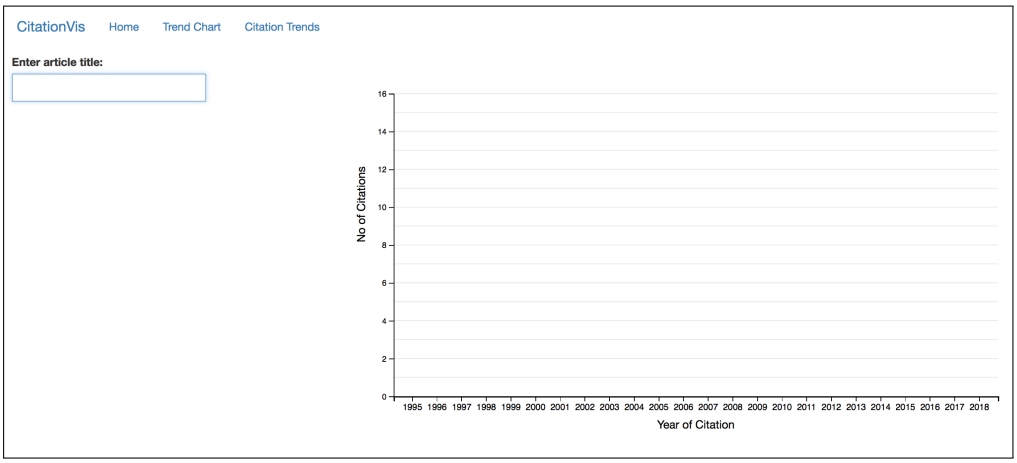

Figure 5.3: The Figure shows the interface of CitationVis Web Application. This interface contains mainly the menu bar to switch different visualization, Search bar to search the article, and the visualization SVG position to draw the citation graph against the article.

filtering option as well. The main visualization has years on x-axis from 1995 to 2018 and citation count on the y-axis. The value of these two scales are constant, but x-axis is independent and y-axis dependent on the x-axis. The user interaction visualization displays within the visualization area. A user can search article in the search box, and the visualization graph will be drawn within the visualization area.

Visualization relies on the user interaction. So, the user has to decide which visualization procedure wants to try either the Trend Chart or Citation Trends. Trend Chart will display visualization in a histogram. A user can search typing an article title in the search box, select an article title from a drop-down list and get it's line chart or histogram chart of the citation trends over the years. Compare its citations between different years. Now the user can get knowledge how the citation has changed over the year. It gives the idea that how the article topic is active and being referred to other papers. It also becomes an important factor when a user wants to compare citation changes over time.

Similarly, Citation Trends visualization helps the user to search an article by title and visualize citation frequency over time and compare different articles by their number of citations. The x-axis and y-axis show the same value in this line chart as well. This overview is about the CitationVis interface.

5.3.1 Citation Trends with Histogram

The Citation Trends visualization displays an article received all citations over the years using a simple histogram chart. A user can search all the articles published in TVCG

journal between 1995-2018. A user can search by title of the article in the search box and get it all citation details showing in histogram graph. But, it is hard to remember a full title of the article. So, whenever a user types a letter on the search box, an auto search suggested the title of papers and from the drop-down menu bar. The title can be selected and view the citation overview in time-series in the visualization area. The Figure 5.4 shows autocompleted search helps to see all the title of articles once we type any alphabet.

Once the article is selected, then a histogram graph is drawn within in visualization area. Histogram graph shows the overview of citation trends of an article over the years. The height of the block indicates the number of citations. A user can hover the mouse see details of a particular year. Each vertical block of the chart shows the total number of citation received for the particular year. The details of the article for a specific year can be seen by clicking on any bar. As user clicks, a table is displayed with more information such as title, year, URL, publication date. We are visualizing all citation within the range of 1995-2018. So, there are 24 years of citation records if a paper published in 1995.

The graph displays two discrete categorical variables, a dependent variable as citation received per year and independent variable as the year. The histogram graph is simple and easy to interpret about the citation data. The x-axis shows the time-dependent variable that is the year of citation and the y-axis show the total number of citation received in that particular year. The height of each block indicates the total number of citations received for a specific year. The width of each block is equal in size. The y-axis shows total citation count.

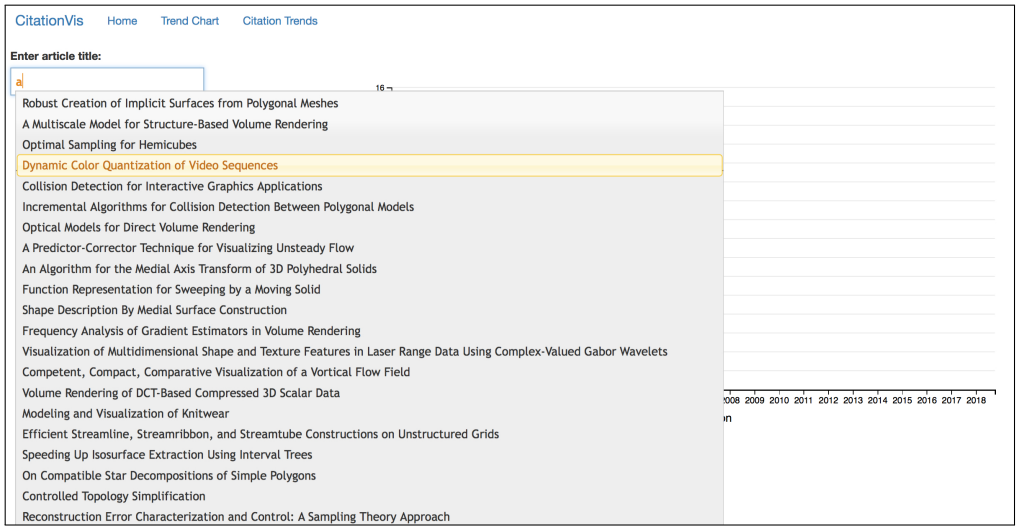

Figure 5.4: The Figure shows an autocompleted list of all article title. When a user type any alphabet, and then this autocomplete search works and displays all the list of the article title.

The Figure 5.5 shows an article is searched in the search box and the article title is "Optical Models for Direct Volume Rendering." A histogram graph is drawn against of the search article. The bar in the chart shows the number of citation received in the y-axis and the citation happened in the particular year in the x-axis.

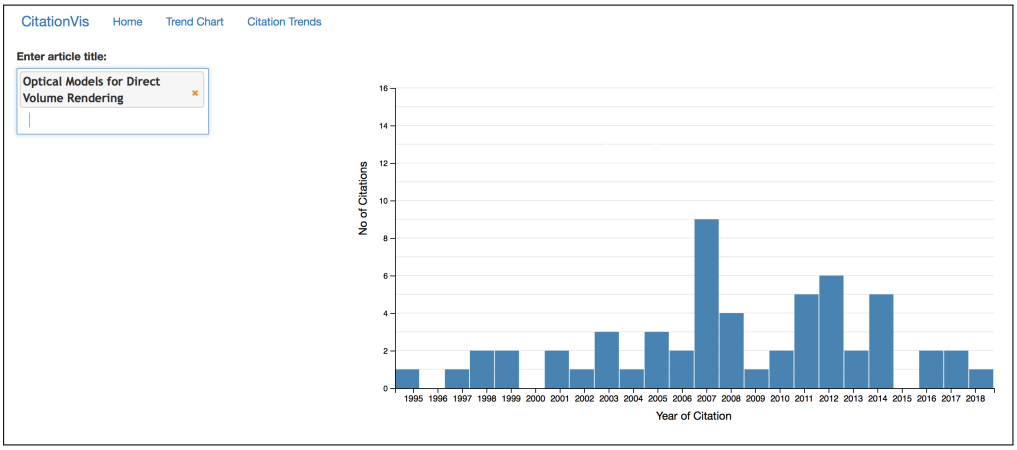

Figure 5.5: The Figure shows an article is searched and a histogram chart is drawn. The histogram chart shows the overview of all citation received in different years. The paper was published in 1995. The graph shows the trend of citation received within 1995-2018.

Now a user can hover the mouse over each bar in the chart and then the color of that bar will change to strong red color from the strong cyan-blue color. Also, there a quick view of the total citation number and the year of citation can be seen in the toolkit as well. When the page is refreshed, then the most updated data will be visualized in the bar chart. Every time the data will update for hovering the bar. The Figure 5.6 shows mouse hovered over a bar and it shows the total number of citation received in the specific year.

So, how it is happening. At first, When a user clicks on the Citation Trends button in the menu bar, then it sends an AJAX request to the database (publication) through the PHP back-end. The data is returned by inner join and group by query statement between article table and article citation table using a MySQL query. Then the queried data parsed by PHP built-in parser function into JSON format. After that, the JSON data return to JavaScript D3 function. Now, each citation data item such as the number of citation is represented as value and year. This parsing is done by JavaScript D3 each function. Finally, this citation data is sent to visualization function which creates the histogram graph visualization using the D3 library.

The citation trends histogram chart showing in the Figure 5.6 that the highest number of citation happened in 2007 with nine citation count after publishing the paper in 1995.

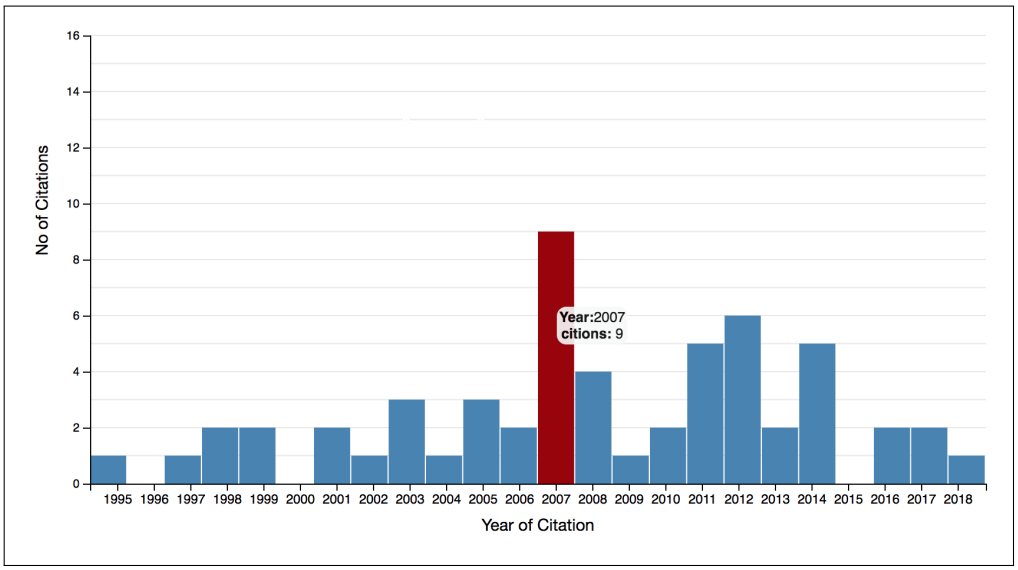

Figure 5.6: The Figure shows a mouse hover over an event in 2007. Then the color of the bar is changed to strong red color from the strong cyan-blue. It also displays the total citation count and the year of citation in a pop-up text.

While there are some years, no citation is done. Overall, the graph shows the topic was mostly cited between 2001 to 2014 with a high number of citations in 2007. After that, the citation falls down. There are few exceptions found such as no citation in 1996, 2000, and 2015 whereas the most citation occurred in 2007.

This citation trends visualization includes another feature that helps to the user to know more details about the citations. For instance, what are the articles cited in a specific year? When did the citation happen? What are the titles of the articles? Where is citation page references link? A user can get all the information by clicking on any bar of the graph. In the Figure 5.7 shows the table of contents after clicking on the citation bar of 2007. Also, the bar color has changed to strong red color from its original color.

So, every time a user clicks on another bar in the graph, a new table is created for that particular year. This new table contains the article title, publication date, and URL, extracting the data from tables (article, publication date, URL) of the publication database. A scroll up/down option is there on the table. In this table, a user can scroll up and down to see all of the cited articles and check their details. We only considered the title, publication data, and URL link displaying in the table. The URL connected to the original article page. A user can also visit the actual reference article online clicking the URL. The article will open via internet browser in a new window. The Figure 5.8 shows the table name with all the cited paper details namely article title, publication date, and clickable URL links for all the papers.

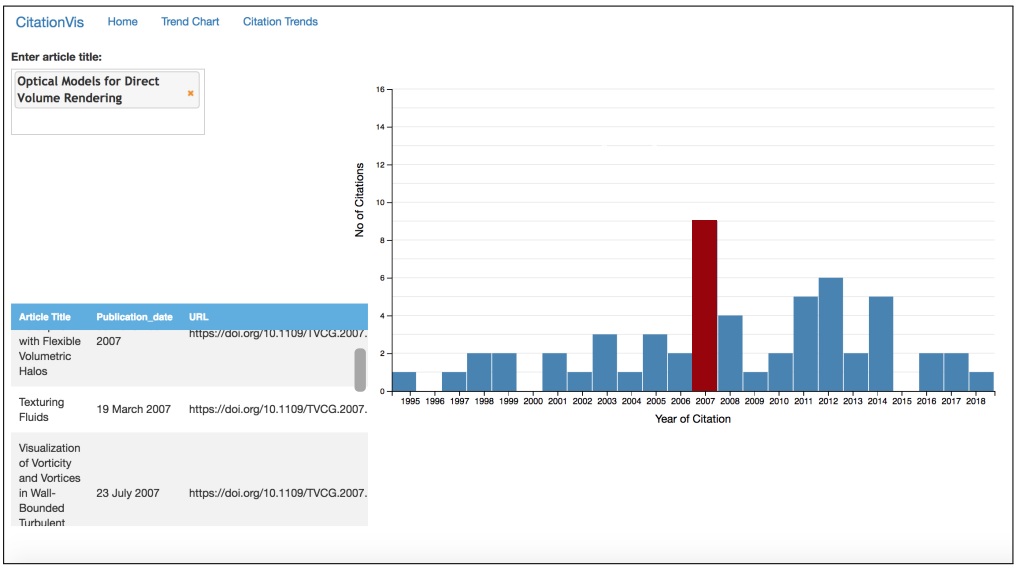

Figure 5.7: Shows a mouse hover in 2007. Then the color of bar is changed to strong red and also showing the total citation received in this particular year in pop up text.

5.3.2 Citation Trends with Line Graph

This Line graph also consists of a horizontal x-axis demonstrate all the years and the vertical y-axis illustrate the number of citations received. The citation count is either zero or positive values. So, The axes intersect near the bottom of the y-axis and the left end of the x-axis. The point at which the axes converge is always (0, 0). Each axis is labeled with a data type. For example, the x-axis as years and the y-axis is the number of citations. If there is a citation, then it indicates with a dot. Wherever a citation is present, there is a dot. And the citation dot points are connected by a line.

The x-axis is used to indicate all the years separately. We are visualizing citation trends per year. Our citation data are from the year 1995 to 2018, so there are 24 years. All the years are independent because the year does not depend on other value. But the number of citation received in the y-axis dependent on x-axis time interval. We are going to represent the citation data of articles so it is important to visualize in line chart as we can identify the trends of the citation. We want to visualize and compare citation data over time in the line chart. So, a user can get the idea how a particular article's citation value changes over time. Therefore, we can compare multiple article's citation trends.

In the last section, we have discussed histogram graph visualization. Now we want to switch to line chart visualization, and for that, we have to click to Trend Chart in

Article Title	Publication_date	URL
Perception with Flexible Volumetric Halos	05 November 2007	https://doi.org/10.1109/TVCG.2007.
Texturing Fluids	19 March 2007	https://doi.org/10.1109/TVCG.2007.
Visualization of Vorticity and Vortices		

Figure 5.8: The Figure shows a table of all the citations received in 2007. It includes article title, publication data, and URL. The URL link is clickable to browse the article webpage. The contents of the table can be scrolled to see all the citation details as well.

the main menu bar. On click, the previous visualization will disappear, and line chart visualization will appear in the display. Now the interface looks similar to the histogram chart visualization. It contains all the same option such as search box. This search box accepts autocomplete and multiple title section. So, when a user types a letter in the search box, it started matching with all of the titles in the database by each letter and loaded the title into the autocomplete search bar. A user can select an article and view the citation trends of the article. The Figure 5.9 shows a title of the article chosen "Optical Models for Direct Volume Rendering" and its line chart is drawn in the SVG visualization area.

In this visualization, a slider is included to the application for filtering the citation values by year. This slider will give the flexibility to interact with the line chart as it is a timeline slider. This slider range between 1995 to 2018 timeline interval. So, when a user moves the slider bar, the line chart also changes accordingly. Citation information will show in the line chart depending on where the slider is stopped. This is done by filtering in client side. When the slider stops moving, then another AJAX call is sent to the database, and then it collects the number of citation received for the selected years

in the slider. So, the citation data in line chart will changes when user move the slider within the slider range. The maximum is 2018, and the minimum is 1995. The slider is useful when there are many years of citation.

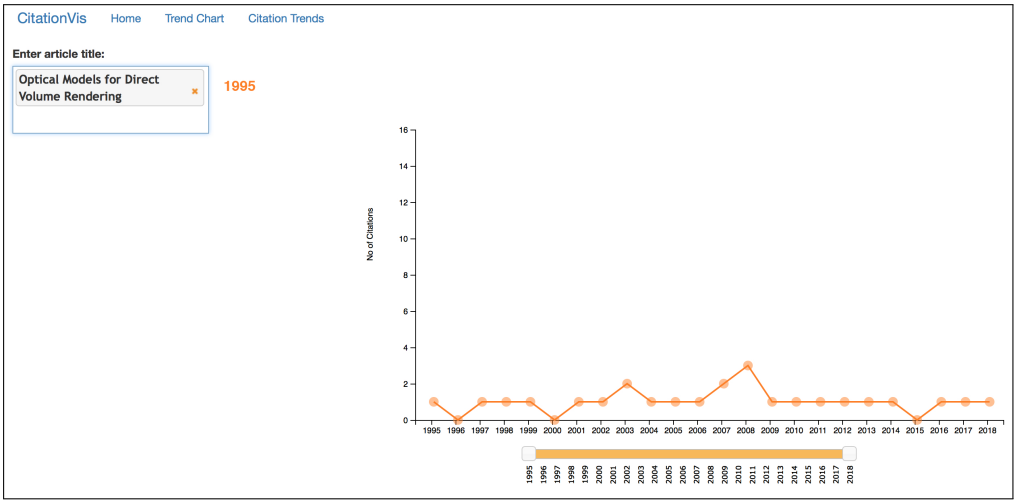

Figure 5.9: The Figure shows citation line chart is drawn for the selected title published in 1995. A slider is also available under the visualization.

The searched title has a close icon, and the search title can be removed from the search box by clicking the close icon. Next, to that, the publishing year is present there. In the searched title, we see it is in orange color. This color code is used to identify citation trends in the line chart for that particular article. If a user hovers the mouse on the citation line, the color becomes dark and highlighted. The citation point also increased and if hover mouse on the spot, we see its details in the toolkit. For example total citation count, and the year of publication. We used the line chart idea to know the citation trend in line so that we can see when the citation line peak and when the citation is broken down. Now if a user wants to close the previous search title, on click to the close icon can remove it. Therefore the citation line chart of that title will be removed from the visualization area as well.

We mentioned earlier that this line chart is also used for comparing multiple articles citation. The Figure 5.10 shows the citation comparison of two different articles published in 1995 and 1996. Here the orange color is used for the first searched article and the green color is used for the second article. Mouse hover on the line highlights the first article citation. This citation shows that almost every year, the article received the citation from other paper. The line also shows a break down in 2015 with zero citation. In most of the documents, we did not find any citations in 2015. On the other hand, another paper published in 1996 received a couple of citations within next year of publication. Later this paper was not being cited anymore. However, a user can see

the differences of citations received for two different article those published in 1995 and 1996. A user can understand citation differences between two papers and also can conclude a decision that which paper cited most over the time.

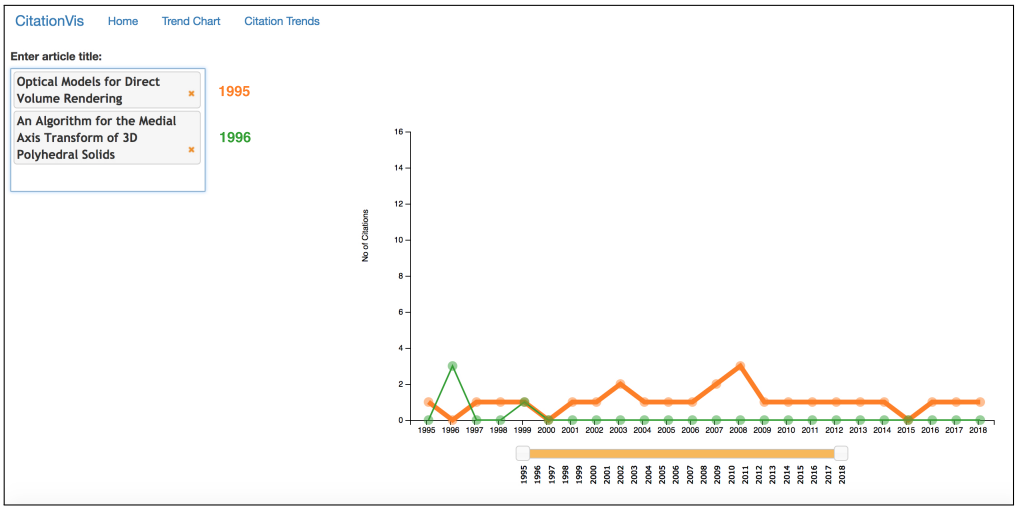

Figure 5.10: The Figure shows two articles are searched in the search box, published in 1995 and 1996. Then their citation line chart is drowned. The random color code is used to identify both citation value in the line chart. Also the color code visible and comparable between different articles.

A user can close any article by clicking the close icon. Then that article will be removed from the list and also the line chart will be removed from the visualization. If the user wants to add another article then can do it by simply searching and selecting the article. This way, a user can include multiple articles to see the overview of citation trends and can compare citation trends changes over time among various articles.

The Figure 5.11 shows multiple articles are selected to see their citation trends and comparison over time changes. The red highlight line chart shows most citation received in 2017 compared to other citations. This article is about D3. Therefore, we can conclude that in the advanced visualization, the D3 is the most cited and used in the data visualization field.

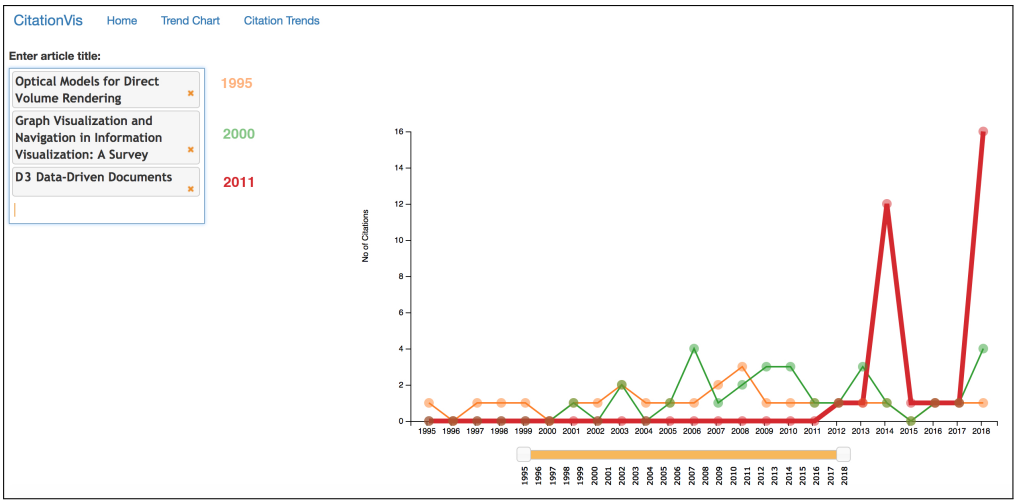

Figure 5.11: The Figure shows multiple articles searched in the search box, published in 1995, 2000, and 2011. The citation lines are drawn respectively. The color code is used to identify different articles. The D3 articles published in 2011 and mostly cited after 2013 among other papers. The highlighted line chart in red color indicates citations received for the D3 articles.

6 Conclusion

In this paper, we presented an automatic dynamic web scraper that can scrape citation data of articles published in TVCG journal of IEEE website. Later, we visualize the scraped citation data for citation trend analysis using CitationVis web-based interactive application. Since this work is an extension of previous TrendVis application [27], we got the database schema and we created a database followed the last design of the database. We used their static web scraper to extract article title, abstract, year, URL, DOI, keywords and store the data into the database. We modified this static scraper to included the publication date of each articles as we want to visualize exact data of citation happened.

This work aims to visualize citation trends over time, scraping citation data from TVCG articles and stored in the database. Citation data of articles are only available in dynamic load pages where static web scraping does not work. So, we have built an automatic web scraper which can traverse throughout the dynamic DOM and scrape the citation data such as DOI, from TVCG journal. This dynamic web scraper selects an article URL, traverse DOM of all the citation URL for the selected article and then scraped DOI from cited article homepage by Nightmare.js evaluate function. Finally, insert the DOI of cited articles into the database against each article in newly table.

For the visualization, we developed the CitationVis which is a web-based interactive application helps to visualize the citation trends of the article using histogram chart and line chart visualization methods. Line chart used to analyze the citation trends, compare citation ratio between multiple articles that change over time. Histogram chart used to visualize citation trends shows in the block over the years. Each block of the year in the chart used to explore in details of the cited papers. For example, a user can see the exact citation date. Also, the user can access the cited full article on click to the URL.

IEEE Explore has many metadata of articles available such as abstract, authors, references, citations, and keywords. TrendVis [27] worked with keywords and abstract of articles from TVCG IEEE Explore. We have extended to the project and worked with citations of TVCG articles. In future, the other metadata of article can be scraped and visualize them according to their expectation. The existing database is flexible to extend. We focused to collect citation data only from IEEE publications. The citation data from other publications are not considered with this work. As an extension of this work, citations from other publications can be scraped. Our scraper is extendable so

that anyone can reuse for scraping other website. And visualize their citation data and compare with IEEE publications vs. other publications. This work only included articles in TVCG journal. However, it can be used to include other journals as well.

List of Figures

List of Tables

Bibliography

[1] "Introduction to web scraping," 04 2018. [Online]. Available: https://data-lessons.github.io/library-webscraping/

[2] B. Adelberg, "Nodose?a tool for semi-automatically extracting structured and semistructured data from text documents," in *ACM Sigmod Record*, vol. 27, no. 2. ACM, 1998, pp. 283–294.

[3] A. Arasu and H. Garcia-Molina, "Extracting structured data from web pages," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 2003, pp. 337–348.

[4] G. Boeing and P. Waddell, "New insights into rental housing markets across the united states: web scraping and analyzing craigslist rental listings," *Journal of Planning Education and Research*, vol. 37, no. 4, pp. 457–476, 2017.

[5] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[6] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[7] J. Clark, "Neoformix blog," 2009.

[8] V. Crescenzi, G. Mecca, P. Merialdo *et al.*, "Roadrunner: Towards automatic data extraction from large web sites," in *VLDB*, vol. 1, 2001, pp. 109–118.

[9] T. Dönz, "Extracting structured data from web pages," 2005.

[10] N. Elmqvist and P. Tsigas, "Citewiz: a tool for the visualization of scientific citation networks," *Information Visualization*, vol. 6, no. 3, pp. 215–232, 2007.

[11] D. W. Embley, D. M. Campbell, Y. S. Jiang, S. W. Liddle, D. W. Lonsdale, Y.-K. Ng, and R. D. Smith, "Conceptual-model-based data extraction from multiple-record web pages," *Data & Knowledge Engineering*, vol. 31, no. 3, pp. 227–251, 1999.

[12] E. Garfield, “Citation indexing its theory and application in science, technology and humanities (information science s.),” *John Wiley & Sons Inc.*, 1973.

[13] D. George and R. Knegjens, “Paperscape,” *URL: http://blog. paperscape. org*.

[14] M. Ghoniem, J.-D. Fekete, and P. Castagliola, “On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis,” *Information Visualization*, vol. 4, no. 2, pp. 114–135, 2005.

[15] C. Görg, Z. Liu, J. Kihm, J. Choo, H. Park, and J. Stasko, “Combining computational analyses and interactive visualization for document exploration and sensemaking in jigsaw,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 10, pp. 1646–1663, 2013.

[16] F. Horn, “Interactive exploration and discovery of scientific publications with pubvis,” *arXiv preprint arXiv:1706.08094*, 2017.

[17] N. P. Hummon and P. Dereian, “Connectivity in a citation network: The development of dna theory,” *Social networks*, vol. 11, no. 1, pp. 39–63, 1989.

[18] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. D. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko, “vispubdata. org: A metadata collection about ieee visualization (vis) publications,” *IEEE transactions on visualization and computer graphics*, vol. 23, no. 9, pp. 2199–2206, 2017.

[19] J. Kamps and M. Marx, “Visualizing wordnet structure,” in *Proc. of the 1st International Conference on Global WordNet*, 2002, pp. 182–186.

[20] D. A. Keim, “Visual exploration of large data sets,” *Communications of the ACM*, vol. 44, no. 8, pp. 38–44, 2001.

[21] ——, “Information visualization and visual data mining,” *IEEE transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.

[22] K. Kucher and A. Kerren, “Text visualization techniques: Taxonomy, visual survey, and community insights,” in *Visualization Symposium (PacificVis), 2015 IEEE Pacific*. IEEE, 2015, pp. 117–121.

[23] A. H. Laender, B. A. Ribeiro-Neto, A. S. Da Silva, and J. S. Teixeira, “A brief survey of web data extraction tools,” *ACM Sigmod Record*, vol. 31, no. 2, pp. 84–93, 2002.

[24] S. Liu, W. Cui, Y. Wu, and M. Liu, “A survey on information visualization: recent advances and challenges,” *The Visual Computer*, vol. 30, no. 12, pp. 1373–1393, 2014.

[25] S. Mahajan and N. Kumar, “A web scraping approach in node. js,” *International Journal of Science, Engineering and Technology Research (IJSETR)*, pp. 909–912.

[26] J. Matejka, T. Grossman, and G. Fitzmaurice, “Citeology: visualizing paper genealogy,” in *CHI’12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2012, pp. 181–190.

[27] I. Mehmood, “Interactive visual trend analysis of tvcg articles,” Master’s thesis, University of Ulm, 2017.

[28] R. Mitchell, *Web scraping with Python: collecting data from the modern web*. " O’Reilly Media, Inc.", 2015.

[29] B. Shneiderman, “The eyes have it: A task by data type taxonomy for information visualizations,” in *The Craft of Information Visualization*. Elsevier, 2003, pp. 364–371.

[30] H. Small, “Co-citation in the scientific literature: A new measure of the relationship between two documents,” *Journal of the Association for Information Science and Technology*, vol. 24, no. 4, pp. 265–269, 1973.

[31] J. Stasko, J. Choo, Y. Han, M. Hu, H. Pileggi, R. Sadana, and C. D. Stolper, “Citevis: Exploring conference paper citation data visually,” *Posters of IEEE InfoVis*, vol. 2, 2013.

[32] T. F. Sturm, *LATEX : Einführung in das Textsatzsystem*, 9th ed., ser. RRZN-Handbuch. Hannover [u.a.]: Regionales Rechenzentrum für Niedersachsen, RRZN, 2012.

[33] M. Theus *et al.*, “Interactive data visualization using mondrian,” *Journal of Statistical Software*, vol. 7, no. 11, pp. 1–9, 2002.

[34] M. Weber, M. Alexa, and W. Müller, “Visualizing time-series on spirals.” in *Infovis*, vol. 1, 2001, pp. 7–14.

[35] P. Xu, C. Stolper, A. Sainath, and J. Stasko, “Vis 25?all the papers and citations,” *Website: http://www. cc. gatech. edu/gvu/ii/citevis/VIS25*, 2014.

Declaration

I hereby declare that this thesis titled:

Interactive Citation Trend Analysis of TVCG Articles

is the product of my own independent work and that I have used no sources or materials other than those specified. The passages taken from other works, either verbatim or paraphrased in the spirit of the original quote, are identified in each individual case by indicating the source. I further declare that all my academic work was written in line with the principles of proper academic research according to the official "Satzung der Universität Ulm zur Sicherung guter wissenschaftlicher Praxis" (University Statute for the Safeguarding of Proper Academic Practice).

Ulm, ...

Nitai Chandro Roy