De transnationale Universiteit Limburg is een uniek samenwerkingsverband van twee universiteiten in twee landen: de Universiteit Hasselt en Maastricht University.

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculteit Wetenschappen

School voor Informatietechnologie

master in de informatica

Masterthesis

Exploring the utilisation of Large Language Models within document relation visualisations

Jan Robyns

Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :
Prof. dr. Gustavo Alberto ROVELO RUIZ

COPROMOTOR :
Prof. dr. Kris LUYTEN

BEGELEIDER :
De heer Gilles EERLINGS

2024
2025

Faculteit Wetenschappen
School voor Informatietechnologie

master in de informatica

Masterthesis

Exploring the utilisation of Large Language Models within document relation visualisations

Jan Robyns

Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :

Prof. dr. Gustavo Alberto ROVELO RUIZ

COPROMOTOR :

Prof. dr. Kris LUYTEN

BEGELEIDER :

De heer Gilles EERLINGS

Acknowledgements

Before heading into this thesis, I want to thank some people who have helped me throughout the period of writing my thesis, as well as throughout university. First, I want to thank my promoter Prof. Dr. Gustavo Alberto Rovelo Ruiz for his assistance and honest feedback throughout the thesis. His opinions during meetings always gave me a clear goal to work towards, making the master's thesis a slightly more pleasant and rewarding experience. I want to thank Prof. Dr. Kris Luyten and supervisor Gilles Eerlings for helping me with questions about AI and approaches towards Large Language Models and proofreading my thesis.

Besides the thesis, I want to thank my girlfriend, Lise, and close friends, Wesley and Mathias, who supported me throughout my academic journey. Finally, I want to thank my family, especially Prof. Dr. Pieter Robyns, for the inspiration and ideas that contributed to my thesis. Thanks to the support of my family, I managed to get through everything, even the most hectic times.

Samenvatting

Afgelopen jaren is het aantal academische publicaties sterk toegenomen, met jaarlijks duizenden onderzoeksartikelen en conferenties die bijdragen aan de onderzoekswereld. Online platformen zoals ArXiv, PubMed en IEEE Explore hebben het voor onderzoekers gemakkelijker gemaakt om hun publicaties te delen. Deze snelle groei heeft echter ook een keerzijde: een overweldigende hoeveelheid informatie, waardoor het steeds moeilijker wordt voor onderzoekers om relevante publicaties binnen hun vakgebied te vinden en te verwerken. Het principe van *information retrieval* verhelpt dit probleem deels te verlichten door middel van zoekmachines voor gerichtere zoekopdrachten, maar het probleem blijft bestaan vanwege de enorme hoeveelheid tekst en data die verwerkt moet worden om te bepalen of publicaties relevant zijn voor de gebruiker.

Naarmate de academische wereld groeide, ontwikkelde ook het vakgebied van Natural Language Processing (NLP) binnen AI zich verder. De ontwikkeling van de Transformer-architectuur en Large Language Models (LLMs) heeft de tekstgeneratiemogelijkheden van AI gerevolutioneerd. Modellen zoals ChatGPT en LLama worden nu veelvuldig gebruikt voor taken zoals schrijven, proeflezen en andere tekstgerelateerde taken. Hoewel deze evolutie al veel impact heeft gehad, bieden deze technologieën ook aanzienlijke mogelijkheden binnen de academische wereld. Zo kunnen LLMs publicaties samenvatten tot beknopte teksten, waardoor onderzoekers sneller kunnen bepalen of een paper relevant is. Platformen zoals Elicit en Scite gebruiken dit, maar sommige streven ervoor om verder te kijken dan een individuele publicatie. Het visualiseren van hoe publicaties zich tot elkaar verhouden en elkaar beïnvloeden is waardevol, omdat dit onderzoekers helpt om een breder inzicht te krijgen in hun vakgebied en de onderlinge verbanden. Deze thesis wil dit overzicht bereiken met visualisaties van publicaties met een assisterende kracht van de LLMs. Om dit doel te bereiken, wordt eerst een literatuurstudie uitgevoerd voorafgaand aan de ontwikkeling van de applicatie.

Literatuurstudie

Deze thesis heeft twee hoofdgebieden als literatuurstudie. Het eerste gebied bestudeert de ontwikkelingen binnen AI, met name in NLP en de recente vooruitgang op het gebied van LLMs. Om dieper in te gaan op LLMs is er onderzoek gedaan naar hoe deze modellen en NLP visualisaties kunnen ondersteunen. Het tweede gebied bestudeert hoe publicaties worden weergegeven door middel van visualisaties, met de nadruk op relaties op basis van citaties, publicatiedatums en onderwerpen. Deze kunnen worden weergegeven in de vorm van grafen en tijdlijnen, wat gebruikers de mogelijkheid biedt om trends, invloeden en relaties tussen publicaties te begrijpen. Verder bestuderen we representatiemethoden en *best practices* voor het presenteren van grafen, waarbij ook inzichten van de Gestalt-principes en de richtlijnen van Tufte worden meegenomen.

AI en NLP

Binnen het kader van het onderzoek over AI, is er diepgang over de evolutie van NLP en Transformers. Een eerste verbetering bevindt zich in representatie van woorden vanuit het perspectief van de computer. Via *word embeddings* kunnen computers woorden beter in een

context plaatsen doordat ze gerepresenteerd zijn in n-dimensionele vectoren in een vectorruimte. Op deze manier kunnen de computers de woorden mappen naar deze ruimte en woorden met een gelijkaardige semantiek samen plaatsen. Een verdere evolutie kwam er door het ontwerp van *Recurrent Neural Networks* (RNN's) die de mogelijkheid aanbiedt om invoerwaarden zonder een vaste lengte te kunnen verwerken. Dit is handig voor zinnen te verwerken, aangezien deze een variabele lengte hebben. Traditionele neurale netwerken zoals *Convolutional Neural Networks* (CNN's) hebben een vaste invoergrootte, wat er dus voor zorgt dat ze niet kunnen werken met sequentiële data. Hoewel de RNN's zeker een goede stap vooruit zijn, zijn er beperkingen, zoals het verlies van context. Voor dit tegen te gaan, gebruiken nieuwere modellen een *attention* mechanisme. Dit zorgt ervoor dat de modellen beter mee zijn met de context en dus betere resultaten leveren. Hoewel de resultaten verbeterd zijn, zijn deze modellen niet snel te trainen door hun sequentiële werking. Dit lost de Transformer architectuur op door parallellisatie toe te voegen. Daarbovenop maakt de architectuur ook gebruik van een verbeterd *attention* mechanisme genaamd *self-attention*.

Hoewel deze evolutie op zichzelf al nuttig is, onderzoeken we binnen deze thesis hoe LLMs ondersteunend kunnen zijn binnen visualisaties. We hebben toepassingen gevonden waarbij LLMs worden gebruikt in visualisaties zoals tijdlijnen, waarin *storytelling* en annotaties worden gebruikt om gebruikers te betrekken en te informeren over de gegevens in de grafieken. Binnen deze scriptie onderscheiden we drie ondersteunende rollen van LLMs: samenvatten, aanbevelingen en het labelen van onderwerpen.

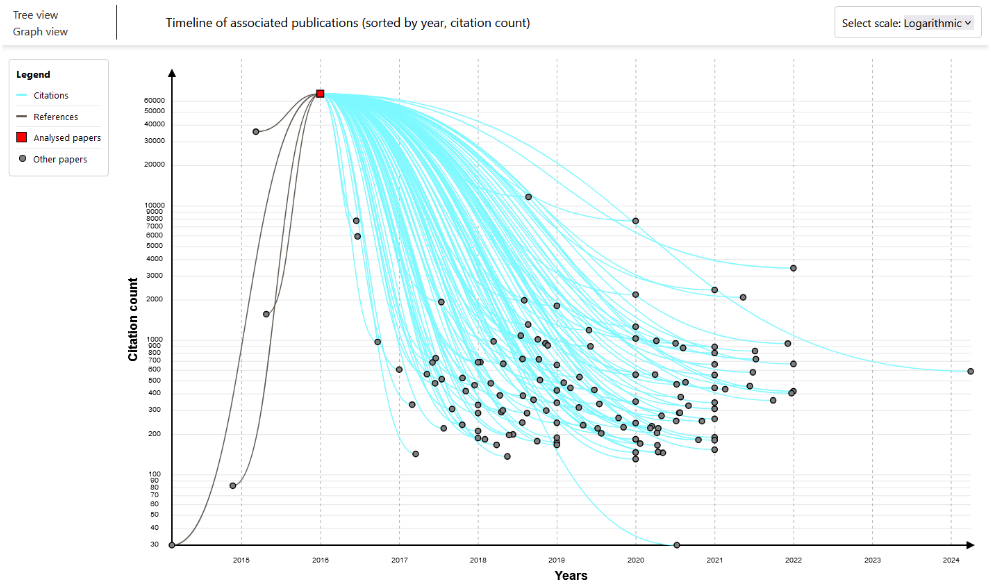
Visualisaties van publicaties

Binnen het kader van onderzoek naar publicaties, onderzoeken we manieren hoe we data van publicaties kunnen weergeven. Een veel voorkomende representatie is het gebruikmaken van een citatienetwerk of ook wel een graaf van publicaties. Hierbij zijn de nodes de publicaties en de linken tussen deze nodes de relatie van publicaties tussen elkaar. Hoewel het een goede visualisatie is, heeft het een probleem met de schaalbaarheid van de visualisatie. Als men een groot aantal nodes en linken probeert weer te geven, kan het groot aantal elementen voor extra visuele overbelasting zorgen. Dit kan opgelost worden met technieken zoals *edge bundeling*, waarbij de linken van een node samengebundeld worden. Een andere techniek is *transitive reduction*, wat ervoor zorgt dat indirect links tussen nodes werden uitgehaald. Een andere methode voor de visuele overbelasting tegen te gaan is door nodes te groeperen op basis van hun onderwerpen. Dit gaat op twee manieren: met *topic modeling* of met *clustering. Topic modeling* zorgt ervoor dat datapunten samen onder eenzelfde label vallen, maar geeft ook de mogelijkheid dat een datapunt onder meerdere onderwerpen valt. Het algoritme sorteert deze datapunten op basis van statistische modellen. *Clustering* daarentegen groepeert de datapunten altijd onder één cluster, wat ervoor zorgt dat een datapunt niet bij meerdere clusters kan zitten. *Clustering* bepaalt dit met de afstand tussen datapunten. Een voorbeeld hiervan is *agglomerative clustering*, waarbij elk punt als een cluster begint. Als de afstand tussen twee punten kleiner is dan een voorafbepaalde limiet, worden de twee punten samengevoegd onder dezelfde cluster. Het algoritme eindigt als er geen punten meer samengevoegd worden.

Hoewel deze visualisatietechnieken interessant zijn om een overzicht te creëren van de onderwerpen die er voorkomen binnen een citatienetwerk, is het ook interessant om tijdlijnen in het achterhoofd te houden. Binnen de context van publicaties, kunnen tijdlijnen helpen om een evolutie binnen een onderwerp weer te geven. Zo kan men bepalen welke richting een onderwerp kan uitgaan en welke invloeden publicaties hadden. Hoewel deze visualisaties goede voorbeelden zijn, zijn er ook nog de fundamentele principes die men kan volgen. Hierbij komen de Gestalt-principes en de richtlijnen van Tufte bij kijken, die algemene regels opstellen bij het maken van visualisaties. Deze regels kunnen dan ook meegenomen worden naar de ontwikkeling van de applicatie

Ontwikkeling

Tijdens de ontwikkeling van de applicatie zijn twee visualisaties gemaakt die samen een systeem vormen bestaande uit een front- en backend. De frontend bestaat uit een webpagina ontwikkeld met D3.js en Svelte om de twee visualisaties voor te stellen. Op de eerste pagina bevindt zich de tijdlijn die is opgesteld op basis van een geanalyseerde publicatie. De tijdlijn toont de referenties en citaties van het artikel door de jaren heen. De publicaties worden weergegeven als datapunten, waarbij interactie mogelijk is om informatie over elke publicatie te tonen. Binnen de visualisatie vertegenwoordigt de x-as de tijd per jaar, terwijl de y-as het aantal citaties van een publicatie weergeeft. Met deze visualisatie zien we dat veel geciteerde artikelen duidelijk opvallen. De volledige visualisatie is te zien in Figuur 1.



Timeline of associated publications (sorted by year, citation count)

Select scale: Logarithmic

Figure 1: De tijdlijn van een vooraf geselecteerd artikel.

De tweede visualisatie is een geclusterde citatiegrafiek die publicaties groepeert per onderwerp, waardoor onderzoekers zich kunnen richten op specifieke vakgebieden. De clustering wordt uitgevoerd in de backend, waar abstracts worden geanalyseerd met behulp van een *semantic* Transformer om *sentence embeddings* te genereren. Vervolgens wordt agglomerative clustering toegepast om abstracts met een vergelijkbare context te groeperen op basis van de afstanden tussen hun embeddings. Na het clusteren gebruiken we LLMs om de onderwerpen van de clusters te definiëren door de abstracts van de publicaties binnen een cluster te analyseren. De resulterende clusters worden weergegeven in de frontend, zoals geïllustreerd in Figuur 2. De visualisatie zorgt ook voor interactie met de publicaties zoals bij de tijdlijn. Via deze interactie kan de gebruiker publicaties samenvatten tot een korter geheel. Naast het clusteren haalt de backend informatie op over publicaties via externe APIs zoals Semantic Scholar en analyseert het de referenties en bijbehorende citaties.

Het backendproces begint met het analyseren van één enkele publicatie, beginnend met de referenties in de bibliografie van de publicatie. De bibliografie wordt geanalyseerd op citatienummers en bepaalt waar elke referentie in de tekst wordt opgehaald. Deze secties worden geparset voor gebruik in de visualisatie om te tonen waar de referenties voorkomen. Vervolgens wordt extra informatie, zoals het aantal citaties, het aantal referenties en andere parameters opgehaald met behulp van de DOI van de referenties via de Semantic Scholar API. Nadat alle referenties zijn geanalyseerd, wordt de DOI van de geanalyseerde publicatie gebruikt om de citaties op te halen. Hierbij wordt gekeken naar publicaties die het artikel citeren. Twee stappen worden genomen om de citaties te analyseren: eerst wordt het artikel gedownload en geanalyseerd om te bepalen in welke secties het originele artikel wordt geciteerd. Vervolgens wordt aanvullende

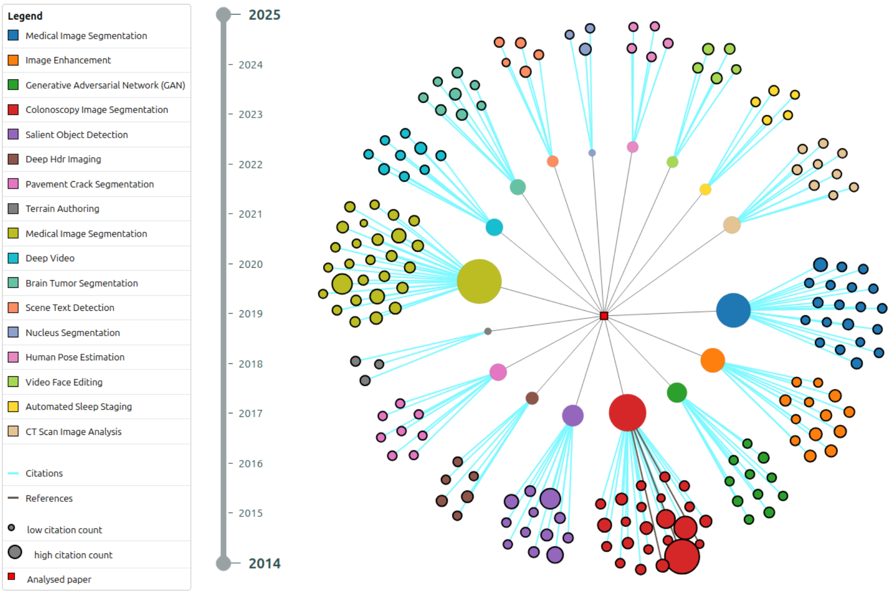


Figure 2: De clustergrafiekweergave van een vooraf geselecteerd artikel.

informatie opgehaald via de Semantic Scholar API. Wanneer deze stappen zijn voltooid, worden de gegevens van de referenties en citaties gecombineerd en naar de frontend gestuurd.

We voeren een vergelijkende studie uit van toepassingen die visualisaties en AI gebruiken en evalueren hun prestaties en functies in vergelijking met onze applicatie. Uit onze bevindingen blijkt dat de keuze van de applicatie grotendeels afhankelijk is van de doelen van de gebruiker. Gebruikers die een onderwerp breed willen verkennen en gerelateerde inhoud willen doorzoeken, zullen eerder kiezen voor verkennende tools zoals de onze. Daarentegen geven gebruikers die een diepe, gerichte analyse willen uitvoeren de voorkeur aan meer gespecialiseerde oplossingen.

Hoewel het ontwikkelproces positieve resultaten heeft opgeleverd, blijven er enkele beperkingen. Prestatieproblemen kunnen het gebruiksgemak voor onderzoekers verminderen, omdat de backend-analyse lang duurt. Hierdoor konden we geen uitgebreide gebruikerstest uitvoeren met onze applicatie. Dit zorgt ervoor dat we minder zicht hebben op de bruikbaarheid van de applicatie, maar zorgt wel voor een basis voor toekomstig werk.

Conclusie

Deze thesis heeft de mogelijkheden onderzocht van het combineren van LLMs met visualisaties. Door gebruik te maken van geavanceerde AI-technologieën zoals LLMs hebben we een tool en visualisaties ontwikkeld die onderzoekers helpen bij het navigeren van grote hoeveelheden academische literatuur. Hiervoor hebben we een visualisatietool ontwikkeld bestaande uit een backend voor het verwerken en analyseren van publicatiegegevens en een frontend voor het visualiseren van de publicaties. We hebben LLMs op verschillende manieren ingezet om de visualisaties te ondersteunen, terwijl de focus op de visualisatie zelf behouden bleef. Ondanks positieve resultaten zijn er enkele beperkingen, zoals prestatieproblemen en het ontbreken van een gebruikersonderzoek, die mogelijkheden bieden voor toekomstig werk.

Summary

Over the years, the volume of academic publications has surged, with thousands of research papers and conferences contributing annually to the global body of knowledge. Online platforms such as ArXiv, PubMed, and IEEE Explore have made it easier for researchers to share their findings. However, this rapid growth has a downside: an overwhelming influx of information, making it increasingly challenging for researchers to identify and process relevant papers within their areas of study. While information retrieval systems mitigate this challenge by offering search engines for more targeted searches, the issue persists due to the extensive amount of reading required to process the vast number of papers and determine their content and relevance.

As the academic world has grown, so has the Natural Language Processing (NLP) field within AI. The development of Transformer architecture and Large Language Models (LLMs) has revolutionised AI's text generation capabilities. Models such as ChatGPT and Llama are now widely used by the public for tasks like writing, proofreading, and other text-related activities. While this evolution has been transformative, these technologies also hold significant potential in the academic world. For instance, LLMs can summarise publications into concise texts, helping researchers quickly determine a paper's relevance. Platforms like Elicit and Scite aim to facilitate this, but some seek to expand their scope beyond the content of individual publications. Visualising how publications relate to and influence one another would be valuable, enabling researchers to grasp their field's broader context and interconnections. This thesis aims to achieve this through the visualisation of the publications and using the power of the LLMs in an assisting role. To achieve this goal, a literature review is conducted before the application's development.

Literature Review

This thesis focuses on two primary areas of research. The first area explores advancements in AI, particularly in NLP and its recent developments involving LLMs. To further expand on LLMs, research was conducted to understand how LLMs and natural language applications can enhance the effectiveness of a visualisation. The second area examines how publications are represented through visualisations, focusing on aspects such as citation relationships, publication dates and topics. These can be visualised in graphs and timelines, giving the users the possibility to understand trends, influence and relations between publications. This includes studying representation methods and best practices for presenting graphs and drawing insights from Gestalt principles and Tufte's guidelines.

AI and NLP

This thesis's exploration of AI builds upon the evolution of NLP. A first evolution is focused on the word embedding representation, which transforms words into n-dimensional vectors representing a word's semantic meaning in a vector space. A pivotal development in NLP was the introduction of Recurrent Neural Networks (RNNs), which excel at handling sequences, such as sentences, due to their ability to process inputs of varying lengths. However, RNNs faced limitations that necessitated improvements, particularly in retaining contextual information. To address these challenges, attention mechanisms were introduced, enabling the model to focus on relevant parts of the input. Building on this, the most significant breakthrough came with the Transformer architecture, which incorporated self-attention mechanisms. This allowed the model to capture relationships within the data better while enabling parallelisation, significantly improving training efficiency and overall model performance.

While this evolution is helpful in its usage, we look at the options available within visualisation, specifically in a supporting role. We have found that other applications use LLMs within visualisations, such as timelines, use storytelling and annotation to further engage and inform the user about the information in the graphs. Within the thesis, we utilise three categories of supporting roles: summarisation, recommendation and labelling.

Visualisation of citation graphs

Our exploration of visualisations is primarily centred around representing citation graphs. Within the citation graphs, the nodes represent publications, and the links represent the relations between papers. As the amount of data increases, visual clutter can arise. Techniques such as edge bundling and transitive reduction reduce this clutter. A common practice is to group the publications based on their topics to combat the clutter. This grouping can be done using topic modeling and label propagation algorithms. Some researchers use algorithms like agglomerative clustering that assign publications to a cluster with similar content. These clustering techniques are widely utilised in citation graphs, but are not the only approaches available. Exploring other visualisation methods can further enhance how academic relationships and trends are presented.

For instance, creating a timeline with the publication release dates can reveal the evolution of topics over time. Applying the Gestalt principles and Tufte's guidelines is essential when considering these examples. These foundational rules ensure the creation of clear and intuitive visualisations, minimising confusion for viewers and enabling practical interpretation of the data. By applying these foundational principles and insights from existing visualisation techniques, we can develop clear and insightful visualisations.

Development

During the development process, two visualisations were created to achieve our objectives, which exist in a system consisting of a front and backend. The frontend consists of a webpage developed with D3.js and Svelte to create the two visualisations. On one page is the timeline visualisation based on a pre-determined analysis of a paper. The timeline visualises the papers' references and citations throughout the years. Using the publications as data points, the visualisation incorporates interactivity by displaying information about each publication. Within the visualisation, the x-axis represents the time per year. In contrast, the y-axis of the timeline represents the number of citations a publication has. This method of visualisation enables highly cited papers to stand out. The entire visualisation can be seen in Figure 3.

The second visualisation is a clustered citation graph that organises publications by topic, enabling researchers to focus on specific research fields relevant to their work. The clustering is performed in the backend, where abstracts are embedded using a semantic Transformer to generate sentence embeddings. Agglomerative clustering is then applied, grouping abstracts with similar contexts based on the distances between their embeddings. After this clustering, we use LLMs to define the topics of the clusters by analysing the abstracts of each paper belonging to their cluster. The resulting topic-based clusters are displayed in the frontend, as illustrated in Figure 4. The visualisation provides the means to summarise a publication that the user is interested in and allows the recommendation of unfamiliar topics for an author. In addition to clustering, the backend is tasked with retrieving information about publications

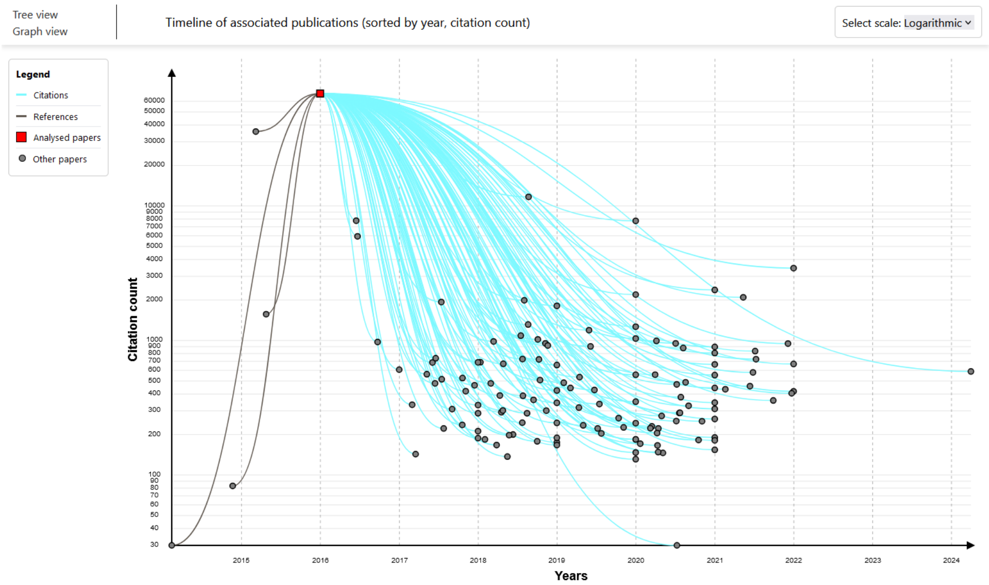


Figure 3: The timeline view of a pre-selected paper.

using external APIs, such as Semantic Scholar, and analysing the references and associated citations.

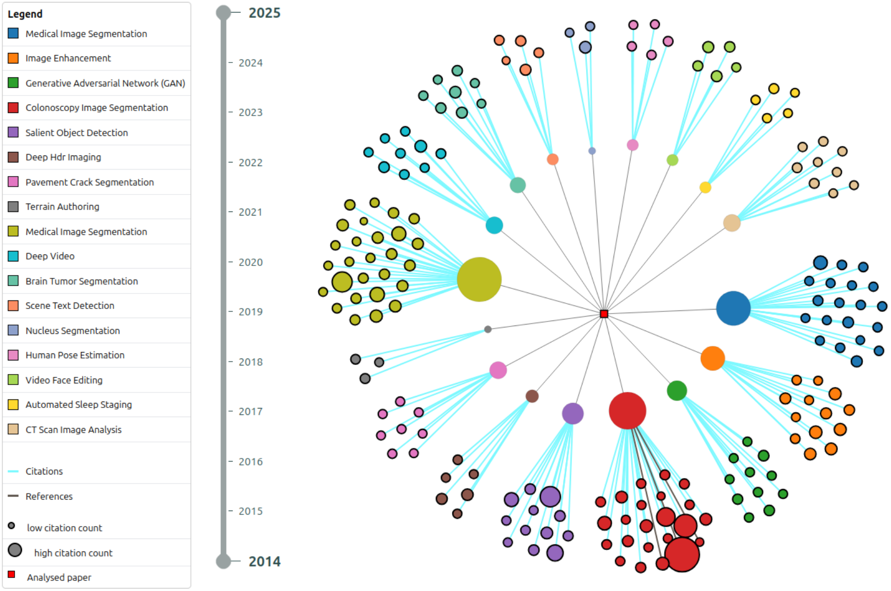


Figure 4: The cluster graph view of a pre-selected paper.

To begin this process, the backend begins by analysing a single publication, starting with its references in the paper's bibliography. It examines the bibliography for citation numbers and determines where each reference is cited within the paper. It then parses these sections for use in the frontend visualisation to show where the references occur. Subsequently, additional information like the paper's citation count, the citation velocity and other parameters is retrieved using the DOI of each reference from the Semantic Scholar API. Once all references are analysed, the DOI of the analysed publication is used to fetch its citations. To clarify, the citations refer to publications referencing the analysed paper in their text. Two steps are taken to analyse the citations: First, the paper is downloaded and analysed to identify the sections where the original paper is cited. Next, similar to the references, additional information is retrieved from the Semantic Scholar API. When these steps are completed, the data of the references and citations is combined and sent to the frontend.

We conduct a comparative study of various applications utilising visualisations and AI, evaluating their performance and features against our application. Our findings suggest that the choice of application largely depends on the user's goals. Users seeking to explore a topic broadly and navigate related content are more inclined to use exploratory tools like ours. In contrast, those aiming for a deep, focused analysis may prefer more specialised solutions.

While the development process yielded positive results, certain limitations remain. Performance issues may hinder the application's usability for researchers, as the backend analysis can be time-consuming. Additionally, due to the application's performance limitations, a user study was not conducted. This limits our oversight on the tools' usability, but lays the ground for potential future work.

Conclusion

This thesis has explored the possibilities of utilising LLMs in combination with visualisations. By leveraging advanced AI technologies such as LLMs, we developed tools and visualisations to assist researchers in navigating and understanding the vast landscape of academic literature. To do this, we created a visualisation tool consisting of a backend for processing and analysing the publication data and a frontend for visualising the processed publications. We applied LLMS to assist with visualisations in multiple ways that help represent data, while keeping the primary focus on the visualisation itself. With the development, limitations such as performance issues and a missing user study highlight areas for future work.

Contents

Chapter 1

Introduction

Within the scientific community, researchers share their findings, discover unknown research areas, and take risks when researching something new [For+18]. Other researchers then build upon the new branches, and the cycle of scientific growth repeats. The scientific world has become extensive, with many different topics and fields of study, and continues to grow during the 21st century. The internet was born along with the extensive development within the scientific world. Thanks to the development of the internet, people can share their research beyond conferences and presentations. Researchers can freely publish their findings on sites like ArXiv.com, ResearchGate.com and PubMed.com for other researchers to find. A steady growth can be seen in the data of these sites. For example, in computer science, a popular website to visit for publications is DBLP. The number of publications in this academic research database has grown yearly, reaching over 7 million across various types, as illustrated in Figure 1.1. However, this growth leads to the challenge of information overload. Researchers may struggle to identify high-quality, highly cited papers amid the vast number of publications, as they must classify the relevant from the irrelevant works. Conversely, researchers may search for papers containing the necessary information, but these papers may go unnoticed due to their low citation count.

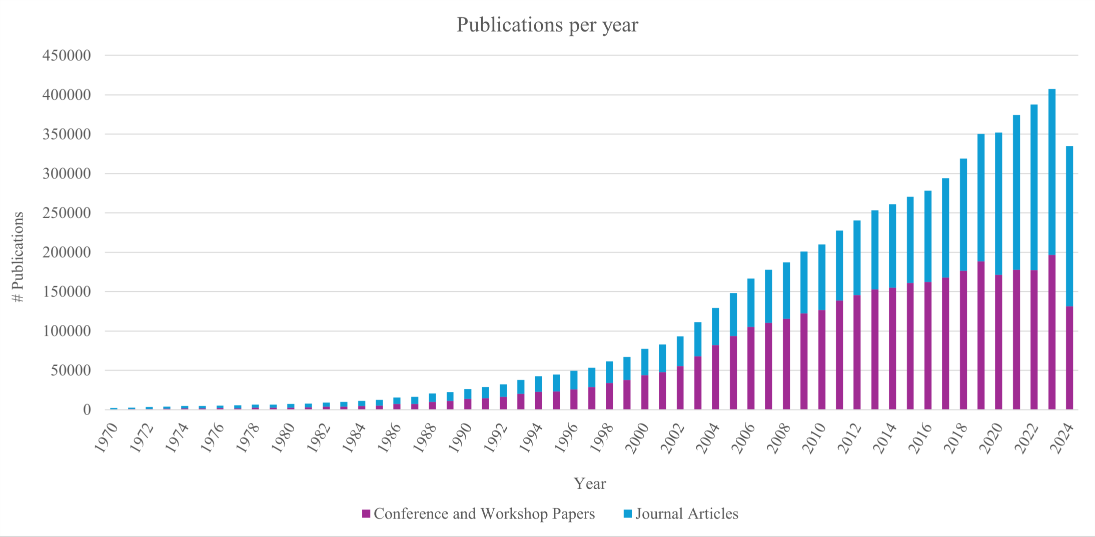


Figure 1.1: The number of publications over the years published to DBLP.org. Data taken from [dbl24] and consulted in November 2024.

To solve the problem of information overload, it is essential to develop efficient ways to navigate these spaces and collect relevant publications. For the general public, information can be gathered within web browsers, using search engines like Google, Bing and other popular engines. These engines are well-suited for a quick search, for example, to get a recipe, find a phone number for a business, and look up other daily information. However, researchers rely on similar tools to obtain the necessary information. Google Scholar, for example, is widely used by academics for its focus on research publications. It allows users to view citation counts, save publications, and even cite them directly. To further enhance literature reviews, researchers can also turn to specialised sites like DBLP, which provides an API for querying data via SPARQL, or Semantic Scholar, which offers similar API endpoints. These possibilities of using search engines and APIs contribute to a specific field called information retrieval (IR). Information retrieval concerns the organisation, processing, and access to all forms and formats of information. This enables users to find relevant information from an organised collection of documents [Cho10].

However, with the many possibilities of different IR systems, they present several challenges. With this large amount of data added to the internet, IR systems can have difficulty returning relevant information. Each IR model presents its own challenges. Take, for example, the Boolean IR model. This model uses purely boolean operations, specifically the AND, OR and NOT operations. Using this model, the query and terms within a document must be an exact match. Due to this characteristic, the model may return an excessive number of results if the query is too broad or only a limited number of results if the query is too strict [MRS08; RN19]. Another information system model is the Vector Space Model (VSM). VSM represents texts as vectors and determines these vectors with Term Frequency-Inverse Document Frequency (TF-IDF). When a query is prompted to this model, the query is converted to a vector. To match the query against the documents, the model uses cosine similarity to retrieve semantically similar documents. A significant drawback of this method is that it assumes that the words in the query are statistically independent, which is not always the case [MRS08; NMC17]. These challenges highlight a gap in existing systems, which poses the question: how can these systems be improved? While one approach could involve refining the systems, leveraging Artificial Intelligence (AI) advancements is another promising avenue. Over the past decades, AI has undergone rapid and transformative advancements, particularly in text processing. These developments present promising solutions to address the limitations of traditional IR systems. Unlike conventional methods that depend on specific keywords, Large Language Models (LLMs) can process natural language queries seamlessly. Moreover, LLMs maintain contextual awareness across a series of queries, enabling them to handle complex inquiries effectively. To understand how AI, especially LLMs, has reached this level of sophistication, it is vital to examine the broader trajectory of AI's growth.

From the 1960s until now, AI has steadily grown in various fields. Especially within the last two decades, AI has overtaken humans by performing tasks like reading comprehension, image recognition and language understanding, as seen in Figure 1.2. Each line indicates a benchmark used to determine the performance of different AI tasks, whereas the black line indicates the baseline for human performance. For example, GLUE is a benchmark for understanding LLMs in natural language. Despite these rapid gains, the results depend on specific use cases and benchmarks, which may give a distorted view of the actual AI evolution [Gia+23]. However, the evolution continues since data shows that LLMs become more potent as computation capabilities grow. With AI continuing to progress, it is becoming more accessible to the public, specifically LLMs [Gia+23]. Since the development of the Transformer architecture by Google [Vas+23], Natural Language Processing (NLP) has significantly advanced in its field. The architecture of the Transformer introduced the mechanism of self-attention, which allows LLMs to capture relationships between distant elements in a sequence. The invention of the Transformer model also allowed for scalability, processing much more data. With this evolution within the NLP field, multiple companies like OpenAI and Google invested and developed chat models that consumers can use to make their lives easier and search the internet. Models like ChatGPT and Gemini have cemented themselves as valuable tools for helping with question-answering and summarisation tasks.

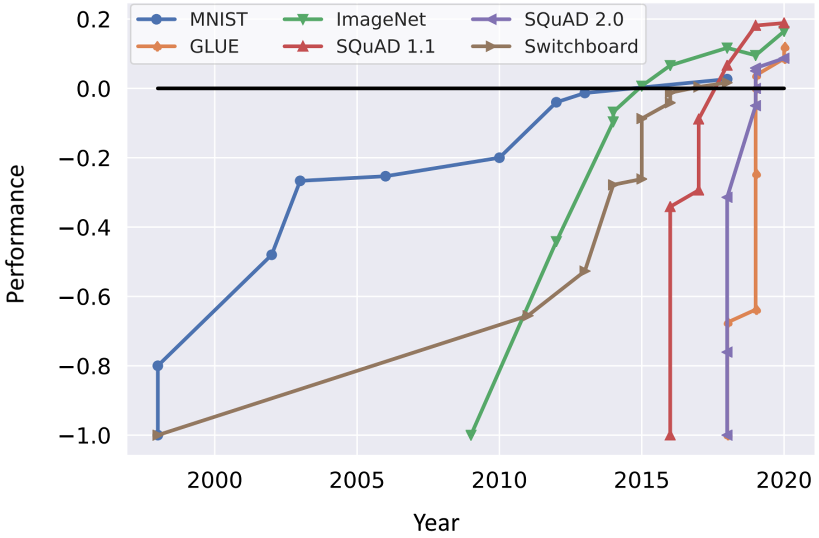


Figure 1.2: Different benchmarks for AI applications compared to human performance baseline (represented with the black line). (Visualisation taken from [Kie+23].)

With the expanding capabilities of AI and the growing volume of academic online data, there is an opportunity to leverage AI to process research publications more effectively. The internet hosts vast collections of academic papers, conference proceedings, and other scholarly resources, much of which remain underutilised due to their volume and complexity. Specific AI solutions may be more suitable depending on the type of data and the specific goals. As discussed earlier, Transformers have become a cornerstone of AI advancements, demonstrating their potential to revolutionise how academic data is processed and understood [Lin+22]. Multiple solutions like Elicit and Scite.ai already combine LLMs' abilities with publication data. However, only a few solutions, like ScienceOS, visualise the relationships within the data. As mentioned earlier, Elicit is an example that uses LLMs to summarise publications. Researchers can prompt the AI with a question. The AI fetches related papers with a source and answers the question based on those sources. After that, the researchers can determine if they want to inspect the publications displayed in a separate table. With this table, it is possible to grab specific sections from those publications and summarise them with AI. This enables researchers to compare and identify similar papers by reviewing the summarised sections. However, the links cannot be represented in a visualisation that encapsulates the relationship between the publications. While visualisations are optional and can be complex to represent correctly, they can be a great addition to interpreting data clumps and seeing relations that could not be seen beforehand. Good visualisation can give an overview of the information and allow clear communication with peers [Hea24]. An example of such a visualisation is by Morris et al. [Mor+03], which gives information about the *research fronts*. These *research fronts* indicate the evolution of research topics that have developed over time, providing an overview of the different branches within a topic.

While using AI and visualisation is a start for academic visualisations, other document types can be explored. Take, for example, the documents within a company, such as meeting notes and client agreements. Many companies use meeting notes to keep track of discussions in meetings with partners or other clients. For a larger company, this volume could lead to overwhelming documents, potentially causing confusion among employees who lack oversight of the bigger picture. Confusion can be prevented by giving IDs to these documents and linking them together. An interesting addition would be to attach an LLM to the company documents to easily retrieve information by querying meetings or contract details. A timeline visualisation of the documents could provide insight into the progression of meetings. While these ideas are interesting for future work, the main focus will be on academic publications as a proof of concept.

This thesis combines LLMs and publication data to improve research methods and processes while providing the option to have better oversight of the surrounding research through visualisations. Visualising the relationships between citations can provide insights into trends, showing how publications contribute to their topic and how authors could contribute to other fields. The visualisation allows one to click through the different citing papers and show relations to discover new papers within a similar field. In the visualisation, the other publications are grouped by topic to make navigating and filtering various research areas easier. This grouping is possible by clustering the various publications based on the contents of the abstract. Visualising simplifies the provided data and makes it easier for users to understand the content. This approach is a common practice in visualising citation graphs. However, this thesis enhances the experience by leveraging LLMs to identify topics and streamline the research process through publication summarisation. This method enables the dynamic generation of topics, uncovering new clusters that novice researchers might have overlooked. While keeping in mind these discussed points, the following research question is proposed:

How can LLMs be an addition in related document visualisations within an academic context?

To make the main objective more manageable, this broad research question is divided into two focused sub-questions:

- *How can LLMs support in visualising relationships between documents?*
- *How can we visualise relationships between related documents?*

To answer these questions, we first broaden our knowledge about LLMs and visualisations in the context of academic documents. With this knowledge, we build an application that assists researchers when doing a literature review through visualising a timeline and cluster graph with the assistance of LLMs. After developing the application, we compare its design and use of LLMs to other applications. This comparison is made between applications with a visual emphasis, such as CiteSpace and Connected Papers and applications with a heavy-focused AI implementation, such as Elicit and Scite.ai. However, to see the advantages of the LLM, we first need to understand the basics of how the LLM works, beginning with an introduction to Natural Language Processing (NLP).

Chapter 2

Related Work

This chapter addresses the topics relevant to answering the research questions. First, there is a discussion about NLP, including its evolution, the rise of Transformers, and applications of NLP within visualisations. Second, it examines different visualisations for citation graphs and highlights techniques that can improve representations involving publication data. Lastly, we discuss design principles according to the Gestalt Laws and Tufte's principles.

2.1 Natural Language Processing

This thesis focuses on identifying connections between academic publications, a task that can be approached in multiple ways. One way involves analysing the citation relationships, including both a publication's references and the papers that cite it, which can be retrieved from online databases. Another approach is to examine the content of the publications directly, grouping them into topics based on their semantic meaning. This analysis can be done using techniques from the NLP field [Joh+21]. Over time, NLP has used different methods and models to analyse texts. This section outlines the evolution within the field of NLP, going through the various models that led to the creation of the LLM.

2.1.1 Word embeddings

Before processing textual data, it is essential to find an efficient way for computers to represent words, which can be achieved through word embeddings. Word embeddings provide a method for machines to interpret words by capturing their semantic meaning and the context in which they appear. This representation is crucial for enabling computers to process text and establish connections between words based on their meaning and context.

A word embedding can be seen as a vector in an n-dimensional vector space. These can be classified into two types of embeddings: sparse and dense embeddings. First, sparse embeddings can be acquired by algorithms like TF-IDF and Pointwise Mutual Information (PMI). Second, dense embeddings can be acquired using the skip-gram algorithm. The sparse embeddings are known to have large dimensions and only have a small subset of non-zero elements. Dense embeddings only contain the non-zero values and have a lower dimension size than sparse embeddings. Because of this property, dense embeddings perform better than sparse embeddings since the dimensionality of the embeddings is reduced, meaning there are fewer weights to train when using AI models [JM24].

Embeddings provide an algebraic property, which means vector operations like addition and subtraction are possible. This property leans on the King-Queen example seen in [Mik+13] and Drozd, Gladkova, and Matsuoka [DGM16]. Because of this example, it can be said that the distance of word embeddings is correlated to the semantic meaning of the embeddings. This makes the distance between word embeddings an indicator of how similar two words are.

2.1.2 Recurrent Neural Networks

Although word embeddings are helpful, they are not used on their own. Multiple neural network approaches are possible, depending on the usage of the word embeddings. When trying to process sentences, there is a need to support a variable input size. Traditional networks like feedforward neural networks or multi-layer perceptrons have a pre-determined or fixed number of inputs, while sentences can be variable in length. This limit indicates that traditional networks cannot process sentences efficiently.

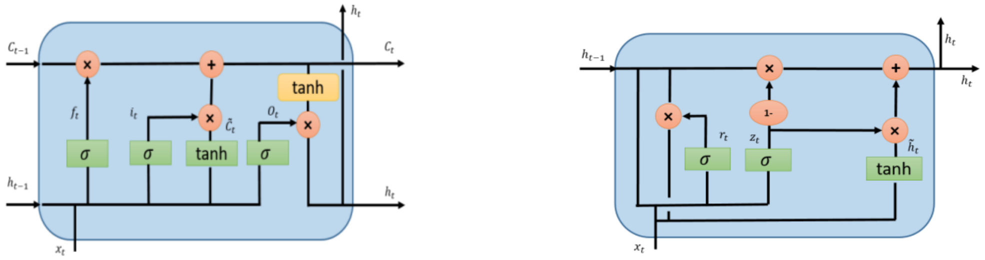
Another possible set of networks which can take a variable size of inputs is the Recurrent Neural Network (RNN) [Alo+18]. These models are used to learn long-term dependencies and sequential data types like time-series data [Sal+18]. Since sentences are like sequences of long words, these models can be used for language processing. RNN models have the unique property that the output of one step is fed back into the input of the next step. This mechanism gives the architecture a recurrent property to maintain information from previously processed information. However, due to this property, the RNN architecture struggles with the vanishing gradient problem [GSC00; Alo+18].

2.1.3 Neural networks: a simple overview

To explain the vanishing gradient problem, we need to take a step back and look at the basics of a neural network. A basic neural network consists of an input layer, hidden layers, and an output layer. When processing information through a neural network, we speak of forward propagation, so data flows from the input layers through the hidden layers into the output layer. The important thing to note is that the processing depends on the sum of the input value, a given weight, and a bias. This sum is then passed through a non-linear activation function, determining whether a node activates and information is passed on to the next layer. The actual learning of a neural network happens through backpropagation. This begins with comparing the network's output against the target and evaluating it with a loss function. After this evaluation, the network adjusts its weights using gradients to minimise the loss. When backpropagation is performed within RNNs, the vanishing gradient problem occurs. This occurs because RNNs need to backpropagate over long sequences, which makes the gradient shrink layer by layer [Che18]. Recall that the weights are updated based on these gradients and that the weights are used in the forward propagation to process the network input. If the weights do not get adjusted, the network does not learn. Two advanced architectures were developed to address this: Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs). These models introduced gating mechanisms to mitigate the vanishing gradient, enabling the network to effectively retain and process information over time [HS97].

2.1.4 LSTMs and GRUs

LSTMs use a cell state, which helps add or remove information that needs to be remembered. They do this by having three different gates, as shown in Figure 2.1a. These three gates are the input gate, forget gate and output gate. The input and output gates are meant to receive the previous iteration's output and pass on relevant information to the next iteration. The forget gate helps reset memory and blocks information once it is outdated [GSC00]. GRUs evolved from LSTMs since they have a similar structure, as seen in Figure 2.1b. The difference is that GRUs combine the input and forget gates, forming the new update gate to process information. This simplification gives the advantage of GRUs being computationally more efficient and faster to train. The drawback of this simplification is that the performance is lower than that of the LSTMs. While LSTMs and GRUs are a good start for NLP, there is a need for models that can remember information for longer texts and keep context in mind. An approach to doing so is chaining multiple RNNs in an architecture seen in sequence-to-sequence models.



(a) The LSTM model structure [Alo+18]. This structure includes the three gates: forget gate (f_i), input gate (i_t) and output gate (o_t). These gates regulate the flow of information, with the cell state (C_t) retraining long-term dependencies.

(b) Structure of a GRU model [Alo+18]. The structure uses two gates: the reset gate (r_t) to forget irrelevant past information and the update gate (z_t) to control how much new information is added. The hidden state (h_t) updates using the gates, combining past and current information.

Figure 2.1: The architecture of the LSTM and GRU models.

2.1.5 Sequence-to-sequence

LSTMs are often employed to handle the sequencing of multiple RNNs due to their ability to manage sequential data effectively. In a sequence-to-sequence architecture, one LSTM processes the input sequence and generates an intermediate representation of the input. This intermediate output is then passed to a second LSTM, which processes it further to produce the final output. This design underpins the sequence-to-sequence model's approach. Cho et al. [Cho+14] attempted to use this architecture, but they hit limitations regarding the performance on longer sequence sentences. Sutskever, Vinyals, and Le [SVL14] tried the same approach but had more success with three primary differences. First, they used two different LSTMs to process the input and output. Second, they opted for deep LSTMs, which outperform shallow LSTMs since they have more computational layers. Lastly, they mapped their input sequences in reverse, making the translation closer to their original input sequence. All these differences contributed to improved performance for translation tasks. Using these advancements, the sequence-to-sequence models outperform the standard LSTMs, thus advancing the goal of interpreting text.

Since the sequence-to-sequence model uses two models where one encodes and the other decodes, the model falls under the encoder-decoder family. The general working of these models is that the encoder must encode the sentence into a fixed-length vector. The decoder then decodes the outputs of the encoder into the required output. An example of this thought process is translating a sentence from one language to another, where the brain encodes the original sentence and simultaneously decodes it into the target language. The encoder-decoder structure has its flaws. To successfully process a sentence, the encoder needs to fit the sentence into a fixed-length vector, which can be challenging for long sentences. Bahdanau, Cho, and Bengio [BCB16] provides a solution by implementing an attention mechanism in the decoder. The decoder pays attention to specific parts of the source sentence. This way, the encoder does not have to encode all the information in the source sentence. This approach is a good improvement, but other factors now need to improve the effectiveness of the encoder-decoder model. There are three main problems. First of all, the structure has a slow training and inference speed. Second, it has trouble dealing with rare words. Lastly, it fails to translate all the words in the source sentence [Wu+16]. Wu et al. [Wu+16] tried to solve these issues in multiple ways. First, they connect the attention from the decoder network's bottom layer to the encoder network's top layer to address the performance issue. Through this connection, parallelism is introduced, which allows faster computing. They employ low-precision arithmetic on Tensor Processor Units (TPUs) for the inference issue. They use sub-word units for inputs and outputs to handle the rare words. Lastly, they employ a beam search technique to address the issue of only translating some parts of the source text. However, sequence-to-sequence models had a slow training process, making them hard to train. This issue was solved with the Transformer architecture, along with extra improvements surrounding the attention mechanism.

2.1.6 Transformers

The attention mechanism was a general improvement in the encoder-decoder architecture. Vaswani et al. [Vas+23] brought further additions to the architecture by releasing the Transformer architecture, the foundational architecture for multiple LLMs such as BERT and GPT models [Dev+19]. The main improvement is the introduction of self-attention, eliminating the need for recurrence in RNNs, which slows down the training process and enables Transformers to parallelise. Self-attention also handles long-range dependencies more efficiently, which RNNs struggled with. Since the Transformers do not use the sequential nature of RNNs, they lack the context of the order of input tokens and thus do not know how to interpret sentences correctly. However, this issue is solved with positional encodings, which are added to the input embeddings, giving information about the positions of tokens. Another key feature of the Transformer architecture is the Multi-Head Attention mechanism. This enhancement introduces multiple attention layers, each focusing on different aspects of token relationships, thereby improving the efficiency and accuracy of identifying connections between words. These additions resulted in the following model architecture that can be seen in Figure 2.2. While the past architectures primarily focused on Neural Machine Translation (NMT), the Transformer architecture opened the door for more applications like text generation, summarisation and question-answering [SZD24]. This led to other researchers trying to find optimisations for their cause. For example, Meta developed their open-source LLM, consisting of several modifications in the Transformer architecture and the training process [Tou+23]. Thanks to the Transformer architecture, it is now possible to have a high-level understanding of a text through summarisation techniques.

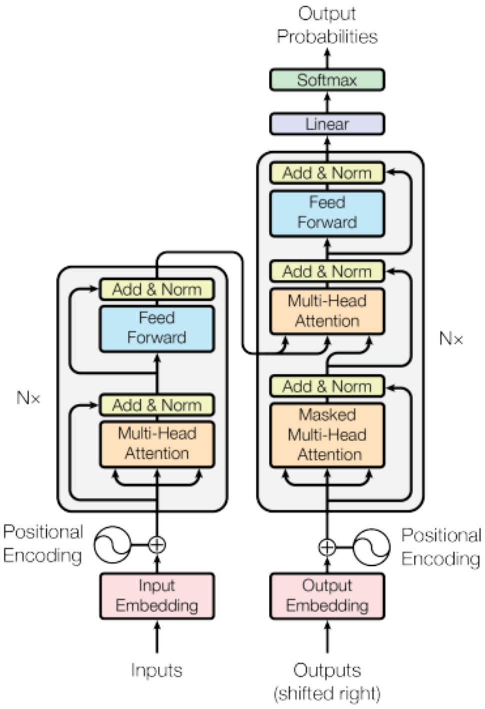


Figure 2.2: The Transformer architecture Vaswani et al. [Vas+23]. The Transformer uses multi-head self-attention to process input tokens in parallel, capturing relationships across sequences. The encoder (left) encodes input embeddings, while the decoder (right) combines context and masked attention to generate outputs.

By having an overview of the NLP evolution, we can see its importance in analysing text-based information and how the analysis of texts can be done through the final Transformer structure. LLMs are currently the go-to way to analyse text and can provide an easy experience when analysing long texts. Thus, they can be proven helpful for analysing academic texts. But how can LLMs enhance visualisations?

2.2 Exploring LLMs within visualisations

In this thesis, it is essential to understand how LLMs are used within visualisations to assess their usefulness in different scenarios. According to [Hut+24], there are several established applications of LLM technology, along with emerging opportunities and challenges in its utilisation. However, we will focus on how LLMs assist in visualisations, specifically enhancing their usability rather than generating visualisations. In this section, we will explore this particular application of LLMs.

2.2.1 Data processing and management

Data processing is a crucial component and the core of every visualisation. While it is essential for visualisation to provide a clear representation of the data, ensuring the data is accurate and complete is equally important. This can be achieved through pre-processing techniques tailored to the visualised data. [Hut+24] argues that integrating LLMs enables the pre-processing of large volumes of unstructured text, reducing the burden of data cleaning. For instance, LLMs can generate missing data, enhancing completeness and consistency. This principle is demonstrated in [Zha+23], which evaluates multiple LLMs, such as GPT-4 and Llama, for data processing and wrangling. The study examines their performance on tabulated data using a data-wrangling framework that incorporates prompting techniques, contextualisation, and feature selection, as shown in Figure 2.3. The results indicate that GPT-4 achieved a 90 percent success rate in error detection, data imputation, and entity matching, further reinforcing the capabilities of LLMs in text processing.

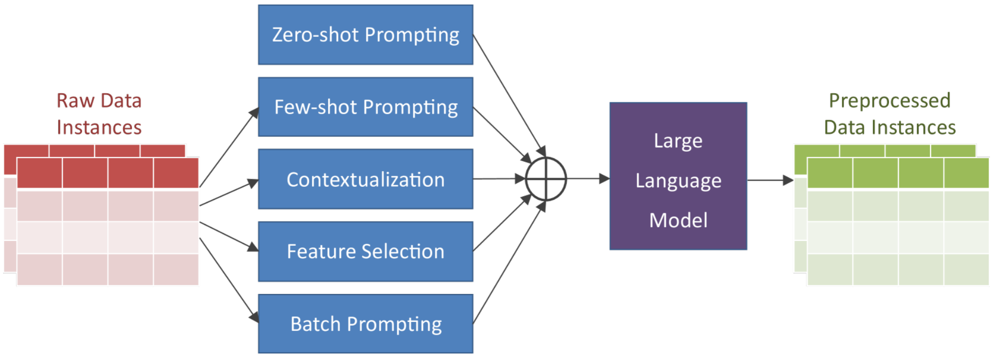


Figure 2.3: The data-wrangling framework from [Zha+23] illustrated. The framework represents the process that was used to test the different LLMs in terms of their capabilities of data augmentation through multiple prompting techniques (zero- and few-shot prompting) combined with contextualisation and feature selection. Batch prompting was used to lower the costs of the prompting. This pipeline results in LLMs producing preprocessed data instances.

This ability is also emphasised in [Din+24], which explores data labelling and other data augmentation techniques. The survey concludes that LLMs offer higher accuracy and lower bias in data labelling. Additionally, LLMs outperform crowd-worker annotations in annotation tasks, though they trade off some generalisation. Nonetheless, LLMs remain a reliable asset in data labelling.

2.2.2 Language interaction

A second enhancement within visualisations, according to [Hut+24], is the ability to have more interaction through language with the visualisation. This way of interacting provides the ability to query the visualisation through natural language, resulting in an accessible way of exploring the visualisation. A side effect of this interaction is the ability to navigate the visualisation without using English as the standard language. Thanks to the machine translation capabilities of LLMs, users can query in various languages. Voigt et al. [Voi+22] further highlights the areas where language interaction can occur within visualisation. They base their survey on the taxonomy of abstract visualisation tasks by explaining *why* and *how* interaction is performed with visualisations. This results in four visualisation tasks: present, discover, enjoy and produce. These are further divided into subtasks, highlighting how language can be used within visualisations per task. However, not all of these subtasks are relevant to the thesis. As mentioned before, this thesis only considers LLM implementations that have a supporting role in visualisations rather than a creative function.

Present

The primary goal of the present task is to guide the viewer through visualisation while efficiently showcasing the data. LLMs can accomplish this goal through two distinct methods: visual storytelling and explanation generation.

Visual storytelling involves telling a story through textual information while matching the visualisation. As mentioned in Voigt et al. [Voi+22], with visual storytelling, the users are guided through visualisation by interacting with generated stories. Through system animations, the system highlights essential details relating to the interacted text passage. An example of this can be seen in Metoyer et al. [Met+18], which investigates the assistance of natural language processing in sports journalism articles. They specifically focused on the NBA to couple pre-written articles with supporting visualisations that ground the article's narrative. They do this by performing text analysis, starting with identifying story elements. This analysis identifies the who, what, when, where, and why factors considered necessary when telling a story. These "Ws" can be uncovered using various text processing techniques. For the "What" question, they build a "story grammar", which is used to parse the text, which is a set of rules that identify or match various ways of mentioning statistics of players. When identified, they are used to train a support vector machine (SVM) classifier. To couple the text to the visualisation, they store the list of analysed Ws into a JSON file, which is then sent to a coupling component. The coupling component links sentences to the appropriate visualisation. After connecting the components to the text, the user can hover over the text, revealing the data related to the graphics.

Explanation generation differs from visual storytelling since the explanation is not bound to a driven narrative or linked to visual elements. The emphasis is more on explaining the visualisation rather than enhancing it. According to Voigt et al. [Voi+22], this opens the door for defining more complex visualisations and results. This complexity can be seen in Sevastjanova et al. [Sev+18], which discusses applying this principle to explaining machine learning models.

Discover

The discover task aims to help users explore and retrieve information from visualisations. This goal is achieved by providing tools that allow users to navigate the data, such as keyword searches, posing questions to an LLM for insights, or receiving browsing recommendations. Within Voigt et al. [Voi+22], two methods of achieving this are related to the goals of the thesis. The first one is discovering through the usage of keyword searches, and the second is discovering through browsing the visualisation.

Keyword search is a principle that has been applied outside of visualisations. Searching on the web happens with keyword searches and allows the user to browse through the vast amount of available data. Within the area of visualisation, the principle remains the same. When keywords that are important to the user are given, the visualisation gets filtered to show only the relevant results related to the query. An example of keyword search within a visualisation can be seen in Schleußinger and Henkel [SH18], which uses a search index that uses ElasticSearch. ElasticSearch[1] is a distributed search engine which provides search and analysis features for textual data such as logs, vector databases, or databases with textual data. Along with ElasticSearch and a TF-IDF model, they visualise only the relevant data for the user's query search. While this does not explicitly use LLMs as a means to discover information, the ability to give the user freedom through natural language is an essential aspect of visualisation.

Browsing allows the user to discover and explore a visualisation without any concrete questions, but it is guided by system-provided recommendations. This method gives the user freedom in exploration but leaves the window open for potentially interesting insights that the users have not considered. In Lee et al. [Lee+21], they create a system with multiple recommendation systems to guide the user and provide insights for datasets by adjusting three insight scores. The recommendation manager handles these scores and consists of *interestingness*, *relevance*, and *timeliness*, as well as insights from the data. *Interestingness* tries to capture how interesting a particular insight is. This score is calculated using the statistical properties and metrics of the underlying data. An example of this is the correlation recommender, which goes over the qualitative attribute pairs and uses the Pearson correlation score. When a certain threshold is crossed, the correlation is deemed interesting. *Relevance* is computed by matching it to the user's queries. For example, if the user wants to know more about "sales", then the *relevance* score should be higher for results that involve "sales". To properly get the relation between the query and the insight, they use a Vector Space Model (VSM). They can compute relevancy scores using this model by converting the context and the insights into vectors. Lastly, the *timeliness* score measures how relevant an insight is at a given moment in the conversation. This score helps prevent users from repeatedly seeing the same insight over time. To achieve this, the application tracks the frequency of displayed insights, phasing out those that appear too often, regardless of their *interestingness* or *relevance*. By balancing these three parameters, the application provides users with diverse insights based on their broad guidelines while avoiding repetition.

Within this thesis, the primary goals of the LLMs are to help researchers discover publications while playing a more background role. This assistance is delivered through the backend and frontend, with data processing and cleaning being a large part of the backend, while services such as summarisation and topic recommendation play a bigger part in the frontend. A detailed explanation of how this was implemented can be seen in Section 3.1.2 and Section 3.2.2

While the applications of LLMs and natural language extend beyond these examples, it is essential to understand their potential for representing visualisations, particularly in academic networks.

2.3 Visualisation of academic relationships

The visualisation of academic relationships is a broad topic. Many researchers have tried to find creative and new solutions that fully visualise the relationships between publications, authors and topics. This section explores visualisations designed to represent academic relationships and publication data. It also addresses challenges such as visual clutter and strategies to resolve these issues based on the type of visualisation. Finally, it discusses key principles to consider for creating clear, effective, and valid visualisations through the Gestalt Laws and Tufte's principles within the context of academic publications.

[1] https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro-what-is-es.html

2.3.1 Citation graph representations

The visualisation of relationships between academic publications is a thoroughly researched field that contains many ideas on properly presenting the information for different relations between papers. The primary visualisation method is a citation graph, where the graph nodes represent publications, and the graph's edges represent the relations between the papers. An example of this can be seen within Tan et al. [Tan+16], where this is coined the "Fundamental display". When the user selects a paper that they want to analyse, a network is generated from the references and citations that are linked to this paper. This paper is surrounded by other paper nodes, where each link to a node represents a reference depending on the direction of the arrow. Users can then click on nodes to expand the graph, showing references to the clicked node. An example can be seen of this in Figure 2.4, which shows the progression of clicking through multiple papers that expand the displayed graph.

This visualisation is a valid approach for a small dataset, but can quickly become chaotic as more publications join the graph. Due to this chaos and disorder, it is hard to present the citation graph. While metadata like the author name or the paper titles are valuable for the user to identify the data and gain additional information on the publications, the data is left out for enhancing the clarity of the visualisation [KH17]. A solution to this is to bundle many edges together. This way, visual clutter from the edges is reduced. Within the graph, there may be arrows drawn to indicate citations. This approach can be conceptualised as a directed graph, in which the nodes represent individual publications and the directed edges denote citation relationships. A directed edge from one publication to another indicates that the latter cites the former. The arrow can also be an addition to visual clutter, making it easier to represent the edges for different colours. One colour can be for references and another for actual citations [NIS15]. This example can be seen in Figure 2.5, which shows that when many edges connect to a node, they get bundled into one larger edge. Another approach to combat visual clutter is the reduction of unnecessary edges. This practice is called transitive reduction [vW14a; Clo+14]. Within the context of a citation graph, this method can be used to reveal academic papers that play an essential role in the causal structure of the network. Publications with a higher citation count are less critical regarding causal structure. As seen in Figure 2.6, there is a significant reduction in the number of edges due to the transitive reduction algorithm. In addition to improving edge visibility, another approach to enhance the readability and interpretability of citation graphs is the grouping or clustering of nodes based on shared characteristics or relationships, which helps to highlight patterns and structural connections within the network.

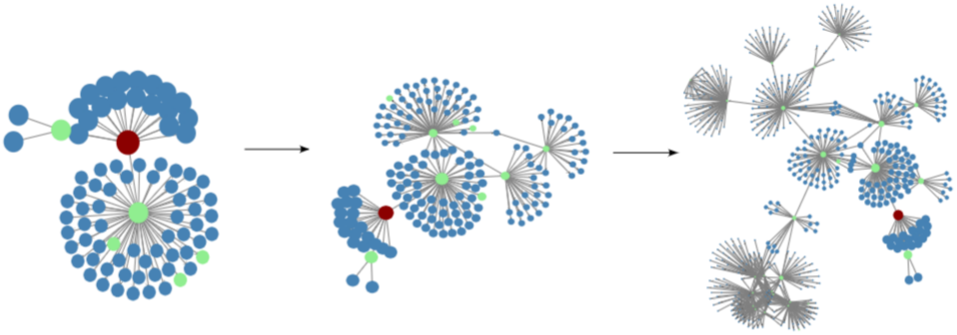


Figure 2.4: The fundamental display of a citation graph, showing the progression of clicking through different nodes, expanding the citation network [Tan+16].

2.3.2 Grouping algorithms

Various algorithms are used to group nodes or publications into thematic categories, enabling researchers to analyse citation graphs efficiently. A first example, as seen in Nakazawa, Itoh,

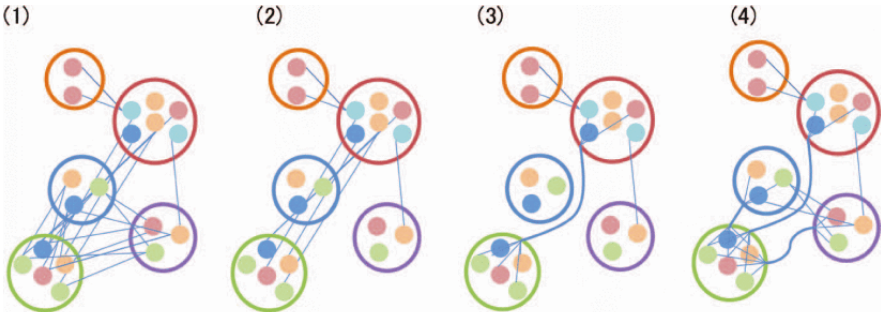


Figure 2.5: The process of edge bundling in a citation graph [NIS15].

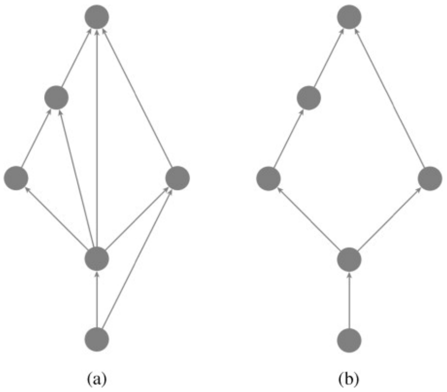


Figure 2.6: The transitive reduction algorithm on an example graph [Clo+14].

and Saito [NIS15], implements a grouping method that categorises papers based on their content with a technique called Latent Dirichlet Allocation (LDA). LDA is a simple topic modelling technique that assumes documents are composed of multiple topics. For example, an article may discuss topics like machine learning, genetics, and physics, with words strongly linked to each topic. LDA models these topics statistically and identifies the underlying topic structures of a document collection as if they were generated through a probabilistic process [Ble12]. Second, other researchers employ alternative topic models or clustering algorithms to classify publications [EW17; vW14b]. Topic models apply unsupervised learning to analyse text, producing sets of terms that define emergent topics [JM24]. This approach is convenient for classifying publications with overlapping themes where the exact keywords appear in different contexts. Lastly, the Label Propagation Algorithm (LPA) presents another way of forming clusters. Each data point starts with a unique label. Each node updates its label at each iteration to match the one most frequently held by its neighbours. If there's a tie between multiple labels, one is chosen randomly with equal probability. While this propagates through the network, the dense node groups reach a consensus on which label they belong to. At the end, nodes having the same label are formed as a community [RAK07]. Tan et al. [Tan+16] uses this method for their hierarchical display. After the algorithm is done, they analyse the publication titles per cluster to find the common keywords for that cluster. Based on the common keywords, a topic is picked as the primary keyword to describe the cluster. This grouping example can be seen in Figure 2.7, where it is visualised in a circle packing diagram, with each circle being a topic. While topic modelling is a valid option, other options like clustering are used to group data together.

Clustering is another unsupervised machine-learning technique that combines data points based on distance metrics or the similarity of two data points [Gao+23]. There are various types of clustering, such as partition-based and hierarchical clustering, each with strengths and weaknesses depending on the use case [Fah+14]. Within partition-based clustering, k-means clus-

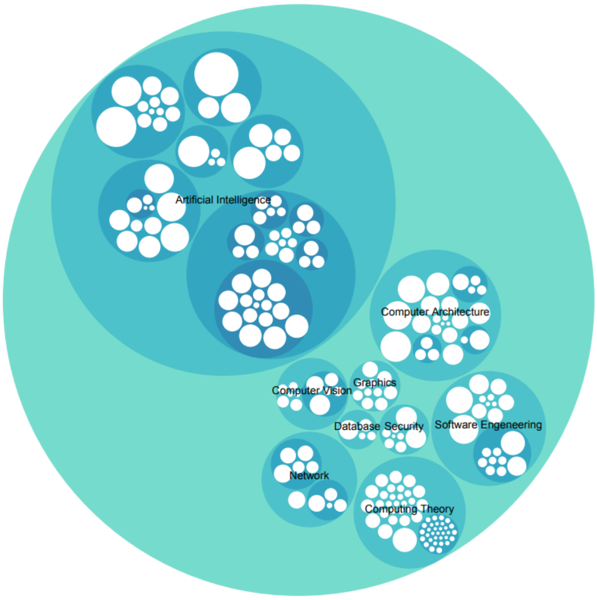


Figure 2.7: A circle packing diagram displaying the topics formed from the Label Propagation Algorithm [Tan+16].

tering is a well-known algorithm with the core idea of iteratively updating the cluster centres, which are calculated as the mean of the data points within each cluster. This process continues until a specified convergence criterion is met. An advantage of k-means is that it is highly computation-efficient, with the drawback being that there must be a pre-determined number of clusters. The opposite can be said about hierarchical clustering, where agglomerative clustering is a popular algorithm. With agglomerative clustering, the algorithm goes bottom-up, where each data point begins as a cluster, and clusters merge based on a pre-determined threshold, making it a valuable algorithm when there is no need to have a pre-determined number of clusters [Fah+14; Ezu+22]. An example of hierarchical clustering can be seen within Morris et al. [Mor+03], where the influences of papers can be shown on different research fronts. To determine the research fronts, the authors performed hierarchical clustering on the abstract to get different research fronts. This can be seen on the graph in the y-axis in Figure 2.8, where the dendrogram is displayed. Despite the differences between topic modelling and traditional clustering, both methods are valuable for grouping data. Topic modelling captures overlapping themes and relationships, while clustering is more suited for exclusive group assignments. The choice between these approaches depends on the specific requirements of the analysis. The clustering of topics is a good general approach. However, alternative representations can address temporal questions that clustering alone cannot answer. For instance, questions like “Is this paper still being cited today?” or “How old are the references cited in this paper?”

2.3.3 Temporal representations

While the citation graph is a common way to represent publications, there are various ways to represent the data without forming a graph. One way is to present the publications in a timeline with a temporal scale and the publication’s release date. This representation shows which papers are more recent and distinguishes between older and newer publications. This presentation is relevant when researchers need to search for more recent publications and information. This representation gives us another dimension to work with and visualise relationships between citations using time as a relation. There are different ways to represent such a timeline. Matejka, Grossman, and Fitzmaurice [MGF12] uses a grand timeline where the selected papers’ ancestors and predecessors are visually highlighted. The visualisation gives the ability to determine the

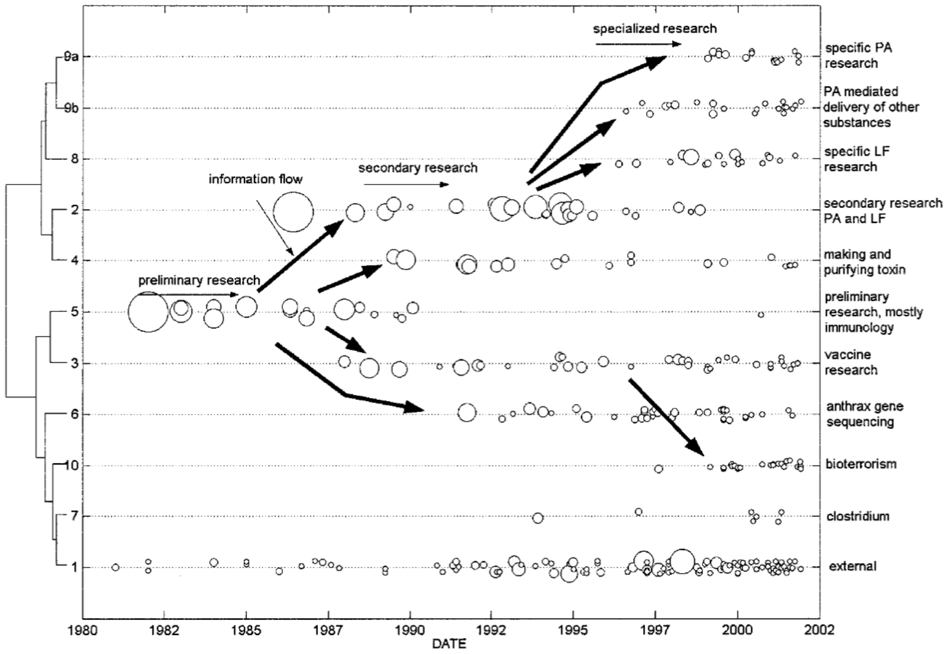


Figure 2.8: A visualisation that uses hierarchical clustering to group papers within multiple topics, forming research groups. The dendrogram of the clusters can be seen on the y-axis Morris et al. [Mor+03].

number of generations a user can go backwards or forward through the visualisation. Another timeline representation is in the circular tree form in Choi et al. [Cho+18]. Each node is a paper, and its size represents the number of citations. The inner circle represents the earliest time of a released paper, and the outer ring represents the newest released paper. The papers are clustered based on citations to create a citation network for the research papers. Another way to represent the timeline is by visualising the edges differently. Zhang, Jing, and Zhou [ZJZ18] embeds the temporal data inside the graph. This design was invented through a series of steps. First, they use arrows to represent the flow of temporal data, indicating which node comes first. This approach is possible for small datasets, but as mentioned before, this becomes problematic in larger datasets. They solve this problem by utilising curved edges and following them clockwise to represent the flow of time. Lastly, they add visual rectangles along the edges to represent temporal data. Each rectangle is a specific point in time, and the colour intensity indicates its importance. This visualisation results in a dual representation, with the edge indicating time flow and the rectangles indicating the details of the temporal data. This design process can be seen in Figure 2.9. Another approach to view the citation timeline is a paper's influence on new research fronts. This concept was discussed earlier with Morris et al. [Mor+03], where hierarchical clustering led to the dendrogram displayed in Figure 2.8. The timeline in this visualisation represents the influence that the analysed paper can have on future research. It could be considered an indicator of progress within specific fields, as well as an indicator of the possibility of expansion within topics. Although timelines are a fascinating visualisation aspect, other approaches also explore different methods for representing citation networks.

2.3.4 Other representations

While representations such as graphs and timeline visualisations are more conventional, others adopt alternative approaches. One example is provided by Brandes and Willhalm [BW02], who visualise citation networks using a landscape metaphor. In their approach, highly cited

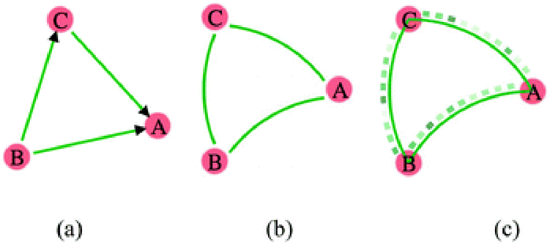


Figure 2.9: This shows the design process of how to represent temporal data in citation graphs using the edges of the graph [ZJZ18].

publications are represented as mountains, while less cited papers appear as minor elevations or flat areas. Through clustering, groups of less-cited publications are shown as villages at the base of these citation mountains, illustrating how these works relate to and build upon the central, highly influential publications. Another visualisation in Sun et al. [Sun+24] comprises multiple visualisations forming a composite visualisation. Composite visualisations represent designs with multiple visuals in a view [Zhu+25]. They argue that the existing academic tools are oversimplified and fail to convey the impact authors can have within their field. The visualisation represents a 3D prism-shaped figure that highlights the contributions of scholars, shows how topics evolve and tries to identify key works that contributed to their respective evolutions. The visualisation consists of three different modules. The first one is the GeneticPrism module, which is the 3D prism-shaped visualisation, where the top view represents the inter-topic citation influences with a polygon chord diagram. On the side panels, hierarchical graphs display the intra-topic evolution of research contributions over time. Second is the GeneticScroll, a topic-specific exploration that can be seen as a deep dive into a single research topic. Through this graph, the dynamics and connections can be seen with other topics. While the citation graph is the module's main presentation, multiple other visualisations like stream graphs, flow maps, and hexagonal glyphs exist. The last visualisation module is the topic chord diagram, which is a stand-alone view of inter-topic interactions that summarises the impact of a scholar's work within different fields. While the GeneticPrism is a large visualisation with many options, some researchers opt for a more compressed visualisation. Khazaei and Hoeber [KH17] try to improve search spaces like the internet by creating a compact visualisation for publications. They argue that conventional list-based search interfaces fail to support exploratory tasks. They try to improve this by introducing the Bow Tie Academic Search system, where visual encodings of citation data are added in search results, consisting of references and citations. This implementation aims to improve the comparison between articles and enables navigation through forward and backward citations. The visualisation in this paper is the bow tie representation. The structure consists of the centre of the bow tie, with the supporting left and right wings. The centre represents the article, with the left wing representing the references cited by the article and the right wing being the articles citing the centre article. The width and the height of the wings are also representation factors within the visualisation. The width of the wings represents the number of citations or references, while the height represents a temporal range, which enables the identification of older or newer publications. Additionally, they implemented a keyword visualisation, representing the most frequent keywords from the search results and a relation grid linking the keywords to articles. An example of the bow tie visualisation can be seen in Figure 2.10 along with the browser implementation of the bow tie in Figure 2.11.

Another example of a more compact visualisation is presented in [Dör+12] through the PivotPaths application. This visual interface represents data by linking authors to their resources and associating those resources with relevant concepts. The primary aim of the paper is to relate facets and items, encourage pivoting, and enable gradual view adjustments based on pivot interactions. The key contribution of this work lies in its approach to navigating the visualisation, referred to as "pivoting." Pivoting allows users to efficiently switch between filters,

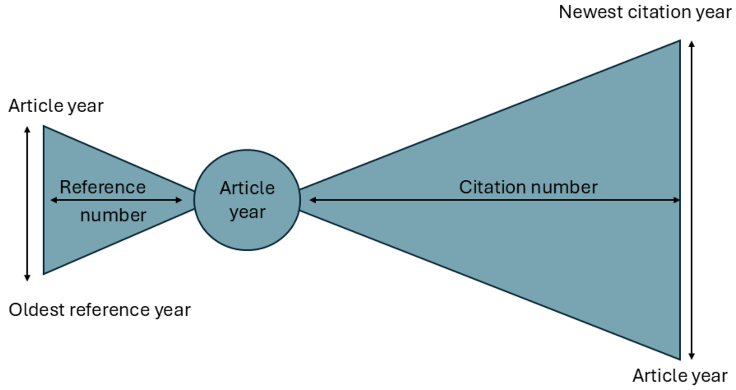
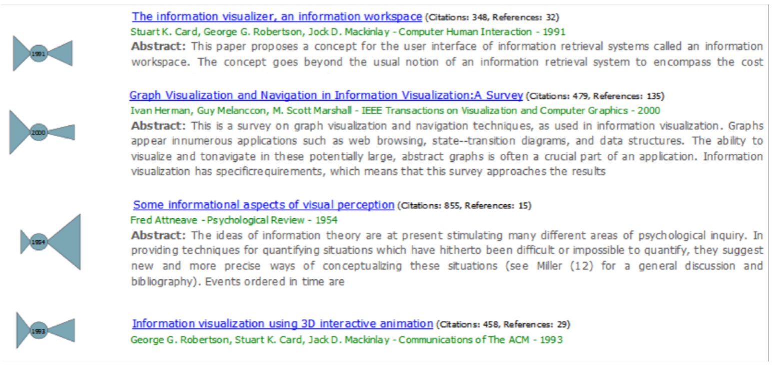


Figure 2.10: The bow tie structure, along with the representation of each element within the visualisation. (recreated figure from [KH17])



The information visualizer, an information workspace (Citations: 348, References: 32)
Stuart K. Card, George G. Robertson, Jock D. Mackinlay - Computer Human Interaction - 1991
Abstract: This paper proposes a concept for the user interface of information retrieval systems called an information workspace. The concept goes beyond the usual notion of an information retrieval system to encompass the cost

Graph Visualization and Navigation in Information Visualization:A Survey (Citations: 479, References: 135)
Ivan Herman, Guy Melanccon, M. Scott Marshall - IEEE Transactions on Visualization and Computer Graphics - 2000
Abstract: This is a survey on graph visualization and navigation techniques, as used in information visualization. Graphs appear innumerous applications such as web browsing, state--transition diagrams, and data structures. The ability to visualize and tonavigate in these potentially large, abstract graphs is often a crucial part of an application. Information visualization has specificrequirements, which means that this survey approaches the results

Some informational aspects of visual perception (Citations: 855, References: 15)
Fred Attneave - Psychological Review - 1954
Abstract: The ideas of information theory are at present stimulating many different areas of psychological inquiry. In providing techniques for quantifying situations which have hitherto been difficult or impossible to quantify, they suggest new and more precise ways of conceptualizing these situations (see Miller (12) for a general discussion and bibliography). Events ordered in time are

Information visualization using 3D interactive animation (Citations: 458, References: 29)
George G. Robertson, Stuart K. Card, Jack D. Mackinlay - Communications of The ACM - 1993

Figure 2.11: The bow tie visualisation implemented in the browser. (taken from [KH17])

resulting in an evolving visualisation with seamless animations. While primarily designed for academic purposes, this method can be applied in other contexts, such as visualising movie data—connecting actors, their films, and the categories associated with each movie. The main structure of the data is illustrated in Figure 2.12, which depicts the relationships between resources, authors, and topics. When integrated into a broader context, it forms the visualisation shown in Figure 2.13, where elements are filtered based on publications by C. Plaisant, highlighting collaborators and the topics influenced by the author.

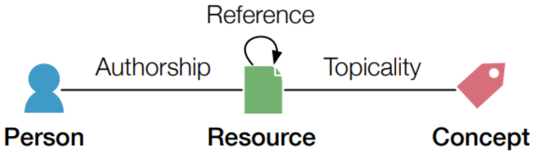


Figure 2.12: The structure of data represented in the PivotPaths application [Dör+12].

While all these examples are purely through the lens of viewing citation graphs, other visualisations can be used as inspiration. An example is the visualisation depicted in Peeters et al. [Pee+24], where they propose a new visualisation method for an overview of microbiome compositions in collected samples. While this is not directly useful for citation graphs, they give a nice visualisation of nodes in a snowflake structure, as seen in Figure 2.14. Instead of representing these microbiome compositions, clusters could serve as topics with surrounding

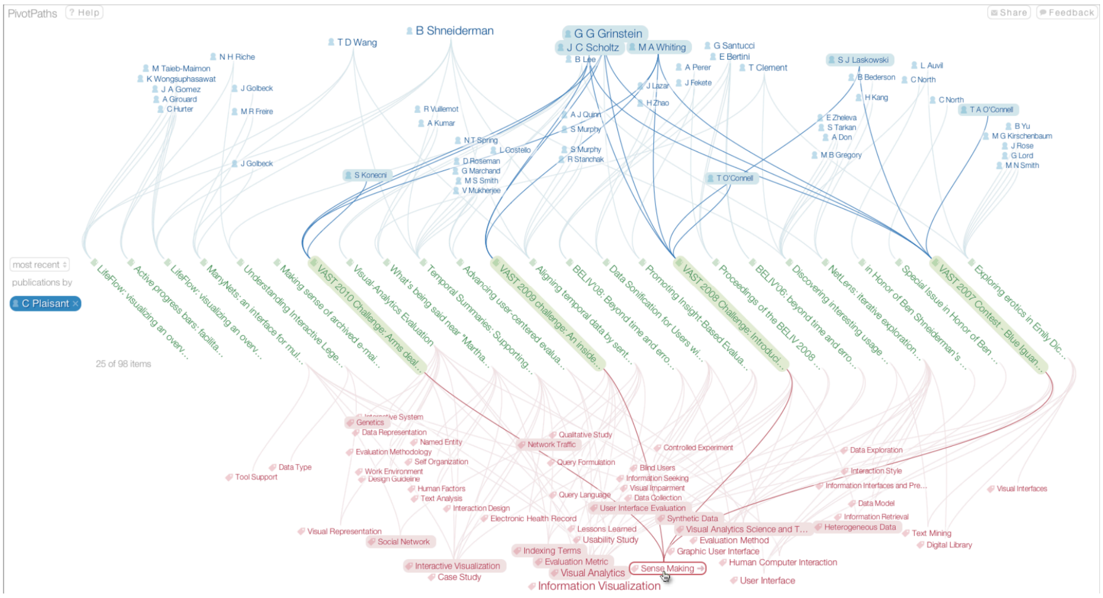

PivotPaths
? Help
Share
Feedback
B Shneiderman
T D Wang
G G Grinstein
J C Scholtz
M A Whiting
G Santucci
E Bertini
T Clement
S J Laskowski
L Auvil
C North
T A O'Connell
S Konecni
T O'Connell
most recent
publications by
C Plaisant
25 of 98 items
LifeFlow: visualizing an overv...
Active progress bars: facilita...
ManyNets: an interface for mul...
Understanding Interactive Lege...
Making sense of archived e-mai...
VAST 2010 Challenge: Arms deal...
Visual Analytics Evaluation
What's being said near "Martha...
Temporal Summaries: Supporting...
Advancing user-centered evalua...
VAST 2009 challenge: An inside...
Aligning temporal data by sent...
BELIV'08: Beyond time and erro...
Data Sonification for Users wi...
Promoting Insight-Based Evalua...
VAST 2008 Challenge: Introduci...
Proceedings of the BELIV 2008 ...
BELIV'06: beyond time and erro...
Discovering interesting usage ...
NetLens: iterative exploration...
In Honor of Ben Shneiderman's ...
Special Issue in Honor of Ben ...
VAST 2007 Contest - Blue Iguan...
Exploring erotics in Emily Dic...
Genetics
Data Representation
Network Traffic
Qualitative Study
Controlled Experiment
Data Exploration
Interaction Style
Information Interfaces and Pre...
Visual Interfaces
Data Type
Tool Support
Query Formulation
Blind Users
Information Seeking
Visual Impairment
Data Collection
User Interface Evaluation
Synthetic Data
Data Model
Information Retrieval
Heterogeneous Data
Visual Analytics Science and T...
Text Mining
Digital Library
Visual Representation
Social Network
Electronic Health Record
Lessons Learned
Usability Study
Evaluation Method
Graphic User Interface
Indexing Terms
Evaluation Metric
Interactive Visualization
Case Study
Visual Analytics
Sense Making
Human Computer Interaction
User Interface
Information Visualization

Figure 2.13: An example of the PivotPaths application with a filter applied, revealing collaborations, contributions, and topics related to the author C. Plaisant. Hovering over elements applies a linking and brushing technique, enhancing the clarity of visual connections [Dör+12].

papers and thus be modified to the current use case.

A large part of the visualisations that we discussed are valid for an academic context, but for this thesis, we choose to utilise a timeline graph, which will indicate when publications are released along with their citation count. With this visualisation, the user can see important papers over time. To identify important topics and growing research trends, we utilise a combination of clustered graphs and grouping algorithms to assist the user in researching relevant publications suited towards the user. We further explain our implementation in Chapter 3.

2.4 Design Principles

While there are different visualisations to represent the citation graphs and information about the publications, some design principles are recommended for creating these visualisations. During *Information Visualisation*, it was taught that these principles help make a clear visualisation that avoids confusion. However, in this thesis, only some principles are helpful for these visualisations. In this section, we discuss these different principles.

2.4.1 Gestalt laws

As described in [War21], the Gestalt laws were founded by a group of German psychologists. These laws describe how humans perceive patterns and organise visual elements into meaningful wholes. When applied to information visualisation, the Gestalt laws form a set of design principles that guide how users interpret visual patterns in data. There are eight Gestalt principles commonly cited in this context. Still, in the following sections, we focus on proximity, similarity, connectedness and continuity, as they are most relevant to the thesis.

Proximity

The principle of proximity indicates the importance of spacing between elements. This principle states that if data points are close together, they form a group. This principle can be seen by the

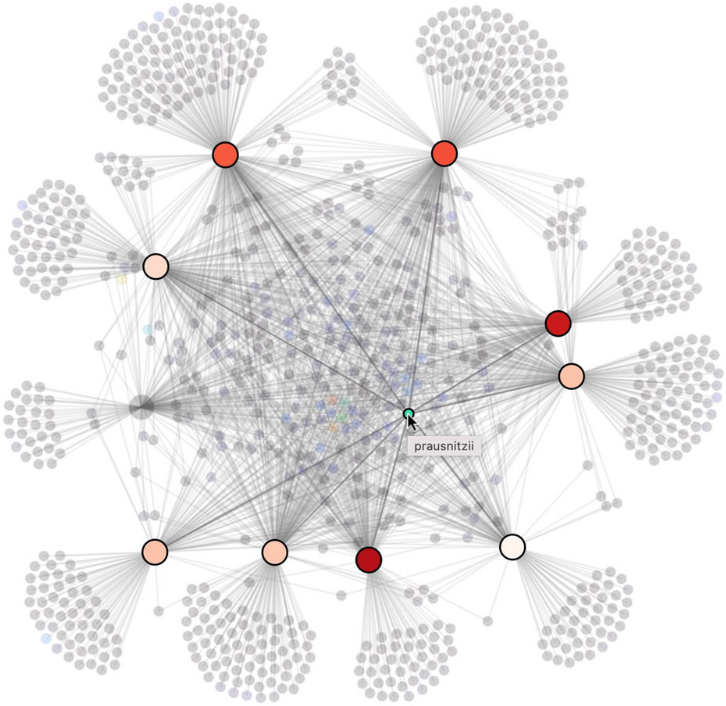


Figure 2.14: An example of the snowflake representation. (taken and modified from Peeters et al. [Pee+24])

proximity of the elements in Figure 2.15. Within the context of the citation graphs, elements that are close together are grouped into the same cluster, indicating that they are related.

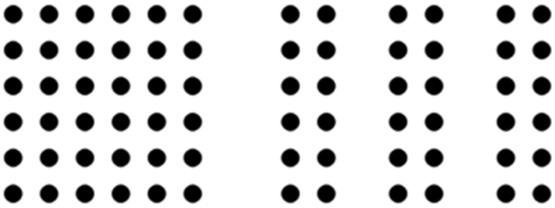


Figure 2.15: An example of proximity in visualisation. Data positioned close together are perceived as a group, whereas data separated by a significant distance are perceived as members of another group [YS19].

Similarity

The similarity principle indicates that objects are grouped together based on their shapes. Other attributes like shape and colour can also be indicators of similarity, indicating a grouping. For clustering graphs, this can be seen quite often, with different clusters having different colours to create distinct groups. An example of similarity by shape or by colour can be seen in Figure 2.16.

Connectedness

Connectedness indicates the relationship between concepts and is essential for node-link diagrams. Since citation graphs are node-link diagrams, the importance of connectedness through lines is not to be underestimated. According to [War21], connecting pieces can be a more powerful grouping method than having close proximity or assigning shapes and colours. This principle can be seen in Figure 2.17.

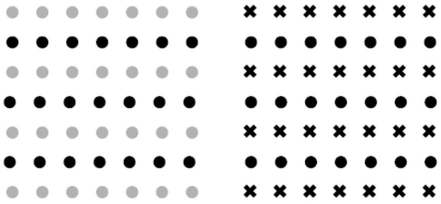


Figure 2.16: An example demonstrating the importance of shape and colour in figures, acknowledging the importance of the principle of similarity [War21].

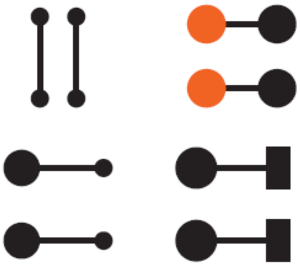


Figure 2.17: An example demonstrating connectedness: Even when elements differ in shape, size, or colour, their connections create a strong perception of unity, making the connected elements appear closely related despite their differences. (Figure taken and modified from [War21])

Continuity

Continuity is about the smoothness of lines and how they are visually interpreted. Lines that have abrupt changes in direction and are not smoothly connected are harder to interpret. By drawing smooth lines, it is easier to determine the source and destination of lines, giving a better user experience. This concept is essential for graphs since the edges are lines that can be drawn smoothly or abruptly. The difference between the fluent and abrupt lines can be seen in Figure 2.18, where the left example has smooth lines, and the right example has abrupt lines. Continuity was also an issue in [NIS15], where the straight lines in a previous implementation caused confusion. They then solved this by implementing smoother curves by using spline curves.

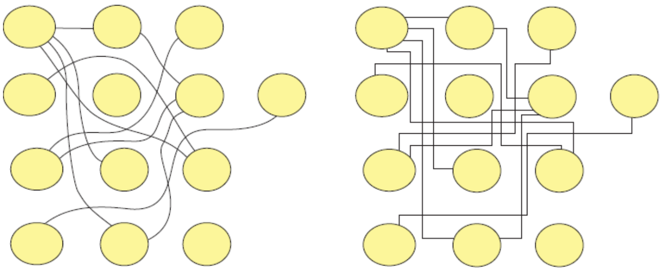
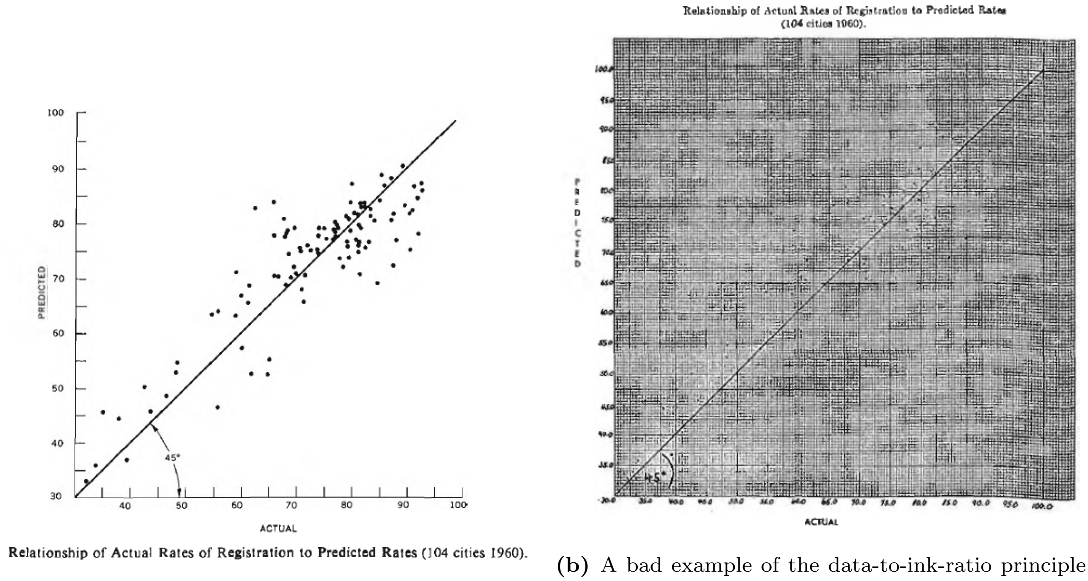


Figure 2.18: An example of using smooth versus straight lines in a graph. The straight lines are hard to follow, while the smooth lines give a clear start and endpoint. (Figure taken and modified from [War21])

While the Gestalt laws provide a solid foundation for designing visualisations, Tufte's principles offer another valuable perspective. Focused on clarity and simplicity, Tufte's approach emphasises minimising unnecessary elements and maximising the effective presentation of data. Visualisations should be able to show data in a transparent and efficient way so that the viewer can explore relationships and patterns without any distractions inserted in the graphs.

2.4.2 Tufte's Principles

Tufte's principles give another perspective on the world of data visualisation through the emphasis on clarity and efficiency. According to Tufte, *Graphical excellence* is achieved when statistical graphics consisting of complex ideas are communicated with clarity, precision and efficiency. An important principle that is required to follow this excellence is to have the maximum data-to-ink ratio. The data-to-ink ratio is relevant so that the design of the visualisation does not overshadow the data that is represented. Unnecessary grid lines, borders or other embellishments should be avoided to make room for the data. Only use as much graphical ink as necessary to represent relationships or events in data. Figure 2.19 provides an example with two graphs representing the same data: one adheres to the data-ink ratio principle, which is shown in Figure 2.19a. The other graph depicted in Figure 2.19b does not adhere to the principle because the amount of grid lines obscures the actual data that needs to be observed [TG83].



(a) A good example of the data-to-ink-ratio principle. The data is visible and can be interpreted by the user.

(b) A bad example of the data-to-ink-ratio principle. The grid lines overpower the data in the visualisation, making it hard for the viewer to interpret the data.

Figure 2.19: Examples of the data-ink ratio principle [TG83].

A similar principle is to avoid chart junk. This principle states that information that does not add to the visualisation should be removed, such as decorative elements. The visualisation's main priority should be the chart's functionality and usefulness, and not the aesthetics. With these principles in mind, together with the Gestalt principles, these frameworks help create visualisations that are both intuitive and informative for the end-user. Keeping these principles in mind, we utilise them in our frontend to give users a smoother experience navigating our visualisations, which is further explained in Section 3.2.1.

Chapter 3

Implementation

This section outlines the technical implementation of the thesis, which serves as a practical example and contributes to answering the research question. The main goal is to present a straightforward visualisation that represents the relationships between documents, which allows users to explore the data in an efficient manner. Depending on the user's end goals, the user should be able to go through large datasets, identify related content, and uncover relations between documents that were not visible before. Having this visualisation should help in information retrieval, topic exploration and knowledge discovery.

Within this thesis, the application of the LLMs is kept to an assisting role, giving the main priority to the visualisations. The visualisation consists of two representations: a timeline and a citation graph based on clustering topics. Through the timeline, researchers can review the history of an analysed publication. Aside from the timeline, the clustered citation graph displays the analysed paper and its influenced topics.

3.1 Large Language Models

Implementing LLMs requires careful planning, particularly as they play a supporting role in the context of the research question. As discussed in Section 2.2, LLMs offer several strengths that enhance, rather than dominate, the visualisation. For this thesis, three tasks have been selected to assist users while maintaining a non-disruptive integration within the visualisation: text labelling, summarisation, and identifying relationships between categories for recommendations. The application of these tasks will be discussed in the following sections.

3.1.1 Text Labelling

For the citation graph, we need to cluster the various publications and group them into the relevant category to which they belong. Within this thesis, we chose the approach of grouping with agglomerative clustering. As mentioned in Section 2.3.2, there are multiple other approaches like LPA and LDA, but we want the LLM to classify a publication under one specific topic. Agglomerative clustering also allows for clustering without a predetermined number of clusters. This characteristic of agglomerative clustering is essential since we are working with variable data, and the number of clusters can change based on the number of processed papers. Other clustering categories, such as k-means or topic modelling, need a pre-determined number of clusters to execute correctly, which makes them unsuitable for the current use case.

Another consideration to make is based on what we are grouping. Grouping based on the titles leaves us with little context about the subject and little data to process. Conversely, grouping based on the entire text can be a long process. Because of these reasons, we chose a clustering approach based on the publication's abstract. A publication's abstract is a summary of the publication, which makes the abstract a valid starting point for encapsulating the context of a publication, but also small in size to process. While the abstracts are a good start for grouping the publications per topic, some preprocessing is needed to optimise the results. Not all the text in the abstract is essential, bringing us to the principles of lemmatisation and removing stop words. Stop words are common words like "the" and "and" which do not add information to the context and need to be removed so that they do not influence the results of the grouping [LRU14]. On the other hand, Lemmatisation does not remove words but is the process of grouping word forms. For example, the English words "studying", "studies", and "study" are assigned to the lemma "study". So, multiple words with the same meaning get grouped under that word's lemma [GS12]. These techniques are utilised so that there is an emphasis on the essential words and filtering on less significant details. If preprocessing is not done in this way, we might get groups with common words, such as stop words or words belonging to the same group.

After preprocessing, the grouping and text labelling process can begin. As seen in Section 2.2.1, we can utilise the strengths of Transformers and LLMs to help with the text labelling. To do this, we produce embeddings using a Sentence Transformer [RG19]. This type of Transformer can process the semantic meaning of sentences within the embeddings, creating sentence embeddings. Sentence embeddings are similar to the word embeddings seen in Section 2.1.1, but word embeddings can lack the full context of a sentence. To clarify, we look at the example in table 3.1, which shows two different meanings of the word "bank". With word embeddings, there is no difference between the embeddings for the word bank, while there is a semantic differentiation. With sentence embeddings, however, the whole context is considered, creating different embeddings for the word bank, which results in properly handling the context.

Example 1: I deposited money at the bank.
Example 2: I had a picnic by the bank.

Table 3.1: An example of how the word "bank" may influence the context based on the embedding technique used. Using word embeddings, "bank" is in its own context and has the same word embedding in both sentences. This outcome results in a loss of context, with both "banks" being interpreted as the same "bank". With sentence embeddings, the word bank is tied to the context of the sentence, resulting in different embeddings and preserving the context of the word within the sentence.

The sentence embeddings can then be used to calculate the similarity between sentences through distance metrics. Based on these distance metrics, clustering can occur to identify which sentences are close together. In this case, the distance represents the semantic similarity between the sentences used in the abstracts. With these embeddings, agglomerative clustering can begin forming the groups. The algorithm starts assigning each data point to a cluster. Each iteration merges clusters if the distance between the two clusters is less than the predetermined distance threshold. The algorithm stops if no cluster combinations with a distance exceeding the threshold exist. After the clustering, KeyBert gets the keywords per cluster to better represent these clusters on the frontend. KeyBert is another Transformer that uses the abstracts per cluster to generate a predetermined number of keywords. KeyBert first calculates the embeddings of the passage. After this, individual word embeddings are generated in n-grams. Then, these two embeddings are compared against each other to determine which terms are relevant to the text. This way, the most relevant keywords can be filtered for the specific passage. After these keywords are specified, an Ollama model is prompted to summarise the keywords into a fitting topic. These topics are then used to label the clusters and categorise the legend in the frontend. When a new graph is made with other fetched papers, this process can start again. It will group the publications in another set of clusters, independent of the number of incoming publications.

3.1.2 Summarisation and Topic Recommendation

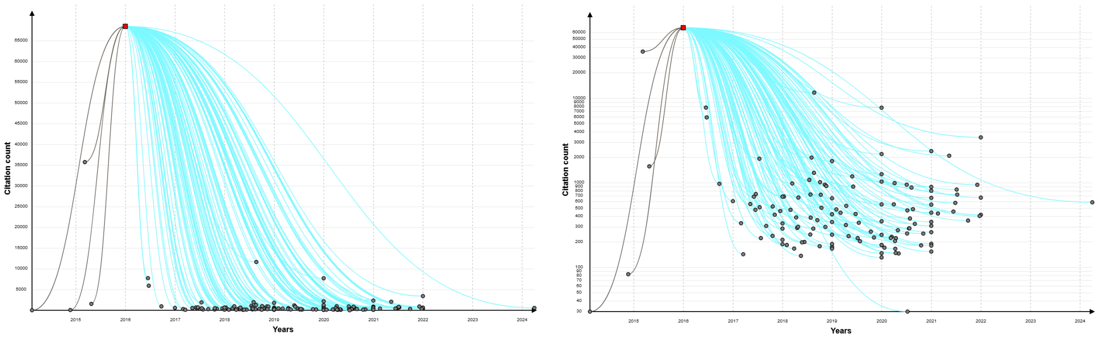
Both summarisation and topic recommendation are done in a similar way. In the application's backend, two endpoints handle summarisation and recommendation. The goal of the summarisation is to take the text of a paper and return it in a summarised manner to the user on the frontend. The recommendation is meant for the authors who have already released publications surrounding the analysed paper, but may want to research another topic they have not interacted with. To reach both goals, we use the Ollama LLM combined with the LangChain Python library. LangChain is a library that assists in developing LLM applications and provides functions to easily set up an LLM environment with additional support for creating chatbots, Retrieval Augmented Generation setups (RAG) and other utilities. We want to ensure we forge the right prompts for each endpoint in this situation. For the summarisation, we provide a prompt that tells the LLM to summarise the whole text of a publication, and we give the text along with the section titles of the paper. With this setup, the LLM returns the section header and a small summary of the section. We can then attach a formatter that reshapes the output of the LLM into the desired format for the frontend. In LangChain, forming a prompt, getting the output of the LLM and attaching a formatter is called a "chain". For the recommendation of the topics to the authors, we form another different chain. As input, the LLM will receive a set of topics the author has already published on and a set of topics that the author is unfamiliar with. The prompt indicates that the LLM must try to return the top three most fitting topics of the second set as a recommendation for the author. The formatter in this chain is also different, as we want to receive JSON format to process this information on the frontend properly. Both of these endpoints are designed to be in a supporting sense of the frontend, which we will discuss next.

3.2 Visualisation Tool

The visualisation tool consists of two parts: first, we have the backend, which processes the publications for use in the frontend. This consists of analysing a selected paper, downloading the references and citations of the analysed paper in PDF format, using the PDFs for in-document analysis and fetching information from API endpoints about the fetched documents. The analysis is the primary source of information presented on the frontend. The backend has extra functionalities that support the frontend using the LLMs and provide the functionalities mentioned in Section 3.1. Second, we have the frontend, which represents these processed papers in a timeline and clustered visualisation. There is also the possibility of interacting with the frontend by clicking data points and analysing the contents of publications, such as the citation count, references, and other statistics.

3.2.1 Frontend

For the frontend, we made two visualisations that represent ways to visualise a citation network. First, there is the timeline that represents the temporal information of the citation network. To achieve this, we leverage the built-in capabilities of D3.js and opt for a hierarchical clustering tree. This approach enables a parent-child structure, where the root node represents the parent, and the references and citations are depicted as the children. Since a paper's references are always older than the paper itself, and citations are always more recent, the relationships between publications are represented using distinct colours: cyan for citations and a brownish tone for references. While the x-axis represents the release years of the publications, the y-axis represents the citation amount per publication. The user can also switch between a logarithmic scale and a linear scale for the y-axis. This option is helpful when there is a big difference between citation counts, creating the problem of a considerable distance between scales. This difference is evident in the example screenshots shown in Figure 3.1a and Figure 3.1b. In the linear scale, a significant clustering of data points occurs near the bottom of the graph due to the wide disparity in citation counts. Conversely, the logarithmic scale provides a more evenly distributed data visualisation. Within the timeline, data points represent the references and citations of a root paper, with the root paper being the analysed paper. The data points can be interacted with, providing information about the publications like the DOI, the title, authors, the abstract and the occurrences of where the root node was mentioned, which shows in a window which can be seen in Figure 3.2a. For each data point, there is also the option to fetch new publications. The fetching of the paper happens in the same way that the original data is provided, and an endpoint is used to fetch the papers using the submitted DOI ID. This results in the current node becoming the root paper, which generates a new timeline for that root paper. There is also the possibility of going back to the previous timeline by pressing the back button when selecting a data point. The different timelines are kept in memory on a stack data structure for navigation, which means that the user saves time by not needing to redo the fetching process of the original timeline.



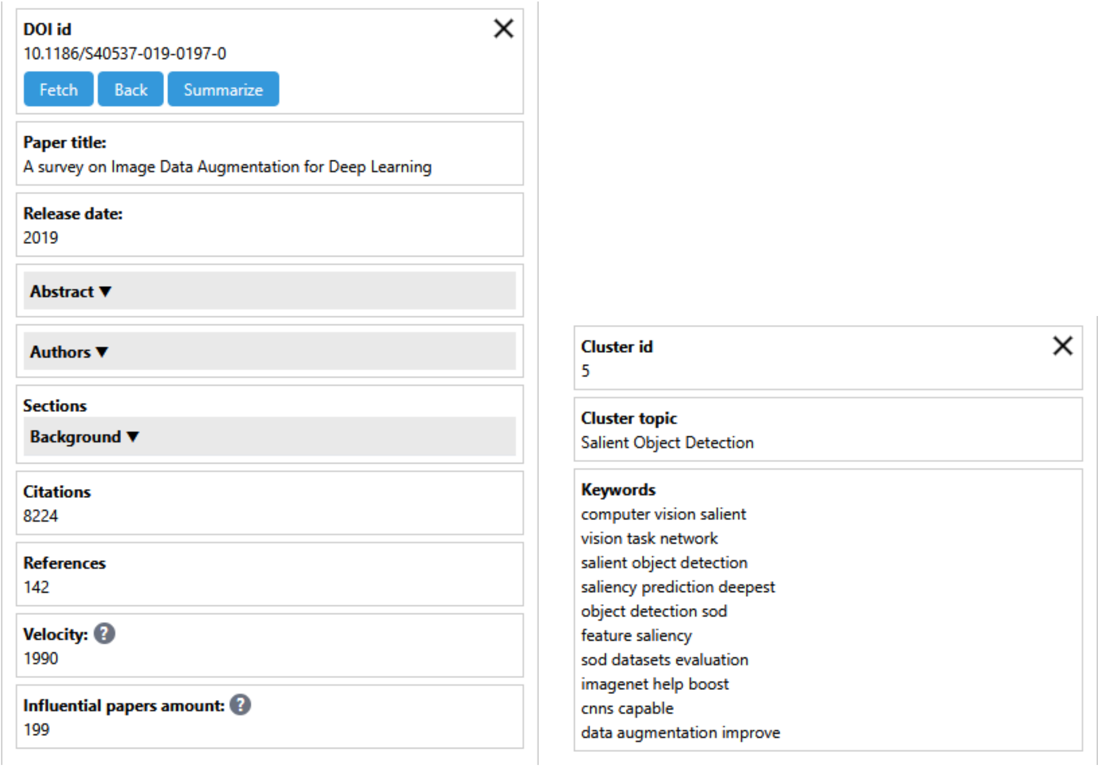
(a) The timeline visualisation with a linear scale for the citation count in the y-axis.

(b) The timeline visualisation with a logarithmic scale for the citation count on the y-axis.

Figure 3.1: The difference between a logarithmic and linear representation of the timeline graph. In Figure 3.1a, we can see that most of the data is on the bottom of the y-axis because most papers are outperformed by a major publication cited in great quantities. In Figure 3.1b, we can see that the logarithmic scale helps with visualising large disparities between data points, making for a more balanced visualisation.

The other representation option is the citation graph, which can be seen in Figure 3.3. Switching between the timeline and the citation graph is possible using the navigation bar at the top left of the page. When opening the citation graph, the user can see that the graph consists of nodes with different colours. Each colour represents a topic, which can be seen by consulting the legend. There are three types of nodes: the root, cluster, and paper nodes. The root node is indicated with a red colour in the middle, and the nodes with a thick black border are the paper nodes. Similar to the timeline graph, the paper nodes have links to the cluster nodes depending on whether they are a reference or a citation of the analysed paper. The remaining nodes are the cluster nodes, which have the paper nodes as children if they fall under the topic of that cluster. Each cluster node has a different size, representing how many paper nodes are under a specific topic. This visualisation indicates which topics have been influenced the most by the analysed paper, since more papers have been released than on other topics. The individual size of the paper nodes visualises how many citations the paper has. When clicking the cluster nodes, they give the general topic of the cluster and the keywords that the cluster represents, as seen in Figure 3.2b. The same functionality comes back for the paper nodes as with the timeline, so there is the option to fetch new papers and return to the original representation. An additional feature enables the summarisation of an entire paper. The backend processes the paper section by section, using the LLM to generate concise summaries for each part.

Regarding accessibility, it may be hard for colour-blind users to see the difference between the multiple topics. To combat this issue, the user is given three different palettes, which can be adjusted to account for colour-blindness. The supported types of colour blindness are protanomaly, a reduced sensitivity to red light; deuteranomaly, a red-green colour vision deficiency; and tritanomaly, a blue-yellow colour vision deficiency. An example can be seen in Figure 3.4a, showing the colours that suit people with protanomaly. Figure 3.4b shows how



DOI id
10.1186/S40537-019-0197-0

Paper title:
A survey on Image Data Augmentation for Deep Learning

Release date:
2019

Abstract ▼

Authors ▼

Sections
Background ▼

Citations
8224

References
142

Velocity: ?
1990

Influential papers amount: ?
199

Cluster id
5

Cluster topic
Salient Object Detection

Keywords
computer vision salient
vision task network
salient object detection
saliency prediction deepest
object detection sod
feature saliency
sod datasets evaluation
imagenet help boost
cnns capable
data augmentation improve

(a) The window shown when a publication is interacted with in the cluster or timeline graph. The window displays information such as the DOI ID, the release date and other extra information about the publication. From this window, the user can choose to fetch the publication, go back to the previous version of a visualisation or summarise the paper. When the "Summarise" button is clicked, a new field appears with the summary of the publication. The sections field contains information about the analysed paper, like where it was cited or where this publication was referenced in the analysed publication.

(b) The window shown when a cluster data point is interacted with. The window displays information about the overall cluster topic and the keywords that are given to that cluster by the LLM. The ten keywords were used by the LLM to determine the final "cluster topic".

Figure 3.2: The different tooltips of the application. The tooltip shown in Figure 3.2a is available in both visualisations, while the tooltip in Figure 3.2b is only available in the cluster graph, since the timeline visualisation does not have any clusters.

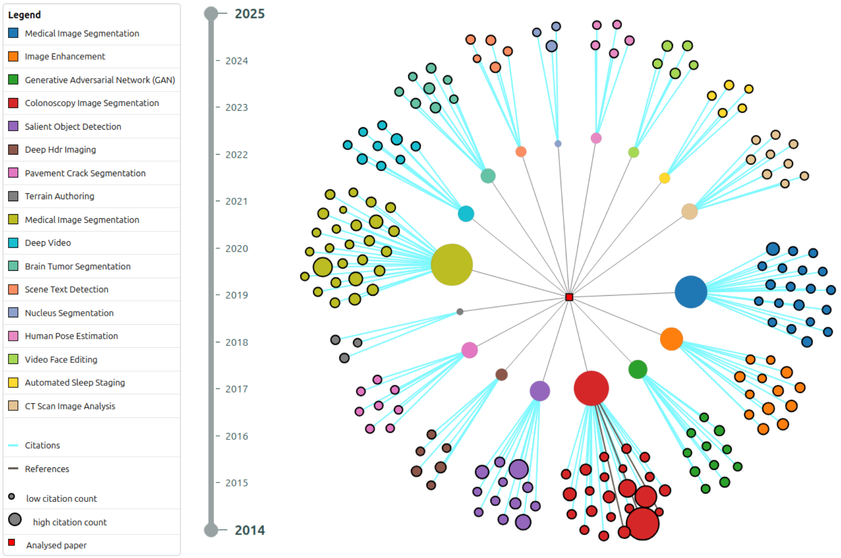


Figure 3.3: Cluster graph visualisation of citations and references for the analysed U-Net paper. The analysed paper (red square) is linked to its references (brown edges) and citations (cyan edges). The citations and references are categorised under various research domains. "Colonoscopy Image Segmentation" is particularly important for the references, highlighting the topic's influence and connections with the analysed paper.

someone with protanomaly would see the graph.

To further improve the general usability of the graph visualisation, there is a way to single out different categories through linking and brushing. When hovering over a group of nodes, the rest of the graph dims out and highlights the group of nodes. In the legend, the corresponding group is also highlighted. This mechanism also works when hovering over an element within the legend, highlighting the group linked to that legend element. This principle can be seen in Figure 3.5, which displays both the usage of linking and brushing through the legend of the graph (Figure 3.5a), as well as on the nodes in the graph (Figure 3.5b).

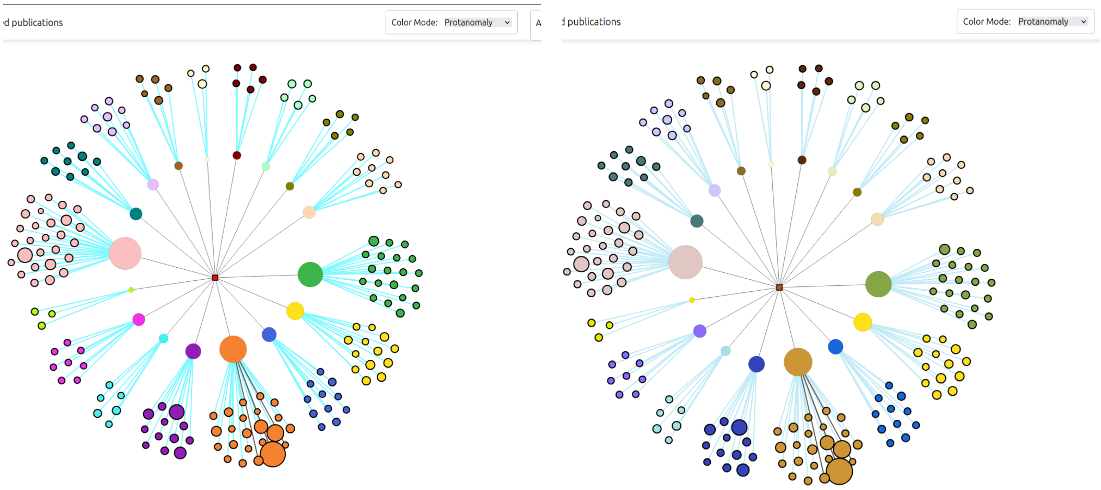
Both the visualisations follow the four Gestalt laws and Tufte's principles mentioned in Section 2.4. Connectedness is a general principle for both the timeline and the cluster graphs. Connected nodes mean relationships, a reference or citation, between those nodes. This also counts for continuity since the links in both graphs do not make any abrupt changes, making it easy for the user to follow the links from source to destination.

For the timeline graph, the citation count defines the proximity, meaning that data points with a similar citation count are grouped. While it could be argued that the dotted lines in the timeline graph are cluttered because the visualisation violates the data-to-ink principle, we claim it helps guide the user to read the chart more easily.

In the cluster graph, however, proximity is determined by the topic defined for the publication, indicating that the proximity is related to the topic. The similarity law can be seen in the cluster graph when looking at the borders of the nodes. Nodes with a black border are publications, while clusters are nodes without a black border.

3.2.2 Backend

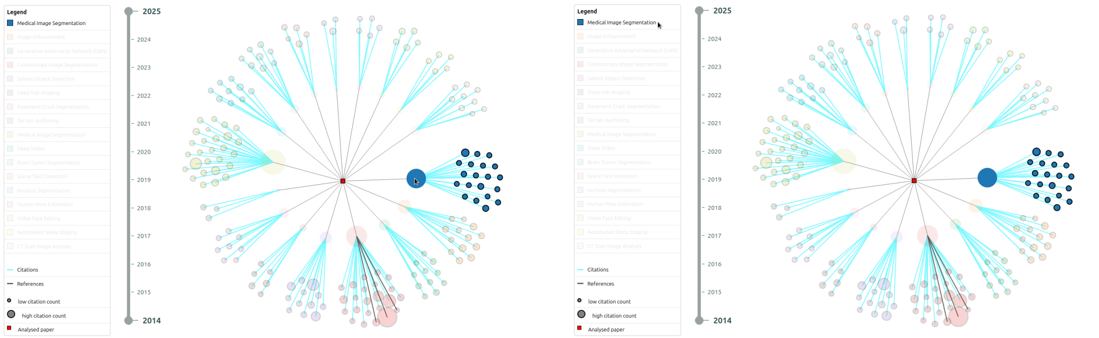
Within the backend, the main goal is to process and analyse a selected paper. This starts with the DOI of the selected paper that the user wants to see analysed. Using the DOI of the publication, the paper is downloaded via an external source to get the PDF. In some occurrences, the PDF might be corrupted or not properly downloaded, which means that it



(a) Graph visualisation with a colour scheme tailored for users with protanomaly, ensuring accessible and clear differentiation of clusters and relationships.

(b) The graph visualisation viewed through a lens as someone with protanomaly (using Coblis colour blindness simulator).

Figure 3.4: The colour mode implementation within the application. Figure 3.4a shows how the application looks when a user without protanomaly would view the application. Figure 3.4b shows how the user would see the visualisation with protanomaly.



(a) The linking and brushing technique. Hovering over a cluster or data point within the citation graph highlights the related element in the legend and dims other elements unrelated to the element that is being hovered over.

(b) The linking and brushing technique. Hovering over an element within the legend highlights the related cluster in the citation graph and dims other elements unrelated to the element that is being hovered over.

Figure 3.5: The linking and brushing implementation on the cluster graph nodes. Both figures show the technique by hovering over the legend and over the nodes. With this technique, the user can easily distinguish related elements by hovering over the desired element.

can not be analysed and could cause the application to crash. To prevent this, we perform a backup action that looks up the same DOI to the Semantic Scholar API [Kin+23]. This lookup results in the ArXiv ID of the publication, allowing for the download of the PDF with the ArXiv API. When the PDF is downloaded, it gets processed by GROBID. GROBID is a machine learning library for extracting and parsing raw documents like PDFs to structured XML. With the publication processed by GROBID, the XML can be analysed instead of the raw PDF [GRO08]. The resulting XML file contains every section separated, such as the headers, which contain the title and the authors, the text itself, which is separated per section, and the reference list of the publication. To handle this data, the Element Tree library in Python is utilised to process the XML. This way, the reference list can be cycled through in a tree structure, parsing the different references and analysing the paper where this reference is used. This analysis is needed so that, in the frontend, the user can see where this reference is used in the analysed paper and in what context it was provided.

For the citations, the DOI of the analysed publication is queried in the DBLP SparQL API. This query returns the DOIs of the citing papers, which can be used to download the PDFs of the citing papers. When these PDFs are downloaded, they undergo a similar analysis using GROBID. With the resulting XML from GROBID, we process the citation. First, the analysed paper is sought after in the reference list of the citation. When the analysed paper is found, the citation number is parsed from the results. After this, the whole paper is parsed to find this citation number and extract the passages using the citation number. This information is used to show where the analysed paper was cited in the citing paper. This information makes it easier for the user to determine in what context the publication was cited, resulting in a more transparent overview of the utilisation of the analysed paper in the cited paper. Once the analysis of the PDFs is done, the backend fetches more information about the references and citing papers by using the DOI to query a legacy Semantic Scholar endpoint. This endpoint provides the release date of a publication, the number of references and citations, the number of influential citations, and the citation velocity. Semantic Scholar explains that influential citations are highlighted citations where one publication significantly influences another, making it easier to see how research builds on and connects with previous work. This parameter is decided by a machine learning model, considering the number of citations and the surrounding context of the publication [VHE15]. The velocity indicates how popular or impactful the publication is after several years. This parameter is determined by taking the weighted average number of citations over the last three years, with recent papers getting less weight. After the processing is done, the information gets stored in a JSON file. The frontend processes this file, and D3.js uses its data to build a hierarchical structure based on the parent-child relationships defined in the JSON file. This entire pipeline can be seen in Figure 3.6. For the references of the analysed paper, a similar approach is utilised to get the results. The difference within the process is that the reference must be sought after in the original paper. As a result, we show the user where the referenced paper is cited in the publication.

3.3 Use cases

To demonstrate the practical application of the implemented system, there are different use cases where researchers utilise the tool to achieve their goal. Depending on the situation, the different graphs can be used to identify patterns. The timeline can be used to identify the importance of a selected paper on future trends, depending on the number of preceding publications. In this section, we will discuss several of these use cases, highlighting the contributions of this thesis.

Identifying emerging research trends

For this first use case, the researcher selects a publication they want to analyse. A timeline graph is generated by plugging the publication into the tool, revealing the citations and references related to the publication. This timeline is insightful, as it can indicate in what year the analysed

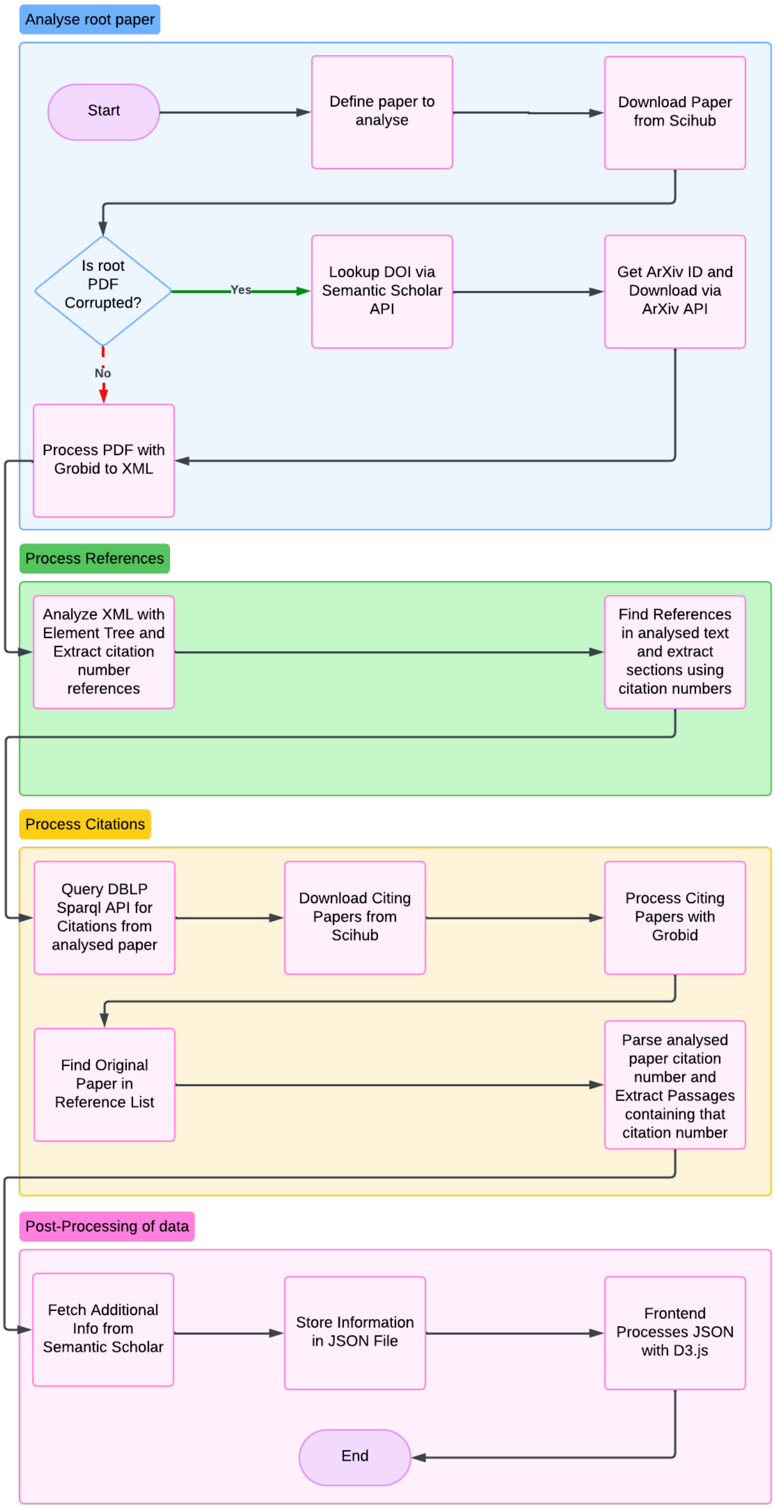
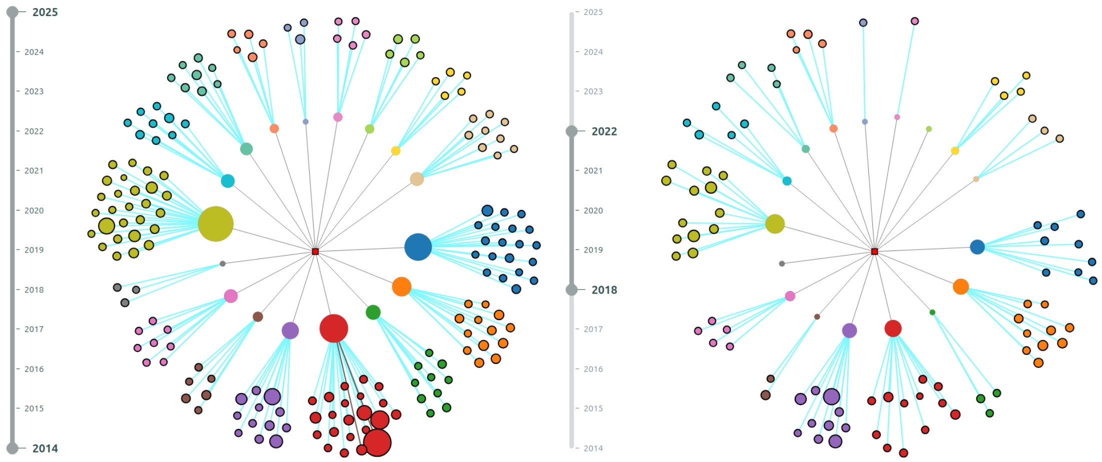

Analyse root paper
Start
Define paper to analyse
Download Paper from Scihub
Is root PDF Corrupted?
Yes
Lookup DOI via Semantic Scholar API
Get ArXiv ID and Download via ArXiv API
No
Process PDF with Grobid to XML
Process References
Analyze XML with Element Tree and Extract citation number references
Find References in analysed text and extract sections using citation numbers
Process Citations
Query DBLP Sparql API for Citations from analysed paper
Download Citing Papers from Scihub
Process Citing Papers with Grobid
Find Original Paper in Reference List
Parse analysed paper citation number and Extract Passages containing that citation number
Post-Processing of data
Fetch Additional Info from Semantic Scholar
Store Information in JSON File
Frontend Processes JSON with D3.js
End

Figure 3.6: The UML flowchart describes the process of the backend, going over the various different processes that happen in the backend, like analysis of the root paper, the analysis of citations and references and post-processing.

publication is cited the most, potentially revealing that the publication has contributed to a trend within a specific year. The researcher switches to the clustered citation graph to identify the influence on different topics. By glancing over the graph's legend, the researcher can see the topics where the publication has been cited. Depending on the size of the nodes, the researcher can determine if the publication had a significant influence on the specific topic, and they may decide to investigate the trend further by generating a new graph using a different node with a specified topic. Additionally, the graph contains a slider that can be used to display papers within various time ranges. These ranges modify the cluster graph, only revealing publications published within the time frame indicated on the slider. Consequently, the cluster nodes' size is also modified, signifying how big the topic is during a specific period. This example can be seen in Figure 3.7, where the usage of the slider and the effects can be seen. Besides these graphs, the nodes contain information about the citation velocity and other influence indicators, assisting in the researcher's assessment of identifying emerging research trends.



(a) The visualisation before setting a date range results in the full visualisation without any missing data. The cluster nodes have their full size, and every node is visible.

(b) The visualisation after setting a date range results in a more limited visualisation where data that is not between the date range is filtered out. The cluster nodes change in size accordingly. Only publications between 2018 and 2022 are shown.

Figure 3.7: The working of the slider that controls the date range in the cluster visualisation. Using the slider modifies the visualisation, as seen in Figure 3.7b, and filters publications that fall outside of the range. Clusters that do not have children indicate that the topic does not have any publications in that year. In the visualisation, this can be seen in two different clusters, where it does not contain any papers between 2018 and 2022.

Discovering new topics

Relating to identifying the emerging research trends, the researcher can also look for new topics that have emerged. Since the clustering and determination of topics are done using an LLM using the abstracts of the publications, there are no predefined topics or categorisation. This way of working opens the door for new topics from niche papers. This makes the application robust against the changing environment of the academic world, where the tool can adapt to evolutions throughout time. The researcher can look at these niche topics and further investigate if the topic has been widely researched or if the topic is relatively new by generating a new graph about the new topic.

Literature review acceleration

Another use case that shows the potential helps accelerate the literature review before starting new research. A researcher begins a new paper but is not well-versed in the topic. The researcher might already have a publication related to the topic, but does not know where to head next. By plugging the publication into the tool, he can see in which year the topic was popular and other publications and topics where the publication was mentioned. This action gives a way to explore multiple alleys to different subtopics that can be present in the main topic of the researcher. If the researcher wants to limit themselves to a specific time range, they can use the timeline slider implemented on the cluster graph and restrict themselves to the time range. To further assist in the research, the tool provides the ability to get AI-generated summaries, condensing the paper content and leading to quicker comprehension and filtering of publications.

Identifying influential topics

While identifying emerging research trends is possible, the visualisation also enables categorising references within the field. By examining the link colours between nodes, we can distinguish references from citations, revealing the groups to which the references belong and their influence on the creation of the analysed papers. This, in turn, provides an overview of the domains that have shaped the analysed papers, leading to insights into the nature of the publication. An example of this is shown in Figure 3.3. The analysed paper focuses on U-Net, a convolutional neural network (CNN) designed for biomedical image segmentation. Within the application, the references are categorised under “Colonoscopy Image Segmentation,” indicating that the topic had a particular influence on the development of the U-net publication.

Author similar topic recommendation

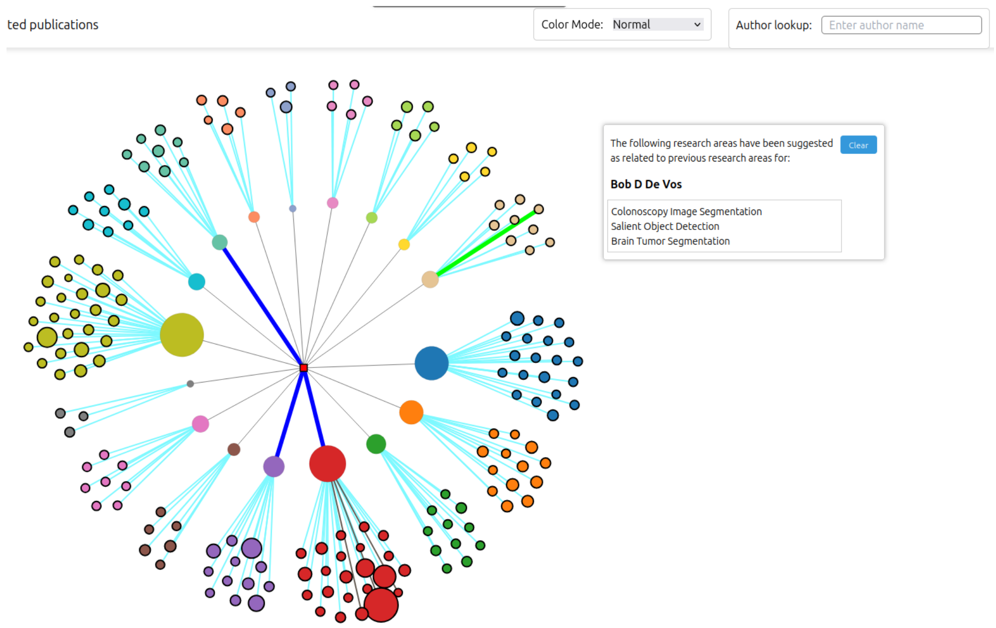
For the final use case, authors can use the application to determine the field they may want to research next. If the author is present within the visualisation, they can look up their name by inputting it in a text field in the top-right of the application. After selecting their name, the green links reveal their contribution to the analysed publication. In the meantime, the topics the author contributed to and the other topics are sent to an endpoint in the backend. This endpoint uses an LLM that receives the lists of topics and determines which topics are recommended. This recommendation is purely based on the titles of the topics. After processing, the visualisation displays the topics by highlighting the cluster links in blue. These blue-coloured links are represented as “recommended” research areas for the researcher to discover further. An example of this can be seen in Figure 3.8, which describes the recommended research areas for “Bob D De Vos”.

3.4 Comparing other applications

To further emphasise the importance of different elements provided in our application, we compared other applications and our application, which have pros and cons for the user experience. These points are primarily based on how the user could interpret information applications and the design choices. The comparison focuses mainly on the two aspects of the thesis: the visual representation of academic networks and how AIs and/or LLMs are utilised in the applications, if applicable. This section will first evaluate the visualisations and examine the usage of AI within individual applications. The visualisation analysis is based on the Gestalt laws and Tufte’s principles. The usage of AI is evaluated based on the user’s needs and capabilities. Finally, the section ends with an overall comparison between the applications.

3.4.1 CiteSpace

CiteSpace is an application based on the publication of Chen [Che04]. The visualisation plays a primary role in showcasing the relationships between a predetermined dataset of articles and discovering the “intellectual turning points” within scientific domains. The interface allows users to select the articles they want to analyse, allowing for data sources like Web of Science (WoS), Scopus or PubMed. Depending on the use case, the user can decide to process the



ted publications

Color Mode: Normal

Author lookup: Enter author name

The following research areas have been suggested as related to previous research areas for:

Bob D De Vos

Colonoscopy Image Segmentation
Salient Object Detection
Brain Tumor Segmentation

Figure 3.8: The recommendation system of publication topics in working. The visualisation shows the works of "Bob D De Vos" in green, which belong to the "Medical Image Segmentation cluster". The application suggests that the author can explore similar topics like "Colonoscopy Image Segmentation", which are indicated in blue.

data before generating the graph, like generating time slices. When the user has set their preferences for the data analysis, they can create a cluster graph that showcases the clusters based on their citations and connections to other clusters. An example[1] of the clustered graph can be seen in Figure 3.9, which shows the emerging technologies between 1990 and 2024. When the graph is generated, the user can modify the graph, such as the background colour, which colours the graph needs to use by adjusting the colour palette, which labels must be shown and other configurable options. The size of the nodes within the clusters represents the citation count of the publication, enabling the user to see which publications are more popular than others.

Besides the cluster graph, the application allows a swap to a landscape representation, which displays all clusters separately on their landscape. The y-axis per landscape represents the citations per topic over the years. This representation allows the user to identify when a particular topic has "peaked" throughout the years. Besides the data's cluster graph and landscape view, the user can also switch to a timeline, circular, and time zone view. These options allow users to adjust and find the visualisation that fits their needs.

While this application does not use AI to help investigate publication data, its primary purpose is to visualise the data, so it is essential to look at how the Gestalt Principles can apply to this visualisation. First, the principle of proximity does occur within the visualisation. The clusters indicate which node belongs to which group, supported by the colours given by the application. Second, the colour of the nodes also shows which node belongs to which group, which is based on the similarity principle. Third, the connections between nodes are clear, and a relationship is indicated when needed. Last, the continuity principle is still considered since the links all use bent lines, which helps the user track the source and destination.

The application is designed to visualise large networks. It automatically generates clusters

[1] https://citespace.podia.com/exemplars

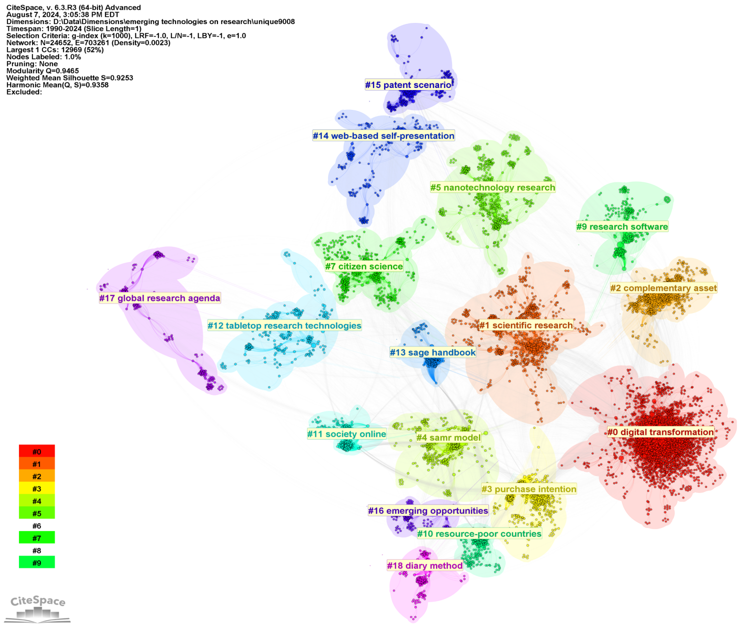

CiteSpace, v. 6.3.R3 (64-bit) Advanced
August 7, 2024, 3:05:38 PM EDT
Dimensions: D:\Data\Dimensions\emerging technologies on research\unique9008
Timespan: 1990-2024 (Slice Length=1)
Selection Criteria: g-index (k=1000), LRF=-1.0, L/N=-1, LBY=-1, e=1.0
Network: N=24652, E=703261 (Density=0.0023)
Largest 1 CCs: 12969 (52%)
Nodes Labeled: 1.0%
Pruning: None
Modularity Q=0.9465
Weighted Mean Silhouette S=0.9253
Harmonic Mean(Q, S)=0.9358
Excluded:
#15 patent scenario
#14 web-based self-presentation
#5 nanotechnology research
#9 research software
#7 citizen science
#2 complementary asset
#17 global research agenda
#12 tabletop research technologies
#1 scientific research
#13 sage handbook
#11 society online
#4 samr model
#0 digital transformation
#3 purchase intention
#16 emerging opportunities
#10 resource-poor countries
#18 diary method
#0
#1
#2
#3
#4
#5
#6
#7
#8
#9
CiteSpace

Figure 3.9: An example of a cluster graph in CiteSpace (Example taken from the CiteSpace website). The graph shows emerging technologies between 1990 and 2024. The graph shows that the clusters “digital transformation” and “scientific research” contain the largest number of articles, thus being the biggest emerging technologies. The light-grey lines between clusters indicate relations between the clusters. Along with the graph, the application also gives extra information in the top-left corner, such as the modularity score and the weighted mean silhouette score, indicating the clustering quality.

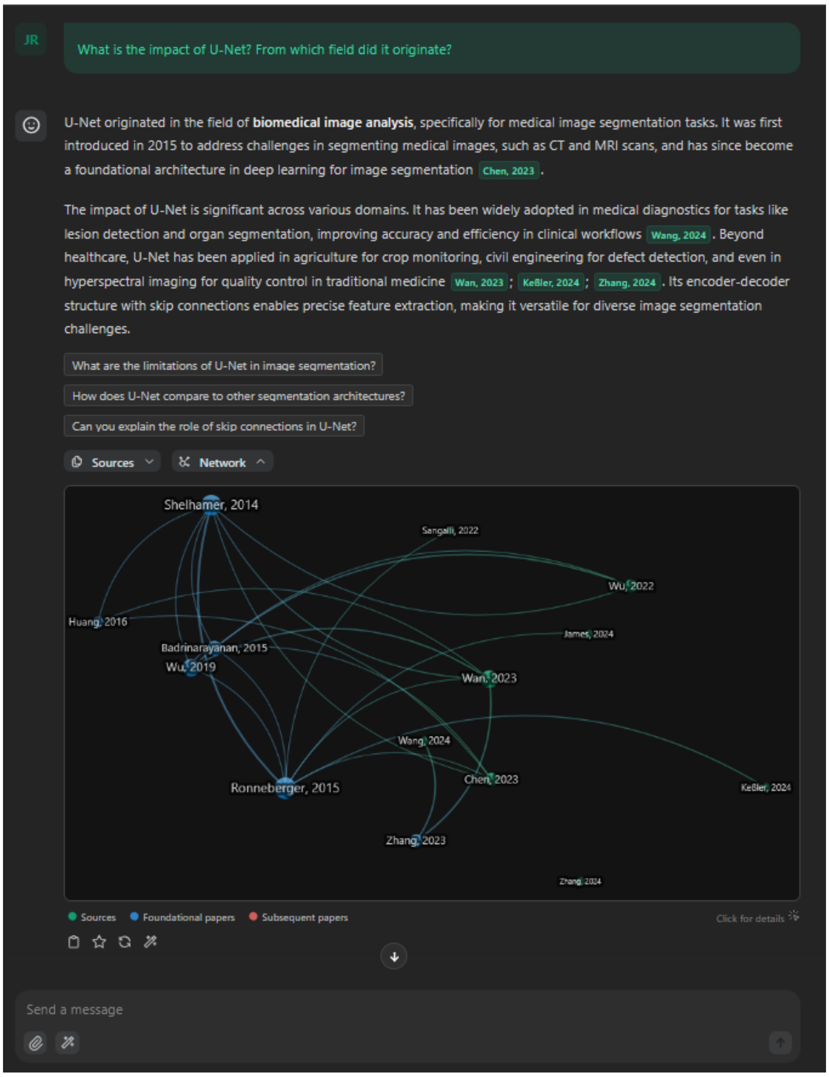
for the user, but the user must provide a file containing the publications they wish to visualise.

3.4.2 ScienceOS

ScienceOS is a web-based application with a chat interface that can answer questions while providing academic-based sources. When provided with an answer, the application provides the sources and potential follow-up questions the user may want. Another feature we are primarily interested in is visualising the used sources. This visualisation consists of a network of nodes divided into three groups: sources, foundational papers and subsequent papers. The sources are publications that are used within the chat assistant to ground the response of the LLM. The foundational and subsequent papers are publications related to the sources, further underscoring the grounding. The graph can be interacted with by clicking the nodes and providing details about the title, authors, TLDRs, and abstracts. It also provides a drill-down mechanism, where a new network can be generated, and it provides citations of those papers. Hovering over the nodes results in a similar linking and brushing technique, which can also be seen in our implementation, which helps follow the relationship between nodes. An example can be seen in Figure 3.10, where a prompt about U-net is provided to replicate results with our data.

This application’s primary use case is to answer users’ questions with a grounded response, while the visualisation plays more of a background role. This indicates that the application uses the LLM, which uses publications to ground its responses. The advantage for the user is that they can have answers without seeking specific publications to ground a given statement.

When looking at the visualisations, the Gestalt laws are still in effect. Regarding similarity, the colours relating to the groups indicate sources, foundational papers and subsequent publications.



JR

What is the impact of U-Net? From which field did it originate?

U-Net originated in the field of **biomedical image analysis**, specifically for medical image segmentation tasks. It was first introduced in 2015 to address challenges in segmenting medical images, such as CT and MRI scans, and has since become a foundational architecture in deep learning for image segmentation Chen, 2023 .

The impact of U-Net is significant across various domains. It has been widely adopted in medical diagnostics for tasks like lesion detection and organ segmentation, improving accuracy and efficiency in clinical workflows Wang, 2024 . Beyond healthcare, U-Net has been applied in agriculture for crop monitoring, civil engineering for defect detection, and even in hyperspectral imaging for quality control in traditional medicine Wan, 2023 ; Keßler, 2024 ; Zhang, 2024 . Its encoder-decoder structure with skip connections enables precise feature extraction, making it versatile for diverse image segmentation challenges.

What are the limitations of U-Net in image segmentation?

How does U-Net compare to other segmentation architectures?

Can you explain the role of skip connections in U-Net?

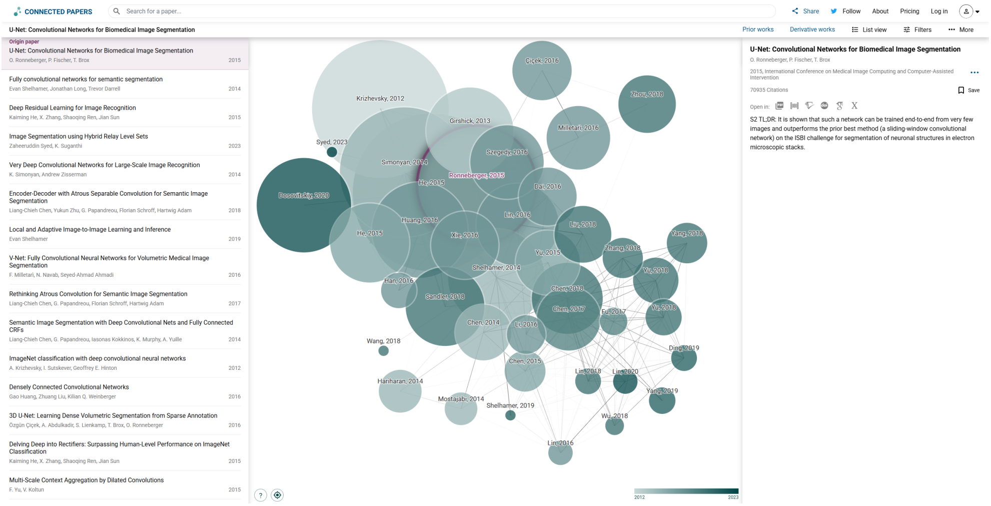
Figure 3.10: An overview of the ScienceOS chat. The prompt is about U-net, and the LLM gives us more information through textual generation backed by sources and a generated graph. The generated graph represents the relations between the sources. These consist of the sources backing the generated answer, the foundational papers substantiating the sources and the subsequent papers.

The laws of connectedness and continuity are maintained, but the law of proximity is not applied within the visualisation. The position of the nodes is random and does not have any meaning. While this is not a significant problem, it does mean that a dimension of the visualisation is lost, as the position could have been used to represent another data axis. For example, grouping the publications per year causes the publications to be grouped per year, applying the proximity law as an indicator of the release date.

In general, ScienceOS possess good qualities in both the usage of AI and the visualisation aspect. While the visualisation gives limited information, it is a good entry point to visualise the sources mentioned in the generated content. This mechanic allows the user to explore the sources related to the information they need.

3.4.3 Connected Papers

Connected Papers is another web-based application that utilises a search bar to collect the necessary papers. In the search bar, the user can insert the title of a publication, keywords, DOI or other identifiers. A network of related publications is built around the origin paper when given a publication. The network consists of publications represented as nodes, with the year and author given for that publication. The size and colour of the nodes represent the citation count and year of release, respectively. The similarity to the origin paper determines the position of the nodes in the graphs, so papers that are more related to the origin paper are closer to it. Hovering over the nodes reveals the title of the paper and other metadata, such as the authors, the journal in which it was released, citation count, and additional information. The application allows for different publications to be added as origin paper, which expands the graph with other publications related to that paper, creating a larger graph. Lastly, the application provides the option to switch between list views, such as prior works and derivative works and provides a way to apply filters to these lists. An example of the application can be seen in Figure 3.11, with the U-net paper used as the origin paper. The primary purpose of this tool is to visualise publications in a structured manner. The visualisation gives an overview of the relations between the publications and provides a supporting role.



U-Net: Convolutional Networks for Biomedical Image Segmentation

Origin paper
U-Net: Convolutional Networks for Biomedical Image Segmentation
O. Ronneberger, P. Fischer, T. Brox 2015

Fully convolutional networks for semantic segmentation
Evan Shelhamer, Jonathan Long, Trevor Darrell 2014

Deep Residual Learning for Image Recognition
Kaiming He, X. Zhang, Shaoqing Ren, Jian Sun 2015

Image Segmentation using Hybrid Relay Level Sets
Zaheeruddin Syed, K. Suganthi 2023

Very Deep Convolutional Networks for Large-Scale Image Recognition
K. Simonyan, Andrew Zisserman 2014

Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation
Liang-Chieh Chen, Yukun Zhu, G. Papandreou, Florian Schroff, Hartwig Adam 2018

Local and Adaptive Image-to-Image Learning and Inference
Evan Shelhamer 2019

V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation
F. Milletarì, N. Navab, Seyed-Ahmad Ahmadi 2016

Rethinking Atrous Convolution for Semantic Image Segmentation
Liang-Chieh Chen, G. Papandreou, Florian Schroff, Hartwig Adam 2017

Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs
Liang-Chieh Chen, G. Papandreou, Iasonas Kokkinos, K. Murphy, A. Yuille 2014

ImageNet classification with deep convolutional neural networks
A. Krizhevsky, I. Sutskever, Geoffrey E. Hinton 2012

Densely Connected Convolutional Networks
Gao Huang, Zhuang Liu, Kilian Q. Weinberger 2016

3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation
Özgün Çiçek, A. Abdulkadir, S. Lienkamp, T. Brox, O. Ronneberger 2016

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification
Kaiming He, X. Zhang, Shaoqing Ren, Jian Sun 2015

Multi-Scale Context Aggregation by Dilated Convolutions
F. Yu, V. Koltun 2015

U-Net: Convolutional Networks for Biomedical Image Segmentation
O. Ronneberger, P. Fischer, T. Brox
2015, International Conference on Medical Image Computing and Computer-Assisted Intervention
70935 Citations
Save
Open in:
S2 TL;DR: It is shown that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks.

Figure 3.11: An example of the working of Connected papers. The example provides a graph of connection with the publication “U-Net: Convolutional Networks for Biomedical Image Segmentation”. The visualisation represents the connections between the origin paper and other publications. The proximity and size of the nodes indicate the similarity between nodes and citation count, respectively. The list on the left side of the screen represents a table format of the visualisation.

According to the website, the similarity between the publications determines the nodes’ structure and positioning of the graph. While this indicates that machine learning or NLP techniques determine the similarity between publications, the website uses co-citations and bibliographic coupling instead. Bibliographic coupling refers to two publications referencing the same work. For example, publications A and B reference publication C. Co-citation is the opposite of bibliographic coupling. When publication A has publications B and C in their references, publications B and C are co-citations [Sur+11]. Publications with highly overlapping citations and references are presumed to have a related subject, making them similar. Because of this system, they do not use any AI or LLM to determine the similarity between publications.

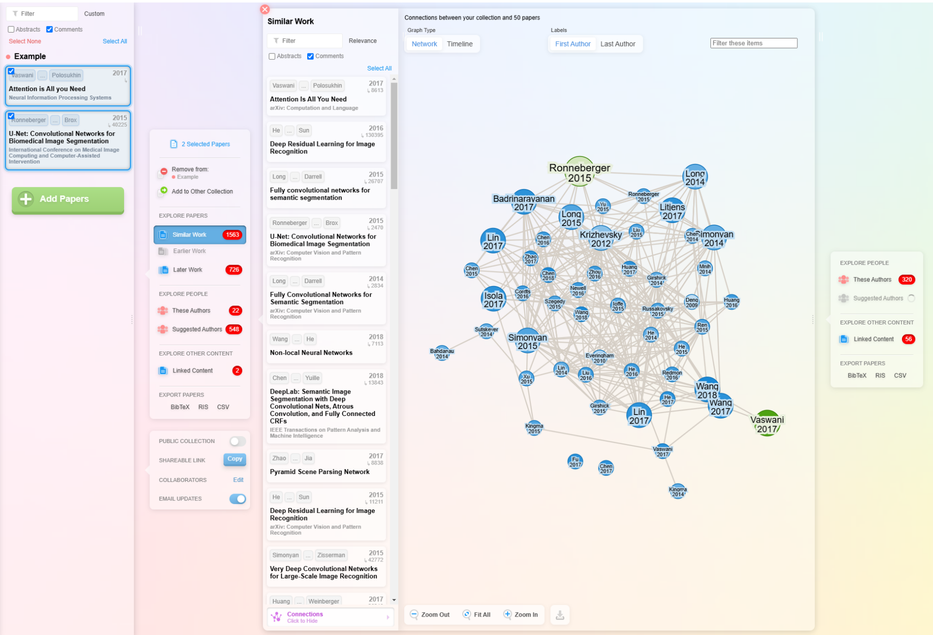
For the Gestalt Laws, the visualisation uses proximity to indicate similarity between the publications. The colour of the nodes can be an indication of similarity, which represents the year when the publication was released. In terms of connectedness and continuity, they implement a strategy similar to the other applications. The links are direct and do not have any abrupt changes in direction.

3.4.4 Research Rabbit

Research Rabbit is a web-based application that the user can utilise to create collections of publications. These collections can make two types of graphs: a citation network and a timeline. Within the citation network, the nodes have labels of the first author's name and the year the publication was released. The colours of nodes indicate the relationship towards the user. Green nodes indicate that the user has the publication in their collection and can be used to create new graphs to fetch similar content. Blue nodes within the network indicate that the paper is related to the publications indicated in green. This relation is either a direct one as a reference or citation, or through an indirect direction by a determined similarity. The intensity of the colour indicates the recency of the publication paper, with the darker colours being more recent and the older publications being lighter colours. A similar graph can be generated, but with the connections between authors, as well as suggested authors. The timeline consists of two columns, where one side consists of the publications in the collection on the left and the related publications on the right. The publications are sorted according to their publication year, with the oldest publications at the bottom and the newest ones at the top. Both the citation network graph and the timeline have a similar way of interacting with the nodes. When hovering the nodes, the application highlights the essential relations between the hovered node and the surrounding nodes. Clicking on the nodes creates a new pop-up window, which delivers additional information about the publication. This interaction also allows the publication to be added to the collection and generate a new graph similar to the original graph. This graph makes it possible to click through the graphs and publications. An example of this application can be seen in Figure 3.12. The use case of this application is to collect publications that are related to the user's research. The visualisations are supporting elements which can be used to argue why a publication is helpful for a collection.

While Research Rabbit is not open about using AI or LLMs, we learn from the website's FAQ that they use a recommendation system for recommending papers. Based on the publications the user reads or that are in the user's collection, the application recommends similar publications that the user has shown interest in.

The network representation of Research Rabbit follows all Gestalt laws, but the law of proximity is not applied in this representation. This is because Research Rabbit chooses to grant more interaction to the user. The user can drag around the nodes and position them as they want so that they can get rid of any clutter. In the timeline visualisation, the proximity is preserved by binding the proximity to the release year of the publication. Through colour, Research Rabbit applies the similarity law, as green nodes are publications that are present in the collection, and blue nodes are publications that are recommended. For continuity, they provide straight lines, and as an extra, they provide linking and brushing when a user hovers over the nodes, revealing which publication it cites and has cited.



☑ Vaswani ... Polosukhin 2017
Attention is All you Need
Neural Information Processing Systems

☑ Ronneberger ... Brox 2015 ↳ 40225
U-Net: Convolutional Networks for Biomedical Image Segmentation
International Conference on Medical Image Computing and Computer-Assisted Intervention

Vaswani ... Polosukhin 2017 ↳ 8613
Attention Is All You Need
arXiv: Computation and Language

He ... Sun 2016 ↳ 130395
Deep Residual Learning for Image Recognition

Long ... Darrell 2015 ↳ 26707
Fully convolutional networks for semantic segmentation

Ronneberger ... Brox 2015 ↳ 2470
U-Net: Convolutional Networks for Biomedical Image Segmentation
arXiv: Computer Vision and Pattern Recognition

Long ... Darrell 2014 ↳ 2834
Fully Convolutional Networks for Semantic Segmentation
arXiv: Computer Vision and Pattern Recognition

Wang ... He 2018 ↳ 7113
Non-local Neural Networks

Chen ... Yuille 2018 ↳ 13843
DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs
IEEE Transactions on Pattern Analysis and Machine Intelligence

Zhao ... Jia 2017 ↳ 8838
Pyramid Scene Parsing Network

He ... Sun 2015 ↳ 11211
Deep Residual Learning for Image Recognition
arXiv: Computer Vision and Pattern Recognition

Simonyan ... Zisserman 2015 ↳ 42772
Very Deep Convolutional Networks for Large-Scale Image Recognition

Huang ... Weinberger 2017

Figure 3.12: An example of the Research Rabbit website. For the generation of the graph, "U-Net: Convolutional Networks for Biomedical Image Segmentation" and "Attention is all you need" have been selected as the primary papers to generate the graph. Within the visualisation, they are represented as green nodes. The other nodes (in blue) are publications seen as similar work. The application allows for the generation of other graphs containing later work or that are related to the authors.

3.4.5 Litmaps

Litmaps follows a similar way of maintaining documents within a collection, such as Research Rabbit and Connected Papers. The website generates a graph visualisation after adding a publication to a list of documents. This graph tries to link all the selected publications by recommending publications that have cited all the selected publications in the collection. This recommendation is based on the algorithm used to search shared citations and references. Litmaps permits users to select other algorithms, such as searching papers with familiar authors or trying to fetch publications with similar text based on the titles and abstracts. The application allows four data visualisations: standard, ring, side-by-side (publications and citations) and by author. Furthermore, the website enables the modification of the x and y axes, allowing them to be sorted by citation or reference count, publication date, momentum or map connectivity. Finally, the website lets users position the nodes directly, highlighting them in purple if a modification occurs. An example of the application can be seen in Figure 3.13, where we show the standard representation of three selected publications. Interacting with the nodes shows the publication's abstract, the number of references and citations, and allows the user to refine their search by adding it to their collection.

As mentioned, Litmaps uses multiple algorithms that influence the publications' recommendations. The only one that uses AI is the analysis of similar text. The algorithm provides publications with similar content by analysing the publication title and abstract. This algorithm is practical if publications have fewer citations or are hard to find in a general scope.

Litmaps provides the user with numerous options to modify the graphs of recommended publications. As mentioned, the user can choose between four representations and modifications of the x-axis and y-axis. The laws of similarity, connectedness and continuity all apply. The similarity between the groups of recommended papers and papers in the collection can be seen. This approach is similar to Research Rabbit's visualisation of recommended versus collected

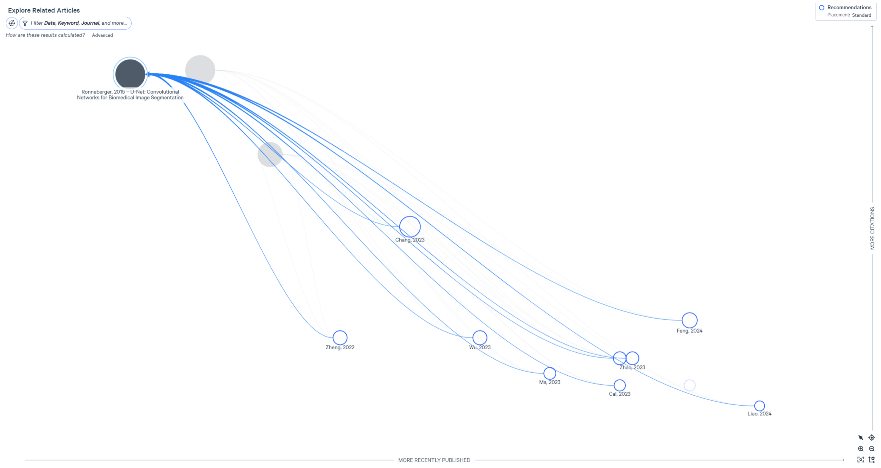


Figure 3.13: An example of the usage of Litmaps. Using three selected publications, we get a graph that shows the citations and references that are related to these publications. While hovering over the node of one publication from the collection, it reveals the relations to the other documents not in the collection. A grey fill represents nodes in the collection, while nodes not present do not have a fill.

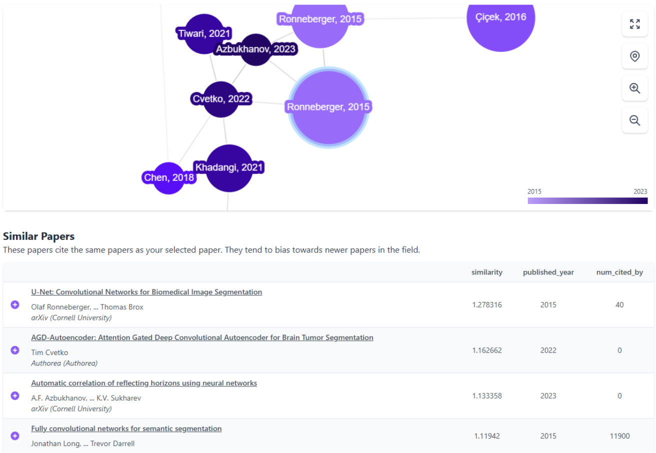
publications. The connectedness and continuity principles are applied through the links of the graph, which are present in each visualisation. Finally, the relevance of proximity is defined by the user. Based on the axes of the standard visualisation, the proximity of nodes has a different meaning. The user can also decide the position of themselves for ordering the visualisation for their own needs, but this takes away any pre-defined meaning in the position of the nodes.

Litmaps is a solid application for collecting publications and saving them in a collection for further reading. The visualisation aspect of Litmaps gives a lot of freedom for the user, allowing them to get an overview of the recommended papers and collected publications. Different search algorithms can assist in a varied search depending on the user's needs.

3.4.6 Inciteful

The last application with a visualisation aspect is Inciteful. Inciteful allows users to look up a publication they want more information about. After looking up other publications related to the submitted publications, a graph and multiple tables that relate to the publication are generated. The tables provided are similar papers, essential papers, review papers, important recent papers and other data. The results give a graph network that shows the connections between the submitted publication and the tables. The nodes represent the publications, with their colour indicating the year of their release. Interacting with the nodes creates a new window, showing the publication's metadata and providing a new visualisation. The visualisation follows the same scheme as the earlier visualisation. An example of the visualisation and the table can be seen in Figure 3.14. To counter the potential cluttering of the graph by having too many related publications, the application introduces a "locking" mechanism. With this mechanism, the user can choose which publications they want. This limits the number of publications in the visualisation, showing only the publications that are locked, along with the publications that are related to the publication. This significantly reduces nodes, which can be seen in Figure 3.15.

An additional functionality of Inciteful is the tool to see the connection between two publications, which is called the "Literature Connector". This tool is meant to bridge the gap between two research fields and how a chain of publications can connect those two. When using it, the application will ask the user to give two publications, the start and end points. It then



Similar Papers

These papers cite the same papers as your selected paper. They tend to bias towards newer papers in the field.

	similarity	published_year	num_cited_by
U-Net: Convolutional Networks for Biomedical Image Segmentation Olaf Ronneberger, ... Thomas Brox *arXiv (Cornell University)*	1.278316	2015	40
AGD-Autoencoder: Attention Gated Deep Convolutional Autoencoder for Brain Tumor Segmentation Tim Cvetko *Authorea (Authorea)*	1.162662	2022	0
Automatic correlation of reflecting horizons using neural networks A.F. Azbukhanov, ... K.V. Sukharev *arXiv (Cornell University)*	1.133358	2023	0
Fully convolutional networks for semantic segmentation Jonathan Long, ... Trevor Darrell	1.11942	2015	11900

Figure 3.14: An example of visualisation and the first table after inputting a publication into Inciteful. The visualisation gathers the information and publications from the different tables and forms a graph with the publications as nodes. The nodes are colour-coded according to the release date of the publication. The table represents which papers are similar to the given publication.

generates a graph similar to the representation in Figure 3.14 and Figure 3.15b.

Insightful recommends publications to users based on two key parameters: importance and similarity. To determine importance, it employs the PageRank algorithm. As defined by Leskovec, Rajaraman, and Ullman [LRU14],"*PageRank is a function that assigns a real number to each page in the Web (or at least to that portion of the Web that has been crawled and its links discovered). The intent is that the higher the PageRank of a page, the more 'important' it is.*". Although this definition initially applies to webpages, Insightful adapts it to assess the importance of publications. While PageRank effectively highlights high-quality publications, it can also cause lesser-known but potentially relevant works to remain unnoticed by users.

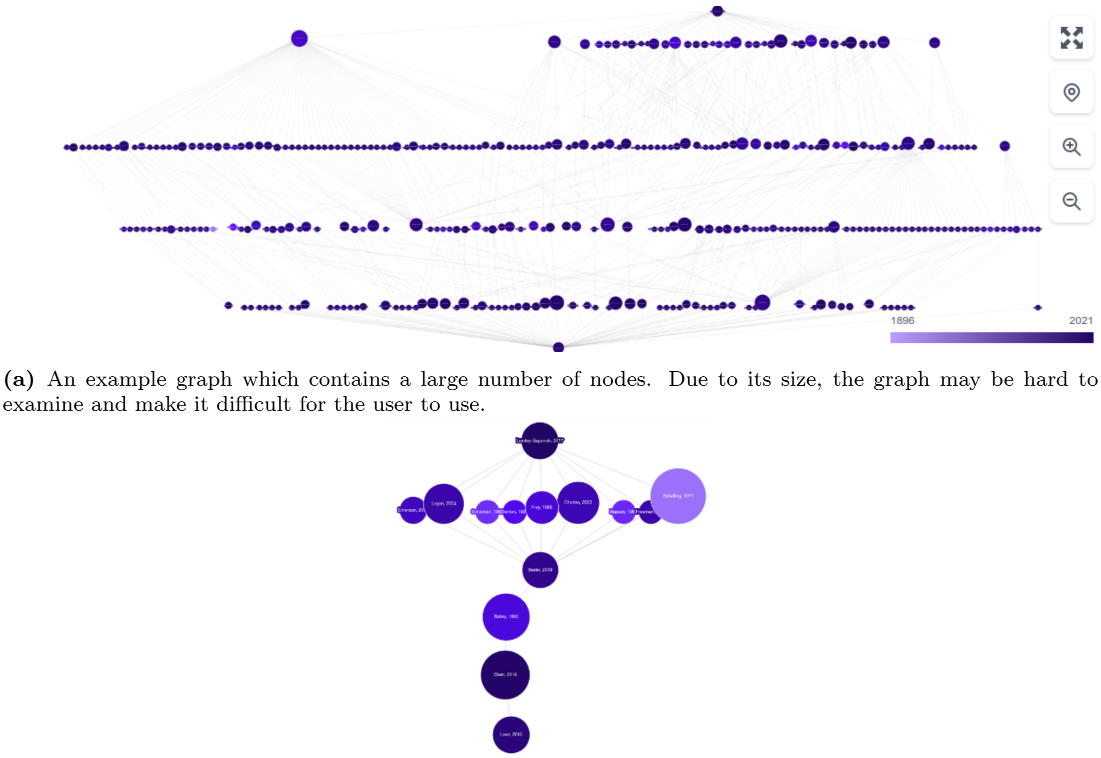
Inciteful uses bibliographic coupling and co-citation to determine the similarity between publications. Depending on the number of publications the user selects, the measurement of similarity changes. If the user has chosen only one publication to compare to, this publication is used to check for similarity. When the user selects multiple publications, coined "seed papers", the application creates a fake publication node, which links the publications together. This fake node is then used to compare the similarity between publications. To further enhance the results coming from the bibliographic coupling and co-citation metric, they use Adamic/Adar and the Salton index to solve some edge cases that both metrics have.

The visualisation of the publications applies to almost all of the mentioned Gestalt Laws. The colours of the nodes in the graph determine when they were released, grouping them per year. Connectedness and continuity are preserved through the links, which are straight lines, making them easy to follow. Proximity is not applied in Inciteful, as the user can determine the position of the nodes according to their needs.

Inciteful is best used to link domains by using the Literature Connector and connecting familiar explored research with unknown domains. Another advantage is that the user can round out their research using a publication to expand on a topic further.

3.4.7 Elicit

Elicit is a predominantly text-based AI-assisted research tool. The central premise of the application is to pose a research question, which will be used to identify relevant papers related to that research question. Upon entering the research question, the tool evaluates the quality of the question, readjusting if there are any missing factors which could improve the question.



(a) An example graph which contains a large number of nodes. Due to its size, the graph may be hard to examine and make it difficult for the user to use.

(b) An example graph that uses the locking mechanism. Thanks to the locking, the graph is drastically reduced in size, making it easier to analyse, which results in a better user experience.

Figure 3.15: An example of the locking mechanism. Figure 3.15a shows an example graph that is large in size, which results in a cluttered visualisation without much oversight. Figure 3.15b shows the same example but with locked publications. This results in a cleaner visualisation, which results in a more user-friendly experience.

When the question is submitted, the application will gather the top four publications related to the question and summarise the contents based on the user's preference. The summaries are added to a table, which users can modify column by column. The table compares the publications based on what the user wants to compare. For example, suppose the user is solely interested in the different methodologies the publications have used. In that case, the user can click to add a column to the table, which will have a summary of the methodology of every publication that the application deems related to the research question. When the user has found their relevant papers, they can choose to continue this research process by creating a new table, doing a new analysis on the selected papers, or "chat" with the publication to further gather insights about the selected paper.

Compared to the application presented in the thesis, Elicit provides more compartmentalised ways to analyse publications per section. This is useful for researchers who have already established a set of publications and want to compare them on a deeper level. Since the application does not use visualisation to show the relation between publications, it could be hard for the user to see the overview of what relates a publication to another.

3.4.8 Scite

Scite is similar to Elicit, as it is a predominantly text-based AI-assisted research tool. The application starts by prompting the user to ask a question, which is then processed by the application. While processing is ongoing, the application gathers multiple sources to ground answers to the prompted question. While this grounding process happens, the application generates an answer to the question while citing sources on the places needed. After the generation is done, the user can ask further questions, which will restart the process, or investigate the sources provided. The sources can be found on the right side of the screen in a reference list. Users can also analyse the search strategy that the AI has used to gather the sources. The application uses three search terms to find relevant papers related to the provided query and takes most of this list as sources. A more in-depth discussion of the work behind Scite can be found in Nicholson et al. [Nic+21], which talks about the processing of scholarly documents and the application's inner workings.

Scite.ai shows similar advantages to those seen in ScienceOS but without visualisation. As discussed, the advantage of this application for the user is that the user can ask a question, and it gets grounded, just like with ScienceOS. The disadvantage, however, is that the user can not see the relations between the sources, making it harder to string publications together. This could make it harder to discover topics.

3.4.9 Overall Comparison

To end the section, we gathered information on previous applications and created a summarising table for a quick overview of the advantages and differences with our table. While each application has its own use cases, they all link to understanding academic connections.

From the table 3.2, we can conclude that most of the tools follow the Gestalt Laws, which is advantageous for the user who utilises the tool. The primary decider, which determines which tool to use, is the AI features the tools provide. Tools like ScienceOS and Scite provide chatbots, which can be used to ask questions about topics and return a summarised answer to the question, providing the sources needed to back up that generated answer, making these tools useful for quick answers. Tools like Research Rabbit, Connected Papers, Litmaps, and Inciteful are the most efficient tools for paper exploration. Their visualisations make it easy to see connections between publications and related topics. When users already have a paper they like, they can submit it to these applications to get similar recommendations, making it easier to expand on their research. Finally, we have the tools that go more in-depth into the publications through summarisation, like Elicit. It allows for a summarisation of individual publications, making it easier to skim over the contents that might be harder to read, speeding up the research process. Our application crosses the bridge between discovery and deeper research while allowing authors to explore new topics they have not encountered. While it lacks features of the other applications, it reveals interesting partitions between topics. It will enable the user to see different fields that a publication could influence.

While comparisons to other applications are insightful, it is essential to reflect on the limitations of the application and consider future directions that could further elevate the experience of the tool. The comparison between previous tools and our application gives us insights into where our implementation could improve. These insights will be further discussed in the next chapter.

Tool	Visualisation				AI utilisation
	Proximity	Similarity	Continuity	Connectedness	
Our application	✓	✓	✓	✓	Summarisation, topic labelling, and topic recommendation
CiteSpace	✓	✓	✓	✓	Topic labelling and grouping
ScienceOS	✗	✓	✓	✓	Chat generation with grounding and publication recommendation
Connected Papers	✓	✓	✓	✓	Paper recommendation
Research Rabbit	✓	✓	✓	✓	Paper recommendation
Litmaps	✓	✓	✓	✓	Multiple paper recommendation algorithms; algorithm based on text similarity
Inciteful	✗	✓	✓	✓	Publication recommendation and connecting topics
Elicit		N/A			Publication summarisation and recommendation
Scite		N/A			Text generation with grounding and publication recommendation

Table 3.2: Comparison of Literature Tools by Visualisation and AI Features

Chapter 4

Discussion

In this chapter, we discuss the limitations of our work and possible changes for future work. Further, I evaluate my performance surrounding the work done for this thesis, discussing my approach, workflows, and strong and weak points.

4.1 Limitations

The tool is developed to help researchers identify relevant papers for their research by showing a timeline to determine recent papers and a citation graph to group papers according to their topic. However, some application goals have not been reached due to constraints. Throughout this section, we discuss the limitations that the user can experience through the front or backend.

4.1.1 Backend

Within the backend, there is room for improvement in multiple areas. The first problem is performance issues that slow down the application. By downloading the papers and analysing them section by section, the application is severely bottlenecked. To cut down on running time within the application, there is a fixed limit for fetching five references and five citations, which reduces the wait times for the user but results in a new graph with limited data. This limit can easily be adjusted, but due to performance, users may have to wait a long time to generate graphs with extensive references and citations. This slow performance can also be seen in the topic recommendations for the authors and the summarisation of publications. This issue primarily persists due to performance issues on the host computer, which cannot use the LLM properly. A solution for this would be to use external LLMs like the OpenAI endpoints for ChatGPT, but this has two consequences. First, there is the cost of using external solutions and second, uploaded documents may not be private, which compromises the ability to use private documents such as company documents. These general performance challenges can result in delays that affect the tool's overall usability and could be bothersome for the users. Another major limitation lies in the reliability of sources for downloading papers.

The tool uses an unreliable source to download the PDFs of publications used as references and citations. While some publications are downloaded without problems, some downloads are corrupted and can not be processed by GROBID, causing some citations or references not to be processed and included in the data. For the root node of the graphs, this can cause the program to crash if there is no root node to process. For this reason, a fail-safe looks up the DOI of the root node on the Semantic Scholar API, gets the paper's ArXiv ID, and downloads the paper using the ArXiv API. While this is implemented for the root node, extending the fail-safe further to the citations and references would be interesting to ensure that all publications are correctly downloaded. While this fail-safe is a proper guardrail for the application, the Semantic Scholar API does have some restrictions.

While the Semantic Scholar API is a reliable way of getting information, it has a strict rate-limit policy without an API key. Depending on which endpoints are used, the rate limit varies. While the legacy endpoints are barely rate-limited, the most recent API does not accept big requests due to the rate limits. To solve this issue, a request was made for an API key that introduces lighter rate limits, but due to the high demand for requests received by others, our request has not been accepted by the Semantic Scholar team. Due to this, the legacy API was primarily used to gather information about publications, which provided limited information. With the API key, it is possible to request bigger batches of information, which could lead to better application performance with fewer requests. However, due to not being able to acquire the API key, the requests now happen individually and with limited data, which is not performant.

Another issue occurred with the processing of the publication abstracts. Some publications and clusters are not processed correctly due to improper filtering of abstracts. Some results from the clustering algorithm resulted in duplicate keywords used for different clusters, which can be confusing. Some abstracts contained a picture, which GROBID analysed as the abstract only having a figure, resulting in the text "fig" as the analysed text. This issue caused the abstracts to be clustered in their own cluster while the contents of the publications did not relate to each other. Another example of a similar problem is that the abstract only contained GitHub links, which caused the publications to be clustered based on these links. This improper processing also happened with some titles of publications, causing some publications to have their release date as the title of the publication. A similar issue occurs with the dates of publications that are visualised in the timeline graph. Due to GROBID not being able to parse a release date in the paper, it is not filled in. This parsing issue is problematic since the timeline needs to be visualised, which requires these dates. As a solution, the Semantic Scholar legacy API is used to retrieve the release date. However, the API only provides the year of release, limiting the level of detail we can utilise. To address this limitation, the frontend compensates by assigning the latest possible date within the given year, displaying the paper as being released on December 31st of that year. This solution could be a misleading factor for the visualisation. While backend issues impact the tool's ability to fetch and process data effectively, certain frontend limitations hinder the user experience in exploring the data.

4.1.2 Frontend

Some features of the frontend could be improved to enhance the user experience, particularly in data exploration and design. Currently, clicking the fetch button within nodes of the timeline or cluster graph causes the entire graph to be replaced with incoming data. This reload disrupts the flow of exploration, making it difficult for users to maintain context or engage in a seamless click-through experience. Furthermore, several functionalities are constrained due to the absence of required endpoints in the backend. For example, the possibility to use a different publication to visualise, as outlined in Section 4.1.1. These limitations reduce the interactivity and flexibility of the frontend, hindering the overall user experience. Within both representations, the information state can be modified by fetching papers or using the back button. This back button uses a stack implementation that stores new data states. Each time, when fetching a paper, the current application state is pushed onto the top of the stack. When removing a state from the top of the stack, the corresponding state is discarded and no longer saved. Because of this implementation, previously fetched papers must be re-fetched, which is inefficient. Finally, as mentioned before, there are duplicate entries in the legend of the citation graph. While the keywords of the clusters are different, the end topic remains the same, which could confuse the user in the frontend.

Compared to other applications, users may also find this application to be less customisable to their liking. While other applications provide colour palettes or the ability to move around nodes, our application only supports the three options for colour-blind users. Giving users the ability to customise enables them to further personalise the visualisation to their liking, which can be helpful if they want to present the visualisation in their way.

4.2 Future work

The tool could be improved with the limitations in mind for future work. The performance is slow, so performing multiple processes on different threads could speed up the process. Further, to mitigate the data reliability, getting an API key for the Semantic Scholar API could improve the amount of data that can be handled successfully. This improvement also allows the ability to gather more information on the publications, resulting in more information to show on the frontend. The API key helps with the API rate limit, which allows for more simultaneous calls to the endpoints and helps with the application's performance. The data processing could also be solved by fetching proper data from the API instead of trying to analyse the papers. Then, the API or the analysis could be a backup to ensure the displayed data is correct. While this thesis uses a clustering approach, implementing a topic modeling solution that uses LLMs as topic models could also be an option. For example, the topics could be decided by BERTopic [Gro22]. In the frontend, it would be better if the original state of the graph stays after fetching a paper. This expansion of the graph allows for a broader visualisation. The trade-off, however, is that the graph itself may get much more cluttered due to the introduction of extra data. Although these improvements address the application's limitations, additional opportunities exist to expand the work further. While these are points of improvement based on the limitations of the work, there is the possibility of adding extras to the tool as well.

Visualising the comparison of papers based on available data could provide valuable insights into the influence of one paper on a research topic compared to another. This approach offers an intriguing perspective on how different works contribute to a field. A comparison could also consist of taking sections from the papers and comparing them based on the provided text. An example would be to use an LLM to compare two methodology sections to compare the approach of publications on a similar topic. Differences in opinions between papers could provide valuable insights into the evolution of a research topic. For instance, a publication from 1980 might present conclusion A, while a more recent publication from 2020 could argue conclusion B. Visualising these differences and tracking how perspectives shift over time would offer a compelling way to understand the development of the field. This mechanism could be implemented by creating a pipeline where a user can submit a claim. With a RAG system, relevant documents can be fetched to support that claim. These supporting claims can be reformulated into contradicting claims and given to the RAG system to search for claims contradicting the passages. Another addition could be a filtering function for the two graphs, limiting the amount of visualised data and giving a more transparent view if there is a lot of data on the screen.

Depending on the user's use case, different directions can be seen for the application. If the user wants more of an AI focus on the application with a more in-depth analysis of the publications, the application allows for expansion on that use case. The application already keeps track of the sections in publications thanks to the analysis of GROBID. Further utilisation of the LLMs could result in a deeper analysis of the sections. Still, it could not be done in the current state of the application due to resource issues mentioned in Section 4.1.1. Another use case would be the collection of publications, as seen in Research Rabbit and Connected Papers, where saving publications to a collection is the primary functionality. While no current functionalities could be used to support this use case, it could be seen as an advantageous function for users through user profiles. User profiles could also be used for the authors, who would get their topic recommendation based on the publications that they wrote and get suggested publications based on the author's publications in their collection.

Another addition would be to expand on the research question. The sub-questions are answered through practical application, as seen in the representations of the visualisations in the application and the usage of the LLMs throughout the workflow of the program. The research question is further answered through theoretical examples seen throughout Chapter 2. Expansion on the research question would be to test other visual implementations and further expand on LLMs beyond an assisting role.

Lastly, a user study would be an interesting direction for evaluating the application's efficiency. This was not pursued during the thesis, as the focus was primarily on technical implementation and prototyping. The application is not yet suitable for user testing at its current stage, as key elements such as accessibility, customisation, and performance require further refinement. Enhancing these aspects would lead to a more polished user experience, laying a stronger foundation for a meaningful user study and enabling comparisons with existing solutions.

While these are all considerations for future work, this work has given me the time to self-reflect on my skills as a programmer and a student.

4.3 Self-reflection

Throughout this thesis, I gained valuable knowledge across multiple domains, from understanding LLMs to developing a robust application integrating advanced technologies. Working with LangChain to set up an environment for LLM applications taught me the importance of structuring systems involving components like word embeddings, vector stores, and chat models. The visualisation development was another significant learning opportunity for my career in frontend development. While it was my first time working with D3.js and Svelte, I enjoyed the challenge of integrating these libraries to create meaningful and interactive visualisations. This process also served as a refresher in frontend and backend development, reinforcing skills in full-stack development. Despite the challenges, the overall application development was a rewarding and successful process, despite significant room for improvement.

One key challenge I faced was poor time management and prioritisation. I focused heavily on developing the application, often at the expense of writing the thesis. As a result, the writing process felt rushed and left minimal time for refining the application. In hindsight, balancing the technical and written components of the project more effectively would have improved both the thesis and the application. Additionally, while the tool functions as intended, it feels underwhelming compared to my initial vision. Many features could be expanded to improve the application's performance and user experience. Future projects will benefit from dedicating more time to refining details and setting realistic goals for what can be achieved within the given timeline. Another problem is that I went too fast in development before considering what I wanted to create. After analysing the other applications after development to perform a comparison, I realised that there are some aspects I have missed in use cases, which could have been advantageous for the user. Working with user profiles would have been a great addition to the application, but due to the pre-built structure of the application, this would not have been easy to implement.

Beyond my technical skills, I have gained confidence as a developer through hard work and perseverance. While this thesis was a challenging and ambitious project, it taught me the importance of maintaining momentum and not giving up. Throughout the process of writing and development, I experienced moments of doubt and struggled to stay confident. However, by continuing to push forward despite these challenges, I now feel more assured in my abilities and more capable in the field of IT.

Despite these challenges, this experience has been instrumental in shaping my approach to future projects. The skills I developed in managing complex systems, integrating novel tools, and creating interactive visualisations have provided a strong foundation for my professional growth. Moving forward, I plan to implement better time management strategies, such as setting milestones for technical and documentation tasks, to ensure balanced progress. Besides this, it is crucial for me to occasionally step back and look at the bigger picture of what I have created. Stepping back allows me to ask questions about the application, evaluate if I am proud of the application and question why I have approached a problem the way I did. This mentality also applies to my writing of code, which should be done with more planning. While this project revealed areas for improvement, it has also equipped me with the knowledge and experience to tackle similar challenges with greater confidence.

Chapter 5

Conclusion

To conclude the thesis, we highlight the findings throughout the thesis, discuss the results, and further discuss the changes and future work.

This thesis is a bundle of information surrounding integrating LLMs into visualisations, as well as a resource for exploring visualisations for citation networks. With this, we built a system that supports researchers in their literature reviews and helps them discover relations between publications. We explored the assistance of LLMs by introducing them to a custom visualisation of citation networks. This assistance gives researchers extra avenues to explore citation graphs and helps them interpret data. We keep two things in mind. First, there is a need to examine how LLMs can assist with visualisations and make them more effective. Second, we want to see what representations can be used to visualise relations between academic literature properly. To reach the first goal, we explored the fundamentals of LLMs, examining their development processes and the factors that have shaped their current state. With this, we became aware of the strengths of the LLMs and how we can utilise them through summarisation, labelling, and recommendation.

After exploring the possibilities of LLMs, we focus on the second part of the research question by discovering numerous ways in which citation graphs can be visualised, which focuses on visualisation possibilities within citation graphs and networks. Most visualisations revolve around citation graphs and graph networks as a visualisation, representing publications as nodes and links as relations between publications. While this is a popular approach to represent the citation graphs, others use visualisations to reveal ties between publications that are relevant to the viewer. An interesting dimension to explore is the temporal relations between the publications, allowing users to seek ties between temporal data and the publications. While these two representations are valid, more visualisations beyond the topic of publications are possible and could be helpful within this field. While keeping these visualisations in mind, we revisit the Gestalt laws and Tufte's principles to follow guidelines while we develop the application.

The developed system is a web application consisting of a timeline and cluster graph visualisation. For the timeline, the publications form a tree, which is ordered by time of release, with the second parameter being the citation count. With this visualisation, users can see impactful publications and when these publications were released, compared to the analysed publication. The cluster graph represents the citations and references of the analysed publication in a grouped manner, with the groups represented as topics. This grouping is done by comparing abstracts and using agglomerative clustering. These clustering results are then used to form groups, which a Transformer can use to determine the topics of the clusters, which label the group. Through this visualisation, the user can determine which groups were influenced, which have grown the most over the years and other use cases. Additionally, the application supports summarisation and topic recommendation through LLMs, further enhancing the research aspect of the application through AI. By comparing our application to others, we can say we have advantages similar to those of other major applications, depending on the use case fit for the user.

While we explored the assistance of LLMs within visualisations through summarisation, topic recommendation and topic labelling, there are more areas where LLMs can be further used, such as storytelling and annotations. We can conclude that our research answers the first sub-question of our research question through practical implementation and research seen in Section 2.2. While we explored timelines and cluster visualisations as possibilities in the context of citation networks, it answered the second research question through practical implementation and research. The representation of related documents can be done by timelines and graphs, but is not limited to those representations, as seen in Section 2.3.4.

The development of the application was generally successful, but there are notable limitations that may hinder its full potential. Due to a lack of resources, the performance of the LLMs and the analysis of the publications are slowed down, which can be frustrating for the user when utilising the application. Additionally, there is room for expanding the visualisation's exploration features, such as enabling comparisons between publications or further customisability. Lastly, because of some mental blocks, the development did have some problems, which resulted in a less performative application.

Despite these limitations and areas for improvement, this project has provided valuable learning opportunities. I gained a deeper understanding of LLMs' inner workings and enhanced my skills in developing visualisations and frontend development. While challenges remain, such as improving my prioritisation and time management, I take pride in the work I have accomplished for this thesis.

Bibliography

[Alo+18] Md Zahangir Alom et al. *The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches.* 2018. arXiv: 1803.01164 [cs.CV]. URL: https://arxiv.org/abs/1803.01164.

[BCB16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate.* 2016. arXiv: 1409.0473 [cs.CL]. URL: https://arxiv.org/abs/1409.0473.

[Ble12] David M. Blei. "Probabilistic topic models". In: *Commun. ACM* 55.4 (Apr. 2012), pp. 77–84. ISSN: 0001-0782. DOI: 10.1145/2133806.2133826. URL: https://doi.org/10.1145/2133806.2133826.

[BW02] U. Brandes and T. Willhalm. "Visualization of bibliographic networks with a reshaped landscape metaphor". In: *Proceedings of the Symposium on Data Visualisation 2002.* VISSYM '02. Barcelona, Spain: Eurographics Association, 2002, 159–ff. ISBN: 158113536X.

[Che04] Chaomei Chen. "Searching for intellectual turning points: Progressive knowledge domain visualization". In: *Proceedings of the National Academy of Sciences* 101.suppl_1 (2004), pp. 5303–5310. DOI: 10.1073/pnas.0307513100. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.0307513100. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0307513100.

[Che18] Gang Chen. *A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation.* 2018. arXiv: 1610.02583 [cs.LG]. URL: https://arxiv.org/abs/1610.02583.

[Cho+14] Kyunghyun Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* 2014. arXiv: 1406.1078 [cs.CL]. URL: https://arxiv.org/abs/1406.1078.

[Cho+18] Gyeongcheol Choi et al. "Citation Network Visualization of Reference Papers Based on Influence Groups". In: *2018 IEEE 8th Symposium on Large Data Analysis and Visualization (LDAV).* 2018, pp. 96–97. DOI: 10.1109/LDAV.2018.8739176.

[Cho10] Gobinda G Chowdhury. *Introduction to modern information retrieval.* Facet publishing, 2010.

[Clo+14] J. R. Clough et al. "Transitive reduction of citation networks". In: *Journal of Complex Networks* 3.2 (Nov. 2014), pp. 189–203. ISSN: 2051-1329. DOI: 10.1093/comnet/cnu039. URL: http://dx.doi.org/10.1093/comnet/cnu039.

[dbl24] dblp. *dblp: Publications per year.* Nov. 2024. URL: https://dblp.org/statistics/publicationsperyear.html.

[Dev+19] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2019. arXiv: `1810.04805 [cs.CL]`. URL: `https://arxiv.org/abs/1810.04805`.

[DGM16] Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. "Word Embeddings, Analogies, and Machine Learning: Beyond king - man + woman = queen". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers.* Ed. by Yuji Matsumoto and Rashmi Prasad. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 3519–3530. URL: `https://aclanthology.org/C16-1332`.

[Din+24] Bosheng Ding et al. "Data augmentation using llms: Data perspectives, learning paradigms and challenges". In: *Findings of the Association for Computational Linguistics ACL 2024.* 2024, pp. 1679–1705.

[Dör+12] Marian Dörk et al. "PivotPaths: Strolling through Faceted Information Spaces". In: *IEEE Transactions on Visualization and Computer Graphics* 18 (2012), pp. 2709–2718. URL: `https://api.semanticscholar.org/CorpusID:2057737`.

[EW17] Nees Jan van Eck and Ludo Waltman. "Citation-based clustering of publications using CitNetExplorer and VOSviewer". In: *Scientometrics* 111.2 (May 2017), pp. 1053–1070. ISSN: 1588-2861. DOI: `10.1007/s11192-017-2300-7`. URL: `https://doi.org/10.1007/s11192-017-2300-7`.

[Ezu+22] Absalom E. Ezugwu et al. "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects". In: *Engineering Applications of Artificial Intelligence* 110 (2022), p. 104743. ISSN: 0952-1976. DOI: `https://doi.org/10.1016/j.engappai.2022.104743`. URL: `https://www.sciencedirect.com/science/article/pii/S095219762200046X`.

[Fah+14] Adil Fahad et al. "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis". In: *IEEE Transactions on Emerging Topics in Computing* 2.3 (2014), pp. 267–279. DOI: `10.1109/TETC.2014.2330519`.

[For+18] Santo Fortunato et al. "Science of science". In: *Science* 359.6379 (2018), eaao0185. DOI: `10.1126/science.aao0185`. eprint: `https://www.science.org/doi/pdf/10.1126/science.aao0185`. URL: `https://www.science.org/doi/abs/10.1126/science.aao0185`.

[Gao+23] Caroline X. Gao et al. "An overview of clustering methods with guidelines for application in mental health research". In: *Psychiatry Research* 327 (2023), p. 115265. ISSN: 0165-1781. DOI: `https://doi.org/10.1016/j.psychres.2023.115265`. URL: `https://www.sciencedirect.com/science/article/pii/S0165178123002159`.

[Gia+23] Charlie Giattino et al. "Artificial Intelligence". In: *Our World in Data* (2023). https://ourworldindata.org/artificial-intelligence.

[GRO08] GROBID. *GROBID.* `https://github.com/kermitt2/grobid`. 2008–2024. swh: `1:dir:dab86b296e3c3216e2241968f0d63b68e8209d3c`.

[Gro22] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[GS12] Andrea Gesmundo and Tanja Samardžić. "Lemmatisation as a tagging task". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2.* ACL '12. Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 368–372.

[GSC00] Felix Gers, Jürgen Schmidhuber, and Fred Cummins. "Learning to Forget: Continual Prediction with LSTM". In: *Neural computation* 12 (Oct. 2000), pp. 2451–71. DOI: 10.1162/089976600300015015.

[Hea24] K. Healy. *Data Visualization: A Practical Introduction*. Princeton University Press, 2024. ISBN: 9780691270845. URL: https://books.google.be/books?id=v7gfEQAAQBAJ.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[Hut+24] Maeve Hutchinson et al. *LLM-Assisted Visual Analytics: Opportunities and Challenges*. 2024. arXiv: 2409.02691 [cs.HC]. URL: https://arxiv.org/abs/2409.02691.

[JM24] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. Online manuscript released August 20, 2024. 2024. URL: https://web.stanford.edu/~jurafsky/slp3/.

[Joh+21] Prashant Johri et al. "Natural Language Processing: History, Evolution, Application, and Future Work". In: *Proceedings of 3rd International Conference on Computing Informatics and Networks*. Ed. by Ajith Abraham, Oscar Castillo, and Deepali Virmani. Singapore: Springer Singapore, 2021, pp. 365–375. ISBN: 978-981-15-9712-1.

[KH17] Taraneh Khazaei and Orland Hoeber. "Supporting academic search tasks through citation visualization and exploration". In: *International Journal on Digital Libraries* 18.1 (Mar. 2017), pp. 59–72. ISSN: 1432-1300. DOI: 10.1007/s00799-016-0170-x. URL: https://doi.org/10.1007/s00799-016-0170-x.

[Kie+23] Douwe Kiela et al. "Plotting Progress in AI". In: *Contextual AI Blog* (2023). https://contextual.ai/blog/plotting-progress.

[Kin+23] Rodney Michael Kinney et al. "The Semantic Scholar Open Data Platform". In: *ArXiv* abs/2301.10140 (2023). URL: https://api.semanticscholar.org/CorpusID:256194545.

[Lee+21] Doris Jung-Lin Lee et al. "Boomerang: Proactive Insight-Based Recommendations for Guiding Conversational Data Analysis". In: *Proceedings of the 2021 International Conference on Management of Data*. SIGMOD '21. Virtual Event, China: Association for Computing Machinery, 2021, pp. 2750–2754. ISBN: 9781450383431. DOI: 10.1145/3448016.3452748. URL: https://doi.org/10.1145/3448016.3452748.

[Lin+22] Tianyang Lin et al. "A survey of transformers". In: *AI Open* 3 (2022), pp. 111–132. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2022.10.001. URL: https://www.sciencedirect.com/science/article/pii/S2666651022000146.

[LRU14] Jure Leskovec, Anand Rajaraman, and Jeffrey D. Ullman. *Mining of Massive Datasets*. Second. Cambridge University Press, 2014. ISBN: 1107015359. URL: http://mmds.org.

[Met+18] Ronald Metoyer et al. "Coupling Story to Visualization: Using Textual Analysis as a Bridge Between Data and Interpretation". In: *Proceedings of the 23rd International Conference on Intelligent User Interfaces*. IUI '18. Tokyo, Japan: Association for Computing Machinery, 2018, pp. 503–507. ISBN: 9781450349451. DOI: 10.1145/3172944.3173007. URL: https://doi.org/10.1145/3172944.3173007.

[MGF12] Justin Matejka, Tovi Grossman, and George Fitzmaurice. "Citeology: visualizing paper genealogy". In: *CHI '12 Extended Abstracts on Human Factors in Computing Systems.* CHI EA '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 181–190. ISBN: 9781450310161. DOI: `10.1145/2212776.2212796`. URL: `https://doi.org/10.1145/2212776.2212796`.

[Mik+13] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space.* 2013. arXiv: `1301.3781 [cs.CL]`. URL: `https://arxiv.org/abs/1301.3781`.

[Mor+03] Steven A. Morris et al. "Time line visualization of research fronts". In: *Journal of the American Society for Information Science and Technology* 54.5 (2003), pp. 413–422. DOI: `https://doi.org/10.1002/asi.10227`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.10227`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.10227`.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2008. URL: `http://nlp.stanford.edu/IR-book/`.

[Nic+21] Josh M. Nicholson et al. "scite: A smart citation index that displays the context of citations and classifies their intent using deep learning". In: *Quantitative Science Studies* 2.3 (Nov. 2021), pp. 882–898. ISSN: 2641-3337. DOI: `10.1162/qss_a_00146`. eprint: `https://direct.mit.edu/qss/article-pdf/2/3/882/1970740/qss\_a\_00146.pdf`. URL: `https://doi.org/10.1162/qss%5C_a%5C_00146`.

[NIS15] Rina Nakazawa, Takayuki Itoh, and Takafumi Saito. "A Visualization of Research Papers Based on the Topics and Citation Network". In: *2015 19th International Conference on Information Visualisation.* 2015, pp. 283–289. DOI: `10.1109/iV.2015.58`.

[NMC17] Mang'are Fridah Nyamisa, Waweru Mwangi, and Wilson Cheruiyot. "A Survey of Information Retrieval Techniques". In: *Advances in Networks* 5.2 (2017), pp. 40–46. DOI: `10.11648/j.net.20170502.12`. eprint: `https://article.sciencepublishinggroup.com/pdf/10.11648.j.net.20170502.12`. URL: `https://doi.org/10.11648/j.net.20170502.12`.

[Pee+24] Jannes Peeters et al. "Snowflake: visualizing microbiome abundance tables as multivariate bipartite graphs". In: *Frontiers in Bioinformatics* 4 (2024). ISSN: 2673-7647. DOI: `10.3389/fbinf.2024.1331043`. URL: `https://www.frontiersin.org/journals/bioinformatics/articles/10.3389/fbinf.2024.1331043`.

[RAK07] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks". In: *Physical Review E* 76.3 (Sept. 2007). ISSN: 1550-2376. DOI: `10.1103/physreve.76.036106`. URL: `http://dx.doi.org/10.1103/PhysRevE.76.036106`.

[RG19] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Nov. 2019. URL: `https://arxiv.org/abs/1908.10084`.

[RN19] Francis A. Ruambo and Mrindoko R. Nicholaus. "Towards Enhancing Information Retrieval Systems: A Brief Survey of Strategies and Challenges". In: *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT).* 2019, pp. 1–8. DOI: `10.1109/ICUMT48472.2019.8970954`.

[Sal+18] Hojjat Salehinejad et al. “Recent Advances in Recurrent Neural Networks”. In: *CoRR* abs/1801.01078 (2018). arXiv: 1801.01078. URL: http://arxiv.org/abs/1801.01078.

[Sev+18] Rita Sevastjanova et al. “Going beyond Visualization. Verbalization as Complementary Medium to Explain Machine Learning Models”. In: 2018. URL: https://api.semanticscholar.org/CorpusID:52834158.

[SH18] Maurice Schleußinger and Maria Henkel. “Knowde: A Visual Search Interface”. In: June 2018, pp. 191–198. ISBN: 978-3-319-92269-0. DOI: 10.1007/978-3-319-92270-6_26.

[Sun+24] Ye Sun et al. *GeneticPrism: Multifaceted Visualization of Scientific Impact Evolutions.* 2024. arXiv: 2408.08912 [cs.DL]. URL: https://arxiv.org/abs/2408.08912.

[Sur+11] Ganesh Surwase et al. “Co-citation Analysis: An Overview”. In: (Sept. 2011).

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks.* 2014. arXiv: 1409.3215 [cs.CL]. URL: https://arxiv.org/abs/1409.3215.

[SZD24] Ali Reza Sajun, Imran Zualkernan, and Sankalpa Donthi. “A Historical Survey of Advances in Transformer Architectures”. In: *Applied Sciences* 14 (May 2024), p. 4316. DOI: 10.3390/app14104316.

[Tan+16] Zhaowei Tan et al. “AceMap: A Novel Approach towards Displaying Relationship among Academic Literatures”. In: *Proceedings of the 25th International Conference Companion on World Wide Web.* WWW ’16 Companion. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016, pp. 437–442. ISBN: 9781450341448. DOI: 10.1145/2872518.2890514. URL: https://doi.org/10.1145/2872518.2890514.

[TG83] Edward R Tufte and Peter R Graves-Morris. *The visual display of quantitative information.* Vol. 2. Graphics press Cheshire, CT, 1983.

[Tou+23] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models.* 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[Vas+23] Ashish Vaswani et al. *Attention Is All You Need.* 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[VHE15] Marco Antonio Valenzuela-Escarcega, Vu A. Ha, and Oren Etzioni. “Identifying Meaningful Citations”. In: *AAAI Workshop: Scholarly Big Data.* 2015. URL: https://api.semanticscholar.org/CorpusID:2538517.

[Voi+22] Henrik Voigt et al. “The Why and The How: A Survey on Natural Language Interaction in Visualization”. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 348–374. DOI: 10.18653/v1/2022.naacl-main.27. URL: https://aclanthology.org/2022.naacl-main.27/.

[vW14a] Nees Jan van Eck and Ludo Waltman. “CitNetExplorer: A new software tool for analyzing and visualizing citation networks”. In: *Journal of Informetrics* 8.4 (2014), pp. 802–823. ISSN: 1751-1577. DOI: https://doi.org/10.1016/j.joi.2014.

07.006. URL: https://www.sciencedirect.com/science/article/pii/S1751157714000662.

[vW14b] Nees Jan van Eck and Ludo Waltman. "CitNetExplorer: A new software tool for analyzing and visualizing citation networks". In: *Journal of Informetrics* 8.4 (2014), pp. 802–823. ISSN: 1751-1577. DOI: https://doi.org/10.1016/j.joi.2014.07.006. URL: https://www.sciencedirect.com/science/article/pii/S1751157714000662.

[War21] Colin Ware. "Chapter Six - Static and Moving Patterns". In: *Information Visualization (Fourth Edition)*. Ed. by Colin Ware. Fourth Edition. Interactive Technologies. Morgan Kaufmann, 2021, pp. 183–243. ISBN: 978-0-12-812875-6. DOI: https://doi.org/10.1016/B978-0-12-812875-6.00006-2. URL: https://www.sciencedirect.com/science/article/pii/B9780128128756000062.

[Wu+16] Yonghui Wu et al. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016. arXiv: 1609.08144 [cs.CL]. URL: https://arxiv.org/abs/1609.08144.

[YS19] Mehmet Yalcinkaya and Vishal Singh. "Exploring the use of Gestalt's principles in improving the visualization, user experience and comprehension of COBie data extension". In: *Engineering, Construction and Architectural Management* 26.6 (Jan. 2019), pp. 1024–1046. ISSN: 0969-9988. DOI: 10.1108/ECAM-10-2017-0226. URL: https://doi.org/10.1108/ECAM-10-2017-0226.

[Zha+23] Haochen Zhang et al. "Large language models as data preprocessors". In: *arXiv preprint arXiv:2308.16361* (2023).

[Zhu+25] Qian Zhu et al. "CompositingVis: Exploring Interactions for Creating Composite Visualizations in Immersive Environments". In: *IEEE Transactions on Visualization and Computer Graphics* 31.1 (2025), pp. 591–601. DOI: 10.1109/TVCG.2024.3456210.

[ZJZ18] Li Zhang, Ming Jing, and Yongli Zhou. "Embedded Temporal Visualization of Collaboration Networks". In: *Advances in Multimedia Information Processing – PCM 2018*. Ed. by Richang Hong et al. Cham: Springer International Publishing, 2018, pp. 89–98. ISBN: 978-3-030-00764-5.