Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Master's Thesis Nr. 298

Systems Group, Department of Computer Science, ETH Zurich

The state of network research

by

Haoyu Zhu

Supervised by

Prof. Ankit Singla, PhD student Maximilian Grüner

February 2020–August 2020

inf | Informatik
Computer Science

The state of network research

HAOYU ZHU

Master's Programme, Communication Systems, 120 credits
Date: August 14, 2020
Supervisor: Ankit Singla
Examiner: Peter Sjödin
School of Electrical Engineering and Computer Science
Host organization: ETH Zurich
Swedish title: Tillståndet för nätverksforskning

The state of network research / Tillståndet för nätverksforskning

© 2020 Haoyu Zhu

Abstract

In the past decades, Networking researches experienced great changes. Being familiar with the development of the networking researches is the first step for most scholar to start their work. The targeted areas, useful documents and active institutions are helpful to set up the new research. This project, focused on developing an assistant tool based on public accessed papers and information over internet. Which allows researchers to view most cited papers in networking conferences and journals. NLP tools are implemented over crawled full-text in order to classify the paper and extract the keywords. Papers are located based on authors to show the most active countries around the world that working in this area. References are analyzed to view the most cited topic, and detailed paper information. We draws some interesting conclusion from our system, showing that some topic does attracts more attaction in the past decades.

Keywords

Natural Language Processing, Network Spider, Network, Topic Modelling, State of Research.

Sammanfattning

Under de senaste decennierna upplevde nätverksundersökningar stora förändringar. Att känna till utvecklingen av nätverksundersökningar är det första steget för de flesta forskare att starta sitt arbete. De riktade områdena, användbara dokument och aktiva institutioner är användbara för att skapa den nya forskningen. Projektet fokuserade på att utveckla ett assistentverktyg baserat på offentliga åtkomstpapper och information via internet. Som gör det möjligt för forskare att se de mest citerade artiklarna i nätverkskonferenser och tidskrifter. NLP-verktyg implementeras över genomsökt fulltext för att klassificera papperet och extrahera nyckelorden. Artiklar är baserade på författare för att visa de mest aktiva länderna runt om i världen som arbetar inom detta område. Hänvisningar analyseras för att se det mest citerade ämnet och detaljerad pappersinformation. Vi drar några intressanta slutsatser från vårt system och visar att något ämne inte lockar till sig mer under de senaste decennierna.

Nyckelord

Naturlig Språkbearbetning, Nätverksspindel, Nätverk, Ämnesmodellering, Forskningstillstånd.

Acknowledgments

In my two-years master studies, I have many friends, professors, staff, and strangers to thank. They helped me a lot when I was facing problems. Without them, it is not possible for me to type these words here.

I would like to thank my thesis supervisor Professor Ankit Singla and Maximilian Grüner. You provide help and assistance in my entire work. Your kindness and selflessness helped me a lot, especially when facing the new COVID-19 virus.

I would like to thank my friends, Chen Tang, Haoran Yao, Ming Cui. At the struggling beginning of our life in Sweden, We four helped each other in settling. You give me the faith to live and study here.

I would like to thank my friends I met in Switzerland. You just like lights, brightening my way when I was taking exchange studies alone in Switzerland.

I would like to thank my house lord, Mr. Fritz and Mrs. Esther. During the thesis time you really make me feel at home. It is one of the happiest and relaxed time in my life.

I would like to thank my parent. You support me during the epidemic, pacify my felling and encourages me to continue working and studying.

I would like to thank my girlfriend. The time when I was struggling for the projects and thesis, you stay with me.

There are still many other people that I would like to thank. I can't list you all in this short acknowledgment. But I would like to say Thank You to all of you!

Stockholm, August 2020
Haoyu Zhu

Contents

List of Figures

List of acronyms and abbreviations

BS BeautifulSoup

CERMINE Content ExtRactor and MINEr

ESA Explicit Semantic Analysis

ETD Emerging Trend Detection

HTML HyperText Markup Language

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

MDUI Material Design User Interface

NMF Non-negative Matrix Factorization

SVM Support Vector Machine

Chapter 1

Introduction

It has been a long time when people started trying to connect scientific researches together, meanwhile the concept of *Research Front* has been introduced by Price[1]. Citations help a lot in building such networks in identifying research fronts when machine learning and NLP are not even introduced in 1965. But world changes, so as the technologies. In 2004, Kontostathis et.al tried to identify the emerging research trend using textual data mining[2] using new methods and set up Emerging Trend Detection (ETD) systems. An ETD system is designed as automatic or semi-automatic detecting new research trends over the internet. As a general and commercial software, it doesn't provide much valuable information about network researches. Our idea is to build up a new, web-crawler and NLP based web system, which assists scholars to start their research about networks and communications. This assistant mainly focuses on the topic of papers, their reference relationship, active institution and main individuals.

1.1 Background

The Internet is public for everyone. Knowledge is always shared on internet. To start research, researchers usually need to go through tons of documents to find the topic. It is not wise to build up a specific tool for each research. That is the main idea of this paper: To build up a universal tool for these researchers on networking area. The system is based on open accessed papers on internet, using topic modeling and keyword extraction to extract topics and keyword information from these papers. And shows them on an interactive website so that researchers can get their customized information easily and quickly. As the research is pushing forward day and night, The system is designed as a

scale-free platform with the ability to crawl, resolve and add new papers to it.

1.2 Problem

The system can be approximately divided into several parts:

1. Data collection and formatting
2. Data analysis
3. Outcome display

Relatively, the system is facing problems on every part of the system:

1. How to obtain research paper metadata.
2. Download research paper full-text in a legal way.
3. Find a suitable algorithm to model the topic, extract the keyword and build reference relations.
4. Build up a platform to display the result we got.

Apart from these main problems, There are also many tiny, but cannot be ignored problems: PDF resolving, Data structure, Information extraction, Server side programming and data visualization. These problems will be discussed in detail in the next chapters.

1.3 Contribution

The main contribution can be summarized as follow:

- We obtained the metadata from semantic scholar's open corpus[3].
- We modeled research topics based on corpus from top conferences and journals in network area. These topics are displayed in an interactive graph with special filters.
- We extracted the keyword from the abstract and full-text of research articles, and build a keyword cloud based on keywords.
- We developed reference relationships between topics and shows using a chord dependency diagram.
- We summarized active authors and institutions around the world in this area.

1.4 Research Methodology

This project will be focusing on several parts: data collection, formatting, processing and visualization. The paper metadata will be collected if they are top conferences and top journals in network area. The source of metadata is mainly from two metadata providers: crossref[4] and semantic scholar open corpus[3]. As the metadata is the basis of this project, We then continue crawling full-text on internet using web spiders. The full-text is then used as the input for topic modeling to identify the paper topics. In this part, tf-idf and nmf are used for topic modeling algorithms. Text rank is introduced into keyword extraction which provides an acceptable accuracy on keyword extraction from abstract or full-text. MongoDB is used to store the database and provide back-end data services on web data visualization. A website is designed using several frameworks and libraries to visualize the result and provides and interactive view.

1.5 Delimitation

This system is not perfect, There are a lot of things to do ahead. As a web spider based system, it can access those public published websites and files. I.e. author's personal webpage, Institution's publication page and conference's program page. The quality of full-text completely based on the author himself: whether we can get a full text mostly depends on if he or she is willing to publish full-text and make it available online.

The download and extraction of papers are time-consuming so the system is designed using abstract for topic-modeling and keyword extraction first. The resolution of PDF full-text is another shortcoming. Some character is not resolved correctly, figures and tables are somehow embedded into the paragraph, fonts and styles are different in different articles, formula and numbers are hard to convert into linear expressions.

There are few organizations provide research article metadata online. Among these providers, there is a little information we can get: title, abstract, author, references, publish date, full-text urls. But some information like affiliation is not provided in the metadata so that we can only get from full-text, which leads to low accuracy.

1.6 Structure of the thesis

This paper includes 7 chapters.

Chapter 1, Introduction, mainly focuses on the overall information of this project, introduces the main framework of this project, and explains the methodology in this project.

Chapter 2, Background and Related works, introduces the previous works that have been done related to our project, along with the inspiration of the project.

Chapter 3, System design and Implementation, describes the details about system architecture and components. Besides, the implementation detail and main algorithms are being discussed.

Chapter 4, Contribution, accounts the main contribution we have done to this project.

Chapter 5, Results and Analysis, talks about analysis outcomes from the researches, including hot topics, hot areas, active institutes and important individuals.

Chapter 6, Discussion, aims at discussing the outcomes we talked in Ch.5, explains the reason for results and their meaning.

Chapter 7, Conclusion and Future work, summarizes the project, points out the main academic achievement in this paper. The shortcoming of this paper and its unfinished work are also talked as future work.

Chapter 2

Background and related works

2.1 Metadata and full-text crawling

2.1.1 Web spiders

In 1994, the concept of web crawling was firstly introduced on the 2nd World Wide Web conference[5]. The definition of web-spider is an auto robot that goes through websites that we would like to visit and capturing information from them. The robot has the ability to find next entry to visit from existing pages and builds a visiting network like a spider, thus it was called web spiders. Web spiders are widely used in today's search engines like Google and Bing[6]. To download full-text from the internet, we need a web spider to find information and link of full-text.

2.1.2 HTML resolution

HyperText Markup Language (HTML) is a language for displaying contents for web pages[7]. HTML page is a tree-like document including top nodes like <body>, <head> etc. BeautifulSoup (BS) is a widely used library for resolving HTML and XML documents[8]. BS resolves html documents into a document tree, thus we can use BS selectors to locate nodes and extract information.

2.1.3 Similarity check

The distance between two points can be easily represented as the length of the line linking these two points. Similarly, the distance between the two vectors can be represented by cosine value of the angle between two vectors, which is also known as cosine similarity[9]. Considering mapping non-stop words

of two sentence (A, B) into an n-dimension space, where n is the number of different non-stop words in the sentence (A, B), we can have two vectors representing sentence (A, B). I.e. sentence A is "A cat is not a dog", sentence B is "A dog likes the cat, but the cat does not". Let the space be (cat, dog, likes). Vector A thus becomes (1, 1, 0), similarly, vector B becomes (1, 1, 1). Which means the angle θ between vector A and vector B becomes:

$$cos(\theta) = \frac{A \cdot B}{|A| \cdot |B|} \tag{2.1}$$

We can easily found that if the angle between two vectors is in range $(0, \pi)$. In our definition, the vector can only have a positive value in each dimension, So the angle between these two vectors is in range $(0, \frac{\pi}{2})$. Thus when two vectors is more near, their angle will be smaller, thus the cosine value is larger. This algorithm has many variations on constructing the vectors and spaces: The space can be constructed with or without sequential number, The factor in vector can be word frequency or word existence. Different construction method leads to different similarity results between sentences.

2.1.4 PDF resolving

Content ExtRactor and MINEr (CERMINE)[10] is an open-source tool used for extracting contents from full-text pdf files using Support Vector Machine (SVM). The output from this tool is a xml-like file called *cermxml*. Which can be resolved by BS[8].

2.2 Full-text analysis

2.2.1 Topic modelling

Topic model was firstly introduced by Papadimitriou in 1998[11]. It can be seen as a probabilistic classification and clustering in text mining. It creates topics from given document collections and gives probabilistic results of the relation between each document and each topic. Thus we can use topic modeling to mine useful information from our document collections. Many algorithms can be used for topic modelling such as Non-negative Matrix Factorization (NMF), Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA) and Explicit Semantic Analysis (ESA). These algorithms tried to fit maximum likelihood between document and generated topics to create

more informative and relative topics.

2.2.2 Keyword extraction

TextRank[12] is an algorithm on the inspiration by PageRank[13]. The core idea in PageRank is reference. I.e. if some page is linked by many other pages, it should have a higher rank, in other words, important. Thus we can create a linking network and update the rank of each page until it converges. The idea in TextRank is similar, in PageRank the reference connects pages, in TextRank follow works. If some word follows many other words, it will be considered as high rank. E.g. in a piece of sentence: *Routing protocols is the basic of internet, routing devices are called routers.* **routing** have two different words following: **protocol** and **devices**. Thus it will have higher ranks and can be considered as an idea of this sentence. In TextRank, if another word follows this high-rank word, the rank of this word will increase accordingly. This following relationship also constructs a connecting network, which can be updated to find the highest-ranking words.

2.3 Data visualization

Data visualization is used to visualize, analysis and display numeric data. D3.js[14] is an open-source JavaScript library that visualizes data using SVG, JS and HTML. It is easy to use, embed and customize in our project.

2.4 Related work

Researchers are always interested in new trends in their research areas. In 1965, Price introduced the concept of *Research Front* and build up a network using citation data to identify the latest but most cited papers [1]. He tried to find new trends and concepts from these newly published but hot papers. Later in 1990s and 2000s, NLP and semantic analysis were introduced into this area[15, 16, 17]. Their researches mostly have a common idea is to mine, extract or generate new trend from textual materials or corpus.

TimeMine[18] created timelines from free news corpus. The main purpose is to find emerging topics in an area and gives an overview of the development of topics in a timeline view.

William and Ting-hao introduced their achievement of a universal trend detecting system in their paper[15]. They build up a neural network model

to identify and extract emerging trends from a collection of corpus. The performance is also discussed when adapting to different networks in order to find the best model.

Chapter 3

System design and implementation

3.1 System architecture

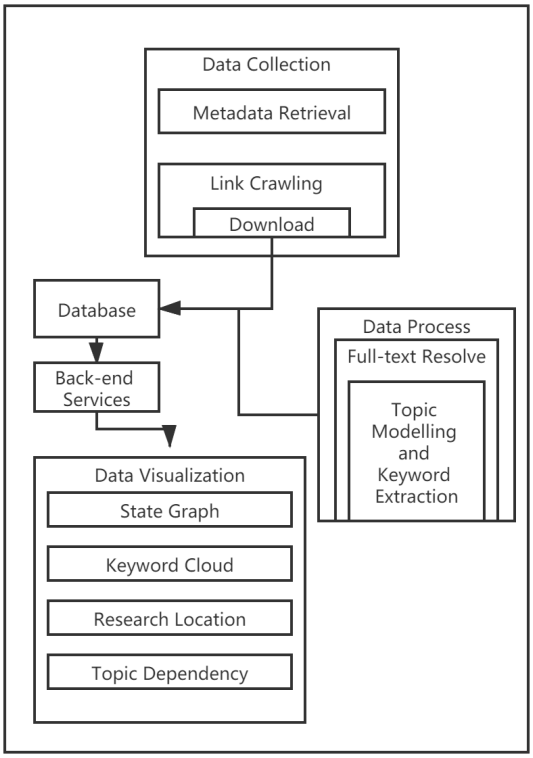

Figure 3.1: System Diagram

As fig 3.1 shows, We divide the whole system into three parts: Data

collection. Data visualization and Data process. Data collection has several functions: retrieval metadata, crawling full-text links and downloading. Data process takes the metadata, full-text or abstract as input, and gives the information about the papers: topic and keywords. Data visualization is the main part that we display to public. It contains several visualization pages. These parts are connected by a database and a back-end service. The database collects and stores the outputs from data collection and data processing part. The back-end service takes data from database and provides RESTful apis for websites in data visualization part.

3.2 Data collection

3.2.1 Metadata obtain

Metadata, defined as the data to describe other data[19]. Metadata of research articles normally contain several fields: title, doi or id, publish date, journal or conference information, authors, citations and references. Some may contain abstract, research area or even full-text link data[3]. As we can see, metadata of research article normally contains no information about keywords and classifications. Some publisher like IEEE, ACM and Elsevier shows metadata on their article sites, which is illegal form web crawlers to download. Our source of metadata are two research organizations: Crossref and Semantic Scholar by AI2.

Crossref

Crossref is a non-profit association set by thousands of academic publishers. It collects metadata from members(participated publishers) and distributes it to millions of researchers. To obtain metadata from Crossref, we need to make requests through a RESTful api. This api provides several filtering methods: to filter publish date, type, member(publisher) and some other fields. But mostly, The metadata from crossref contains no information about affiliations and abstract. However, It is enough as navigation.

Semantic Scholar

Semantic Scholar is another organization that provides metadata download. Semantic scholar is build by Allen Institute for AI. It provides a open-corpus that contains all metadata we can get from semantic scholar, 185 million items.

A good thing is that the metadata from semantic scholar contains abstract, citation, reference and pdf link. But this open corpus is for download, which means we need to set up a local database and query metadata from it.

3.2.2 Metadata formatting and storage

In this project we uses MongoDB as our database for metadata storage. MongoDB is a document-based Non-SQL database. Data is stored as documents in collections. To better maintain metadata storage and promote access speed from client side, The database is designed as follows:

Metadata collection contains metadata information, including those we retrieved from Crossref and Semantic Scholar, crawling and downloading status.

Topic collection contains the topic information we get from topic modelling.

Journal and **Event** collection contains top journal and conference that we added to our database.

Country_paper collection contains the location and affiliation information of these papers.

Keyword collection contains keyword summary of each year from 2000.

3.2.3 Full-text acquire

Now, In our system we have only metadata and abstract(from semantic scholar). Full-text is another important data that in our project. Luckily semantic scholar open corpus provides pdf urls for papers in its database. But some links are blank or not accessible. Another backup is to crawl full-text links from search engine like DuckDuckGo.

Link crawling

For most papers in our database, semantic scholar provides a list of full-text links in its metadata. But unluckily, some links are broken due to time. In other words, some paper may not have useful links from semantic scholar. For those papers, a back-up plan of crawling from search engine is enabled. DuckDuckGo is a search engine that we used in this scenario. The result from legacy page "https://html.duckduckgo.com/lite" is easy to be resolved. So here we used html-session for python[20] to simulate get requests and crawling the result from DuckDuckGo. In DuckDuckGO request we set query as (article title) + (filetype:pdf) to filter PDF full-text results, and stores them into MongoDB database.

Full-text download and check

When the link for full-text is ready, download process is activated. The links we get is somehow not correct, it can be presentation slides, CV contains title of that research paper, similar document or even uncorrelated documents. So it is necessary to do a check before and after download to make sure that the full-text we got is correct. Here we divide the check into two parts: title check before downloading and pdf check after downloading. The method we used at here is cosine similarity algorithm[9] we introduced in the last chapter. If two vectors are the same, the angle will be $\frac{\pi}{2}$ thus the cosine is 1. We use this algorithm to check if the title in metadata and in web resources are the same. If the cosine between metadata title and link title is larger than the threshold, the download will start. When the full-text is downloaded, the title of full-text is checked as a double-check. Meanwhile, the page size is checked to make sure it is an article rather than a presentation slide.

3.3 Data analysis and processing

When full-text or abstract is ready, we start the data processing part. For pre-processing, The full-text will be extracted and formatted into database. These full-text or abstract will then be used for NLP purposes.

3.3.1 Full-text extraction

CERMINE tried to extract any information in full-text file and classifies them. The output xml-like file uses labels to identify full-text, abstract, author, contact information and references. However, the paper styles vary. There may have some accuracy problems for CERMINE extraction. Here we take only full-text body and authors as output and stores to the database.

3.3.2 Topic modelling

NMF[21] is a linear algebra based algorithm used in many areas including signal processing, Pattern Recognition and Natural language processing[22]. The aim for NMF in topic modeling is to factorize the document matrix into two new matrices with the smallest error, one matrix mapping the document to topic matrix while te other mapping the words to topic matrix[23]. Scikit learn, a famous, open source and widely used machine learning library[24] which contains encapsulated NMF functions and libraries is used here to

deploy the NMF part. It contains several coefficients that may influence the result of NMF topic modeling. As a mathematical algorithm, NMF doesn't contain the ability to automatically determine the topic counts. To show a comprehensive result from our dataset, we let the topic counts in a range between five and twelve, and can be switched when visiting our websites. The other coefficients are adapted manually to get the best classification results.

To estimate the result of topic modeling, t-SNE is used to visualize high dimension results from NMF into a 2D graph[25]. As fig3.2 shows, the accuracy of topic modeling results can be approximately estimated from t-SNE graph by the degree of decentralization from the graph. In fig3.2, topic 0, 1, 3 are well distributed, but topic 2, 4, 5 seems to be more separate, especially topic 4. It is convenient to find problems if there is some bad topics. I.e. the result distributes in the whole graph, or even into other topics like topic 4.

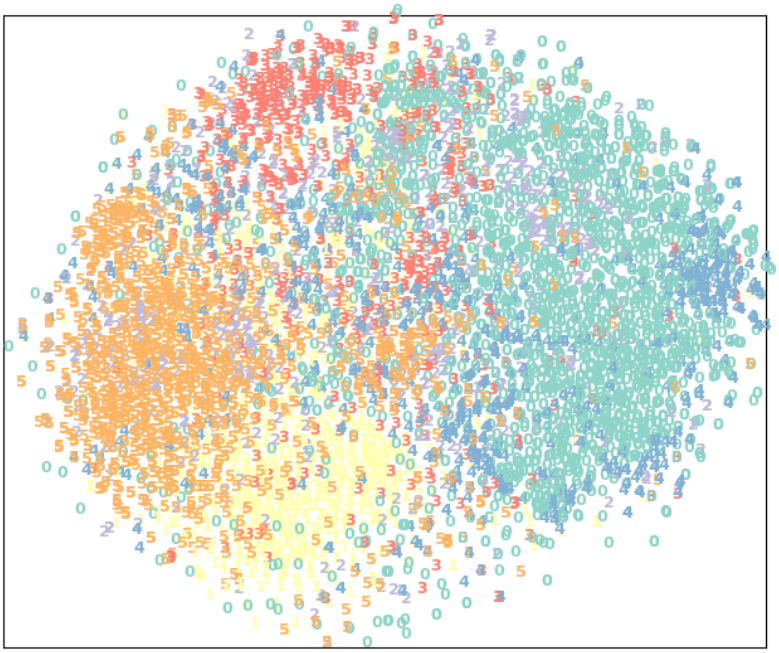

Figure 3.2: t-SNE data visualization

3.3.3 Keyword extraction

In Ch.2 we have introduced TextRank algorithm for keyword extraction. Obviously, the words that have the highest ranks should represent the main idea of a piece of article[26]. This leads to our main method used in keyword extraction.

However, words have many forms: None, Verb, Adjective, Adverb, Complex, Past tense, etc. It is not a good idea to count them separately, thus before TextRanking, a lemmatization is required to mapping different forms of words into a common form to prevent repeated keywords.

3.3.4 Affiliation extraction

Normally, when authors submitted their papers, they will attach their affiliation in their paper and submitted metadata. But almost all paper in the database from Crossref contains no information of affiliation of each author. Thus it is tricky to identify the affiliation of papers in our database. Considering that most authors use an organization email address as their contact information, we extract E-mail addresses from full-text and classify them into different affiliations so that we can easily identify paper's affiliation. However, different authors in one paper may come from different organizations, So one paper may affiliate to more than one organization.

3.3.5 Topic dependency

The metadata in Semantic Scholar contains information of citations and references of one paper. Thus we made updates to our database to include information about topic citations. Thus from our database we can easily draw connections between topics and find most cited topics and papers.

3.4 Data visualization

In previous sections we introduced the data collection and processing part that outputs results for data visualization. D3.js[14] and chart.js[27] are two main library we used here for data visualization. D3.js provides many highly customized data visualization components. We here uses word cloud, map, chord dependency diagram for data visualization.

3.4.1 Server side implementation

Like other modern websites, we designed our website framework as front and back end separated. A server-side application is running on bach machine of D-INFK ETH Zurich. It is hosted using Node.js to capture REST requests. All database operations are put on server side to enhance efficiency and security.

3.4.2 Website infrastructure design

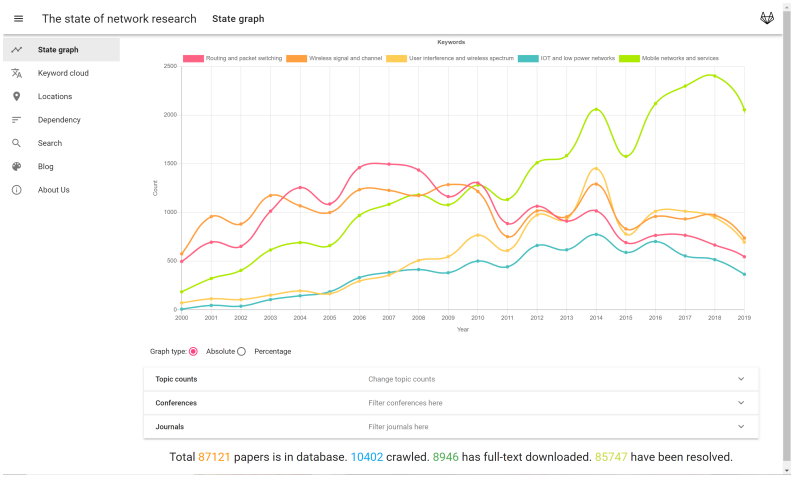

Figure 3.3: Website overview

Material Design User Interface (MDUI) is picked up by us for interface framework. Here we divided the website into 7 pages:

State Graph

State graph describes how the topic in network research develops. It has four main filters as fig3.4 : type, topic count, conference and journal. The type filter can be set to absolute data or percentage data, in order to lower the influence of paper count differentiation between years. The topic count filter can be used for switching topic counts. Conference and journal filters can be used to select whether these specific conferences and journals will be included in the graph data. When hovering over points, there will have a tooltip showing summary information about this topic in this year. When clicking on the points on graph, we can have a look on the detailed data about the topic in this year as fig3.5 shows. The paper is ranked by citation counts and have a full-text link for visitors.

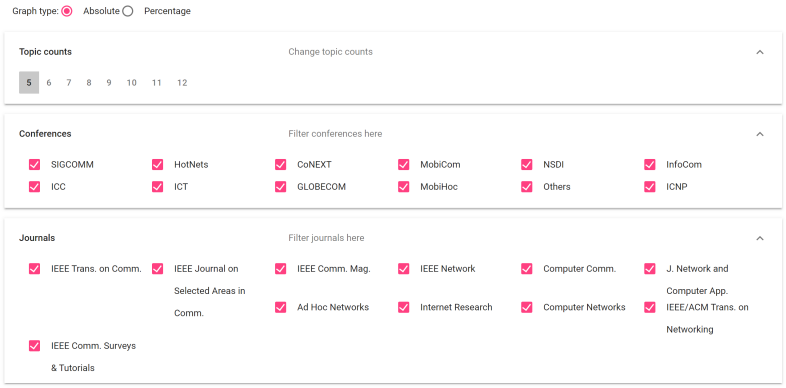

Figure 3.4: State graph filters

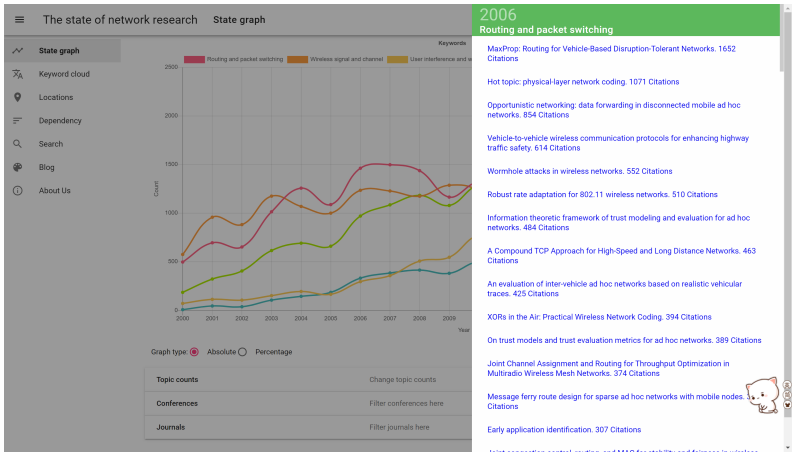

Figure 3.5: Detailed information when clicked

Keyword Cloud

Keyword cloud shows the frequency of top keywords displayed in papers. As is shown in fig3.6, The keyword cloud page has a slide that can switch years from 2000 to 2019, which is useful to show the trend for top keywords between years.

Locations

Research locations shows where researches happen, and the most active institutions in the world as fig3.7. It contains an animation control button to let years change from 2000 to 2019 to view changes. Countries are also clickable and most active institutions thus will be displayed as in fig3.8.

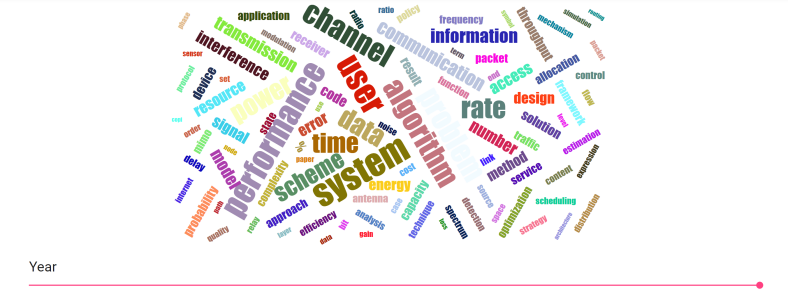

Figure 3.6: Keyword Cloud

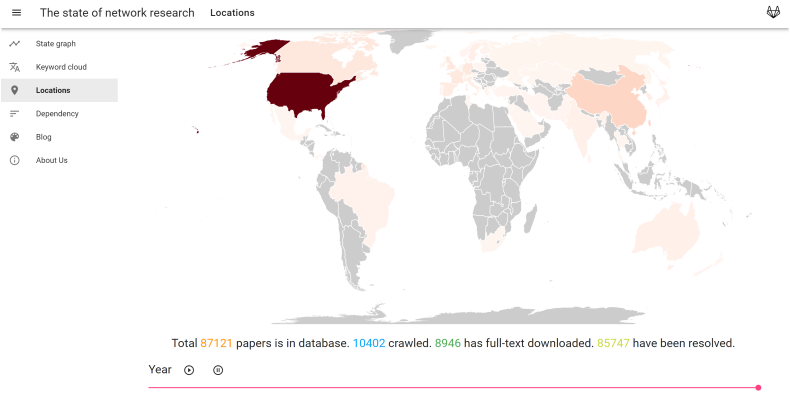

Figure 3.7: Research location

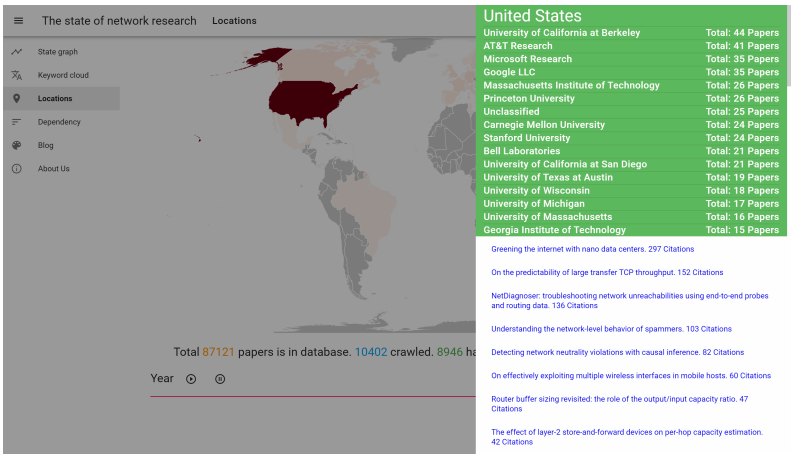

Figure 3.8: United States research details

Topic Dependency

Topic dependency describes how the topic cited each topic using a chord-dependency diagram. It has the same filters as we introduced in state graph. When clicking on topics, It will also provide citation and reference paper details in a citation way.

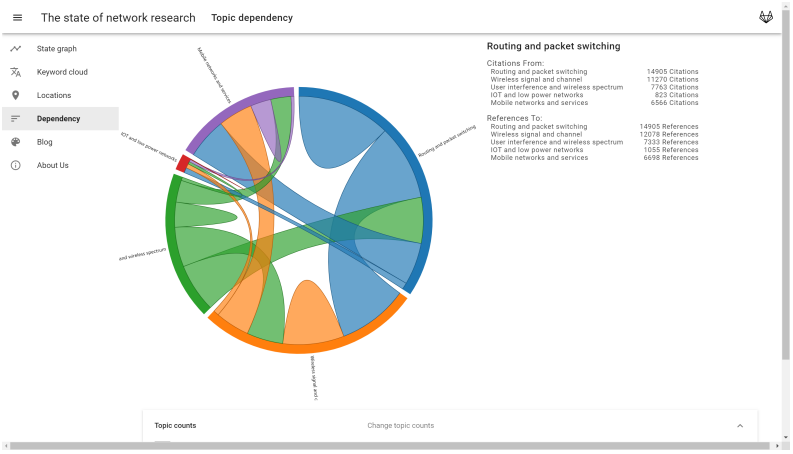

Figure 3.9: Topic dependency diagram

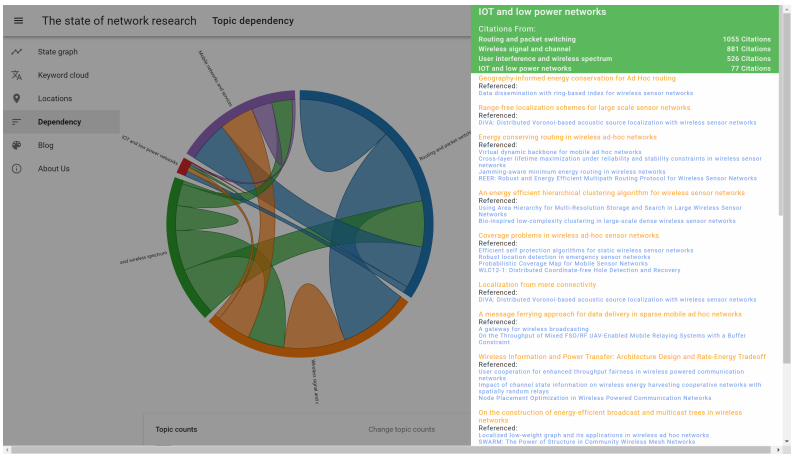

Figure 3.10: Topic dependency details

Blog

Blog is the place where authors and supervisors can publish their short articles.

Search

Search provides the ability to query results from our metadata database.

About Us

About us introduces the project and website.

3.5 Scalability

This project is designed as a scale-free system, Maintainers can easily import new papers into metadata database. The model for topic modeling is stored locally so the topic and keyword of new paper are easily and quickly obtained when added to database. For efficiency purposes, we've made some computing-heavy results persistent available in database, e.g. reference dependency, keyword and research locations. When new papers are imported to database, some scripts need to be run to update this pre-computed information.

Chapter 4

Contribution

In this project, our works including many parts:

- In data collection part, we programmed requests to filter metadata files from Crossref and format it into database.
- We obtained semantic scholar open corpus and build up a local database to store and query metadata from it.
- We developed DuckDuckGo result crawler for full-text link crawling, and downloader to acquire full-text.
- We implemented cosine similarity algorithm to check the correctness of downloaded full-text.
- We used TextRank and Lemmatization to extract keyword and pre-loaded them into database.
- We implemented NMF algorithm to do topic modeling for papers.
- The affiliation is identified by using emails of authors from paper full-text.
- We build up connections between topics by using citation counts.
- Server side application is programmed to process requests from client.
- We have made several pages: State graph, Keyword cloud, Research location, Topic dependency and Blog to show our results in an interactive way.

Chapter 5

Results and Analysis

By looking our data visualization website, some interesting results can be found and the reasons for these results is worth to analysis, as an inspiration for new researchers.

5.1 Major results

5.1.1 State graph

The state graph is the main point of our research. We have made topic counts available from five to twelve to have more interesting results. We have some interesting findings in many areas.

Firstly let's look at the overview of state graph. When topic count is 5, the topics look very uncorrelated: 1. Routing and packet switching, 2. Wireless signal and channel, 3. User interference and wireless spectrum, 4. IOT and low power networks and 5. Mobile networks and services. From here we can select each conference or journal to view which area they focus most. E.g. SIGCOMM focus mainly on the software part, It contains few papers about topic 2, 3 and 4. MobiCom focuses nearly on in mobile networks. But IEEE transactions on communications focus mainly in wireless networks, i.e. topic 2 and topic 3.

when topic count increases, some more interesting topic appears. Thus some more interesting results can be drawn from graph. When the topic count is 5, the paper about mobile networks increases dramatically, and stands on the absolutely top after the year 2011. When topic count rises to 10, this trend is more clear: all other topics keep stable while mobile networks keep increasing. When topic count increases to 12, we can see that the papers for user resource

allocation in wireless networks increased during years. Data encoding and decoding are always discussed but never become a top topic.

5.1.2 Keyword cloud

The keyword cloud always shows some common keywords during years like performance, system, algorithm, channel, etc. But some special keywords can still be found when changing years.

5.1.3 Research location

Of course, United States is the top research country. It is several times as the second country: China. The most famous and known labs and universities contribute most. Some countries also have a surprising performance: Brazil, Sweden, the United Kingdom, Canada, Japan and South Korea have many papers.

5.1.4 Topic dependency

We can also draw some interesting conclusions from topic dependency. When the topic count is 5, we saw that user interference and wireless spectrum cited almost every other topic. Routing and packet switching makes self-reference inside this topic as many as references to Wireless signal and channel. When topic count rises to 12, user resource allocation in wireless networks referenced a lot other topics but has fewer citations in.

5.2 Analysis

The results from our website are quite interesting.The state graph shows an overview trend for topics. But it does show something special: Mobile users are increasing dramatically, So the discussion of Mobile networks and services never stops. And User interference, Cognitive radio becomes more popular when spectrum resource is being exhausted nowadays. And new technologies, wireless relay becomes more famous after 2009. These conclusions we draw from our system can always have a reasonable explanation, which means our accuracy is high.

Chapter 6

Discussion

How is everything goes in network research? What is the trend for new topics? How should I start my research? In this project we have made answers available. Not everything is perfectly answered but researchers can be inspired by our outputs and may have a special idea of their research, Then our goal is achieved. Mining information from textual corpus is a painful work, especially when you are having hundreds of thousands of papers to read. We hope our works helps.

Chapter 7

Conclusions and Future work

7.1 Conclusions

In this project, we built, implemented and examined our state tracking system for network research. This system based on public published paper full-text and abstract, using topic modeling and TextRank keyword extraction for machine learning, mined and displayed information from tens of thousands of papers. The result is quite interesting and more information is waiting to be mined from our system. Our system is designed not only to simply display a single conclusion, but to provide an interactive data visualization for researchers to explore. The system is designed as scale-free, and is easy for the maintainer to add paper to it.

7.2 Limitations

In this project, around 80,000 papers were added to the database, however the curve in state graph is not smooth enough. For 12 topics, averagely there are around 350 papers of each topic per year, which is obviously not enough. The numbers of papers in database have a significant influence to machine learning algorithms like topic modeling. And for our topic modeling results, It is quite hard to adapt parameters when we have 80,000 documents, It has plenty of space for promotion. Moreover, the accuracy of getting affiliation data from e-mail addresses is not high since the e-mail address is not always easy to extract from paper, and some scholars use personal e-mail address instead. The keyword extracted from abstract is sometimes not typical and requires full-text for keyword extraction.

7.3 Future work

Our system is designed to continuously crawling data from internet and make output. Now most part is auto-updated, but to make it completely rid of maintainers, we have a lot of work to do. Also we can have some more interesting pages from our corpus, e.g. the page shows citations between authors, and the main topic of authors, so that researchers can find famous scholars to contact. Moreover, the resolution of PDF files has many many small bugs like symbol and character mis-resolving, which also needs to be solved. The project won't stop here, It will continues to be hosted and updated.

References

[1] D. J. de Solla Price, "Networks of scientific papers," *Science*, vol. 149, no. 3683, pp. 510–515, 1965. doi: 10.1126/science.149.3683.510. [Online]. Available: https://science.sciencemag.org/content/149/3683/510

[2] A. Kontostathis, L. M. Galitsky, W. M. Pottenger, S. Roy, and D. J. Phelps, *A Survey of Emerging Trend Detection in Textual Data Mining*. New York, NY: Springer New York, 2004, pp. 185–224. ISBN 978-1-4757-4305-0. [Online]. Available: https://doi.org/10.1007/978-1-4757-4305-0_9

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

[4] J. Kemp, "Metadata retrieval," [EB/OL], https://www.crossref.org/services/metadata-retrieval/ Accessed Aug 10, 2020.

[5] D. Eichmann, "The rbse spider — balancing effective search against web load," *Computer Networks and Isdn Systems*, vol. 27, p. 308, 1994.

[6] B. Pinkerton, E. D. Lazowska, and J. Zahorjan, "Webcrawler: finding what people want," 2000.

[7] D. Raggett, A. Le Hors, I. Jacobs *et al.*, "Html 4.01 specification," *W3C recommendation*, vol. 24, 1999.

[8] L. Richardson, "Beautiful soup: a library designed for screen-scraping html and xml," [EB/OL], https://www.crummy.com/software/BeautifulSoup/ Accessed Aug 10, 2020.

[9] A. Huang, "Similarity measures for text document clustering," in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, vol. 4, 2008, pp. 9–56.

[10] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek, and Bolikowski, "Cermine: automatic extraction of structured metadata from scientific literature," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 4, pp. 317–335, 2015. doi: 10.1007/s10032-015-0249-8. [Online]. Available: https://doi.org/10.1007/s10032-015-0249-8

[11] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," in *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, ser. PODS '98. New York, NY, USA: Association for Computing Machinery, 1998. doi: 10.1145/275487.275505. ISBN 0897919963 p. 159–168. [Online]. Available: https://doi.org/10.1145/275487.275505

[12] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411.

[13] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 30, pp. 107–117, 1998. [Online]. Available: http://www-db.stanford.edu/~backrub/google.html

[14] M. Bostock, "D3.js - data-driven documents," [EB/OL], https://d3js.org/ Accessed Aug 10, 2020.

[15] W. Pottenger, "Detecting emerging concepts in textual data mining," 10 2002.

[16] Y. Yang, J. G. Carbonell, R. D. Brown, T. Pierce, B. T. Archibald, and X. Liu, "Learning approaches for detecting and tracking news events," *IEEE Intelligent Systems and their Applications*, vol. 14, no. 4, pp. 32–43, 1999.

[17] R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. Namburu, *Data mining for scientific and engineering applications*. Springer Science & Business Media, 2013, vol. 2.

[18] R. Swan and D. Jensen, "Timemines: Constructing timelines with statistical models of word usage," 12 2001.

[19] L. Woolcott, "Understanding metadata: What is metadata, and what is it for?,," *Cataloging & Classification Quarterly*, vol. 55, no. 7-8, pp. 669–670, 2017. doi: 10.1080/01639374.2017.1358232. [Online]. Available: https://doi.org/10.1080/01639374.2017.1358232

[20] K. Reitz, "psf/requests-html," [EB/OL], https://github.com/psf/requests-html Accessed Aug 10, 2020.

[21] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999. doi: 10.1038/44565. [Online]. Available: https://doi.org/10.1038/44565

[22] Y. Wang and Y. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[23] A. Bakharia, "Topic modeling with scikit learn," [EB/OL], https://medium.com/mlreview/topic-modeling-with-scikit-learn-e80d33668730 Accessed Aug 10, 2020.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[25] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[26] X. LIANG, "Understand textrank for keyword extraction by python," [EB/OL], https://towardsdatascience.com/textrank-for-keyword-extraction-by-python-c0bae21bcec0 Accessed Aug 10, 2020.

[27] N. Downie, “Chart. js| open source html5 charts for your website,” *Chart. js*, 2015.