SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Informatics

User-Centric Design of an Exploratory Search System for Scholarly Entities in Natural Language Processing

Ferdy Dermawan Hadiwijaya

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Informatics

User-Centric Design of an Exploratory Search System for Scholarly Entities in Natural Language Processing

Nutzerzentierter Design eines explorativen Suchsystems für wissenschaftliche Entitäten in Natural Language Processing

Author:	Ferdy Dermawan Hadiwijaya
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Tim Schopf
Submission Date:	16.10.2023

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, 16.10.2023 Ferdy Dermawan Hadiwijaya

Acknowledgments

First and foremost, I would like to thank to my advisor, Tim Schopf, for his continued support during the thesis. Thanks for taking a chance on me. This thesis would not have been successful without your input. I would also like to thank Prof. Dr. Florian Matthes for supervising this thesis.

I would like to also extend my thanks to my family and friends for all the support they have given me up to this point. You have made me "me" and I'd like to return your support with all I can.

After living four years in Germany, I take comfort in my first milestone. I am grateful for these experiences, and I am thankful for everyone who has made it possible.

Abstract

Natural Language Processing (NLP) as a research field has gained an increasing popularity for the last few decades, evident by the ever-increasing number of publications related to NLP. The rapid developments in NLP research pose a problem to its researchers. Notably, someone without prior knowledge faces a barrier to an unknown research field. While there are numerous tools enable exploratory search of scientific literature in NLP, none has directly communicated the pain point of this rapidly expanding research field.

To tackle this, we proposed the development of an exploratory search system for scholarly entities in Natural Language Processing. The development followed the user-centric design framework, putting usability and user experience as primary objectives. Our proposed solution includes a frontend, search engine, a graph database, and a vector database to enable exploratory search. This solution emphasizes on the importance of graph connections of entities and a semantic powered search engine. The development is divided to two iterations, both characterized by the user evaluation at the end of its phase.

Kurzfassung

Das Forschungsgebiet des Natural Language Processing (NLP) hat in den letzten Jahrzehnten zunehmend an Popularität gewonnen, wie durch die kontinuierlich steigende Anzahl von NLP-bezogenen Veröffentlichungen deutlich wird. Die rasche Entwicklung in der NLP-Forschung stellt für Wissenschaftler eine Herausforderung dar, insbesondere für diejenigen ohne Vorkenntnisse in diesem speziellen Forschungsfeld. Es gibt zwar viele Tools, die die explorative Suche in der wissenschaftlichen Literatur im Bereich NLP ermöglichen, aber keines hat bisher die rasanten Entwicklungen in diesem Feld direkt adressiert.

Um dieses Problem anzugehen, haben wir die Entwicklung eines explorativen Suchsystems für wissenschaftliche Entitäten im Bereich Natural Language Processing vorgeschlagen. Diese Entwicklung folgte dem benutzerzentrierten Designansatz, bei dem Benutzerfreundlichkeit und die Erfahrung der Nutzer als Hauptziele gesetzt wurden. Unsere vorgeschlagene Lösung beinhaltet ein Frontend, eine Suchmaschine, eine Graphdatenbank und eine Vektordatenbank, um eine explorative Suche zu ermöglichen. Besonders betonen wir die Bedeutung von Verbindungen zwischen Entitäten in Form von Graphen und die Nutzung einer semantisch gestützten Suchmaschine. Die Entwicklung erfolgte in zwei Iterationen, wobei jede Phase durch eine Bewertung der Nutzer am Ende der Phase gekennzeichnet war.

Contents

1 Introduction

1.1 Motivation

Natural Language Processing (NLP) is a subset of computer science concerned with the computational analysis or synthesis of natural language. NLP as a research field has gained even more popularity since the adoption of pre-trained language models. The popularity is evident from the ever-increasing number of publications related to NLP.

Additionally, the number of fields of study within the NLP domain has grown. The rapid developments in NLP research pose a problem to its researchers. Professional researchers can face difficulty keeping up with state-of-the-art papers, whereas getting into a new research subfield of NLP is more challenging due to the sheer amount of existing publications.

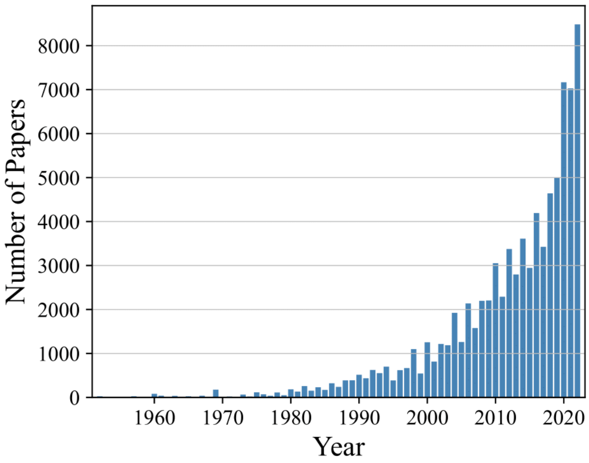


Figure 1.1: Number of NLP papers in ACL Anthology from [SAM23].

Numerous tools enable the exploration and search of scientific literature in NLP. Because these tools complement each other and a researcher has to manage multiple simultaneously, their research workflow is considerably complicated. This fragmentation of functions can also hinder the researcher's ability to recognize connections between different publications. The added complexity consumes valuable time and may result in overlooking critical insights and trends within the NLP domain. Furthermore, the existing tools are general scientific platforms that have existed for years with minimum changes, lacking features specific to NLP and its fast-paced, wide-ranging nature.

This thesis is part of the research project *Natural Language Processing Knowledge Graph (NLP-KG)* [Schb] conducted at the Software Engineering for Business Information Systems chair at Technical University of Munich (TUM). The thesis serves as a proof of concept and user interface to the data set.

1.2 Proposed Solution

To tackle the problems mentioned above head-on, we propose a User-Centric Design (UCD) of an exploratory search system for scholarly entities in NLP. This approach aims to identify users' problems and iteratively develop a solution that brings real value to the users and satisfies their immediate needs. More specifically, the solution is a web application that combines multiple modern technologies such as static site generation for the frontend, semantic search for the search engine, and graph database for exploration. Such technologies aim to enable a performant web interface and relevant search results.

This customized solution intends to streamline the exploratory search process and address the unique needs of NLP researchers. Our web app should filter and present information specific to its use more effectively while highlighting the connections between entities such as publications, fields of study, venues, and researchers. By providing an integrated platform, we strive to enhance the research process of the ever-evolving landscape of NLP research.

1.3 Research Questions

Our study aims to design and develop a web application tailored to the needs of NLP researchers. This effort is aimed at addressing the distinctive challenges faced by NLP researchers. The research questions related to these objectives are presented as follows:

RQ1 *What are the existing approaches of researchers to search, explore, and keep up with the NLP research?*

Identifying the methodologies used by NLP researchers in literature research involves examining current approach's pain points. These issues are planned to be addressed through our proposed solution that strives for incremental improvements over existing methods.

RQ2 *How can we curate and present information in a web application to support user-friendly search and exploration of scholarly entities in NLP?*

The users' primary requirement on the application involves identifying and extracting relevant information, such as publications. It should enable users to efficiently navigate the web application and access all the integrated features. Therefore, it is essential to determine the most effective way to present this information without introducing unnecessary complexities.

RQ3 *What approaches can we use to achieve a performant semantic search and exploration of relevant scholarly entities in Natural Language Processing?*

Performance plays a substantial role in enhancing the user experience within an exploratory search system. This study aims to integrate state-of-the-art approaches in semantic search and web application engineering, providing our users with a tool for accurate and performant literature research in the NLP domain.

RQ4 *How can we systematically evaluate the usefulness of our proposed approach to our target users?*

In a user-centric development approach, users play a crucial role in influencing and guiding changes within the product. Thus, user experience and usability evaluation analyses are imperative in each iteration. Questions will be designed to gather pain points, pinpoint potential areas for improvement, and achieve an evaluation of our web application.

2 Theoretical Foundations

This study is grounded in two relevant theories — User-Centric Design (UCD) and Semantic Search. We will delve into these concepts to provide a clear context and establish their significance within the study. Our user-centric development of an exploratory search system calls for a robust theoretical foundation as a guide. In this chapter, we will explore the theories that support each implementation phase.

2.1 User-Centric Design

User-centric design is an iterative design framework focusing on the user's experience at each stage of the design process. The widespread recognition of this framework can mainly be attributed to the books 'User Centered System Design: New Perspectives on Human-Computer Interaction' [ND86] and 'The Design of Everyday Things' [Nor02]. Despite the name *design*, user-centric design does not mean it is limited to graphical design processes. Instead, UCD implies that the whole process is centered around the user. The users drive the design, development, and evaluation of the project.

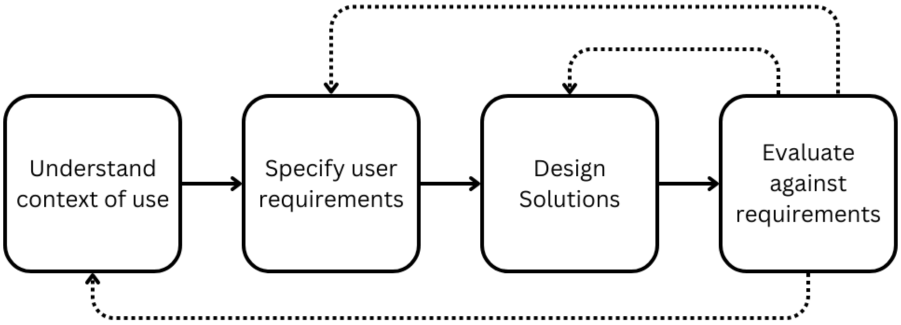

Understand context of use
Specify user requirements
Design Solutions
Evaluate against requirements

Figure 2.1: The iterative process of UCD.

Figure 2.1 shows the general workflow of User-Centric Design. Each iteration of the UCD approach involves four distinct phases:

1. Specify the context of use: Identify the people who will use the product, what they will use it for, and under what conditions they will use it.
2. Specify requirements: Identify any business requirements or user goals that must be met for the product to be successful.
3. Create design solutions: This part of the process may be done in stages, building from a rough concept to a complete design.
4. Evaluate designs: Evaluation — ideally through usability testing with actual users — is as integral to good software development as quality testing.

The four processes are conducted iteratively until the evaluation results align with the specific context of users and meet their relevant needs. UCD focuses on addressing the overall user experience through conscious attention to users.

In an ideal scenario, a UCD-oriented development involves people from multiple disciplines, domain experts, stakeholders, and users. Furthermore, it is crucial to maintain long-term usage monitoring to gather data for ongoing enhancements in user experience.

2.2 Semantic Search

Semantic search is a search technique that focuses on understanding the meaning of the words and context in a query to provide more relevant search results. It is an alternative to the traditional keyword-based search, which relies primarily on matching specific words. The difference lies in how both approaches attempt to find relevant results. This difference gives semantic search advantages over keyword-based search in certain use cases.

In semantic search, the exact wording becomes less relevant. For instance, consider the search query “Best ice cream in Munich”. Traditional syntactic search algorithms may prioritize a result like “Best Schnitzel in Munich” over “Munich’s most delicious ice-creams”. This preference can be attributed to the syntactic similarities: "Best", "in", and "Munich". In contrast, semantic search recognizes the similarities between words and that “Best ice-cream in Munich” is more closely related in meaning to “Munich’s most delicious ice-creams” than to “Best Schnitzel in Munich”. The semantic capability is crucial for modern search engines because there is often a discrepancy between the query’s wording and the results’ wording.

2.2.1 Embedding: The numerical representation

In order to understand the meaning of a word, these words have to be translated into a language that computers understand: numbers. This numerical representation of natural language is called semantic embedding. This embedding takes the shape of vectors of real numbers that encode meaningful semantic information.

2.2.2 Pre-trained Language Models

A pre-trained language model is a deep learning model previously trained on a large corpus of text data. Language models are designed to tackle complex tasks in NLP, such as language translation. Semantic search requires a language model that generates semantic embeddings from publications to find the most similar publication to the search query.

2.2.3 Similarity Search

So far, we have learned about semantic search, how a text is represented, and how to generate them. Now, we discuss how to determine the similarity of two embedded texts.

Distance Metrics

There are multiple measures to determine how similar two embeddings are. As previously mentioned in subsection 2.2.1, the embeddings have the form of vectors. Both terms will be used interchangeably in this subsection. In this thesis, we use Cosine Similarity to determine the vector similarity. As the name implies, cosine similarity uses a cosine function to measure the angle between two vectors projected in a multidimensional space.

Cosine Similarity between two vectors $\vec{A}$ and $\vec{B}$ is calculated as:

$$\text{Cosine Similarity}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{||\vec{A}|| ||\vec{B}||} = \frac{\sum_1^n A_i B_i}{\sqrt{\sum_1^n A_i^2} \sqrt{\sum_1^n B_i^2}}$$

Cosine Similarity has a value bound by a constrained range of 0 and 1. This range can then determine how similar two natural language texts are, given their embeddings in the form of vectors. A Cosine Similarity of 1 implies that two texts are exactly alike. A cosine Similarity of 0 would conclude that there are no similarities between the two texts.

Cosine Similarity is advantageous to other distance measures in the context of semantic search because of its properties. Cosine similarity measures how similar two texts are, irrespective of their size. Two similar documents with varying lengths will still have a smaller angle between them. The smaller the angle, the higher the similarity.

3 Related Works

No research was found at the intersection of exploratory search systems, user-centric design, and natural language processing research. This thesis aims to fill this gap and explore the application of user-centric design to tackle the problem mentioned earlier in natural language processing research. To achieve our scientific goals, we systematically identified and analyzed existing freely accessible research solutions.

- **Google Scholar** is an academic search engine that indexes scholarly literature online. The developers leveraged a web crawler to identify papers for inclusion in the search results. Google Scholar includes academic journals and books, conference papers, thesis and dissertations, preprints, abstracts, technical reports, and other forms of scholarly literature, including court opinions and patents. Although the index has more than 300 million documents [Gus19], it is important to note that many of these documents suffer from inaccuracies in their metadata [Jac10]. Furthermore, this index includes freely accessible open-access materials and documents locked behind a paywall.

- **Semantic Scholar** is a search engine for scientific literature developed at the Allen Institute for AI (AllenAI)—the tool leveraged machine learning techniques to improve the search process for scholarly papers. Semantic Scholar's scope was initially limited to papers in computer science [Jon15]. However, the database has since expanded to encompass over 200 million publications spanning various science disciplines [Mat21]. The developers included paper summaries and an adaptive research recommendation system. Additionally, it offers easy access to graphs and data extracted from the papers while emphasizing connections and relationships between publications using graph structures. Unlike Google Scholar, Semantic Scholar only includes free open-access papers.

- **DBLP** is a public computer science bibliography database founded at the computer science department of the University of Trier and currently managed by Schloss Dagstuhl — Leibniz Center for Informatics [DBLb]. DBLP originally stood for Database Systems and Logic Programming but has been taken to stand for Digital Bibliography and Library Project. The database provides open bibliographic information on major computer science journals and proceedings, counting over six million. To update its database, it mainly seeks to obtain the data from the major publisher directly [DBLa]. The team behind DBLP mainly focuses on the quality of the metadata database, resulting in a lower database count and poor search features on the website.

- **Papers with Code** is an open online database for Machine Learning papers, code, datasets, methods, and benchmarks [Pap]. The core Papers with Code is based on Meta AI Research. The data is annotated manually by the community or the team. Its key advantage is its task classification of the Machine Learning domain. Each task has a list of papers, codes, benchmarks, and datasets, making it super convenient for Machine Learning engineers to browse the state-of-the-art approaches in Machine Learning.

- **Connected Papers** is an exploration tool for scientific papers. The tool is unique in that it visualizes the graph connections between papers. The graph is arranged according to the similarity score of the papers. The similarity score considers two papers similar if they have highly overlapping citations and references [Con]. This website is used for getting a visual overview of a new academic field, highlighting relevant papers, and creating the bibliography for theses.

It is important to note that the list presented here is not exhaustive but rather a curated selection of the most widely recognized tools. These tools serve different purposes in literature research: search tool (Google Scholar and Semantic Scholar), bibliographic database (DBLP), graph exploration tool (connected papers), and community-driven code, papers, and benchmark database (Papers with code). These tools complement one another by offering distinct functionalities.

Our initial analysis affirms the hypothesis that a considerable fragmentation of tools exists within the NLP domain. This fragmentation presents challenges in understanding the connections and trends in the field.

In the following, we will discuss further solutions: Google Scholar and Semantic Scholar. These tools enjoy widespread recognition and are comparable to our proposed exploratory search system.

Google Scholar’s primary strength lies in its extensive repository of scientific literature, coupled with its integration into mainstream Google Search results. Nevertheless, Google Scholar’s search functionality relies on keyword-based search and cannot capture the nuance of a search query. Furthermore, its advanced search capabilities are limited to rule-based syntactic search and date filtering.

Semantic Scholar, on the other hand, distinguishes itself through its machine-learning-powered features. It offers a semantic search feature, generates paper summaries, and extracts essential elements from the PDF-format paper, such as figures, data, tables, and citations. While navigation is possible between certain entities like citation graphs and researchers, Semantic Scholar does not extract information such as fields of study and venues from papers, limiting the navigation to two entities.

This preliminary survey of related tools gives us a first insight into the current status of tools for literature research. Notably, Google Scholar and Semantic Scholar each possess unique strengths and limitations. Our thesis aims to tackle the limitations of existing tools, improving exploratory search for scholarly entities in NLP.

4 Methodology

We previously presented our primary objective of enhancing the exploratory search process for NLP researchers. We proposed a web-based solution to facilitate a more streamlined exploratory search, emphasizing users significantly during its development. The research questions discussed in chapter 1 provide a structure for us to pursue this goal.

The research questions reflect the iterative process of User-Centric Design. First, we need to understand the specific challenges NLP researchers face and understand what they truly need (RQ1). Subsequently, we developed a web application to address these challenges. Our development process encompasses frontend (RQ2) and search engine (RQ3) perspectives. The solution is subject to systemic evaluation by actual users (RQ4), from which evaluation will be handled in the next iteration.

The user-centric design framework outlines our development approach, ensuring that user experience remains the focus of each development phase. In the following sections, we will elaborate on the concrete steps undertaken to realize the user-centric development of our exploratory search system for scholarly entities in the Natural Language Processing domain.

4.1 User Research

Our proposed exploratory search system should target various search and exploration processes of scholarly entities in NLP. Therefore, it aspires to serve any individuals engaged in NLP Research, regardless of their level of expertise or affiliations with for-profit enterprises, public institutions, or academic establishments. The intended user ranges from students researching sub-fields of NLP for their master's theses to industry professionals seeking to stay updated with the latest advancements in the field.

4.1.1 Initial user interviews

To identify the typical research workflow of NLP researchers, verify our initial hypotheses, and discover the pain point of existing NLP research tools, we interviewed two early-career NLP researchers currently active at TUM.

We purposefully selected early-career researchers for our interviews. As early-career researchers, they need to constantly learn about new domains, making them frequent users of exploratory search systems. Additionally, they are not entrenched in the current workflow and tools and are open to new ideas and tools.

While our sample size is small, it was intentionally chosen to allow for more in-depth discussions with the interviewees. We believe that the insights gained from these interviews can be valuable, even though the sample size is not representative of the full range of users we target.

The interview questions can be divided into three sections:

1. Investigation of exploratory search workflow: This section explored the interviewees' research workflow concerning searching for and exploring scientific papers.
2. Pain points in NLP researchers' workflow: Six statements about possible problems in NLP research workflow were posed as questions with a Likert scale response, ranging from 1 (strongly disagree) to 5 (strongly agree).
3. Possible solution: This section discussed a possible solution to the pain points identified in the previous two sections.

4.1.2 Crafting User Personas

Persona is a fictional representation of a target user based on real-world data. It describes the user's demographics, psychographics, and motivation [Blo02]. Illustrating the characteristics of a few fictional figures can provide a holistic understanding of the target audience. Demographics such as age and occupation reveal users' basic behaviors, while psychographics such as interest and values delve deeper and provide a more nuanced understanding of the users. Furthermore, empathy toward users' motivation guides us to align the application with users' desires and aspirations for a better user experience.

We created two user personas as follows:

- **Persona 1: Wesley, Master's Student**
  - Name: Wesley Christiansen
  - Age: 28
  - Occupation: PhD student
  - Research Interest: Explainable Large Language Model
  - Years of Experience: 3

  - Location: Aarhus, Denmark
  - Background: Wesley started pursuing a Ph.D. program at Aarhus University two years ago, where he wants to invent a novel approach to the Explainable Large Language Model field, his research focus. He gained interest in NLP early in his master's program. So, he worked as an assistant researcher in the NLP research group of his university before deciding to start his Ph.D. program.
  - Problems: Wesley faces challenges in locating relevant and significant papers and gaining a comprehensive understanding of emerging research areas due to the rapidly evolving nature of explainability and Large Language Models. As a Ph.D. candidate, he finds it hard to find time to sort, filter, and read the relevant papers from the ever-expanding literature.
  - Motivation: Wesley wants to contribute to top journals and help develop innovative approaches. In order to achieve this, he needs another way to filter publications more effectively and recognize which papers are most relevant to read.

- **Persona 2: Emily, Machine Learning professor**
  - Name: Emily Lee
  - Age: 53
  - Occupation: Professor
  - Research Interest: Newest advancements in NLP
  - Years of Experience: 20
  - Location: New Jersey, United States of America
  - Background: Emily has been teaching at a university in New Jersey for five years. Though not actively researching and developing new techniques, she wants to stay updated with the latest advancements in applied Machine Learning, including NLP.
  - Problems: She does not have much free time as a professor. With the vast amount of NLP publications released each year, she finds it hard to prioritize which publications to read first. Additionally, Emily is often overwhelmed with new research areas in NLP. Because the research areas under NLP are dynamic, Emily has to spend many hours to get an overview of a research subfield of NLP.
  - Motivation: Emily wants a way to get a quick overview of new research fields she is interested in. She wants to explore recent publications more efficiently

by seeing how things are related. She wants a more holistic understanding of a research paper and its area.

In conclusion, a persona can help developers understand and empathize with their users and create web applications that meet users' needs. Personas are essential for user-centric design at every stage, from ideation to evaluation, to help ensure the application is user-friendly and successful.

4.1.3 User-Centered Evaluation and Requirements Elicitation

The iterative development process incorporates user feedback as a fundamental component. Following each development cycle is a methodical evaluation centered around the users' perspectives. This user evaluation aims to assess the system's usability, identify possible pain points, and gather more requirements for the next iteration. To facilitate this, we employ the System Usability Scale for usability testing, followed by open discussion.

This study incorporates two user evaluation rounds for the two iterations, later outlined in Chapter 5. These rounds consist of three and five participants, respectively. Of these participants, five are master's students from TUM who have used search engines to find NLP publications, and three are Ph.D. candidates from TUM with a research focus on NLP. The number of participants follows the guide as described by Nielsen et al. (May 1994).

The System Usability Scale (SUS) is an easy method for assessing system usability. It comprises ten items that offer a global view of subjective usability assessments [Bro96]. SUS measures usability through Likert scale questionnaires that give respondents five response options, ranging from Strongly Agree to Strongly Disagree. The System Usability Scale is employed after the respondents use the system but before any debriefing or discussion takes place. The respondents should answer the questions immediately rather than thinking about items for a long time.

The System Usability Scale measures different aspects of the system, such as effectiveness (i.e., the ability of users to successfully achieve their objectives), efficiency (the effort and resources expended in achieving said objectives), and satisfaction (reflecting the overall user experience). This approach facilitates a systematic and reliable means of obtaining a high-level understanding of a system's usability status. Ultimately, the System Usability Scale supports the evaluation and improvement of the system from the users' perspectives.

4.2 Prototyping

Prototyping is the starting point of the actual implementation. Prototypes are a tool to test and validate a vision concretely through preliminary user feedback. Contrary to the recommendations made by Walker et al. (Sept. 2002), we opted to develop a medium-to-high fidelity prototype for several reasons. Our decision was informed by a study that argued that there was no significant difference in usability testing outcomes between low and high-fidelity prototypes.

While low-fidelity prototypes offer the advantage of facilitating faster iteration during the initial phase of development, we, however, decided to proceed with creating a high-fidelity prototype. This decision can be attributed to the time constraints, limiting the possibility of any iteration at the prototyping.

The prototyping and design process was conducted using the Figma platform, and the result was used for an early user evaluation and the anticipated web application implementation.

4.3 Dataset

As part of the "Natural Language Processing Knowledge Graph" research project, the application leverages an existing dataset sourced from open-access archives with scientific literature related to Natural Language Processing [Schb]. These archives include arXiv [Arx], Semantic Scholar [Scha], and ACL Anthology [Ant].

A previous related study by Schopf, Arabi, and Matthes (2023) extracted a taxonomy of the research fields that are part of the NLP domain. Additional entities and relations have been extracted since then and stored in a graph database.

By the time of this thesis's publication, this database encompassed the following four primary entity types:

- **Publication** represents the category of all instances of NLP publications identified within the dataset. These publications include conference papers, journal papers, workshop papers, demonstrations, evaluation papers, theses, and dissertations. We must note that we do not differentiate these publications based on their sources; however, each publication may possess optional attributes from the sources.

- **FieldOfStudy** represents the category of various research fields within the domain of NLP. These fields of study have been established according to the taxonomy of research domains by Schopf et al. (2023).

- **Researcher** represents the category of authors and researchers found in the data source who have written at least one publication in the field of NLP.
- **Venue** represents the category of various entities such as journals, organizations, magazines, or institutions that serve as hosts or publishers for Publications within the domain of NLP.

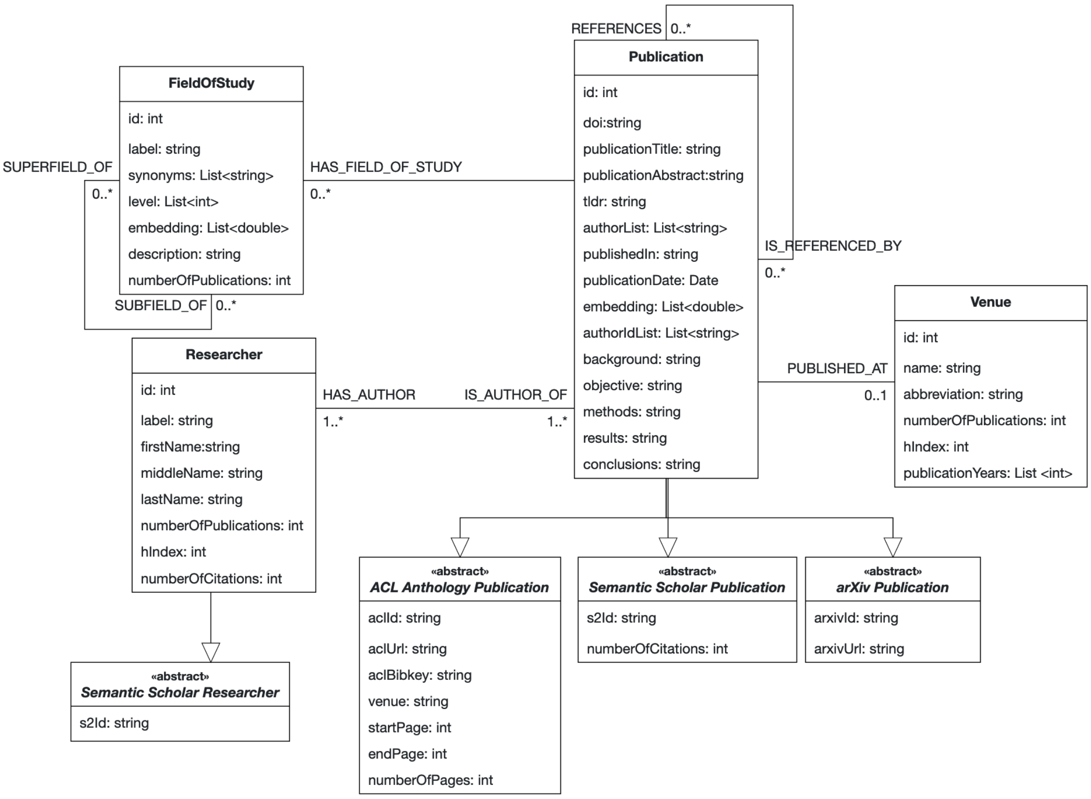


Figure 4.1: Entity Relationship Diagram of our graph dataset

4.4 Software Engineering

The user experience is a central piece of this study. This goal requires thorough planning in the engineering of our application. Our application is made of four integral components: the frontend, the search engine, the graph database, and the vector database. Figure 4.2 depicts the relations between each component.

There are two main design goals for the applications:

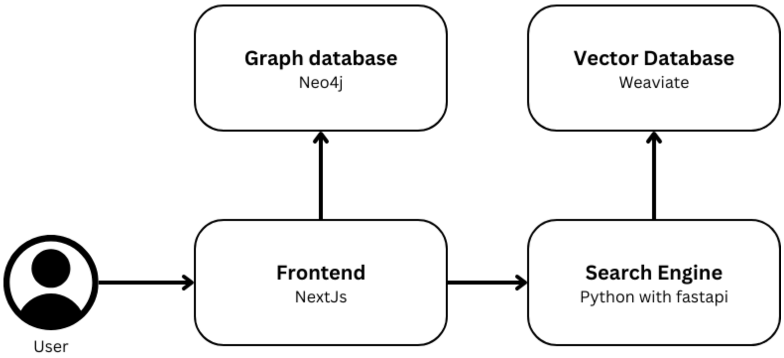


Figure 4.2: The software architecture of our web application

- **Separation of Concerns** is an essential principle for building any application. It promotes modularity, maintainability, and scalability in software development. Hence, we divide the application into the four previously mentioned components with different functions. This division enables better development, testing, and maintenance.

  When any parts of the application need to handle more workload, they can be scaled independently for performance improvements. Furthermore, the separation of concerns supports greater flexibility and adaptability because individual components can be replaced without affecting the entire application.

- **Performance** refers to the overall application speed. This goal depends on the efficiency of each component and the speed at which the components communicate with one another. For user experience, we aim to improve runtime speed at the cost of build time and greater memory and disk usage. The trade-off can be achieved by executing resource-hungry computations in advance. While optimization is not the main goal of this thesis, we aim to keep a high performance to improve user experience.

4.4.1 Frontend Development (Next.js)

Frontend is the part of an application that provides the user interface. Thus, developing the frontend component is important and requires extensive attention. The selection of frontend technology is influenced by three key factors: design considerations, development speed, and performance metrics.

Styling is a big part of frontend development, and there are many frameworks to choose from. We utilize Tailwind CSS as our main styling method for its utility CSS classes. These classes correspond to a CSS style, which can be used to style an HTML component quickly [CSS]. However, styling a component typically requires multiple classes, slowing the initial development. So, we use component classes from DaisyUI for the earlier stage of development [UI]. The DaisyUI component classes will mostly be replaced with pure Tailwind CSS classes for more customization.

An equally important decision was the selection of a UI framework, library, or paradigm. Our application relies on Next.js, a meta-framework for React. React itself is a simple JavaScript library for building interactive user interfaces [Rea]. It offers useful functions that improve development efficiency and experience compared to raw JavaScript. React lets programmers put components together, but it does not prescribe how to do routing and data fetching. This is where Next.js comes in. Next.js handles the tooling and configuration needed for React and provides additional structure, features, and optimization for the application [Nex].

Next.js supports static site generation, permitting the retrieval of data and the construction of web pages during the build phase instead of runtime execution. Static Site Generation mitigates database access requirements for certain pages while delivering accelerated page rendering. This does come with a trade-off involving memory consumption for storing static sites/pages and a longer build-time duration. However, this trade-off justifies the enhanced user experience resulting from swifter page loading.

Besides its simplicity and robust framework, our decision for Next.js and React was based on two other factors: familiarity and static side generation feature. Our familiarity with React supports smoother and faster development. Furthermore, Next.js supports static site generation, which fetches data and builds web pages at build-time instead of runtime. Static Site Generation lowers page render time and reduces database access for some pages, with the trade-off of memory needed to contain these static sites/pages and longer build-time. A faster page load would enable a better user experience, so we took on this trade-off.

In addition to React, Next.js, Tailwind CSS, and DaisyUI, our development stack incorporates TypeScript and tRPC. TypeScript enables type safety for JavaScript [Typ], while tRPC is an API library that enables typesafe procedure calls between Next.js' server and client component [tRP]. These additions facilitate better developer experience and faster development.

4.4.2 Databases (Neo4j and Weaviate)

Our application implements two distinct databases, namely Neo4j and Weaviate, each serving different roles within the system architecture.

Neo4j is the primary repository for our graph data and manages our dataset. The frontend component interacts with this database during the build-time and runtime phases. During build-time, neo4j allows the frontend component to generate static pages of every entity, while in the runtime, neo4j completes the search and exploration feature. The decision for the database owes to the fact that neo4j is an established industry standard for graph databases.

On the other hand, Weaviate functions as a vector database. It contains the embeddings of all publications and some relevant metadata. Embeddings and metadata of the publications are required for a hybrid search, combining semantic and keyword-based searches. Weaviate supports two indices: The approximate nearest neighbor(ANN) index for semantic search and the inverted index for keyword-based search [Wea]. Hybrid search combines both search indices to enhance search result precision and relevance. Notably, Weaviate uses the Hierarchical Navigable Small World algorithm [MY16] for ANN indexing and BM25 [RZ09] for keyword-based search. Its speed and search accuracy supports the decision for Weaviate.

4.4.3 Search Engine (Python)

The ability to search is an important feature of the application. Consequently, we set up a dedicated RESTful server component that generates search recommendations. This component is built using Python due to its machine learning support [RPN20]. Its features allow for quick development, suitable for our goals. The search engine provides RESTful API endpoints through FastAPI framework, a simple and performant web framework for Python [Fas].

The search process consists of two distinct stages: retrieval and reranking. During retrieval, the server interfaces with Weaviate, our vector database, employing the hybrid search methodology elucidated earlier. Subsequently, publications are subject to the ranking by a reranking algorithm, which considers semantic relevance, syntactic similarity, citation count, and temporal recency for default relevancy sorting.

Due to time constraints, the scope of our search engine is limited to searching publications. The search engine is structured into two stages: retrieval and re-ranker. During retrieval, the search engine queries relevant publications from Weaviate, our vector database, through the hybrid search. Then, the publications will be ranked through a re-ranker algorithm based on semantic meaning, syntactic similarity, number of citations, and recency.

The embeddings used for publication retrieval are generated using a pre-trained language model called "Scientific Paper Embeddings using Citation-informed Transformers (SPECTER)." SPECTER takes a publication's title and abstract as input to generate a document-level embedding [Coh+20]. It is based on SciBERT [BLC19], a pre-trained language model for general scientific texts.

For the default sort option based on relevancy, the search engine uses the open-sourced re-ranker model used by Semantic Scholar [Fel], which also includes a finetuned language model. Otherwise, the retrieved publications will be sorted based on the metadata.

It is essential to acknowledge possible performance challenges with this approach. The redundancy of retrieving and reranking publications for identical queries exists but with a different offset or sort option. This redundancy worsens the application performance and negatively impacts the overall user experience. To mitigate this, we implement caches required to speed up the search process.

Additionally, a balance must be struck between the amount of retrieved publications and an accurate search result. An excessive number of retrieved publications might slow down the whole speed process, while an insufficient number could result in the required publication not being retrieved at all.

We will discuss the result of the development of the search engine in the subsequent chapter of this thesis.

4.4.4 Development Setup

Leibniz-Rechenzentrum (Leibniz Supercomputing Center) of the Bavarian Academy of Sciences and Humanities hosts our application on two servers.

One of the servers was previously established for the ongoing research project "Natural Language Processing Knowledge Graph", running the neo4j database instance. At the same time, the other contains the rest of the application components, namely the frontend, search engine, and the Weaviate database. The latter runs the application through Docker for easy maintenance, security, and fast deployment [BBA17].

5 Results

This chapter presents the results obtained from the study conducted, which emphasized a user-centric approach to the design of an exploratory search system tailored for scholarly entities within the realm of Natural Language Processing.

The previous chapter explained in detail the methodology of this study. A major part of our methodology is the adoption of the iterative development model based on the user. So, the research began with initial user study and prototyping, followed by the iterative phases of development and user evaluation. In the following, the results of our study will be laid out in detail and arranged in chronological order.

5.1 Initial user interviews

Subsection 4.1.1 examined the methodology employed during the initial user interviews. These interviews serve as the foundation of our research, upon which the subsequent design and development phases are constructed.

The initial user interviews yielded several significant findings. These findings encompassed the following dimensions:

5.1.1 Existing Approaches

This section delved into the existing approaches identified from the interviewees' workflow. The exploratory search for scholarly entities in the field of Natural Language Processing can primarily be categorized into exploration and search methods.

Exploration Methods

The exploration of relevant publications within NLP research typically involves three distinct methods:

- **Subscription to Scientific Journals and Venues:** Many NLP researchers subscribe to specific scientific journals and academic venues or follow notable researchers in the field through social media platforms like Twitter. However, these sources often introduce noise and irrelevant publications into the researcher's feed.

- **Keyword-Based Searches:** Experienced researchers frequently employ specific keyword searches using tools like Google Scholar and Semantic Scholar. These keywords are typically related to their research interests and may include the name of an NLP task, techniques, or the broader research area.
- **Navigating Scholarly Entity Connections:** Some researchers opt for exploration by navigating through scholarly entity connections, which can involve traversing citation graphs or exploring a researcher's publications.

Search Methods

The search process in literature research largely relies on search engines, some of which are listed in Chapter 3. Two tools commonly used in this context are:

- **Google Scholar:** This widely-used platform supports an extensive collection of academic papers. A factor of its popularity can be attributed to its association with Google, the world's most utilized search engine. However, Google Scholar predominantly relies on syntactic search, potentially missing the nuanced aspects of a query. Its extended search features are primarily rule-based syntactic search and date filters.
- **Semantic Scholar:** Leveraging NLP techniques, Semantic Scholar streamlines literature research and highlights connections within a research field using graph structures. The platform hosts the entities and relationships of researchers and publications. It serves as a benchmark in this research, given its advanced features. Semantic Scholar also used a two-stage search engine, with a keyword-based search for retrieval and S2Ranker for reranking. Although the search result has been satisfactory, its search engine is slower than Google Scholar's by far.

Research implications

Our proposed solution seeks to enhance the exploratory search experience in NLP by focusing on search and navigation through additional scholarly entity connections, e.g., field of study. The existing methods often require some level of prior knowledge about a specific research field, potentially restricting entry for unfamiliar users. Moreover, Google Scholar and Semantic Scholar lack additional but relevant entities to enable seamless navigation, offering room for improvements.

5.1.2 Requirements

An analysis ensued from both explicit and implicit user feedback. This analysis, combined with a comparative evaluation of comparative evaluation of existing NLP research approaches, yielded the following functional and non-functional requirements.

Functional Requirements

- **FR1: Navigational Entities.** The system must facilitate the exploration of scholarly entities, including publications, fields of study, and researchers within the NLP domain. Dedicated pages for each entity type should display their connections, allowing users to navigate seamlessly between related entities through hyperlinks.
- **FR2: Semantic Search for Publications.** The application must support a semantic search for publications. The results must allow for searching relevant publications with different phrasings of two words with similar meanings.
- **FR3: Sort and Filter options.** The search functionality should offer sorting options based on the number of citations, recency, and a custom re-ranker optimizing for the publication's semantic and syntactic relevance as well as both recency and citations. Users should be able to filter search results based on selected fields of study for publications and sort researchers by metrics such as citations, publications, and h-index.
- **FR4: Fields of Study Hierarchy.** The system should allow users to navigate through a hierarchical structure of fields of study. The taxonomy of fields of study must be readily accessible, and the hierarchy should be visible on both publication and field of study pages, complete with navigable links and connections.
- **FR5: Information Completeness.** The system must ensure that users can access and view all relevant information on each application page. Data, including metrics such as h-index, citations, and publications, should be consistent with the latest database updates.

Non Functional Requirements (NFR)

- **NFR1: Minimalist user interface.** The user interface must be designed clearly and concisely such that the information is displayed without unnecessary clutter and distraction.
- **NFR2: Optimal Response Time.** The system must prioritize optimal response times, minimizing delays during user interactions. Swift response times should

be maintained for navigation, data retrieval, and search operations. In cases where slower responses are unavoidable, immediate feedback mechanisms, such as placeholders with shimmer effects, should be implemented.

- **NFR3: Accurate and Consistent Search Results.** The system must consistently deliver accurate search results. Accuracy is determined by the relevance of scholarly entities in terms of recency, popularity, and their proximity to the search query. Consistency ensures that search results remain stable and dependable, yielding identical outcomes for the same query.
- **NFR4: Reliable Control and Navigation Behaviour.** The control and navigation behavior of the system must be reliable and predictable, offering an intuitive and user-friendly user interface. User controls, such as buttons and navigational links, should be clearly identifiable, enhancing the overall user experience.

5.2 Early Design Prototypes

This section delved into the initial prototypes of our exploratory search system for scholarly entities in NLP. In the previous section, we identified the key requirements, both functional and non-functional, for the development of the proposed solution. These requirements provide the foundations for the proposed solution, encompassing both the user interface and the technical aspect of the system. The primary objective of these early prototypes was to visualize the search user interface. Specifically, three unique views were designed. We attached screenshots of these views in the appendix. Figure 1 shows the main search interface with publications, fields of study, and researchers. Figure 2 shows the main search interface with an expanded field of study section. Lastly, Figure 3 shows a dedicated page for researchers.

It is important to note that user input was not incorporated after prototyping due to time constraints. Nevertheless, we should evaluate the design prototypes in alignment with the defined functional and non-functional requirements.

The initial prototypes effectively addressed FR1, FR3, and NFR1. Upon entering a search query and executing the search, the system displayed relevant publications, researchers, and fields of study. This display facilitates navigation between these entities through the underlying graph database relationships (FR1). The prototypes also presented a unified control button for sorting and filtering options to align with FR3 from the design aspect. Notably, the design of these prototypes adhered to a minimalist aesthetic, avoiding unnecessary colors and distractions, thereby aligning with NFR1.

It should be noted that fulfillment of FR3, FR4, and FR5 was not possible at this point in time. The changing dataset, ambiguity in certain requirements, and uncertainties regarding technical feasibility were impeding factors in the direct handling of these requirements. The other functional and non-functional requirements, such as FR2, NFR2, NFR3, and NFR4, will be addressed in subsequent stages of development.

The design of the prototypes in this study drew inspiration from existing search engines, such as the layout of featured snippets employed by Google Search [Loc] and the publication list format, reminiscent of Semantic Search's design. This design decision was intended to provide a sense of familiarity and ease of use for new users. The simple design ultimately aimed to lower the learning curve associated with our application.

The design prototypes are not intended to represent the final user interface. Rather, we still integrated wireframe-like components in the prototypes, such as buttons and information displays. These components can facilitate design flexibility. This approach allows us to avoid commitment to a single and final design direction and to adapt the design later according to our styling technology capabilities.

5.3 First Iteration

This iteration started the technical implementation of the web application. During this iteration, we focused on the user interface, postponing the search engine implementation for the subsequent iteration. The transition between this phase and the subsequent iteration is marked by the user evaluation round. The first iteration spanned approximately one and a half months,

5.3.1 Development

The initial iteration encompassed the following notable development milestones:

- **Docker Implementation:** The application was instantiated through Docker, enhancing portability, consistency, and scalability. This architectural choice ensured that the application could be seamlessly executed on any computer without necessitating configuration alterations.

- **Syntactic search:** As previously mentioned, the semantic search engine will be implemented in the subsequent iteration. In its place, a syntactic search feature was implemented using Neo4j with simple string matching to filter entities whose names contain the query string precisely. This functionality enabled searches for scholarly Publications, Researchers, and Fields of Study.

- **Implemented the search page:** Users can now access the search page, enter their search query, and receive results as outlined in the previous point. This implementation featured dynamic updates without necessitating page reloads for a more seamless user experience.

- **Sort and filter options:** We implemented a filter for publications and sort for both publications and researchers were implemented to complement the search feature. Users could sort Publications based on their number of citations or recency, while Researchers could be sorted based on their number of publications, citations, or h-index. Additionally, users can filter publications by one or more fields of study. The system only displays publications associated with all selected fields of study.
- **Implemented the publication page:** A publication page was created to show more details, notably the related fields of study, researchers, citing papers, and referenced publications, as outlined by the underlying graph dataset.
- **Implemented the researcher page:** A researcher page was created to show more details, notably their publications and other researchers that coauthored at least one publication, as outlined by the underlying graph dataset.
- **Implemented the field of study page:** A field of study page was created to show more details, notably related publications, associated sub-fields, and parent fields of study, as outlined by the underlying graph dataset.
- **Static site generation at build-time:** During the production build process, static versions of pages for Publications, Researchers, and Fields of Study were generated.
- **Implemented fields of study hierarchy visualization:** A visualization of the field of study hierarchy was implemented to show the connections between fields of study. This component was visible in the field of study view and publication view.

5.3.2 User Evaluation

After the implementation in the first iteration, user evaluation was conducted to assess essential features such as navigational pages and keyword-matching search. This evaluation round brought critical insights to light from the perspectives of usability, design, and requirements completion.

- **Unclear User Interface:** Some issues have been brought up regarding the display of information and clarity of controls. There were instances where the purpose

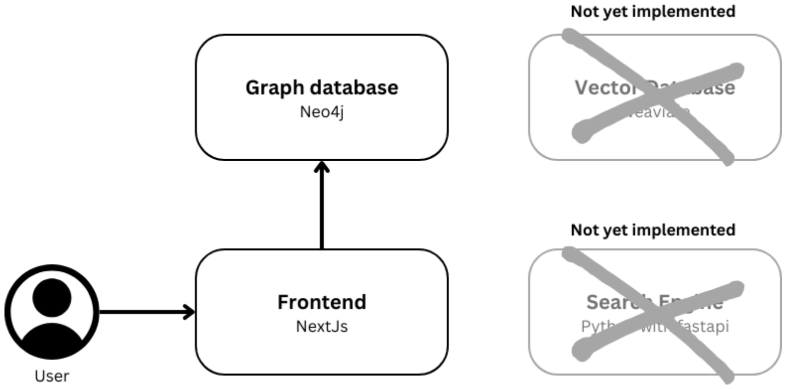


Figure 5.1: The implemented parts of our intended software architecture

of certain numerical values, text, and graphical elements was not easily comprehensible to users. Additionally, the functionality of a few interface elements, including buttons, hyperlinks, and sort/filter options, was not intuitive for some users. These issues require the integration of tooltips for select elements and a select UI redesign to improve the user experience.

- **Slow Responsiveness:** Users reported suboptimal responsiveness during search and navigation, attributed primarily to suboptimal data retrieval and graph data storage. To address this issue, plans were outlined to optimize graph connections and data retrieval, sometimes extracting the data into properties of an entity type. For other processes with unavoidable time delay, UI component placeholders and shimmer effects will be implemented for immediate feedback, for example, during the search.
- **Incomplete and Inaccurate Search:** A user reported the current limitations with search functionality reduced usability. Pagination was not yet supported, which is a crucial component for a complete search feature. The search results were also inaccurate and unsatisfactory due to the overly simplistic search feature, failing to capture users' nuanced queries.

In spite of these limitations, the application was well-received. The minimalist user interface was commended for delivering an overall user experience on par with existing exploratory tools. Most controls, including buttons and links, were found to offer clear signifiers for their functions. Moreover, the relations between publications, researchers, and fields of study were deemed useful for exploratory search. The application offered a promising exploratory search solution, though incomplete and required further development to achieve production readiness.

5.4 Second Iteration

Following the first iteration, our subsequent iteration, spanning approximately six weeks, was defined by the focus on addressing critical issues concerning usability, performance enhancement, and the integration of semantic search capabilities. During the second iteration, the dataset was expanded with another scholarly entity, "Venue." Another objective in this iteration was the addition of a dedicated page for the venue entity, enabling more options for exploration.

5.4.1 Development

An emphasis on semantic search integration marked this phase of the project. We undertook several significant efforts, notably:

- **Migration to Leibniz Supercomputing Center:** To accommodate the computational demands of the required language models, our application moved from a local environment to a virtual machine at the Leibniz supercomputing center equipped with 100GB disk space, 20 vCPUs, and 90GB RAM. This transition occurred smoothly due to our previous Dockerization of the application.

- **Weaviate Integration:** The Weaviate vector database was introduced to support hybrid search functionality. Weaviate was orchestrated as a Docker container, with data and embeddings sourced from the Neo4j database through a Python server.

- **Python-based Search Engine:** A Python service was set up to provide REST API endpoints. FastAPI was selected as our framework, motivated by its performance. It begins with support for transferring publications from Neo4j to Weaviate. To tackle problems related to performance and responsiveness, we implemented a background task to synchronize the data, ensuring that only a manageable amount of batches of publications were queried from Neo4j and sent to Weaviate. Furthermore, we developed a query mechanism to access publications from Weaviate through its hybrid search API. We experimented with several configurations, ultimately settling on a retrieval limit of one thousand publications with an emphasis on BM25 (syntactic search) over HNSW (semantic search) to support the conventional emphasis on keyword utilization in exploratory search. Additionally, we optimized the reranking process by caching and limiting the number of publications to enhance speed, potentially sacrificing accuracy in the trade-off.

  Pagination Implementation: A limited pagination feature was implemented to support exploring more search results. Due to time constraints, the pagination was limited to publication searches, as this pagination served as the most important search feature. Although incomplete, our implementation proved that the features can be expanded further.

- **Graph Connection Preprocessing** Calculating some information about an entity based on its graph connections at run-time costs a big computational power. So, upon adding new data, we extracted information such as the h-index for researchers.
- **Implemented venue page:** A venue page was created to show more details, notably the associated publications, as outlined by the underlying graph dataset.
- **Replaced fields of study hierarchy visualization:**

  Due to the previous usability shortcomings, we replaced the implementation of our hierarchy visualization with a new implementation with a scrollable and dragable interface. Furthermore, a page is implemented that shows the entire hierarchy and enables exploration of all fields of study.
- **Tooltip Integration:** More tooltips were implemented to improve usability and clarify the significance and purpose of UI elements.
- **UI Component Skeletons and Shimmer Effect:** Skeleton of UI components and shimmer effect were implemented to provide immediate feedback upon searching. These visual cues helped maintain system responsiveness before search data was actually retrieved.

At this point, we have successfully incorporated all essential features of an exploratory search. The architecture of our application, as outlined in figure 4.2 was implemented. While there is further room for improvement, this second iteration has achieved its objectives.

5.4.2 User Evaluation

Upon completion of the implementation phase, a holistic final user evaluation was conducted to assess the usability and overall performance of the exploratory search system. Due to the essential features having been implemented, the feedback collection was more expansive and could touch on all aspects of the web application, contrary to the limited features at the time of the first user evaluation.

During the interviews, we investigated how users perceived our proposed solutions compared to the existing solutions. The responses were characterized by expressions of interest, with statements similar to "I will try it on my own" and "I might use it on occasion." Nevertheless, a clear endorsement was non-existent, primarily due to the dominance of existing tools on the internet. Furthermore, there exist inherent limitations associated with new users unfamiliar with the system, making it difficult to collect user feedback that resulted from long-time consideration.

Nevertheless, several noteworthy observations emerged from the user feedback. Users expressed satisfaction with the system's navigation and its performance. Moreover, the system facilitated fields of study hierarchy traversal, even in the absence of well-defined search goals.

However, the users weren't satisfied with the system's sub-optimal performance in recommending relevant search results. Despite the newly integrated search engine for publications, users' nuanced queries often remained ineffectively captured. Additionally, the search for researchers and fields of study remained limited to keyword-based queries, forcing users to know precise keywords. Furthermore, issues in the database were identified, with some researcher entities exhibiting duplication or incomplete names. This issue further worsened the user experience of the search feature.

There are some additional insights, such as:

- **Desire for Information Completion:** Some users expressed interest in the explanation and information of various controls and data elements within the application. Clarity and comprehensiveness are relevant for researchers with an interest in how the re-ranker operates, the specific functionalities of different components, and the precise scope of our dataset.

- **Database limitations:** An interviewee reported that the existing database is too restrictive for his use case. Although the thesis enabled an exploratory search for scholarly domains in NLP, researchers often require interdisciplinary research, as exemplified by this interviewee's work in the domain of social computing. Although the expansion of the database is out of this thesis's scope, it is important to note that the lack of coverage requires another search engine in addition to our solution in their workflow.

- **Enhanced Feature Set:** Interviewees exhibited enthusiasm for the application but emphasized that for it to supersede existing solutions, it must offer a substantial improvement in terms of functionality and quality-of-life features. This includes the integration of features already present in alternative solutions and other quality-of-life features.

- **Identification of New Bugs:** Users identified several bugs, such as occasional failure to load citing papers. These issues detract from the overall user experience and should be addressed in future iterations to enhance the application's reliability.

Our graph connections demonstrated an improvement over existing methods. On the other hand, the search feature has failed to deliver a robust solution for finding scholarly entities. This deficiency and other previously mentioned points necessitate further development efforts to enhance usability.

At this point, the user interfaces were fully implemented, as exemplified by Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, and Figure 11 in the appendix.

6 Discussion

In this chapter, we will present an in-depth analysis of iterative development and its accompanying user evaluation. The discussion will center around their alignment with the research problem and objectives, highlighting accomplishments, shortcomings, and potential future works for scholarly development.

The primary objective of this thesis was to address the challenges encountered in the realm of exploratory search for scholarly entities within the field of Natural Language Processing. To guide our research, we formulated research questions accordingly. The approach involved an initial investigation into existing solutions, the identification of pain points experienced by users, and the subsequent development of a solution in two iterative phases. The proposed solution comprises two key components: the frontend interface and the search engine. Following the completion of these iterations, we conducted a thorough evaluation of the resulting application, soliciting user feedback.

The methodology of this study adhered to the principles of user-centric design, with a particular emphasis on usability and user experience. To this end, we undertook an initial user study, created user personas to represent our target audience, and pursued iterative development and evaluation. Notably, a quantification effort to asses usability was introduced through the system usability scale.

Now that the web application has been implemented, we proceed to dissect its merits and demerits. This examination is structured to first discuss the application in relation to the artifacts generated during this thesis. Subsequently, we will delve into what worked well, what did not, and the overarching limitations encountered.

6.1 Requirements Review

We evaluate our application against the objectives defined by both functional and non-functional requirements.

Functional Requirements:

- **FR1 (Navigational Entities):** We successfully supported entity navigation by implementing a unique detailed entity page with links to connect the entities with one another.

- **FR2 (Semantic Search for Publications):** A search engine with semantic capabilities was implemented. User satisfaction stands as an indicator of success due to the lack of metrics and historical data as a benchmark to assess the search engine in a quantifiable manner. It is noteworthy that users expressed dissatisfaction with exploratory searches, but known-item searches with exact queries yielded favorable results.

- **FR3 (Sort and Filter options):** The intended sort and filter options have been implemented and are accessible to users.

- **FR4 (Fields of Study Hierarchy):** A dedicated page displays the field of study hierarchy, allowing users to expand and collapse nodes for easy navigation to relevant field of study pages.

- **FR5 (Information Completeness):** The web application effectively presents relevant information, encompassing most of the entity properties from the dataset.

Non Functional Requirements:

- **NFR1 (Minimalist user interface):** Our application embodies a minimalist user interface with limited use of colors, great contrasts, and clearly visible components.

- **NFR2 (Optimal Response Time):** The search feature of our application performed at speed faster than Semantic Search and slower than Google Search. Meanwhile, navigation within our application surpassed Semantic Search and is on par with Google Search in terms of perceivable speed.

- **NFR3 (Accurate and Consistent Search Results):** The user interface presents consistent search results. On the other hand, the accuracy falls short of users expectations.

- **NFR4 (Reliable Control and Navigation Behaviour):** Users report a high degree of understanding and consistency in the control and navigation behavior of the system.

6.2 Achievements

In this section, we highlight the accomplishments and contributions achieved in the implementation of an exploratory search system for scholarly entities in Natural Language Processing through the application of user-centric design (UCD) principles.

User-centric design principles have been successfully integrated into the development process. This approach has been fundamental in aligning our efforts closely with the users' needs to emphasize aspects of user experience and usability. For instance, user feedback prompted us to replace and improve the hierarchical tree structure in our system. This modification not only addressed user concerns but also facilitated a more seamless and user-friendly experience. The hierarchical tree structure, in particular, has notably enhanced the quality of life for NLP researchers, as it permits efficient exploration without the need for prior knowledge.

Another significant achievement is the enhanced navigation experience powered by graph connections and static site generation. These two aspects contributed to efficient information exploration and navigation, respectively. Furthermore, the adoption of a modern, minimalist user interface has positively impacted the user experience. This interface design not only aligns with contemporary design standards but also integrates an array of intuitive controls and navigational links, enhancing the user's ability to interact with the system effectively.

Our study contributes to exploratory search in the Natural Language Processing domain by its examination of graph connection navigation and a brief exploration of alternative search engines on user experience. Additionally, we have shown that the graph connection navigation can support exploratory search without the need for prior knowledge. However, more research needs to be conducted to provide a better search engine solution.

6.3 Challenges and limitations

This thesis has encountered a series of challenges and limitations during its course of investigation. First among these limitations was the constraint imposed by the time scope of four months for this study. With only two iterations, the feature development and integration were inherently restricted, resulting in a non-production-ready solution. In the context of an iterative user-centric design, continuous development over multiple iterations is integral for continuous system enhancement. Moreover, user-centric design practices commonly necessitate the involvement of an interdisciplinary team to foster diverse perspectives, which this thesis failed to achieve.

In the conduct of this study, certain inherent limitations deserve attention. Firstly, it is plausible that participant selection introduced a potential bias. Participants were volunteers who might have already expressed an interest in an alternative solution. This personal interest, in turn, might have influenced their feedback on usability. Additionally, the participants had never used the application before, potentially emphasizing their initial impression of the platform rather than a comprehensive evaluation of its functionalities. This limitation may account for the consistency in reported System Usability Scale scores across both development iterations.

Our evaluation of the search engine is limited to user satisfaction and impression and not exact accuracy due to the absence of historical search data, a feature available to existing tools such as Semantic Scholar [Fel].

6.4 Future Work

This study has established the viability of a user-centric design framework for the development of an exploratory search system for scholarly entities in Natural Language Processing. Partly due to the user involvement, we discovered several possible aspects for further enhancement.

A vital element of the exploratory search system is its search engine. Presently, our search engine integrated Weaviate's hybrid search and a re-ranker created by Semantic Scholar. However, the search engine has proved to be short of satisfactory for exploratory search. This deficiency necessitates continuous research and development to improve both precision and performance within the NLP domain.

The graph database is another vital element alongside the search engine. This is precisely why the entities and graph connections should be given attention. The database currently hosts minor inaccuracies, such as duplicated researchers and erroneous researcher entities. Additionally, an extension of the database can improve and streamline exploratory search workflow further, such as with the inclusion of research institutions and external links associated with an entity.

Several improvements can be made to the current search engine:

- **Tailored language model:** The initial language model, "SPECTER," was trained on a broad scientific corpus, potentially impeding its ability to retrieve NLP-specific papers. Therefore, research is needed to develop a language model capable of embedding title, abstract, year, authors, and venue information to optimize search results.

- **Re-ranking Algorithm:** The current state of the re-ranking algorithm requires further study. While Semantic Scholar's approach is robust in terms of accuracy, it suffers from speed limitations due to the sheer volume of papers it retrieves. Investigating alternative strategies to adapt the re-ranker's language model to NLP publications and related entities can push the state-of-the-art approaches. Additionally, consideration should be given to the feasibility of eliminating the second language model and using exclusively the embeddings created for retrieval.

- **Expanding Search Scope:** The search engine should broaden the scope of the search to encompass researchers, fields of study, and venues. This expansion necessitates the development of distinct embeddings to support nuanced semantic capabilities. Further, the incorporation of pagination, sorting, and filtering features should extend to all entities to equip users with more robust exploration features.
- **Enabling Continuous User Feedback and Data Tracking:** To enable continuous development, the usage of the application should be tracked over an extended period. Additionally, implementing mechanisms for users to provide feedback is essential to ensure that their evolving needs and expectations are met.

Additionally, continuous improvements in usability should be pursued through information clarity and completeness, implementation of more robust features, and support for additional screen sizes. This thesis presents many possibilities for future work, a characteristic of continuous development of an iterative user-centric design, proving its effectiveness.

7 Conclusion

In this bachelor's thesis, we have undertaken the task of designing, implementing, and evaluating an exploratory search system for scholarly entities in Natural Language Processing following the user-centric design framework. Our primary focus has been to enhance the overall usability of exploratory search systems within the NLP domain with the integration of graph entities and connections.

The study started with an examination of existing approaches through both independent research and an initial user study. This comprehensive analysis provided a valuable foundation, including identification of the users' pain points with the current solutions. The initial user study also generated notable artifacts, such as user persona and user requirements. Building upon these insights, we started the design and development process in the hope of addressing these identified user challenges. Then, the proposed solution was implemented in two iterations, each consisting of a development and an evaluation phase.

This research holds significance in the cross-section of user-centric design for the development of an exploratory search system that tackles Natural Language Processing research problems. It enriches the scholarly domain with its contribution to an area with a low number of scholarly publications. More notably, it opens a way forward for further development of such scholarly entities in Natural Language Processing.

Our findings demonstrated that graph connections with support of additional entities such as fields of study effectively facilitated navigation, even for those lacking prior extensive knowledge. However, the user interviews indicated the necessity for further research to optimize result accuracy.

Improvements upon existing exploratory search solutions, as exemplified by our proposed solution, stand to substantially benefit researchers and the broader scholarly community. By reducing barriers to scientific exploratory search and enhancing workflow efficiencies, these tools represent valuable contributions. Nevertheless, we acknowledge that further improvements and robust features are imperative to convincingly position this system as the primary tool of choice for our target audience. This work aspires to serve as a solid foundation for subsequent research and development endeavors aimed at similar systems.

Abbreviations

TUM Technical University of Munich

NLP Natural Language Processing

UCD User-Centric Design

System Usability Scale

Scale

1. Strongly Disagree
2. Moderately Disagree
3. Neither Agree nor Disagree
4. Moderately Agree
5. Strongly Agree

The results ranges from A+ to F.

First Iteration Results

		Interviewee		
#	Statements	1	2	3
1	I think that I would like to use this system frequently.	3	4	5
2	I found the system unnecessarily complex.	1	1	1
3	I thought the system was easy to use.	4	5	4
4	I think that I would need the support of a technical person to be able to use this system.	1	1	1
5	I found the various functions in this system were well integrated.	3	5	4
6	I thought there was too much inconsistency in this system.	3	3	2
7	I would imagine that most people would learn to use this system very quickly.	2	5	4
8	I found the system very cumbersome to use.	4	1	4
9	I felt very confident using the system.	5	4	2
10	I needed to learn a lot of things before I could get going with this system.	3	1	3
	Results	62.5	90	82.5

Table 1: The system usability scale results from the first user evaluation.

The results are D(OK), A+(Best), and A(Excellent) respectively.

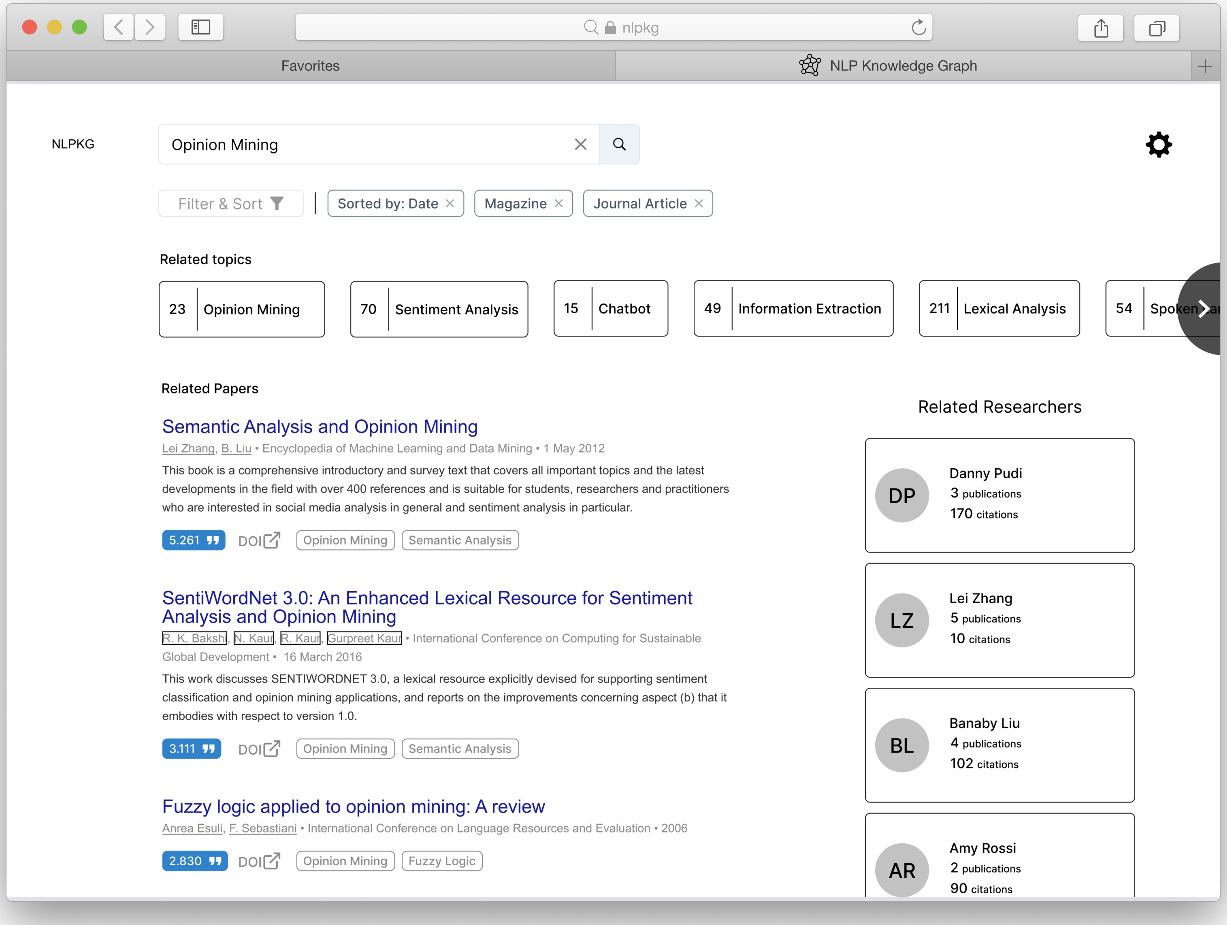
Second Iteration Results

		Interviewee				
#	Statements	4	5	6	7	8
1	I think that I would like to use this system frequently.	3	4	4	5	4
2	I found the system unnecessarily complex.	2	1	2	2	1
3	I thought the system was easy to use.	5	4	5	4	4
4	I think that I would need the support of a technical person to be able to use this system.	3	1	1	1	1
5	I found the various functions in this system were well integrated.	4	5	2	5	4
6	I thought there was too much inconsistency in this system.	1	1	2	1	1
7	I would imagine that most people would learn to use this system very quickly.	4	5	4	4	5
8	I found the system very cumbersome to use.	1	1	2	2	1
9	I felt very confident using the system.	3	4	5	4	5
10	I needed to learn a lot of things before I could get going with this system.	2	1	1	1	1
	Results	75	92.5	80	87.5	92.5

Table 2: The system usability scale results from the second user evaluation.

The results are B(Good), A+(Best), A- ,A+ (Best) and A+ (Best) respectively.

Images



Opinion Mining

Filter & Sort | Sorted by: Date × | Magazine × | Journal Article ×

Related topics

23 Opinion Mining | 70 Sentiment Analysis | 15 Chatbot | 49 Information Extraction | 211 Lexical Analysis | 54 Spoken

Related Papers

Semantic Analysis and Opinion Mining
Lei Zhang, B. Liu • Encyclopedia of Machine Learning and Data Mining • 1 May 2012

This book is a comprehensive introductory and survey text that covers all important topics and the latest developments in the field with over 400 references and is suitable for students, researchers and practitioners who are interested in social media analysis in general and sentiment analysis in particular.

5.261 ❞ DOI Opinion Mining Semantic Analysis

SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining
R. K. Bakshi, N. Kaur, R. Kaur, Gurpreet Kaur • International Conference on Computing for Sustainable Global Development • 16 March 2016

This work discusses SENTIWORDNET 3.0, a lexical resource explicitly devised for supporting sentiment classification and opinion mining applications, and reports on the improvements concerning aspect (b) that it embodies with respect to version 1.0.

3.111 ❞ DOI Opinion Mining Semantic Analysis

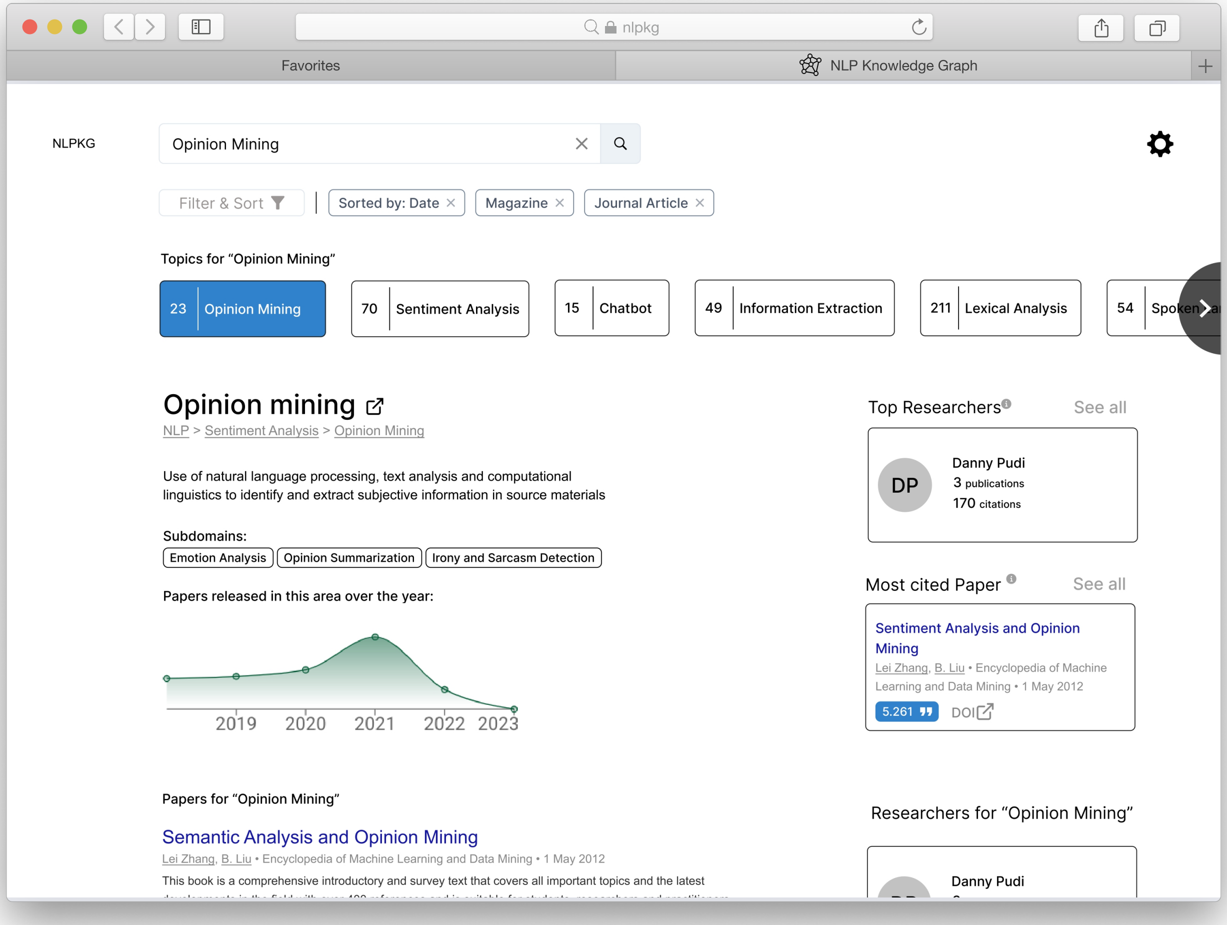
Fuzzy logic applied to opinion mining: A review
Anrea Esuli, F. Sebastiani • International Conference on Language Resources and Evaluation • 2006

2.830 ❞ DOI Opinion Mining Fuzzy Logic

Related Researchers

DP	Danny Pudi	3 publications	170 citations
LZ	Lei Zhang	5 publications	10 citations
BL	Banaby Liu	4 publications	102 citations
AR	Amy Rossi	2 publications	90 citations

Figure 1: Figma Prototype: Main Search



Opinion Mining

Sorted by: Date × | Magazine × | Journal Article ×

Topics for "Opinion Mining"

23 Opinion Mining | 70 Sentiment Analysis | 15 Chatbot | 49 Information Extraction | 211 Lexical Analysis | 54 Spoken

Opinion mining

NLP > Sentiment Analysis > Opinion Mining

Use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials

Subdomains:

Emotion Analysis | Opinion Summarization | Irony and Sarcasm Detection

Papers released in this area over the year:

Top Researchers

DP Danny Pudi
3 publications
170 citations

Most cited Paper

Sentiment Analysis and Opinion Mining
Lei Zhang, B. Liu • Encyclopedia of Machine Learning and Data Mining • 1 May 2012
5.261

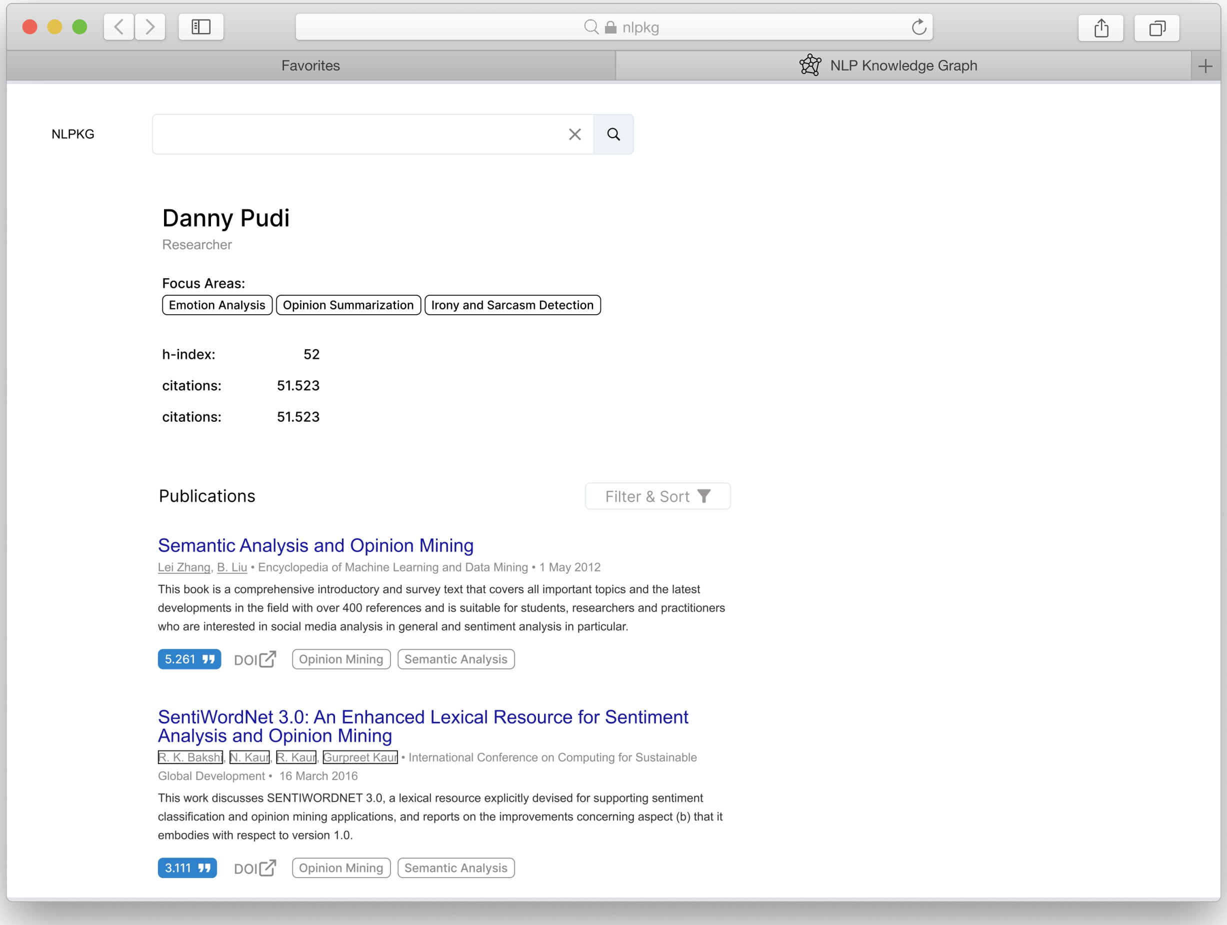
Papers for "Opinion Mining"

Semantic Analysis and Opinion Mining
Lei Zhang, B. Liu • Encyclopedia of Machine Learning and Data Mining • 1 May 2012
This book is a comprehensive introductory and survey text that covers all important topics and the latest developments in the field with over 400 references and is suitable for students, researchers and practitioners

Researchers for "Opinion Mining"

Danny Pudi

Figure 2: Figma Prototype: Field of Study is Expanded



Danny Pudi

Researcher

Focus Areas:

Emotion Analysis | Opinion Summarization | Irony and Sarcasm Detection

h-index:	52
citations:	51.523
citations:	51.523

Publications

Semantic Analysis and Opinion Mining

Lei Zhang, B. Liu • Encyclopedia of Machine Learning and Data Mining • 1 May 2012

This book is a comprehensive introductory and survey text that covers all important topics and the latest developments in the field with over 400 references and is suitable for students, researchers and practitioners who are interested in social media analysis in general and sentiment analysis in particular.

5.261 ❞ DOI Opinion Mining | Semantic Analysis

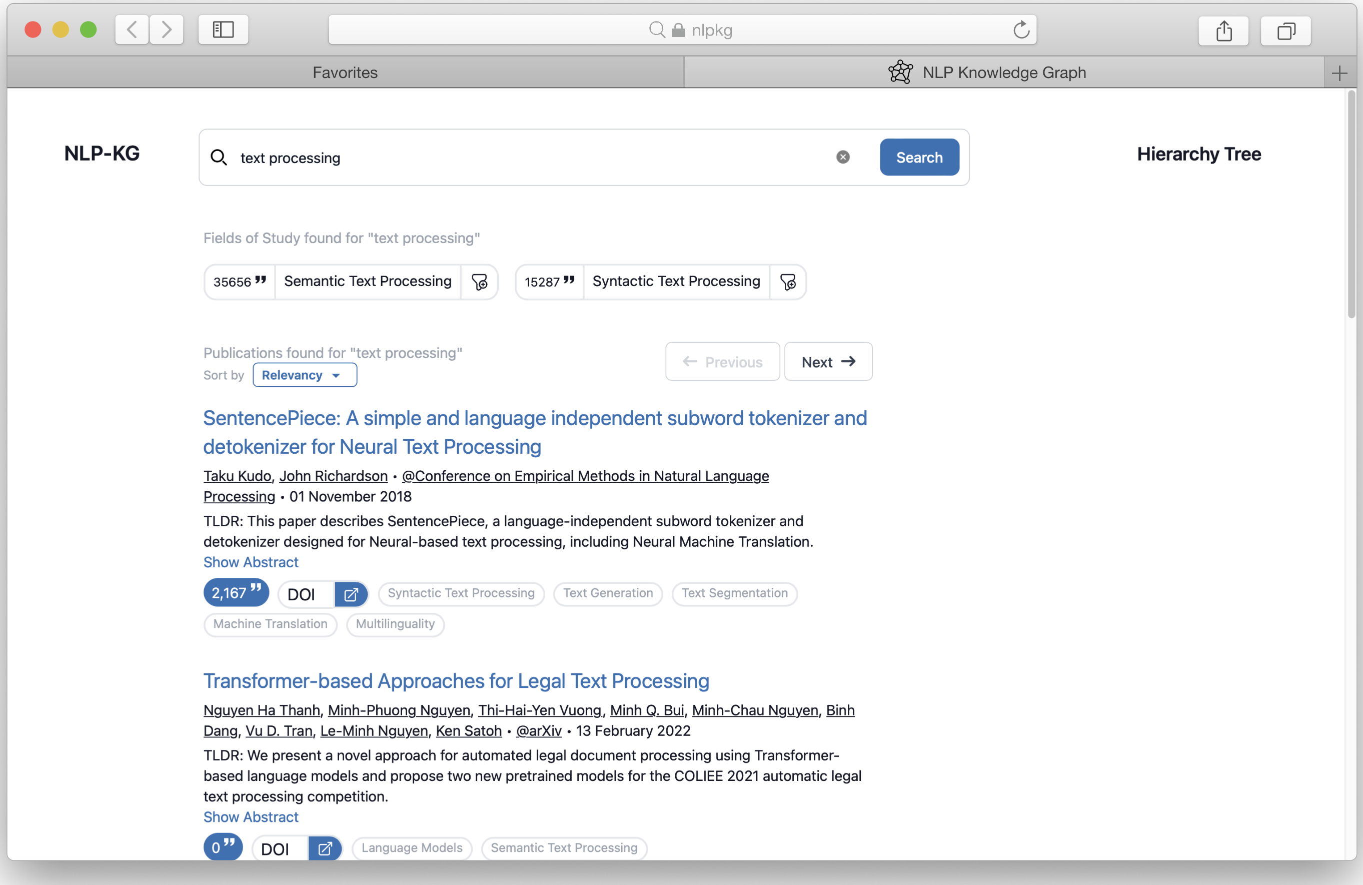
SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining

R. K. Bakshi, N. Kaur, R. Kaur, Gurpreet Kaur • International Conference on Computing for Sustainable Global Development • 16 March 2016

This work discusses SENTIWORDNET 3.0, a lexical resource explicitly devised for supporting sentiment classification and opinion mining applications, and reports on the improvements concerning aspect (b) that it embodies with respect to version 1.0.

3.111 ❞ DOI Opinion Mining | Semantic Analysis

Figure 3: Figma Prototype: Researcher's Page



NLP-KG

text processing

Search

Hierarchy Tree

Fields of Study found for "text processing"

35656 ” Semantic Text Processing

15287 ” Syntactic Text Processing

Publications found for "text processing"

Sort by Relevancy

← Previous

Next →

SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing

Taku Kudo, John Richardson • @Conference on Empirical Methods in Natural Language Processing • 01 November 2018

TLDR: This paper describes SentencePiece, a language-independent subword tokenizer and detokenizer designed for Neural-based text processing, including Neural Machine Translation.

Show Abstract

2,167 ” DOI Syntactic Text Processing Text Generation Text Segmentation Machine Translation Multilinguality

Transformer-based Approaches for Legal Text Processing

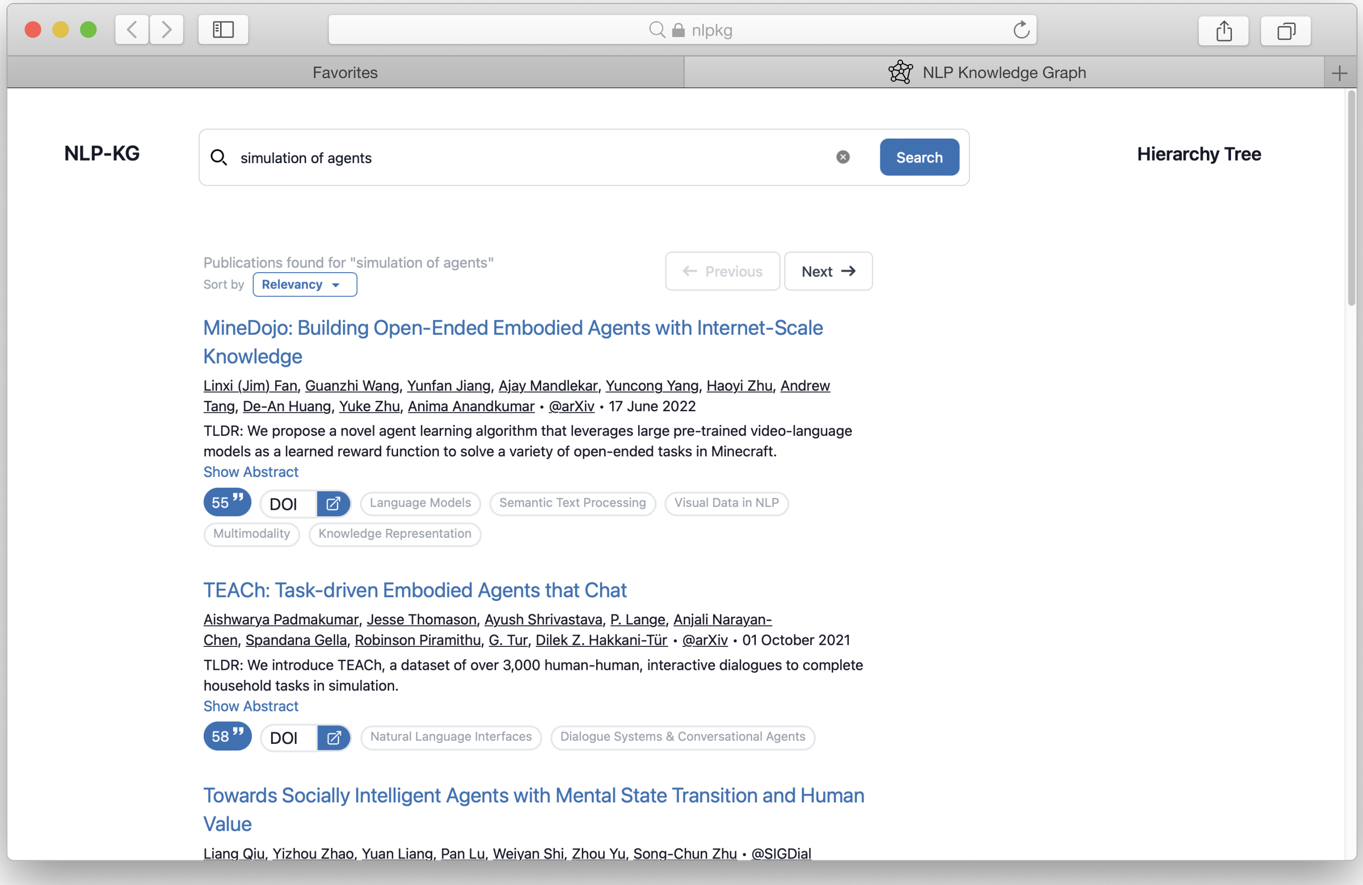
Nguyen Ha Thanh, Minh-Phuong Nguyen, Thi-Hai-Yen Vuong, Minh Q. Bui, Minh-Chau Nguyen, Binh Dang, Vu D. Tran, Le-Minh Nguyen, Ken Satoh • @arXiv • 13 February 2022

TLDR: We present a novel approach for automated legal document processing using Transformer-based language models and propose two new pretrained models for the COLIEE 2021 automatic legal text processing competition.

Show Abstract

0 ” DOI Language Models Semantic Text Processing

Figure 4: Frontend Result: The search for query "text processing" returns relevant fields of study and publications



simulation of agents

Publications found for "simulation of agents"

Sort by Relevancy

MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge

Linxi (Jim) Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, Anima Anandkumar • @arXiv • 17 June 2022

TLDR: We propose a novel agent learning algorithm that leverages large pre-trained video-language models as a learned reward function to solve a variety of open-ended tasks in Minecraft.

Show Abstract

55 ” DOI Language Models Semantic Text Processing Visual Data in NLP Multimodality Knowledge Representation

TEACh: Task-driven Embodied Agents that Chat

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, P. Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramithu, G. Tur, Dilek Z. Hakkani-Tür • @arXiv • 01 October 2021

TLDR: We introduce TEACh, a dataset of over 3,000 human-human, interactive dialogues to complete household tasks in simulation.

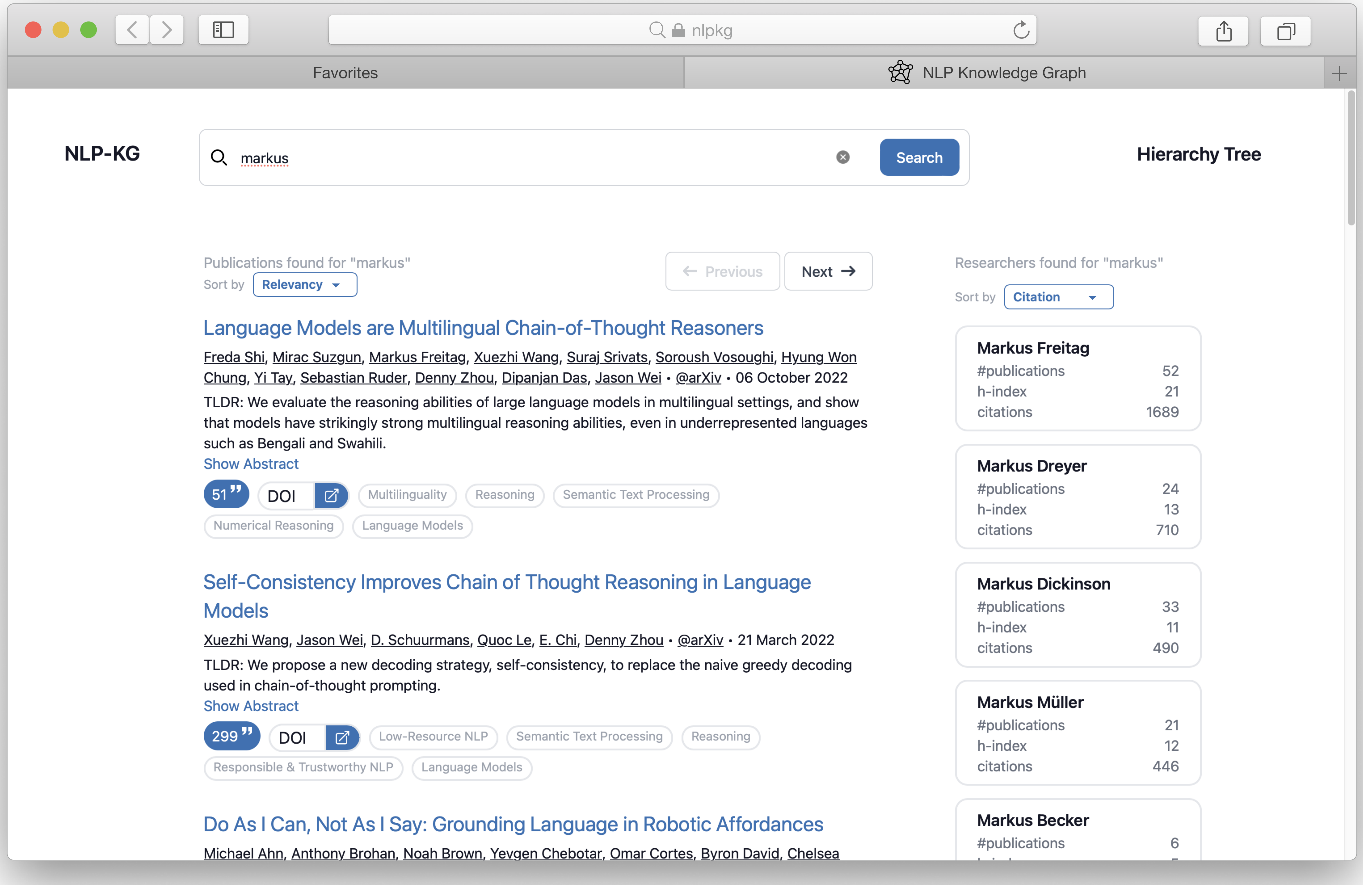
Show Abstract

58 ” DOI Natural Language Interfaces Dialogue Systems & Conversational Agents

Towards Socially Intelligent Agents with Mental State Transition and Human Value

Liang Qiu, Yizhou Zhao, Yuan Liang, Pan Lu, Weiyan Shi, Zhou Yu, Song-Chun Zhu • @SIGDial

Figure 5: Frontend Result: The search for query "simulation of agents" returns relevant publications even with semantic search



Publications found for "markus"
Sort by Relevancy

Language Models are Multilingual Chain-of-Thought Reasoners
Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, Jason Wei • @arXiv • 06 October 2022
TLDR: We evaluate the reasoning abilities of large language models in multilingual settings, and show that models have strikingly strong multilingual reasoning abilities, even in underrepresented languages such as Bengali and Swahili.
51 ” | Multilinguality | Reasoning | Semantic Text Processing | Numerical Reasoning | Language Models

Self-Consistency Improves Chain of Thought Reasoning in Language Models
Xuezhi Wang, Jason Wei, D. Schuurmans, Quoc Le, E. Chi, Denny Zhou • @arXiv • 21 March 2022
TLDR: We propose a new decoding strategy, self-consistency, to replace the naive greedy decoding used in chain-of-thought prompting.
299 ” | Low-Resource NLP | Semantic Text Processing | Reasoning | Responsible & Trustworthy NLP | Language Models

Do As I Can, Not As I Say: Grounding Language in Robotic Affordances
Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea

Researchers found for "markus"
Sort by Citation

Markus Freitag	
#publications	52
h-index	21
citations	1689

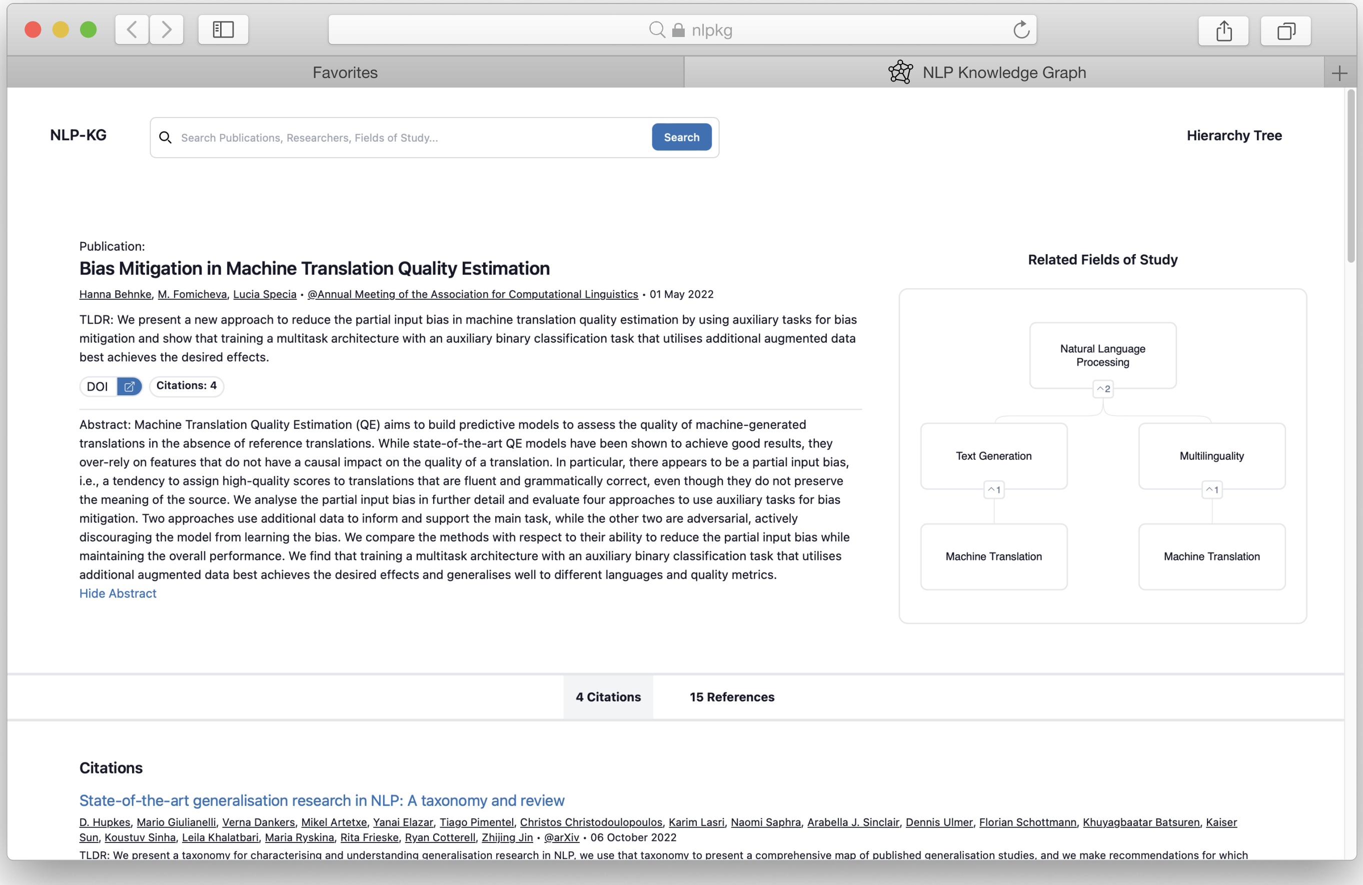
Markus Dreyer	
#publications	24
h-index	13
citations	710

Markus Dickinson	
#publications	33
h-index	11
citations	490

Markus Müller	
#publications	21
h-index	12
citations	446

Markus Becker	
#publications	6

Figure 6: Frontend Result: The search for query "markus" returns relevant publications and researchers



Publication:

Bias Mitigation in Machine Translation Quality Estimation

Hanna Behnke, M. Fomicheva, Lucia Specia • @Annual Meeting of the Association for Computational Linguistics • 01 May 2022

TLDR: We present a new approach to reduce the partial input bias in machine translation quality estimation by using auxiliary tasks for bias mitigation and show that training a multitask architecture with an auxiliary binary classification task that utilises additional augmented data best achieves the desired effects.

DOI Citations: 4

Abstract: Machine Translation Quality Estimation (QE) aims to build predictive models to assess the quality of machine-generated translations in the absence of reference translations. While state-of-the-art QE models have been shown to achieve good results, they over-rely on features that do not have a causal impact on the quality of a translation. In particular, there appears to be a partial input bias, i.e., a tendency to assign high-quality scores to translations that are fluent and grammatically correct, even though they do not preserve the meaning of the source. We analyse the partial input bias in further detail and evaluate four approaches to use auxiliary tasks for bias mitigation. Two approaches use additional data to inform and support the main task, while the other two are adversarial, actively discouraging the model from learning the bias. We compare the methods with respect to their ability to reduce the partial input bias while maintaining the overall performance. We find that training a multitask architecture with an auxiliary binary classification task that utilises additional augmented data best achieves the desired effects and generalises well to different languages and quality metrics.

Hide Abstract

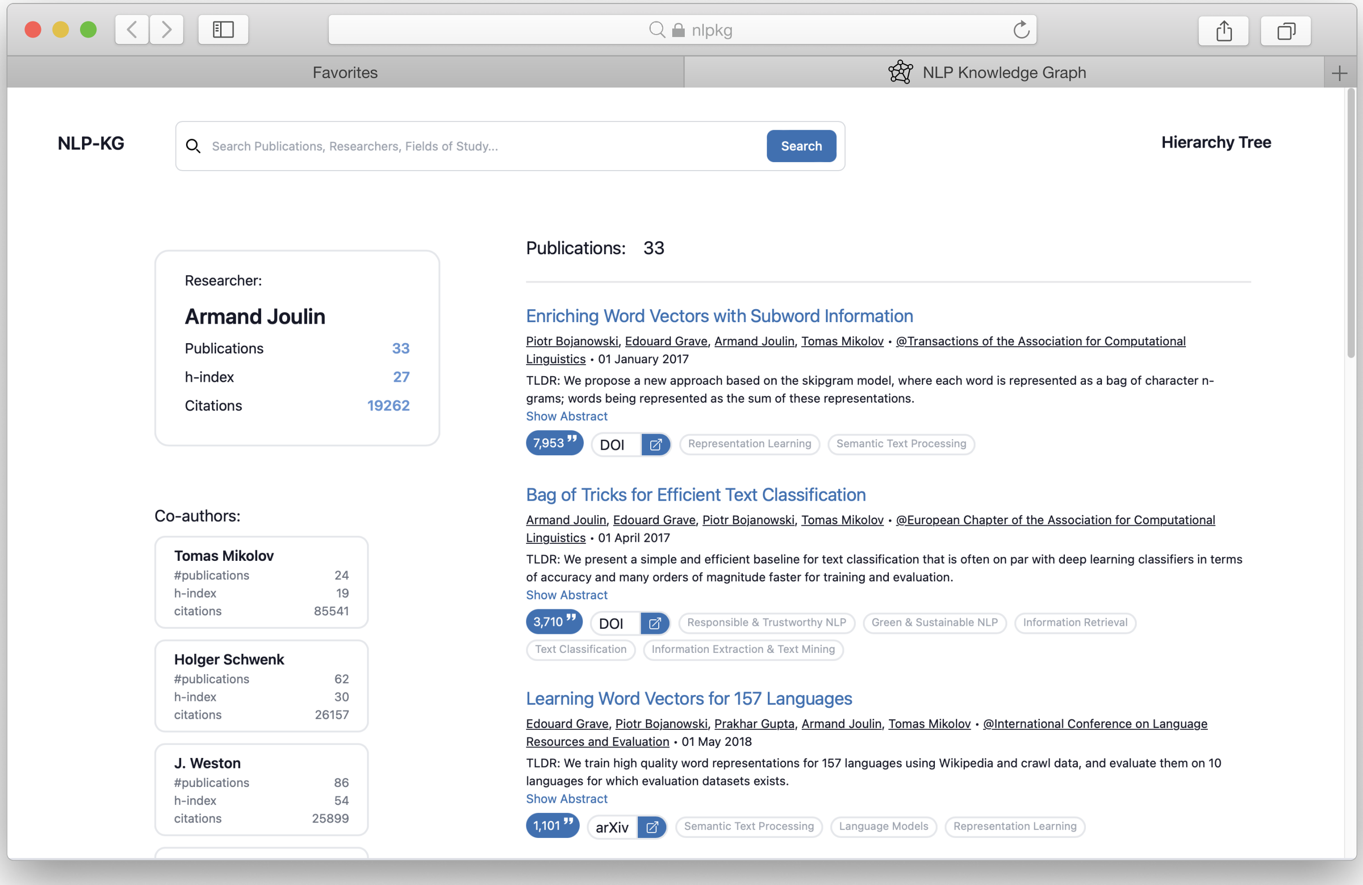
Citations

State-of-the-art generalisation research in NLP: A taxonomy and review

D. Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella J. Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, Zhijing Jin • @arXiv • 06 October 2022

TLDR: We present a taxonomy for characterising and understanding generalisation research in NLP, we use that taxonomy to present a comprehensive map of published generalisation studies, and we make recommendations for which

Figure 7: Frontend Result: The publication page for "Bias Mitigation in Machine Translation Quality Estimation"



Researcher:

Armand Joulin

Publications	33
h-index	27
Citations	19262

Co-authors:

Tomas Mikolov

#publications	24
h-index	19
citations	85541

Holger Schwenk

#publications	62
h-index	30
citations	26157

J. Weston

#publications	86
h-index	54
citations	25899

Publications: 33

Enriching Word Vectors with Subword Information

Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov • @Transactions of the Association for Computational Linguistics • 01 January 2017

TLDR: We propose a new approach based on the skipgram model, where each word is represented as a bag of character n-grams; words being represented as the sum of these representations.

Show Abstract

7,953 ” DOI Representation Learning Semantic Text Processing

Bag of Tricks for Efficient Text Classification

Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov • @European Chapter of the Association for Computational Linguistics • 01 April 2017

TLDR: We present a simple and efficient baseline for text classification that is often on par with deep learning classifiers in terms of accuracy and many orders of magnitude faster for training and evaluation.

Show Abstract

3,710 ” DOI Responsible & Trustworthy NLP Green & Sustainable NLP Information Retrieval Text Classification Information Extraction & Text Mining

Learning Word Vectors for 157 Languages

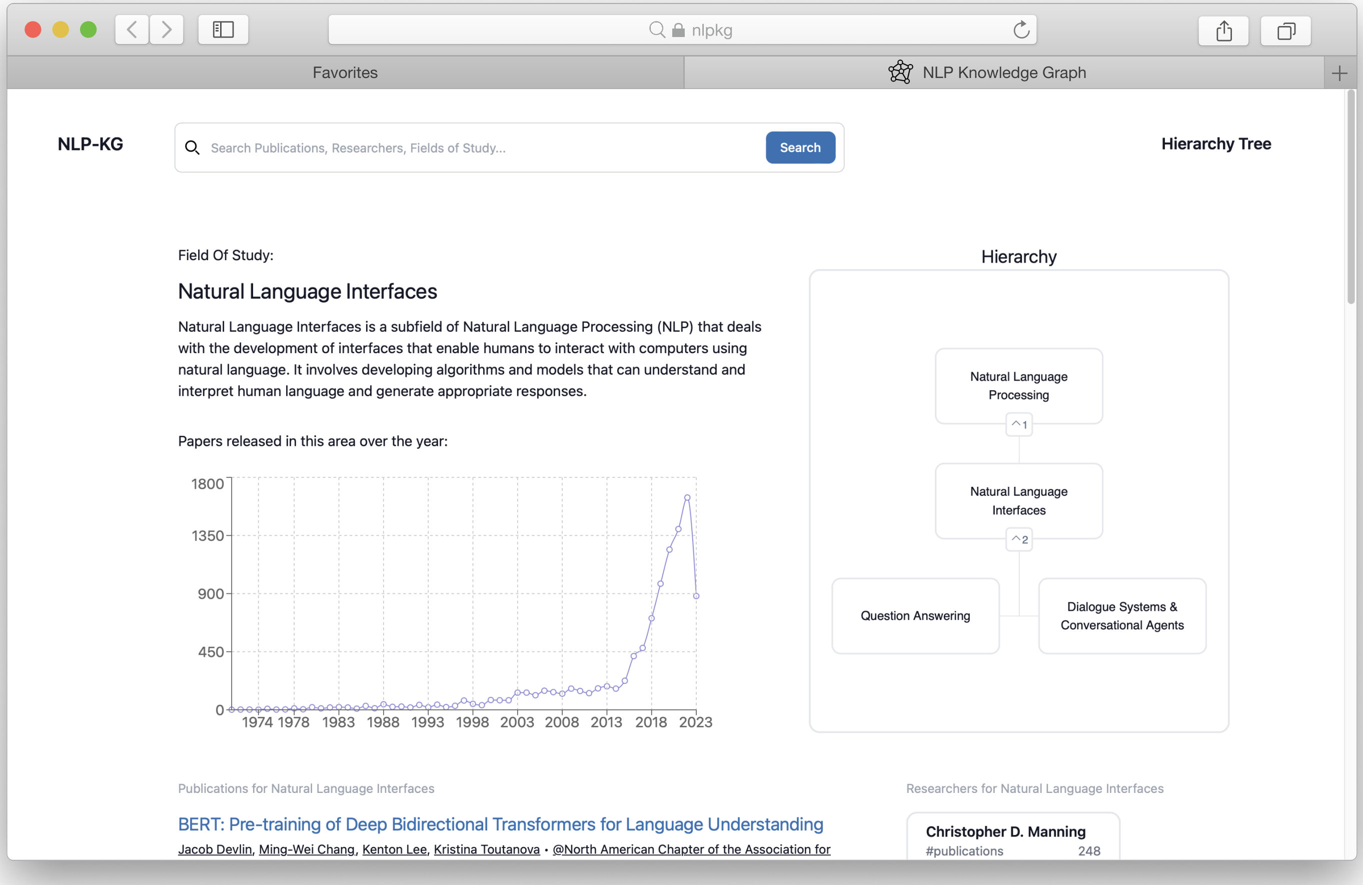
Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, Tomas Mikolov • @International Conference on Language Resources and Evaluation • 01 May 2018

TLDR: We train high quality word representations for 157 languages using Wikipedia and crawl data, and evaluate them on 10 languages for which evaluation datasets exists.

Show Abstract

1,101 ” arXiv Semantic Text Processing Language Models Representation Learning

Figure 8: Frontend Result: The researcher page for "Armand Joulin"



Search Publications, Researchers, Fields of Study...

Field Of Study:

Natural Language Interfaces

Natural Language Interfaces is a subfield of Natural Language Processing (NLP) that deals with the development of interfaces that enable humans to interact with computers using natural language. It involves developing algorithms and models that can understand and interpret human language and generate appropriate responses.

Papers released in this area over the year:

1800
1350
900
450
0
1974
1978
1983
1988
1993
1998
2003
2008
2013
2018
2023
Hierarchy
Natural Language Processing
^1
Natural Language Interfaces
^2
Question Answering
Dialogue Systems & Conversational Agents

Publications for Natural Language Interfaces

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

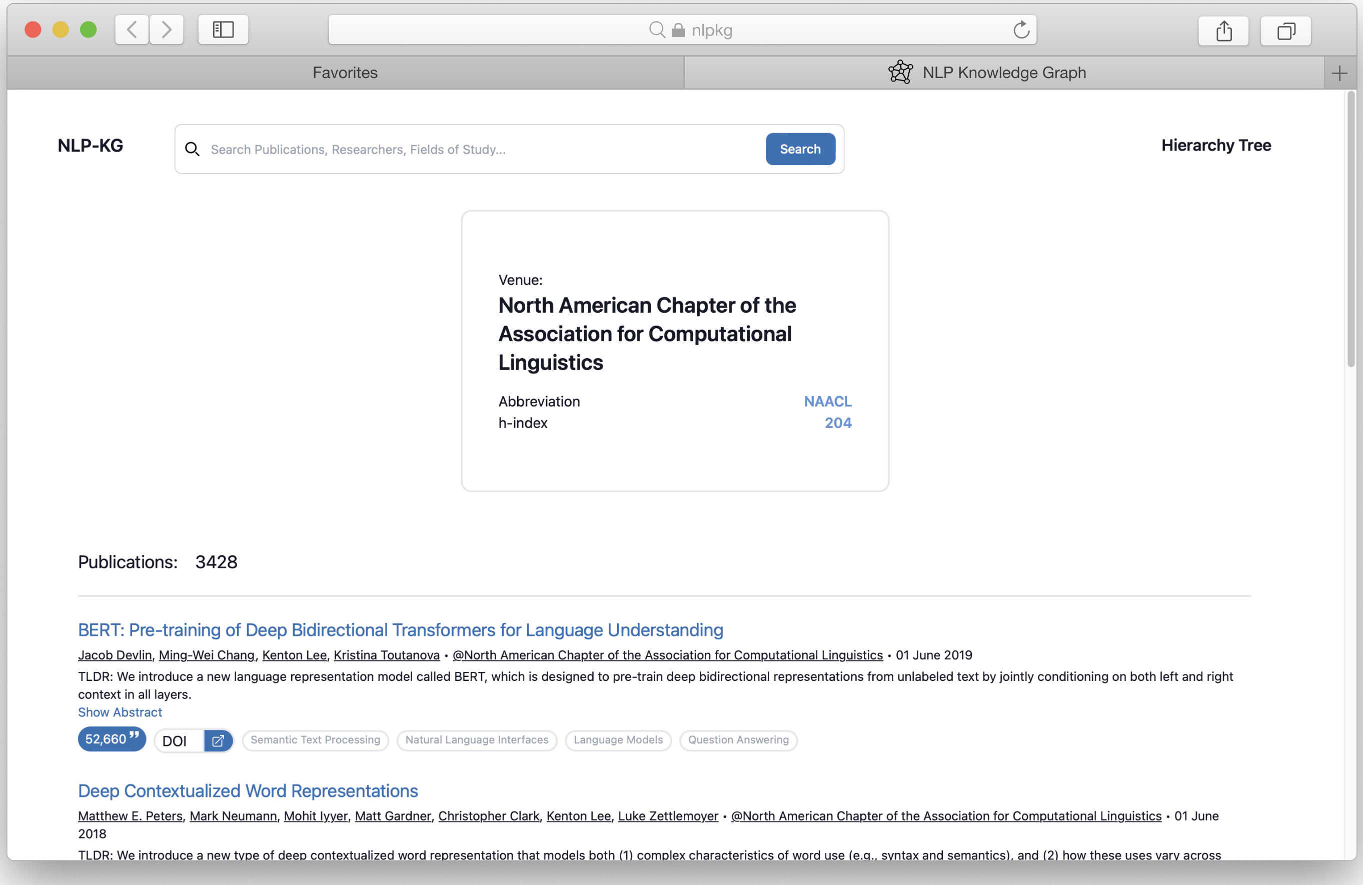
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova • @North American Chapter of the Association for

Researchers for Natural Language Interfaces

Christopher D. Manning

#publications 248

Figure 9: Frontend Result: The field of study page for "Natural Language Interfaces"



Venue:

North American Chapter of the Association for Computational Linguistics

Abbreviation NAACL

h-index 204

Publications: 3428

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova • @North American Chapter of the Association for Computational Linguistics • 01 June 2019

TLDR: We introduce a new language representation model called BERT, which is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.

Show Abstract

52,660 ” DOI Semantic Text Processing Natural Language Interfaces Language Models Question Answering

Deep Contextualized Word Representations

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer • @North American Chapter of the Association for Computational Linguistics • 01 June 2018

TLDR: We introduce a new type of deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across

Figure 10: Frontend Result: The venue page for "North American Chapter of the Association for Computational Linguistics"

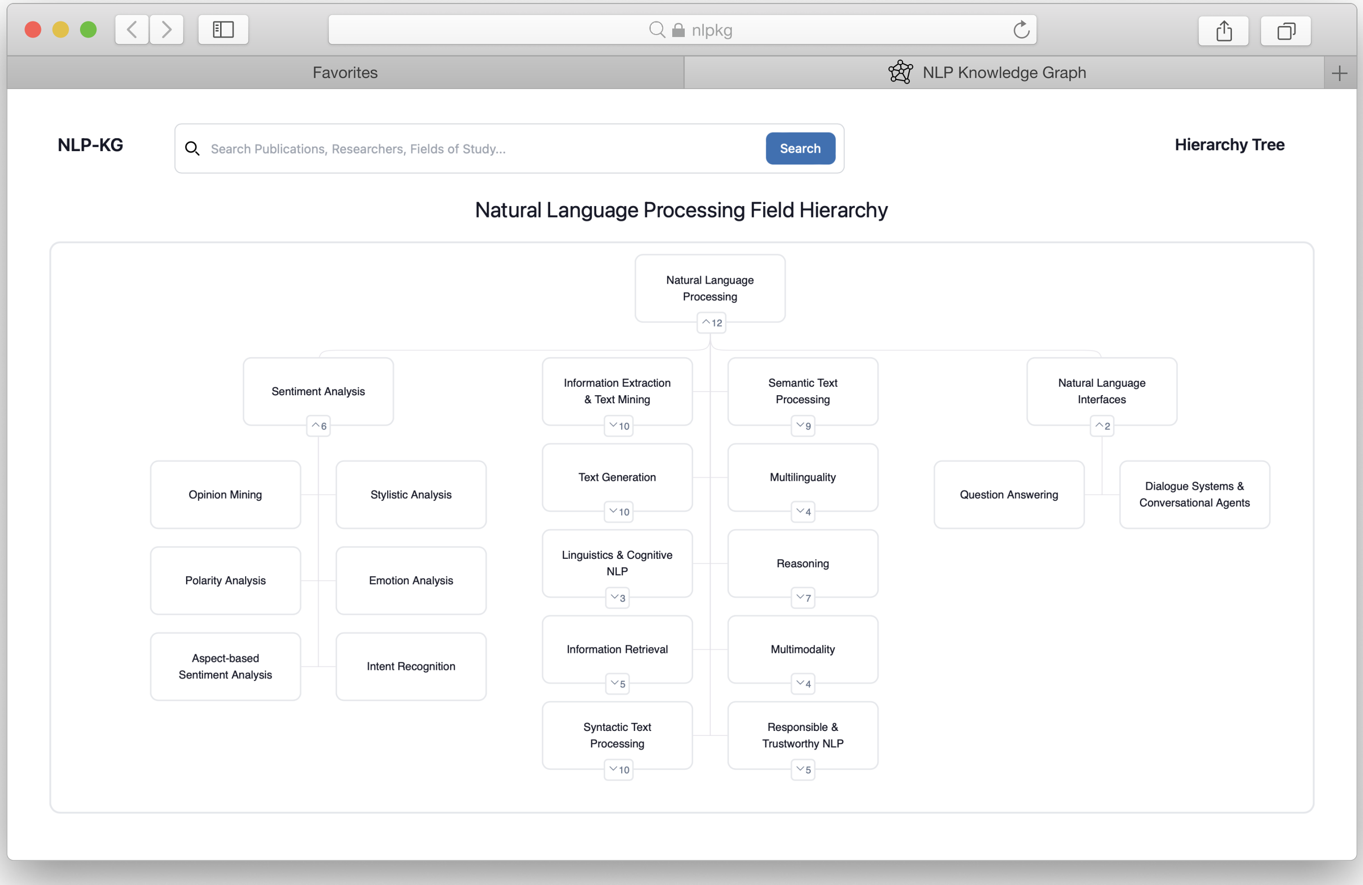

nlpkg
Favorites
NLP Knowledge Graph
NLP-KG
Search Publications, Researchers, Fields of Study...
Search
Hierarchy Tree
Natural Language Processing Field Hierarchy
Natural Language Processing
12
Sentiment Analysis
6
Information Extraction & Text Mining
10
Semantic Text Processing
9
Natural Language Interfaces
2
Opinion Mining
Stylistic Analysis
Text Generation
10
Multilinguality
4
Question Answering
Dialogue Systems & Conversational Agents
Polarity Analysis
Emotion Analysis
Linguistics & Cognitive NLP
3
Reasoning
7
Aspect-based Sentiment Analysis
Intent Recognition
Information Retrieval
5
Multimodality
4
Syntactic Text Processing
10
Responsible & Trustworthy NLP
5

Figure 11: Frontend Result: The complete tree of the natural language processing taxonomy

Bibliography

[Ant] A. Anthology. *ACL Anthology*. URL: https://aclanthology.org/ (visited on 10/14/2023).

[Arx] Arxiv. *About arXiv*. URL: https://info.arxiv.org/about/index.html (visited on 10/14/2023).

[BBA17] B. Bashari Rad, H. Bhatti, and M. Ahmadi. "An Introduction to Docker and Analysis of its Performance." In: *IJCSNS International Journal of Computer Science and Network Security* 173 (Mar. 2017), p. 8.

[BLC19] I. Beltagy, K. Lo, and A. Cohan. *SciBERT: A Pretrained Language Model for Scientific Text*. 2019. arXiv: 1903.10676 [cs.CL].

[Blo02] S. Blomkvist. "Persona - an overview (Extract from the paper The User as a personality. Using Personas as a tool for design. Position paper for the course workshop "Theoretical perspectives in Human-Computer Interaction" at IPLab, KTH, September 3, 2002)." In: (Jan. 2002).

[Bro96] J. Brooke. "SUS – a quick and dirty usability scale." In: Jan. 1996, pp. 189–194.

[Coh+20] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld. *SPECTER: Document-level Representation Learning using Citation-informed Transformers*. 2020. arXiv: 2004.07180 [cs.CL].

[Con] Connected Papers. *About Connected Papers*. URL: https://www.connectedpapers.com/about (visited on 09/30/2023).

[CSS] T. CSS. *Tailwind CSS*. URL: https://tailwindcss.com/ (visited on 10/14/2023).

[DBLa] DBLP. *How can I submit meta data for a complete journal or conference*. URL: https://dblp.org/faq/1474621.html (visited on 09/30/2023).

[DBLb] DBLP. *What institution is behind dblp*. URL: https://dblp.org/faq/1474651.html (visited on 09/30/2023).

[Fas] FastAPI. *FastAPI*. URL: https://fastapi.tiangolo.com/ (visited on 10/14/2023).

[Fel] S. Feldman. *Building a Better Search Engine for Semantic Scholar*. URL: https://blog.allenai.org/building-a-better-search-engine-for-semantic-scholar-ea23a0b661e7.

[Gus19] M. Gusenbauer. "Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases." In: *Scientometrics* 118.1 (2019), pp. 177–214. DOI: 10.1007/s11192-018-2958-5.

[Jac10] P. Jacsó. "Metadata mega mess in Google Scholar." In: *Online Information Review* 34.1 (2023-09-14 2010), pp. 175–191. DOI: 10.1108/14684521011024191.

[Jon15] N. Jones. "Artificial-intelligence institute launches free science search engine." In: *Nature* (2015). DOI: 10.1038/nature.2015.18703.

[Loc] S. Local. *The Layout of Google Search Results Explained*. URL: https://www.successlocal.co.uk/the-layout-of-google-search-results-explained/ (visited on 10/14/2023).

[Mat21] D. Matthews. "Drowning in the literature? these smart software tools can help." In: *Nature* 597.7874 (Sept. 2021), pp. 141–142. DOI: 10.1038/d41586-021-02346-4.

[MY16] Y. A. Malkov and D. A. Yashunin. "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs." In: *CoRR* abs/1603.09320 (2016). arXiv: 1603.09320.

[ND86] D. A. Norman and S. W. Draper. *User Centered System Design: New perspectives on human-computer interaction*. Lawrence Erlbaum, 1986.

[Nex] Next.js. *Next.js*. URL: https://nextjs.org/ (visited on 10/14/2023).

[NM94] J. Nielsen and R. L. Mack, eds. *Usability Inspection Methods*. en. Nashville, TN: John Wiley & Sons, May 1994.

[Nor02] D. A. Norman. *The design of everyday things*. [New York]: Basic Books, 2002. ISBN: 0465067107 9780465067107.

[Pap] Papers with Code. *About Papers with Code*. URL: https://paperswithcode.com/about (visited on 09/30/2023).

[Rea] React. *React*. URL: https://react.dev/ (visited on 10/14/2023).

[RPN20] S. Raschka, J. Patterson, and C. Nolet. "Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence." In: *Information* 11.4 (2020). ISSN: 2078-2489. DOI: 10.3390/info11040193.

[RZ09] S. Robertson and H. Zaragoza. "The Probabilistic Relevance Framework: BM25 and Beyond." In: *Found. Trends Inf. Retr.* 3.4 (Apr. 2009), pp. 333–389. ISSN: 1554-0669. DOI: 10.1561/1500000019.

[SAM23] T. Schopf, K. Arabi, and F. Matthes. *Exploring the Landscape of Natural Language Processing Research.* 2023. arXiv: 2307.10652 [cs.CL].

[Scha] S. Scholar. *About Semantic Scholar.* URL: https://www.semanticscholar.org/about (visited on 10/14/2023).

[Schb] T. Schopf. *Natural Language Processing Knowledge Graph (NLP-KG).* URL: https://wwwmatthes.in.tum.de/pages/57qtab1lrufx/Natural-Language-Processing-Knowledge-Graph-NLP-KG.

[tRP] tRPC. *tRPC.* URL: https://trpc.io/ (visited on 10/14/2023).

[Typ] Typescript. *Typescript.* URL: https://www.typescriptlang.org/ (visited on 10/14/2023).

[UI] D. UI. *Daisy UI.* URL: https://daisyui.com/ (visited on 10/14/2023).

[Wal+02] M. Walker, L. Takayama, J. Landay, and Leila. "High-Fidelity or Low-Fidelity, Paper or Computer Choosing Attributes When Testing Web Prototypes." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 46 (Sept. 2002). DOI: 10.1177/154193120204600513.

[Wea] Weaviate. *Weaviate Indexing.* URL: https://weaviate.io/developers/weaviate/concepts/indexing.