TÉCNICO LISBOA

Management System and Analysis of Scientific Production in the field of Computer Science in Portugal

João Pedro França Pinteus dos Reis Duarte

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisor: Prof. Cláudia Martins Antunes

Examination Committee

Chairperson: Prof. Pedro Tiago Gonçalves Monteiro
Supervisor: Prof. Cláudia Martins Antunes
Member of the Committee: Prof. Rui Miguel Carrasqueiro Henriques

November 2023

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Firstly, I wish to express my heartfelt gratitude to Professor Cláudia Antunes. Her guidance, steadfast support, and insightful advice were invaluable to me throughout the course of this thesis. Her consistent availability and unwavering mentorship have been instrumental in the fruition of this work.

I would also like to extend my profound thanks to Professor João Pavão Martins. His invaluable data contribution concerning the history of AI and its researchers greatly enriched this study.

My family, whose support and patience were unwavering during this process, deserves special acknowledgement. Their steadfast faith and encouragement were my pillars of strength on this academic journey.

Finally, a warm thank you to my friends who have been my constant companions. Their presence and their words of encouragement have been a source of comfort and motivation. Their camaraderie and support played a significant role in the successful completion of this project.

Abstract

This document proposes an information system for supporting the collection, management, and analysis of Portuguese scientific production in the field of computer science. The increasing number of publications and researchers in the field has made it difficult to keep track of the work being developed and to identify key trends, patterns and areas of research.

The system aims to address this challenge by providing a comprehensive and organized view of the scientific production in Portugal. The system will provide a quantitative and qualitative picture of the field, according to different dimensions of analysis such as research groups, time, or scientific topics.

The proposed approach includes building a client-server architecture system, where the server extracts data found online, transforms and loads it on a data warehouse modeled as a star schema, for being analyzed through OLAP operations, clouds of words based on keywords of publications, co-authorship networks, and rankings by author and publication.

Additionally, machine learning algorithms may be applied to classify authors and publications, as illustrated in the presented case study, where authors are assigned to sub fields of computer science, based on the keywords used in their publications.

Keywords

Information System; Scientific Production; Computer Science; Data Warehouse; Data Analysis; Data Mining;

Resumo

Este documento propõe um sistema de informação para apoio à recolha, gestão e análise da produção científica portuguesa no domínio da informática. O crescente número de publicações e de investigadores nesta área tem dificultado o acompanhamento do trabalho desenvolvido e a identificação das principais tendências, padrões e áreas de investigação.

O sistema tem como objectivo responder a este desafio, fornecendo uma visão abrangente e organizada da produção científica em Portugal. O sistema fornecerá uma imagem quantitativa e qualitativa do campo, de acordo com diferentes dimensões de análise, tais como grupos de investigação, tempo ou tópicos científicos.

A abordagem proposta inclui a construção de um sistema de arquitectura cliente-servidor, em que o servidor extrai dados encontrados online, transforma e carrega num armazém de dados modelado como um esquema em estrela, para ser analisado através de operações OLAP, nuvens de palavras, redes de co-autoria, e rankings por autor e publicação.

Adicionalmente, algoritmos de aprendizagem automática podem ser aplicados para classificar autores e publicações, como ilustrado no caso de estudo apresentado, onde autores são atribuídos a sub áreas da informática, baseados nas palavras chave usadas nas suas publicações.

Palavras Chave

Sistema de Informação; Produção Científica; Informática; Armazém de Dados; Análise de Dados; Data Mining;

Contents

List of Figures

Acronyms

APPIA	Associação Portuguesa para a Inteligência Artificial
AI	Artificial Intelligence
DBLP	Digital Bibliography & Library Project
EurAI	European Association for Artificial Intelligence
ECCAI	European Coordinating Committee for Artificial Intelligence
ETL	Extract, Transform, Load
OLAP	Online Analytical Processing
PCA	Principal Component Analysis

1

Introduction

In the last decades, the scientific production in Portugal has increased significantly, leading to an increased diversity of researchers and contributions in the field of computer science. With the growing number of publications and researchers, it has become increasingly difficult to keep track of the work being developed and to identify key trends, patterns and areas of research. This is particularly challenging for those interested in understanding the state of the art in the field, identifying opportunities for collaboration and funding, and for researchers seeking to position their own work within the broader context of the field.

The need to have a comprehensive and organized view of the scientific production in Portugal in the field of computer science has become increasingly important as it allows for better understanding of the current state of research and identification of potential areas for future development. Additionally, having access to such information can also provide valuable insights for policy makers, funding agencies, and academic institutions in terms of identifying areas of strength and areas in need of further support and investment.

The present project aims to address these issues by developing an information system that supports the collection, management, and analysis of Portuguese scientific production in the field of computer science. The system will provide a quantitative and qualitative picture of the field, according to different dimensions of analysis such as research groups, time, or scientific topics. The goal of this project is to make it easier for researchers, policy makers, and funding agencies to keep track of the scientific production in Portugal in the field of computer science and to use this information to make more informed decisions.

The proposed approach for this project is to build a system with a server based on FastAPI and a data warehouse based on PostgresSQL. The system will extract data found online and store it in the data warehouse, where it will be analyzed by applying Online Analytical Processing (OLAP) operations and other techniques. The system will also run machine learning tasks to find patterns and trends in the data, to provide a more comprehensive understanding of the scientific production in the field of computer science in Portugal. By using a server-based on FastAPI, the system will be able to handle large amounts of data and provide a fast and efficient way of extracting, storing and analyzing the data. PostgresSQL will be used as the data warehouse technology, which will help to ensure data integrity and consistency.

The thesis is divided into several chapters, each addressing a different aspect of the project. Chapter 2 provides background information on the current state of the art in the computer science, field associations, and the challenges faced in keeping track of the scientific production in the field of computer science in Portugal. In Chapter 3, the architecture of the system, features and their detailed description are presented. Chapter 4 outlines the data sources and the ETL processes used for fill the database. The Chapter 5 provides the techniques used for the data analysis. Finally, the Chapter 6 provides a case

study, using the system for a specific use case to show the capabilities of the system.

2

Background

Contents

Over the past few decades, there has been a significant rise in scientific publications, culminating in a more diverse pool of researchers and a rich variety of contributions within the realm of computer science. However, with the expansion in both the volume of published works and the number of active researchers, the task of tracking developments, pinpointing prevailing trends, and identifying significant areas of research has grown increasingly complex. This poses a distinct challenge for those interested in assessing the current state of the field, seeking collaborative and funding opportunities, or for researchers aiming to situate their work within the wider field.

The urgency for a thorough and structured overview of Portugal's computer science scientific output has increased. This comprehensive view allows for a more accurate understanding of the present research climate and aids in pinpointing promising areas for future growth. Access to this wealth of information can offer critical insights for policymakers, funding bodies, and educational institutions, aiding in the recognition of areas of proficiency and those requiring additional support and investment.

2.1 Portuguese associations

The Association for Portuguese Artificial Intelligence (Associação Portuguesa para a Inteligência Artificial (APPIA)) is a non-profit organization committed to advancing research, education, and awareness of Artificial Intelligence (AI) as a scientific field in Portugal. As a member of the European Association for Artificial Intelligence (European Association for Artificial Intelligence (EurAI), formerly known as European Coordinating Committee for Artificial Intelligence (ECCAI)), APPIA plays a key role in promoting AI within the country.

APPIA has a rich history of organizing conferences and events to further the understanding and development of AI. The organization's website provides a comprehensive list of past events, as well as resources for those interested in learning more about the field. Additionally, the website features an interactive map highlighting the locations of AI-related meetings that have taken place in Portugal, providing a visual representation of the growth and impact of the organization's efforts [1].

Overall, APPIA's mission is to support the development of AI in Portugal and to foster collaboration between researchers, practitioners, and industry leaders in the field, providing a valuable platform for sharing knowledge and ideas.

INForum is an important event that brings together the national community in the various branches of computer science. The event serves as an ideal platform for young researchers who are looking to showcase their work, receive constructive feedback, and be encouraged in their pursuits.

The INForum event is non-profit, which means its main goal is to provide an opportunity for knowledge exchange and fostering community within the computer science field. It aims to provide a space for young researchers to establish connections and exchange ideas with their peers and with established

figures in the field [2].

2.2 Other non-profit organizations

IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity. IEEE and its members inspire a global community through its highly cited publications, conferences, technology standards, and professional and educational activities.

The IEEE offers a wide range of features designed to support the needs of its users in various areas of their professional lives.

One key feature of the IEEE website is its membership information and benefits. The website provides detailed information about the different types of membership available, as well as the benefits of being an IEEE member. This includes access to exclusive content, discounts on products and services, and opportunities for networking and professional development.

The IEEE website also has a calendar of upcoming events, including conferences, workshops, and webinars organized by IEEE or its affiliates. This is a useful resource for members and visitors who are interested in staying up-to-date on the latest developments in their field and connecting with other professionals.

In addition to its event calendar, the IEEE website also provides access to the full text of IEEE journals and conference proceedings. This is a valuable resource for researchers and professionals looking to stay current on the latest research and developments in their field. The website also offers tools for searching and browsing this content, making it easy to find relevant articles and papers.

The IEEE website also offers professional development resources, such as courses, certification programs, and continuing education opportunities. These resources can help members and visitors to further their careers and stay up-to-date on the latest trends and best practices in their field.

Another important feature of the IEEE website is its information about the technical committees and standards-setting activities of the organization. The website provides details about the development and maintenance of technical standards, as well as information about the work of the various technical committees.

Finally, the IEEE website has a range of career resources to help members with their career development. This includes job listings, career advice, and networking opportunities, as well as information about professional development resources and other career-related resources.

ACM is the world's largest educational and scientific computing society, delivers resources that advance computing as a science and a profession, provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

Doing the research about the features of ACM website we found they are almost the same as IEEE and it makes sense since both professional organizations have similar vision and purposes but while IEEE is more broad focusing in advancing the fields of electrical engineering, computer science and related fields, the ACM focuses specifically on computer science and related fields.

Consequently, to figure out the technologies which are being used by them, we will use a tool called called **Wappalyzer**.

Wappalyzer is a browser extension and a software tool that analyzes web pages and detects the software technologies and frameworks that are used to build them, in this case we insert the website link and it does a study of which technologies are being used in the respective website.

Regarding IEEE website we can see it's using Angular framework and JQuery javascript library for its client side while in ACM website we can see it's using JQuery javascript library and D3.js for data visualization.

When attempting to analyze the client-side technologies of these websites, the information obtained is not comprehensive enough to make reliable conclusions.

Regarding the server we couldn't find nothing that we consider important, we know this information is often hidden and some more complex web applications use different technologies in their servers.

2.3 Server technologies

When it comes to developing software, there is often a debate about whether to use or not use frameworks. Both approaches have their own set of benefits and drawbacks, and the decision ultimately comes down to the specific needs and goals of a project.

On the one hand, frameworks can provide a number of benefits for developers. They offer a set of pre-built, modular components that can be easily integrated into a project, saving time and effort. They also often come with a built-in structure and set of best practices, which can help to ensure that the code is organized and maintainable.

However, there are also some potential downsides to using frameworks. One concern is that they can be inflexible, as the developer is limited to using the components and patterns provided by the framework. This can make it difficult to customize the project or to introduce new features that fall outside of the framework's capabilities. In addition, frameworks can be heavyweight and can add unnecessary complexity to a project, making it more difficult to understand and maintain.

Ultimately, the choice between using or not using frameworks will depend on the specific needs and goals of a project. For small or simple projects, not using frameworks may be sufficient. For larger or more complex projects, a framework may be more helpful in providing structure and saving time. It is important to carefully consider the trade-offs and choose the approach that best meets the needs of the

project.

This way, we have several programming languages that are commonly used to build servers. Some of the most popular include:

- **Java**: Java is a popular, object-oriented programming language that is widely used for building enterprise-level applications.
- **Kotlin**: Kotlin is a modern programming language that is fully compatible with Java. It is designed to improve upon some of the shortcomings of Java, such as verbosity and null-safety. Kotlin is widely used for building server-side applications and is supported by many popular web development frameworks such as Spring and Vert.x.
- **Python**: Python is a high-level, general-purpose programming language that is widely used for building web applications, scientific applications, and data analysis tools.
- **Go** (also known as Golang): Go is a statically-typed, open-source programming language that is designed for building high-performance and concurrent systems. It is widely used for building server-side applications, particularly in the field of networking, and it has a built-in concurrency model that makes it well-suited for building highly concurrent systems.
- **C#**: C# is a popular programming language that is designed for building a wide range of applications, including web applications.
- **JavaScript**: JavaScript is a popular programming language that is commonly used on the frontend, while in the beginning it was a language that only could be used on frontend, nowadays it can also be used on the backend with the help of runtime environments such as Node.js. Express is an example of a popular Javascript framework capable to build server applications [3].

From this, we have to filter this languages by the ones who are better in the field of data science and machine learning. This way, Python and Java seems to be the best for it.

Python is a versatile and powerful programming language that is well-suited for data science and machine learning tasks. We will explore some of the reasons why Python is a popular choice for data science and machine learning.

One of the main reasons why Python is so popular for data science and machine learning is its large and active community. The Python community is constantly developing new libraries, frameworks, and resources for data science and machine learning. This means that there is a wealth of tools available to data scientists and machine learning engineers, which can save them a lot of time and effort.

Another reason why Python is good for data science and machine learning is its simplicity and ease of use. Python has a simple, easy-to-read syntax, which makes it easy to learn and use, even for those

with little programming experience. This makes it a great choice for data scientists and machine learning engineers who want to focus on their data and models, rather than struggling with the intricacies of a complex programming language.

Python also has a number of powerful libraries for data science and machine learning. These libraries, such as NumPy, pandas, and scikit-learn, provide a wide range of tools for data manipulation, analysis, and visualization. These tools are essential for data science and machine learning tasks, and they can save data scientists and machine learning engineers a lot of time and effort.

Python is also a language that can be integrated with other languages, such as R, Java, and C++. This makes it easy to use with other existing software and allows for interoperability. This means that we can use the best tools from different languages, depending on the task at hand.

Lastly, Python has many libraries for deep learning such as TensorFlow, Keras, PyTorch and many more, which provides easy to use API for creating and training neural network models. This makes it easy to create and train sophisticated deep learning models, which are becoming increasingly important in data science and machine learning.

In conclusion, Python is a popular choice for data science and machine learning tasks due to its large and active community, simplicity and ease of use, powerful data science libraries, interoperability, flexibility, and support for deep learning. These features make it a great choice for who wants to focus on their data and models, rather than struggling with the intricacies of a complex programming language.

On the other hand, **Java** is also a popular programming language that is well-suited for data science and machine learning tasks. We will explore some of the reasons why Java is a good choice for these tasks.

One of the main reasons why Java is good for data science and machine learning is its platform independence. Java is designed to be platform-independent, which means that it can be run on a variety of operating systems and environments. This makes it a great choice for data science and machine learning tasks, as the same code can be used across different systems. This is especially useful for organizations that work with multiple operating systems and need a language that can be used across all of them.

Another reason why Java is good for data science and machine learning is its large and active community. Java has a large and active community of developers and users, which provides a wealth of libraries, frameworks, and resources for data science and machine learning. This means that there is a wealth of tools available to data scientists and machine learning engineers, which can save them a lot of time and effort.

Java is known for its good performance, which is particularly important for data science and machine learning tasks that involve large amounts of data and complex models. Java is designed to be scalable, which makes it well-suited for data science and machine learning tasks that involve large amounts of

data and complex models.

Java has built-in support for distributed computing, which makes it easy to scale up data science and machine learning tasks to run on a cluster of machines. This feature is especially useful to process and analyze big data sets. Java also has libraries such as Weka, Deeplearning4j, and MOA which are widely used for machine learning.

Java is widely used in industry, and many organizations use it to build production-ready systems. This makes it a good choice for who wants to deploy their models in a production environment.

In summary, Java is a good choice for data science and machine learning tasks due to its platform-independence, large and active community, good performance, scalability, support for distributed systems, libraries for machine learning, integration with big data frameworks, and its ability to be used to build production-ready systems. These features make Java a solid choice to build robust and efficient systems.

Now that we pick the programming languages we should research about modern frameworks to be able to build the server. Thereby, making a research of what frameworks exist in the industry, we could find some alternatives being the most popular ones for each programming language:

2.3.1 Python

- **Django**: Django is a high-level Python web framework that allows for quickly building web applications. It includes a built-in ORM (Object-Relational Mapper) that allows you to interact with databases, as well as a range of tools for working with HTTP requests and responses [4].
- **Flask**: Flask is a lightweight Python web framework that is easy to get started with. It is well-suited for building REST APIs and has a number of extensions that can be used to add async capabilities and support for machine learning tasks [5].
- **FastAPI**: FastAPI is a modern, fast, and lightweight Python web framework that is designed for building APIs. It has built-in support for async tasks and can be used with popular machine learning libraries such as TensorFlow and PyTorch [6].

2.3.2 Java

- **Spring**: Spring is a comprehensive framework that provides a wide range of features for building enterprise-grade applications, including support for dependency injection, data access, and web development [7].
- **Apache Struts**: Apache Struts is an open-source framework for building web applications using the Model-View-Controller (MVC) design pattern [8].

2.4 Data Storage System

A data storage system is a system that is used to store, organize, and manage data. There are many different types of data storage systems, including:

2.4.1 Transactional databases

are structured systems for storing and organizing data in tables. They are commonly used for storing data that needs to be accessed and updated frequently, such as customer records or inventory data.

2.4.2 Data warehouses

Are large-scale, centralized systems for storing and managing data from a variety of sources. They are designed to support fast querying and analysis of data, and they are often used for business intelligence and reporting.

2.4.3 NoSQL databases

Are non-relational databases that are designed to handle large amounts of data that is distributed, unstructured, or has varying schemas. They are often used for storing data that is generated by web applications, such as user activity data or real-time sensor data.

2.4.4 SQL (Relational databases) vs NoSQL (Non-relational databases)

Relational databases and non-relational databases are two different types of data storage systems that are used to store and manage data.

When choosing a modern database, one of the biggest decisions is picking a relational (SQL) or non-relational (NoSQL) data structure. While both are viable options, there are key differences between the two that we must keep in mind when making a decision. [9]

One of the main differences between relational and non-relational databases is the **way that they store data**. Relational databases store data in tables with rows and columns, and they use structured query language (SQL) to manage and manipulate the data. This makes it easy to organize and access data using a fixed schema. Non-relational databases, on the other hand, do not use a fixed schema and may store data in a variety of different formats, such as documents, key-value pairs, or graph data. This makes them more flexible and adaptable to different types of data, but it can also make it more difficult to organize and access data.

Another key difference between relational and non-relational databases is the way that they **support querying and searching data**. Relational databases use SQL to query data, and they support a wide

range of query operations and functions, such as SELECT, INSERT, UPDATE, DELETE, and JOIN. This makes them powerful and flexible for querying and manipulating data. Non-relational databases may use a variety of different query languages or may not support SQL at all. Some non-relational databases may provide more powerful and flexible query capabilities than relational databases, while others may have more limited query capabilities.

Scalability is also an important consideration when choosing between relational and non-relational databases. Non-relational databases are often horizontally scalable, meaning that they can be easily scaled out by adding more servers. This makes them well-suited for handling very large datasets or high levels of traffic. Relational databases may be more difficult to scale out, and they may require more complex clustering or sharding solutions.

Finally, **transactions** are an important feature of relational databases that are not always supported by non-relational databases. Transactions allow multiple related updates to be committed or rolled back as a unit. This makes them well-suited for applications that require strong support for data integrity and consistency. Non-relational databases may not support transactions or may have limited support for them.

In conclusion, relational and non-relational databases have different characteristics and advantages, and the choice of which one to use depends on the specific requirements of the application. Relational databases are powerful and flexible for querying and manipulating data and provide strong support for transactions and data integrity, while non-relational databases are more adaptable and scalable, especially for large datasets and high traffic.

2.4.5 Transactional Databases vs Data Warehousing

When it comes to storing and managing data, there are two main options: transactional databases and data warehouses. They have some key differences that make them better suited to different use cases. Thus, we will compare the two technologies across a range of parameters to help understand which one is best for our needs [10].

One of the main differences between transactional databases and data warehouses is the way they are **used**. Transactional databases are designed to store data, while data warehouses are designed for analyzing data. This means that if we need to store data in real-time, a relational database is likely the better option. If, on the other hand, we need to perform complex analysis on large volumes of data, a data warehouse may be the better choice.

Transactional databases and data warehouses also use different **processing methods**. Relational databases use OLTP (Online Transaction Processing), which is optimized for small transactions. Data warehouses, on the other hand, use OLAP (Online Analytical Processing), which is optimized for complex analysis.

Another key difference is the number of **concurrent users** that each technology can support. Transactional databases are designed to handle thousands of concurrent users, making them a good choice for applications that need to handle high volumes of traffic. Data warehouses, on the other hand, are generally only able to support a limited number of concurrent users, as they are optimized for complex analysis rather than high-volume transactional processing.

The types of **use cases** that are best suited to each technology also differ. Transactional databases are best for small transactions, such as recording data for an online store or a banking system. Data warehouses, on the other hand, are best for complex analysis, such as business intelligence or analytics applications.

Another important consideration is **downtime**. Transactional databases are designed to be always available, as they are used to support mission-critical transactional systems. Data warehouses, on the other hand, may have some scheduled downtime, as they are often used for less time-sensitive applications.

The way that each technology is **optimized** also differs. Transactional databases are optimized for CRUD (create, read, update, delete) operations, which are the basic building blocks of transactional systems. Data warehouses, on the other hand, are optimized for complex analysis, which requires fast, efficient access to large volumes of data.

The **type of data** that each technology is best suited to handle is another key difference. Relational databases are typically used to store real-time detailed data, while data warehouses may be used to store summarized historical data.

In summary, transactional databases and data warehouses are two different technologies that serve different purposes. Transactional databases are best for recording and storing data in real-time, while data warehouses are best for supporting the analysis of large volumes of historical data.

2.5 Data Analytics Tools

Data analytics tools are becoming increasingly popular as organizations strive to make sense of their data and gain insights to inform their decision-making. Among the many data analytics tools available, PowerBI, Tableau, Metabase and QlikView are some of the most popular choices. Consequently, We will explore the features and capabilities of these tools and discuss why they are widely used in the data analytics field.

PowerBI, is a business intelligence tool developed by Microsoft, that is widely used for creating interactive data visualizations and dashboards. It is user-friendly and easy to use, and it provides a wide range of features for data analysis and visualization. PowerBI can connect to a wide range of data sources, including Excel, SQL Server, and cloud-based data sources such as Azure and Google

Analytics. It also provides a suite of data transformation and modeling tools, making it a versatile tool for data analysis [11].

Tableau is another data visualization tool that is widely used for creating interactive and visually appealing data visualizations. It is user-friendly and easy to use, and it can connect to a wide range of data sources. Tableau's drag-and-drop interface makes it easy to create and share interactive visualizations, and its wide range of data visualization options allows users to explore and analyze their data in new and powerful ways [12].

Metabase is an open-source data analytics tool that is widely used for creating and sharing data visualizations. It is designed to be simple and easy to use, making it accessible to users with little or no programming experience. With Metabase, we can easily connect to their data sources, create and share interactive visualizations, and build custom dashboards. Additionally, Metabase provides a SQL interface, allowing us to write complex queries and analyze data [13].

Qlik Sense is a data visualization and business intelligence (BI) software that allows users to explore and analyze data from multiple sources. It is a self-service BI tool that enables users to create and share interactive dashboards, reports, and visualizations without the need for IT support. Qlik Sense offers a range of built-in data connectors, data modeling, and data visualization capabilities, as well as a scripting language called QlikScript, which allows users to create custom data visualizations and automate data analysis tasks. Additionally, it also offers collaboration tools, and it can be integrated with other systems and platforms, such as cloud services, big data platforms, and enterprise systems [14].

In conclusion, these are some of the most popular data analytics tools available. Each of these tools provides a wide range of features and capabilities for data analysis and visualization, making them well-suited for a variety of data analytics tasks. They are designed to be easy to use, flexible, and can connect to a wide range of data sources.

2.6 Client technologies

There are many frontend technologies and frameworks options we can choose, being the most popular:

- **Angular**: Angular is a popular frontend framework that is built and maintained by Google. It is designed to help developers build large and complex applications in an organized and maintainable way. Angular includes a wide range of features and tools, such as a powerful templating system and dependency injection, that can help you build high-quality applications quickly and efficiently [15].

- **React**: React is a popular frontend framework that is developed and maintained by Facebook. It is designed to help developers create user interfaces that are fast, scalable, and easy to maintain.

React uses a declarative programming style, which makes it easy to reason about and debug your application's behavior. It also includes a powerful library of reusable components, called React components, that you can use to build your application's user interface [16].

- **Vue.js**: Vue.js is a popular frontend framework that is designed to be lightweight, fast, and flexible. It is easy to learn and use, and provides a powerful set of tools and features that can help you build high-quality applications quickly and efficiently. Vue.js includes a powerful templating system, a reactive data-binding model, and a tool called Vue CLI that can help you scaffold and build your application [17].

2.7 Containerization

Containerization is a method of packaging software applications in a way that makes them easier to deploy and manage. It involves packaging an application and all its dependencies, such as libraries and runtime, into a container. The container can then be run on any machine that has a container runtime installed, regardless of the underlying operating system or infrastructure. This makes it **easier to deploy and run applications in different environments**, such as on-premises servers, in the cloud, or on a developer's local machine.

Containers are a popular tool in software development because they provide a lightweight and portable way to package and distribute applications. They allow developers to isolate their applications from the underlying infrastructure, making it easier to test and deploy applications in different environments. Containerization is often used in conjunction with container orchestration tools, such as Kubernetes, to manage and scale containerized applications in production environments.

There are some existing container technologies although there is one that is largely the most popular, **Docker**. According to stackoverflow survey 2022, 68.57% of professional developers use docker on their job [18].

Docker is a containerization platform that allows developers to package and deploy applications in containers. It is one of the most widely used container technologies, with a large user base and a vast ecosystem of tools and services built around it [19].

3

Solution Architecture

Contents

The goal of this project is to develop a system for the storage and analysis of scientific publications written by any author affiliated to an institution in any area and identify patterns and trends within the field. To achieve this goal, several requirements have been identified for the system.

3.1 Requirements

First, the system must be able to extract data from the web. This is an essential component of the system, as it will allow the system to gather and collect the necessary scientific publications.

Next, the system must be able to load the data into a storage system. This is crucial for the system, as it will allow the data to be easily accessed and analyzed. The storage system must be designed in such a way that it is easy to analyze, allowing for efficient data mining and machine learning tasks.

Additionally, the system must be capable of executing scheduled data updates between a time interval to keep data updated. This will ensure that the system remains current and relevant, providing the most up-to-date data for analysis.

Furthermore, the system must be capable of running machine learning tasks, which will be used to analyze the data and extract valuable insights. This is an essential component of the system, as it will allow for the identification of patterns and trends within the data.

Finally, to see the results, the system also needs to provide a way for users visualizing data and its analysis results.

One of the main objectives of this project is to make the system as easy to maintain as possible. To achieve this, the system will be built in a way that respects SOLID principles by Robert C. Martin, which are a set of guidelines for writing maintainable and scalable code [20]. Additionally, the system will be designed to be easily scalable, so that it can accommodate an increasing amount of data over time and we or other team can easily add features in the future.

Another important consideration for this project is the choice of technologies to use. Because the project is being developed *pro bono*, we will be selecting free and/or open-source technologies to build the system.

Overall, the goal of this project is to develop a system that can effectively store and analyze scientific publications. By meeting the requirements outlined above, the system will be able to extract, load, and analyze data in an efficient and effective manner.

3.2 Funcionalities

3.2.1 Data Summarization

The system will utilize advanced data summarization techniques, specifically OLAP operations, to present the data in a variety of ways.

We will be able to view general statistics such as the total number of published works, as well as more specific data such as the number of publications per author, per group, per time interval, per field, per publication type, and per publisher. This granular level of data summarization will allow to gain a deeper understanding of data.

Another features of the system is the ability to view data on an author-by-author basis. This will enable to understand the productivity and impact of individual authors, and how their work compares to their peers.

The system will also allow to view published works per time interval and per field, which will enable to understand how the field has evolved over time, and what the current research trends are. Additionally, the system will allow to view published works per publication type and per publisher, which will enable to understand which types of publications are most prevalent and which publishers are the most active in the field.

3.2.2 Word Cloud Visualization of Publication Keywords

One of the central components of our system designed for analyzing the historic and current state in the field is the employment of word cloud visualizations. Word clouds are generated using keywords extracted from the publications. This data visualization tool facilitates an immediate understanding of the salient themes, topics, and research focuses. By emphasizing the keywords that appear more frequently across the collected publications, word clouds provide an intuitive representation of predominant discussions in this rapidly evolving field.

3.2.3 Co-Authorship Networks

Another integral feature of our system is the visualization of co-authorship networks. Collaboration is a cornerstone of progress within research community, and the co-authorship network offers valuable insights into these collaborative patterns. By representing authors as nodes and their joint publications as edges, this network helps decode patterns of collaboration between authors and research groups. The resulting analysis can spotlight leading contributors and influential clusters within the community, enhancing understanding of the collaborative structure and shared expertise in this field.

3.2.4 Author and Publication Ranking Metrics

The system further integrates advanced author and publication ranking mechanisms, relying on a range of quantitative metrics. These metrics not only encompass the number of an author's publications but also take into account the temporal aspect (year, decade, and century) of the publications.

Author rankings provide a quantitative snapshot of an author's productivity and influence within the domain.

Publication rankings, on the other hand, contribute to our understanding of the evolution of research topics over time, and highlight the contribution of various publishers to the field. By enabling systematic evaluation of these factors, we can foster a more granular understanding of the dynamics of any field, illuminating both historical progress and potential future directions.

3.2.5 Data Mining

In addition to the above features, our system is equipped with machine learning capabilities for data mining, thereby enhancing its analytical potential. It leverages machine learning algorithms to label authors into different fields based on the keywords found in their published works. This approach enables the system to identify and understand the primary research areas of individual authors, offering a more nuanced perspective of the research landscape. Additionally, it will also be possible to predict the number of publications that an author is likely to have in the near future, discover trends in the use of keywords over time, and more.

3.2.6 Website for Data Visualization

In addition to data accessibility, our system includes a user-friendly web interface designed to visually present the analyzed data and its features. This web-based platform serves as a visual gateway to the wealth of information extracted and analyzed by our system, showcasing the word cloud visualizations, co-authorship networks, ranking metrics, and other data views. By offering a comprehensive and intuitive interface, the website enhances user engagement and aids in the dissemination of the insights derived from the system.

In conclusion, the system offers a robust and comprehensive framework for extracting and analyzing scientific publications. Through the synthesis of advanced features such as OLAP-based data summarization, machine learning capabilities, and a user-friendly web interface, the system provides a detailed picture of the research landscape within an area. These tools collectively enhance our understanding of research trends, collaboration patterns, influential individuals, and publishers, offering valuable insights into both the historical trajectory and the current state of any area.

3.3 Solution Architecture

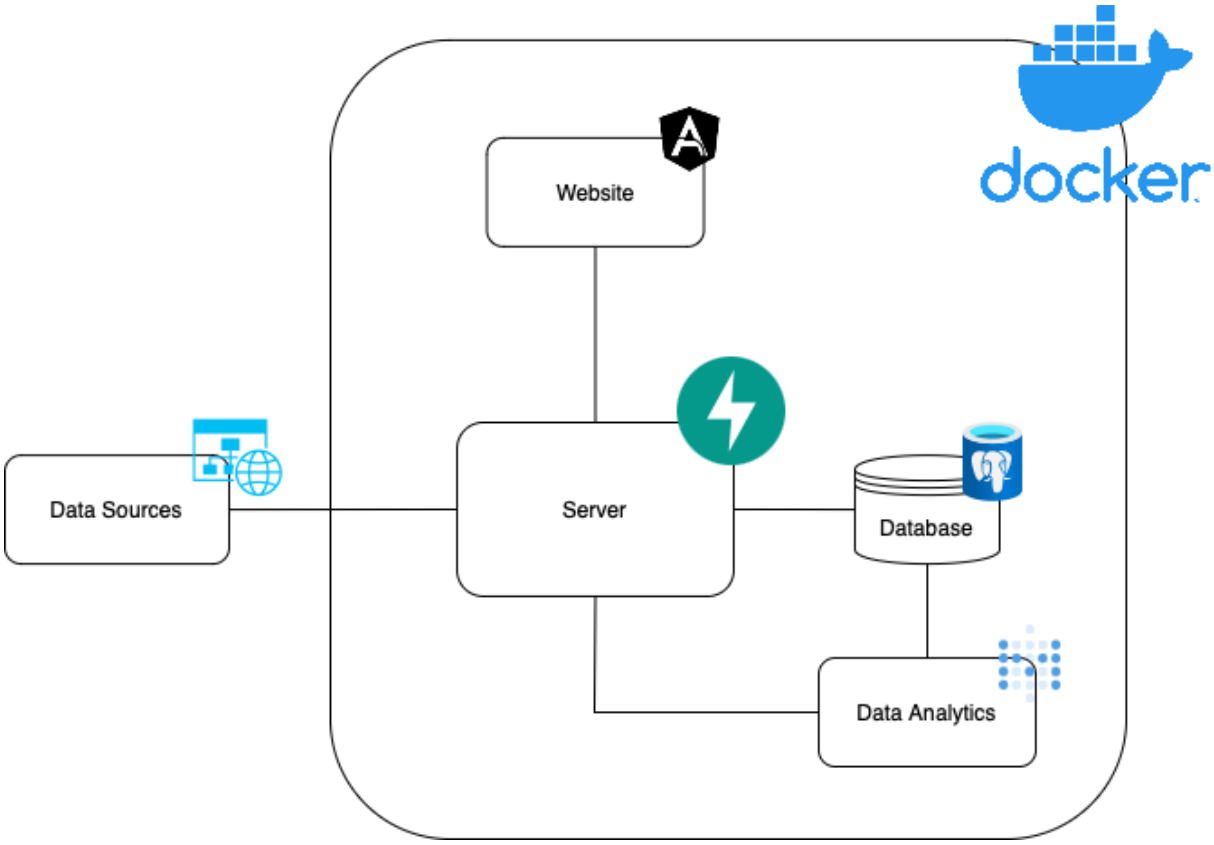

Figure 3.1: Architecture

The architecture of our system, depicted in Figure 3.1, is modular. It comprises several key components, each fulfilling a specific role, that work in harmony to facilitate the extraction, analysis, and visualization of scientific productions. The diagram provides an overview of the architecture, indicating how data flows between these interconnected components, thus giving a sense of the system's functionality as a whole. This modular approach not only enhances efficiency but also allows for scalability and adaptability to evolving needs.

The **server** component, implemented using FastAPI [6], forms the backbone of the system, interacting with various elements to orchestrate the flow of data. It is directly linked to the data sources, enabling the retrieval and storage of data. Furthermore, it interacts with the database component for data storage and retrieval purposes. This server also communicates with the web interface to serve necessary data upon request and is linked to the data analytics tool to make the data-driven dashboards accessible via the website.

The **web** interface, built with Angular [15], serves as the front-end of the system. It communicates with the server to obtain data from the data warehouse and presents it in a user-friendly manner. The

interface provides a variety of visualizations, including graphs, charts, and tables, offering users a comprehensive and intuitive understanding of the data.

The **data analytics** tool, developed using Metabase [13], directly connects to the database. It primarily aids development efforts by facilitating the exploration and analysis of data housed in the data warehouse. This component is crucial for creating easily understandable and insightful data dashboards, thereby enhancing our system's analytical capabilities.

To promote seamless deployment, scalability, and uniform functioning of the system across environments, all components are **containerized using Docker** [19]. This process packages the components with their dependencies into standardized units called containers, ensuring efficient management and reliable operation of the application regardless of the deployment environment.

In essence, the system architecture comprises the FastAPI-based server, Angular-based web interface, Metabase-based data analytics tool, and Docker-based containerization. Working synergistically, these components ensure the effective and efficient operation of the system, facilitating the extraction, analysis, and presentation of valuable insights from scientific productions.

3.3.1 Server

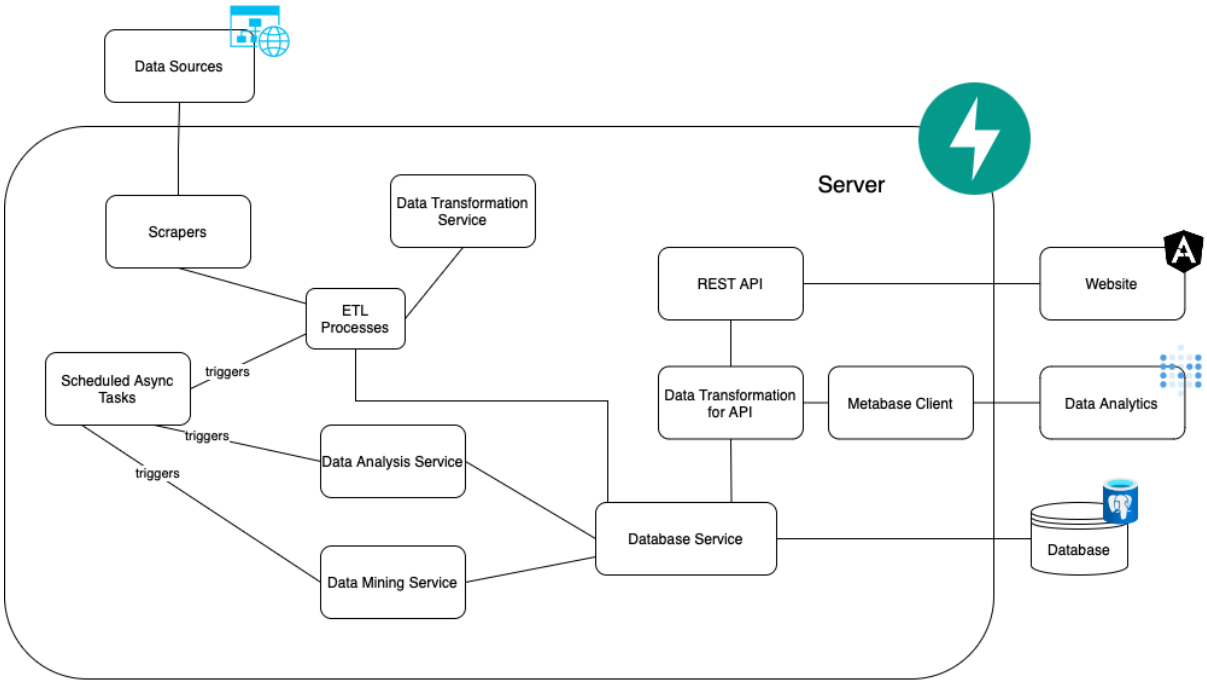

Figure 3.2: Architecture

At the core of our system lies the server component, organized meticulously using a modular approach. This architecture has been carefully chosen to streamline functions and enhance the maintain-

ability and scalability of the system.

The technology that the server will be based is FastAPI [6], the FastAPI framework is the chosen one because it's designed with a specific goal which is to simplify the creation of web services that allow different applications to talk to each other, user friendly documentation and support to asynchronous tasks which is useful to run machine learning tasks and to update our data regularly. On the other hand, it's the youngest Python framework of the ones listed above, so it's possible that it has limited community support, however it's increasing a lot in popularity since its release and its documentation is very user friendly in contrast to what we see in the Flask framework [5] .

Regarding the alternatives, Django [4] is excessively complex for the problems we want to solve since it's a complete framework with frontend and backend and with a considerable amount of built-in modules from scratch, and we will not use most of them, which may make the system more heavy than necessary. Even being the most popular, its learning curve is the highest because of its complexity.

Regarding Flask, it is also a web framework but much more lightweight than Django because we have the power to choose which modules we want to use, but its documentation doesn't seem friendly comparing to FastAPI.

Primarily, the **ETL Processes** Module is one of the cornerstones of the system. It handles the essential task of extracting raw data from various data sources.

Once retrieved, it proceeds to transform the data, making it compatible with the system's specifications and requirements.

Finally, it loads the processed data into the database, where it is stored for future retrieval and use. This systematic workflow ensures the smooth operation of the ETL processes, forming the backbone of our data pipeline.

This module liaises closely with the **Scraper Service**, an integral part of the system that deals with data extraction from online sources that do not provide easily accessible APIs or have data presented in an unwieldy format. The Scraper performs HTTP requests to specific websites, parses the HTML responses, and extracts relevant data.

In parallel, the **Data Analysis Service** works tirelessly to conduct a broad spectrum of data analyses. Its key function involves constructing intricate co-authorship networks that provide invaluable insights into the patterns of author collaboration. However, its design is flexible and scalable, providing a robust foundation for incorporating additional analysis operations as the system evolves and the need for more complex analyses arises.

Complementing the Data Analysis Service, the **Data Mining Service** contributes to the system's analytical prowess by labeling authors into various fields, which will be explained more in-depth on the Data Analysis chapter. This task is performed by scrutinizing the authors' published works and their associated keywords. Like the Data Analysis Service, the Data Mining Service has been designed with

extensibility in mind, enabling potential expansion to include other sophisticated data mining operations in the future.

The **Scheduled Async Tasks module** plays a crucial role in the system by ensuring that the data remains current and relevant. By initiating the ETL processes, Data Analysis Services, and Data Mining Services at regular time intervals, it allows the system to stay up-to-date with the latest data, thereby enhancing the accuracy and reliability of the system's output.

To facilitate smooth interactions with the database, the **Database Service** acts as an abstraction layer, simplifying SQL operations and improving the overall manageability of the system.

Finally, the **REST API** module serves as the system's bridge to the outside world, offering various endpoints that allow external entities to access the data. Operating within the parameters of the HTTP protocol, the REST API is primarily responsible for facilitating data retrieval requests from the database. Built on architectural principles like client-server architecture, statelessness, cacheability, and a layered system, the REST API module is designed to provide lightweight, scalable, and easy-to-use access to system data. It effectively separates the front-end interface from the back-end logic and data storage, enhances scalability by maintaining stateless operations, reduces server load by being cacheable, and offers simplified interaction through a fixed set of APIs, regardless of the underlying infrastructure [21].

In summary, our server architecture is a meticulous blend of several modules, each working in harmony to enable efficient and effective operations. The modular approach offers the flexibility needed for future enhancements, facilitating comprehensive, up-to-date analysis of scientific productions within any area.

3.4 Data Model

A data model is a way of organizing and structuring data so that it can be stored, managed, and accessed efficiently.

For this project we choose to use a data warehouse, as the main purpose of this project is to analyze data and execute data mining processes. Other reason, is that the database updates will be large (biannually) since we don't need real-time data, so using a relational database is not required in terms of performance.

Main reasons to choose a data warehouse schema are:

- **Data integration**: Data warehouses can be used to integrate data from a variety of sources, such as transactional databases, log files, and external data sources, and we will probably extract data from several data sources such as bibtex files and website scraping.
- **Centralized data**: A data warehouse provides a single, centralized repository for storing and

managing structured data, which can make it easier to access and analyze data from multiple sources.

- **Data quality**: Data warehouses often include processes and tools for cleaning, transforming, and validating data, which can improve the quality and reliability of the data.
- **Historical data**: Data warehouses typically store historical data as well as current data, which can be useful for trend analysis and other types of business intelligence and analytics activities that require a long-term view of an organization's data.

3.4.1 Schema

Star schema.

A star schema is a type of database schema where a central table, known as the fact table, is connected to one or more dimension tables. The fact table contains the measures or facts of the data, such as Publications and Affiliations while the dimension tables contain the attributes or characteristics of the data, such as title, abstract, publisher [10].

One of the main characteristics of a star schema is that it is highly denormalized. This means that the same data is repeated in dimension tables. Additionally, the fact table and dimension tables are connected through foreign key relationships, allowing for quick and efficient joins.

Another characteristic of a star schema is its simplicity. Because the schema is easy to understand and navigate, it is well-suited for business users and analysts who need to quickly access and analyze the data. This makes it a popular choice for data warehouses, where fast query performance and ease of use are critical.

A star schema also allows for easy aggregation of data. Because the fact table contains the measures of the data, it is simple to sum, average, or perform other calculations on the data. This makes it a great choice for creating reports and dashboards.

Next, we will list and describe the dimension and fact tables on the data model, presented on Figure 3.3, and how they can be used to gain insights into various business processes.

First, let's take a look at the **dimension tables** in this schema. The dimension tables include Author, Institution, Scientific area, Published Work, Date and Term. The **Author** dimension table contains data about the authors such as name, date of birth, birth place, and nationality. This data can be used to identify the authors of published works and to gain insights into their demographics. The **Institution** dimension table contains data about the institutions where authors are and were affiliated, such as institution name, university name, city, country, and type (research center or education center). This data can be used to identify the institutions where authors are affiliated and to gain insights into the types of institutions that are producing research in a particular field. The **Scientific Area** dimension

table contains data about the field of study, sub-field, domain, and area of published works. This data can be used to identify the scientific areas in which published works are categorized and to gain insights into the specific topics that are being studied. The **Published Work** dimension table contains data about the published works themselves, such as title, abstract, type, publisher, publication name, DOI, source url, and pdf url. This data can be used to identify specific published works and to gain insights into the characteristics of those works. The **Date** dimension table contains data about the date that a published work was published, such as month, year, decade, and century. This data can be used to identify when a published work was published and to gain insights into trends in published works over time. Finally, the **Term** dimension table represents the keywords used in published works.

Now, let's take a look at the **fact tables** in this schema. The fact tables include PhD, Affiliation, Publication, and Keywords. The **PhD** fact table contains data about the authors, supervisors, institutions, and published PhD thesis. The **Affiliation** fact table contains data about the authors, institutions, start of affiliation (Date dimension) and end of affiliation (Date dimension). This data can be used to track the past and current affiliations of authors and to gain insights into the institutions that are producing research in a particular field. The **Publication** fact table contains data about the authors, published works, scientific areas and date of publication. This data can be used to track the publications of authors and to gain insights into the scientific areas in which those publications are categorized. Finally, the **Keyword** fact table contains data about the published works and the terms associated with them. This data can be used to track the keywords associated with published works and to gain insights into the specific topics that are being studied or were studied in the past.

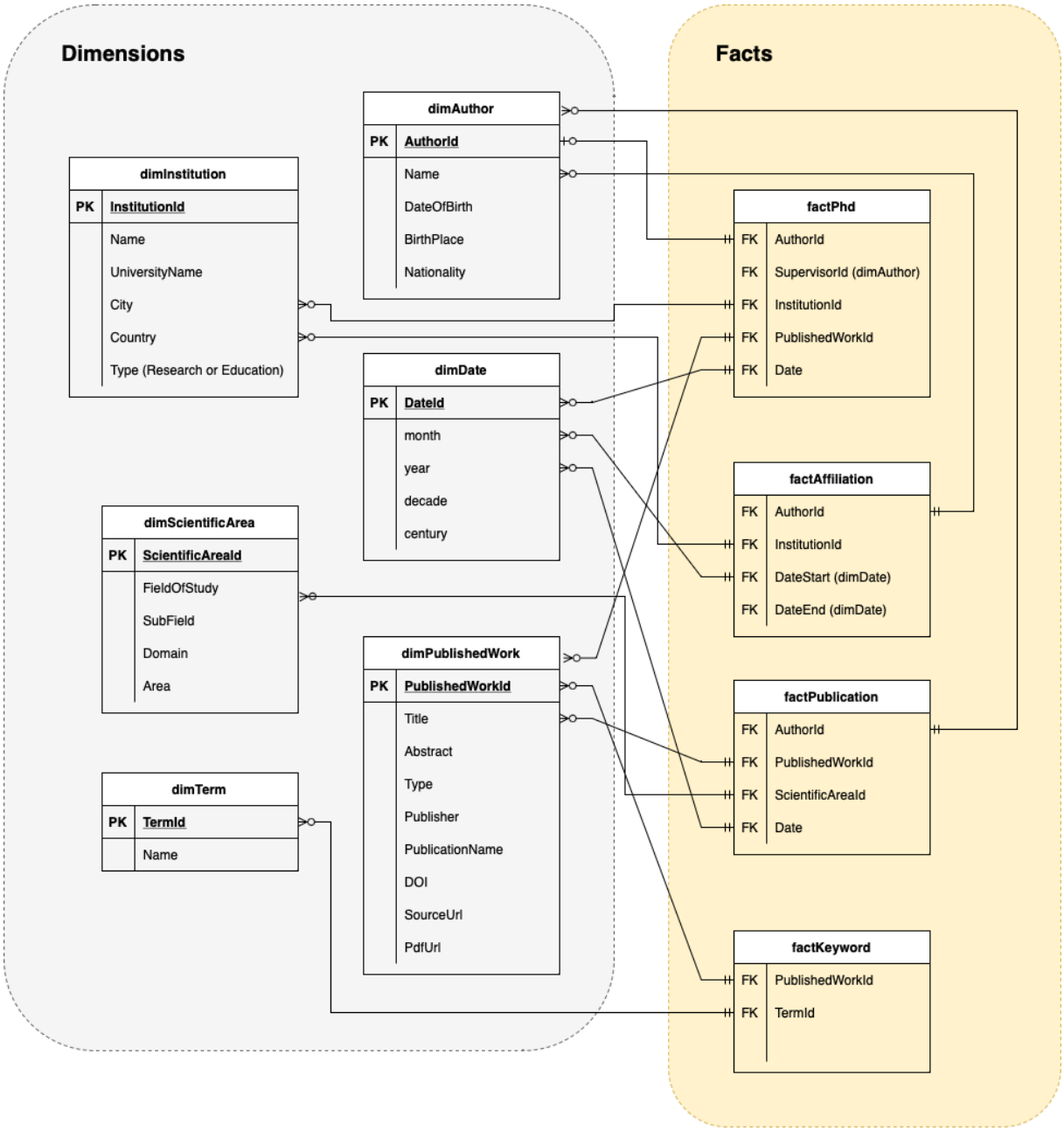

Figure 3.3: Data Model

4

ETL

Contents

In this chapter, we will delve into the vital underpinnings of our system, Extract, Transform, Load (ETL) processes. These processes are at the heart of our data handling operations, ensuring that we collect, refine, and store data in a manner that is optimally structured for our analytical needs.

We will begin with a comprehensive exploration of our data sources, giving you insights into where our information originates, and why these particular sources are valuable for our objectives. The diversity and richness of these data sources form the foundational blocks of our system, and understanding them is key to appreciating how our system operates.

Next, we will walk you through our data extraction process. This is where we gather raw data from our various sources and prepare it for further processing. We will elaborate on the techniques and tools we use to ensure this process is both accurate and efficient.

Finally, we will detail our ETL processes, explaining how we transform raw, unstructured data into a clear, organized format, and load it into our database. The ETL process is where the real magic happens, turning scattered data into a cohesive, structured format that enables us to perform meaningful analysis and generate insightful visualizations.

4.1 Data Sources

In any data-driven project, the process of finding and identifying the right data sources is a crucial first step. To successfully extract the necessary data, we embarked on an extensive journey to uncover the most relevant and reliable sources. The process began with a comprehensive review of available online platforms specializing in computer science research. We sought out renowned academic sources, digital libraries, and prominent platforms dedicated to hosting scientific publications. This exhaustive search aimed to ensure that we could access a diverse and representative range of research works in the Portuguese context.

One of the primary sources we utilized in our quest for data was the **Digital Bibliography & Library Project (DBLP)** website. This reputable computer science bibliography website served as a valuable resource for retrieving publication information, including titles, authors, and publisher. By leveraging the extensive database of DBLP, we were able to access a vast collection of publications relevant to our research domain.

In addition to DBLP, we recognized the significance of the **Ciência Vitae** platform as a comprehensive source of information about Portuguese researchers and their scientific activities. This national platform provided an avenue to explore detailed profiles, affiliations, and publications of authors.

Other source that was provided to us, sent by Professor João Pavão Martins, a bibtex file with publications in Portugal related to AI field, and also text documents about the history of AI in Portugal, documenting its evolution and the researchers of the field.

We also recognized the potential value of incorporating Google Scholar into our data extraction process. Google Scholar, with its extensive collection of scholarly articles and publications, held the promise of providing valuable insights into the research landscape. However, when we tried to retrieve the data from this source we found out that it is protected from automating scraping, the only way to extract data from this source is using a paid API but since this project is being developed *pro bono*, it's not an option to be considered.

As a result, we focused our efforts on DBLP and Ciência Vitae, which provided substantial data for our analysis. These sources offered comprehensive information on authors, publications, affiliations, and other relevant data points, enabling us to construct a robust database..

By leveraging these data sources, we embarked on a journey to extract the data. The abundance of data from these sources laid the foundation for our subsequent data extraction, transformation, and loading processes, enabling us to build a comprehensive database to support our research objectives.

4.2 Data Extraction

The information system for supporting the collection, management, and analysis of scientific production requires a robust data extraction process. The data extraction process involves identifying the relevant data sources and extracting the necessary information from each source.

The data extraction process will be based on the data model defined in the data model section. The model defines the data elements required for the system. The data extraction process will involve developing scripts that can automatically extract the required data from online sources.

Once the data is extracted, it will be cleaned and preprocessed to remove any irrelevant or duplicate data, standardize the data format, and resolve any inconsistencies. The cleaned data will then be ready for being store in our database.

4.2.1 Author's Data

In our quest to extract data on authors, we devised a strategy that involved leveraging the network of coauthors. Our approach aimed to uncover a comprehensive list of authors by starting with a select group of authors with known DBLP pages.

To begin, we focused on these initial list of authors and examined their coauthors and checked if they had any affiliation with a Portuguese institution. If an affiliation was found, we recorded the author's information and proceeded to explore the network further by examining the coauthors of the identified authors in a recursive way, and in the end, the goal is to get a comprehensive list of the authors.

This process of traversing the network of authors proved to be a time-consuming endeavor. The script responsible for extracting the authors had to navigate through a vast network, ensuring that every

possible connection was explored.

The data extraction process of each author included extracting author's data, including their name, date of birth, birthplace, and nationality as defined in the data model.

In this case, the author's data was extracted from their DBLP page. However, only the author's name was found on it. To retrieve the remaining data, we realized that the author's page URL on could be useful for further searching on publications data.

Additionally, we also extracted the Ciência Vitae page from their DBLP page and saved it since it will be useful for extracting their affiliations history.

Therefore, the author's attributes have now been updated to include their name, DBLP page URL, and Ciência Vitae page URL. By saving these URLs, we can now access the data about the authors.

In this process, we encountered the challenge of entity recognition - ensuring that every author was uniquely identified. To address this, we discovered that each author has a unique pid, or identifier, embedded in their DBLP page URL. For instance, consider the URL 'https://dblp.org/pid/12/2004'. Here, the unique pid that identifies the author is located at the end of the URL, specifically '12/2004'.

Leveraging this unique pid, we were able to establish a distinct identity for each author and effectively solve the problem of entity recognition. This strategy enabled us to maintain the integrity of our data by preventing duplication or misidentification of authors.

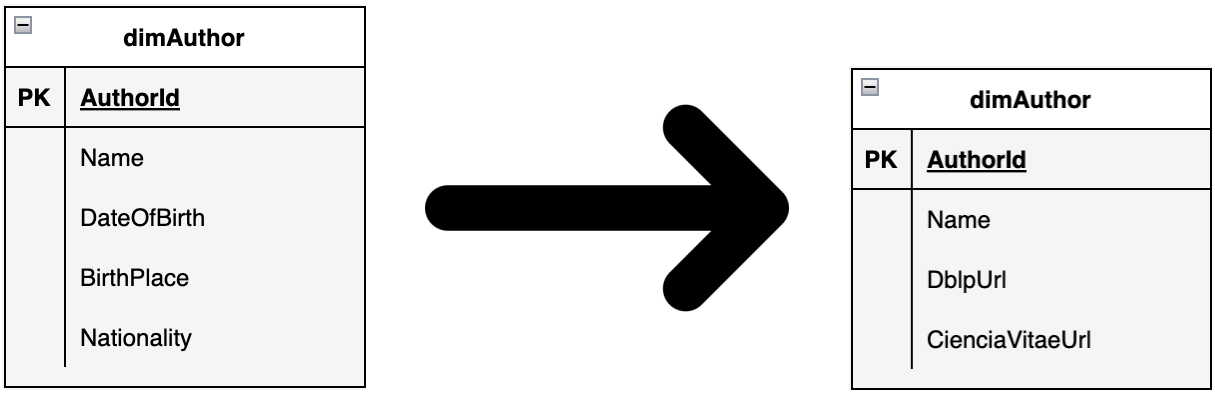

Figure 4.1: Author Dimension modification

4.2.2 Publication's Data

After extracting the author's data, including their name, DBLP page URL, and Ciência Vitae page URL, we began the process of extracting their publication data. The goal was to fill the published work dimension and publication fact with relevant data to analyze the author's research output.

We used the author's DBLP page URL to access his/her DBLP page to extract their publication data. Using scripts, we accessed the DBLP page which contains a list of all publications of the author and

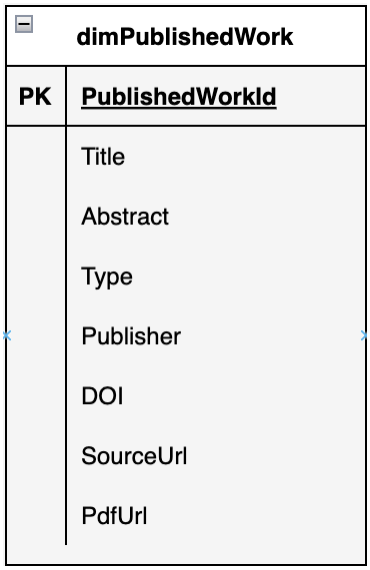

Figure 4.2: Published Work Dimension

extracted the relevant information of each publication such as the title, type, publisher, DOI, journal and year.

In some cases, we also found the publication's source URL, and its PDF URL on the DBLP page. However, this data was not available for all publications.

Using the extracted data, we filled the published work dimension with the relevant data for each publication. We then used the publication fact table to link the publications to the published work, the author, and the date. By including the published work ID, author ID, and date ID in the publication fact table, we were able to track the author's publication history over time.

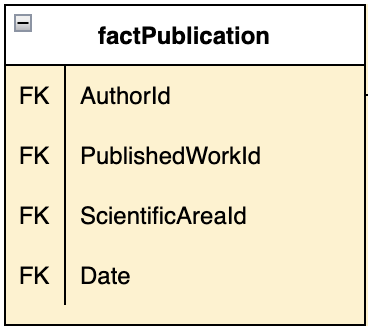

Figure 4.3: Publication Fact Dimension

4.2.3 Date Dimension

With the year data successfully extracted from each publication, we were able to proceed with populating the date dimension table. Initially, we had assumed that the granularity of the date would be on a monthly

basis. However, since the articles only had the publication year available, we needed to modify the table attributes to only include the year.

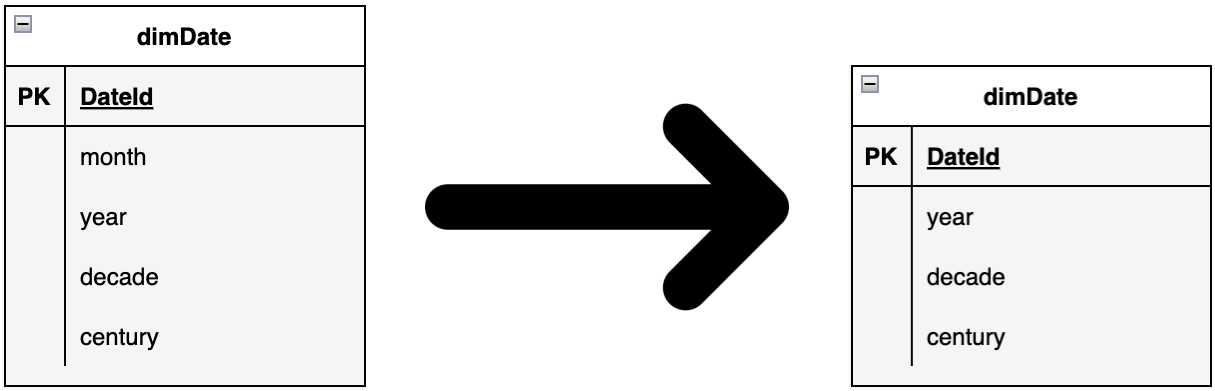

Figure 4.4: Date Dimension modification

4.2.4 Exploring Publication Details

After extracting the publication data for our research analysis, we wanted to explore further details about each publication, such as the PDF, abstract, and keywords. However, this proved to be a challenge as there were multiple source websites, and we needed to build a scrapper for each one.

To overcome this challenge, we started by identifying the most used source types for the publications. We then manually checked where we could find the useful data for each source type and built a scraping script to extract it. In the future, it would be possible to improve this scraping tool by adding more sources.

For the data scraping, we utilized the BeautifulSoup Python library. BeautifulSoup is a powerful library commonly used for web scraping purposes to pull the data out of HTML and XML files. It provides Pythonic idioms for iterating, searching, and modifying the parse tree, making it very easy to extract information from complex web pages. [22]

The process of data scraping followed a structured approach. The steps involved in the scraping process were as follows:

Initially, we verified the presence of the specific data elements we intended to extract on a given website domain. This involved exploring the website and inspecting its HTML structure to identify whether it contained the information we were seeking.

Once we confirmed the existence of the required data, we inspected the HTML structure of the website further to identify unique classes, ids, or other attributes that could aid us in pinpointing and extracting the targeted information.

With the specific elements identified, we proceeded to construct a scraper using the BeautifulSoup library. The scraper was tailored to the unique structure of each domain and designed to accurately

locate and extract the specific data elements from the website.

4.2.5 PDF URL

Our first priority was to find the PDF URL for each publication. This was crucial, as it would allow us to parse the keywords and abstract from the article itself in the next steps of our data extraction process. To accomplish this, we systematically checked each publication's source URL for a direct link to the PDF file. If a direct link was available, we extracted it and saved it in our database.

4.2.6 Keywords

In the next step of our data extraction process, we focused on extracting keywords from each article. We began by attempting to extract the keywords from the source URL. If the keywords were not found in the source URL, we attempted to extract them from the PDF.

To extract keywords from the source, we utilized the same strategy as we did for the PDF URL extraction. We manually checked each source to determine if the keywords were present. If they were, we built a script to extract them.

If we were unable to find the keywords in the source, we attempted to parse them from the PDF, but this was a more complex process. First, we downloaded the PDF and used a library to read it and convert it to a text format. Next, we searched for the word ”Keywords:” in the text and cut everything in the back. Then, we split the text by ”,” or ”;” to create an array and selected the array indexes until we found a line break or if a position had more than 4 words.

To ensure the effectiveness and accuracy of our keyword extraction strategy, we employed a series of unit tests using 13 PDFs as test cases with and without keywords. Our goal was to validate the functionality of the extraction method and identify any areas for improvement.

Out of the 13 PDFs, our strategy was successful for 12 of them. Only one PDF presented challenges and was not successful in extracting the keywords.

Despite encountering some challenges in extracting keywords from all 13 PDFs, our overall success rate has proven to be high, indicating the effectiveness of our approach for keyword extraction from a large volume of articles. This is a critical step in organizing, categorizing, and analyzing information, which can lead to more insightful conclusions and better decision-making.

Once we have extracted the keywords, we take an additional step to clean and format them for storage in the database. We capitalize the keywords and remove any unexpected characters or formatting that may have been present in the source material. For example, we encountered keywords such as ”Artificial Intelligence (Cs.Ai),” which contained unexpected characters. To ensure consistency and accuracy, we removed the unnecessary ”(Cs.Ai)” portion using a regular expression before storing it in the

database.

This cleaning and formatting process allows us to maintain a high level of data quality and consistency across all keywords, which is essential for accurate analysis and reporting. Additionally, it streamlines the process of storing and managing the extracted keywords, making it easier to access and utilize this information for future projects.

Overall, our approach to keyword extraction from articles and PDFs involved a combination of techniques, including review of the source URL, parsing of PDFs, and rigorous testing. By employing these methods, we were able to streamline the categorization process and provide valuable insights for our project.

4.2.7 Abstract

Expanding on our efforts to streamline the categorization and analysis, we identified a potential source of valuable information - article abstracts that are available on some websites. To incorporate this data into our analysis, we developed a strategy for identifying and extracting abstracts from these sources.

To identify article abstracts, we used the same technique as we did for the PDF URLs and keywords extraction, checking each website domain to see if an abstract is present. This approach involves scraping each website and checking for the presence of an abstract, which can then be extracted and stored along with the keywords for further analysis.

While we explored the possibility of extracting article abstracts from PDFs, we found that it was challenging to accurately identify where the abstract ends. PDFs often have different layouts and formats, making it difficult to create a standardized approach to extracting abstracts. In addition, abstracts in PDFs may be embedded within the text of the article, making it harder to isolate and extract them.

Given these challenges, we decided to focus on extracting abstracts from website domains where the abstracts were more easily identifiable and consistently formatted. By using a targeted approach, we were able to achieve a high degree of accuracy and reliability in our abstract extraction process.

To ensure the effectiveness and accuracy of our scraper, we developed a set of seven unit tests to validate the parser's functionality. These tests have been successful, confirming the reliability of our approach to identifying and extracting article abstracts.

By incorporating article abstracts into our analysis, we are able to gain a more comprehensive understanding of the content and context of each article, enabling us to draw more informed conclusions and make more impactful decisions. Additionally, our rigorous testing and validation process ensure the accuracy and reliability of the data, providing a strong foundation for future analyses and initiatives.

Overall, our efforts to identify and extract article abstracts have proven to be successful, expanding the scope and depth of our data analysis capabilities.

4.2.8 Affiliations

With the extraction of authors' and publications' data completed, we moved on to extract affiliations' data for each author. Our goal was to fill our affiliation fact table and institution dimension table with this data.

To extract this data, we turned to the ciência vitae page of each author. Here, we could successfully extract the year interval in which an author was associated to an institution. This year interval and the corresponding institution is considered as an affiliation, and our goal was to extract all affiliations of each author.

With this data, we were able to fill our affiliation fact table and institution dimension table with the relevant data. However, in the institution dimension table, we could only extract the name of the institution, in the next step we tried to retrieve other institutions details. It's worth noting that we did not make any changes to the affiliations and institution tables at this point.

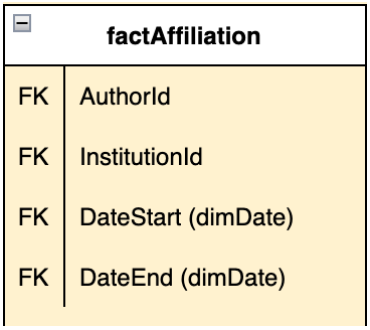

Figure 4.5: Affiliation Fact

4.2.9 Institution Dimension

After extracting data on authors, publications, and affiliations, our next step was to extract data on institutions to fill the institution dimension table. However, we faced a challenge in automating the process of extracting this data. Additionally, we realized that the details of an institution were not crucial to the goals of this project.

As a result, we decided to simplify the institution dimension table by removing its attributes and focusing solely on the name of the institution. This change allowed us to streamline our data processing and focus on the more critical aspects of the project.

4.2.10 Phd Fact Table

We had initially planned to extract data related to the PhD's article of each author and create a fact table for this purpose. However, despite our best efforts, we were unable to find a reliable source of

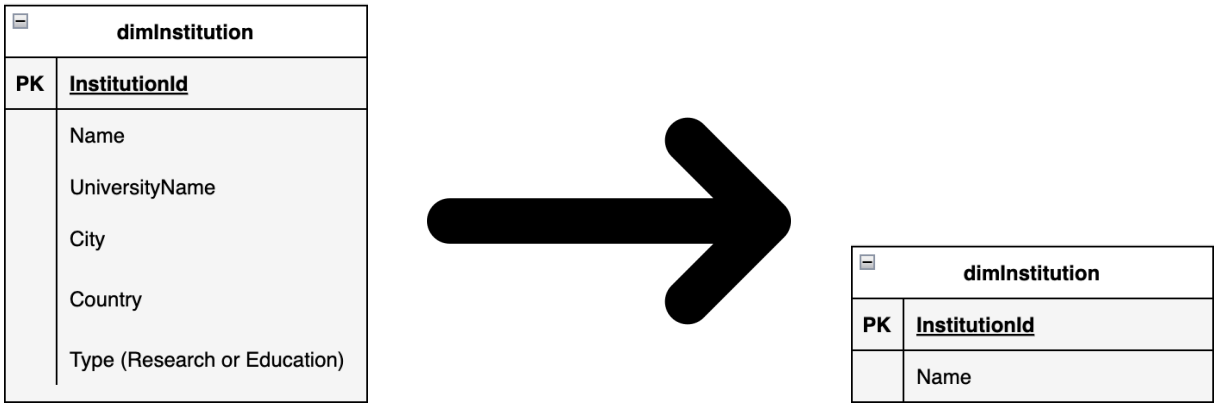

Figure 4.6: Institution Dimension modification

information that could provide us with this data. We searched extensively through various sources, including the CiênciaVitae page, DBLP page and the publications themselves, but to no avail. As a result, we were unable to fill the PhD fact table with any meaningful data, and had to remove it from the data model.

4.2.11 Scientific Area Dimension

In our effort to extract as much relevant data as possible for our research analysis, we also attempted to retrieve information for the Scientific Area dimension table. However we couldn't find the scientific area of the publications on the available sources. Therefore, we were forced to remove it from the data model. It's important to note, though, that while this data was not readily available, we intend to undertake a mining process in the future, which aims to unearth this data and enrich our data set further. This represents an avenue for potential improvement and expansion in our system's data gathering capabilities.

4.2.12 Data model after extraction

In conclusion, after the data extraction process was completed, it was necessary to make changes to the database schema. These changes were made to ensure that the schema accurately reflected the data extracted and was optimized for the analysis of the data. The following changes were made to the database schema:

The **dimAuthor** table was modified to remove the columns for DateOfBirth, BirthPlace, and Nationality, and new columns for DblpUrl and CienciaVitaeUrl were added.

The **dimDate** table was modified to remove the month column.

The **dimPublishedWork** table was modified to add a new column for the journal.

The **dimInstitution** table was modified to only have the name of the institution.

The **dimScientificArea** table was removed because there was no relevant information on the publications that could be used to populate this table.

The **factPhd** table was removed because there was no source of data available to populate this table.

The **factKeyword** table was modified to add a foreign key to the dimDate table. This change was made because the date was needed to generate the word clouds, which were essential to the project's goals.

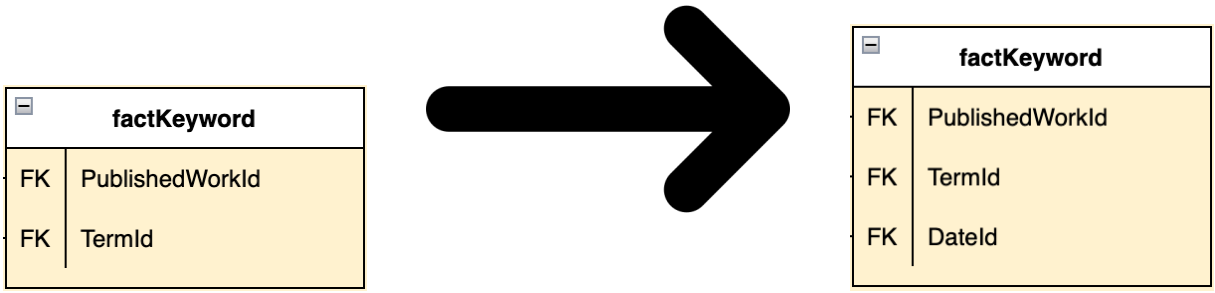

Figure 4.7: Keyword fact modification

The final data model is presented in Figure 4.8.

Overall, these changes would ensure that the database schema was efficient and effective in supporting the project's goals.

4.3 Processes

An ETL process is a fundamental part of any data warehousing project. It refers to a set of procedures used to integrate data from various sources, transform it into a consistent format, and load it into a target database. In the case of this project, the ETL process was initiated after the extraction parsers were completed. The aim was to extract and load the initial data to create the data model for the project. The ETL process includes several sub-processes, which are also meant to be reused to update or add data over time. The following is the definition of each ETL process.

The first ETL process (Figure 4.9) involves extract the authors and its pages, the Dblp page and Ciência Vitae page, using the strategy described on the author's data section of Data model extraction. The JSON file used in the start only has some authors to start the extraction and this data will be stored on another JSON file in order to be able to add authors manually or edit it if required

The second ETL process (Figure 4.10) involves loading author data from the JSON file result from the previous etl process. The JSON file contains a list of authors, where each author has their name,

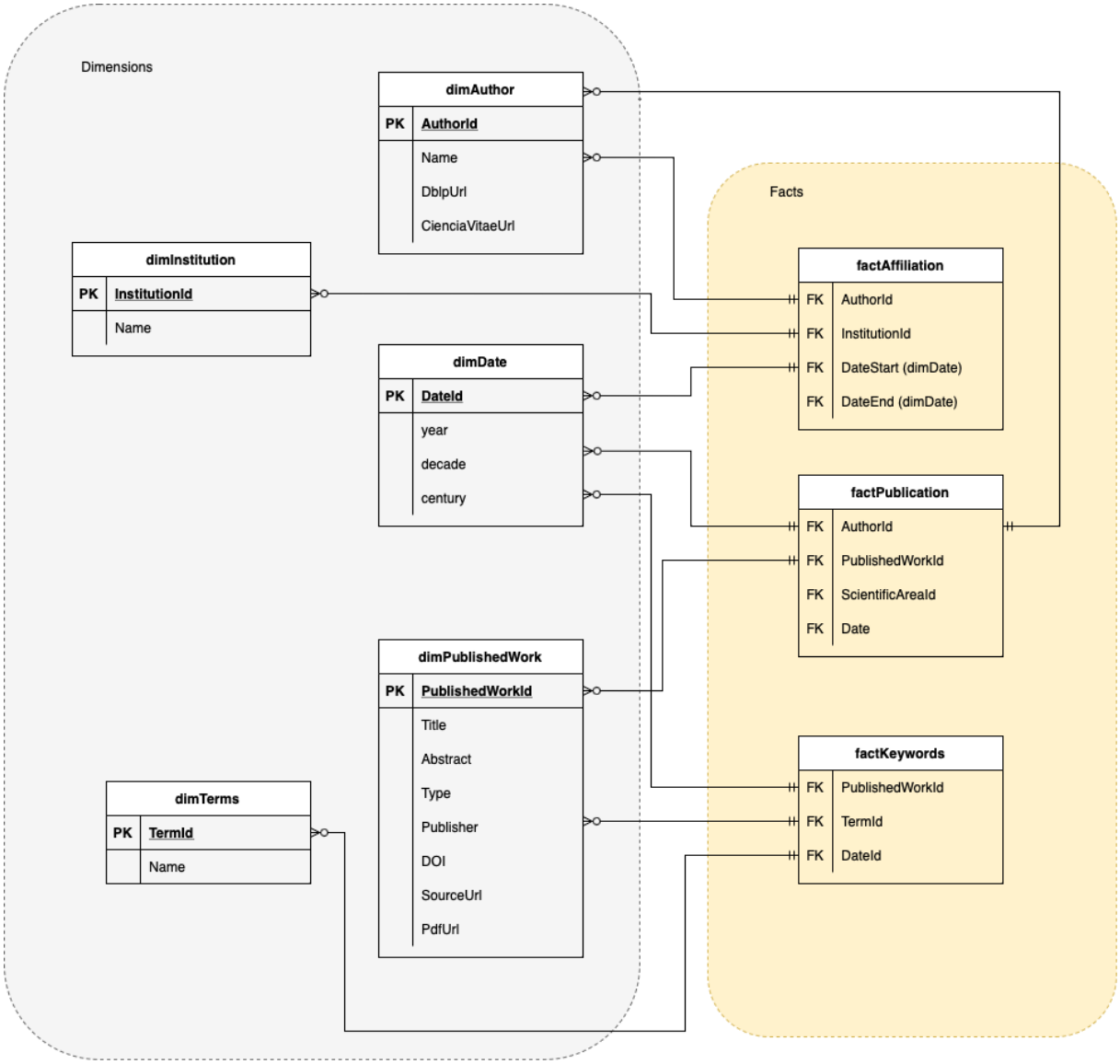

Figure 4.8: Data model after extraction

Figure 4.9: First ETL Process - get authors

dblp page URL, and cienciavitae page URL. The process of loading this data involves extracting the information from the JSON file and transforming it into a format suitable for storage in a database. Once the data is transformed, it is loaded into the database, making it available for further processing.

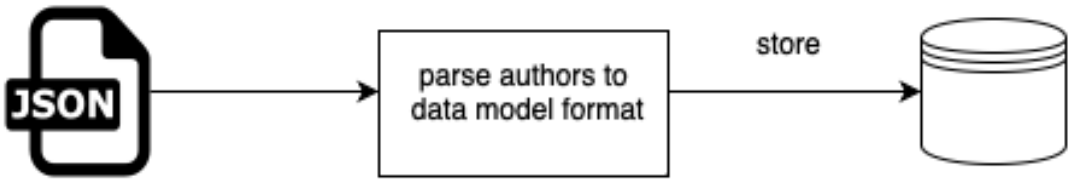

Figure 4.10: Second ETL Process - store authors

The third ETL process (Figure 4.11) involves loading articles from the dblp page of each author and storing them in the database. This process is crucial in gathering all the relevant publications of an author in a single location. By loading articles, the system can create links between authors and their publications.

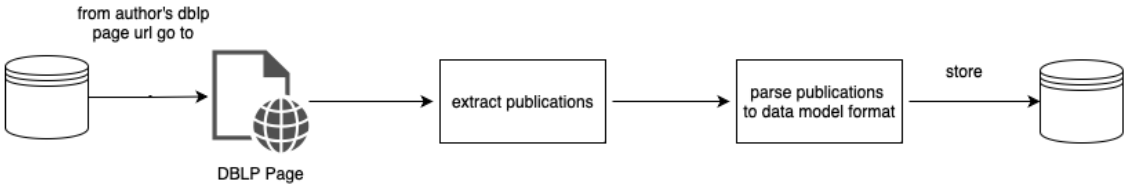

Figure 4.11: Third ETL Process - get publications

The Fourth ETL process (Figure 4.12 involves loading author affiliations from the cienciavitae page of each author and storing them in the database. This process is vital in establishing the author's organizational affiliations and academic background, which can be essential in evaluating their research work.

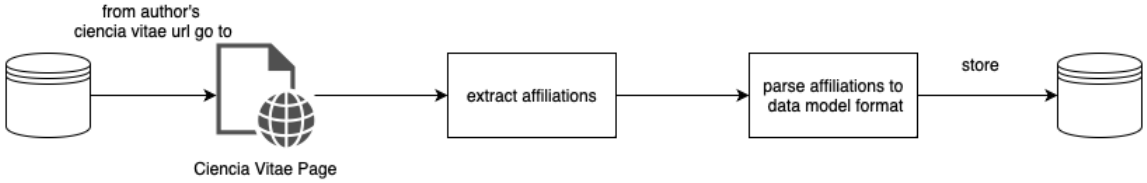

Figure 4.12: Fourth ETL Process - get affiliations

The Fifth ETL process (Figure 4.13) involves loading published work details, aiming to extract more data from the publications already stored in the database. These details include pdf URLs, keywords, and abstracts, which can provide additional insights into the publications and their authors. This information is crucial in identifying research trends and evaluating the impact of a particular publication.

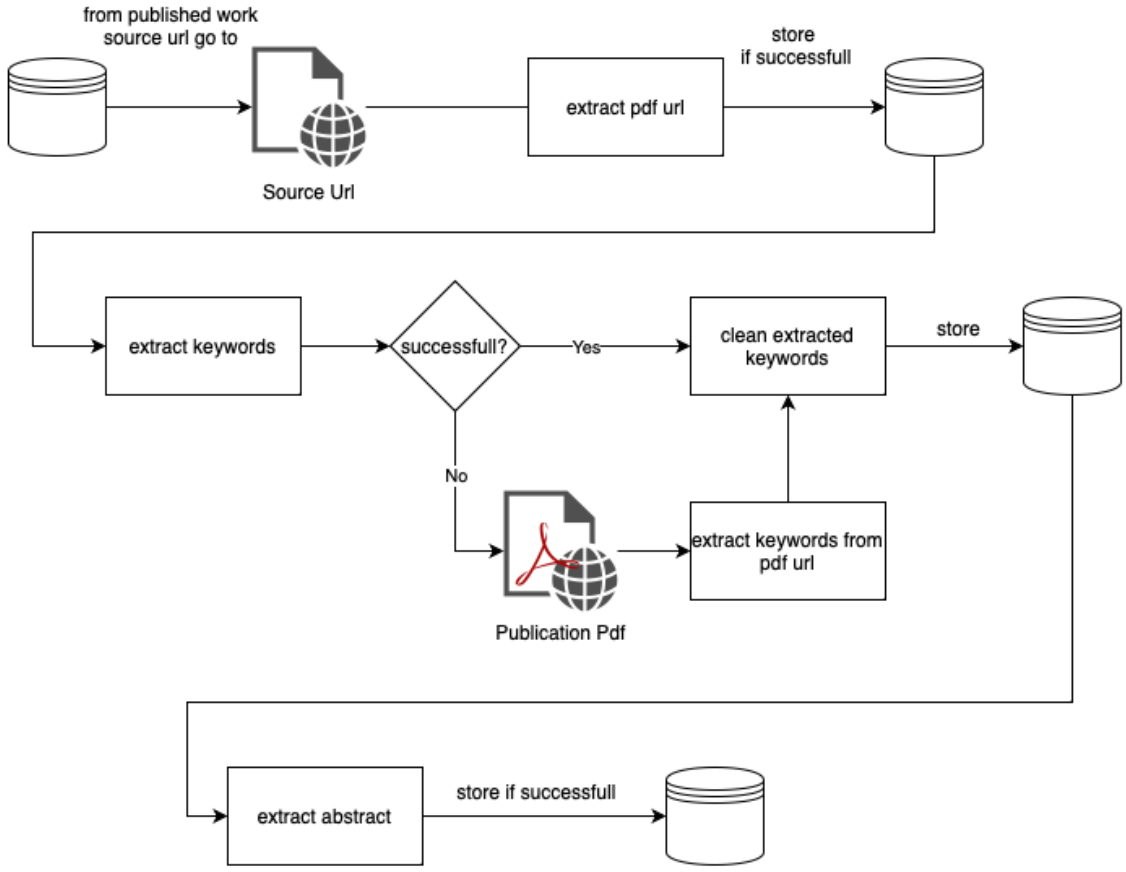

Figure 4.13: Fifth ETL Process - get publications details

Overall, the ETL process plays a critical role in ensuring that the extracted data is transformed into a consistent format and loaded into the target database. By utilizing this process, the data model can be created, and the data can be stored efficiently, enabling researchers to query and analyze the data effectively. Also, this etl process sequence not only will be necessary to store the initial data but also to update and add data regularly, therefore they're designed to not duplicate data on the database even when they run multiple times.

5

Data Science

Contents

In this chapter, we delve into the comprehensive data analysis conducted in this project. Our approach incorporated a range of techniques including data summarization, author and keyword rankings, word clouds, co-authorship network graphs, and data mining. Each method provided distinct, yet interconnected insights into our dataset, offering a multi-faceted understanding of the underlying patterns and relationships. In the following sections, we explore each of these techniques in detail, elaborating on their implementation and the valuable information they revealed.

5.1 Summarization

Data summarization served as the cornerstone of our initial analysis, granting us a valuable overview of the data encapsulated in our tables. This stage facilitated the inspection of fundamental aspects of our data such as the number of rows, occurrence of duplicates, and presence of missing values, complemented with an evaluation of their percentages.

Moreover, we scrutinized the distribution of data across what we identified as significant variables. For example, the distribution of publications was explored by year and by decade, exposing potential temporal trends and patterns within the data.

We offered a representative sample of data from each table to provide a snapshot of our data.

All the insights derived from the data summarization process were visualized and shared using Metabase [13], our data analytics tool. This tool enabled us to present our findings in a clear, interactive format, which facilitated better understanding of the patterns and trends we identified in the data. While implemented through Metabase [13], this tool provides an interface to accomplish a full OLAP analysis.

To manifest these charts and dashboards, we needed to use SQL queries within Metabase [13]. As illustrative examples, we will present some of these queries - namely those that were utilized to ascertain the number of authors, identify missing values of the authors dimension table, and enumerate the number of publications per year.

Number of authors:

```
SELECT count(*) FROM dim_authors;
```

Missing values of the authors dimension table:

```
SELECT
  'total_records' as attribute,
  COUNT(*) AS count
```

```
  FROM
    dim_author

  UNION ALL

  SELECT
    'name',
    SUM(CASE WHEN name IS NULL OR TRIM(name) = '' THEN 1 ELSE 0 END)
  FROM
    dim_author

  UNION ALL

  SELECT
    'dblp_page_url',
    SUM(CASE WHEN dblp_page_url IS NULL OR TRIM(dblp_page_url) = '' THEN 1 ELSE 0 END)
  FROM
    dim_author

  UNION ALL

  SELECT
    'ciencia_vitae_url',
    SUM(CASE WHEN ciencia_vitae_url IS NULL OR TRIM(ciencia_vitae_url) = '' THEN 1 ELSE 0 END)
  FROM
dim_author;
```

Number of Publications per Year:

```
  SELECT
    d.year,
    COUNT(DISTINCT p.published_work_id) AS unique_published_work_count
  FROM publication p
  JOIN dim_date d ON p.date_id = d.id
  GROUP BY d.year
  ORDER BY d.year ASC;
```

By executing a detailed data summarization, we went beyond mere observation and ventured into an intricate analysis of the data. By making these assessments, we were able to expose trends and patterns that would remain hidden within the raw data, enabling us to comprehend the data's complexities on a deeper level.

5.2 Rankings

In our analysis, we also applied various ranking techniques to highlight the most prolific entities in our data.

Our rankings examined a variety of dimensions, such as identifying authors with the highest number of publications. We also evaluated these prolific authors based on the volume of their work across different centuries and decades. In addition to authors, we highlighted the decades and years that saw the highest volume of publications. Finally, we assessed keyword usage, pinpointing the most frequently used keywords and their prevalence over different centuries and decades.

Each ranking was produced by executing SQL queries that utilized aggregation operations, predominantly using the 'Count' aggregate function.

Authors with more Publications query:

```
SELECT name as "Authors", Count as "Number of Publications"
FROM (
    SELECT author_id, count(*)
    FROM publication
    GROUP BY author_id
) as f
JOIN dim_author da ON da.id = f.author_id
ORDER BY Count desc;
```

To visualize these rankings, we employed Metabase as well. This platform allowed us to present our findings using horizontal bar charts, providing an intuitive and interactive way to digest the data. The graphics not only emphasized the ranking order but also illustrated the relative differences between each data point.

Through these rankings, we are able to identify the most influential authors, the most active periods for publications, the most prolific publishers, and the most commonly used keywords in our dataset. This exploration provided an additional layer of understanding about the trends and patterns encapsulated within our data.

5.3 Word Clouds

In our journey to unearth the hidden patterns within our data, we leveraged the power of visualization, specifically through the creation of word clouds. Word clouds provided an efficient and visually appealing way to depict the frequency of keyword usage. We constructed word clouds for all keywords and additionally partitioned them by century and decade to examine the shift in keyword trends over time.

The data used to create these word clouds were extracted using SQL queries, which fetched the keywords and their occurrence frequency from the keyword fact table for the respective timespan. Post extraction, the data underwent a transformation process to fit the required format for creating the word cloud, which resembled a list of dictionary structure, with each keyword paired with its total number of occurrences, for instance, [{"word": "keyword1", "count": 6}, {"word": "keyword2", "count": 3}].

The execution of these processes was handled by our server, which interfaced directly with the database. To visualize the word clouds, we employed Angular [15] coupled with the angular-d3-cloud library [23] on our website. The data required for these visualizations was made accessible to the website through an API endpoint on our server, ensuring a seamless flow of information from the data warehouse to the user-facing interface.

The creation of these word clouds not only enhanced the aesthetic appeal of our data presentation but also facilitated a quick, comprehensive overview of the keyword frequency, offering another avenue to comprehend the evolution of trends captured within our data.

5.4 Co-Authorship Network

In our exploration, we ventured into network analysis with the intent to construct a co-authorship network graph. This visualization provides a graphic representation of collaborative relationships between authors, helping us understand the underlying collaborative structures and dynamics.

We built the co-authorship network using the NetworkX library on the server. NetworkX is a Python package that simplifies the creation, manipulation, and study of complex networks.

The process involved retrieving data from the publication fact table to identify pairs of authors who had collaborated on the same published works. If, for instance, Author1 and Author2 had three published works in common, we would include three tuples of (Author1, Author2) in our list of pairs. This format is directly compatible with NetworkX, which made it straightforward to import our data into the graph structure.

Once the data was represented as a graph using NetworkX, we visualized and saved the resulting co-authorship network graph on our server. This graphical depiction provided a clear and succinct way to understand co-authorship patterns among the authors, facilitating deeper insights into the nature and extent of scholarly collaboration.

5.5 Labelling authors per field

This section is dedicated to elucidating the process of labeling authors based on their specific research interests, extracted from the keywords associated with their publications. Distinguishing authors into distinct categories of research fields presents a detailed overview of the research landscape, allowing us to identify the experts in each domain.

To determine this, we will be employing machine learning to assist us in this task, specifically, the unsupervised learning technique known as K-means clustering [24]. This method allows us to assign authors into clusters, or groups, where each group represents a specific field of research.

In order to compile our initial dataset for the labeling study, we embarked on an extraction process that centered around the authors dimension table. Our primary objective was to ascertain the frequency of keyword usage for each author relative to their total number of publications. This was essential to ensure that our data was normalised and, thus, would yield more accurate and reliable results in the subsequent clustering process.

We initiated this process by extracting all the authors listed in the authors dimension table. For each author, we executed an SQL query to determine the frequency of each keyword they used, dividing this by their total number of publications. This gave us a frequency percentage for each keyword per author.

The aim was to then create a data structure where each row represented an author (identified by their ID), the columns represented the keywords, and the values indicated the frequency percentage of the keywords. This structuring of our dataset facilitated a more streamlined and efficient process when conducting the clustering analysis. A portion of the data is presented on Figure 5.1.

	id	Signal Processing	Biometry	Electrocardiography	Biosignals	Meanwave	Knowledge Acquisition System	Classification	Ontology
0	1	0.33333	0.33333	0.33333	0.33333	0.66667	0.33333	0.33333	0.33333
1	2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	8	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	9	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	10	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
10	11	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
11	12	0.04762	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.14286
12	13	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
13	14	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
14	15	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.03636
15	16	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.04167
16	17	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.10667

Figure 5.1: Portion of the Initial Dataframe

Following data extraction, the next crucial step was pre-processing our dataset. This process was geared towards refining our data and ensuring it was in an optimal state for further analysis and model training. To guide this process and benchmark our progress, we used K-means clustering as our primary algorithm.

Pre-processing involved the implementation of various techniques such as Principal Component Analysis (PCA) [25] and feature selection. Our objective was to identify and select the most informative attributes in the dataset, thereby enhancing the performance and precision of our subsequent K-means clustering.

Specific strategies employed during this stage involved removing keywords with higher variance, as these could skew our results by disproportionately influencing the clustering process. Similarly, we aimed to remove keywords based on their correlation, as highly correlated keywords can introduce redundancy into our model, detracting from its efficiency and effectiveness.

Throughout this process, our goal was to identify the pre-processing techniques and parameters that yielded the best silhouette score in our K-means clustering. The silhouette score serves as a key metric that assesses the quality of the clusters produced by the algorithm. By striving for the optimal silhouette score, we were not only able to refine our pre-processing techniques, but also pinpoint the ideal number of clusters for the dataset.

Having identified the optimal pre-processing settings, we proceeded to apply the K-means clustering algorithm to the dataset. The outcome of this process was a set of clusters, with each author assigned to a specific cluster based on their keyword frequency. However, a significant challenge arose at this point – the cluster labels generated by K-means were essentially numerical and lacked any inherent meaning or description.

To address this, we devised a strategy to assign a name to each cluster. We first extracted the centroids of each cluster, which are essentially the mean of all the data points within a particular cluster. The centroids, in our context, represented the 'average' keyword frequency of each cluster. We then identified the two keywords with the highest values in each centroid. These two keywords, we hypothesized, had the most influence in the assignment of authors to each cluster.

Thus, our method transformed the otherwise abstract cluster labels into something more insightful and intuitive. Each cluster was now represented by the two most influential keywords, shedding light on the dominant themes or areas within each cluster. This, in turn, allowed us to discern the likely field of research for each author based on their cluster assignment.

6

Case Study

Contents

In this chapter, we will delve into the heart of the system's operation, illuminating the practical execution of the strategies and methods previously discussed. Our exploration will take us through each step of the process, from the extraction of raw data to its subsequent profiling and analysis.

We will begin by outlining the methods and techniques utilized for data extraction, and how this data was organized to populate the database. The journey will continue with a detailed profiling of the extracted data, highlighting its structure, attributes, and overall characteristics. Further analysis of this data will provide us with deeper insights and facilitate a more comprehensive understanding of the data.

The chapter will then transition into a discussion of the data analysis process, demonstrating the application of the various analytical tools at our disposal. This will include the rankings, creation of keyword word clouds, the construction of the co-authorship network, and the classification of authors into specific fields based on their publication keywords.

Also it's important to note that the examination of user interaction quality was not included within the scope of this project. However, it is recommended that this aspect is addressed in a subsequent version of the system.

The ultimate aim of this chapter is to provide a practical demonstration of the system at work, showing not only how it operates, but also the value and insights it can generate from the data.

6.1 Filling the database

The journey to populate our database was an exhaustive process that commenced with data extraction, as detailed earlier. We initiated this task with a list of authors, adopting a recursive approach to trace their network of co-authors. This strategy allowed us to collate an extensive dataset representing the interconnections within the AI area.

In this case study, we began with a selection of 17 authors and their DBLP page url, selected due to their pioneering contributions to AI research in Portugal. This initial list served as the starting point from which we explored the network of co-authors, using the technique described in Section 4.2.1, aiming to capture a comprehensive list of authors that reflects the landscape of AI research in Portugal.

Once this complete list of authors was gathered, we triggered the series of ETL processes, outlined in the Section 4.3. These processes were integral in ensuring that the raw data was transformed and structured properly for seamless integration into our database.

The magnitude of this undertaking is best understood when considering the cumulative execution time of all ETL processes, which totaled approximately 34 hours. This time frame is a testament to the complex and voluminous nature of the work carried out.

This extensive execution time underscores the heavy lifting carried out in the extraction, transformation, and structuring of data to ensure it is in the right form for successful integration into our database.

It illustrates the substantial effort and resources dedicated to this critical phase of the project, which formed a significant portion of the overall work. The long execution time, thus, also reflects the depth and detail of the data that we aimed to capture in order to create a robust and comprehensive data model.

Now with the data extracted and stored, we will focus on an essential step, profiling the data. Now that we've gathered our data, we are poised to assess and understand its content and quality. Through data profiling, we delve into each table of our database, investigating the specific attributes and nuances of our collected data.

The process doesn't just involve statistical evaluation; it's an exploratory journey into the essence of our data. It allows us to gauge its integrity, detect anomalies, and ensure it is suitable for the tasks ahead.

In our exploration, the Metabase tool [13] will assist us in digesting the information at hand, allowing us to glean valuable insights about the overall structure and individual variables within our data. The analysis and discussion of each table will shed light on our data's strengths, weaknesses, and potential, shaping our understanding and strategies for further data operations.

Starting with author's dimension table (dashboard on Figure 6.1). The data extraction technique, as previously described, enabled us to acquire a robust data pool consisting of 1196 authors.

Our exploration of the data reveals varying degrees of data completeness. All authors are accounted for in terms of their names and DBLP pages. However, only about 58% of the authors have an available Ciência Vitae URL, a valuable data point that furnishes us with data on affiliations. Therefore, affiliations data will be missing for at least the remaining 42% of authors who lack this URL.

Nonetheless, the absence of duplicate entries is a promising sign. The zero duplicate count aligns with the nature of dimension tables, which should, ideally, only contain unique instances.

Although the data is not perfect, the breadth of the Author dimension table is quite commendable. The sizable amount of author data collected can adequately mirror the landscape of AI research in Portugal. It could offer a rich source for analysis, capable of yielding meaningful insights into the domain we are studying.

When examining the Institution dimension table (dashboard on Figure 6.2), we identified 292 different entries. It's important to clarify that the number of institutions we have in our data exceeds the total number of actual institutions in Portugal by a considerable margin.

This discrepancy is attributed to the fact that our data not only encompasses the overarching institution but also captures the specific departments within each institution. For instance, we have distinct entries for 'Universidade Nova de Lisboa, Portugal' and 'Universidade Nova de Lisboa Departamento de Física, Portugal'. Such granularity might be due to the potential mobility of researchers between different departments within the same institution over their academic careers.

Figure 6.1: Dim Author Profiling Dashboard

Additionally, we also uncovered the presence of foreign institutions from countries such as France, Australia, and the UK within our data, this can be interpreted as collaborations with foreign researchers that are doing internships in Portugal.

Regarding missing values, our data integrity is commendable as all institutions have corresponding names. Furthermore, we identified no duplicates, which aligns with our expectation for a dimension table that should ideally consist of unique records.

Assessing the overall quality of the Institution dimension, it falls short of our initial expectations. The failure to isolate overarching institutions from their constituent departments hinders our ability to gain a clear understanding of research contributions at the institutional level. Therefore, the utility of the current form of this table for our purposes is limited.

One of the tasks to do as future work, is to create a taxonomy for the institution dimension, where each one can belong to a larger unity, as for example an University.

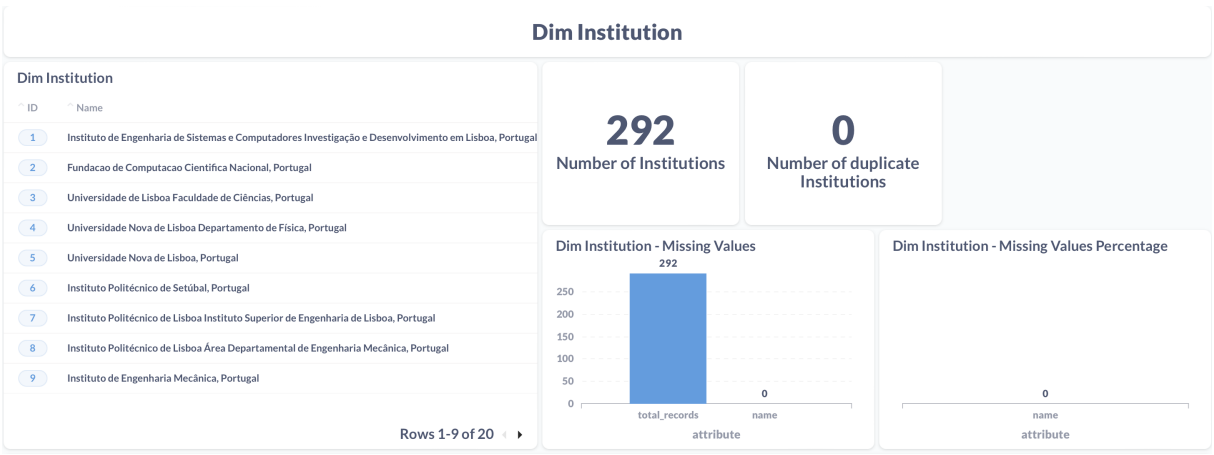

Figure 6.2: Dim Institution Profiling Dashboard

Upon reviewing the Date dimension table (dashboard on Figure 6.3), we find it contains a total of 51 entries, which correspond to a span of 51 years. The earliest year recorded in our data is 1974, while the later is 2035.

This presence of the years 2025 and 2035 is puzzling. We do not expect to encounter such future dates in our data. These entries could be attributable to errors or typos during the data entry process. Alternatively, they could potentially denote end dates for certain affiliation contracts.

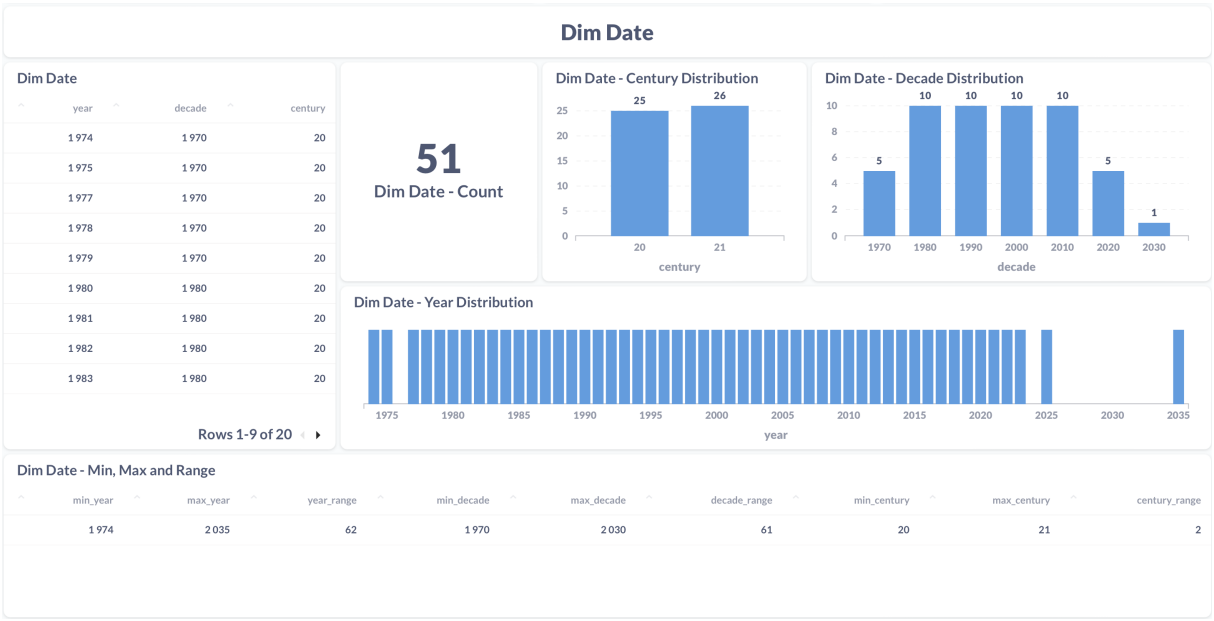

Figure 6.3: Dim Date Profiling Dashboard

When it comes to profiling the Published Work dimension table (dashboard on Figure 6.4 and Figure 6.5). Our data houses a significant collection of 29120 published works, providing a substantial basis for exploring and understanding the landscape of the field. This data, free from any duplicate entries, is a testament to the success of our data extraction and aggregation methods.

Although we observe a sizable proportion of missing values, particularly in the fields of abstracts and PDF URLs, which stand at approximately 64%, this does not substantially impact our objectives. The absence of these details does not significantly hamper our analytical processes as the source URLs contain the necessary keyword data. Furthermore, our research goals do not directly depend on the availability of the PDFs, nor the presence of the abstracts.

In terms of data quality, we consider the data to be largely satisfactory for our current objectives. While there are noticeable gaps, these do not hinder our overall analysis. The areas with missing data could also be potentially addressed in future iterations of this project by implementing additional scrapers, thereby improving the comprehensiveness of the data.

In our Term dimension table (dashboard on Figure 6.6), we have identified a substantial total of 24703

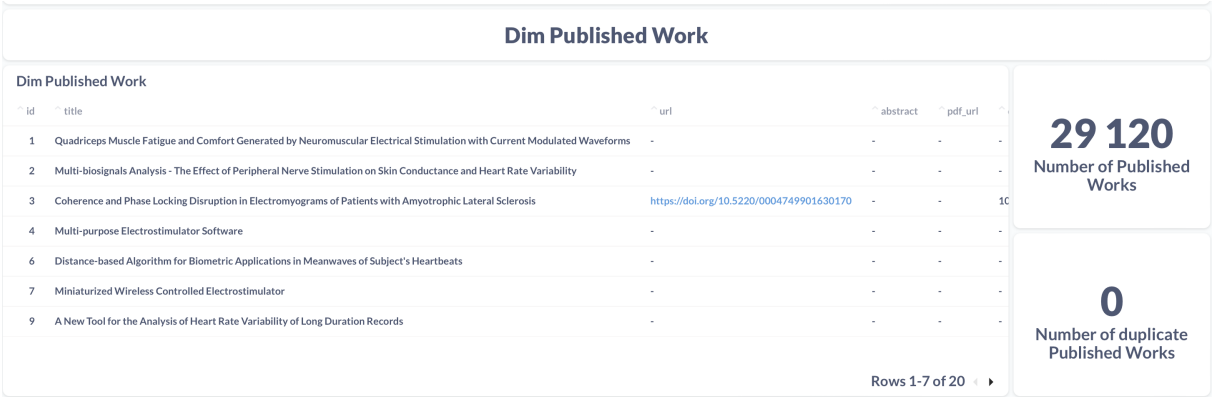

Figure 6.4: Dim Published Work Profiling Dashboard

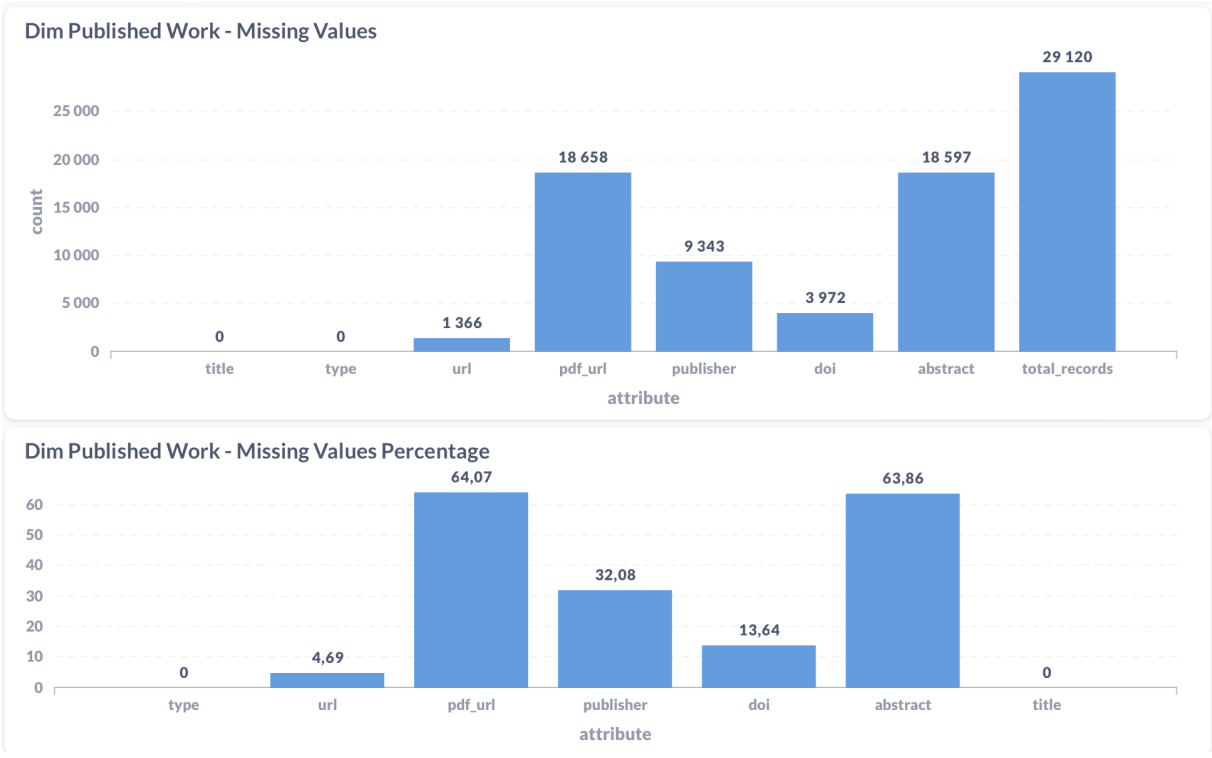

Figure 6.5: Dim Published Work Profiling Dashboard Missing Values

distinct terms. This rich collection of terms plays a crucial role in our upcoming author classification task and word clouds formation, where it will allow us to draw robust and comprehensive conclusions.

The table contains no duplicate entries. Moreover, no missing values were detected in this dataset since each record is represented solely by the term name. This clean, expansive, and fully populated table aids in reinforcing the quality and reliability of our dataset for further stages of analysis.

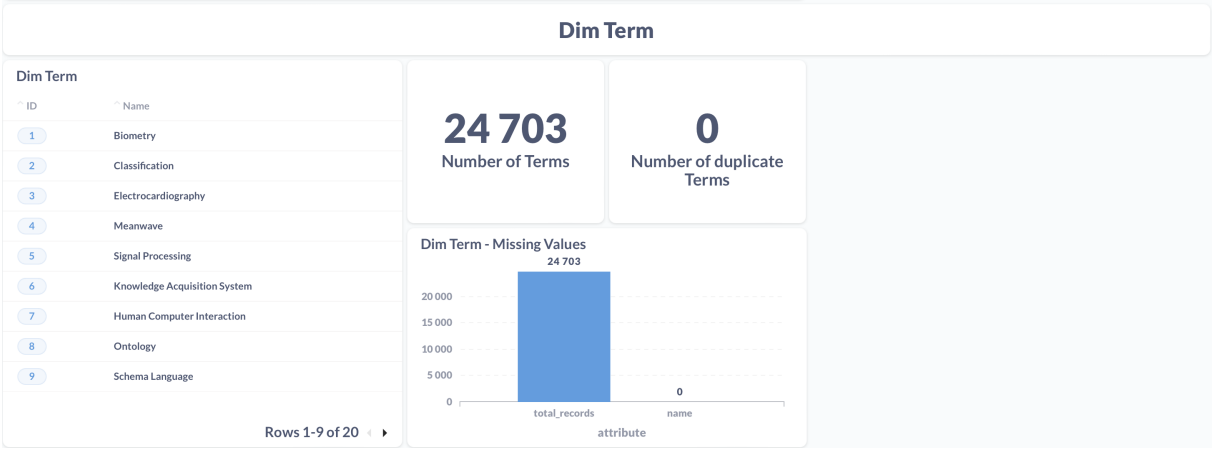

Figure 6.6: Dim Term Profiling Dashboard

In examining our Publication Fact table (dashboard on Figure 6.7 and Figure 6.8), it becomes evident that the table houses a considerable number of entries, totaling up to 42411. This table encapsulates the key facts related to authors, published work, and dates, painting a comprehensive picture of the research landscape.

Through the analysis of this table, we can trace the progression of unique published works across various time frames - centuries, decades, and years. As we observe, there has been a general increase in the volume of publications over the years, mirroring the growth and evolution of the field.

Further, our table is free from any duplicates. Moreover, we can derive certain averages from our data that provide intriguing insights. For instance, the average number of published works per author is estimated to be around 35 to 36. Similarly, the average number of authors per published work is found to be between 1 and 2. This suggests that a significant portion of research is conducted either independently or in small teams.

As we delve into the Affiliation Fact table (dashboard on Figure 6.9 and Figure 6.10), it is observed that this table houses 686 entries, free of any duplicates.

However, a significant observation from our analysis is the dearth of affiliations for a large proportion of our authors. Specifically, we lack affiliation data for approximately 883 authors out of the total 1196. This represents almost 74% of our author population - a quite substantial percentage. Given the scale of missing data, it becomes evident that attempting to glean insights or draw conclusions based on affiliations could potentially lead to skewed or misleading results. Therefore, we acknowledge that our

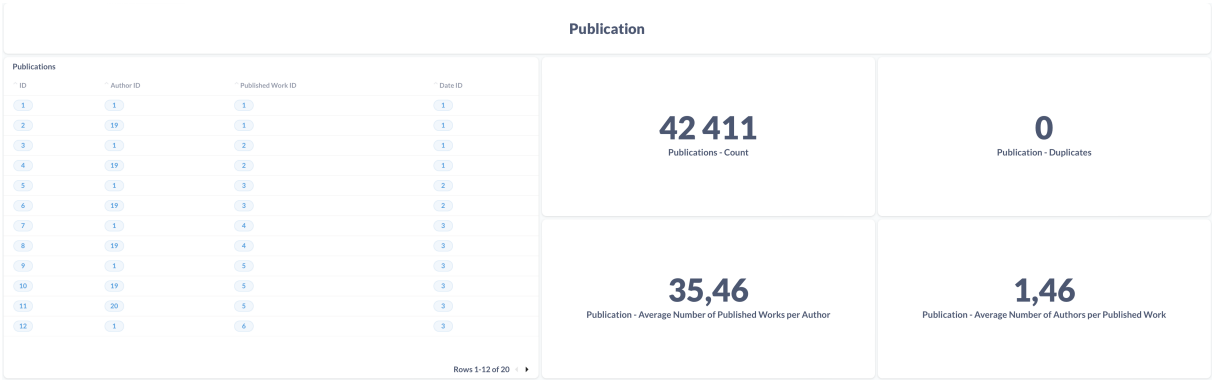

Figure 6.7: Publication profiling part 1

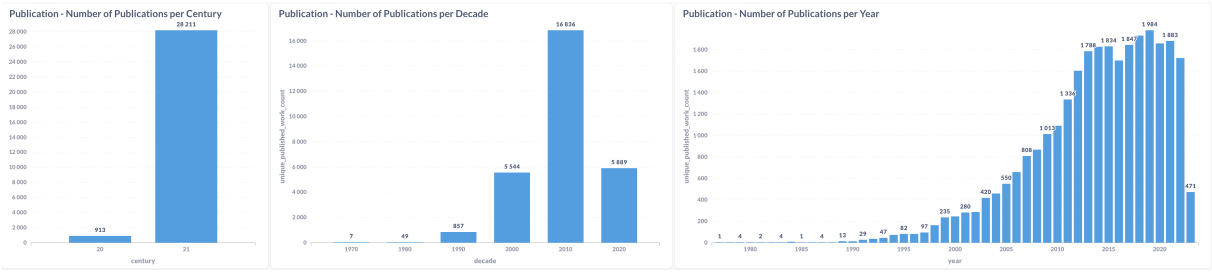

Figure 6.8: Publication profiling part 2

current dataset may not be sufficiently robust for affiliation-related discoveries.

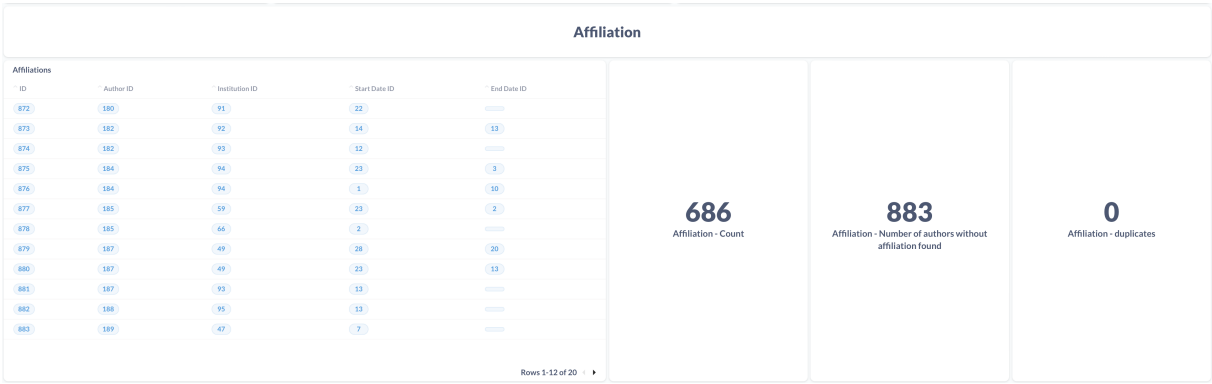

Figure 6.9: Affiliation profiling part 1

Diving into the Keyword Fact table (dashboard on Figure 6.11 and Figure 6.12), it's discerned that we have amassed 55641 unique entries, without any duplicate instances.

An analysis was carried out to ascertain the distribution of keywords across different time frames, assessing the feasibility of temporal discoveries. It was observed that, starting from 1999, we consistently extracted over 300 keywords per year. This constitutes an excellent dataset for our explorations. However, as we journey further into the past, the keyword density decreases, with no keywords recorded prior to 1982.

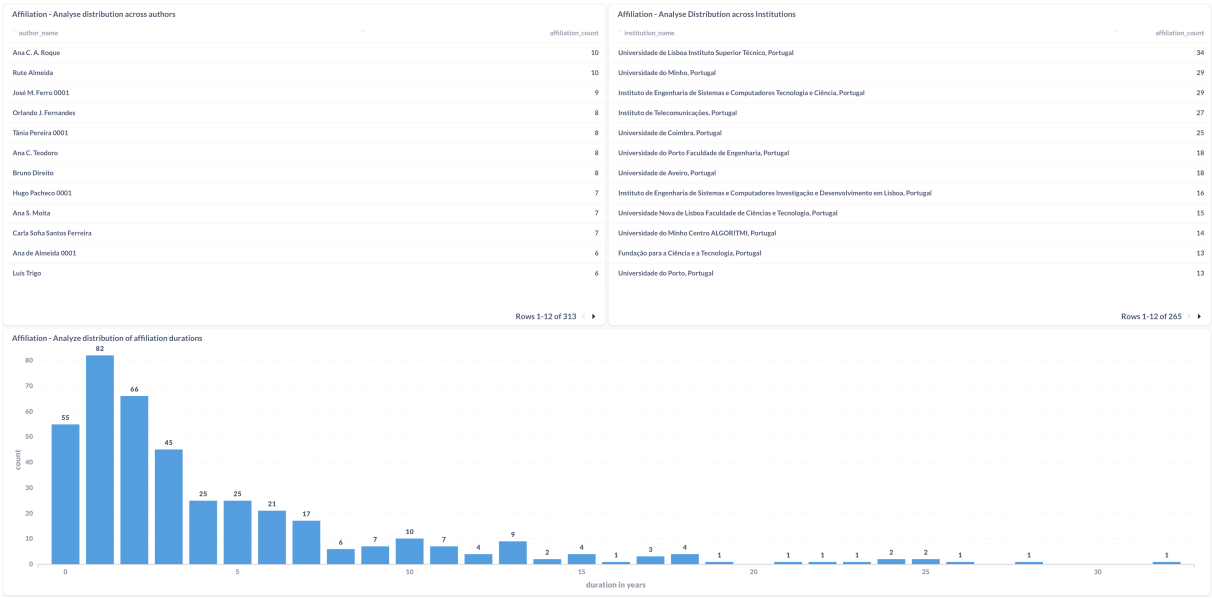

Figure 6.10: Affiliation profiling part 2

An evaluation of the keyword distribution by decades revealed a similar trend. The 1990s yielded 1085 entries, a sufficient amount to glean insights. This trend of growing keyword density persists in the subsequent decades, with 9541 entries in the 2000s, 31611 in the 2010s, and 13368 in the 2020s. However, a noticeable dip is observed in the 1980s, which only contributed 36 entries, with no keywords recorded prior to that decade.

Considering the data from a centennial perspective, the 20th century saw a modest 1121 keyword entries, while the 21st century demonstrated a significant leap with 54520 entries. This suggests that our keyword-based discovery will yield more reliable and substantial insights for the 21st century.

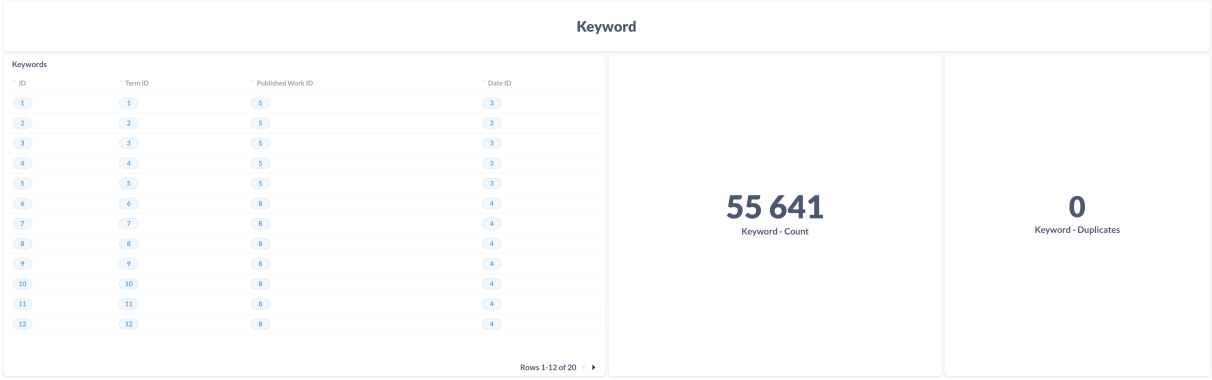

Figure 6.11: Keyword profiling part 1

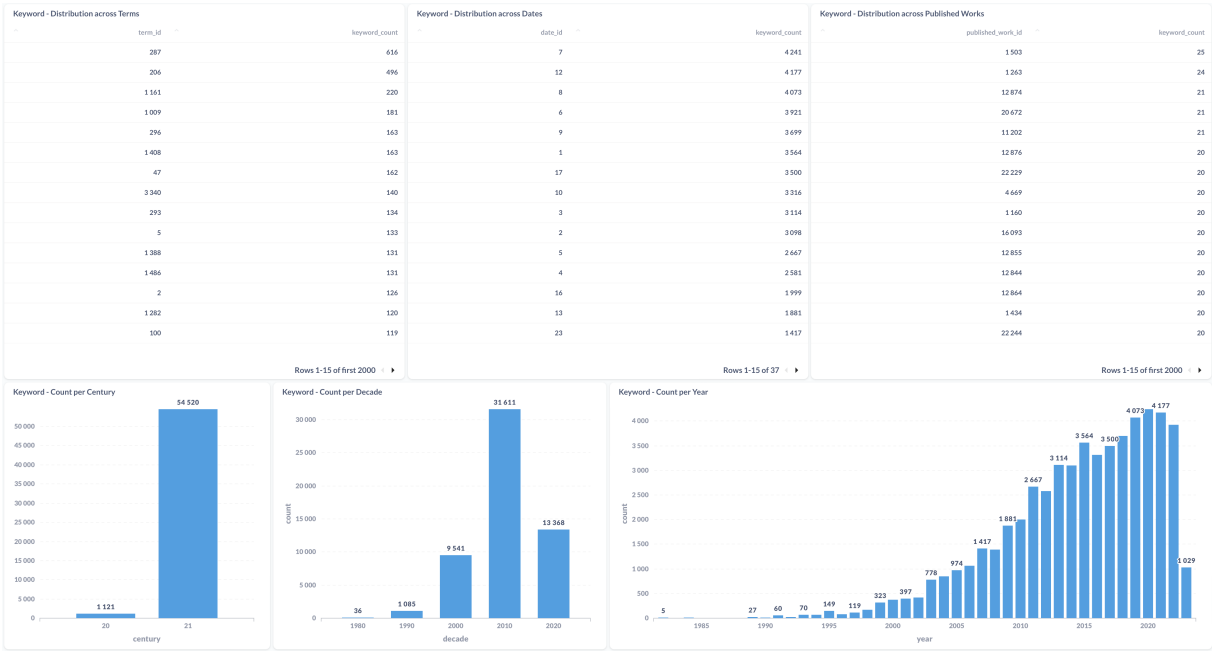

Figure 6.12: Keyword profiling part 2

6.2 Discovery

As we journey deeper into our data, we will now transition from data profiling to a more investigative role, presenting the findings of our analytical exercises. In the forthcoming section, we will navigate through a diverse array of data explorations, from ranking analysis to keyword visualizations, from scrutinizing the intricacies of co-authorship networks to deciphering authors' field classifications. These data discovery operations are aimed at uncovering hidden patterns, identifying key contributors, and gaining an understanding of the overarching themes present in our data.

6.2.1 Rankings

Now we embark on an exploration of rankings. By organizing our data into various orders of precedence, we aim to highlight key trends and dominant features within our data. This methodical analysis can reveal the most prolific authors, the busiest periods for publication, and the most frequently used keywords, amongst other insights. By condensing the complexity of our data into these succinct and easily interpretable rankings, we strive to distil the essence of our findings and provide a clear overview of the landscape of AI research in Portugal. Let's delve into these rankings and unearth the story they tell.

In this section, we are presenting multiple visual representations that provide comprehensive rankings of authors based on their publication output. These rankings are not just limited to an overall mea-

sure but are also segmented into different time frames, such as by century and by decade. By breaking down these rankings into specific periods, we obtain a detailed account of the authors' productivity over time.

These time-bound insights help paint a picture of how AI research has evolved and which authors have made significant contributions during these periods. However, it's crucial to remember that the quantity of publications, while indicative of productivity, does not solely determine an author's influence or impact. As we move forward, these rankings will be further contextualized within our broader analysis and interpretation of the data.

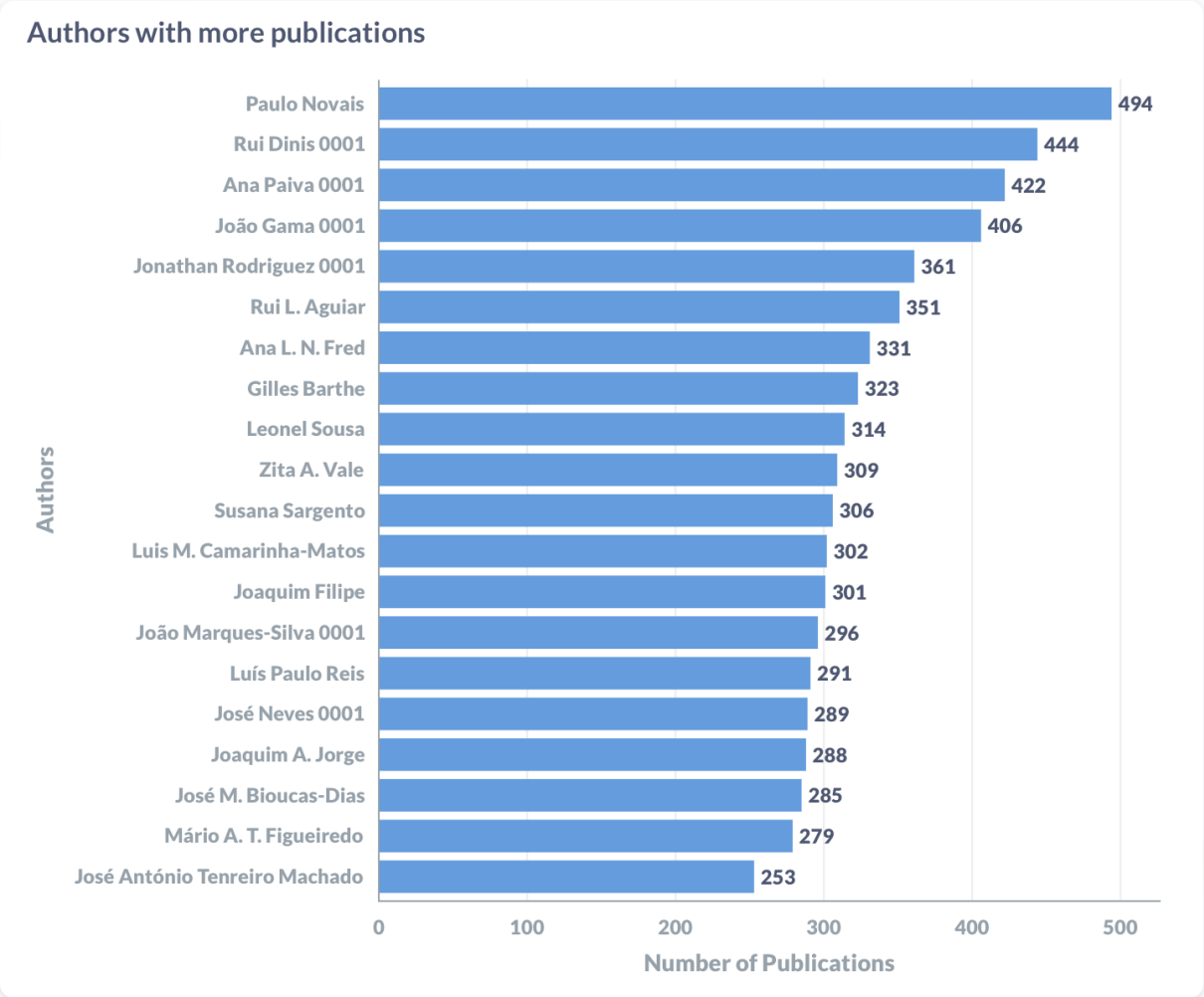

Figure 6.13: Authors with more publications

Next, we delve into the rankings of time frames that saw the highest volume of publications. By assessing the quantity of publications within each century and decade, we can trace the trajectory of AI research over time. These rankings offer a temporal view of AI research output, illustrating periods of intense activity and highlighting significant historical moments in the evolution of the field.

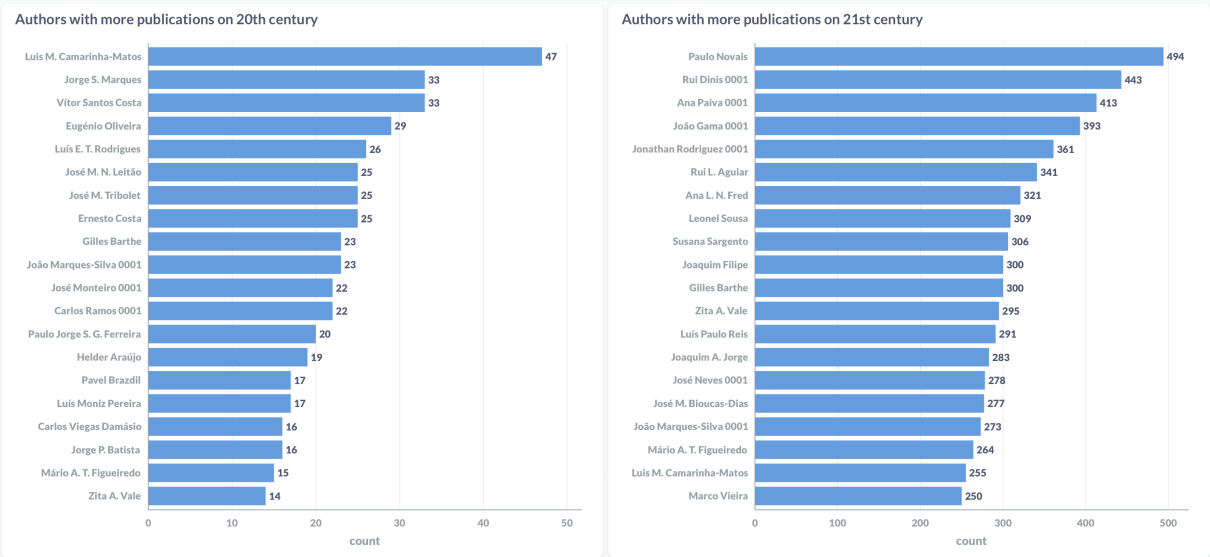

Figure 6.14: Authors with more publications per century

Figure 6.15: Authors with more publications between decades 1970 and 1990

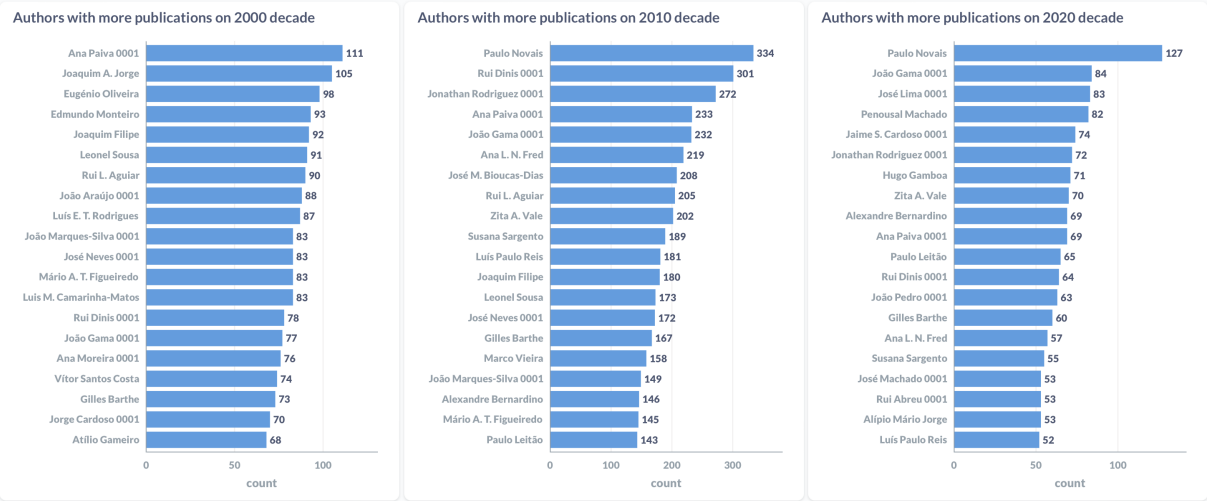

Figure 6.16: Authors with more publications between decades 2000 and 2020

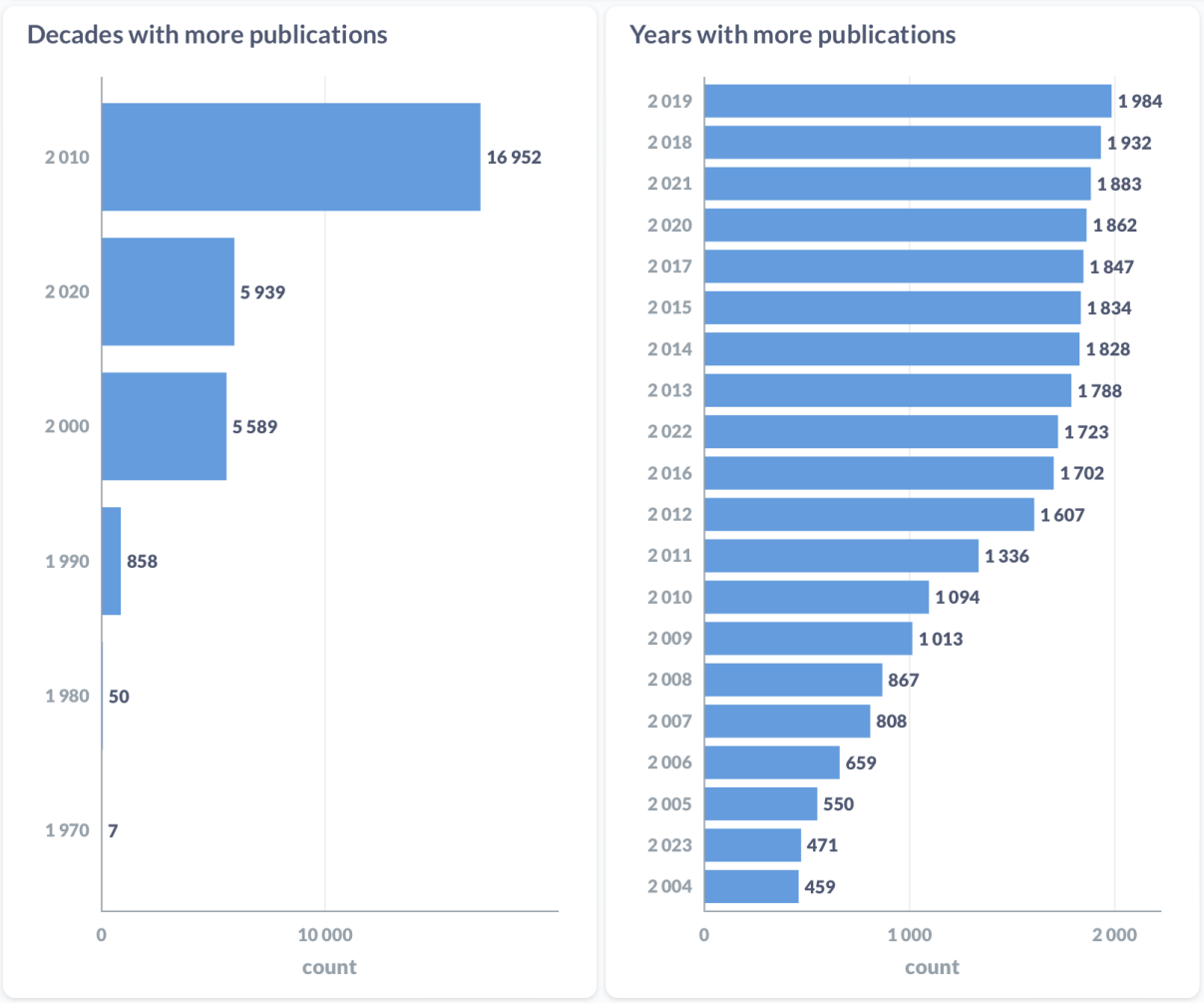

Figure 6.17: Decades and years with more publications

Moving on, we explore the rankings based on keyword usage. These rankings provide valuable insight into the most prominent themes and topics within the AI research community, both in general and over specific time frames.

In particular, we present rankings for the most commonly used keywords across all publications, as well as the most frequently used keywords segmented by century and decade. This breakdown allows us to monitor the progression of research interests and highlights the key areas of focus within each time period.

These keyword rankings serve as a reflection of the primary areas of interest in the AI field, and changes over time could signal shifts in research focus, emerging trends, or technological advancements.

Visualizing these rankings can facilitate a more intuitive understanding of these data points. Seeing the keywords displayed in order of frequency can quickly highlight prevalent research themes and illuminate their relative prominence in the field. With this information, we can better understand the broad strokes of AI research and how it has evolved over time.

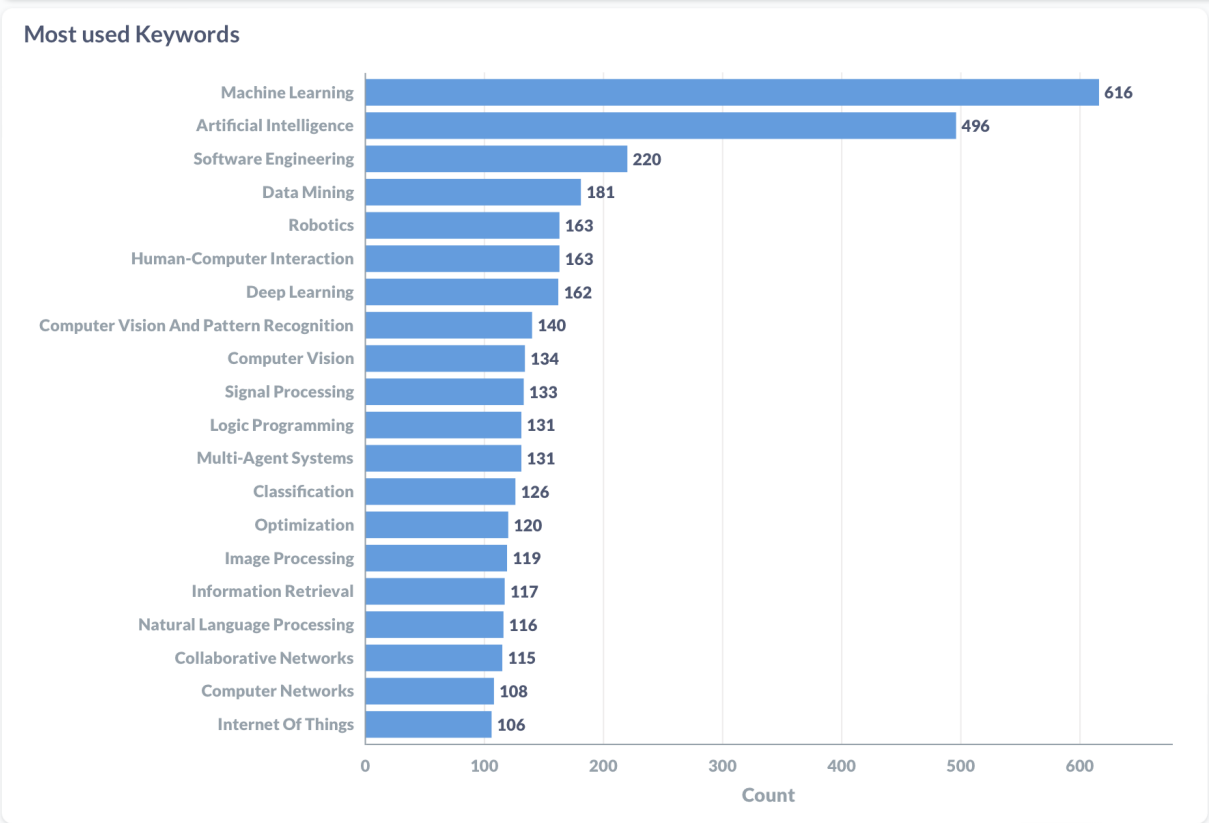

Figure 6.18: Most Used Keywords

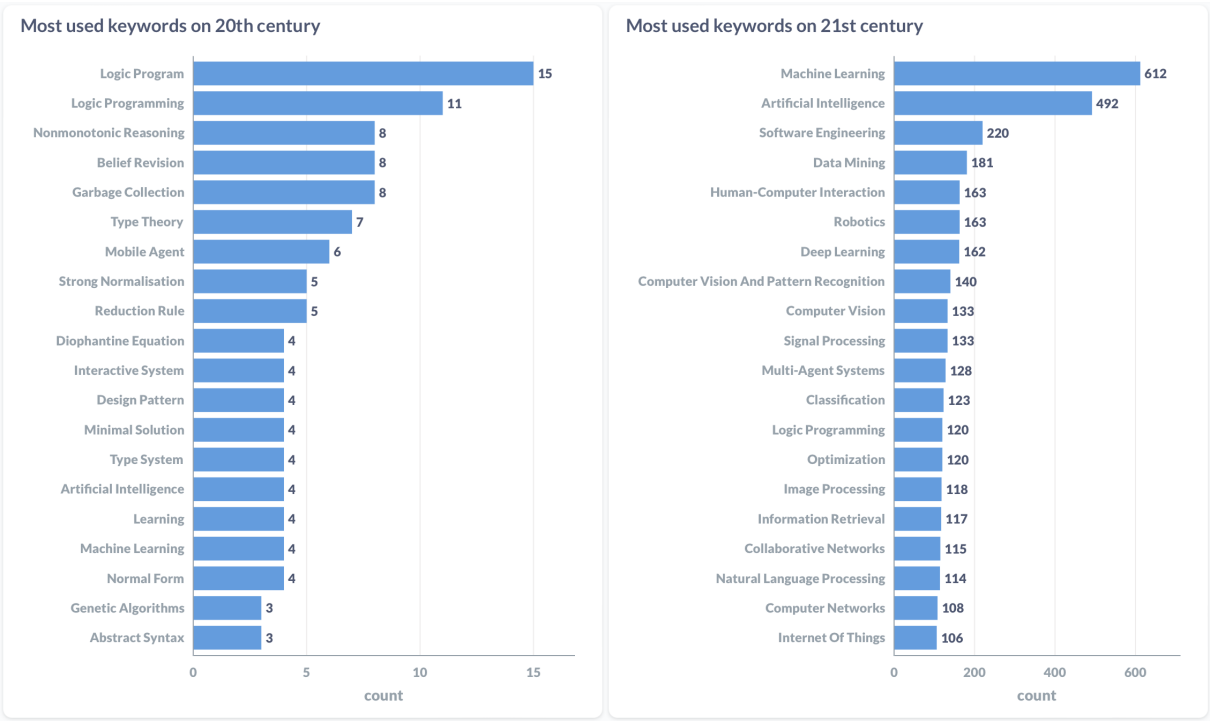

Figure 6.19: Most Used Keywords per Century

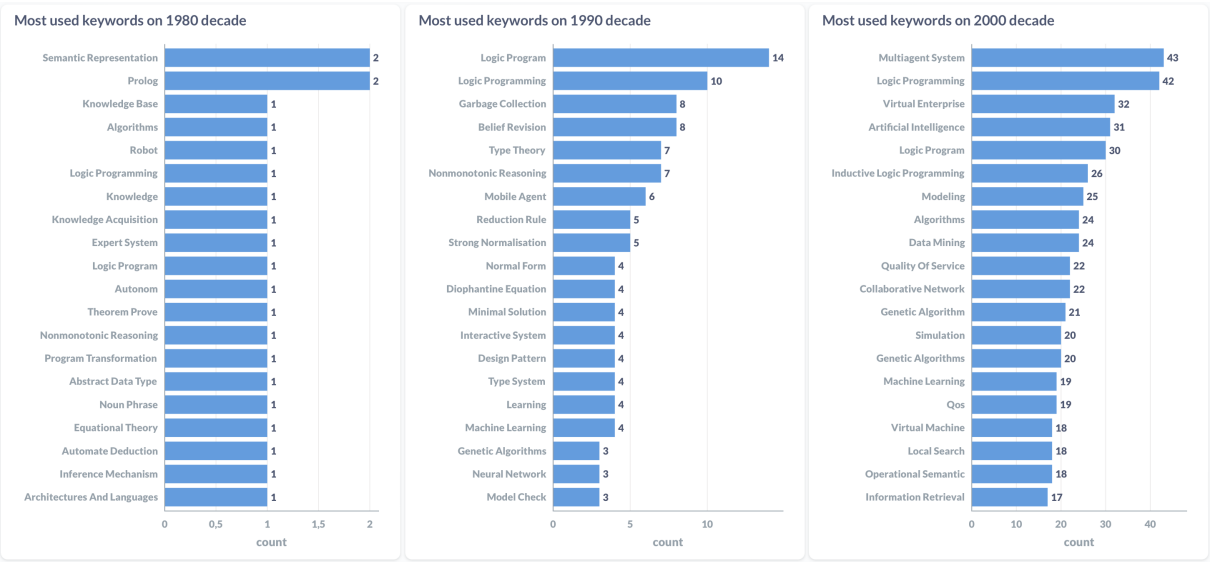

Figure 6.20: Most Used Keywords per Decade 1

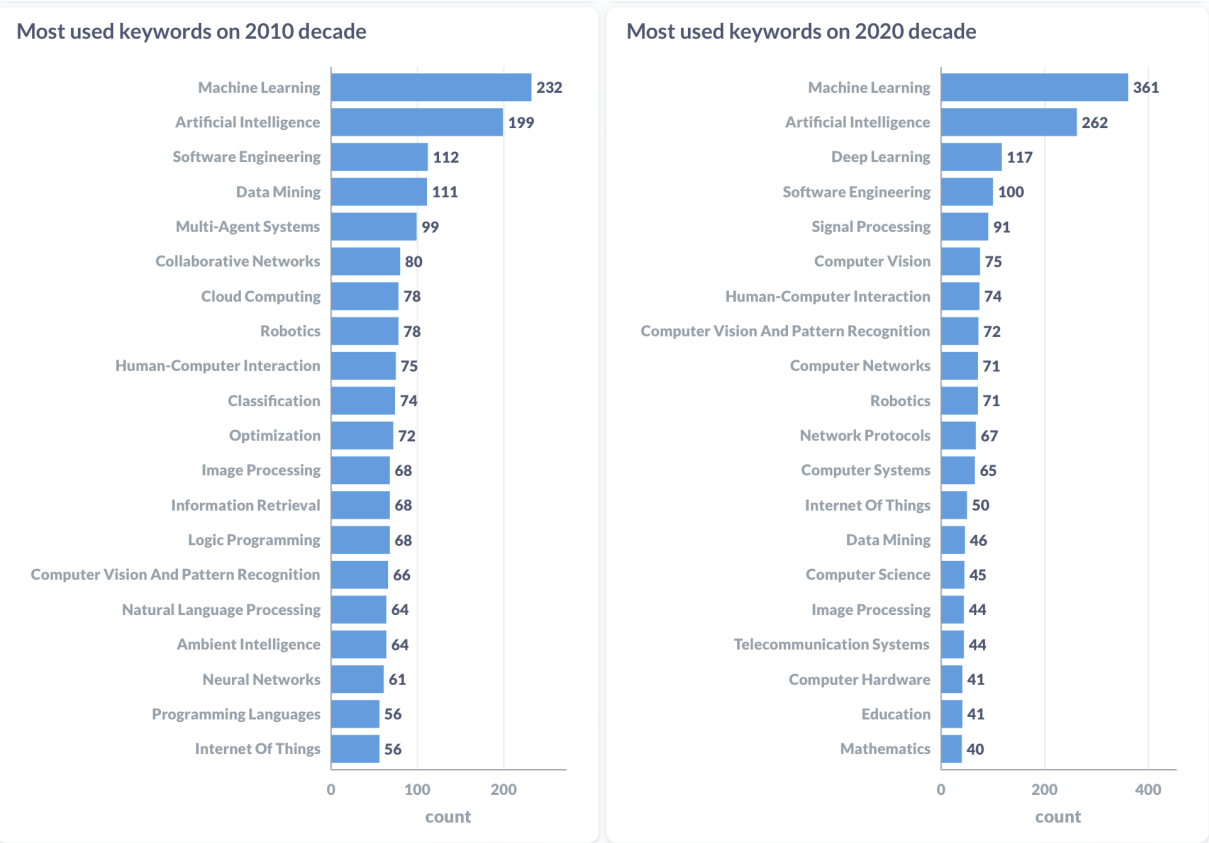

Figure 6.21: Most Used Keywords per Decade 2

6.2.2 Co-Authorship Network

The co-authorship network, on Figure 6.22, presents an interesting perspective on the collaboration patterns among the AI researchers within our data. Each dot in the network represents an author, and each line or edge represents a collaboration between two authors, as reflected in their joint publications.

The main cluster of the network, where most of the dots are densely packed together, signifies a closely-knit community of researchers who have frequently collaborated on AI research. This might indicate a strong collaborative culture within this community, fostering knowledge exchange and interdisciplinary work.

On the other hand, we notice several dots linked to the main cluster but situated farther away. These outliers might represent authors who, despite having collaborated with the researchers within the main cluster, generally engage in isolated or more specialized areas of research. These authors might be key connectors to external research communities, bringing novel ideas into the network and diversifying the knowledge base.

This co-authorship network, thus, not only reflects the strength and extent of collaboration among the AI researchers but also points to the diversity and breadth of research within this field. The rich interconnections within the network indicate the potential for the development of new ideas and the generation of innovative solutions through collaboration.

6.2.3 Word Clouds

Having examined the co-authorship network, we now turn our attention to a more visual representation of keyword usage in AI research - word clouds. This section is dedicated to the creation and interpretation of word clouds based on our data.

Word clouds offer a unique, intuitive, and visually engaging way to understand the frequency of keyword usage. Larger text size corresponds to higher frequency, allowing viewers to instantly grasp the most common themes in the field at a glance.

In this section, we will present word clouds for all keywords, as well as specific ones created for different centuries and decades. This temporal segmentation allows us to examine the shifts and trends in AI research themes over time, enriching our understanding of the development of the field. Let's delve into these visual representations and unravel the patterns they reveal.

Upon examining the word cloud representing all keywords used (Figure 6.23), several key trends emerge. The most frequently occurring keywords, depicted through larger text, are 'machine learning', 'artificial intelligence', 'software engineering', 'data mining', and 'human-computer interaction'.

This indicates that these themes are the most prevalent in our dataset. Their prominence reflects the central role these topics play in the field. Machine learning and Artificial Intelligence are the most

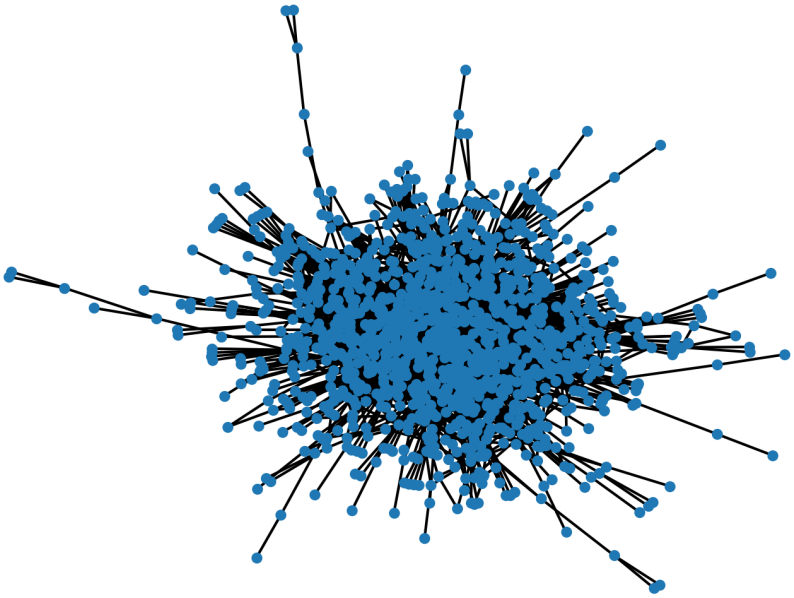

Figure 6.22: Co-Authorship Network graph

prominent, understandably.

Simultaneously, the prevalence of 'Software Engineering' demonstrates the foundational role it plays in implementing AI systems. The term 'Data Mining' was prevalently used between the years 1990 and 2015 to describe what we currently refer to as Data Science, a crucial aspect of AI applications. The frequency of 'Human-Computer Interaction' suggests a strong interest in making AI technologies user-friendly and understanding the ways humans interact with these systems.

The high frequency of these keywords indicates that research efforts in AI in Portugal align with global trends in the field, which is dominated by these core areas of interest.

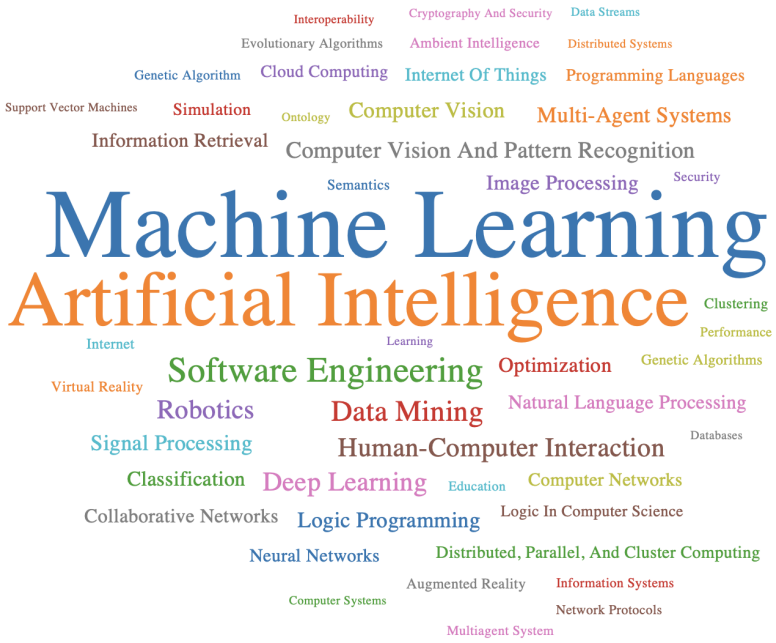

Figure 6.23: All Keywords

Turning our attention to the word cloud representing keywords used in the 20th century (Figure 6.24), it's clear that 'logic programming' and 'nonmonotonic reasoning' were the most prevalent topics of that era. However, it's essential to note that the overall diversity of keywords during this period is considerably less compared to more recent years.

'Logic programming' refers to a type of programming paradigm which is largely based on formal logic, and its prominence suggests a strong focus on theoretical foundations of AI during this period. 'Nonmonotonic reasoning', a form of reasoning used in AI where the introduction of new knowledge can invalidate old conclusions, also points to an era where foundational concepts and approaches in AI were being established.

Nonetheless, due to the limited number of keywords, it is prudent to interpret the results with caution.

The relative scarcity of data from this period may not fully capture the breadth of AI research undertaken in the 20th century.

Shifting to the 21st century (Figure 6.25), we find that the keyword distribution closely mirrors that of the global word cloud, reflecting the recent surge in AI research.

However, it's important to keep in mind that our 21st century word cloud is heavily influenced by the sheer volume of research produced during this time, far surpassing the volume from the 20th century. As such, this word cloud may not only reflect the changing trends in AI research but also the overall growth and expansion of the field itself.

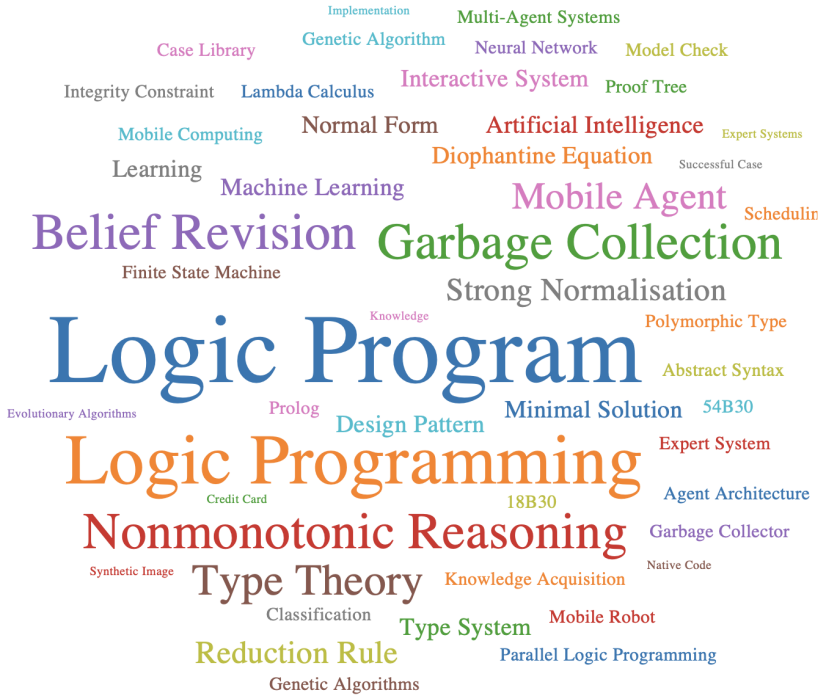

Figure 6.24: Keywords XX Century

As we delve into the word clouds by decade, we are met with fascinating insights regarding the shifting trends and focus areas in AI research.

During the 1980s (Figure 6.26), the data we have is sparse, yet we see 'Prolog' distinctly standing out. However, due to the limited amount of data, we should exercise caution in interpreting this as a reflection of the state of the field during that decade.

The 1990s (Figure 6.27), bring forth a clearer image with keywords such as 'Logic Programming', 'Nonmonotonic Reasoning', 'Garbage Collection', 'Belief Revision', and 'Type Theory' at the forefront. These terms are indicative of the areas that dominated AI research during that time.

As we transition into the 2000s (Figure 6.28), the focus appears to shift, with 'Inductive Logic Programming', 'Virtual Enterprise', and 'Quality of Service' gaining prominence. The term 'Artificial Intel-

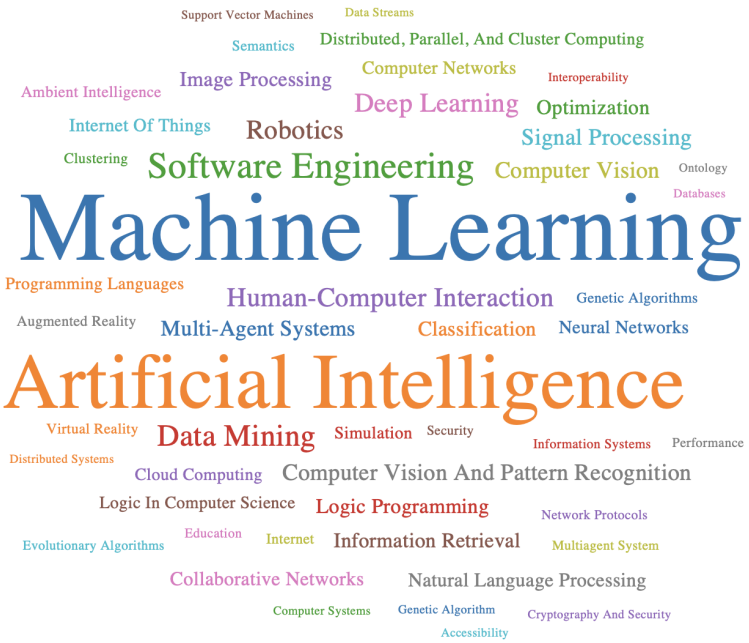

Figure 6.25: Keywords XXI Century

Figure 6.26: Keywords 1980 Decade

Figure 6.27: Keywords 1990 Decade

ligence' also notably emerges as a dominant keyword, potentially a result of the term becoming more mainstream and broadly applied.

In the 2010s (Figure 6.29), there's a clear shift towards more applied areas with 'Machine Learning', 'Artificial Intelligence', 'Software Engineering', 'Data Mining', and 'Multi-Agent Systems' leading the way. These keywords signal the growing importance and ubiquity of these areas in AI research.

Finally, in the 2020s (Figure 6.30), the word cloud is highly similar to the one from the 2010s. Notably, we see the addition of 'Signal Processing' and 'Deep Learning', indicating the rising prominence of these fields.

These word clouds, while offering valuable insights into the dominant areas of research during each decade, also mirror the broader trends and transitions within the field of AI, underscoring the evolution from more theoretical to applied research areas.

6.2.4 Labelling authors per field

As we delve into the outcomes of applying our labelling process, as described in Section 5.5, to our dataset, intriguing patterns and findings emerge. Our exploratory journey began with an extensive variance study on the dataset employing the K-means clustering algorithm. We adopted an iterative approach, testing a variety of variance thresholds to eliminate keywords with low variance and concurrently

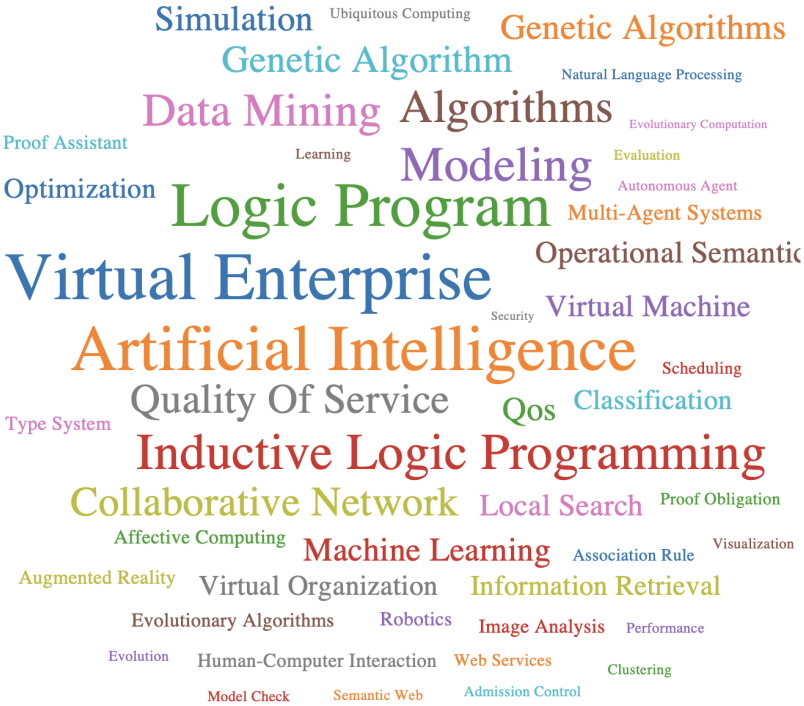

Figure 6.28: Keywords 2000 Decade

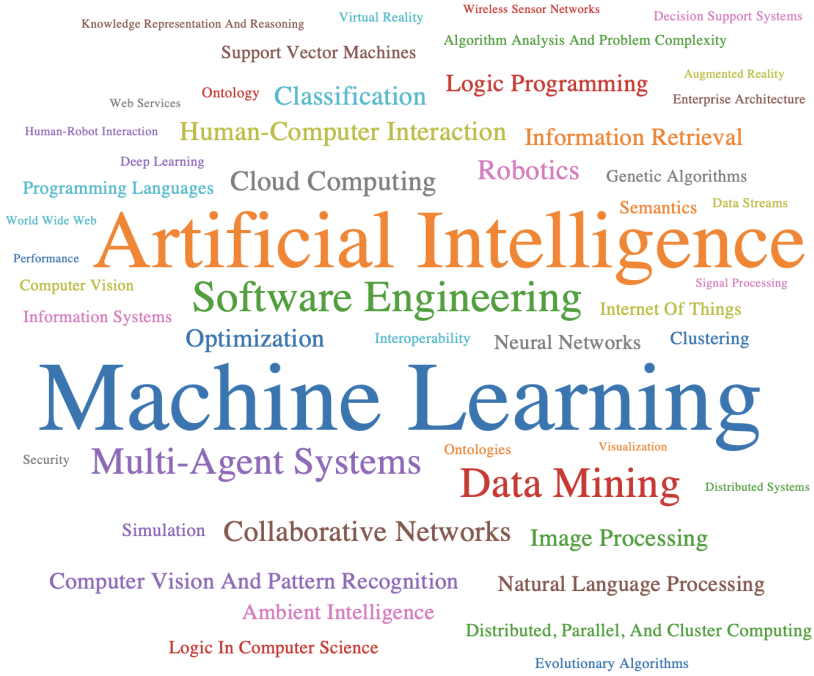

Figure 6.29: Keywords 2010 Decade

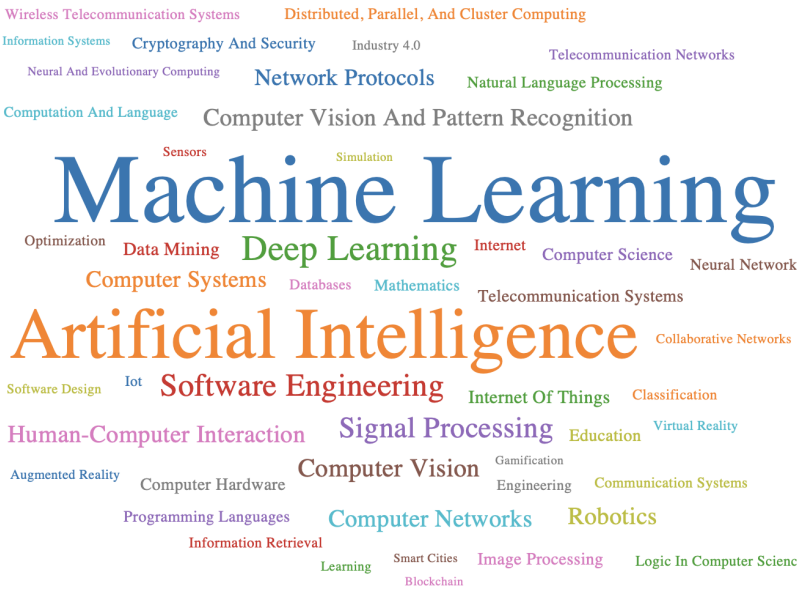

Figure 6.30: Keywords 2020 Decade

experimenting with different cluster numbers in the K-means algorithm.

Our aim was to identify the combination of variance threshold and cluster number that yielded the optimal Silhouette score, a measure of how similar an object is to its own cluster compared to other clusters, using cosine similarity. The results indicated that a variance threshold of 0.004 proved to be the most effective, as depicted in Figure 6.31. This finding reflects the balance we sought to achieve between retaining meaningful variability in the data while removing noise or less significant features.

Continuing our pursuit for optimization, we ventured to test alternate preprocessing techniques to discern their potential for enhancing the results. The Principal Component Analysis (PCA) was one such method, which we employed as a feature extraction technique on the initial dataset. Similar to the earlier variance study, we undertook an iterative examination, testing various thresholds to identify the most effective one, with the K-means algorithm and the Silhouette score serving as our performance benchmark.

Our findings revealed that a PCA threshold of 0.07 yielded the best results, as illustrated in Figure 6.32. This means that in the PCA dimensionality reduction process, we maintained enough principal components such that they explained 7% of the total variance in our data. However, when juxtaposed with the outcomes obtained from the variance study, it became apparent that PCA was less effective in our dataset. The results derived from applying PCA did not outperform the ones attained from the variance technique. Given these observations, we made the decision to discard PCA as a preprocessing technique for our dataset, affirming the superior performance of the variance technique.

Building on our prior findings, we proceeded to apply the variance technique to our dataset, aiming

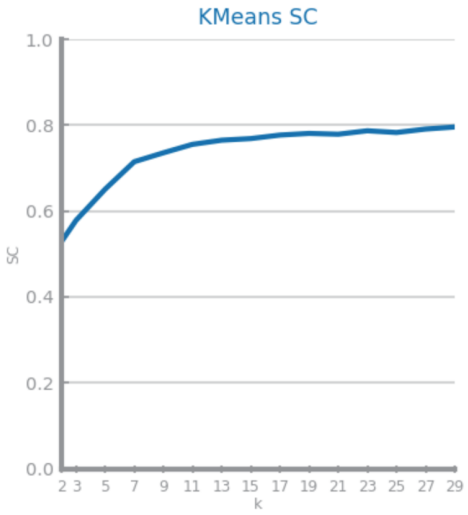

Figure 6.31: Optimal variance result

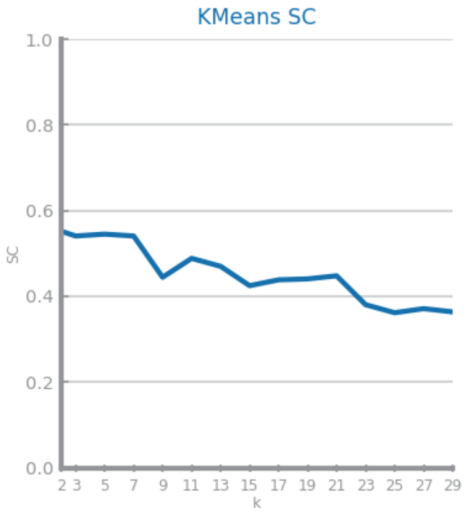

Figure 6.32: Optimal PCA result

to understand the number and the nature of keywords that survived the process. Our analysis revealed a dramatic reduction in the keyword count. Of the original 24,703 keywords present in the dataset, only 10 managed to weather the variance technique. These resilient keywords, pivotal to our dataset, were as follows: *Signal Processing*, *Classification*, *Deep Learning*, *Internet of Things*, *Artificial Intelligence*, *Computer Vision*, *Sensors*, *Machine Learning*, *Software Engineering*, and *Robotics*.

Such a stark reduction in keyword count, while indicative of the robustness of the variance technique, also suggested that further preprocessing might lead to an excessively simplified dataset. Given the fact that the surviving keywords were fundamentally insightful and likely sufficient for constructing meaningful clusters, we decided against further reducing our dataset. The decision was based on maintaining an optimal balance between data simplification and preservation of valuable insights.

In the ensuing phase, we were tasked with the pivotal decision of specifying the number of clusters we wished to adopt for our labelling process. This selection relied on three primary indicators: the count of keywords, our inherent knowledge in the sphere of AI, and the Silhouette score which is a measure of cluster cohesion and separation.

Informed by these factors, we established nine clusters. This choice was influenced by the presence of only ten keywords, the Silhouette score for nine clusters - which although not the highest, was still promising - and our understanding of AI, which is generally perceived to comprise around 6-10 fields, although this is not definitively established and thus, gives us a range to work within.

After applying the K-means clustering with the identified optimal settings, we obtained a variety of clusters, each initially labeled with a unique numerical identifier. However, to lend more interpretability to our results and truly encapsulate the essence of each cluster, we derived a name for each one based on the two most influential keywords within that cluster. The first keyword is considered more influential than the second, therefore, the cluster names were attributed as follows:

Cluster 0: 'Sensors & Signal Processing'

Cluster 1: 'Machine Learning & Artificial Intelligence'

Cluster 2: 'Deep Learning & Machine Learning'

Cluster 3: 'Machine Learning & Signal Processing'

Cluster 4: 'Internet of Things & Computer Vision'

Cluster 5: 'Software Engineering & Machine Learning'

Cluster 6: 'Classification & Machine Learning'

Cluster 7: 'Artificial Intelligence & Signal Processing'

Cluster 8: 'Robotics & Artificial Intelligence'

These names, derived from the top two influential keywords in each cluster, offer an insightful peek into the dominant research themes present in each group.

Having established the cluster names and their key thematic areas, we can now turn our attention

to the results of the clustering process. These results illustrate the distribution of authors across the different clusters, providing a snapshot of the prevalent research trends amongst our author community. In Figure 6.33, we present a detailed breakdown of the number of authors allocated to each cluster. This figure provides a valuable overview of the scale and distribution of research efforts within each identified thematic area.

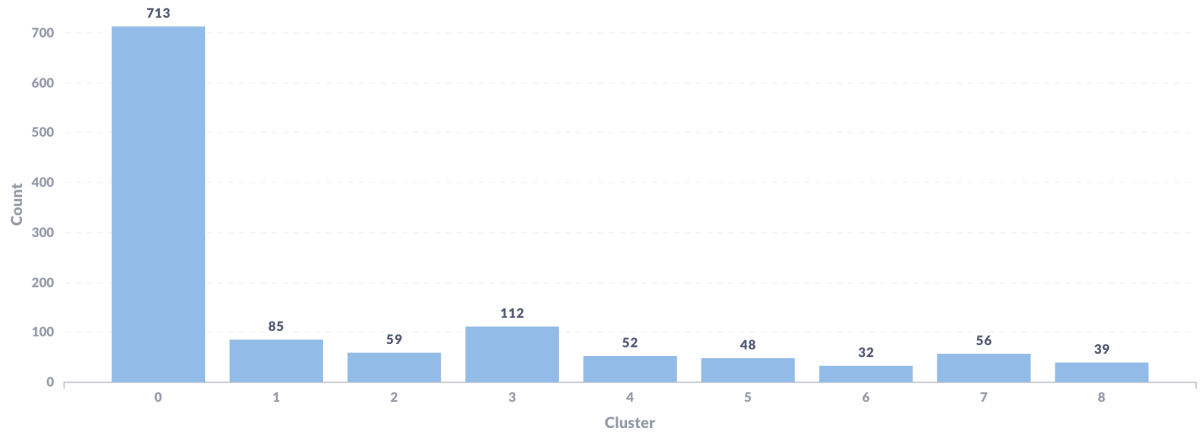

Figure 6.33: Number of authors per cluster

7

Conclusion

Contents

In conclusion, the objective of this project was to propose and develop a system capable of extracting, storing, and analyzing scientific publications produced by researchers in the field of computer science in Portugal. We have managed to build a flexible system that is not limited to this initial scope and can be applied to any field or geographical location. The implementation was primarily focused on the field of computer science, specifically its application to AI research, as illustrated in our case study.

Through this work, we addressed critical concerns in tracking and understanding scientific output. We have devised mechanisms to: monitor the development of research work in real time; detect and interpret patterns, trends, and primary areas of research; gain a thorough understanding of the state of the art in any given field.

This project serves as a significant step towards facilitating a more structured and comprehensive approach to understanding and navigating the vast landscape of scientific research. It stands as a robust platform for future enhancements, capable of contributing to a more informed and interconnected scientific community.

7.1 System Limitations and Future Work

When considering the system's limitations and future improvements, it should be noted that the current structure is primarily designed for the field of computer science. However, it could be expanded to encompass other areas of study by implementing additional scrapers to collect data from other sources.

The data extraction phase is one aspect that could benefit from refinement. Currently, the time required to retrieve all data is significant; potential solutions could involve optimizing the existing processes or implementing parallelized tasks to expedite the process.

Furthermore, the extraction of details from publications, such as keywords, could be enhanced. This could be achieved by integrating more domain-specific scrapers or even by employing AI tools capable of extracting keywords from the content of available PDFs of publications.

Finally, the system could also accommodate other sophisticated data mining features. Future iterations could introduce predictive capabilities to forecast trends in publications and keywords, thereby providing a more dynamic and forward-looking perspective on the evolution of research fields.

7.2 Practical Implementation of the System

In a practical implementation of the system, the extracted and stored data of our case study was shared with an organization named KBAI Research. KBAI Research is committed to offering tools that facilitate the management and expansion of knowledge related to work, personal interests, societal matters, or daily activities.

Their key objective was to develop an editor and a visualizer of structured knowledge bases as a foundation for implementing automatic reasoning algorithms. In this context, our database provided them with an invaluable resource. They approached us with a request to share our database, as the wealth of information within it served as a powerful knowledge base to support their work. The real-world application of our data by KBAI Research is a testament to the practical and significant value the system offers.

Bibliography

[1] “Portal appia,” ”https://www.appia.pt”.

[2] “Inforum,” ”https://inforum.org.pt”.

[3] “Express,” ”https://expressjs.com”.

[4] “Django,” ”https://www.djangoproject.com”.

[5] “Welcome to flask,” ”https://flask.palletsprojects.com”.

[6] “Fastapi,” ”https://fastapi.tiangolo.com”.

[7] “Spring,” ”https://spring.io”.

[8] “Apache struts,” ”https://struts.apache.org”.

[9] D. Bartholomew, “Sql vs. nosql,” *Linux Journal*, vol. 2010, no. 195, p. 4, 2010.

[10] R. Ramakrishnan and J. Gehrke, *Database management systems*. Mcgraw-Hill Education, 2003.

[11] “Data visualization: Microsoft power bi,” ”https://powerbi.microsoft.com”.

[12] “Tableau,” ”https://www.tableau.com/”.

[13] “Metabase,” ”https://www.metabase.com”.

[14] “Qlik sense — modern cloud analytics,” ”https://www.qlik.com/us/products/qlik-sense”.

[15] “Angular,” ”https://angular.io”.

[16] “React – a javascript library for building user interfaces,” ”https://reactjs.org/”.

[17] “Vue.js - the progressive javascript framework — vue.js,” ”https://vuejs.org/”.

[18] “Stack overflow developer survey 2022,” ”https://survey.stackoverflow.co/2022/”.

[19] “Docker,” ”https://www.docker.com”.

[20] R. C. Martin, *Clean Architecture : a craftsman's guide to software structure and design*. Prentice Hall, 2018.

[21] H. Subramanian and P. Raj, *Hands-on RESTful API Design Patterns and Best Practices : Design, Develop, and Deploy Highly Adaptable, Scalable, and Secure RESTful Web APIs*. Birmingham, UK: Packt Publishing Limited, 2019.

[22] "Beautifulsoup library," "https://www.crummy.com/software/BeautifulSoup/bs4/doc/".

[23] "Angular d3 cloud library," "https://github.com/maitrungduc1410/d3-cloud-angular".

[24] "Kmeans clustering," "https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html".

[25] "Pca," "https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html".