

# Enhanced biomedical data extraction from scientific publications

by

**Markus Almendral Berggrav**



Master's Thesis  
Department of Informatics  
University of Bergen

June 1, 2023

# Abstract

The field of scientific research is constantly expanding, with thousands of new articles being published every day. As online databases grow, so does the need for technologies capable of navigating and extracting key information from the stored publications. In the biomedical field, these articles lay the foundation for advancing our understanding of human health and improving medical practices. With such a vast amount of data available, it can be difficult for researchers to quickly and efficiently extract the information they need. The challenge is compounded by the fact that many existing tools are expensive, hard to learn and not compatible with all article types. To address this, a prototype was developed. This prototype leverages the PubMed API to provide researchers access to the information in numerous open access articles. Features include the tracking of keywords and high frequent words along with the possibility of extracting table content. The prototype is designed to streamline the process of extracting data from research articles, allowing researchers to more efficiently analyze and synthesize information from multiple sources.

# Acknowledgements

I would like to extend my sincere thanks to my supervisor, Harald Barsnes, and my co-supervisor, Yehia Farag, for their guidance and support throughout my research. I am grateful for their expertise and insights, which have significantly shaped my work. I am also appreciative of my family and my girlfriend for their support and encouragement throughout my study. Their belief in me has been a constant source of motivation.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>1 Preface</b>	<b>1</b>
1.1 Objective . . . . .	1
1.2 Thesis outline . . . . .	1
<b>2 Background</b>	<b>3</b>
2.1 Systematic review and meta-analysis . . . . .	3
2.2 The PubMed database . . . . .	5
2.3 Article screening . . . . .	6
2.4 Copyright restrictions . . . . .	7
2.5 Article structure . . . . .	8
2.6 Manual data extraction . . . . .	9
2.7 Automated data extraction . . . . .	11
2.7.1 Parsers and data mining . . . . .	11
2.7.2 Existing technologies . . . . .	12
2.8 Data repositories . . . . .	13
<b>3 Methods</b>	<b>15</b>
3.1 File handling . . . . .	15
3.1.1 PDF . . . . .	16
3.1.2 HTML . . . . .	17
3.1.3 XML . . . . .	18
3.1.4 CSV . . . . .	19

---

3.2	PubMed API . . . . .	20
3.2.1	E-utilities . . . . .	20
3.2.2	ID converter . . . . .	22
3.2.3	PMC OA Web Service API . . . . .	23
<b>4</b>	<b>Results</b>	<b>24</b>
4.1	Project structure . . . . .	24
4.2	Retrieving the articles . . . . .	26
4.3	Multiple article inspection . . . . .	29
4.3.1	Presenting the articles . . . . .	30
4.4	Single article inspection . . . . .	32
4.4.1	Graphical user interface . . . . .	34
<b>5</b>	<b>Discussion</b>	<b>40</b>
5.1	Human-machine interaction . . . . .	40
5.2	Benefits of using an API . . . . .	41
5.3	Developing a good relevance criteria . . . . .	42
5.4	Highlighting data vs. extracting data . . . . .	44
<b>6</b>	<b>Future work</b>	<b>47</b>
6.1	Improving the graphical user interface . . . . .	47
6.2	Additional use of the PubMed API . . . . .	48
6.3	Utilizing new technology . . . . .	48
<b>7</b>	<b>Conclusion</b>	<b>51</b>

# List of Figures

2.1	Quality of evidence . . . . .	4
2.2	The four main steps performed when gathering data . . . . .	4
2.3	Example of a PubMed search in the web browser. . . . .	6
2.4	Screenshot of a protein plot in CSF-PR. . . . .	14
4.1	Diagram of the project structure. . . . .	25
4.2	Example of the output of an E-utilities search in the PMC article subset. . . . .	26
4.3	Example of the output of the PubMed ID converter. . . . .	27
4.4	Example of how to retrieve the downloadable article file. . . . .	27
4.5	Locating the title in an articles XML file. . . . .	28
4.6	Diagram over components in article classification and their characteristics. . . . .	29
4.7	Presenting multiple articles. . . . .	30
4.8	Comparing relevancy of two articles. . . . .	31
4.9	Diagram over components in article inspection and their characteristics. . . . .	34
4.10	Inspecting a single article. . . . .	35
4.11	Overview of the table presenting the keyword frequencies. . . . .	36
4.12	Overview of the most frequent words in an article text. . . . .	37
4.13	Example of how tables are presented in the GUI. . . . .	38
4.14	Example of output when storing selected article columns. . . . .	38

# List of Tables

2.1	Inclusion and exclusion criteria in the screening process. . . . .	7
2.2	Steps when performing a systematic review. . . . .	10
2.3	Common steps performed when parsing an article. . . . .	12
2.4	Examples of available technologies for data extraction and data handling. . . . .	13
3.1	Commonly used file types in scientific research. . . . .	16
3.2	Overview of the API-tools in E-utilities. . . . .	21

# Chapter 1

## Preface

This chapter introduces the objective of the thesis and gives a general overview of the topics that will be discussed.

### 1.1 Objective

The objective of this thesis is to enhance the process of extracting biomedical data from scientific publications. To achieve this goal, technologies capable of processing online articles will be explored, through the development of an innovative prototype extraction tool. This prototype will search and extract data from the PubMed Central open access subset of the PubMed database. The biomedical field is in need of an extraction tool which is tailored to the unique characteristics, requirements and formats of this domain. Enabling researchers to quickly and efficiently process large volumes of text can facilitate the identification of trends and patterns across multiple studies, potentially leading to new insights and discoveries.

### 1.2 Thesis outline

The thesis begins by establishing the scientific context for where an extraction tool can be of use. Here the process of conducting a systematic review is introduced along with how data is gathered for this type of study. Next, the focus is on how file formats influence the data extraction process, this includes: what file types are common in scientific journals, how these files can be retrieved, and how different files are traversed and parsed. The



implementation and design is then detailed, with an overview of the project structure, explaining how different parts of the developed prototype interact with each other to display the article content. Finally, possible future improvements are outlined.

# Chapter 2

## Background

To understand the benefit of data extraction tools in scientific research, the process of conducting a systematic review will be explained along with how a data extraction tool can be a great resource when creating data repositories.

### 2.1 Systematic review and meta-analysis

A systematic review is a research technique consisting of collecting all possible studies related to a given topic, and analyzing their results. The results provided by each study consists of data such as sample size, study design, and effect sizes [1]. These elements are analyzed using statistical techniques to calculate an overall effect size or summary statistics that represents the combined results of all the studies, referred to as a meta-analysis. The goal of a meta-analysis is to provide a more comprehensive and objective understanding of the research evidence on a particular topic than can be obtained from any individual study.

Systematic reviews provide several benefits over individual studies, including increased statistical power, the ability to identify patterns or trends across studies, and the ability to test hypothesis that may not have been possible to test in individual studies due to small sample sizes [2]. In recent years meta-analyses have been actively performed in various fields and made easier with the large volumes of information available online [1]. *Figure 2.1* shows the quality of evidence in different types of research studies and indicates that the highest quality of evidence can be accomplished in a systematic review study.

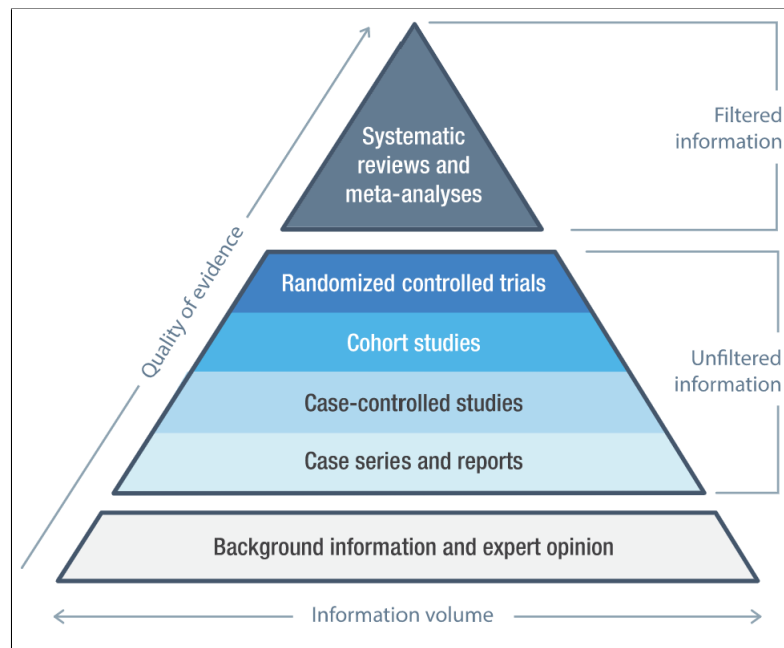


Figure 2.1: *Quality of evidence* [3].

In general there are four main steps researchers follow while conducting this type of study (see *Figure 2.2*) [4] : i) preprocessing, ii) localization of relevant articles, iii) extraction of relevant data, and iv) result reporting. Traditionally, this was done manually, but the development of data extraction tools has now made it possible to automate parts of this process.

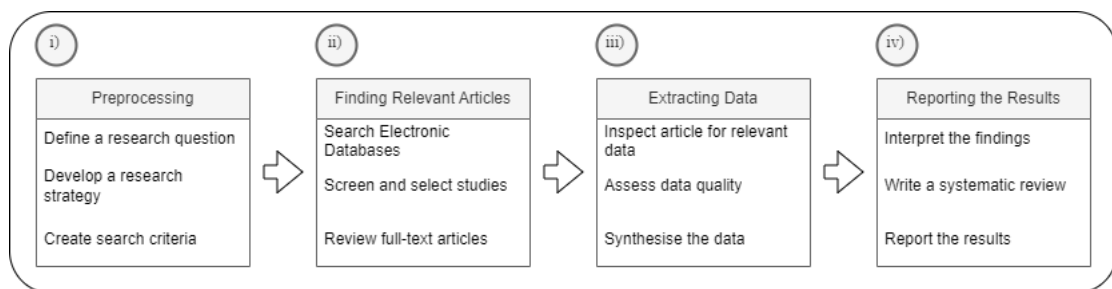


Figure 2.2: *The four main steps performed when gathering data.*

The initial preprocessing step involves defining a clear research question, developing a research strategy and creating a search criteria. A clear research question will simplify the search and ensure that relevant data is collected. Developing a research strategy involves identifying and planning the necessary steps for conducting the research. Cre-

ating a set of search criteria include finding keywords, databases, or other search terms which can help with identifying relevant studies.

A tool for data extraction can assist researchers in finding relevant articles (Step 2) and in extracting data (Step 3) [5]. Since articles are automatically processed, the extracted data is less prone to be affected by human errors. In addition, ensuring that the data is extracted in a uniform way, reduces the risk of inconsistencies or bias that can occur when doing it manually.

## 2.2 The PubMed database

PubMed is a database that, at the current time of writing, comprises of over 35 million citations of biomedical data from numerous different journals [6]. It has a sophisticated search algorithm and its comprehensive database makes it a valuable resource for researchers, healthcare professionals, and others seeking information on a wide range of biomedical topics. PubMed is maintained by the National Library of Medicine (NLM) at the National Institute of Health (NIH) in the United States.

In PubMed a user can perform a search with one or more search terms and get access to research articles that are related to these terms. Terms can be keywords, phrases, or MeSH (Medical Subject Headings) terms and are connected using operators such as AND, OR and NOT. Using comma as operator will invoke inclusive OR, meaning that both AND and OR will be used when searching for articles. PubMed's search algorithm takes into account factors such as the relevance of the search terms to the article, the articles popularity and the recency of the publication [7].

When performing a search, the algorithm retrieves the articles that matches the search criteria and ranks them based on relevance. If this search ends up giving too many results, it is possible to refine the search by using different filters. *Figure 2.3* shows an example of a PubMed search.

The screenshot displays a PubMed search interface. At the top, the search bar contains the keyword "CSF-PR". Below the search bar, there are options for "Advanced", "Create alert", and "Create RSS". The search results are sorted by "Best match" and show 3 results. The left sidebar contains filters for "MY NCBI FILTERS", "RESULTS BY YEAR", "TEXT AVAILABILITY", "ARTICLE ATTRIBUTE", "ARTICLE TYPE", and "PUBLICATION DATE". The "Free full text" filter is selected under "TEXT AVAILABILITY". The "1 year" filter is selected under "PUBLICATION DATE". The search results list three articles, each with a checkbox, a title, authors, journal information, and a "Free PMC article" link.

Figure 2.3: Example of a PubMed search using keyword "CSF-PR" and filter "Free full text".

## 2.3 Article screening

After the number of publications have been narrowed down, the next step is to look at the abstracts and titles of the articles. PubMed provides abstracts for all its articles and links to where the full-texts can be located. By looking at the abstracts a researcher gets an idea of what an article is about, but the full-text may not be open access. PubMed provides links to full-text of articles that are freely available, as well as links to articles that require a subscription or payment to access.

The process of selecting articles to include in a systematic review is called the screening process. Under the screening process researchers apply inclusion/exclusion criteria to the articles that seem relevant which help shorten the list. Inclusion/exclusion criteria

are predetermined criteria used to determine whether an article should be included or excluded from a study [8]. Some factors to consider when applying these criteria are outlined in *Table 2.1*.

<b>Study design</b>	Is it randomized controlled trial, observation study, or case study? Depending on the research question, articles may be included or excluded based on their study design.
<b>Population</b>	Is it a relevant population for the research question? For example, when studying a specific disease, the article must include patients with this disease.
<b>Intervention / exposure</b>	For example, when studying the effectiveness of a new drug, you may want to include articles that study the use of this drug.
<b>Outcome</b>	For example, when studying the effectiveness of a new drug, articles which report on the efficacy of this drug may be included.
<b>Publication date</b>	Depending on the research question, only new, only old, or all articles may be included.
<b>Article quality</b>	Consider the article quality, such as whether they have been peer-reviewed or have a high risk of bias.

*Table 2.1: Inclusion and exclusion criteria in the screening process.*

## 2.4 Copyright restrictions

Copyright restrictions can pose a problem under the screening process of a systematic review because they limit the access to certain articles. In many cases, articles are protected by legal agreements or licensing agreements that prohibit automated data extraction without permission from the copyright owners [9]. Additionally, even if data is publicly available, it may still be subject to copyright restrictions that limit how the data can be used. For example, a database of academic articles such as PubMed may be publicly available, but individual articles may require special permissions depending on

where the full-text is located. If data extraction is used to extract and use copyrighted data without permission it can raise ethical concerns and lead to legal disputes.

There are two ways of avoiding these issues. The first is to get access to the appropriate permissions and legal agreements before performing the data extraction. This may involve obtaining permissions from the owners of the article or journal of choice. However, this often requires a lot of work, especially when performing big systematic reviews. The second way is to only include sources of data that are freely available. The PubMed database as a whole does not have general copyright restrictions, but the restrictions vary depending on the article subset. The largest and most widely used subset of PubMed is called MEDLINE [10]. MEDLINE is however restricted to journal subscribers, meaning that special permissions are required.

If the article is open access in PubMed, it is most likely part of the digital archive called PubMed Central (PMC). Of the 35 million articles stored in PubMed, around 9 million of them are part of the PubMed Central Open Access (PMC-OA) subset [11]. What sets PMC apart is its focus on making valuable scientific research freely available. The open access principles, allows for the unrestricted use and distribution of the published articles. Data extracted from these studies are therefore copyright-friendly and can safely be used [12].

## **2.5 Article structure**

Originally, there were no academic conventions which ensured that research articles were built on the same rules. However, many journals have now developed standardized reporting guidelines or checklists, aiming to improve the quality and transparency of research articles [13].

The location of data in research articles can vary depending on the study design and the type of data being presented. The result section is typically the primary location.

Data is often presented in tables, figures, and graphs that illustrate the study findings. In addition, the result section often includes the descriptive statistics such as means and standard deviations. For information related to how the data was collected, the method section is a good place to look. Here data such as study design, sampling methods, and data collection instruments can be found. In addition, this section often includes information about the statistical analyses that were performed on the data.

If the data is not located directly in an article, it can either be in the supplementary materials or appendix. Many journals require authors to provide supplementary materials, such as additional tables or figures, as part of their submission. These materials provide additional detail about the data presented in the main text of the article. Appendices serve the same purpose and may include raw data, survey instruments, or coding manuals used in the study. The contents in the appendix can provide readers with additional information about the study and allow them to further explore the data presented in the main text of the article.

By having guidelines for reporting key information in a standardized way, it facilitates the interpretation and synthesis of research findings making the development of extraction tools easier.

## **2.6 Manual data extraction**

Manual data extraction refers to the process of gathering data by hand, without the use of automated tools or software. In this process, researchers manually look for data points such as study design, sample size, intervention details, outcome measures and statistical results in an article. These data points are written down and the process is repeated for each article included in the systematic review. Since researchers need to thoroughly read and understand each source to extract the desired data, it can be a time-consuming and labor intensive process.



An advantage of manual data extraction is that it allows researchers to evaluate contextual information, evaluate the quality of data, and make subjective decisions based on their expertise. When extracting data manually, several key steps are typically required. While the specific process may vary depending on the research question and methodology, the general steps involved are presented in *Table 2.2* [14].

<b>1) Define data extraction criteria</b>	Determine the specific data points and variables that need to be extracted from the study.
<b>2) Create an extraction form</b>	Develop a structured data extraction form that captures the predetermined data points (spreadsheets, tables, electronic forms).
<b>3) Extract data</b>	Systematically extract the relevant data from the study and record it in the data extraction form. Read through the study and identify the required information.
<b>4) Quality assessment</b>	Implement measures to ensure the quality and reliability of the extracted data.
<b>5) Data management</b>	Organize and manage the extracted data in a structured manner. With proper labeling, categorization and documentation.
<b>6) Data verification</b>	Double-check the extracted data for accuracy and completeness.
<b>7) Data synthesis</b>	The data that has been collected needs to be synthesized to be able to draw conclusions about the research question. This may involve statistical analysis or other methods of data synthesis.

*Table 2.2: Steps when performing a systematic review.*

## 2.7 Automated data extraction

Automation refers to the use of technology to perform tasks that were previously done manually. This involves the use of machines or software to perform tasks with little or no human intervention. The goal of automation is to increase efficiency and productivity by reducing errors and costs associated with human labor. In this section the focus is on which parts of the data extraction process that can be automated and the challenges associated with this process.

### 2.7.1 Parsers and data mining

A parser is a program or software component that analyzes the structure of a text and determines its grammatical structure according to a specific set of rules. The goal of an article parser is to automatically identify and extract key pieces of information, such as author names, affiliations, publication dates, titles, abstract, tables and keywords. This process is also referred to as data mining and simplifies the organization of information in a structured matter [15]. Article parsers can be used in combination with natural language processing techniques, machine learning algorithms, and rule-based approaches to analyze text and identify relevant information [16]. Parsing an article involves several steps, which can vary depending on the specific parser and the type of article being parsed. However, the steps shown in *Table 2.3* are typically involved.

<b>Preprocessing</b>	The article text is first preprocessed to remove any unwanted formatting or characters. The text may also be segmented into sentences or paragraphs, depending on the parser.
<b>Tokenization</b>	The article is then tokenized, which involves splitting it into individual words or tokens.
<b>Part-of-speech (POS) tagging</b>	Each token is assigned a part-of-speech tag, such as noun, verb, or adjective, based on its grammatical role in the sentence. POS tagging is useful for identifying key pieces of information such as names, dates or locations.

<b>Named-entity-recognition (NER)</b>	The article is analyzed to identify person names, organization names, or locations. This is typically done using machine learning algorithms, which can learn to recognize patterns and features that are indicative of named entities.
<b>Relation extraction</b>	Once the named entities have been identified, the relationship between them are extracted. This step may involve additional processing, such as syntactic parsing or semantic analysis.
<b>Structuring</b>	Finally, the extracted information is structured in a standardized format, such as a database, which can be analyzed or visualized.

*Table 2.3: Common steps performed when parsing an article.*

## 2.7.2 Existing technologies

Currently there are many data extraction tools available for public use. Some require paid subscriptions while others are open access. In addition, the tools can be categorized as either semi-automated or fully automated. Semi-automated extraction relies on software to automate some of the process while fully automatic extraction relies on software to extract all of the data without human intervention.

*Table 2.4* presents different extraction technologies. The choice of technology will depend on the data to be analyzed. Different tools vary in their setup cost and difficulty. In addition they have different ability to manage, present, store, and retrieve data.

Type	Examples	Description
Web scraping	BeautifulSoup, Scrapy, Selenium	Extraction of data from any website without the need for human intervention.
Screen scraping	WinAutomation, UiPath	Extract data from computer screens and other graphical user interfaces (GUI's)

Scientific Data extraction	Octoparse	Extract structured data from research articles, such as numerical data or tables. Using techniques such as data mining and machine learning to identify and extract the data.
ETL (extract, transform, load) tools	Talend, Informatica, Pentaho	These tools are used to extract data from multiple sources, transform it into a consistent format, and load it into a target database.
Web services	PubMed API	Allow for easy navigation and access of information stored in online databases. Provide consistent formatting and easy-to-use search features.
Language models	ChatGPT	Powerful language processing algorithm which can be trained on a variety of datasets.

*Table 2.4: Examples of available technologies for data extraction and data handling.*

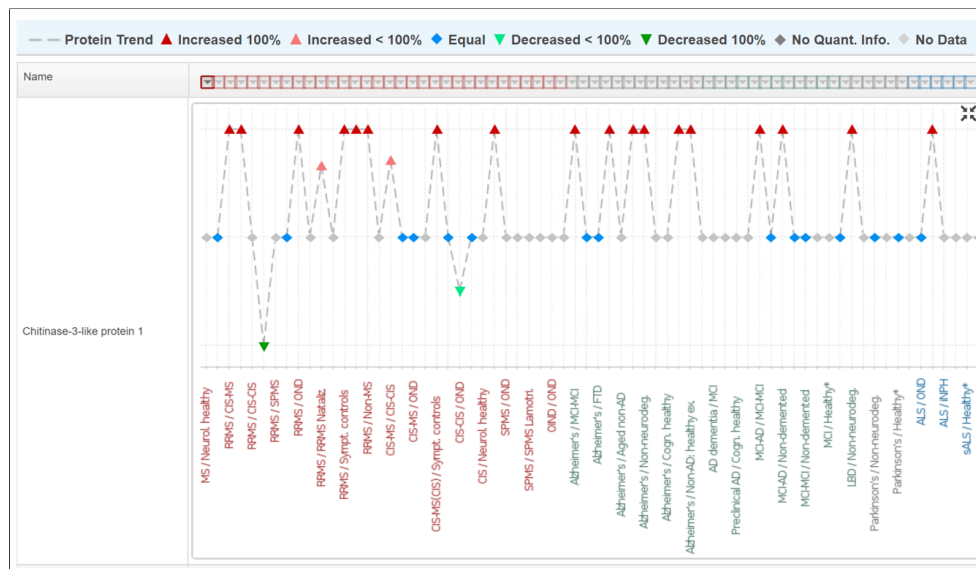
## 2.8 Data repositories

The extracted data is often stored in repositories. A data repository is a centralized location where data is stored and made available for access and sharing by researchers, scientists or other interested partners. Data repositories can take various forms, such as online databases, archives or libraries. They can be hosted by academic institutions, government agencies, non-profit organizations, or private companies.

Data repositories play an important role in promoting data sharing and transparency in scientific research. By providing a location for data storage and access, it makes it easier for researchers to access and use data from a wide range of sources, facilitating collaborations and interdisciplinary research. In addition, it also ensures the long-term preservation and accessibility of research data, ensuring that it remains available for future use. One way of expanding data repositories is by extracting data from articles and storing them in these repositories.

## CSF Proteome Resource

The intention behind the prototype is to collect information to populate the data repository called CSF Proteome Resource (CSF-PR) [17]. CSF-PR is an online data repository of mass spectrometry based proteomics experiments on human cerebrospinal fluid (CSF). CSF can give indications about the state of the central nervous system and is particularly relevant for neurodegenerative diseases such as Multiple Sclerosis (MS), Alzheimer's disease (AD), Amyotrophic Lateral Sclerosis (ALS) and Parkinson's disease (PD). In CSF-PR the regulation states of proteins are compared between control groups and different types of neurodegenerative diseases. The regulation data is collected from research papers and the metadata is used to see if a protein shows increased, decreased or no regulation between two groups. An example of how these regulation states are visualized in CSF-PR is presented in *Figure 2.4*.



*Figure 2.4: Screenshot of a protein plot in CSF-PR.*

The idea is that CSF-PR can be used to locate proteins that are associated with different diseases, ultimately leading to the development of new targeted medicine. This is dependent on the database including enough data to generalize the results and be updated with data from the most recent publications.

# Chapter 3

## Methods

### 3.1 File handling

An important step in automated data extraction is to understand how different file types impact the process. Here important aspects to take into consideration are which file types should be compatible with the tool, how the article files should be retrieved and the challenges associated with parsing different files. There are several common file types used in academic research. Different file types serve different purposes and in most cases articles have both a file containing the article text and supplementary files containing the data. Some of the most commonly used file types are listed in *Table 3.1*.

Filename	Name	Description
<b>PDF</b>	Portable Document Format	PDFs are often used in sharing research articles, reports, and other documents as they can be easily viewed across different platforms and devices.
<b>HTML</b>	Hypertext Markup Language	Popular file format used for publishing online publications. It gives flexibility when formatting and displaying articles on the web.
<b>XML</b>	Extensible Markup Language	XML files have similar features as HTML files, but are designed for storing and transporting data. However, they do not have formatting instructions.
<b>CSV</b>	Comma-Separated Values	CSV files are commonly used for sharing data as they can be easily imported and exported across different software platforms and databases.

---

*Table 3.1: Commonly used file types in scientific research.*

When submitting a manuscript for publication, authors are typically required to format their manuscript according to the guidelines of the target journal or publisher, which may include specific font sizes, margins and headings. The manuscript is then deposited through the journals submission system where they undergo conversions to be compatible with the journal database format. The journals have built in methods for displaying and formatting articles on the web. In most cases this will be as HTML files, but they will provide PDF versions when possible.

### **3.1.1 PDF**

PDF files are a popular file format used to present and exchange documents across different platforms and devices. One key feature of a PDF file is its ability to preserve the visual structure of a document, including fonts, images and formatting. This is achieved through the use of a fixed layout format with specific rules for element style, positioning and size.

However, a PDF file is binary in nature, meaning that specialized software such as Adobe Acrobat has to be used to present it. In addition, the visual representation of the content does not necessarily correlate with its logical structure. This has implications when extracting data since the data is not accessible using traditional parsing techniques [18, 19]. Since PDF files are designed for presenting data and not extracting data, computers will have a hard time localizing the data of interest.

There exist developed tools with machine learning algorithms that can parse PDF files, but these algorithms come with their own problems. Instead it is better to choose a file format where the structural information is preserved such as HTML or XML.

### 3.1.2 HTML

HTML is a markup language used for creating web pages and other types of digital documents that can be viewed in web browsers. HTML uses a series of tags and attributes to structure and format content within a web page. Tags are used to mark different types of content, such as headings, paragraphs, lists, images, and links. Attributes are used to provide additional information about the content, such as the size or color of an image, or the destination of a link.

To be able to extract information from HTML documents, an HTML parser has to be used. A good HTML-parser should be able to handle a wide range of HTML documents and accurately extract the relevant information. In addition, it should have methods for extracting different types of elements such as, text, images, links, and other multimedia elements. When extracting data from HTML documents the initial step revolves around filtering out elements that do not have semantic properties. After the initial filtering, the remaining elements should be split into text components, table components and figure components.

Since all articles in PubMed Central are published online, there is always a HTML version available. The HTML version of a file can be accessed through its unique URL (Uniform Resource Locators). When parsing HTML files, text elements can easily be identified and separated. Paragraphs have the tag "<p>" and headers have the tags "<h1> , <h2> , <h3>" where the numbers specify if it is a title, header or sub-header. This means that the underlying structure of an article can easily be retrieved and analyzed.

However, for components like images, tables and other non-textual elements this is a harder task. How the journal chooses to display an article is different from journal to journal. This impacts the way the HTML file is structured and has implications on the parsing algorithm. When performing filtering on tags, this will result in the parser sometimes filtering out too much and sometimes filtering out too little information. There has



to be a consistency to the way an article is presented and this becomes difficult if the tool is built on a web page parser.

Another limitation to building an extraction tool on a HTML parser, is that some articles are in journals that require paid subscriptions. Some journals have copyright restrictions which prevent a user from downloading articles. PubMed has an internal server listener that tracks the amount of times an article is downloaded. If they find the activity suspicious, it results in the IP address of the user being blocked.

### 3.1.3 XML

PubMed prefers that developers retrieve articles and article meta-data through their API service and they have multiple tools which simplify this process. However, through the PubMed API, article meta-data and full text are only available in XML-format. XML stands for "Extensible Markup Language" and has many similar features as HTML, but is not used for displaying web pages like HTML. This means that many HTML parsers also can parse XML files. XML is designed to store and transport data, making it a popular choice for data interchange between applications and systems. In XML, data is stored in a structured format that is both human-readable and machine readable. It contains tags and attributes that define the structure of the data, allowing it to be easily parsed and manipulated.

Creating a tool that is compatible with the PubMed API and able to parse XML files enhances overall performance at the expense of limiting article compatibility. The rules used to parse these files will only be valid for PubMed Central articles, meaning that it cannot be generalized to include other databases. However, the benefit of working with XML files is that there is less variability in the article structure. PubMed Central has strict rules on how the article's XML file should be built, with specific tags representing specific parts of the article [20]. This means that the extraction process becomes much easier. For example the abstract of an article is always stored with the tag "<abstract>".

In addition, through the PubMed ftp-service, a user can download a folder containing the article's XML file along with all images and supplementary files associated with the article. References to the position of these files are made inside the main XML file so it opens the possibility of also including figures when displaying the article in a graphical user interface (GUI).

In XML-files, the article meta-data is easily extracted resulting in an organized way of presenting multiple articles. By including pictures and supplementary files, as well as having an easy way of extracting different components, the inspection part becomes less prone to errors. In addition, by downloading the articles, it is no longer necessary to connect to the articles website, resulting in fewer parsing errors.

### 3.1.4 CSV

A CSV file is a plain text file format used to store tabular data. It is a simple and widely supported file format for data exchange between different applications and platforms. In a CSV file, each line represents a data record, and each record is divided into columns separated by a delimiter. This delimiter is usually a comma, but can also be tabs or semicolons.

In scientific research CSV files are used for data collection, analysis and sharing. They provide a convenient format for organizing and recording research data. Each row typically represents a data point or observation, and columns represent different variables or measurements. CSV files are compatible with statistical software and programming languages, opening up the possibility of performing data analysis or generate visualizations.

When extracting data from scientific publications, CSV files become a valuable resource for researchers. CSV files can be provided as additional material to the article text and can along with other tabular data provide the foundation of the topics discussed. There-

fore, it is equally important to be able to retrieve the supplementary files, as it is retrieving the article text.

## 3.2 PubMed API

PubMed provides several API's that allow software developers and researchers to access its data and build custom applications. The API's serve different purposes and when exploring the different API's available, there were especially three that could help our cause.

- i) **E-utilities:** Used for accessing and retrieving data from the PubMed databases
- ii) **ID converter:** Convert between ID's used in the different PubMed databases.
- iii) **PubMed Central API:** Download directories containing an articles XML file, figures and supplementary materials.

In order to use the PubMed API, the tool has to get an API key from the National Center for Biotechnology Information (NCBI). This key is used to authenticate and authorize access to the API. Once authenticated, the tool has access to performing queries in the database using a range of search parameters. These parameters includes keywords, publication dates, and author names. The PubMed API returns results in the form of metadata, which includes information such as article titles, authors, abstracts, publication dates and journal ID's. The metadata can then be used to retrieve the full text of articles from PMC.

### 3.2.1 E-utilities

Entrez Utilities (E-utilities) is a set of eight web based tools developed by NCBI that allows users to access and retrieve data from their databases. E-utilities allow users to search for information and retrieve records in XML format. In *Table 3.2* the E-utilities

API tools are presented.

<b>1) ESearch</b>	Allows users to search the PubMed database using specific search terms.
<b>2) EFetch</b>	Retrieves specific records from PubMed in various formats.
<b>3) ESummary</b>	Provides a summary of the data associated with a specific PubMed ID.
<b>4) ELink</b>	Provides links to related articles in PubMed, based on a specific PubMed ID.
<b>5) EGQuery</b>	Performs a Global Search of the NCBI databases.
<b>6) ESpell</b>	Provides suggestions for spelling errors in search terms.
<b>7) EInfo</b>	Provides metadata for a specific NCBI database.
<b>8) EPost</b>	Allows users to upload a set of PubMed ID's and retrieve data associated with those ID's.

*Table 3.2: Overview of the API-tools in E-utilities.*

Even though these tools can provide a range of features, the most powerful is ESearch. ESearch is built in a way that it performs a search similarly to how a search is performed in the web page, but everything inside of a single URL call. The result of the call is an XML file containing the PubMed ID's that are relevant to the input search query. Since the article subset of interest is the open access subset, the call can be modified to only retrieve the PubMed ID's where the full text is freely available. This is similar to performing a PubMed search in the web platform with the filter "Full-free text" (*Figure 2.3*). The benefit of performing the initial search for articles in the PubMed database and not directly in the PubMed Central database is so that there is a consistency between the articles presented in the tool and the articles presented in the web page.

The base URL of the E-search call is:

<https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?>

The initial part, in black, is shared by all E-utilities calls and the last part, in red, specify that it is the E-search engine that should be used. Parameters are chained onto the base call. Some of the most common parameters include:

`db=pubmed` - This parameter specifies the database to search. In this case the search will be performed in PubMed.

`&term=<keyword1>,<keyword2>` - This parameter specifies the search terms. Each term can be separated by a "," or the operator's "+AND+" or "+OR+" can be used.

`&retstart=<int>` - This parameter specifies the starting index of the results to return.

`&retmax=<int>` - This parameter specifies the ending index of the results to return.

`&datetype=mdat&mindate=2020/01/01&maxdate=2023/05/04` - This parameter specifies the date range of articles to search. In this case only publications from 2020 to 2023 will be retrieved.

`&retmode=xml` - Specifies the return mode of the call. By default this is in XML.

### 3.2.2 ID converter

Since the initial search was done in PubMed and not in PMC, the resulting ID's are PubMed ID's and not PMC ID's. The PubMed identifications and the PMC identifications are different so to be able to locate the article file in the PMC database, the ID's have to be converted. PubMed provides multiple ways of doing this, but the easiest way is to use their ID Converter. This ID converter takes one or multiple ID's and outputs the article ID's in all PubMed databases. By parsing the output XML file, the PMC ID's can be obtained.

The ID converter API is the backend service that is used by the web-based PMC ID Converter [21]. This API allows users to convert between the various IDs used in the PubMed system. The base URL of this call is:

---

```
https://www.ncbi.nlm.nih.gov/pmc/utils/idconv/v1.0/?
```

All calls require the registered name of the tool and email to identify the application making the request. The only other parameter is the parameter: `ids=<ids>`. Here the Ids that are going to be converted are separated by commas.

### 3.2.3 PMC OA Web Service API

Finally, the last part is finding the downloadable file using the PMC OA Web Service API. This API allows users to discover downloadable resources from the PMC Open Access Subset. These articles are stored in an online database and accessible through the PMC FTP Service as tgz (tar,gzipped) format, or, for those articles that have them, in PDF as well.

The base URL for this service is:

```
https://www.ncbi.nlm.nih.gov/pmc/utils/oa/oa.fcgi?
```

Adding the parameter `id=<PMCID>` to the request causes the service to return a result set response which includes information about the record in the database. This will provide the article's citation, license, and retraction status, as well as information about any downloadable resources.

# Chapter 4

## Results

The goal of this thesis was to enhance biomedical data extraction from scientific publications with a special emphasis in extracting data for usage in the online data repository, CSF-PR. This involved exploring different technologies and determining how they could be utilized when creating our prototype. Overall, the problem can be divided into: i) how we can determine the relevancy of an article to an input query and ii) how we can efficiently perform the inspection and extraction of data. In the following chapter the developed prototype will be presented along with the development process. Because of the prototype's ability to explore articles in PubMed Central, it has been given the name "PMC Explorer".

The Java source code along with a step by step guide on how to use it can be obtained here: <https://github.com/barsnes-group/PMCEXplorer>

### 4.1 Project structure

PMC Explorer was built in an object oriented manner meaning that different classes are responsible for storing and presenting different information. The visual part is managed by the four classes MultipleArticles, SingleArticle, TablePanel and TextPanel. These classes have the methods responsible for creating containers for displaying the text, creating the tables and performing actions on button click. However, the information and algorithms for retrieving the information to fill these containers are stored in other classes. In *Figure 4.1* a class diagram shows how the different classes interact with each other to

retrieve the information.

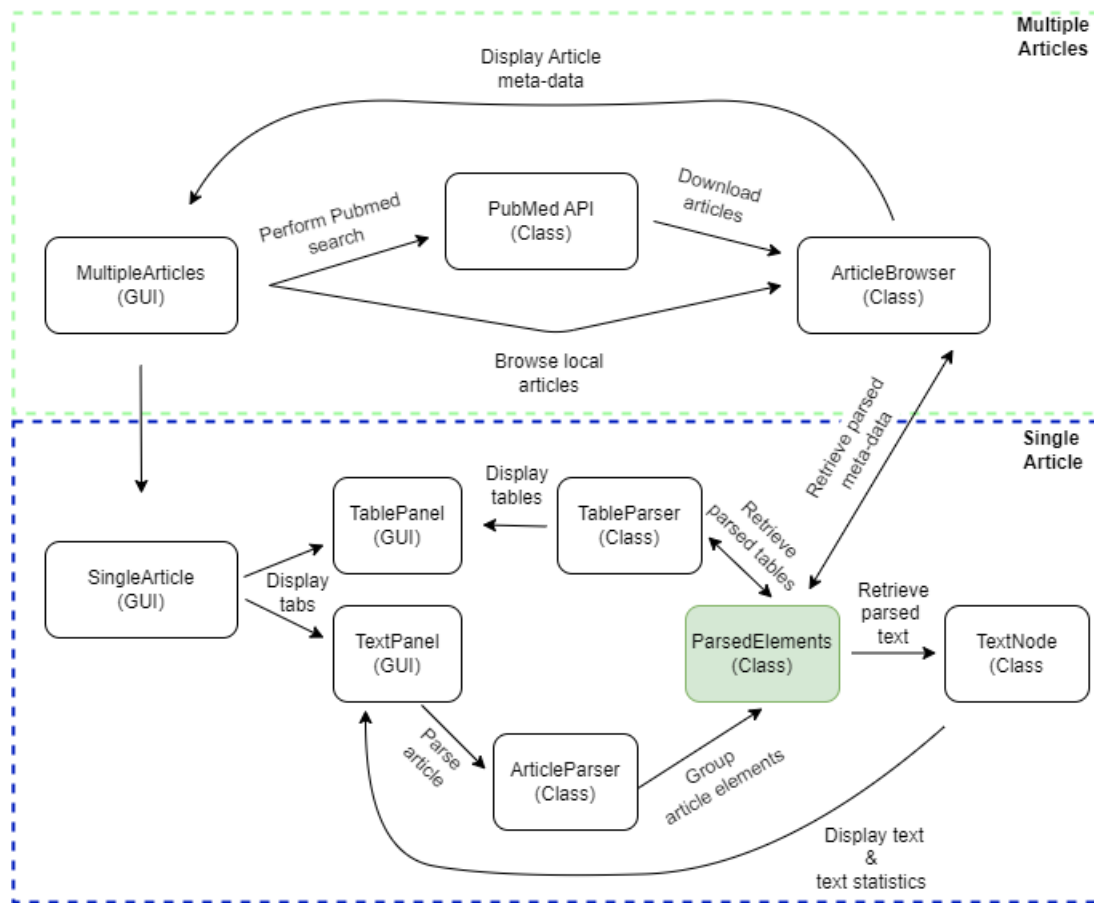


Figure 4.1: Diagram of the project structure.

The ArticleBrowser class is responsible for collecting the meta-data and setting the relevance status of each article that is displayed in the MultipleArticles class. The articles to be displayed are either selected through the PubMed API or from the local drive. For each article the meta-data is retrieved by parsing the XML file. The parsing is performed in the ParsedElements class where information such as title, author and year is retrieved. However, information such as article status and keyword frequency needs further processing for it to be available. This processing is done in the TextNode class.

Since the ParsedElements class contains the parsed article it can be regarded as the main component. The parser used is called Jsoup and it has many powerful methods for element selection [22]. In this class text-elements, headers, tables and meta-data are extracted separately and prepared for further processing. Since XML-files do not contain



formatting instructions, the extracted elements had to be converted to HTML to make them displayable in the GUI. This process involved changing tag-names and making the figures HTML compatible.

The parsed text-elements are passed on to the TextNode class. Here the text is tokenized and the keywords and most frequent words are counted. The parsed tables are passed on to the TableParser class where they are converted to a standardized format which allows for column selection in the GUI.

## 4.2 Retrieving the articles

Initially the article data was retrieved by parsing HTML pages. However, because of the limitations discussed in section 3.1.2, the choice was to instead use the PubMed API. Since the PubMed API is structured using XML files, an important step is to describe how an XML parser can be used to download the articles of interest. There are four instances when an XML parser becomes useful:

### 1) Performing a search query in E-utilities

An example of an E-utilities search in the PMC subset using the keyword "CSF-PR" can be seen in *Figure 4.2*. The hierarchical structure of this XML-file makes it easy to locate relevant information. In this file the information of interest are the ID's which are in the group "<IdList>" with the tag "<Id>".

```
▼ <eSearchResult>
  <Count>2</Count>
  <RetMax>2</RetMax>
  <RetStart>0</RetStart>
  ▼ <IdList>
    <Id>32963504</Id>
    <Id>25038066</Id>
  </IdList>
  <TranslationSet/>
  <QueryTranslation>"CSF-PR"[All Fields] AND "pubmed pmc open access"[Filter]</QueryTranslation>
</eSearchResult>
```

*Figure 4.2: Example of the output of an E-utilities search in the PMC article subset.*

## 2) Converting the PubMed ID's to PMC IDS

After the ID's have been retrieved the next step is to convert the ID's. This can be done with the ID converter API and the result can be seen in *Figure 4.3*. The different journal ID's are stored as record attributes. To extract the PMC ID, the parser has to locate the "<record>" tag and select the attribute "pmcid".

```

▼<pmcids status="ok">
  ▼<request idtype="pmid" pmids="" versions="yes" showaid="no">
    <echo>tool=DataExtractorUIB;email=maberggrav%40gmail.com;ids=32963504%2C%2C25038066</echo>
  </request>
  ▼<record requested-id="25038066" pmcid="PMC4223498" pmid="25038066" doi="10.1074/mcp.M114.038554">
    ▼<versions>
      <version pmcid="PMC4223498.1" current="true"/>
    </versions>
  </record>
  ▼<record requested-id="32963504" pmcid="PMC7499868" pmid="32963504" doi="10.1186/s12014-020-09296-5">
    ▼<versions>
      <version pmcid="PMC7499868.1" current="true"/>
    </versions>
  </record>
</pmcids>

```

*Figure 4.3: Example of the output of the PubMed ID converter.*

## 3) Retrieving the downloadable file

Now that all the ID's have been located and converted, the third step is downloading the article folder. For each PMC ID retrieved in the previous step, an URL call has to be made. An example of how the output looks like is shown in *Figure 4.4*. The XML can be located by following the link provided under the tag "<link> href=". Since the file is compressed using both .tar and .gzip, it has to be unpacked twice before further processing.

```

▼<OA>
  <responseDate>2023-04-29 11:16:25</responseDate>
  <request id="PMC4223498">https://ncbi.nlm.nih.gov/pmc/utis/oa/oa.fcgi?id=PMC4223498</request>
  ▼<records returned-count="1" total-count="1">
    ▼<record id="PMC4223498" citation="Mol Cell Proteomics. 2014 Nov 18; 13(11):3152-3163" license="none" retracted="no">
      <link format="tgz" updated="2016-11-07 14:51:12" href="ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_package/7f/29/PMC4223498.tar.gz"/>
    </record>
  </records>
</OA>

```

*Figure 4.4: Example of how to retrieve the downloadable article file.*

## 4) Parsing the article XML-file

The downloaded article folder contains the figures used in the article, the supplementary files and an XML file with the articles text and structural information. The XML file follows PubMeds guidelines for tagging meaning that there is a consistency to how the elements are structured. There are three main components to each file:

- `<front>`: Includes the article meta-data, licensing status and abstract. Here information such as author, publication date and article-id can be extracted.
- `<body>`: Contains the body of the article with headers, paragraphs and references to figures included in the article folder.
- `<back>`: Contains the remaining information associated with the article such as references, supplementary material and acknowledgments.

Parsers often provide methods for navigating an XML file, making it possible to locate and extract data from different levels. In *Figure 4.5* a part of an article XML is provided along with how the article title can be located. To extract the title from this segment, requires the parser to look in the `<front>` component, then move into the `<title-group>` component and finally into the `<article-title>` component.

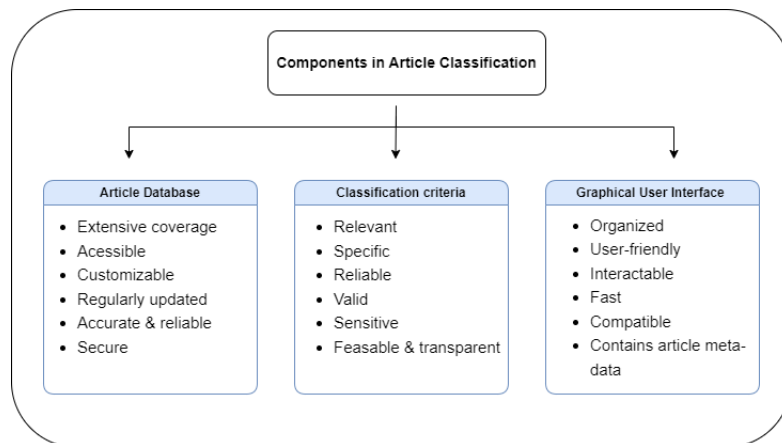
```

▼ <front>
  <journal-meta>
    <journal-id journal-id-type="nlm-ta">Clin Proteomics</journal-id>
    <journal-id journal-id-type="iso-abbrev">Clin Proteomics</journal-id>
    ▼ <journal-title-group>
      <journal-title>Clinical Proteomics</journal-title>
    </journal-title-group>
    <issn pub-type="ppub">1542-6416</issn>
    <issn pub-type="epub">1559-0275</issn>
    ▼ <publisher>
      <publisher-name>BioMed Central</publisher-name>
      <publisher-loc>London</publisher-loc>
    </publisher>
  </journal-meta>
  ▼ <article-meta>
    <article-id pub-id-type="pmid">32963504</article-id>
    <article-id pub-id-type="pmc">7499868</article-id>
    <article-id pub-id-type="publisher-id">9296</article-id>
    <article-id pub-id-type="doi">10.1186/s12014-020-09296-5</article-id>
    ▼ <article-categories>
      ▼ <subj-group subj-group-type="heading">
        <subject>Research</subject>
      </subj-group>
    </article-categories>
    ▼ <title-group>
      <article-title>Development of robust targeted proteomics assays for cerebrospinal fluid biomarkers in multiple sclerosis</article-title>
    </title-group>
  
```

*Figure 4.5: Locating the title in an articles XML file.*

### 4.3 Multiple article inspection

In general there are three components that come into play when making this part of the prototype. The first part is selecting a database and making the prototype compatible with the articles contained in this database. The second is generating a classification criteria. PMC Explorer is not supposed to replace the already existing algorithm in PubMed. Instead it will provide a more specific analysis of the articles that have been determined to be relevant. The final component is an organized GUI which gives the user extensive enough information about an article, that it can be deemed relevant without further inspection. The three components and their characteristics are shown in *Figure 4.6*.



*Figure 4.6: Diagram over components in article classification and their characteristics.*

To put things into perspective let's look at how data can be collected for CSF-PR. CSF-PR contains data regarding proteins regulation state in different neurodegenerative diseases. Keywords here would include for example "cerebrospinal fluid", "Parkinson's disease", "Alzheimer's disease" or "Multiple Sclerosis". Performing a PubMed search with any combination of these keywords or other keywords relevant to the topic, would end up displaying the articles the PubMed algorithm determined to be relevant. However, how relevant these articles are to the input keywords and which keywords that got emphasized the most is not known. PubMed gives an article a high relevancy score if the input

keywords can be found in the title, abstract and full-text [7]. However, in some cases even though a keyword is present in all these parts, the article could still not be relevant enough to pass the screening process. Therefore a more sophisticated algorithm can give researchers a more reliable way of finding relevant articles.

### 4.3.1 Presenting the articles

The initial window presented to the user when opening PMC Explorer is shown in *Figure 4.7*. In this example, the articles are first downloaded using the keywords; Multiple Sclerosis, Parkinson's disease and Alzheimer's disease. Then to determine relevancy the abbreviations ms, pd and ad are used as search criteria on the downloaded files. The GUI class that holds the visual information in this window is called MultipleArticles. Here information such as author, title and year are presented for each article giving the user the ability to easily distinguish between the different articles.

Input keywords:  
ms,pd,ad

Browse Local Files  Browse PubMed

RUN

Nr	Notes	Author	Title	Year	PMC-id	Status	Frequencies	View Article
1	Notes	Li et al.	The role of plasma cortisol in dementia, epilepsy, and multiple sclerosis: A Mendelian randomization study	2023	PMC10050717			Inspect Article
2	Notes	Hu et al.	Infections among individuals with multiple sclerosis, Alzheimer's disease and Parkinson's disease	2023	PMC10053639			Inspect Article
3	Notes	Kesidou et al.	CNS Ageing in Health and Neurodegenerative Disorders	2023	PMC10054919			Inspect Article
4	Notes	Kountouras et al.	Controlling the Impact of Helicobacter pylori-Related Hyperhomocysteinemia on Neurodegeneration	2023	PMC10056452			Inspect Article
5	Notes	Kurowska et al.	The Role of Diet as a Modulator of the Inflammatory Process in the Neurological Diseases	2023	PMC10057655			Inspect Article
6	Notes	Redeňek Trampuž et al.	Shared miRNA landscapes of COVID-19 and neurodegeneration confirm neuroinflammation as an important overlapping feature	2023	PMC10064073			Inspect Article
7	Notes	de Oliveira et al.	The impact of the COVID-19 pandemic on neuropsychiatric and sleep disorders, and quality of life in individuals with neurodegenerative and demyelinating diseases: a systematic	2023	PMC10091330			Inspect Article
8	Notes	Bianco et al.	Sex and Gender Differences in Neurodegenerative Diseases: Challenges for Therapeutic Opportunities	2023	PMC10093984			Inspect Article
9	Notes	Russo et al.	Chitinase Signature in the Plasticity of Neurodegenerative Diseases	2023	PMC10094409			Inspect Article
10	Notes	Wijeweera et al.	Therapeutic Implications of Some Natural Products for Neuroimmune Diseases: A Narrative of Clinical Studies Review	2023	PMC10118888			Inspect Article
11	Notes	Chauhan et al.	Comparative in-silico analysis of microbial dysbiosis discern potential metabolic link in neurodegenerative diseases	2023	PMC10126365			Inspect Article
12	Notes	Yang et al.	Active constituents of saffron (Crocus sativus L.) and their prospects in treating neurodegenerative diseases (Review)	2023	PMC10127217			Inspect Article
13	Notes	Qian et al.	Large-Scale Integration of Single-Cell RNA-Seq Data Reveals Astrocyte Diversity and Transcriptomic Modules across Six Central Nervous System Disorders	2023	PMC10135484			Inspect Article

PREVIOUS 0/20 NEXT

*Figure 4.7: Presenting multiple articles.*

Articles are displayed based on which of the two criteria are chosen when the search is performed. The first one, called "Browse files", displays the articles that have been downloaded and gives the user the flexibility of performing multiple search queries with

different keywords on the local article files. The second one, "Browse PubMed", connects to the PubMed API and downloads open access articles relevant to the input keywords. The idea here is that the users can first download articles from the PMC database that are relevant to their research question. Then the user can explore the downloaded articles using different keywords to get a better understanding of what the articles are about.

The status column and frequencies column can be used to determine if an article is relevant based on the search performed. The status column indicates if an article has been determined as relevant by PMC Explorer's algorithm. An article gets the status green if one of the keywords used is part of the most frequent words in the text, if not, it presents the color yellow. The status column alone does not provide information on which of the keywords that triggered a green color, only that one of them did. Therefore, the frequencies column was made to give more information about each individual keyword.

Title	Year	PMC-id	Status	Frequencies	View Article
The role of plasma cortisol in dementia, epilepsy, and multiple sclerosis: A Mendelian randomization study	2023	PMC10050717	Green	Light Green, Yellow, Light Green, Yellow, Light Green	Inspect Article
Infections among individuals with multiple sclerosis, Alzheimer's disease and Parkinson's disease	2023	PMC10053639	Green	Dark Green, Dark Green, Dark Green, Dark Green, Dark Green	Inspect Article

Figure 4.8: Comparing relevancy of two articles.

The Frequencies column displays a colored box for each of the input keywords used. Each box is a shade of green, and the darker the color, the more frequent the keyword. In Figure 4.8 the status and frequencies of two articles is presented. The keywords used are the same as in Figure 4.7. By analyzing this figure one can see that all the keywords are more prevalent in the second article compared to the first. This may not be of significance because maybe the second article is longer or the author is more fond of using these abbreviations. The more interesting part is however to look at the differences in frequencies in a single article. In the first article the first keyword (ms) and the last keyword (ad) have a more intense green color than the middle keyword (pd). This can indicate that this article is not really about Parkinson's disease. By changing the keywords and looking at the resulting frequencies, the user can explore multiple articles at

the same time.

If a search displays something interesting on a specific article, the user can open the "Notes" cell in the article row. The "Notes" cell accesses the articles unique text file where the user can write down interesting observations. For example in *Figure 4.8*, an interesting observation is that the keyword "pd" is not mentioned many times in the text so this article does not probably contain data associated with Parkinson's disease. In addition a user can write down other comments which can aid them in the research process. For example, if the article has been analyzed before, it could be smart to write a comment so that an article does not get reviewed twice.

After the articles have been explored and those of interest selected, the next step is to inspect the articles of choice. In the table there are two options that give this possibility. The first option is to click on the PMC-id of the article. This will open the article in the PMC website and gives the user the possibility of reading the original article. The other option is to open the inspection window of the prototype. This gives a more advanced view of the article, with highlight possibilities, keyword statistics and data extraction options.

#### **4.4 Single article inspection**

At first, the goal was to see what parts of the process could be automated and what parts required human involvement. The underlying problem was that the relevant data was not always located in the same place and different terminology was used within the same topic. Instead of using advanced natural language processing algorithms PMC Explorer's main feature would be to in an easy way highlight the key information in an article.

The initial step was to design a feature responsible for highlighting keywords and displaying the number of times this keyword occurs in the text. In this way the user could

interact with the prototype to find which keywords best fits the text. In addition, by tokenizing the text and keeping track of high occurring words, the user could compare the keywords they regarded as relevant to their research question to the most occurring words in the text. If the input keywords are part of the most occurring words in the text, it may signalize that this text is relevant to the research question. To get a better understanding of the context in which the keywords occur, it would also be feasible to have an algorithm that extracts the sentences in which the keywords occur. To implement these features one would have to iterate the text, keep track of each word, keyword and sentence.

Another important aspect of PMC Explorer would be it's graphical user interface (GUI). Many already existing tools have advanced GUI's with tons of features which in the end could be more confusing than helpful especially for an untrained eye. Therefore, a key aspect of PMC Explorer would be to keep the features simple and easy to learn. For example data is often located in specific parts of the text such as in the tables, supplementary materials or the result section. A way to give the user a better overview of the text could be to split the text into components and display the different parts separately. In addition, it would be smart to have features which are responsible for customizing the visual parts such as highlight color.

Conclusively, to implement the part responsible for article inspection, there are three required components. Firstly, a parser to distinguish between text components, tables and figures. Secondly, a text iterator responsible for finding keywords, their indices and sentences in which they occur. Thirdly, a user-friendly graphical user interface. The three components and their characteristics is shown in *Figure 4.9*.



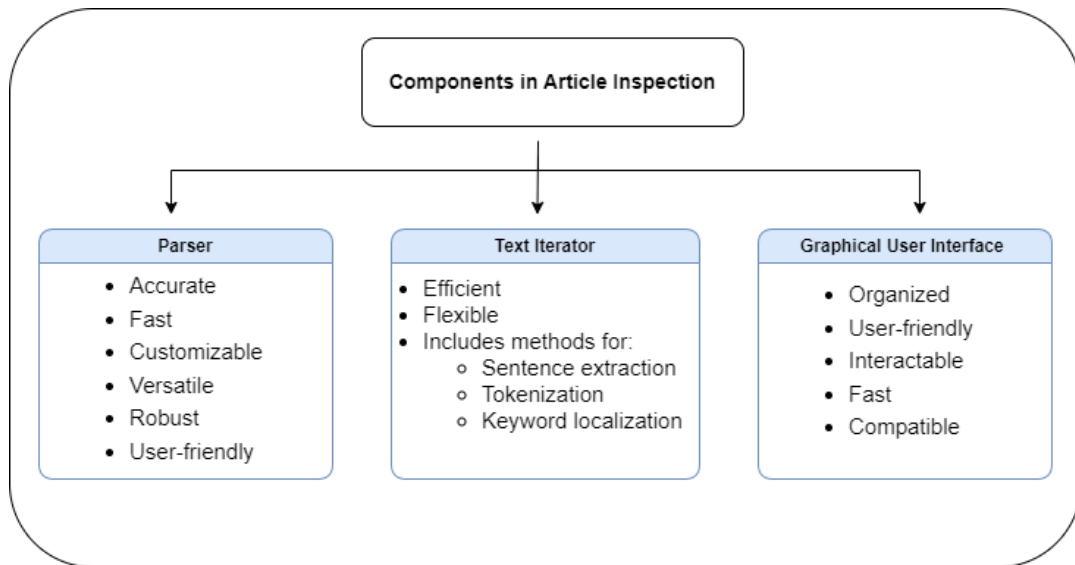


Figure 4.9: Diagram over components in article inspection and their characteristics.

#### 4.4.1 Graphical user interface

To better explain the functionalities of the inspection part of PMC Explorer, the first article in *Figure 4.7* will be used as an example. This article is written by Li et al. and is called "The role of plasma cortisol in dementia, epilepsy, and multiple sclerosis: A Mendelian randomization study" [23]. When opening the inspection window, the user is presented with a frame consisting of an area for changing input keywords, two tables and four tabs that display different parts of the article.

The first tab displays the abstract of the article, the second displays the article text, the third tab displays the relevant sentences in the text and the fourth tab displays the tables in the article. The visual components of this frame is operated by three classes, one which controls the layout and initialization of the keyword search, another which is responsible for presenting the tabs containing text and the last responsible for presenting the tab containing tables. An example of how the components are presented to provide the inspection part is provided in *Figure 4.10*.

Input keywords:  
ms, pd, ad

RUN NOTES

ABSTRACT BODY SENTENCES TABLES

### The role of plasma cortisol in dementia, epilepsy, and multiple sclerosis: A Mendelian randomization study

1

#### Introduction

Dementia, epilepsy and multiple sclerosis (MS) are common neurological disorders for which there are reports of associations with endocrine or immune markers. Hormones are one of the major physiological regulators of brain development, and endocrine and immune system dysfunction may contribute to the development of dementia, epilepsy and MS (1–3).

Approximately 55 million individuals worldwide suffer from dementia, and that number is expected to rise to 78 million by 2030 and 152 million by 2050 (4, 5). In 2019, the number of people with dementia disability globally exceeded 25.27 million (6). Dementia ranks 7th among the top 10 causes of death worldwide, with over 1.62 million deaths from dementia (7). Dementia can be divided into degenerative and nondegenerative categories. The former involves Alzheimer's disease (AD), dementia with Lewy bodies (DLB), Parkinson's disease with dementia (PDD) and frontotemporal dementia (FTD), while the latter includes vascular dementia (VD). AD accounts for 50%–70% of all dementia types (8). DLB is second only to AD in prevalence and accounts for 10%–15% of dementia (9). PDD accounts for approximately 3%–4% of dementia (10). FTD accounts for 5%–10% of dementia (11), and VD is the most widespread type of dementia caused by a nondegenerative disease, accounting for 15%–20% of patients with dementia (12). Dementia and cognitive impairment are major global issues as the world's population ages. Many clinical studies have shown that plasma cortisol has a direct relationship with cognitive impairment, and some observational studies have indicated a causal effect between plasma cortisol and dementia (13–15).

Epilepsy is a common neurological disorder worldwide and affects approximately 70 million individuals globally (16). Every year, there are 34 to 76 new cases diagnosed per 100,000 people (17). Epilepsy has high rates of disability and death, and the social and psychological burdens of the disease are severe, which seriously affects the quality of life of individuals with epilepsy (18). In recent years, many scholars have discovered that epilepsy has biological rhythms that are similar to the circadian rhythms of plasma cortisol (19). The circadian rhythm of plasma cortisol concentrations may affect the balance of neuronal excitability and inhibition, and a correlation exists between plasma cortisol and epilepsy susceptibility (20, 21).

Multiple sclerosis is an immune-mediated demyelinating disease of the central nervous system. It primarily affects individuals between the ages of 20 and 40 and is a major cause of disability in young adults, with serious social and economic burdens (22, 23). The Global Burden of Disease Study reports that the age-standardized MS prevalence is highest in high-income regions of North America, at 164.6 per 100,000 people, and is lowest in Asia (24, 25). Some studies have shown that hypothalamus-pituitary-adrenal (HPA) axis dysfunction is associated with the triggering of or increases in MS symptoms (26, 27).

Traditional observational research designs are case-control studies, whose findings are frequently influenced by confounding factors and where causal effects cannot be determined. In addition, randomized clinical trials can be limited by required ethical considerations. Therefore, valid strategies must be developed to identify causal relationships between exposure factors and response outcomes. Mendelian randomization (MR) is based on Mendel's law of segregation, which states that allele pairs separate or segregate during gamete formation and randomly unite at fertilization (28). The single nucleotide polymorphisms (SNPs) of genes are determined at birth and are not altered by interference from external environmental and behavioral factors, thus, SNPs are considered instrumental variables (IVs), which can be used to enhance the exposure-outcome relationship, prevent reverse causality in the exposure-outcome association and avoid reverse causality. We implemented a two-sample MR analysis using summary statistics and attempted to explore the causal effect of plasma cortisol on dementia, epilepsy, and MS.

2

#### Methods

2.1

##### Study design and MR assumptions

#### Keyword Frequencies:

Nr	Selected	Word	Frequency	Highlight
1	<input checked="" type="checkbox"/>	ad	49	
2	<input checked="" type="checkbox"/>	pd	14	
3	<input checked="" type="checkbox"/>	ms	35	

#### Most Frequent Words:

Nr	Selected	Word	Frequency	Highlight
1	<input type="checkbox"/>	cortisol	76	
2	<input type="checkbox"/>	plasma	60	
3	<input type="checkbox"/>	dementia	32	
4	<input type="checkbox"/>	epilepsy	32	
5	<input type="checkbox"/>	study	27	
6	<input type="checkbox"/>	ms	22	
7	<input type="checkbox"/>	results	22	




Figure 4.10: Inspecting a single article.

The initial layer of the frame is controlled by the class called `SingleArticle` and it is here the user can determine which keywords they want to know more about. Keywords can be separated by using the operators `,` `"` or `+`. Comma should be used as the default operator, but `+` can be used if the user wants further customizability of the extracted sentences. Using `+` between two words will only return the sentences that contain both the keywords. This can be used to for example locate sentences which for example compare two neurological diseases.

By clicking "RUN", three instances of the class `TextPanel` and one instance of the class `TablePanel` gets initialized. The `TextPanel` class is responsible for presenting and formatting the textual information and associated frequency tables. The `TablePanel` is responsible for formatting the in-text tables and the operations that can be performed on these tables. Each time a new search query is performed the article has to be reprocessed to locate the new keywords and indices used for highlighting.

### Text Panel

In *Figure 4.11* the table containing the keyword frequencies is presented. This table contains an option for selecting which keywords that should be highlighted and an option for changing the highlight color. In addition, it contains the number of times each keyword occurs in the text with a green bar making it easy to locate the most frequent one. By changing highlight colors and selection property of a keyword the user has the possibility of easily locating the data associated with the different keywords.

Keyword Frequencies:				
Nr	Selected	Word	Frequency	Highlight
1	<input checked="" type="checkbox"/>	ad	49 	<input type="text" value=""/> ▾
2	<input checked="" type="checkbox"/>	pd	14 	<input type="text" value=""/> ▾
3	<input checked="" type="checkbox"/>	ms	35 	<input type="text" value=""/> ▾

*Figure 4.11: Overview of the table presenting the keyword frequencies.*

In *Figure 4.12* the table containing the most frequent words is presented. The idea behind this table is to give the user a general idea of what the text is about. This table has the same functionalities as the keyword table, but by default the words are not selected. The intention is for the user to use both tables to determine which keywords fit the text best and simplify the process of locating the data of interest. For example the word "results" is not often used in a search query, but may be interesting to highlight. By highlighting for example "ms" and "results" it will be easy to localize which part of the text that presents the results associated with multiple sclerosis.








Most Frequent Words:					
Nr	Selected	Word	Frequency	Highlight	
1	<input type="checkbox"/>	cortisol	70 	<input type="text"/>	<input type="button" value="v"/>
2	<input type="checkbox"/>	plasma	60 	<input type="text"/>	<input type="button" value="v"/>
3	<input type="checkbox"/>	dementia	32 	<input type="text"/>	<input type="button" value="v"/>
4	<input type="checkbox"/>	epilepsy	32 	<input type="text"/>	<input type="button" value="v"/>
5	<input type="checkbox"/>	study	27 	<input type="text"/>	<input type="button" value="v"/>
6	<input type="checkbox"/>	ms	22 	<input type="text"/>	<input type="button" value="v"/>
7	<input type="checkbox"/>	results	22 	<input type="text"/>	<input type="button" value="v"/>

Figure 4.12: Overview of the most frequent words in an article text.

The frequency algorithm used for counting occurrences in *Figure 4.12* is different than the one used in the keyword table (*Figure 4.11*). When creating this table, the text is whitespaced tokenized, leading and trailing symbols are removed, stop-words are excluded, and each word with meaning is counted. However, the algorithm responsible for the keyword-counting also takes into account that the keyword can be part of a word. This means that it is more sensitive and will often have higher counts. For example the keyword "ms" has a frequency of 35 in *Figure 4.11*, but has a frequency of 22 in *Figure 4.12*. This is especially relevant when regarding words such as for example "protein". The first table will include the word "proteins" as an instance of the word "protein" since "protein" is part of the word "proteins". However, the second table will count these two occurrences as different words.

### Table Panel

Column selection is done in the GUI, by holding "ctrl" while selecting the columns. The selected columns can then be extracted by clicking the "SAVE" button. This will save the selected columns in the "Article\_Notes.txt" file. In *Figure 4.13* an example of a table is presented with the first and third column selected.

Table 1 : The characteristics of genome-wide association studies on outcome.

Outcome	Sample size for th...	SNP	Consortium	Ethnicity	Cases	Controls
Alzheimer's disease	26,757	283,086	10,894,596	UKB	European	
Vascular dementia	881	211,508	16,380,457	FinnGen	European	
Parkinson's diseas...	267	216,628	16,380,459	FinnGen	European	
Dementia with Le...	2,591	4,027	7,593,175	NA	European	
Frontotemporal de...	515	2,509	494,577	NA	European	
Epilepsy	929	212,532	16,380,452	FinnGen	European	

Selected columns:

Figure 4.13: Example of how tables are presented in the GUI.

When saving the relevant columns, the data is stored as tab-separated values in the "Article\_Notes" text file. However, since the tables originally are rendered in HTML, it may sometimes result in errors saving special characters because HTML has different character encodings for special characters. In Figure 4.14 the extracted columns from Figure 4.13 are presented. This feature allows for filtering of irrelevant table columns and gives a compact view of the data of interest.

```

Article_Notes.txt
File Rediger Vis

Outcome                               SNP
Alzheimer's disease                    283,086
Vascular dementia                       211,508
Parkinson's disease with dementia       216,628
Dementia with Lewy bodies                4,027
Frontotemporal dementia                 2,509
Epilepsy                                212,532
Multiple sclerosis                       26,703

```

Figure 4.14: Example of output when storing selected article columns.

If the user is interested in extracting data from the supplementary materials, they can locate the files by navigating to the "PMC\_Explorer\_Downloads" folder created when downloading the articles. The directory containing the article data is named after the PMC-id and will in this case be "PMC10050717". The supplementary materials are often stored as excel files which include own methods for data handling.

# Chapter 5

## Discussion

### 5.1 Human-machine interaction

The PMC Explorer is in general semi-automated since it combines elements of manual and automated processes. While the articles are retrieved and parsed automatically, the localization and extraction of data is managed by the user. The benefits of making the user in charge of the management and oversight of the extraction process is that it gives greater flexibility and control. In addition, the option to review and adjust the highlighted data improves accuracy. Since how data is presented in an article varies from article to article, it is beneficial to judge and interpret it before extraction.

A semi-automated tool gives more flexibility and control because it allows researchers to leverage the benefits of both manual and automated processes. With a fully-automated tool, the software algorithms are designed to automatically identify and extract data based on pre-defined rules or criteria. While this can be useful in some situations, it can also be limiting, as the algorithms do not always accurately identify all the relevant data. In addition, the context in which the data occur can be misinterpreted which impacts the output of the algorithm. In contrast, by having a human-machine interaction, researchers can automate parts of the extraction process, while still retaining the ability to manually review and adjust the extracted data.

When dealing with complex or nuanced data, human intervention and oversight can be greatly beneficial. Complex data includes things like unstructured text, inconsistent data

formatting, or data that requires expert knowledge to interpret. Fully automated tools can struggle with accurately extracting data in these situations, as they rely solely on pre-defined rules and algorithms to perform the extraction.

The combination of automated data extraction with human validation and correction takes advantage of the researchers expertise and judgment in the research process. The researcher can make adjustments to the extraction parameters and in this way tailor the parameters to the specific article being analyzed. The adjustments of parameters in PMC Explorer includes modifying the keywords used in the database search and later determining a set of keywords that best highlight the data of a single article.

Since PMC Explorer's main feature is highlighting key findings in an article, it does not have the same data processing steps that other extraction tools often have. In most cases, extraction tools process an article and automatically retrieves the relevant data of this article. When the data is extracted the user can review the extracted data and give feedback. This feedback refines the extraction algorithm leading to more accurate attempts. This however requires knowledge on how to give feedback to a machine learning algorithm. Algorithms are sensitive to feedback and giving it inaccurate or biased feedback may lead to these inaccuracies influencing the results [24].

## **5.2 Benefits of using an API**

In 2022, PubMed updated its E-utilities service with the newest technology in database searching. This update came so that the API service was up to date with the web version of PubMed released in 2020. This update ensured that the results returned when using the API were consistent with the results returned when using the web platform. Even though PMC Explorer does not use the full potential of the E-utilities service, it shows how a simple URL call can be a powerful tool for retrieving articles from online databases and the potential E-utilities bring to the scientific field.



The E-utilities service allows researchers to perform complex literature searches and retrieve relevant articles based on search queries. This can save time and effort compared to manual searches and enable researchers to identify a wider range of relevant articles. In addition, the E-utilities service supports more advanced search features than the web platform which include features such as field tags and search history retrieval. Moreover, it opens the possibility of performing multiple queries and see how the result set gets influenced by the different parameters.

After the result-set has been determined, it is also greatly beneficial for an extraction tool to have a programmatic way of downloading articles. This is particularly useful for systematic reviews that require large numbers of articles. By having easy accessible article meta-data and full-text content it allows for fast data analysis and text mining, ultimately facilitating the identification of key concepts, patterns and trends in scientific literature. In addition, it is also possible to schedule queries to run at regular intervals to constantly keep the tool updated on the newest scientific discoveries.

### **5.3 Developing a good relevance criteria**

Before an article can be determined as relevant or not, it is important to have a clear understanding of what is being looked for. This includes defining a good research question and identifying the key concepts and terms that are central to the inquiry. After the key concepts have been identified, the search can be performed. As mentioned earlier, PubMed uses a ranking algorithm to determine the relevance of articles to a search query. This algorithm takes into account various factors such as the presence of the search term in the title or abstract, the date of publication, the popularity of the article, and other criteria [7].

The ranking algorithm assigns a score to each article based on these criteria, with higher scores indicating greater relevance. The articles are then sorted by score and displayed

to the user in descending order. However, this ranking algorithm is not perfect and may sometimes include irrelevant articles or miss relevant ones. Therefore, by further analyzing the articles and generating a more sensitive criteria, PMC Explorer could help researchers more accurately find useful articles.

One way of determining relevance is by measuring the prevalence of the input keywords in the article. At first the idea was to find a frequency threshold of a keyword for where an article would be determined as relevant. For example if the keyword is "Alzheimer's disease" and the threshold is 50, only the articles where this keyword occurs more than 50 times would be determined as relevant. However, determining a threshold that fits all articles regardless of length was a difficult task. By having a high threshold, all short articles would be determined as irrelevant even though the articles in reality are relevant. On the other side, by having a low threshold, the algorithm would not correctly identify longer articles.

A solution to this idea would be to find how many times a keyword occurs in the text as a proportion of the length of the text. This would accurately determine how prevalent a keyword is based on the length of the text. By counting the number of times a keyword occurs in the text and dividing it by the total word count resulted in small proportions. For example, for an article consisting of 2000 words, where the frequency of the keyword is 20, result in a relevancy score of 0.01. However, if this keyword was present twice as often in the text, it would result in a relevancy score of 0.02. Determining a good threshold score which was specific enough to handle the variability in article length required scaling to actually reflect the semantic properties of the text.

These attempts laid the foundation of the development of the relevance criteria that is present in PMC Explorer. Instead of relying solely on keyword counting, the algorithm also analyzes the content of the article. By tokenizing the text and keeping track of the most occurring meaningful words, the algorithm captures the essence of the text. If an input keyword is present in the most occurring words of the text, it signals that the text

is relevant to that keyword. However, one weakness of this solution is that the statistical properties of a text does not always reflect the semantic properties of a text.

While some researchers have a wide vocabulary and deep knowledge of different terms used in their field, other researchers keep it simple with few descriptive words. This choice influences which words will be part of the most occurring words in their research paper. For example if a researcher is interested in finding research on medicine for a specific disease it would be intuitive to use "medicine" as keyword. However, depending on the context and which semantic property the word has in a sentence, the researcher can use terms such as "medicament", "medication", "drug" or "cure". Even though these words have similar meaning and may be used interchangeably, it may not trigger the relevance threshold, resulting in a wrong classification.

In addition, in most cases researchers start by introducing a term such as for example "Cerebrospinal fluid", but later only refer to this term by the abbreviation "CSF". This will result in the algorithm determining an article as irrelevant when using the keyword "Cerebrospinal fluid", but relevant when using "CSF" as keyword. A possible solution could be to make the tool aware of these abbreviations and treat them similarly, however sometimes the same abbreviation is used for different words. For example "MS" can in one context stand for "Multiple Sclerosis", but in another context for "mass spectrometry". To avoid these issues the best solution would be to give the researcher the ability to provide keywords along with their abbreviations to make the algorithm aware of what it should regard as the same.

## **5.4 Highlighting data vs. extracting data**

How PMC Explorer performs in terms of identifying data for extraction is dependent on what keywords are used. This means that the tool will perform better if the researcher has a good understanding of the topic in question. In addition, it will be greatly beneficial

if the researcher has previous experience reading articles. A good understanding of the terminology used in research articles results in better performance, since it will help locate the text elements that contain the relevant data. For example by using keywords such as "ANOVA", "t-test" or "Chi-square", a researcher can locate which part of the article presents the statistical tests performed in the study.

The reason why this knowledge is necessary is because there is no algorithm implemented that can automatically identify the relevant data. Most extraction tools have a machine learning algorithm that has the ability to perform part-of-speech tagging and named entity recognition. This gives it the ability to locate what parts of a sentence that bear meaning [25]. When these entities have been located it can find relationships between the different parts of the text and return a summarized version. Therefore, a user can give an article as input and as output they will get an overview of what the algorithm determined to be relevant. Even though this gives fast access to the scientific data, there are several challenges and potential problems that can occur under this process [26].

By creating an extraction algorithm which is able to create relationships between different types of data in an article, it has to have the power to resolve ambiguities and filter out noise. Research articles often use technical jargon, abbreviations, and synonyms, which can make it difficult for algorithms to accurately classify the information. After parsing an article, an algorithm can distinguish between the different parts of the article by following given rules. These rules can be tailored to resolve ambiguities in one topic, but by including too many rules it can ultimately end up impacting performance. While a human can resolve these issues because of the ability to see the bigger context, an algorithm needs to be explicitly told so.

Another challenge is the lack of standardization in reporting. Different authors and journals use different formats and terminology for reporting similar information, which can make it difficult to compare and integrate data from multiple sources. This will make it hard to make a general solution which is compatible with all online articles. Especially

since some parsers read the HTML-content of a page, the parsed result will vary based on how the article or page is structured. If the algorithm is prone to misinterpretations and errors, this will result in hindering the scientific process, as researchers must spend time reconciling the methods and data.

Therefore, rather than attempting to implement an algorithm that could extract data from articles, PMC Explorer makes assumptions based on statistical properties alone. In most cases keyword prevalence means article relevance and a good understanding of terminology simplifies the location of data.

Since there already are many existing tools that provide similar functionalities, the intention was to provide something unique. This tool cannot compete with the machine learning algorithms and natural language processing (NLP) algorithms that many existing tools are built upon. They provide features which can extract and process the data in an automatic way, often returning summaries or visual representations of the data. However, since the process of data extraction is done by complex algorithms, it is not easy to follow the data processing workflow. By making each step simple and traceable the goal is to be able to give a better user experience.

# Chapter 6

## Future work

Since the tool is in its prototype stage there are still features that are yet to be implemented, both when regarding the visual aspects and back-end structure. Improvements will result in new possibilities, more customizability and a better user experience.

### 6.1 Improving the graphical user interface

Currently there are limited visual cues for when something goes wrong or if an article file has missing data which impacts the visual presentation. To make the prototype more user-friendly it could give the user feedback to where it was unable to retrieve the requested data. This can then lead to a feature which lets the user manually fill in the information the parser failed to identify.

To take better use of the highlighting functionality, a method for keyword iteration can give a user the ability to put more emphasis on individual keywords. In most web browsers this is done with "control find (ctrl+f)" which highlights keyword and opens a panel for navigating through the keyword locations.

When regarding the algorithm that performs word frequency tracking, it has currently not been taken fully advantage off. To further improve the use case of these statistics there are some functionalities that can be implemented:

- Give the user the ability to customize what words should be excluded when counting the most frequent words. This will make other words increase in priority,

possibly highlighting new sections of the text.

- The filtering of words by length. By having a minimum and maximum character constraint the user can customize what type of words they want to inspect.
- Sorting the most frequent words by relevance to the input keywords or in alphabetical order.

## 6.2 Additional use of the PubMed API

Since the prototype currently only supports the use of keywords as search criteria, this may limit the variability in search results within a specific topic. To make it possible to explore different results, it will be beneficial to include other search filters such as publishing date or language.

Integrating other E-utilities tools into PMC Explorer opens up for more features which can help with retrieving different types of articles. For example by using ELink a user can get access to related articles in PubMed, based on a specific PubMed ID. This will be useful if the initial search only returned a handful of articles. Then the user could use ELink to retrieve similar articles based on the initial search. Another feature that could be implemented is spelling suggestion. This can be done with the ESpell tool.

## 6.3 Utilizing new technology

Although there are many benefits with using semi-automatic extraction tools, technology is evolving each day leading to improved fully automatic extraction algorithms. Since the availability of high-quality training data has improved, it has allowed machine learning algorithms to be trained on more diverse and representative data sets [27]. As the rules and algorithms improve, so does the extracted data.

Technological advances will be especially beneficial when conducting big systematic

reviews that involve processing of large volumes of data [28]. As well as extracting elements such as images, tables, and figures, these algorithms can perform more complex tasks such as finding relationships between different types of data. These relationships can include gene-disease associations, drug-target interactions, or protein-protein interactions which can help researchers discover new insights and connections between different areas of research [29].

A technological innovation that has gained popularity these last few months is OpenAI's new language model called ChatGPT [30]. Most people use ChatGPT as a communication partner because of its human like responses and vast knowledge. However, ChatGPT can also be used for data extraction. Because of its powerful language processing algorithm it can process text and identify relevant information. To use ChatGPT as an extraction tool it has to be trained on a set of articles and be provided with specific instructions and guidelines on what kind of data it should extract. Queries can be formulated to search for and extract specific information, such as numerical values, experimental results, or key findings.

In addition, many extraction tools can be integrated with easy-to-use analytical software. One popular business analytical tool is Microsoft Power BI [31]. Here researchers can visualize, analyze and share data. These tools provide API's which allow developers to programatically interact with and integrate features from their software. By having tools that are compatible with each other, information can be transferred between platforms to analyze different aspects of the data. This will help researchers save time and effort in managing and processing data, while also improving the the accuracy of the analysis.

In the future we may also see good image recognition software which can recognize data plots and extract data from these plots. This is called optical character recognition (OCR) and has undergone a lot of improvements the last few years [32, 33]. OCR software learns patterns and relationships between features and corresponding labels through complex algorithms. Once trained, the software can distinguish between tables,



plots and other images. After the data in a picture has been identified it can be extracted and further processed.

# Chapter 7

## Conclusion

This master thesis focused on the enhancement of data extraction from scientific publications through the development of a prototype extraction tool. The goal was to facilitate and streamline the extraction of relevant information from the vast amount of scientific literature available in the biomedical domain. Through the exploration of various techniques the extraction tool has been designed to semi-automate and optimize the data extraction process. Currently the prototype can identify key findings, such as experimental results, keywords, and statistical information from scientific publications with improved accuracy and efficiency.

It is worth noting that the development of such an extraction tool is an ongoing process. Continuous improvements and refinements are required to address the challenges posed by the ever-evolving nature of scientific literature, such as new publication formats, emerging terminologies, and linguistic variations. However, this research provides a strong foundation for further advancements in the field of biomedical data extraction.

# Bibliography

- [1] E. Ahn and H. Kang, “Introduction to systematic review and meta-analysis,” *PubMed Central*, 2018. Accessed: May 23, 2023. 2.1
- [2] T. P. Peričić and S. Tanveer, “Why systematic reviews matter,” *Elsevier*, 2019. Accessed: May 25, 2023. 2.1
- [3] B. Sullivan, “Levels of Evidence,” *openmd*, 2022. Accessed: May 23, 2023. 2.1
- [4] Khan et al., “Five steps to conducting a systematic review,” *PubMed Central*, 2003. Accessed: May 29, 2023. 2.1
- [5] G. Tsafnat, “Systematic review automation technologies,” *PubMed Central*, 2014. Accessed: May 23, 2023. 2.1
- [6] Peace Ossom Williamson, “Exploring PubMed as a reliable resource for scholarly communications services,” *PubMed Central*, 2019. Accessed: May 29, 2023. 2.2
- [7] M. Collins, “Updated Algorithm for the PubMed Best Match Sort Order,” *NLM Technical Bulletin*, 2018. Accessed: May 29, 2023. 2.2, 4.3, 5.3
- [8] James Cook University, “Systematic Reviews: What are inclusion and exclusion criteria,” *James Cook University*, 2023. Accessed: May 29, 2023. 2.3
- [9] Fiil-Flynn et al., “Legal reform to enhance global text and data mining research,” *Science*, 2022. Accessed: May 29, 2023. 2.4
- [10] NIH, “MEDLINE: Overview.” [https://www.nlm.nih.gov/medline/medline\\_overview.html](https://www.nlm.nih.gov/medline/medline_overview.html), 2022. Accessed: May 23, 2023. 2.4
- [11] NIH, “PMC Open Access Subset.” <https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/>, 2003. Accessed: May 23, 2023. 2.4

- [12] NIH, “PMC Copyright Notice.” <https://www.ncbi.nlm.nih.gov/pmc/about/copyright/>, 2023. Accessed: May 23, 2023. 2.4
- [13] Moher et al., “Guidance for Developers of Health Research Reporting Guidelines,” *PubMed Central*, 2010. Accessed: May 29, 2023. 2.5
- [14] Tawfik et al., “A step by step guide for conducting a systematic review and meta-analysis with simulation data,” *BioMed Central*, 2019. Accessed: May 29, 2023. 2.6
- [15] Thomas et al., “Applications of text mining within systematic reviews,” *Wiley Online Library*, 2011. Accessed: May 29, 2023. 2.7.1
- [16] Min Jiang et al., “Parsing clinical text: how good are the state-of-the-art parsers?,” *PubMed Central*, 2015. Accessed: May 29, 2023. 2.7.1
- [17] Gulbrandsen et al., “CSF-PR 2.0: An Interactive Literature Guide to Quantitative Cerebrospinal Fluid Mass Spectrometry Data from Neurodegenerative Disorders,” *PubMed Central*, 2017. Accessed: May 23, 2023. 2.8
- [18] Endignoux et al., “Caradoc: a pragmatic approach to PDF parsing and validation,” *HAL open science*, 2016. Accessed: May 29, 2023. 3.1.1
- [19] Jose et al., “Assessing the Impact of OCR Errors in Information Retrieval,” *PubMed Central*, 2020. Accessed: May 29, 2023. 3.1.1
- [20] NIH, “PubMed Central Tagging Guidelines.” <https://www.ncbi.nlm.nih.gov/pmc/pmcdoc/tagging-guidelines/article/style.html>. Accessed: May 29, 2023. 3.1.3
- [21] NIH, “ID Converter.” <https://www.ncbi.nlm.nih.gov/pmc/tools/idconv/>. Accessed: May 23, 2023. 3.2.2
- [22] J. Hedley, “jsoup: Java HTML Parser.” <https://jsoup.org/>, 2009-2023. Accessed: May 23, 2023. 4.1

- [23] Li et al., “The role of plasma cortisol in dementia, epilepsy, and multiple sclerosis: A Mendelian randomization study,” *PubMed Central*, 2023. Accessed: May 23, 2023. 4.4.1
- [24] D. Casacuberta, “Bias in a Feedback Loop: Fuelling Algorithmic Injustice,” *CC-CBLAB Cultural Research and Innovation*, 2018. Accessed: May 23, 2023. 5.1
- [25] Excelsior, “Natural Language Processing- How different NLP Algorithms work,” *Medium*, 2022. Accessed: May 23, 2023. 5.4
- [26] O’Mara-Eves et al., “Using text mining for study identification in systematic reviews: a systematic review of current approaches,” *BioMed Central*, 2015. Accessed: May 29, 2023. 5.4
- [27] Lisa Ehrlinger and Wolfram Wöß, “A Survey of Data Quality Measurement and Monitoring Tools,” *PubMed Central*, 2022. Accessed: May 29, 2023. 6.3
- [28] Shemilt et al., “Pinpointing needles in giant haystacks: use of text mining to reduce impractical screening workload in extremely large scoping reviews,” *Wiley Online Library*, 2013. Accessed: May 29, 2023. 6.3
- [29] B. Bhasuran and J. Natarajan, “Automatic extraction of gene-disease associations from literature using joint ensemble learning,” *PubMed Central*, 2018. Accessed: May 23, 2023. 6.3
- [30] OpenAI, “ChatGPT.” <https://openai.com/research/gpt-4>, 2020. Accessed: May 23, 2023. 6.3
- [31] Microsoft, “Microsoft Power BI.” <https://powerbi.microsoft.com/>, 2015. Accessed: May 23, 2023. 6.3
- [32] G. Shperber, “A gentle introduction to OCR,” *TowardsDataScience*, 2018. Accessed: May 29, 2023. 6.3

- [33] Daehyun Kim and Hong Yu, “Figure Text Extraction in Biomedical Literature,” *PubMed Central*, 2011. Accessed: May 29, 2023. 6.3