



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Development of a text-analytics based framework to support automated clinical literature research and study classification

TESI DI LAUREA MAGISTRALE IN
BIOMEDICAL ENGINEERING – TECHNOLOGIES FOR
ELECTRONICS

Author: **Giorgia Mancini**

Student ID: 941582
Advisor: Prof. Enrico Gianluca Caiani
Co-advisor: Prof.ssa Alessia Paglialonga
Academic Year: 2020-21

Abstract

Nowadays, technological innovation is proceeding at an advanced speed, especially in the medical and biomedical fields. The need to search and aggregate medical information from the Web quickly and consistently is increasingly in demand, and for this reason, many platforms have been developed in recent years to guarantee the possibility of speeding up some essential processes for conducting complete and efficient clinical literature research. The automation of such processes continues to present numerous challenges; in fact, many already developed tools still work independently and cannot be combined to include all the steps necessary for such research. This project aims to fill this gap, to provide a tool to be used to search and aggregate information from scientific articles retrieved from two of the most important databases used in research: PubMed and Google Scholar. The developed framework was designed to support researchers in the collection of clinical evidence regarding a specific topic about the medical field, to carry out experiments, studies, or to prove the validity of a device or an application. This is also in line with the requirements of the new regulation entered into force in May 2021 for medical devices (MDR) that requires manufacturers to prove the validity of a given device during the post-market surveillance through clinical evidence, which can also be found in the literature. This project builds the basis for developing the main steps required to complete structured clinical literature research, which could also be useful for building research studies through Systematic Reviews. All phases were automated and implemented with the *Python* programming language, and the intervention of an external user was required only to launch the script and open the final interface. Through this interface, the user can enter a query string that will be automatically searched in the two chosen search engines, by which all the corresponding scientific articles will be downloaded to form a single final database. Once the database was created, a classification algorithm was implemented and extensively tested to categorize the articles according to the type of study (Systematic Review and Meta-Analysis - SRMA, Randomized Clinical Trial - RCT, or Other), by comparing the titles and abstracts with manually created dictionaries. At the end of this operation, data were presented to the user in an intuitive and aggregated way, to provide an overview and a presentation of the most important information about the obtained results.

Sommario

Ad oggi l'innovazione tecnologica procede a velocità avanzata, soprattutto nell'ambito medico/biomedico. La necessità di cercare e aggregare informazioni mediche dal Web in modo rapido e consistente è sempre più richiesta, e per questo negli ultimi anni sono state sviluppate molte piattaforme che garantiscono la possibilità di velocizzare alcuni processi essenziali per condurre una ricerca della letteratura clinica completa ed efficiente. L'automatizzazione di questi processi, però, continua a presentare numerose sfide; infatti molti tools già sviluppati funzionano solo indipendentemente e non riescono a combinarsi tra loro per includere tutti gli step necessari per la ricerca. Questo progetto mira a colmare questo divario con lo scopo di fornire un tool da utilizzare per collezionare articoli scientifici estratti da due tra i più importanti database usati nella ricerca: PubMed e Google Scholar. Il framework sviluppato è stato pensato per supportare i ricercatori nella raccolta dell'evidenza clinica riguardo uno specifico argomento relativo all'ambito medico per effettuare esperimenti, studi, o provare la validità di uno strumento o di un'applicazione. Tutto questo è in linea con il nuovo regolamento entrato in vigore a Maggio 2021 per i dispositivi medici (MDR) che richiede ai produttori la necessità di dimostrare la validità di un determinato dispositivo durante la Post-Market Surveillance attraverso l'evidenza clinica che si può trovare in letteratura. Questo progetto costruisce le basi per sviluppare le principali fasi necessarie per completare una ricerca clinica strutturata, che potrebbero essere utili anche per costruire studi di ricerca attraverso le Revisioni Sistematiche. Tutte le fasi sono automatizzate e implementate con il linguaggio di programmazione Python, e l'intervento di un utente esterno è richiesto solo per lanciare lo script e aprire l'interfaccia finale. Attraverso tale interfaccia l'utente può inserire una stringa che verrà automaticamente cercata nei due motori di ricerca utilizzati, tramite cui tutti gli articoli corrispondenti verranno scaricati per formare un unico database finale. Una volta creato il database, un algoritmo di classificazione è stato implementato e testato per categorizzare gli articoli in base al tipo di studio (Revisioni Sistematiche e Meta-Analisi – SRMA, Studi Clinici Randomizzati – RCT, o altro), attraverso il confronto dei titoli e degli abstract con dei dizionari creati manualmente. Al termine di questa operazione, i dati vengono presentati all'utente in modo intuitivo e aggregato, per avere una panoramica e una presentazione delle informazioni più importanti riguardo ai risultati ottenuti.

Contents

<i>Abstract</i>	3
<i>Sommario</i>	4
<i>List of Figures</i>	8
<i>List of Tables</i>	11
1. Introduction	13
1.1 Structure of a Systematic Review	15
1.2 Structure of a Randomized Controlled Trial	16
1.3 Databases used in clinical literature research	18
1.3.1 PubMed.....	18
1.3.2 Embase	18
1.3.3 Scopus.....	19
1.3.4 Cochrane Library.....	19
1.3.5 Google Scholar.....	19
1.4 Automation of clinical literature research	19
1.4.1 Machine Learning and Natural Language Processing for automation...	21
1.4.2 Web Scraping for automation.....	25
1.5 Clinical evidence in MDR 2017/745.....	26
1.6 Aim of the work.....	28
2. Materials and Methods	30
2.1 Search in PubMed	32
2.1.1 The Entrez library.....	32
2.1.2 The Entrez Programming Utilities.....	33

2.1.3	String creation for PubMed.....	37
2.1.4	Text formats.....	38
2.1.5	Fields Extraction.....	40
2.1.6	DataFrame Creation.....	44
2.2	Search in Google Scholar	49
2.2.1	String creation for Google Scholar.....	49
2.2.2	URL creation.....	50
2.2.3	Scraping Information	51
2.3	Total Database.....	68
2.4	Classification	70
2.4.1	Creation of dictionaries.....	71
2.4.2	Grouping words with Levenshtein distance.....	71
2.4.3	Creation of Regular Expressions	74
2.4.4	Score computation and classification	74
2.5	Development of the Web Interface.....	79
3.	Results.....	81
3.1	Database creation.....	81
3.2	Classification	87
3.2.1	Training phase.....	87
3.2.2	Validation phase.....	95
3.2.3	Test phase	98
3.3	Web Interface.....	99
3.3.1	Tab1: Web Scraper Tool	99
3.3.2	Tab2: Visualization of Results.....	101
4.	Discussion and Conclusion.....	107
4.1	Database creation.....	107
4.2	Classification algorithm	109
4.3	Web Interface.....	112
4.4	Limitations.....	114
4.5	Future Developments	115

4.6 Conclusion.....	115
<i>Appendix A</i>	117
<i>List of Abbreviations</i>	120
<i>Bibliography</i>	122
<i>Sitography</i>	125

List of Figures

Figure 1-1 - Pyramid of evidence. Source: [I].....	14
Figure 1-2 - CONSORT diagram.....	17
Figure 1-3 - Phases of bag of words process. Source: [18].....	22
Figure 1-4 - CNN process to represent texts. Source: [19].....	23
Figure 1-5 - Phases of Bulla et al. AI-assisted framework [22].....	24
Figure 1-6 - Summary of Web Scraping phases. Source: [VII].....	26
Figure 2-1 - General workflow of the thesis.....	32
Figure 2-2 - NCBI search bar.....	33
Figure 2-3 - Example of PubMed record in XML format	39
Figure 2-4 - Example of PubMed record in Medline text format.....	40
Figure 2-5 - Example of lines in a log file	45
Figure 2-6 - Description of database filename	46
Figure 2-7 - Workflow for PubMed database creation.....	48
Figure 2-8 - Example of URL	50
Figure 2-9 - Google Scholar initial page	52
Figure 2-10 - Example of HTML page structure.....	53
Figure 2-11 - On the top (2-11a) the organization of a record in the Google Scholar starting page. On the bottom (2-11b) the corresponding HTML content.....	54
Figure 2-12 - Static Page protocol. Source: [XVI].....	55

Figure 2-13 - Dynamic Page protocol. Source: [XVI].....	55
Figure 2-14 - Block diagram for DOI extraction.....	57
Figure 2-15 - Example of article published in sciencedirect.com: [XVIII].....	59
Figure 2-16 - Example of title modified with CrossRef	62
Figure 2-17 - Example of abstract in CrossRef XML document	62
Figure 2-18 - Example of BibTex format.....	63
Figure 2-19 - Blocks diagram for BibTex information retrieval	65
Figure 2-20 - Blocks diagram for Google Scholar database creation.....	67
Figure 2-21 - Blocks diagram to explain the classification process	76
Figure 2-22 - Confusion Matrix for Binary Classification. Source: [XXVI]	77
Figure 2-23 - Example of ROC curve. Source: [XXVII].....	78
Figure 3-1 - Cases for PubMed database creation.....	82
Figure 3-2 - Cases for Google Scholar database creation.....	82
Figure 3-3 - Example of record in JSON format.....	84
Figure 3-4 - Number of SRMA with Regex match in title (left), match in title and abstract (center), match in abstract (right), no match (bottom).....	88
Figure 3-5 - Number of RCT with Regex match in title (left), match in title and abstract (center), match in abstract (right), no match (bottom).....	88
Figure 3-6 - ROC Curve for Meta-Analysis vs Others.....	89
Figure 3-7 - Confusion Matrices of SRMA vs Others (threshold 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0).....	90
Figure 3-8 - ROC Curve for RCT vs Others	91
Figure 3-9 - Confusion Matrices of RCT vs Others (threshold 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0).....	93
Figure 3-10 - ROC curve for RCT vs Others.....	94
Figure 3-11 - Confusion Matrices of RCT vs Others (thresholds from 0.0 to 30.0)....	95
Figure 3-12 - Confusion Matrix of SRMA vs Others using threshold 30.0	96

Figure 3-13 - Confusion Matrix of RCTs vs Others using threshold 15.0.....	96
Figure 3-14 - Confusion Matrix of SRMA vs Others using threshold 30.0.....	97
Figure 3-15 - Confusion Matrices of RCT vs Others with thresholds 15.0 and 30.0..	97
Figure 3-16 - Confusion Matrices during Test. Figure 3-16a: database “pacemaker”.	
Figure 3-16b: database “artificial pancreas”. Figure 3-16c: database “telemedicine”.	
.....	99
Figure 3-17 - Organization of first Tab, with a zoom on the 'From' dropdown menu	
.....	100
Figure 3-18 - Example of results shown at the bottom of the page.....	101
Figure 3-19 - Dropdown menu of Tab2 clicked	101
Figure 3-20 - List of old databases after clicking the dropdown menu	102
Figure 3-21 - Section ‘Database Overview’ of Tab2: information taken from database	
created with the string "atrial fibrillation"	102
Figure 3-22 - Section ‘Information on Journals’ of Tab2: information taken from	
database created with the string "atrial fibrillation"	103
Figure 3-23 - Section ‘Information on Years’ of Tab2 with the range slider selected:	
information taken from database created with the string "atrial fibrillation"	104
Figure 3-24 - Section ‘Information on Years’ of Tab2: information taken from	
database created with the string "atrial fibrillation"	104
Figure 3-25 - Section ‘DataTable’ of Tab2: information taken from database created	
with the string "atrial fibrillation"	105
Figure 3-26 - Input boxes and buttons at the bottom of the DataTable	106
Figure 3-27 - Example of structured data after clicking the 'Secodnary Source' cell	106

List of Tables

Table 1-1 - List of some of the existing tools for automating SR steps	21
Table 2-1 - List of Python Libraries with corresponding purpose	31
Table 2-2 - Options for 'sort' parameter of ESearch function	35
Table 2-3 - Options for 'datatype' parameter of ESearch function	36
Table 2-4 - List of fields extracted from PubMed.....	41
Table 2-5 - Examples of dates extracted from PubMed with respective conversion	43
Table 2-6 - Examples of Secondary Source information with corresponding Register	44
Table 2-7 - Examples of medical website and corresponding tag for DOI.....	58
Table 2-8 - Examples of medical website and corresponding tag for Abstract	59
Table 2-9 - Example of duplicate record with PubMed and Google Scholar info	69
Table 3-1 - Partial "telemedicine" database	85
Table 3-2 - Partial "mobile health" database.....	86
Table 3-3 - Values of Sensitivity and FPR for Meta-Analysis (thresholds 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0).....	89
Table 3-4 - Values of Sensitivity and FPR for RCT (thresholds 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0).....	91
Table 3-5 - Values of Accuracy, Precision, Number of correct classified RCT with respect to PubMed (thresholds 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0)	94

Table 3-6 - Values of Accuracy, Precision, Number of correct classified RCT with respect to PubMed (threshold from 0.0 to 30.0).....	95
Table 3-7 - Values of Sensitivity, FPR, Accuracy, Precision for thresholds 15.0 and 30.0.....	98
Table 5-1 - RCT dictionary and Meta-Analysis dictionary.....	117
Table 5-2 - Regular Expressions for RCT dictionary	118
Table 5-3 - Regular Expressions for Meta-Analysis dictionary.....	119

1. Introduction

Scientific publications are the main form of communication through which researchers share information about their studies. They are usually texts that describe in detail the methodologies used to prove a scientific discovery and its results. Scientific articles are usually published in specific journals or digitally reported on some online resources.

One of the main aspects that characterize a publication is objectivity: the used procedures and the results obtained during the research must always be clearly explained and reported transparently, in order to make the study reproducible.

Currently, the scientific literature has a huge volume of articles, which are published with the aim of exchange knowledge between researchers and collecting many studies in order to improve clinical evidence.

Clinical evidence represents the best available proof to demonstrate something related to clinical problems [1]. The word 'evidence' is used when it is necessary to confirm and prove a hypothesis, but also to refuse it. The objective is always the improvement of health and well-being by answering specific questions in order to quantify the associated potential risks and benefits.

Whenever clinical evidence is used, it is important to clarify the context of use and why such evidence could reinforce research [2]. Evidence-based practice can be useful to make decisions and to prove the effectiveness of a thesis, but also to improve the characteristics of discovery in the medical field. Even if such practice is always present in the world of healthcare, sometimes it is not enough to make the research perfect. Studies are often affected by external factors that can influence the results [3]. This is the reason why there are many types of studies and experimental designs that can be used to find the best evidence about a specific problem and answer specific questions.

Figure 1-1 shows a pyramid representing a hierarchy created to explain the level of evidence provided by different types of studies. Such studies are sorted from the base to the top, starting from the ones with the lowest level of evidence [4].

On the top, there are Systematic Reviews and Meta-Analyses (SRMA), followed by Randomized Controlled Trials (RCTs), Cohort Studies, Case-control Studies, and Cross-sectional Studies, and at the base, there are basic research and expert opinions.

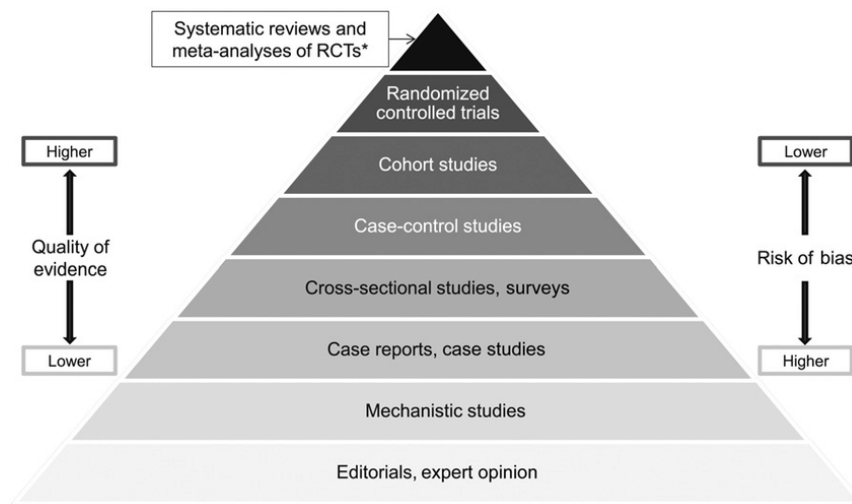


Figure 1-1 - Pyramid of evidence. Source: [1]

Systematic Reviews (SRs) are considered the ones that guarantee the highest level of evidence because they involve systematic research of literature, conducted under certain rules, with the aim to collect all the studies relevant to a specific argument. They are also called “secondary research” because they use the existing research to conduct another study. Their objective is to summarize the results of different trials, in order to provide information for clinical practice and to help the decision-making process [5]. All the studies are selected following rigorous criteria, and according to their heterogeneity, it is possible to use the Meta-Analysis or not. This is a statistical analysis that provides quantitative results merging different trials answering the same clinical question. It aims to integrate the findings of a large number of primary studies, trying to demonstrate the generalizability of the results.

RCTs are below the SRs in terms of the level of evidence. They are considered the studies that best demonstrate the efficacy of an intervention because they compare one or more treatment to a control group, where allocation of subjects is purely random, to find which

one brings more benefits to patients. RCTs are characterized by both internal validity, which aims to avoid bias during the study, and external validity, so to potentially guarantee the study generalizability and applicability in other contexts.

1.1 Structure of a Systematic Review

Given the great variety of studies and publications, when clinical research is implemented, it is important to improve the quality of such research. For this reason, some guidelines are constructed to provide standardized formats to report results in a structured way and to avoid the publication of useless information.

The most important steps necessary to conduct a SR, are summarized below [6]:

- Define the question: it represents the first phase in which the objective of the review is clarified. Reviews should answer well-formulated questions that will guide the development of the literature search.
- Reviewing the literature: it is the main part of the work, where all the most important databases, clinical trials registers, and literature in general are searched to collect the studies relevant to the question previously defined. Usually, a search strategy is established to identify the best results that match the topic of the review.
- Select relevant studies: this is one of the most time-consuming steps because researchers must screen all the results of the published clinical research to find the items that match inclusion and exclusion criteria established a priori. Information, such as the number of participants, the types of treatments and comparators, the type of outcomes, should be extracted and compared with the criteria.
- Assess the quality of studies: this step aims to verify if trials present bias, so to evaluate potential errors that could occur because of an under-estimation or over-estimation of the effect of an intervention.
- Calculate the outcome measures: studies can use binary outcomes or continuous outcomes and they should be as homogeneous as possible in terms of reporting results, to use the Meta-Analysis and summarize findings. Meta-Analysis contributes to

improving the precision of the research and answering questions not posed by individual studies [III]. Alternative methods are available when a statistical analysis is not applicable: they provide limited results, but still better than a narrative synthesis that describes the studies only qualitatively.

- Interpret the results: in this phase, results are visualized with different plots. Forest Plot is used when a Meta-Analysis is performed, and other types of charts are utilized if a statistical analysis is not conducted.

1.2 Structure of a Randomized Controlled Trial

Although RCTs are considered the gold standard for evaluating the effect of an intervention, they may not be reported properly. RCTs should respect a standardized format for reporting information on trials as recommended by the CONSORT¹ Statement, which provides a checklist of fundamental information that should be included in a trial.

The items are [III]:

- Title and Abstract: the title should include that the study is a randomized trial, and the abstract should be structured, with some paragraphs that clearly explain the context, the study design, the methodologies, the results, and the conclusions. Abstracts must be transparent enough to facilitate the information retrieval and evaluation of the results.
- Introduction: usually, the introduction includes the work background and the general trial scheme. Sometimes, also the objective of the trial is included, to provide a clear explanation of the reason for which the study is conducted.
- Methods: this section includes all the characteristics of the study design. The description of the groups with the corresponding treatment, the eligibility criteria for choosing patients, the type of randomization and blinding, are all aspects that should be reported in detail, to make the study reproducible in a different context. The type of outcomes, both primary and secondary, should be inserted too, together with the time of the follow-

¹ <http://www.consort-statement.org/>

up for patients, and the way of computation of such outcomes. The sample size is another important piece of information that must be included, with the method used to estimate it. Figure 1-2 shows the CONSORT diagram, in which the enrollment of patients is described, starting from the eligibility criteria and the subjects' allocations, up to the information regarding the follow-up and the analysis in the study.

- Results: this section explains all the results obtained during the study, reported for each outcome and group.
- Discussion: all the considerations about the study and the obtained results are reported in this part, with the trial limitations, the eventual bias, and the description of the study generalizability.
- Other information: this last section usually contains the Trial Registration number and other information for readers.

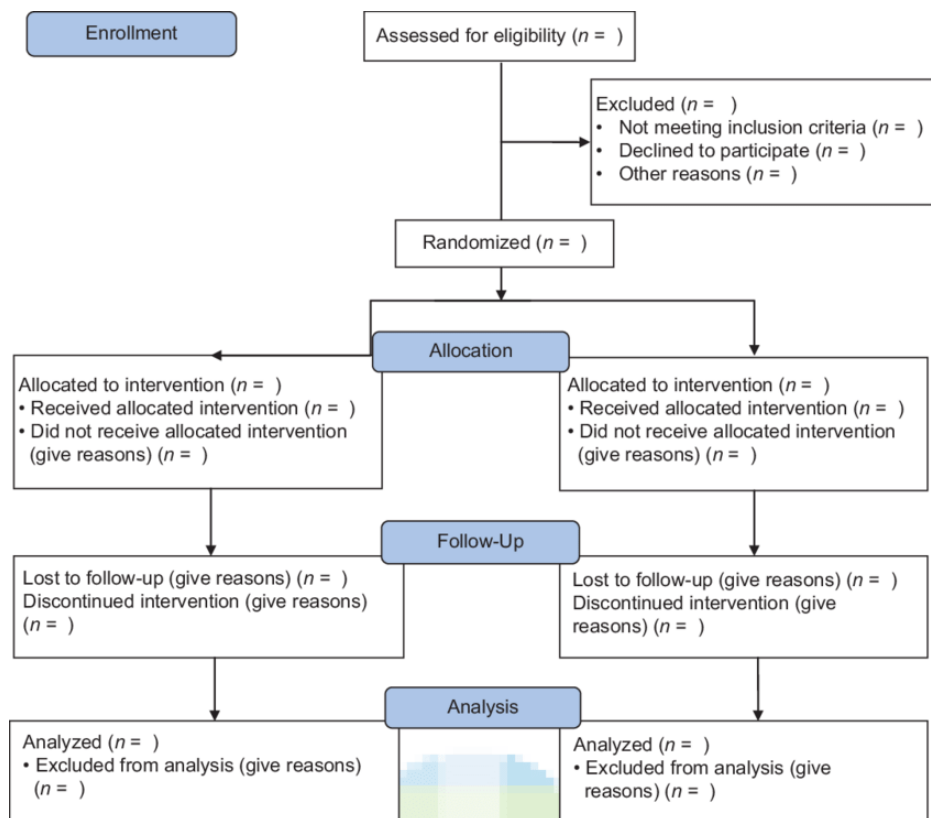


Figure 1-2 - CONSORT diagram

1.3 Databases used in clinical literature research

The searching process is fundamental for performing clinical literature research. With the technology development, this process is becoming more challenging, due to the volume of available data and the variety of sources in which it is possible to find relevant information [7]. Online databases are very important because they contain lots of papers and give the possibility to implement a particular search strategy. The most used databases are reported and briefly described below.

1.3.1 PubMed

PubMed² is an online available resource containing more than 30 million papers regarding biomedical literature. It is a search engine that includes different types of studies such as Meta-Analyses, Clinical Trials, Journal Articles, Reviews, and others. It allows researchers to find evidence about healthcare problems and limit the research thanks to the use of different filters.

Citations belonging to PubMed are published in MEDLINE³, PubMed Central and Bookshelf. They are bibliographic archives that can be screened to retrieve information, and MEDLINE represents the main component because it contains more than 26 million records.

1.3.2 Embase

Embase⁴ is a bibliographic database produced by Elsevier and created in 1946. It contains biomedical and pharmacological information and comprises more than 30 million records. It includes all of MEDLINE citations, but there are over 7 million records that cannot be accessed through PubMed [IV].

² <https://pubmed.ncbi.nlm.nih.gov/about/>

³ https://www.nlm.nih.gov/medline/medline_overview.html

⁴ <https://www.elsevier.com/solutions/embase-biomedical-research>

1.3.3 Scopus

Scopus⁵ is a database created in 2004 that covers literature from any discipline. It contains articles from more than 5000 international editors, but also other types of records like patents or books.

1.3.4 Cochrane Library

The Cochrane Library⁶ is the principal database for SRs that analyze topics related to healthcare. Each review is scrutinized by a Cochrane Review Group to report information correctly and make the study useful for the decision-making process.

1.3.5 Google Scholar

Google Scholar⁷ is the Google search engine for scientific literature. It contains citations like articles, documents, books, conference papers, and it is freely accessible. Records published in this resource can be present in many other databases, and even if the results provided by Google Scholar are limited with respect to others coming from different archives, it represents a way to quickly start research, and to find information in different fields.

1.4 Automation of clinical literature research

Clinical literature research is essential to conduct studies and experiments, especially SRs, but it is a very long process. For this reason, lots of techniques are implemented to reduce the time spent for searching and extracting data from papers. Researchers tried to automate

⁵ <https://www.scopus.com/search/form.uri?display=basic#basic>

⁶ <https://www.cochranelibrary.com/cdsr/about-cdsr>

⁷ <https://scholar.google.it/>

tasks in which they are expert, but to create something useful for lots of people, all the automated tools should work together. To accomplish this objective, the International Collaboration for the Automation of Systematic Reviews (ICASR) was born, including people with different background, such as engineers, researchers, linguists, with the aim to cover all the necessary phases to conduct research and construct a SR [8].

Since the process can require years, automation was introduced to improve all the steps that a SR involves, both in term of time and quality of results. Different tools exist with the aim to automate tasks and improve output, even if most of them are still in development phase. For this reason, one of the principles established by ICASR is to share the code of every tool, and make it open source, with the possibility to have a clear explanation of the tool functionality, but also to improve and contribute to the development process.

Table 1-1 reports some of the existing tools able to automate the necessary phases to conduct a SR.

Tool	Purpose	Functionalities
Metta	Search and retrieval of records	It is an interface created to simplify the process of submitting queries in different databases and optimize the retrieval of relevant papers. It allows four search tracks: general search, SRs search, case reports search, human-related studies search [9].
Thalia⁸	Search literature	It is a semantic search engine for biomedical literature. It can recognize all citations containing specific concepts (chemicals, diseases, drugs, genes, metabolites, proteins, species, and anatomical entities) without taking into consideration the syntax used for explaining them. It is updated daily, and it provides a user interface for searching [10].
Rayyan⁹	Screening citations	It is a free web and mobile app that uses a semi-automated process to screen titles and abstracts. Users can label citations and explain the reasons for

⁸ http://www.nactem.ac.uk/Thalia_BI/

⁹ <http://rayyan.qcri.org>

		exclusion of a paper. The app can also extract metadata from a record and establish a similarity score with other items. The app was tested and evaluated from experts with different levels of competencies [11].
Cochrane Crowd ¹⁰	Categorize papers	It is a global community founded for helping researchers in the classification of papers. The objective of the collaboration is to improve the quality of evidence about healthcare problems.
RevManHAL	Automatic text generation	It is a software constructed as a text editor that produces structured files from unstructured data. Users can insert information in particular templates, and then abstracts, results, and discussion are auto generated [12].
SR Toolbox ¹¹	Different tasks	It is an online archive of automated tools able to perform tasks for completing SRs. It allows users to conduct a “quick search” or an “advance search” to identify the desired tool according to the purpose for which it is created [13].

Table 1-1 - List of some of the existing tools for automating SR steps

1.4.1 Machine Learning and Natural Language Processing for automation

The development of technology and informatics in general has allowed the automation of some phases necessary to conduct research in this field. In particular, Machine Learning (ML) and Natural Language Processing (NLP) are used to develop frameworks capable of performing the actions necessary for clinical literature research, such as text classification and information extraction [14].

¹⁰ <https://crowd.cochrane.org/>

¹¹ <http://systematicreviewtools.com/>

Text classification is used to group documents into a predefined category of interest by screening the text inside and assigning a score representing the probability of falling into a particular class. These tasks can be achieved using ML techniques, which usually represent words as vectors, so sets of letters mapped into numbers with specific rules. There are different methods to make this conversion, and one of them is the *bag of words*: sets of documents are represented as a matrix in which the number of rows corresponds to the number of documents and the columns correspond to unique words. The matrix will be populated with 0s and 1s according to the presence or absence of such words in the document. Then some weights are estimated based on the probability that a word may refer or not to RCT common lexicon (if the objective is to discriminate between RCTs and non-RCTs). These coefficients are multiplied by the vectors, and a probability is estimated and analyzed to understand if the document falls into a class or another. The process is summarized in Figure 1-3.

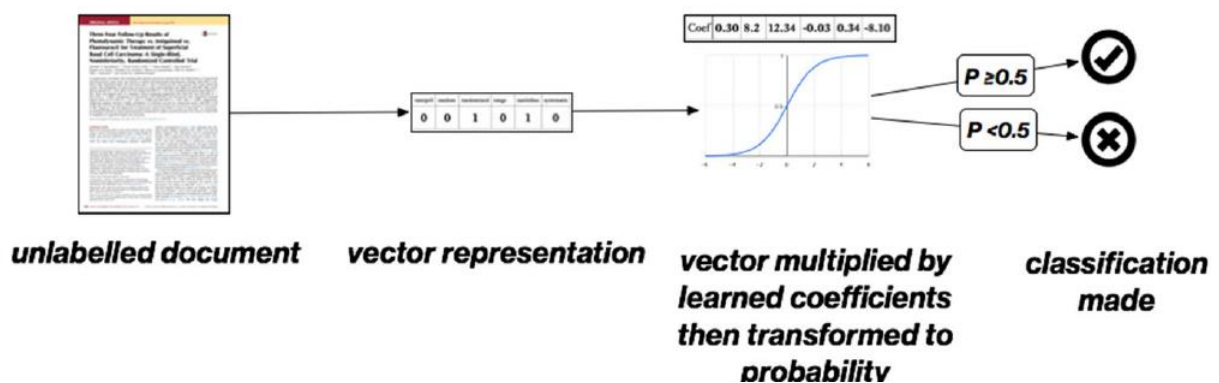


Figure 1-3 - Phases of bag of words process. Source: [18]

Data extraction can be considered similar to text classification because single words are evaluated and considered relevant or not, for extracting particular information. In this case, not only individual words are analyzed, but also the ones that precede and follow the desired information. Typical data extraction processes assign 1 to the word of interest, and 0 to the other contextual information. Sometimes these methods do not produce great results, because there is the risk of not considering adjacent information that can be useful.

Marshall et al. [15] used ML techniques for identifying RCTs in health databases, in particular Support Vector Machine (SVM) and Convolutional Neural Network (CNN). The objective was to demonstrate the efficacy of these methods concerning common filters

available in online search engines. SVM is an algorithm that tries to separate items belonging to two categories inside a plane. The items are mapped as points in space and the objective is to maximize the gap between the two classes. In this case, the two groups were RCT and non-RCTs, which were predicted to belong to a category according to the side in which they fell in the plane. CNN, instead, uses vectors to represent words and matrices to represent pieces of texts. Some filters are then applied to generate a single vector that will be classified in RCT or not-RCT (Figure 1-4).

All the titles and abstracts were tokenized, and the stop words were removed before applying the algorithms. Several combinations of classifiers were used: SVM and CNN were used independently and in combination with the Publication Type (PT) tag of PubMed (PubMed staff manually applies this tag to papers). The results were compared using the Area Under the ROC curve, and the best result was achieved using the combination of SVM, CNN, and PT tag (AUROC = 0.987).

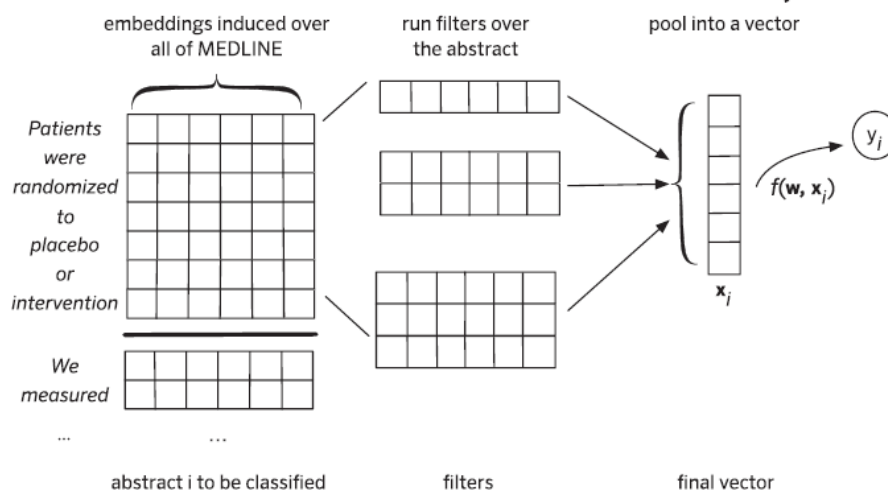


Figure 1-4 - CNN process to represent texts. Source: [19]

The system developed by Thomas et al. [16] includes database searching, ML for identifying papers, and crowd searching, and the objective was to identify the maximum number of RCTs to populate an existing database (called CENTRAL¹²) and reduce the manual workload for excluding non-RCTs. The study was three-stage with training, calibration, and validation phases to evaluate two different classifiers that use SVM. The first one represented

¹² <https://www.cochraneflibrary.com/central/about-central>

text as uni-grams, bi-grams, or tri-grams (so individual words, pairs of words, or triplets of words), while the second one used only uni-grams. Precision and recall were used as evaluation metrics, and it was demonstrated that ML approaches allow obtaining higher precision than common search filters, but some papers are not adequate to be considered as input in ML classifiers because they contain too limited information in the title or abstract and have to be screened manually.

ExaCT is an automatic extraction system integrated with a web browser interface, created for retrieving information from RCTs [17]. It was created to support reviewers by selecting data in the full paper text, and not only in the abstract or title, to not omit information useful for clinical research. This choice was done because usually abstracts describe an overview of the study, and not the specific trial characteristics, like eligibility criteria or types of outcomes. The implemented approach firstly uses a text classifier to select pieces of text that may contain relevant information, and then Regular Expressions (Regex) are used to extract only the desired datum. Records manually selected must be uploaded in the interface in an HTML format, in order to identify sections with specific headings like 'Methods' or 'Results'. Sentences are then represented using the *bag of words* method and then classified using the SVM. At the end, there is the possibility for users to review or modify the extracted data before exporting and using them.

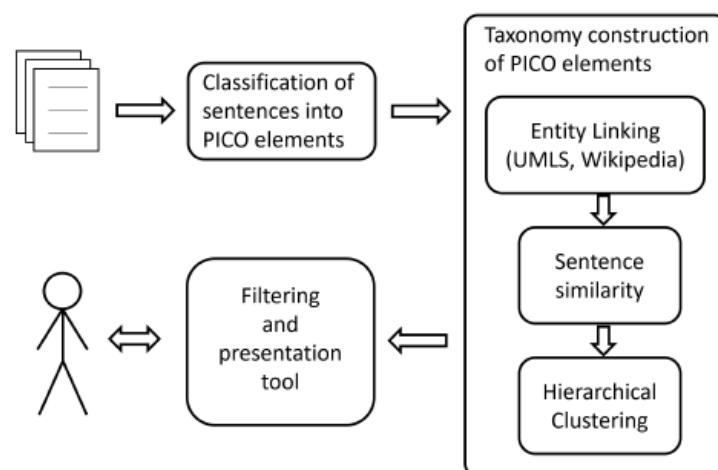


Figure 1-5 - Phases of Bulla et al. AI-assisted framework [22]

Bulla et al. developed an AI-assisted framework with the aim to speed-up the process of performing a SR [18]. They used NLP techniques to extract PICO (Population, Intervention, Comparison, Outcome) information from abstracts and the process is shown in Figure 1-5.

Abstracts texts were analyzed to identify relevant sentences for PICO elements. The sentence classification was performed in three phases: every sentence was converted in a tokens list, then each vector was contextualized with information from other sentences, and at the end the probability of belonging to a particular label was calculated. A taxonomy was implemented for each abstract to connect clinical terms to existing ontologies and vocabularies (like UMLS or SNOMED CT). Then, sentence similarity between abstracts was computed to perform a hierarchical clustering. The filtering tool was inserted as last step to allow users to select one or more interesting concepts to filter the results and obtain only relevant articles with such concepts.

1.4.2 Web Scraping for automation

Web scraping is the automated process of extracting information from the Internet and exporting data to insert them in a file [V]. It consists of a series of techniques used for different purposes, like clinical literature research, news research, or market research, and it aims to reduce the time normally spent for collecting information from the websites.

A web scraping tool is a software that makes HTTP requests to a website and extracts data from it. Usually, only accessible content is extracted, and information is taken from HTML web pages. Since the webpages structures are different, there are many techniques that can be used for parsing pages [VI]. The process of Web Scraping comprises the following phases, summarized in Figure 1-6:

- Identify the desired website
- Collect URLs of the pages where to find information
- Make a request to such URLs to obtain the HTML pages
- Use locators to find the data
- Extract data and save them in the preferred format

In this work, Web Scraping will be used for extracting specific information from the Internet and the phases cited before will be described in detail in Chapter 2.

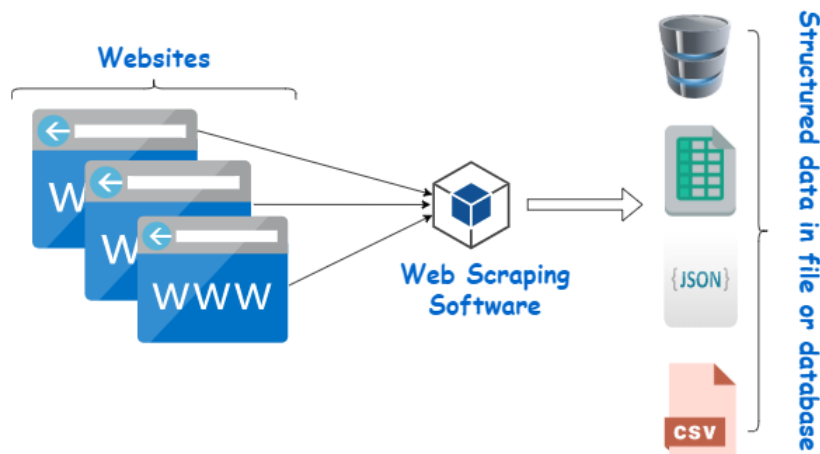


Figure 1-6 - Summary of Web Scraping phases. Source: [VII]

1.5 Clinical evidence in MDR 2017/745

The Medical Device Regulation (MDR) is the regulation created to provide rules for certifying the performance of medical devices, before their launch on the market, and after that, to verify the insurgence of eventual adverse events during post-market surveillance. The new MDR is active since May 26th, 2021, and it has introduced additional rules to the old MDD (Medical Device Directive) for controlling the safety of medical devices. The key aspects of the new MDR are [VIII]:

- Introduction of new stringent classification rules (from 18 in the MDD to 22 in the MDR) with some changes in devices risk classes (I, IIA, IIB, or III).
- Introduction of economic operators (Manufacturer, Authorized Representative, Importer, and Distributor).
- Introduction for Manufacturers of financial coverage, risk management system, post-market surveillance system, reporting incidents system.

- The necessity of demonstrating compliance with clinical data and the necessity to produce technical documentation. Specific documents need to be produced by the Manufacturer: Safety and Clinical Performance Summary for Class III Devices and Implantable Devices, Post-market surveillance report for Class I Devices, and Periodic safety update report for Class IIA, IIB, and III Devices.
- Introduction of device traceability with the creation of the UDI (Unique Device Identification) system.
- The collection of Device information in a single European database (EUDAMED).
- Simplification of conformity assessment procedures based on product quality assurance and statistical product verification.

One of the most important changes of the MDR is related to the clinical evaluation of devices. Article 61 of the new MDR states: *“Confirmation of conformity, evaluation of eventual side-effects, acceptability of the benefit-risk-ratio, should be based on clinical data providing sufficient clinical evidence. The manufacturer shall specify and justify the level of clinical evidence necessary to demonstrate conformity with the relevant general safety and performance requirements. That level of clinical evidence shall be appropriate in view of the characteristics of the device and its intended purpose.”* This means that scientific literature must be reviewed by manufacturers to establish the required level of clinical evidence, together with its safety and performance data, according to the intended purpose of the device, and the results must be included in a Clinical Evaluation Report (CER), that is updated during the entire lifecycle of the device (annually for Class III devices, as explained in MDR, Annex XIV).

The CER is a document that collects information about a specific medical device, by analyzing clinical data collected from scientific literature [IX]. With the new MDR, more stringent requirements are necessary for CERs: it is mandatory to conduct a Clinical Evaluation Plan (CEP) and a Post-Market Clinical Follow-up (PMCF), as explained in MDR, Annex XIV. The CEP should be conducted by the manufacturers with the following information: performance and intended use of the device, clinical benefits for patients, target groups, and all the information related to the device risks. It must be objective and should take into consideration both favorable and unfavorable data [X].

The PMCF is a continuous updating of collecting data from a device already present in the market, with the aim to constantly control the safety of such device and detect eventual adverse events that can occur on patients. The PMCF plan should include information about

the data collection (from scientific literature, feedback from users, or other sources) and evaluation.

The new rules introduced by the MDR demonstrate how the importance of clinical research and literature is growing over the years, not only to provide clinical evidence on experiments in general, but also in the specific field of the medical devices, currently used and distributed all over the world. Each device must adhere to rules to minimize risks for patients, and manufacturers have to constantly conduct clinical research before launching a device on the market (thanks to the analysis of equivalent devices) and after, to monitor its use and ensure benefits to patients who use it.

1.6 Aim of the work

This work aims contributing to the automation of clinical literature research by the extraction of evidence from published studies, starting from querying articles, titles and abstracts screening, up to the classification of papers and graphic visualization of the results.

All these phases are fundamental, and this thesis aims to make them automatic to create a framework that can connect them and intuitively show the results. For this reason, the specific aim of this project is to develop an interface capable of:

- Create a database collecting all literature relevant to the clinical studies related to a certain topic of interest: it is possible to choose a search string that will be applied to different literature search engines on the Internet
- Filter the collected data for the elimination of duplicates
- Automatically classify the resulting documents according to the type of study: this process represents the fundamental step in literature search and is the most expensive in terms of time and resources
- Filter the database based on the type of study and then eventually save the filtered data
- Represent the results in a user-friendly way

All these features will be automated so that the user will be required only to open the interface and navigate it. In Chapter 2 all the methods used to create the entire framework

will be explained; in Chapter 3 the results of the analyses carried out will be shown, and in Chapter 4 a discussion of the entire work, with the limitations and possible future developments, will be given, together with final conclusions.

2. Materials and Methods

This thesis integrates the use of Web Scraping to retrieve information from the Internet, and Web development to show the results to the final user. This chapter describes all the steps followed to create the entire framework. All the phases will be illustrated in detail, citing the technologies and the libraries used.

All workflow steps were implemented using *Python*¹³ (version 3.8), one of the most powerful and used high-level programming languages. Python is flexible and easy to use: it can be utilized by different people, from beginners to experts, to develop projects of any size and type, starting from the implementation of web apps or algorithms with Machine Learning up to the creation of more common software like video games [XI]. Python was chosen thanks to its versatility and adaptability to solve different tasks. It also provides lots of libraries useful to reach the objective of this work: Table 2-1 shows the libraries utilized in this work with the respective purpose.

¹³ [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

Library	Purpose
<i>Bio.Entrez</i>	Compute the research on Pubmed
<i>Bio.Medline</i>	Work on Medline format from NCBI
<i>BeautifulSoup</i>	Static Web Scraping
<i>Selenium</i>	Dynamic Web Scraping
<i>matplotlib.pyplot</i>	Plot graphs
<i>Plotly</i>	Create graphs objects
<i>Dash</i>	Develop web interface

Table 2-1 - List of Python Libraries with corresponding purpose

Figure 2-1 presents a general overview of the most important steps that compose the proposed framework.

In the following sections, each block will be described in detail.

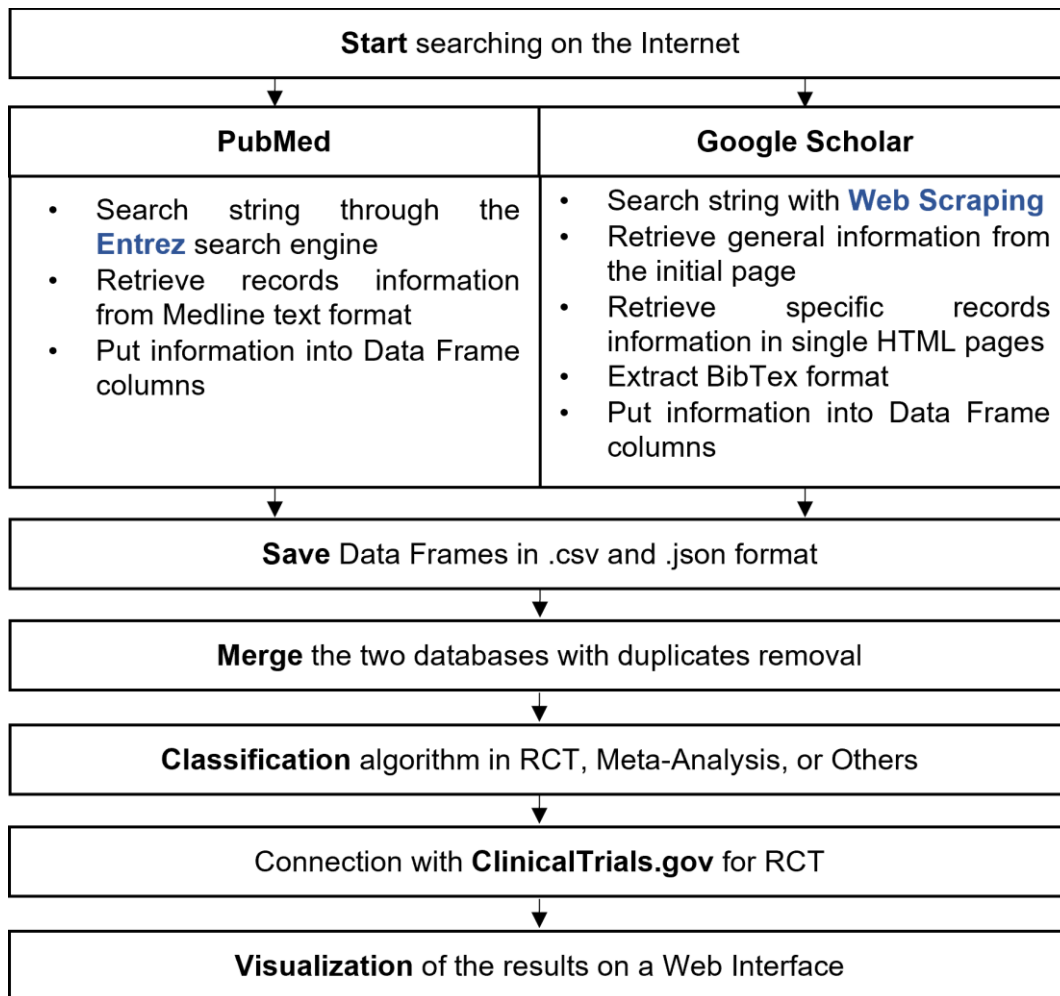


Figure 2-1 - General workflow of the thesis

2.1 Search in PubMed

2.1.1 The Entrez library

The automatic research implemented through the PubMed website was performed thanks to a powerful search engine called Entrez.

Entrez is provided by NCBI (*National Center of Biotechnology Information*) [19] and it is based on a simple interface used to search information within the biomedical literature, as shown in Figure 2-2.

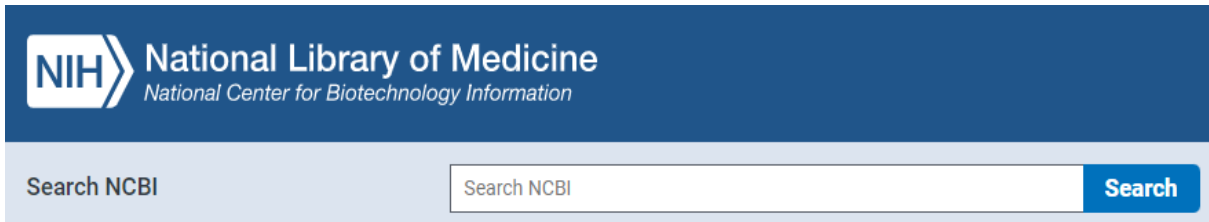


Figure 2-2 - NCBI search bar

Entrez allows the retrieval of information from 38 databases, with a query that can be customized for each of them by different fields, filters, and text formats. They are *Assembly, BioProject, BioSample, BioSystems, Bookshelf, ClinVar, Conserved, Domains, dbGaP, dbVAR, EST, Gene, Genome, GEO Datasets, GEO Profiles, GSS, GTR, HomoloGene, MedGen, MeSH, NCBI Web Site Search, NLM Catalog, Nucleotide, OMIM, PopSet, Probe, Protein, Protein Clusters, PubChem BioAssay, PubChem Compound, PubChem Substance, PubMed, PubMed Central, SNP, SRA, Structure, Taxonomy, UniGene, UniSTS*.

The final interface created in this work has been set to search only in PubMed and PubMed Central, without considering the other databases. However, if a user would like to search in a database different from PubMed, it will be possible to make this change immediately through code.

2.1.2 The Entrez Programming Utilities

The Entrez search engine provides the Entrez Programming Utilities, also called E-Utilities. Developers can use them to connect with the primary interface (both query system and database system) and retrieve different information to organize them directly through programming [20].

The E-Utilities require various parameters to personalize the research: some are needed to run the code, while others are optional. Different programming languages can use E-Utilities to organize data and convert these parameters directly

into NCBI software components to obtain the expected result [XII]. The URL used to send requests to the NCBI website starts with the following prefix:

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

The mandatory parameters, necessary every time a user wants to search for something, are:

- *tool*: string to identify the software which produces the request
- *email*: valid email address of the user, used only in case of NCBI policy violation or to announce software updates.
- *api_key*: key necessary only if the user wants to send more than three requests per second.

The E-Utilities are nine, and they are described in brief in the following lines.

- *EInfo*: provides general information, such as the number of records in the given database, the date of its last update, and eventual links to other Entrez databases.
- *ESearch*: takes as input the query string matching the results of a specific database.
- *EPost*: returns the web environment for the dataset made with the matching results.
- *ESummary*: returns document summaries of the matching results.
- *EFetch*: returns the results in a specified format.
- *ELink*: provides eventual connections between databases, creating hyperlinks for each UIDs.
- *EGQuery*: returns the number of records that match the query.
- *ESpell*: provides spelling suggestions.
- *ECitMatch*: returns PubMed IDs corresponding to a set of citations.

The functions used in this work are ESearch and EFetch: the first one is needed to match the results of the initial query string, while the second one returns such results in the given format.

2.1.2.1 ESearch

The ESearch Utility takes two mandatory parameters as input: *db* indicates the database in which the research will be conducted (if no database is indicated, the default one is PubMed), and the *term* is the string to search. [21]

The parameters to personalize information retrieval are:

- *restart*: UID index from which the research starts. The default value is 0.
- *retmax*: maximum number of records to retrieve. The default value is 20 and the maximum value is 10.000.
- *rettype*: retrieval type
- *retmode*: format of the returned results.
- *sort*: order of the results, that can be set through the options listed in Table 2-2:

Parameter value	Explanation
<i>journal</i>	Alphabetic order by journal title
<i>pub+date</i>	Chronological order by publication date
<i>most+recent</i>	Chronological order by date added to PubMed
<i>relevance</i>	Order by relevance to the query string
<i>title</i>	Alphabetic order by article title
<i>author</i>	Alphabetic order by author name

Table 2-2 - Options for 'sort' parameter of ESearch function

- *field*: field to limit the research only in title, abstract, PT or others (see the Advanced Search in PubMed).
- *idtype*: type of identifier to return (for sequence databases).
- *datetype*: type of date to retrieve, that can be given with the options as described in Table 2-3:

Parameter value	Explanation
<i>crdt</i>	Create Date: date on which the article is added to PubMed
<i>edat</i>	Entrez Date: equal to the Publication Date if the record enters PubMed more than twelve months after the date of publication
<i>pdat</i>	Publication Date of the article
<i>mhda</i>	MeSH Date: date in which the citation was indexed with MeSH.

Table 2-3 - Options for 'datatype' parameter of ESearch function

- *reldate*: integer to specify the days in which the date should be included.
- *mindate*: filter to set the start date.
- *maxdate*: filter to set the end date.

In this thesis the ESearch function will be used as follows:

```
db='pubmed',
sort='most+recent',
retstart=0,
retmax=max_count,
term=final_query,
field='TIAB',
datatype='pdat',
mindate=low_date,
maxdate=high_date
```

PubMed is set as the database of reference in which to perform the search, the records are retrieved from the most recent, starting by index 0, the maximum number of results is set to 10.000, the query string is set by the user, and the results will be only records that match the query string in the title or abstract ('TIAB'). The

type of date is set as the date of publication of the article, and the user can also choose a specific range of dates on which he can focus the research.

2.1.2.2 EFetch

The EFetch Utility requires as input the *db* parameter and the *id*, that is the record Unique Identifier (PMID). The optional parameters are *retstart*, *retmax*, *retmode*, *rettype*, already explained in the previous section.

In this work, the EFetch function was used with the following parameters:

```
db = 'pubmed',  
id=ids,  
retmode='text',  
rettype='medline'
```

The database is PubMed, *ids* is the list of the PMIDs that match the query string as a result of the ESearch, and the other parameters are set to obtain in output a simple Medline text format of the record and retrieve information through its elements.

2.1.3 String creation for PubMed

The *term* parameter is one of the mandatory parameters to give as input in ESearch. It represents the starting point of the research, and it is a string used to connect with the Entrez engine. The user can choose whatever string to explore a variety of results.

The string can be composed of numbers or letters, and also logic operators like AND or OR can be inserted. Some rules are created to obtain the best results for the user: if the query consists of two or more words, double quotation marks are added in order to search the exact expression a user writes in the corresponding search box; if the string is only one word, it is directly sent to the search engine, without any changes.

2.1.4 Text formats

Once the string was created, and the ESearch and Efetch functions were called, all the results were parsed in order to retrieve specific information. Different formats can be used for PubMed records, but the two most important are the *xml* format and the *text* format.

2.1.4.1 XML format

XML¹⁴ is a markup language, used to structure data with a specific syntax. An XML file is characterized by several tags, to organize and store the text in parts, and to clearly explain the data. Each tag describes the content of the subsequent part of the file, in order for programmers to extract specific information, without reading all the content. The example of the XML file of a PubMed record in Figure 2-3 reports only few data: in this case the information can be obtained through coding by setting the name of the tag and then extracting its content.

2.1.4.2 Medline format

The Medline¹⁵ text format is a simple text characterized by different fields, which describes the content inside. In this thesis, the Medline format was chosen instead of XML because it is simpler, more readable, and faster to download. The example in Figure 2-4 shows what a record in Medline text format looks like, with some of the fields that can be retrieved through coding.

¹⁴ <https://www.indeed.com/career-advice/career-development/xml-file>

¹⁵ <https://www.nlm.nih.gov/bsd/mms/medlineelements.html>

```

<PubmedArticleSet>
  <PubmedArticle>
    <MedlineCitation Status="PubMed-not-MEDLINE" Owner="NLM">
      <PMID Version="1">31920910</PMID>
      <DateRevised>
        <Year>2020</Year>
        <Month>09</Month>
        <Day>30</Day>
      </DateRevised>
      <Article PubModel="Electronic-eCollection">
        <Journal>
          <ISSN IssnType="Print">1664-2295</ISSN>
          <Title>Frontiers in neurology</Title>
        </Journal>
        <ArticleTitle>Vitamin K Antagonist Use and Risk for
        Intracranial Carotid Artery Calcification in Patients With
        Intracerebral Hemorrhage.</ArticleTitle>
      </Article>
    </MedlineCitation>
  </PubmedArticle>
</PubmedArticleSet>
<Abstract>
  <AbstractText>
    <b>Background:</b>
    Intracranial carotid artery calcification (ICAC) on computed
    tomography (CT) is a marker ...
    <b>Materials and Methods:</b>
    We retrospectively semiquantified ICAC on brain unenhanced CT of
    consecutive adult patients ...
    <b>Results:</b>
    Three hundred and seventy-six nontraumatic ICH patients were
    included of whom 77 were using VKAs ...
    <b>Conclusions:</b>
    Our findings do not support VKA use as an independent risk ...
  </AbstractText>
</Abstract>

```

Figure 2-3 - Example of PubMed record in XML format

```

{
'PMID': '31920910',
'IS': '1664-2295 (Print) 1664-2295 (Linking)',
'DP': '2019',
'TI': 'Vitamin K Antagonist Use and Risk for Intracranial Carotid
Artery Calcification in Patients With Intracerebral Hemorrhage.',
'AB': 'Background: Intracranial carotid artery calcification (ICAC) on
computed tomography (CT) is a marker of atherosclerosis and an
independent predictor of vascular events including stroke. While vitamin
K antagonists (VKAs) are used to prevent embolic stroke, they have been
shown to increase levels of both coronary and extracoronary artery
calcification...',
'AU': ['Peeters MTJ', 'Houben R', 'Postma AA', 'van Oostenbrugge RJ',
'Schurgers LJ', 'Staals J'],
'LA': ['eng'],
'PT': ['Journal Article'],
'PL': 'Switzerland',
'JT': 'Frontiers in neurology',
'PMC': 'PMC6933022',
'EDAT': '2020/01/11 06:00',
'MHDA': '2020/01/11 06:01',
'CRDT': ['2020/01/11 06:00'],
'AID': ['10.3389/fneur.2019.01278 [doi]']
}

```

Figure 2-4 - Example of PubMed record in Medline text format

2.1.5 Fields Extraction

Several fields can be extracted from PubMed records, as described in the documentation¹⁶. In this work, only the most important ones, listed in Table 2-4, were retrieved from the Medline text format by using the respective tag, in a fast and easy way.

¹⁶ <https://www.nlm.nih.gov/bsd/mms/medlineelements.html>

Field name	Explanation
<i>PMID</i>	PubMed Unique Identifier
<i>AID</i>	DOI
<i>AU</i>	Authors
<i>TI</i>	Title
<i>AB</i>	Abstract
<i>PT</i>	Type of publication
<i>JT</i>	Journal Title
<i>IS</i>	ISSN Journal Code
<i>DP</i>	Date of publication
<i>SI</i>	Different record data (Clinical Trial numbers)

Table 2-4 - List of fields extracted from PubMed

The PMID helps to identify the record in the PubMed database and to create the corresponding link, in addition to the DOI. Other information like Authors, Title, and Abstract simply describe the content of the record. The type of publication gives an indication of the record type, like RCT, Meta-Analysis, Journal Article, or Observational Study (OS). The utility and reliability of this field will be explained better in the next sections.

2.1.5.1 DOI

The DOI¹⁷ (Digital Object Identifier) is another type of identifier assigned to each publication to provide a permanent uniquely web address where to retrieve the record. It is an alphanumeric string that contains a prefix and a suffix separated by the symbol '/', but it always starts with the expression '10.', as follows:

['10.33529/ANGIO2021420']

¹⁷ <https://apastyle.apa.org/learn/faqs/what-is-doi>

The prefix is the number assigned to organizations and the suffix is assigned to the article publisher. The *AID* field extracted from PubMed can also contain another information, called PII¹⁸ (Personally Identifiable Information). It is a simpler and informal identifier used as internal organizations numbering system, in fact it can be included also in the DOI string.

In the following example both parts are reported, but the code developed in this thesis extracts only the DOI code, to avoid redundant information:

```
[ '10.1097/MD.0000000000027906 [doi]', '00005792-202112030-00028 [pii]' ]
```

2.1.5.2 Journal Information

This field contains the full Journal Title in which the record is published and the ISSN¹⁹ (International Standard Serial Number), which is a code related to the corresponding Journal useful to identify it uniquely. It is an 8-digit code, composed of the acronym ISSN and two groups of four digits and it is associated with the publication title. It can be referred to the printed copy of the journal, or to the electronic one. In the databases created in this thesis, all the information related to the ISSN were maintained, without any distinction between print or electronic, to guarantee the possibility to go back to the specific Journal in which the article is published.

2.1.5.3 Publication Date

The date extracted and then inserted in the final database is the Publication Date, so the date on which the record is published on the corresponding Journal. This information has a standardized format with two/three elements: the first one is a 4-digit year, the second one is related to the month or the season and a 1 or 2-digit day can be present or not, to indicate the exact day of publication. Table 2-5 shows different examples of dates extracted from the Medline text format, with all the adjustments made to obtain structured information with the same format of the date

¹⁸ https://www.doi.org/10DEC99_presentation/faq.html#2.6

¹⁹ <https://www.issn.org/understanding-the-issn/what-is-an-issn/>

extracted from Google Scholar records. The last column represents the information inserted in the final database.

Medline Date	Date after conversion	Date in “date” type
2020 Feb	2020 Feb	2020-02-01
2019 Winter	2019 Jan	2019-01-01
2019 Sum	2019 Jul	2019-07-01
2019 Autumn	2019 Oct	2019-10-01
2019 Nov – Dec	2019 Nov	2019-11-01
2017 Sep 12	2017 Sep 12	2017-09-12
2019 Sep/Oct	2019 Sep	2019-09-01
2019 Ago	2019 Aug	2019-08-01

Table 2-5 - Examples of dates extracted from PubMed with respective conversion

Note that if the date contains a season instead of the exact month, this one will be substituted with the corresponding middle month of the season. If there are two months, only the first one will remain in the final database. Some other corrections were made in order to correct eventual typing errors.

2.1.5.4 Secondary Source

Another important retrieved information is the Secondary Source, described in the SI field. It contains information related to the trial number, if present, and it consists of one or more strings composed by a source and a number. Only regularly registered clinical trials contain this information. Table 2-6 reports some examples.

Secondary Source	Corresponding Register
<i>ClinicalTrials.gov/NCT02642419</i>	NIH and FDA Register
<i>ChiCTR/ChiCTR1900025859</i>	Chinese register
<i>NTR/NL4776</i>	Netherlands register
<i>ANZCTR/ACTRN12620000285954</i>	Australian/New Zeland register
<i>UMIN-CTR/UMIN000023403</i>	Japanese register
<i>EudraCT/2019-003756-37</i>	European Union Drug Regulating Authorities Clinical Trials Database

Table 2-6 - Examples of Secondary Source information with corresponding Register

The Secondary Source field importance will be better explained in the Classification section: this expression will be used to classify records directly as RCT.

2.1.6 DataFrame Creation

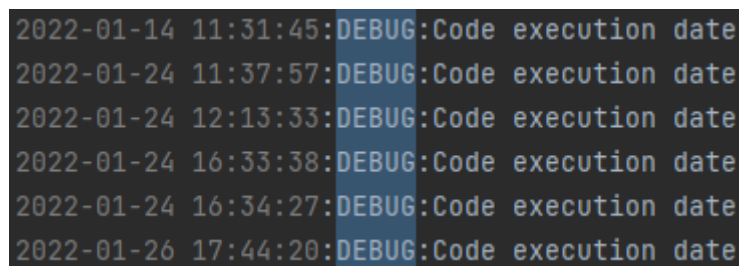
The PubMed Data Frame was created through different steps: the goal was to obtain a complete and new file if no queries were already done with that specific string, or to update the results with only new information if it was already existing. In this way, the process of downloading records is simplified by avoiding retrieving already existent data. In order to do so, the properties of the Logging file will be explained in the next section.

2.1.6.1 Logging File

The *logging*²⁰ module is built-in in Python. It is a sort of document used to keep track of the events in the code. These events can be referred to the simple execution of a function, to some warnings that can slow down the process, or to errors that occur while the code runs. First of all, to create the logger, it is necessary to import the corresponding module from the library, and then the logger can be created and configured. It is mandatory to indicate the filename in which the information will be saved, and the level used. In this thesis the logger was created as follows:

```
logging.basicConfig(filename='pubmed.log',
                    level=logging.DEBUG,
                    format='%(asctime)s:%(levelname)s:%(message)s',
                    datefmt='%Y-%m-%d %H:%M:%S',
                    force=True)
```

The chosen filename was *'pubmed.log'* to save inside this file only PubMed information, the level is *DEBUG*, and the other parameters are the message format that will be displayed in the file and the date format. The debug message *'Code execution date'* was created to help retrieving the information of the timing in which the code is executed in terms of day and time, as shown in Figure 2-5.



```
2022-01-14 11:31:45:DEBUG:Code execution date
2022-01-24 11:37:57:DEBUG:Code execution date
2022-01-24 12:13:33:DEBUG:Code execution date
2022-01-24 16:33:38:DEBUG:Code execution date
2022-01-24 16:34:27:DEBUG:Code execution date
2022-01-26 17:44:20:DEBUG:Code execution date
```

Figure 2-5 - Example of lines in a log file

²⁰ <https://www.geeksforgeeks.org/logging-in-python/#:~:text=Python%20has%20a%20built%2Din,what%20problems%20have%20been%20arisen>

A new line is added in the logging file every time the code runs: then the file is opened in a 'read' mode, and the last line is read to retrieve information about the current date of the code execution.

As shown in Figure 2-6, this information will be automatically used to create the filename of the database, as composed by two parts: the first one is the query string defined by the user, while the second part is the execution date. In this way, the user can keep track of the research already carried out and in terms of the search string and the last date of execution, to understand if the previously extracted database needs an update, or to decide to proceed using a different string.

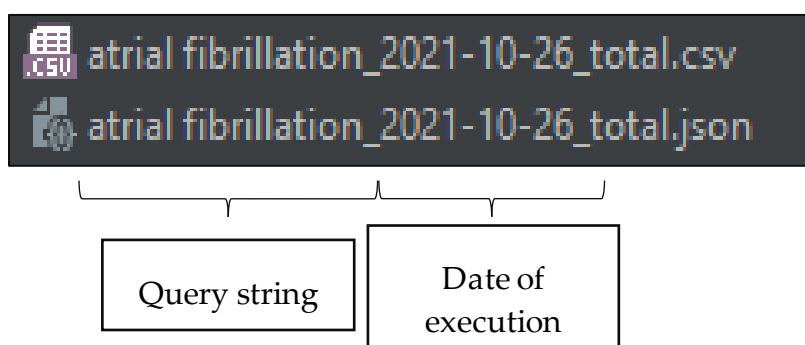


Figure 2-6 - Description of database filename

2.1.6.2 Data Frame Creation steps

After the logging file creation, the user will insert a query string through the corresponding interface. The database, named as the string, will be searched in local folders to check if it already exists or not.

If the database does not exist, all the fields described in Section 2.1.5 will be retrieved for the first time: all the resulting PMIDs will be searched through the Entrez search engine in order to save information like the Article Title, the Article Abstract, the Authors and so on. The final Data Frame columns will contain such information.

If the database already exists, its creation date and the actual date will be compared to verify the time spanned. A rule was introduced to proceed with the update if at least a month was passed, otherwise no new search will start. In the new search for updates, only new PMIDs not present in the old database will be searched and only

new info will be added, in order to reduce the total computational time and to avoid redundant research, with the old and the new databases finally merged to obtain the complete and updated result. Figure 2-7 illustrates the total workflow.

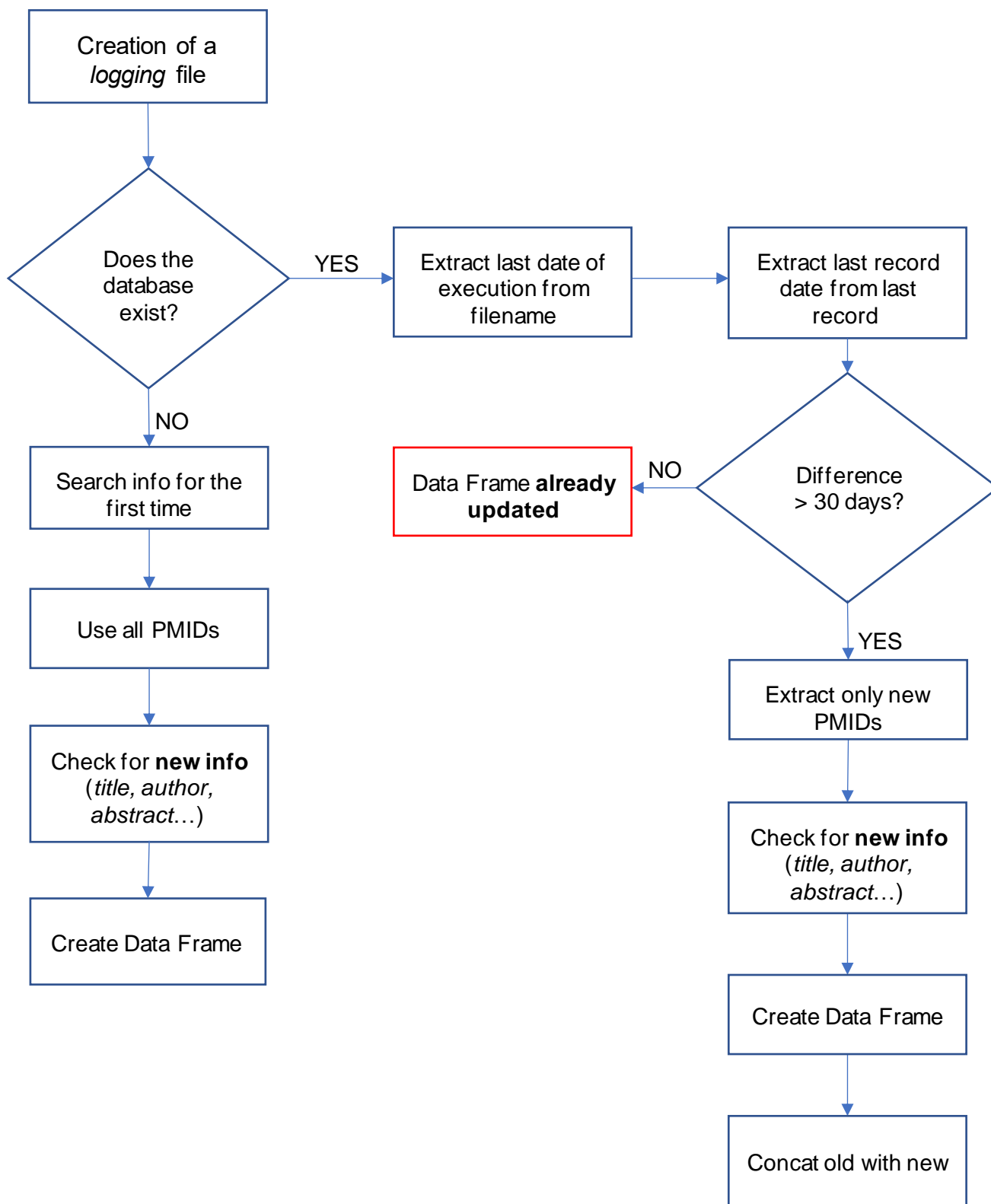


Figure 2-7 - Workflow for PubMed database creation

2.2 Search in Google Scholar

The searching process implemented to access Google Scholar was completely different from the one performed for PubMed, due to their different organization and provided functionalities.

Google Scholar sorts the results by relevance, and it includes a greater variety of publications, including theses, books, or articles, while PubMed is more focused on medical literature. [22] Google Scholar does not allow users to find matches only in specific parts of the text (like title or abstract), but it gives matches between the string and the results in the entire article. PubMed provides more specific filters, like the possibility to filter by type of publication or by journals. Due to the differences between these two search engines a different algorithm was implemented.

2.2.1 String creation for Google Scholar

The creation of the query string for Google Scholar is a bit different from the one described in Section 2.1.3.

Even in this case the string can contain numbers, letters, or logic operators, but an additional rule has been specifically introduced for this work: if the user inserts two or more words, the query is split and a '+' symbol is added in the middle, together with double quotation marks". Otherwise, if the string is one word only, no modifications are made. This was necessary because the final expression will be inserted directly in the URL utilized to create the connection with the search engine.

2.2.2 URL creation

The URL²¹ (Uniform Resource Locator) is necessary to create a connection between the code implemented in Python and the corresponding resource on the Web. It represents an address used to access a single resource, composed of different parts associated with different fields.

Figure 2-8 shows an example of a Google Scholar URL with the following elements:

- *Scheme*: mandatory element to indicate the protocol used to request the resource (in this case it is HTTPS).
- *Authority*: part that contains the domain, so the requested Web server.
- *Path to resource*: path in which the resource will be found.
- *Parameters*: some changeable values to personalize the research.

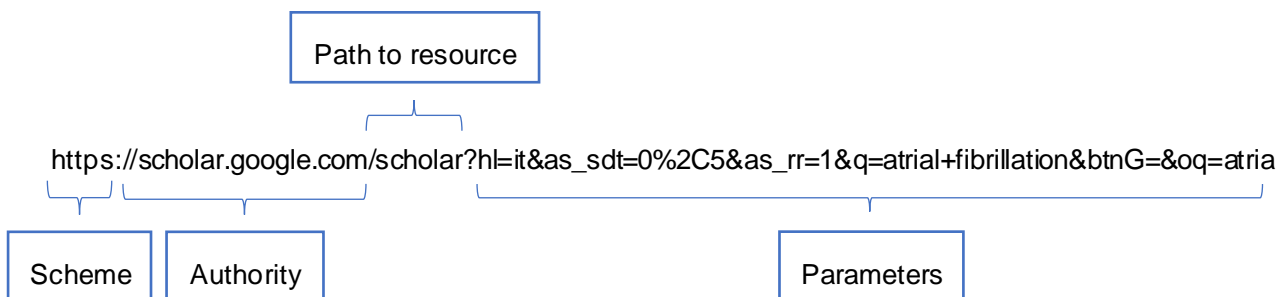


Figure 2-8 - Example of URL

The URL for the research in this thesis was organized as follows:

`https://scholar.google.com/scholar?start=0&q=%22atrial+fibrillation%22&hl=it&scisbd=1&as_sdt=0&as_rr=1`

The scheme and the domain are referred to the Google Scholar website. The '`start=0`' is necessary to indicate that the research starts from the first page. The number 0 is automatically substituted at each iteration in order to conduct the research also on other pages, as only 10 records per page are shown. The '`atrial+fibrillation`'

²¹ https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL

expression is an example of the string inserted by the user and adjusted with the rules explained in Section 2.2.1. The parameter '*scisbd*' set to 1 defines articles sorted by date from the most recent published. The parameter '*as_sdt*' is set to 0 to exclude patents while the parameter '*as_rr*' is set to 1 to include only scientific articles. This URL model was created when the user does not consider the filter by date for the results.

When the date filters were used the URL model becomes as follows:

```
https://scholar.google.com/scholar?start=10&q=%22atrial+fibrillation%22&hl=it&as_sdt=0
&as_ylo=2017&as_yhi=2019&as_rr=1
```

Two more parameters were added: '*as_ylo*' indicates the starting date and '*as_yhi*' indicates the end date.

2.2.3 Scraping Information

Web Scraping was used in this thesis to collect information from the Google Scholar website in a meaningful way and organize it as a database. The next sections will explain in detail the implemented process, which consists of the following phases:

- Access to initial Google Scholar webpage
- Parsing of general HTML pages
- Parsing of specific HTML pages
- Use of CrossRef for information retrieval
- Extraction of BibTex format
- Data Frame creation

2.2.3.1 Access to initial Google Scholar webpage

After the URL creation based on the user requirements, the scraping begins by accessing the starting Google Scholar page (Figure 2-9).

The *requests* library [XIII] in Python allows downloading webpages and retrieving information from them. This module was used to send HTTP requests for opening

URLs on the Internet through code, to provide a response object from which a programmer can extract the data.

This powerful library automatically decodes contents providing the results as text. The used function takes as input two mandatory parameters, the URL and the *headers*. The first one is the specific link to the webpage and the second one is necessary to have the authorization for accessing the website, since Google Scholar requires a login. After sending the request, the response status code is verified to check if the request is considered dangerous or not: if everything is working correctly, the response will be downloaded as text and then parsed to retrieve information.

The screenshot shows the Google Scholar search results for the query "atrial fibrillation". The search bar at the top contains the text "atrial fibrillation" and a search icon. Below the search bar, the results are displayed in a list format. On the left side, there are filters for "Articoli" (Articles) with a count of "Circa 1.540.000 risultati (0,08 sec)". There are also options for "In qualsiasi momento" (Any time) with filters for "Dal 2022", "Dal 2021", and "Dal 2018", and an "Intervallo specifico..." option. There are also options for "Ordina per pertinenza" (Sort by relevance) and "Ordina per data" (Sort by date). On the right side, there are options for "Qualsiasi lingua" (Any language) with "Pagine in Italiano" (Pages in Italian), and "Qualsiasi tipo" (Any type) with "Articoli scientifici" (Scientific articles). There are also checkboxes for "includi brevetti" (include patents) and "includi citazioni" (include citations), and a "Crea avviso" (Create alert) button. The search results are listed in three columns. The first column shows the title and source of the article. The second column shows a brief abstract of the article. The third column shows the full text or PDF link. The first result is "New ideas about atrial fibrillation 50 years on" by S Nattel, published in Nature in 2002. The second result is "Management of atrial fibrillation" by ELC Pritchett, published in the New England Journal of Medicine in 1992. The third result is "Canadian atrial fibrillation anticoagulation (CAFA) study" by SJ Connolly, A Laupacis, M Gent, and RS Roberts, published in the Journal of the American Medical Association in 1991.

Figure 2-9 - Google Scholar initial page

2.2.3.2 Parsing of general HTML pages

The *BeautifulSoup* library was used to do the parsing by navigating into the HTML page already downloaded and retrieving specific information.

HTML pages are structured in sections to organize content on the webpage in a clear way [XIV]. As reported in Figure 2-10, the most important parts are:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
  <HTML>
    <HEAD>
      <TITLE>My first HTML document</TITLE>
    </HEAD>
    <BODY>
      <P>Hello world!
    </BODY>
  </HTML>
```

Figure 2-10 - Example of HTML page structure

- The DOCTYPE declaration that contains HTML version information.
- The HTML section contains information about the language used
- The HEAD section contains information not displayed on the webpage
- The BODY section contains the document content, like text, images, or links
- The TITLE, placed in the HEAD, contains the web browser's title bar.

In this thesis, data were taken from the BODY and Figure 2-11 illustrates how a record is shown in the Google Scholar starting page (2-11a) with the corresponding HTML content (2-11b). Each publication is reported with the title on the top, and other information listed below: authors, journal of publication, year of publication, and the website in which the article was published. Then, a small part of the abstract is shown, with the number of records citations and the button to export the record in BibTex format (Figure 2-11a).

From this page, only title, publication year, and authors were retrieved, because they represent the only complete and clear information. The corresponding tags are highlighted in Figure 2-11b: for example, the tag '*gs_rt*' contains the specific link of the record, so it is possible to obtain the string of the link simply by extracting the content inside such tag, while "*gs_a*" contains information about the authors and the year.

The abstract is reported only partially, so it was not extracted from this page, to avoid redundant code and to reduce the time of downloading information. The

abstract extraction will be explained in the next sections, by searching and scraping each single HTML article page.

Atrial Fibrillation and Dementia: A Report From the AF-SCREEN International Collaboration

L Rivard, L Friberg, D Conen, JS Healey, T Berge... - Circulation 2022
- Am Heart Assoc

12 giorni fa - ... Suspected mechanisms of cognitive impairment in **atrial fibrillation**. Suspected mechanisms linking **atrial fibrillation** (AF) and cognitive impairment are depicted by solid black arrows. AF could lead to cognitive impairment through different mechanisms: cerebral ...

☆ Salva 📄 Cita Tutte e 5 le versioni 📄

[HTML] uwa.edu.au
ACNP Full Text

```

▼<div class="gs_ri">
  ▼<h3 class="gs_rt" onclickstart="gs_evt_dsp(event)">
    ▶<a id="qeYJOFUNr98J" href="https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.121.055018" data-clk="hl=it&sa=T&ct=res&cd=21&d=16118116251045848745&ei=pAQJYtTUE6mVy9YP6eqWsAU" data-clk-atid="qeYJOFUNr98J">...</a>
  </h3>
  ▼<div class="gs_a">
    "L Rivard, L Friberg, "
    <a href="/citations?user=9nezV0sAAAAJ&hl=it&oi=sra">D Conen</a>
    ", "
    <a href="/citations?user=iuyiaJMAAAAJ&hl=it&oi=sra">JS Healey</a>
    ", T Berge...&nbsp;- Circulation, 2022 - Am Heart Assoc"

```

Figure 2-11 - On the top (2-11a) the organization of a record in the Google Scholar starting page. On the bottom (2-11b) the corresponding HTML content

2.2.3.3 Parsing of specific HTML pages

The analysis of the HTML page of every single article was necessary to extract the abstract and the DOI. They are fundamental information, because the abstract will be used to classify the record in the corresponding type of publication (i.e., RCT or SRMA), as it will be explained later on, while the DOI is compulsory to get access to the BibTex format of the article, from which other important information can be retrieved.

The scraping of pages can be static or dynamic [XV], and they will be used both in the proposed approach. Static Web Scraping is used with pages that do not allow interaction with the user, but they show content without changes. The process of

sending requests for a static page is shown in Figure 2-12: when the server receives a request, it provides a response without additional actions, so HTML pages are analyzed simply as texts, and information inside specific tags are extracted using the *BeautifulSoup* library because it converts the web page in a tree of tags, attributes, elements, and values. This library is multiprocessing, so it runs multiple threads simultaneously and can parse webpages in parallel, decreasing the computational time.

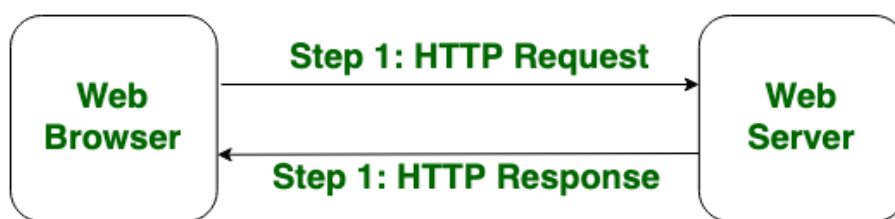


Figure 2-12 - Static Page protocol. Source: [XVI]

Dynamic Web Scraping is used for pages that display different content each time, depending on the type of visitor. They can also change according to the moment of the day in which they are visited [XVII]. Dynamic pages can use *client-side scripting*, so they change in response to simple actions done by the user, or *server-side scripting*, like login pages, which change after the loading.

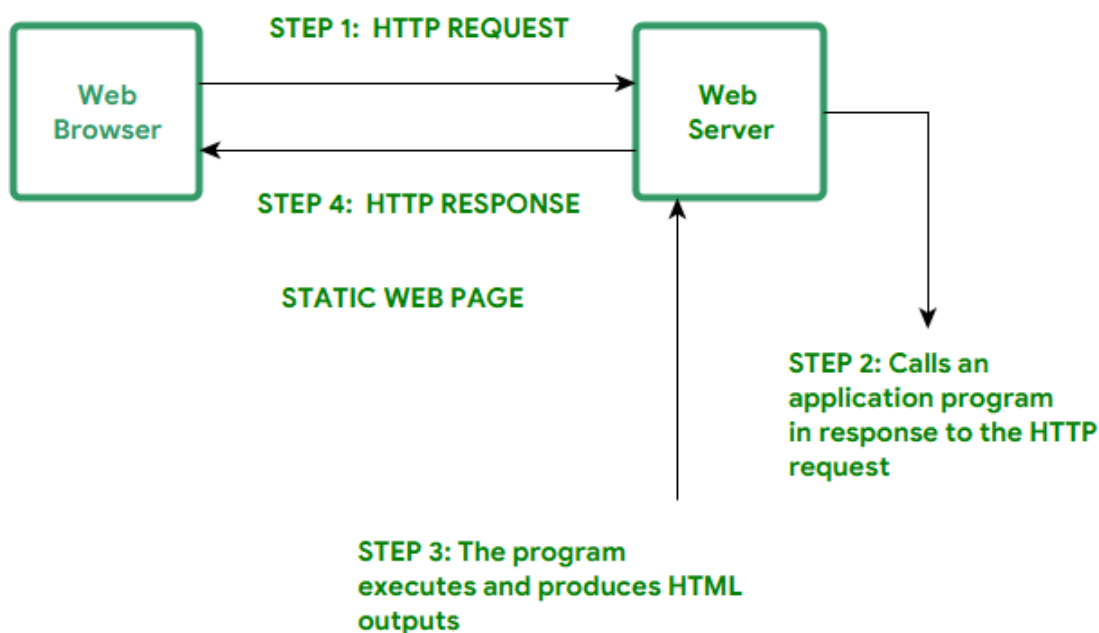


Figure 2-13 - Dynamic Page protocol. Source: [XVI]

They are displayed on the Web like static pages but when a request is sent to open the page, there are internal steps that are executed before sending a response: there is the necessity to read the code written to construct the page, set the page following the instructions, and then remove the code from such page. Figure 2-13 illustrates the process.

Selenium library can scrape this kind of webpages, but the process requires more time than for static Web scraping.

- **DOI extraction in Static Pages**

The DOI information is very important for different reasons: 1) it represents a code to have direct access to the record link, in the specific journal webpage, and it is also mandatory to download the BibTex format of each record; 2) when the search process is completed, the DOI code is used as a common datum between PubMed and Google Scholar to identify eventual duplicates.

Figure 2-14 reports a blocks diagram for the DOI extraction in the case of Static Web Pages. The DOI (as explained in Section 2.1.5.1) is composed of alphanumeric characters, and it always starts with the expression '10.'. This starting expression was used to verify if it was already present in the article link, retrieved from the general Google Scholar page. After scraping all the general pages, the records links were compared with the expression '10.'. If there was a match, it meant that the DOI code was already present in the link, so the string was split and only the last part was extracted and saved as DOI.

If the expression was not present in the link, the Web Scraping of the single HTML page started to retrieve the desired information. A list of the most popular medical journal websites was predefined, in which the HTML tag containing the DOI information was searched, as reported in Table 2-7.

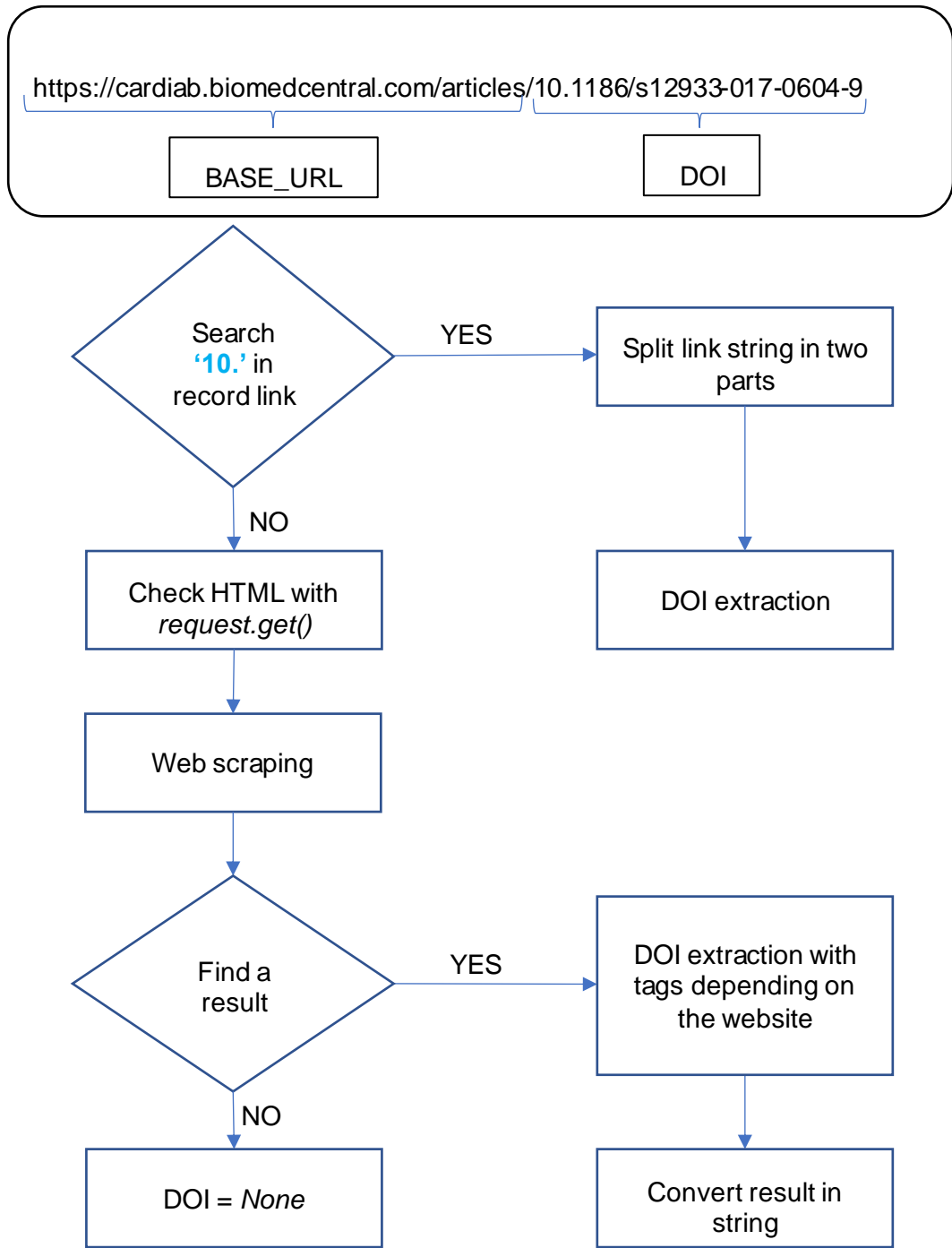


Figure 2-14 - Block diagram for DOI extraction

Website	HTML tag for DOI
Nature.com	<i>c-bibliographic-information__list-item</i> <i>c-bibliographic-information__list-item –doi</i>
Sciencedirect.com	<i>ArticleIdentifierLinks</i>
Jamanetwork.com	<i>pub-history-row clearfix</i>
Academic.oup	<i>citation_doi</i>

Table 2-7 - Examples of medical website and corresponding tag for DOI

The HTML text was scraped with a function that searches if a specific tag is present or not in the text. If the tag was found, the contained information inside is extracted and then converted into a string after the removal of useless characters. Sometimes, even if the DOI was easily extracted, not needed information that could make the DOI string wrong could be present. For this reason, once the DOI list was completed, it was re-checked to delete all the expressions like */html*, */pdf*, */abstract*, because sometimes the code could be at the center of the link string, so characters at the beginning and at the end would need to be removed. This step was necessary because the DOI string will be used in combination with another string (that is the base URL) to form the direct link to the article, and then to download the BibTex format.

- **Abstract extraction in Static Pages**

The extraction process of the abstract for static pages is very similar to the one described in the previous section. Each HTML page was downloaded as text and then scraped to extract the information under the specific tag. If no information was found, the abstract was considered as None; otherwise, the result was converted into string and saved into the Data Frame.

The most important medical websites were analyzed as before, and the Table 2-8 reports the tags corresponding to the abstract content.

Website	HTML tag for abstract
nejm.com	<i>article_Abstract</i>
ahajournals.com	<i>hlfld-Abstract</i>
nature.com	<i>Abs1-content</i>
Sciencedirect.com	<i>abstracts</i>
Jamanetwork.com	<i>citation_abstract</i>

Table 2-8 - Examples of medical website and corresponding tag for Abstract

Figure 2-15 reports an example of a record static webpage. The title is at the top of the page, with authors and citations information, then there are the DOI link and the abstract.

Controversies in atrial fibrillation TITLE

Prof Stanley Nattel MD ^a, , Prof Lionel H Opie FRCP ^b

[Show more](#)

[+](#) Add to Mendeley [🔗](#) Share [📄](#) Cite

[https://doi.org/10.1016/S0140-6736\(06\)68037-9](https://doi.org/10.1016/S0140-6736(06)68037-9)

DOI

Get rights and content

Summary ABSTRACT

Atrial fibrillation is the most common sustained cardiac arrhythmia, and contributes greatly to cardiovascular morbidity and mortality. Many aspects of the management of atrial fibrillation remain controversial. We address nine specific controversies in atrial fibrillation management, briefly focusing on the relations between mechanisms and therapy, the roles of rhythm and rate control, the definition of optimum rate control, the need for early cardioversion to prevent remodelling, the comparison of electrical with pharmacological cardioversion, the selection of patients for long-term oral anticoagulation, the roles of novel long-term anticoagulation approaches and ablation therapy, and the potential usefulness of upstream therapy targeting substrate development. The background of every controversy is reviewed and our opinions expressed. Here, we hope to inform physicians about the most important controversies in this specialty and stimulate investigators to address unresolved issues.

Figure 2-15 - Example of article published in sciencedirect.com: [XVIII]

- **DOI and Abstract extraction in Dynamic Pages**

*Selenium*²² is a web-based automation tool used to scrape information from the Internet.

It requires the installation of a web driver, which will open the URL that is passed inside to navigate the page. The process is time-consuming because the driver waits until the page is fully downloaded [XIX].

In this thesis, data were extracted through XPath, a language that helps identifying specific information inside a page by providing the exact location of all the elements from the root. Since the XPath is different for every single element in a page, only five medical websites were scraped with the Selenium library. They are all online resources in which there are some technical publications about science, engineering, or medicine, and they are listed below:

- *IEEE Xplore* (<https://ieeexplore.ieee.org>)
- *Europe PMC* (<https://europepmc.org>)
- *ScholarWorks* (<https://scholarworks.calstate.edu>)
- *NCBI* (<https://www.ncbi.nlm.nih.gov>)
- *Optica Publishing Group* (<https://www.osapublishing.org>)

The DOI XPath and the abstract XPath are the same for each record published on these websites, so they were manually retrieved once and then used in the code to extract the content in those locations. The process can require several minutes, but it allows to reduce the number of None elements in the final database.

2.2.3.4. Use of CrossRef for information retrieval

CrossRef is an official Registration Agency born in early 2000 used by publishers to enable citation linking in journals using the DOI. It was created to provide access for societies to a huge amount of records, in order to share information in an easy and rapid way [23].

²² <https://selenium-python.readthedocs.io/>

It is used as a large database, in which publishers can upload metadata of records, that can be retrieved through DOI. These data include the URL, the journal information, the title, the publication year, and so on. Once a record is deposited into the database, all the subscribed users can have access to it and can use its information, also to create a bibliography.

Nowadays CrossRef contains about 18 million DOIs referred not only to medical records but also to books, images, and other contents. From 2020, some publishers joined a particular initiative created to provide free access to abstracts records [XX]: this was done in order to facilitate the retrieval of information and it is one of the reasons why this resource is used in this thesis. In fact, sometimes the *Beautiful Soup* library is not enough to extract the abstract, because there were cases in which tags were not found or information was not available. To solve this problem and to reduce the number of None abstracts, the CrossRef API was used to have access to the ones not retrieved through Web Scraping, but also to make title adjustments in case of mistakes.

- **Title Adjustments with CrossRef**

The titles are one of the first data extracted with Web Scraping from the Google Scholar starting page. The problem is that sometimes the information extracted contains also other useless characters (like *[HTML]* or *[PDF]*). These expressions need to be deleted because the title will be used as a comparator between records to identify if there are duplicates in the final database, in case the DOI code is not available. For this reason, the title string should be as similar as possible between PubMed and Google Scholar, otherwise duplicates would not be found.

To solve this issue, once the titles were extracted, they were screened to verify if they contained wrong characters. If not, titles were maintained without changes; if yes, these characters were deleted, and the code saved their position in order to extract them with the CrossRef API.

The CrossRef API allows obtaining an XML file with lots of information. Figure 2-16 reports an example of a title retrieved from Google Scholar with the character *[HTML]*. The position of such a title was saved, and the corresponding DOI was used to search the XML record format. The correct title was extracted and

substituted to the one previously retrieved. In this way, it was possible to obtain a title without any errors.

[HTML] Controversies in atrial fibrillation
S Nattel, LH Opie - The Lancet, 2006 - Elsevier
... Various mechanisms, including rapid local ectopic activity, single-circuit re-entry, and multiple-circuit re-entry, can cause atrial fibrillation... atrial fibrillation is due to a single primary circuit, it can be suppressed by linear ablation within the re-entry pathway; whereas atrial fibrillation ...
☆ Salva Cita Citato da 228 Articoli correlati Tutte e 14 le versioni Web of Science: 127 Importa in BibTeX



```
<titles>  
  <title>Controversies in atrial fibrillation</title>
```

Figure 2-16 - Example of title modified with CrossRef

- **Abstract extraction with CrossRef**

The CrossRef API is a useful resource also in case of missing abstracts. The process was similar to the one implemented for the titles because all the positions of None abstracts were saved at the beginning of the scraping and then the corresponding DOIs were used to download the XML record format. An example of abstract retrieved in this way is reported in Figure 2-17: information under the tags *jats:abstract* and *jats:p* were extracted and inserted into the Data Frame.

```
<jats:abstract xmlns:jats="http://www.ncbi.nlm.nih.gov/JATS1" xml:lang="en">  
  <jats:title>Summary</jats:title>  
  <jats:p>Established primary prevention strategies of cardiovascular diseases are based on understanding of risk factors, but whether the same risk factors are associated with atrial fibrillation (AF) remains unclear. We conducted a systematic review and field synopsis of the associations of 23 cardiovascular risk factors and incident AF, which included 84 reports based on 28 consented and four electronic health record cohorts of 20,420,175 participants and 576,602 AF events...</jats:p>  
</jats:abstract>
```

Figure 2-17 - Example of abstract in CrossRef XML document

With this alternative method it was possible to further reduce the number of None in the final database and to obtain more complete data to make the Classification process as more correct as possible.

2.2.3.5. Extraction of BibTex format

The BibTex format is a standardized format used to simplify the process of storing bibliographic data. Google Scholar gives the possibility of downloading this format and using it to retrieve information [XXI].

```
@article {Hussain 2017,  
  title={Vitamin D supplementation for the management of knee  
  osteoarthritis: a systematic review of randomized controlled  
  trials},  
  volume={37},  
  ISSN={1437-160X},  
  url={http://dx.doi.org/10.1007/s00296-017-3719-0},  
  DOI={10.1007/s00296-017-3719-0},  
  number={9},  
  journal={Rheumatology International},  
  publisher={Springer Science and Business Media LLC},  
  author={Hussain, Salman and Singh, Ambrish and Akhtar, Mohd and  
  Najmi, Abul Kalam}, year={2017},  
  month={Apr},  
  pages={1489-1498}  
}
```

Figure 2-18 - Example of BibTex format

This kind of document always starts with an entry type name that represents the type of the content and a key used to identify the file. An example of a Google Scholar record structured with BibTex is reported in Figure 2-18: the entry type is *article* and it indicates that the contents inside are referred to an article published in a journal, while the identifier is the first author that appears in the list of authors, followed by the year of record publication. The other items represent fields that contain different record information. The data extracted from the BibTex file are the exact date of record publication (*year, month*), the name of the journal (*journal*), and the ISSN code (*ISSN*).

- **Publication date retrieval**

After the extraction of the publication year from the general Google Scholar webpage (as explained in Section 2.2.3.1), some errors emerged. To deal with these mistakes and to obtain correct data, publication dates were also extracted from the BibTex format and then substituted to the data retrieved initially. The following Regex extract the content after a specific string inside parentheses and were used to extract year and month:

$$year=\{[\^]*\}$$
$$month=\{[\^]*\}$$

If the publication year was available in the BibTex file, it substituted the old one retrieved from the starting page. If also the month was present, it was added to the year to have more precise data; otherwise, January was assigned as the month by default. Sometimes the BibTex document can also contain the exact day of publication: in this case, it was extracted and inserted in the same field of year and month; if it was not present, the first day of the month was added by default.

- **Journal information retrieval**

The Journal information was extracted in the same way of year and date, using the following expressions:

$$journal=\{[\^]*\}$$
$$ISSN=\{[\^]*\}$$

These data were retrieved to obtain the same fields extracted in PubMed and helped to verify if the BibTex files had already been downloaded or not. The presence of the 'Journal' field was checked every time a new searching process was starting, to avoid the retrieval of already existent data. If such info was present in the database, it means that BibTex formats were already parsed, and no redundant steps were needed. If the 'Journal' field was present, but some items were missing, it means that some connection problems occurred, so the None positions were saved, and only the BibTex for such positions were downloaded again. Figure 2-19 illustrates the workflow of BibTex information retrieval.

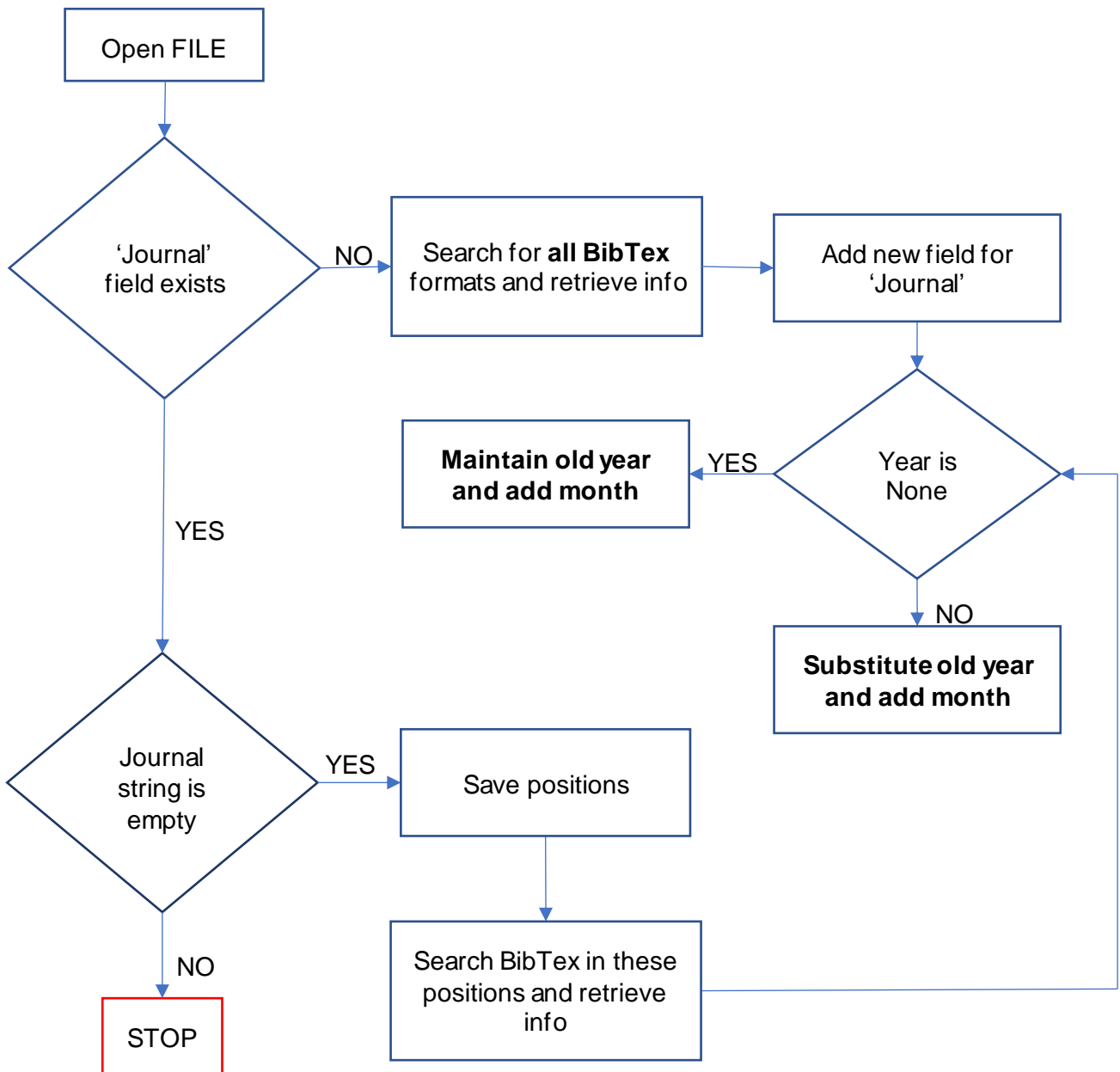


Figure 2-19 - Blocks diagram for BibTeX information retrieval

2.2.3.6. Data Frame Creation

The Data Frame creation for Google Scholar was organized in the same way as PubMed. The idea was to keep it updated every month as before by extracting the date of current code execution from the logging file (as explained in Section 2.1.6.1), now called '*gscholar.log*', and to search only new data if a database already existed.

The difference compared to Pubmed is in the way of retrieving information: Google Scholar records don't have PMIDs, so it was not possible to search only new info through these identifiers. For this reason, if a database did not exist, all the necessary fields were extracted with the methods described in previous sections, but if it did already exist, it was used to compare old data with the new ones, to avoid unnecessary search and to reduce computational time.

All the old records links were collected in a list, and when the new links were extracted from the Google Scholar starting page, they were searched into such list to save the positions of already existing items. These positions will be deleted from the new links list, and the scraping will be done only for the remaining ones. At the end, the Data Frame, containing only new data, will be appended to the old one. Figure 2-20 illustrates the total workflow.

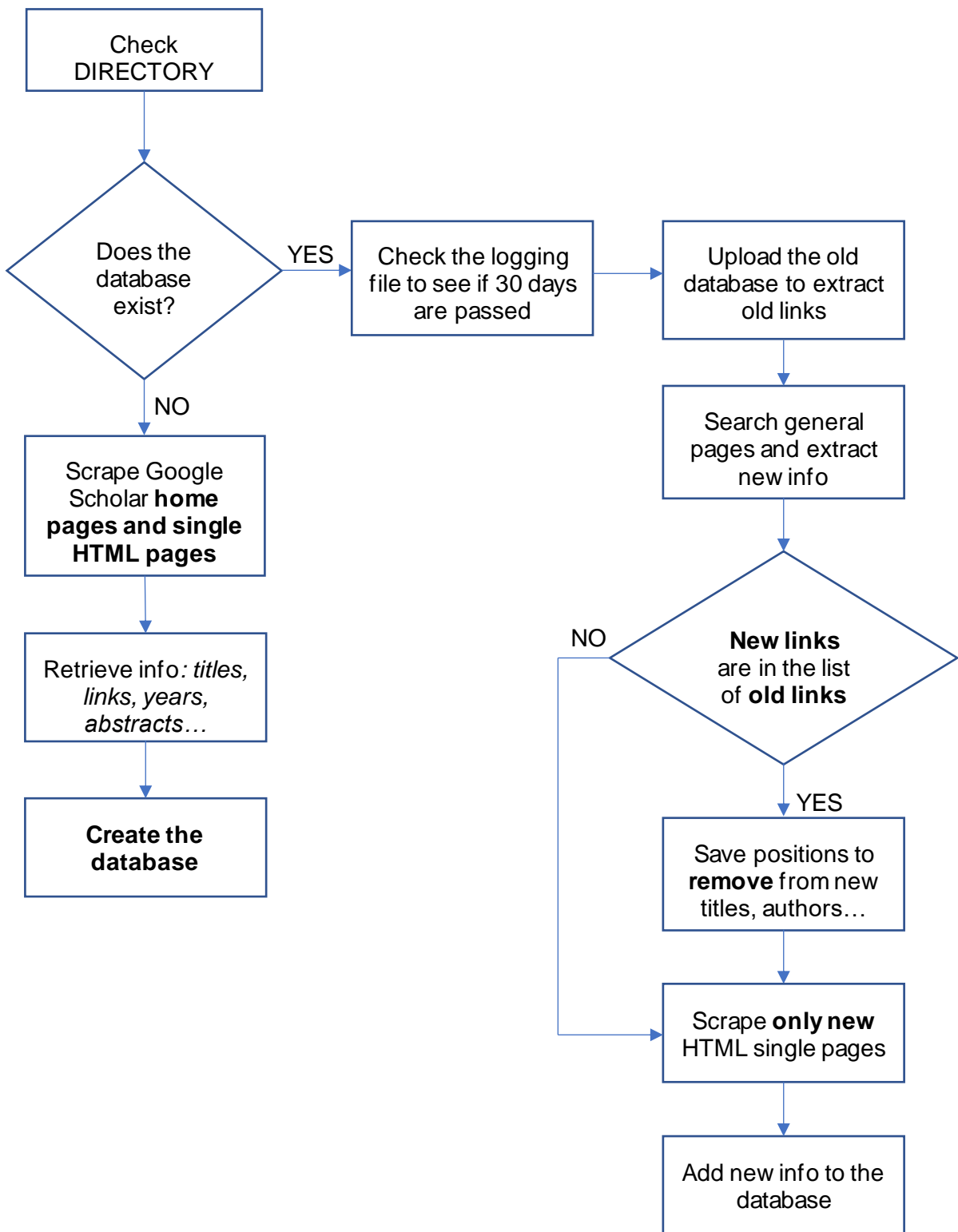


Figure 2-20 - Blocks diagram for Google Scholar database creation

2.3 Total Database

The PubMed and Google Scholar databases were created separately, but at the end of the searching process, they were merged to obtain a unique database.

Also in this case a logging file was created (*'total.log'*) but simply for extracting the current execution date and inserting it into the filename. The principle was the same: if the total database already exists, and there are new data in Google Scholar or PubMed, such data will be inserted in it. The old database will be moved to another folder, to give the user the possibility to also retrieve the old data.

If the total database did not exist, it was created by concatenating the information taken from the searches to the two online resources, after removing duplicate records.

To do so, all the DOIs were compared to verify if some records were present multiple times. The DOI information was chosen because it represents a unique identifier, different for all the records, that never changes.

A specific function was created to identify the duplicate elements in the final DOI list, and then the corresponding Google Scholar record with a duplicate DOI was deleted. The choice of eliminating the record retrieved from Google Scholar was due to the lower precision of its information in each field compared to the record extracted from Pubmed, and fewer None values.

Field	PubMed	Google Scholar
<i>PMID</i>	34689896	
<i>DOI</i>	10.1016/j.ccep.2021.07.006	10.1016/j.ccep.2021.07.006
<i>Title</i>	Congenital Heart Block.	Congenital Heart Block
<i>Abstract</i>	"Congenital complete heart block (CCHB) defines atrioventricular..."	
<i>Link</i>	https://pubmed.ncbi.nlm.nih.gov/34689896/	https://www.cardiacep.theclinics.com/article/S1877-9182(21)00085-X/abstract
<i>Authors</i>	['Steinberg L']	L Steinberg
<i>Date of Publication</i>	2021-12-01	2021-12-01
<i>Journal</i>	Cardiac electrophysiology clinics	Cardiac Electrophysiology Clinics
<i>ISSN</i>	1877-9190 (Electronic) 1877-9182 (Linking)	1877-9182

Table 2-9 - Example of duplicate record with PubMed and Google Scholar info

In Table 2-9 a comparison between a duplicate item is shown: the abstract was present for the PubMed record, but it was not retrieved from Google Scholar. Also, the ISSN code was more complete for PubMed. Another reason for maintaining the PubMed record is that it allows retrieving the NCT Trial Registration number, very useful for record classification.

In case the DOI was not present in a record, because it was not available or the scraping was not efficient enough to retrieve it, the title was used as a comparator. In Table 2-10 it is possible to see that the PubMed title ends with a '.'. For this reason, when the titles were compared, the last two characters were removed from the string to make them equal between the two databases.

Once the duplicates were identified, the corresponding positions were removed from the total database, and the final result was saved both in '.csv' and '.json' format.

2.4 Classification

After completing the searching process and the total database creation, the next step consisted in the classification of papers. The classification phase is a fundamental goal of this thesis, as it aims to contribute to the automation of searching for evidence in clinical research, by trying to reduce the amount of time normally spent in the categorization of the retrieved records [XXII].

The process of classification consists in arranging some input items in groups or categories, following specific criteria. This technique is used to predict a certain output, given different types of inputs. Classification can be binary, if the final categories in which inputs are grouped are only two, or multi-class, if the initial items are then divided into three or more groups. All classification algorithms use a training dataset with some inputs and the known corresponding outputs (labels). The algorithm will be trained on such a dataset to predict the output and to verify if it is correct. Usually, the training phase is done to create a model able to predict the best results, by acting on data and establishing the classification criteria. Then the algorithm is used with another dataset, called validation set, usually smaller than the first one. This phase is useful for adjusting some classification parameters and to have an idea of the final accuracy of the model. The final step is the test, in which completely new data are used and the final output is predicted.

In this thesis, the classification was used to label papers into RCT, SRMA, or Others. PubMed already provide information about the type of classification, but in the next sections it will be demonstrated that this classification is not precise, and the algorithm developed in this thesis results more efficient than the PubMed one. Conversely, Google Scholar does not provide a filter that categorizes papers into their type.

The records classification algorithm proposed in this work consists of the following phases:

- Creation of dictionaries
- Grouping words with the Levenshtein Distance
- Creation of Regular Expressions
- Score computation and classification

2.4.1 Creation of dictionaries

A dictionary is a collection of words used for a particular purpose. Two dictionaries were created manually by collecting the most common words and expressions used for RCTs and SRMA.

A total of 200 SRMA and 200 RCT studies were manually selected, classified, and then read to identify and understand the structure and the lexicon used in the different types of studies. The most used terms and idioms were extrapolated to create a dictionary for the SRMA and RCT outputs. This process was necessary because then all the words in each dictionary will be compared with the title and abstract of each record in the final database, thus resulting in a specific score.

The resulting dictionaries used for RCT and Meta-Analyses are reported in Appendix A in Table 5-1.

2.4.2 Grouping words with Levenshtein distance

The words in dictionaries can be very similar or they can differ for morphological variations, singular/plural, or suffixes. The objective was to obtain a list as general as possible, applicable to a large number of papers, and able to match the largest number of records.

During the classification process, titles and abstracts were compared with the words present in the dictionaries, but to avoid the repetition of very similar expressions during such comparison, the Levenshtein distance was used to create groups of words, that will be compacted in a single statement, thanks to the use of Regex, and then compared to the title and abstract in each record.

The Levenshtein distance is a metric used in informatics to measure the 'distance' between two items. It is used to compare strings and to verify their similarity. From a qualitative point of view, it represents the minimum number of necessary actions (insertions, deletions, or substitutions) to change an expression and make it equal to the comparator. So, if this distance is low, it means that the two items are very similar; otherwise, they are different and lots of changes are needed [XXIII].

A function in Python was used to compute the similarity between words. Given two strings $s1$ and $s2$, the similarity was computed as follows:

$$\left(1 - \frac{dist}{(len(s1) + len(s2))}\right) * 100$$

where $dist$ is the Levenshtein distance and the denominator is the sum of the strings' length. A threshold of > 70 using the Meta-Analysis dictionary, and > 65 using the RCT dictionary was defined, respectively, to reach a similarity. The values were empirically chosen in order to obtain groups of words that can be unified in a single Regex.

If the strings are composed of two or more words, the function executes the following actions: tokenization of sentences, division into individual words, punctuations removal, alphabetically sorting of the tokens, joining of the words to compare the new strings, computation of the distance.

Considering two strings of the RCT dictionary, $s1 = 'randomized comparative trial'$ and $s2 = 'randomised controlled trial'$, they are first tokenized, so divided into individual words as follows:

['randomized', 'controlled', 'trial']

['randomised', 'comparative', 'trial']

The conversion in lower case letters and the punctuation removal is not needed in this case, so they are directly sorted alphabetically as follows:

['controlled', 'randomized', 'trial']

['comparative', 'randomised', 'trial']

Then, the words are reunited with the computation of the distance. To exemplify, a score of 2 is assigned to convert 'randomized' in 'randomised' because 1 deletion and 1 insertion are necessary.

R	A	N	D	O	M	I	Z	E	D
R	A	N	D	O	M	I	S	E	D
							2		

For the words 'controlled' and 'comparative', 2 is assigned for substitutions and 1 for only insertion or deletions.

C	O	N	T		R	O	L	L	E	D
C	O	M	P	A	R	A	T	I	V	E
		2	2	1		2	2	2	1	1

Both strings have the word 'trial' equal so the score for this word is 0. The final Levenshtein distance is 15, and the similarity ratio is 73 (with the formula reported before). This means that the two strings are in the same group because the ratio overcomes the threshold for the RCT dictionary.

2.4.3 Creation of Regular Expressions

A Regular Expression (Regex) consists of a particular sequence of characters used to match combinations of strings. It is utilized for retrieving specific letters or numbers inside a text, including the position, the number of characters repetitions, the beginning or end of words, and so on [XXIV]. Regex can match uppercase or lowercase letters, groups of numbers, new lines, and whatever syntax in a string.

In this thesis, such expressions were used to group in a unique pattern a set of items with similar characters.

An example is reported below, referring to the RCT dictionary:

['randomised trial', 'randomised controlled trial', 'randomized control trial', 'cluster randomized trial', 'randomized controlled trial', 'randomized trial', 'randomized clinical trial']

All these statements are inserted in the RCT dictionary, and this group was created with the use of Levenshtein distance. The Regex to match all the items was:

(?<=randomi.ed).(?=\btrial)*

This expression matches all the strings included between the word 'randomized' and the word 'trial', even if there is only a white space. The '.' in the string 'randomized' indicates a jolly character (in this case it could be 'z' or 's') while the '\b' avoids matching words that contain the substring 'trial' (i.e., atrial). All the defined Regex are reported in Appendix A in Table 5-2 and Table 5-3.

2.4.4 Score computation and classification

Once the Regex were created, the classification process started.

First, the Regex created for the dictionary relevant to SRMA were compared only with the titles, and the positions of records with at least one result were saved. Then all the abstracts were compared both with Regex and the remaining dictionary

words (that were not grouped into a corresponding Regex) with the computation of the score for each abstract.

The score is computed on a range from 0 to 100 in the following way: the mean of the occurrences in the abstract for every Regex or word in the dictionary was computed and considered as the number of minimum words that allows assigning a score of 100%. Then, for each abstract, the computed number of matches was divided for this minimum number and multiplied by 100. So, if the minimum number of words is five, and the abstract has five or more matches with Regex and dictionaries words, the score will be 100. If the matches are only 2, the score will be $2/5 * 100 = 40$.

This score was compared to a threshold value set for SRMA (T_{SRMA}). If the score was above the threshold, the record was classified as SRMA and removed from the total dataset, otherwise, it was maintained.

The process was repeated in the same way for records to be still classified but using the RCT dictionary. Then, the computed score was compared with the threshold value for RCT (T_{RCT}) and classified as such if above the threshold.

In this case, an additional check was performed: when a RCT is registered in ClinicalTrials.gov²³ it has the Trial Registration Number, called NCT code, that uniquely identifies the study. For this reason, the Secondary Source field was analyzed: if it was not null, the paper was considered automatically as RCT by setting the threshold to 100%. If the field was null, the following Regex was searched into the abstract to verify if the NCT code was present:

$'(NCT)\d{8}'$

This expression matches 8 digits after the letters 'NCT' and extracts the code. At the end, all the categorized records were removed from the total dataset and the remaining ones were classified as Others. Figure 2-21 shows the phases of the classification process:

²³ <https://clinicaltrials.gov/>

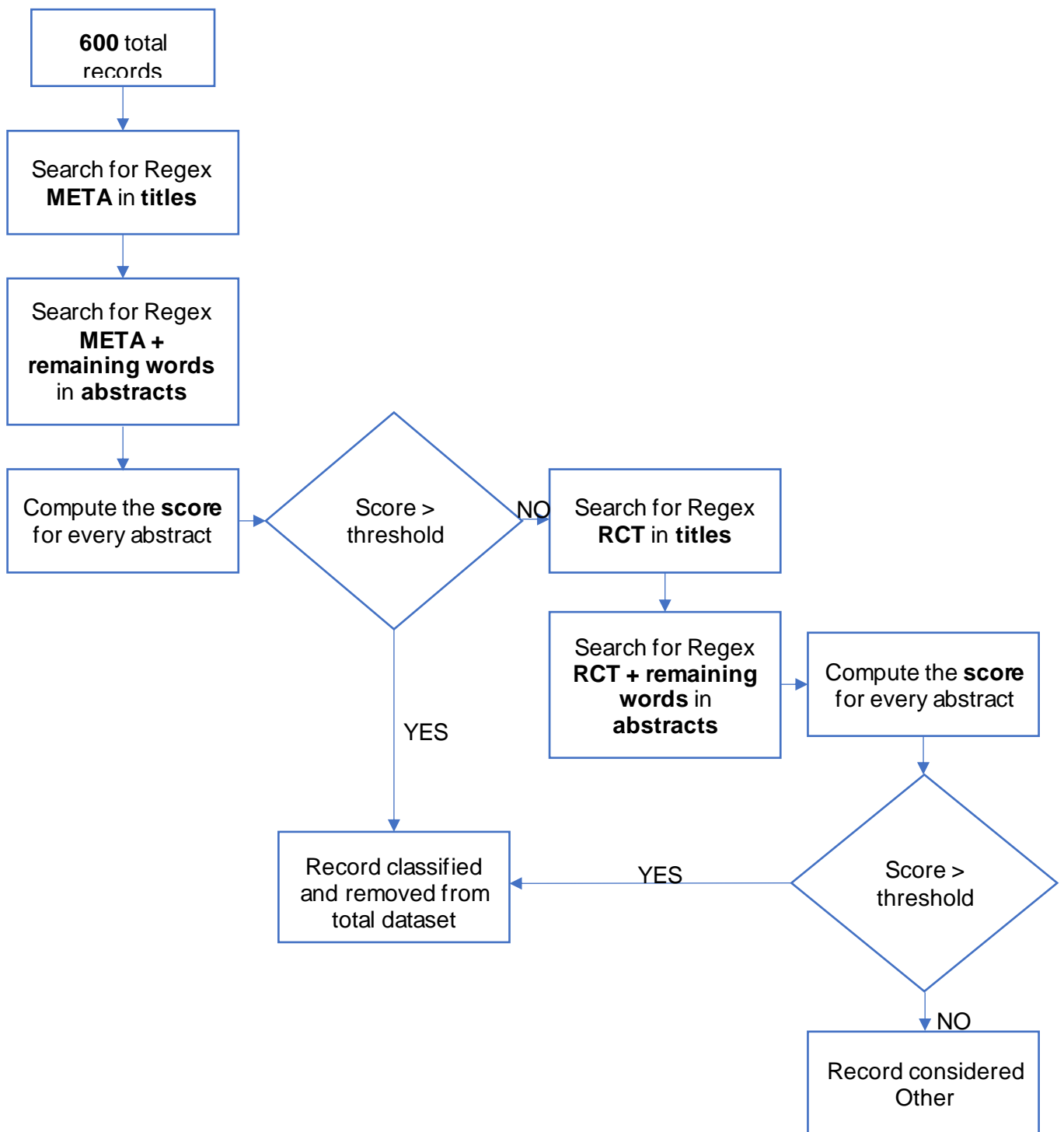


Figure 2-21 - Blocks diagram to explain the classification process

In order to find the best values for T_{SRMA} and T_{RCT} , the following validation protocol was performed, considering 200 records that were manually classified as SRMA, 200 records as RCT, and 200 records as OS.

The algorithm previously described was run iteratively by comparing the scores for each record with the following values:

[0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0]

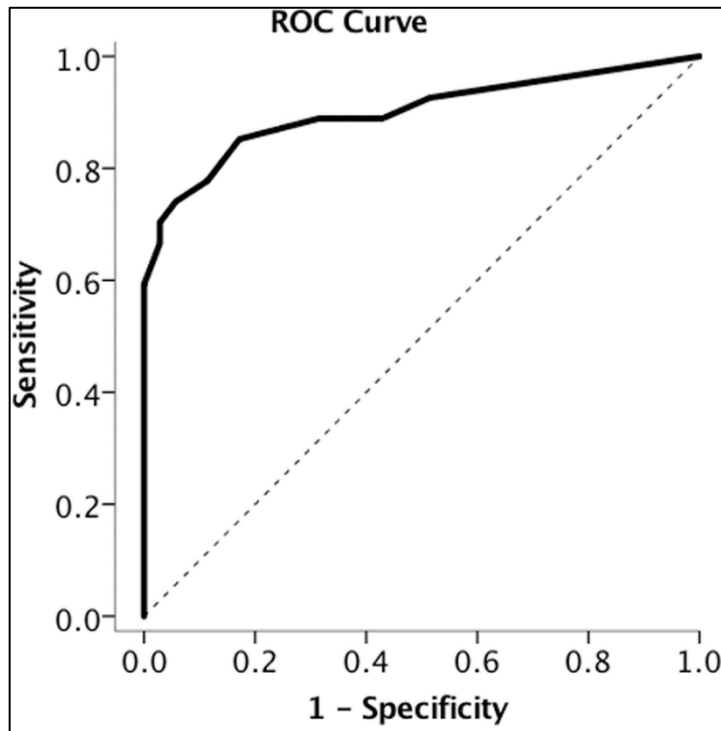
This was done to first determine the best T_{SRMA} and then the best value for T_{RCT} .

For each step, a Confusion Matrix, representing the results of classification as True Positive (TP, items correctly classified as ‘positive’ by the algorithm), True Negative (TN, items correctly classified as ‘negative’ by the algorithm), False Positive (FP, items incorrectly classified as ‘positive’ by the algorithm), False Negative (FN, items incorrectly classified as ‘negative’ by the algorithm) was reported. Figure 2-22 shows an example of a Confusion Matrix in binary classification; for multilabel classification, the number of rows and columns is equal to the number of classes.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 2-22 - Confusion Matrix for Binary Classification.
Source: [XXVI]

A ROC (Receiver Operating Characteristic) curve was generated with on the y-axis the values of True Positive Rate (TPR, Sensitivity) computed for each threshold, and on the x-axis the False Positive Rate values (FPR, 1 – Specificity) [XXV].



$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Figure 2-23 - Example of ROC curve. Source: [XXVII]

Figure 2-23 represents the curve with the AUC (Area Under Curve), that is the area measured below the curve between 0 and 1. If such area is near to 1, it means that the classification is correct, if it is near to 0, there is a higher probability to have incorrect results.

The values of Accuracy and Precision were also computed to evaluate the model performance at the end of the classification process. The Accuracy measures the number of correctly predicted items over the total number of items, while the Precision is the number of TPs over the total items classified as positive. The higher are these values, the better is the classification.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

After having set the best threshold values in the training step, two types of validation were performed: in the first one additional manually selected 100 RCT, 100 SRMA, and 100 OSs were used. In the second one, gold standard RCTs were taken from a freely accessible database, thus analyzing a bigger number of records for this category. The external database²⁴, released from Franck Deroncourt and Ji Young Lee, was downloaded in *.txt* format and consisted of approximately 200,000 RCTs (PMID and abstracts were present); the first 200 PMIDs were extracted from the file and searched with the EFetch Utility in order to retrieve titles and abstract to then compare with the dictionary.

In this phase, during the screening of the records classified as RCTs, it was noticed that a common mistake in FP was including Reviews or OSs in this category. To improve the algorithm performance, once the RCTs were classified, a filtering operation was added to delete records that contained the expressions “review” and “observational study”, thanks to the use of Regex.

2.5 Development of the Web Interface

The last step was the development of a Web Interface, with the aim to integrate all the processes described in the previous sections. Such interface was implemented with the *dash* library in Python (version 3.8), and it was divided into two Tabs.

The first Tab was created to implement the research on PubMed and Google Scholar, giving the possibility to filter the results by date. Through an input box, the user can insert a query string which will be automatically passed into the two search engines. The second Tab displays general information about the created database (total number of records, percentage of papers found in PubMed and Google Scholar, results of the classification), information about the Journals in which the articles were published, the trend in years of papers publication, and a

²⁴ <https://github.com/Franck-Deroncourt/pubmed-rct>

preview of the created database. All the sections will be described in detail at the end of the next chapter.

3. Results

In this chapter the different results obtained, following the same order as in Chapter 2, are presented. Some examples of the generated databases will be illustrated, with a more detailed description of fields and data organization. All the classification results will be explained, and finally, the user interface will be presented.

3.1 Database creation

As already explained, the idea was to keep the database updated on a monthly basis and add only new information if the same string was searched more than once. Figures 3-1 and 3-2 summarize the cases that could occur during a query, both for PubMed and Google Scholar.

Some examples are given below (Tables 3-1 and 3-2 show an extract of the total databases formed by the union of the PubMed database and the Google Scholar one), respectively searching with the string “telemedicine” and “mobile health”, to verify the completeness of the retrieved information and the degree of duplicates found:

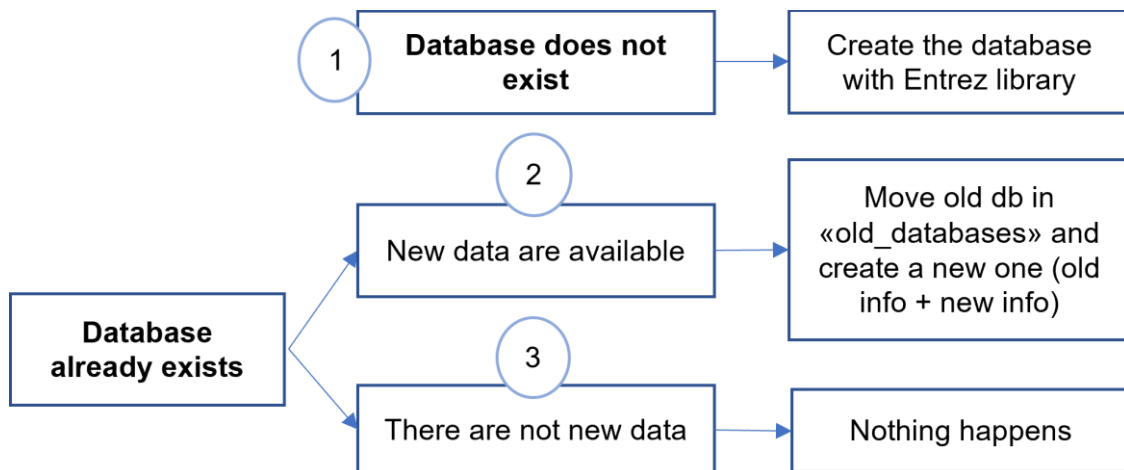


Figure 3-1 - Cases for PubMed database creation

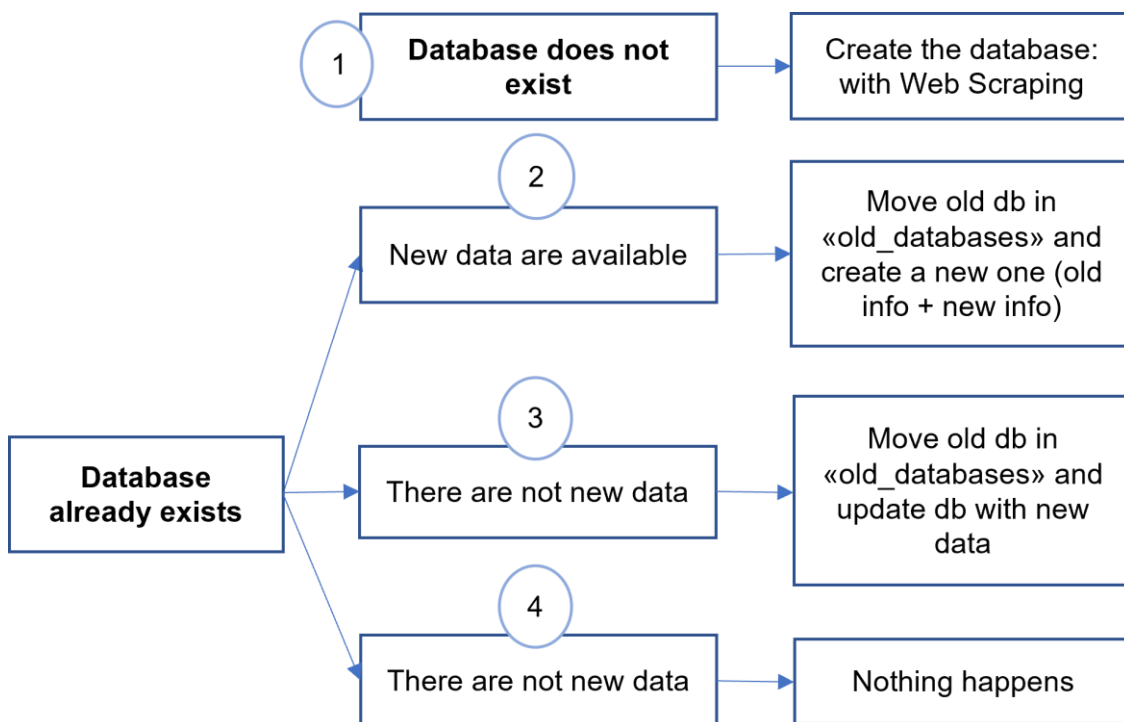


Figure 3-2 - Cases for Google Scholar database creation

- Query “telemedicine”: 510 records were retrieved, 400 from PubMed and 110 from Google Scholar. The research was performed on January 8th, 2022, without using the date filters, so records were automatically downloaded from the most

recent articles published (from January 2020 to January 2022). Subsequently, 41 records were deleted because they were duplicates, thus resulting in 469 items.

Of those, 22 (4.7%) had None values for the DOI field while 49 (10.4%) had None values for the Abstract field. Only for 5/22 records the DOI was not correctly retrieved due to the inefficiency of the techniques described in the previous sections, thus resulting in $5/469 = 1.1\%$ of the created database. The remaining 17 did not have the DOI string written on the web page, so it was not possible to extract such datum from these records.

On the total null abstract values, only 9/49 were not correctly retrieved due to Web Scraping failure, resulting in 1.9% of the created database. The remaining 40 did not have the abstract available.

- Query “mobile health”: 708 records were retrieved, 400 from PubMed and 308 from Google Scholar, of which 46 records were deleted because they were duplicates, thus resulting in 662 items. The research was conducted on the 7th February 2022 without using date filters as before (articles published from January 2020 to February 2022).

Of those, 53 (8%) had None values for the DOI field while 58 (8.7%) had None values for the Abstract field. Only for 6/53 the DOI was not correctly retrieved, resulting in 0.9% of the created database, The remaining records did not have such information on the web page as before. The abstract values were not correctly retrieved in 17/58 records, thus resulting in 2.6% of the created database. The other 41 did not have the abstract.

In the *.json* format, databases were organized as lists of dictionaries, and each paper was reported as in Figure 3-3: all the fields' labels were represented as keys, and the information inside every key were represented as values.

In Tables 3.1 and 3.2, an example of six records in the created databases for “telemedicine” and “mobile health” is shown to appreciate the completeness of the retrieved information. For the fields Title and Abstract, all the relevant text is memorized, but only the beginning is shown in the table.

```
"0": {
  "PMID": 34713028.0,
  "DOI": "10.3389/fdgth.2020.00015",
  "ArticleTitle": "A Smartphone Application Designed to Engage the
Elderly in Home-Based Rehabilitation.",
  "Abstract": "As life expectancy increases, it is imperative that the
elderly take advantage of the benefits of technology to remain active
and independent. Mobile health applications are...",
  "Link": "https://pubmed.ncbi.nlm.nih.gov/34713028/",
  "Authors": "['Androutsou T', 'Kouris I', 'Anastasiou A',
'Pavlopoulos S', 'Mostajeran F', 'Bamiou DE', 'Genna GJ', 'Costafreda
SG', 'Koutsouris D']",
  "PublicationType": "['Journal Article']",
  "PublicationDate": "2020-01-01",
  "Journal": "Frontiers in digital health",
  "ISSN": "2673-253X (Electronic) 2673-253X (Linking)",
  "SecondarySource": null
}
```

Figure 3-3 - Example of record in JSON format

PMID	DOI	Title	Abstract	Link	Authors	Type	Date	Journal	ISSN
	10.1182/blood-2021-152021	The Use of Virtual Care in Patients with Hematologic ...	Background: The use of virtual care, defined as providing ...	https://www.science-direct.com/science/article/pii/S0006497121038969	A Suleman		2021-11-01	Blood	1528-0020
	10.34119/bjhrv4n6-167	Papel da telemedicina em pacientes com ...	Objetivo: Avaliar o papel da telemedicina na vida dos ...	https://www.brazilianjournals.com/ojs/index.php/BJHR/article/view/39800	GD Reis		2021-11-01	Brazilian Journal of Health Review	2595-6825
34901772	10.1002/hbe2.297	Utility of telemedicine in sub-Saharan Africa during	Telemedicine is the use of technology to achieve remote care.	https://pubmed.ncbi.nlm.nih.gov/34901772/	['Chitungo I, Mhango M'...]	['Journal Article', 'Review']	2021-11-02	Human behavior and emerging technologies	2578-1863 (Electronic) 2578-1863 (Linking)
34866028	10.1016/j.jhqr.2021.10.006	E-consultation as a tool for the relationship between ...	INTRODUCTION: Electronic consultation (eConsultation) can ...	https://pubmed.ncbi.nlm.nih.gov/34866028/	['Pavon de Paz I'...]	['English Abstract', 'Journal Article']	2021-11-05	Journal of healthcare quality research	2603-6479 (Electronic) 2603-6479 (Linking)
34903358	10.1016/j.jpedsurg.2021.10.048	Remote treatment of pectus carinatum ...	BACKGROUND/PURPOSE: To report telemedicine's...	https://pubmed.ncbi.nlm.nih.gov/34903358/	['Gigena C, Vincenzo MD'...]	['Journal Article']	2021-11-06	Journal of pediatric surgery	1531-5037 (Electronic) 0022-3468 (Linking)
34864326	10.1016/j.midw.2021.103201	Midwives' perception of advantages of health care at a distance ...	OBJECTIVE: To explore midwives' perceptions of the...	https://pubmed.ncbi.nlm.nih.gov/34864326/	['Gemperle M, Grylka-Baeschlin S'...]	['Journal Article']	2021-11-11	Midwifery	1532-3099 (Electronic) 0266-6138 (Linking)

Table3-1 - Partial "telemedicine" database

PMID	DOI	Title	Abstract	Link	Authors	Type	Date	Journal	ISSN
	10.2196/31097	Exploring the shift in international trends in mobile health research...	Background: Smartphones have become an integral part of our lives...	https://mhealth.jmir.org/2021/9/e31097	J Cao		2021-01-01	JMIR mHealth and uHealth	2291-5222
	10.1080/08039488.2021.1965654	Application of computerized cognitive test...	Background Major depressive disorder (MDD) is a chronic...	https://www.tandfonline.com/doi/abs/10.1080/08039488.2021.1965654	B Cao		2021-01-01	Nordic Journal of Psychiatry	1502-4725
	10.33448/rsd-v10i10.19188	Telemonitorização de sintomas pós quimioterapia...	Abstract The aim of this study was to assess whether there...	https://rsdjournal.org/index.php/rsd/article/view/19188	ETA Moura		2021-01-01	Research, Society and Development	2525-3409
34907785	10.1177/08901171211055317	"Mother's Health and Well-Being Matters: Is a Mediated...	PURPOSE: To test the feasibility of introducing 'Free Time for Wellness' ...	https://pubmed.ncbi.nlm.nih.gov/34907785/	['Jones C, 'Gibbons M'...]	['Journal Article']	2021-12-15	American journal of health promotion : AJHP	2168-6602 (Electronic) 0890-1171 (Linking)
34910541	10.2214/AJR.21.26901	Safeguarding Data Security in the Era of Imaging mHealth.	Mobile health (mHealth) technologies stand poised to...	https://pubmed.ncbi.nlm.nih.gov/34910541/	['Gowda V, 'Cheng G'...]	['Journal Article']	2021-12-15	AJR. American journal of roentgenology	1546-3141 (Electronic) 0361-803X (Linking)
34924318	10.1016/j.pcd.2021.12.005	Technology-based and supervised exercise interventions for ...	AIMS: The purpose of this study was to estimate, for people ...	https://pubmed.ncbi.nlm.nih.gov/34924318/	['Timurtas E, 'Inceer M'...]	['Journal Article']	2021-12-16	Primary care diabetes	1878-0210 (Electronic) 1878-0210 (Linking)

Table 3-2 - Partial "mobile health" database

3.2 Classification

In this section, the results of the classification process, starting from the used datasets during the training and validation phase, up to the final analysis on the total databases, are provided, with some examples to directly explain the obtained results and the comparison with PubMed classification.

3.2.1 Training phase

To verify how many records had at least a match with the words in Regex or generated dictionaries, the titles and abstracts of the manually selected 200 RCT, 200 SRMA, and 200 OS, used as training set, were first compared with the Regex without computing a score but only to verify how many records had at least a match; secondly, the titles and abstracts of papers without any match with Regex, were compared with the other words in the RCT and SRMA dictionaries.

Since the dictionaries were manually created, this preliminary analysis was performed to have a first check of the correctness of such dictionaries. For this reason, the words were compared with the titles and abstracts without computing a score and without verifying if a threshold was exceeded, but only to see if these words were actually present in articles classified as RCT or SRMA.

Figure 3-4 shows the result for SRMA records: 157 papers had a match both in the title and abstract, 30 records only in the title, 13 records only in the abstract. All the records had at least a match with the defined Regex. Figures 3-5 show the result for RCT records: 101 records had matches both in the title and abstract, 8 records only in the title, 74 records only in the abstract. The remaining 17 records were compared with the remaining dictionary words, and only for 5/17 a match was not found: however, in 3/5 the Secondary Source field was present, so they were for sure RCT, and 1/5 had the NCT code written in the abstract. Only 1 record over 200 had neither a match with Regex or words, nor the NCT code (0.5% of the total).

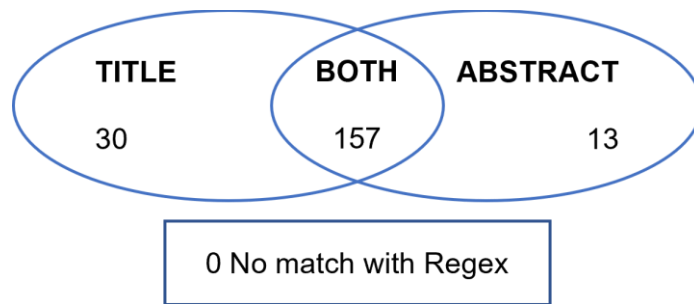


Figure 3-4 - Number of SRMA with Regex match in title (left), match in title and abstract (center), match in abstract (right), no match (bottom)

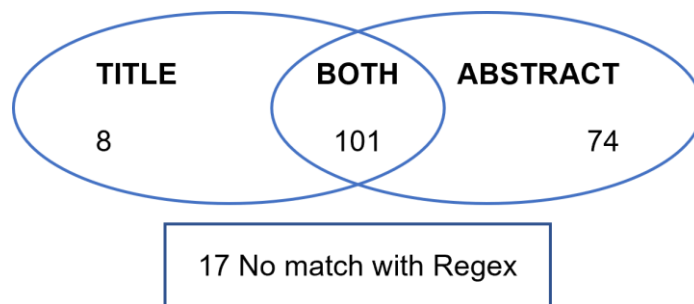


Figure 3-5 - Number of RCT with Regex match in title (left), match in title and abstract (center), match in abstract (right), no match (bottom)

For each record, a score was computed based on the common words found, multiplied by 1.5 for matches found both in the title and in the abstract.

In order to define the best threshold to correctly classify SRMA and RCT, the corresponding T_{SRMA} and T_{RCT} were changed to each of the values defined below:

[0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0]

For each value, the TP, TN, FP, and FN resulting from the classification of the 600 selected records were computed, to construct the Confusion Matrices both for SRMA and RCT.

3.2.1.1 Systematic Reviews and Meta-Analyses vs Others

In this first step, the classification for all the records aimed at classifying first the SRMA was performed, by comparing the computed score with the T_{SRMA} value. The ROC curve was constructed (Table 3-3) to establish the best threshold value (T_{SRMA}).

Threshold	0.0	15.0	30.0	45.0	60.0	75.0	100.0
Sensitivity	1.0	1.0	1.0	1.0	0.96	0.96	0.96
False positive Rate	1.0	0.02	0.02	0.02	0.01	0.01	0.01

Table 3-3 - Values of Sensitivity and FPR for Meta-Analysis (thresholds 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0)

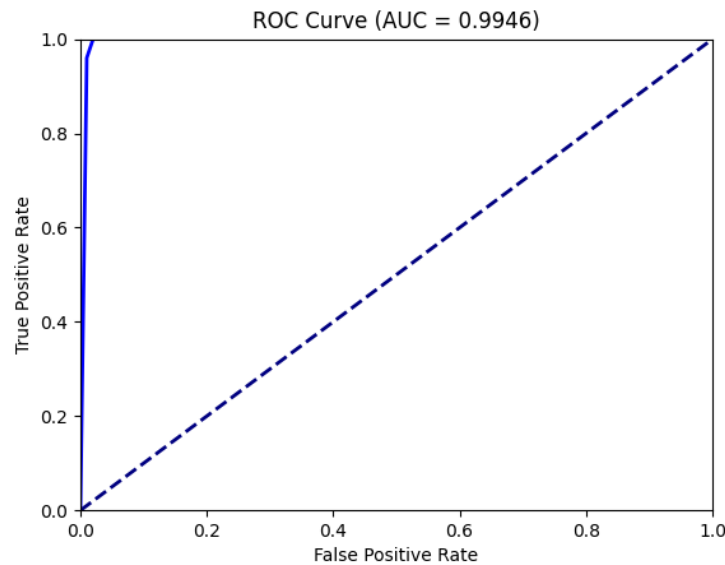


Figure 3-6 - ROC Curve for Meta-Analysis vs Others

Figure 3-6 represents the ROC curve with an AUC close to 1, while Figure 3-7 shows the corresponding matrices for all the threshold values: it is possible to notice that the results were the same for thresholds equal to 15, 30, and 45, with no FN and 9 FP.

An additional comparison was performed to check the correctness of the PubMed classification (retrieved from the PT field) of the 200 Meta-Analyses: 20 records did not have the right PubMed classification, because they were considered as "Journal Article". On the contrary, with the proposed process, they were all correctly classified. As a result of this training, the best T_{SRMA} was set to 30.0.

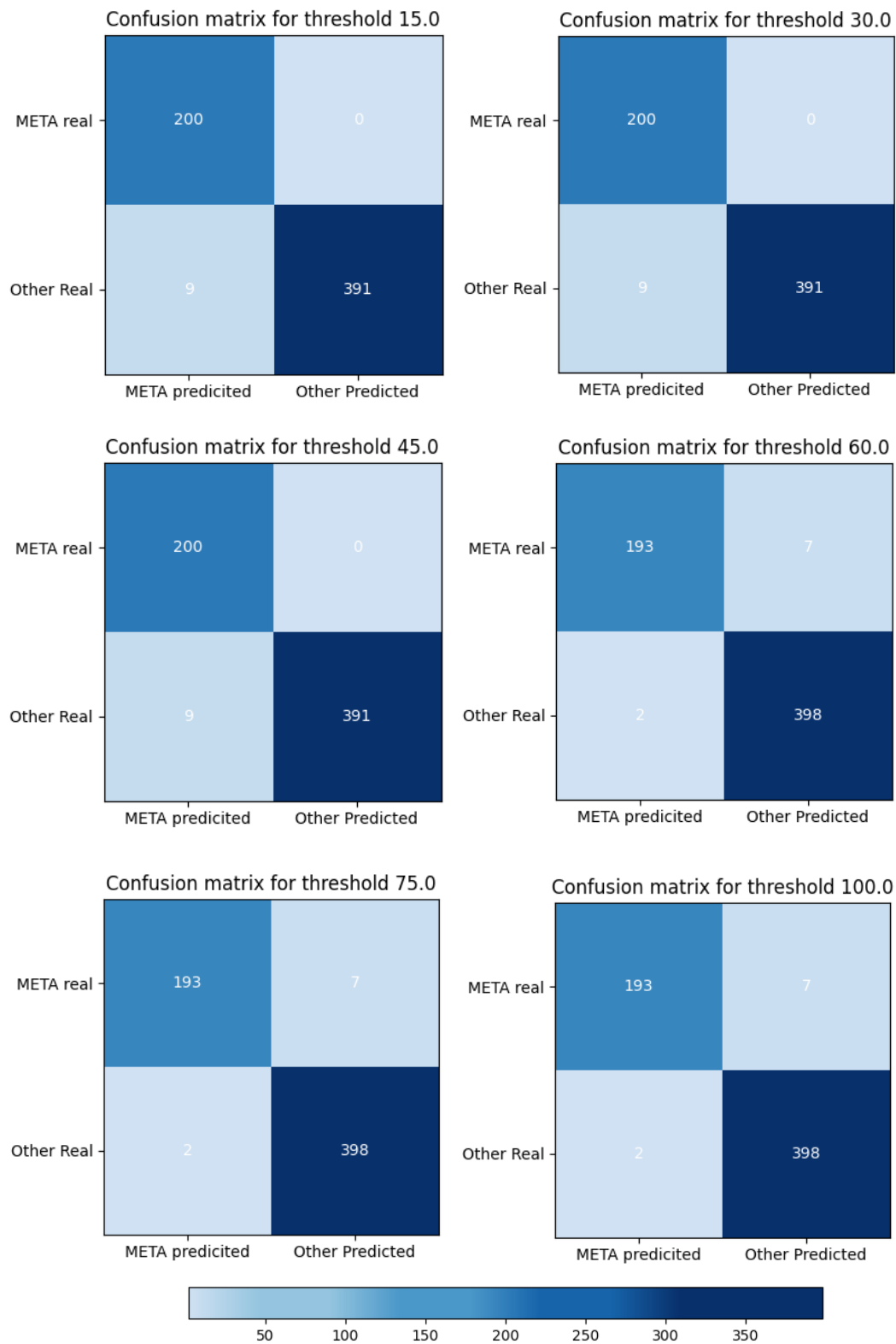


Figure 3-7 - Confusion Matrices of SRMA vs Others (threshold 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0)

3.2.1.2 Randomized Controlled Trials vs Others

After removing the 209 records previously classified as SRMA, the next classification step was performed to compare the score with the T_{RCT} and evaluate its performance. The Sensitivity and FPR values used to construct the ROC curve are reported in Table 3-4.

Threshold	0.0	15.0	30.0	45.0	60.0	75.0	100.0
Sensitivity	1.0	0.97	0.93	0.9	0.83	0.77	0.76
False positive Rate	1.0	0.25	0.15	0.12	0.06	0.04	0.02

Table 3-4 - Values of Sensitivity and FPR for RCT (thresholds 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0)

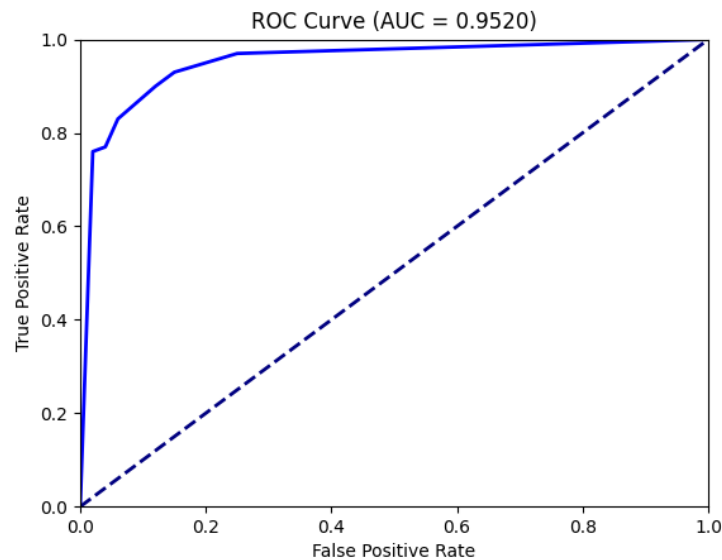


Figure 3-8 - ROC Curve for RCT vs Others

The resulting ROC curve is shown in Figure 3-8, with an AUC equal to 0.9520.

The values of Accuracy and Precision resulted very similar for thresholds 15.0 and 30.0, with the highest number of TP for T_{RCT} set to 15.0, but also the highest number of FP. By increasing T_{RCT} , the FP decreased at expenses of TP, as visible from Figure 3-9. A second analysis was then performed by varying the threshold value from 10.0 to 30.0, with step equal to 1, whose results are shown in Figure 3-10. The Sensitivity and FPR values were constant for T_{RCT} from 10.0 to 16.0, and for values from 17.0 to 30.0.

For $T_{RCT} < 17.0$, the number of RCT correctly classified remained higher, with more FP than for $T_{RCT} \geq 17.0$ (Figure 3-11). The values of Accuracy and Precision (Table 3-6) were higher for a bigger threshold, but since the difference was very small (0.023 for Accuracy and 0.063 for Precision), the final decision was to consider $T_{RCT} = 15.0$ as the best compromise to have the largest number of RCTs correctly classified, at expenses of having more FP (that could be then excluded by manually exploring the results).

The comparison of the classification performance with PubMed classification showed again better results, because 18 studies did not appear as RCT. Conversely, with the proposed approach, and $T_{RCT} = 15.0$, they were correctly identified, as reported in Table 3-5.

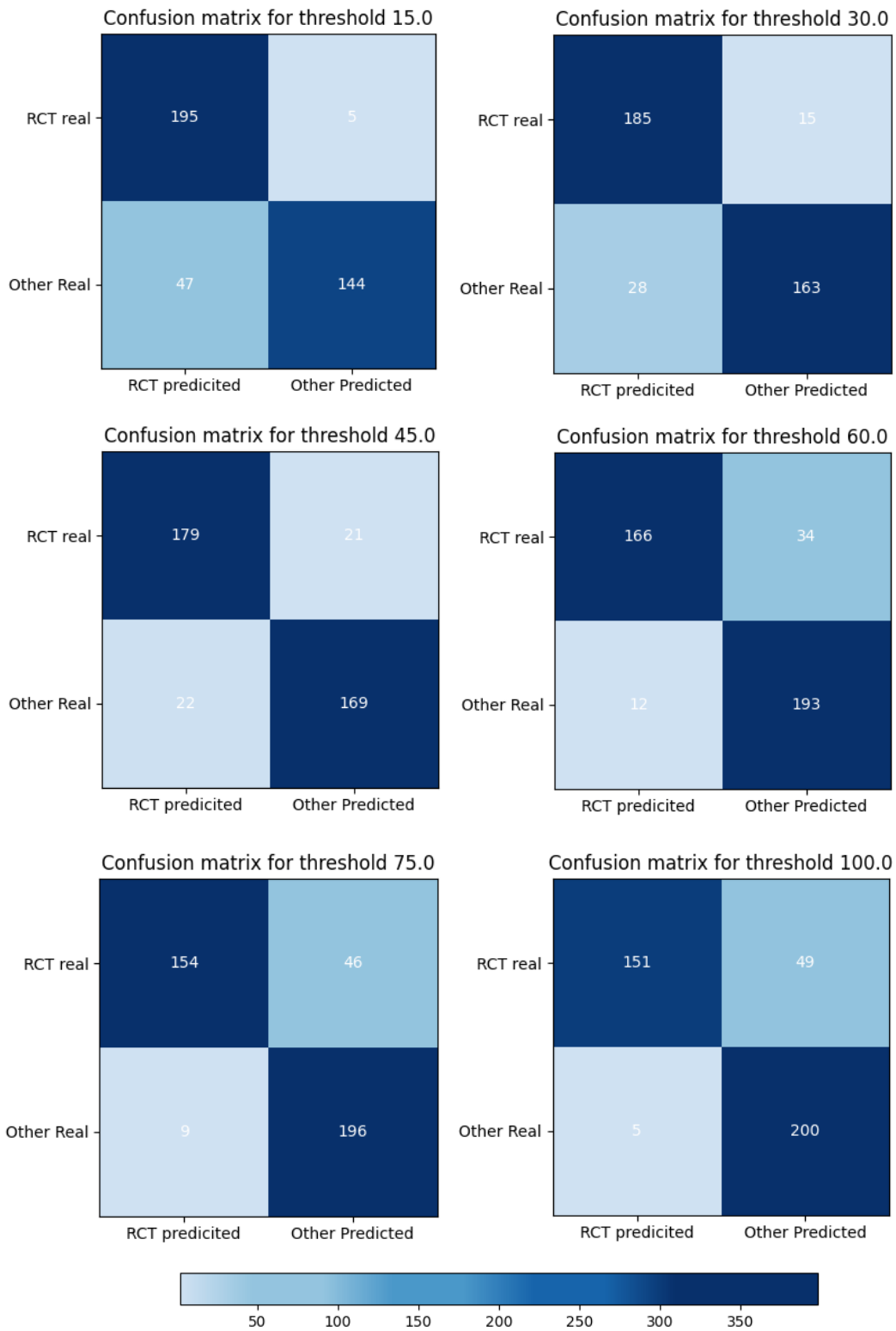


Figure 3-9 - Confusion Matrices of RCT vs Others (threshold 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0)

Threshold	0.0	15.0	30.0	45.0	60.0	75.0	100.0
Accuracy	0.0	0.867	0.89	0.89	0.886	0.864	0.867
Precision	0.0	0.806	0.869	0.891	0.933	0.945	0.968
Correct classification with respect to PubMed	0/18	18/18	17/18	17/18	15/18	13/18	12/18

Table 3-5 - Values of Accuracy, Precision, Number of correct classified RCT with respect to PubMed (thresholds 0.0, 15.0, 30.0, 45.0, 60.0, 75.0, 100.0)

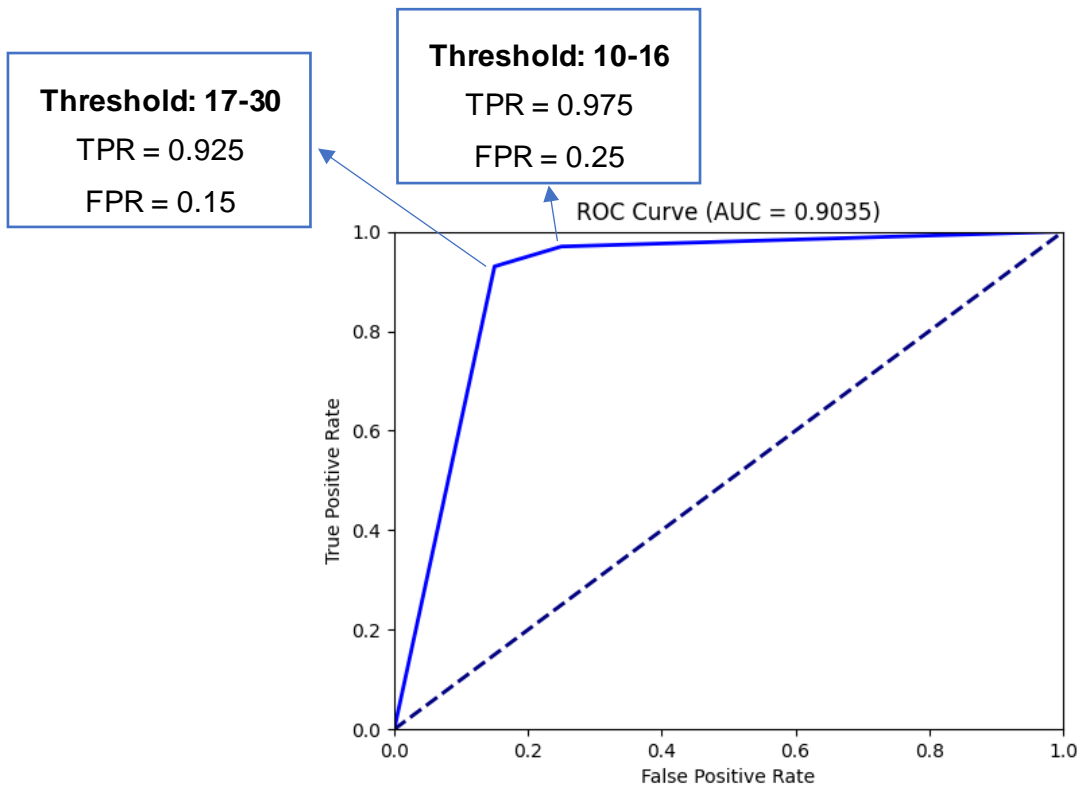


Figure 3-10 - ROC curve for RCT vs Others

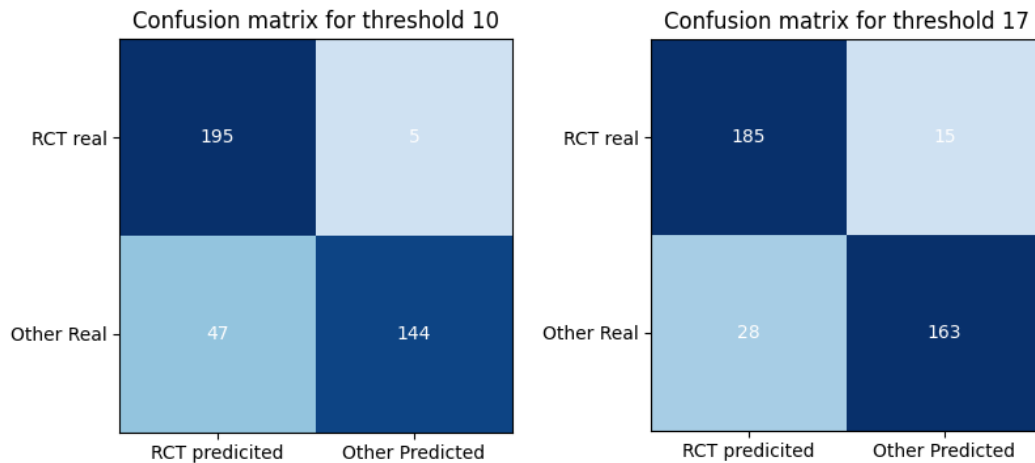


Figure 3-11 - Confusion Matrices of RCT vs Others (thresholds from 0.0 to 30.0)

Threshold	10-16	17-30
True Total	339/391	348/391
True Positive	195/200	185/200
True Negative	144/200	163/200
Accuracy	0.867	0.89
Precision	0.806	0.869

Table 3-6 - Values of Accuracy, Precision, Number of correct classified RCT with respect to PubMed (threshold from 0.0 to 30.0)

3.2.2 Validation phase

3.2.2.1 First Validation (100 vs 100 vs 100)

Figure 3-12 shows the Confusion Matrix that represents the result of the classification as SRMA or Other, using the $T_{SRMA} = 30.0$. All the 100 records manually labelled as SRMA were correctly classified, with only 2 FP and no FN, thus achieving a total Accuracy of 0.993, and a Precision of 0.98.

In comparison to the PubMed classification, 35 records over 100 did not have the correct SRMA label, but they were correctly identified by the proposed approach.

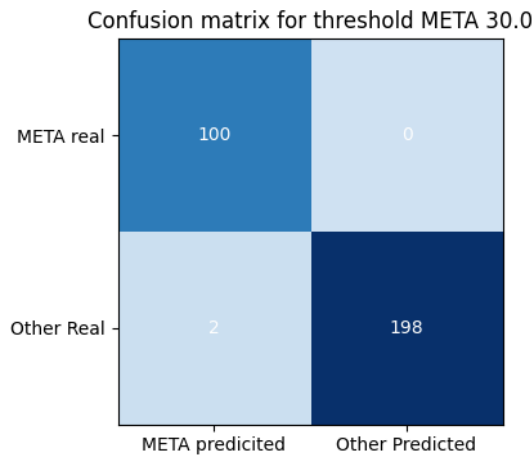


Figure 3-12 - Confusion Matrix of SRMA vs Others using threshold 30.0

Figure 3-13 shows the second Confusion Matrix relevant to the classification of the remaining 198 records as RCT or Other, using the threshold 15.0: 99 records were correctly classified, with 18 FP and no FN, thus achieving a total Accuracy of 0.97, and a Precision of 0.943.

In comparison with PubMed classification, 49 records over the initial 100 did not have the correct label, while 49/49 were correctly identified by the proposed approach.

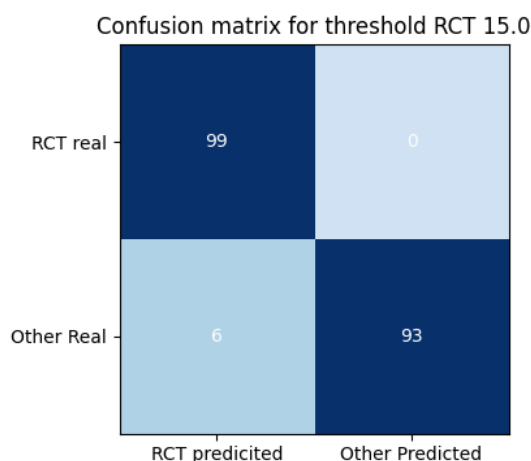


Figure 3-13 - Confusion Matrix of RCTs vs Others using threshold 15.0

3.2.2.2 Validation with database from the Internet (200 vs 200 vs 200)

In this second validation, 200 RCTs whose label was predetermined by the online database, were studied together with 200 SRMA and 200 OS (different from the ones used in the previous phase). All the Meta-Analyses were correctly classified, with 10 FP e no FN, for a total Accuracy of 0.983, and a Precision of 0.952 (Figure 3-14).

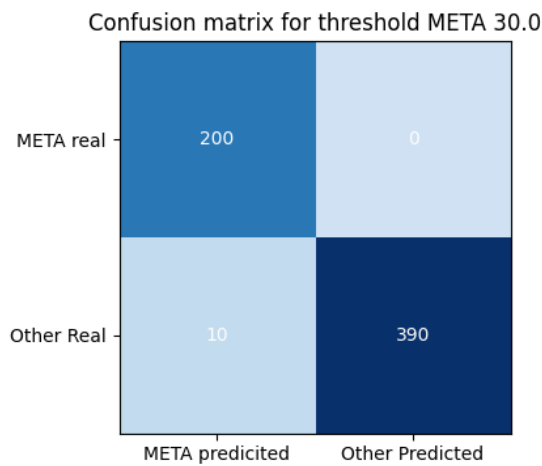


Figure 3-14 - Confusion Matrix of SRMA vs Others using threshold 30.0

For the remaining 390 records, the Confusion Matrix in Figure 3-15 shows the results of the RCTs versus Others classification computed both for T_{RCT} set to 15.0 and to 30.0, to verify the generalizability of the previous settings also when the gold standard was determined by an external entity. From the figure and from Table 3-7 it was possible to notice that the algorithm performance was again better for T_{RCT} set

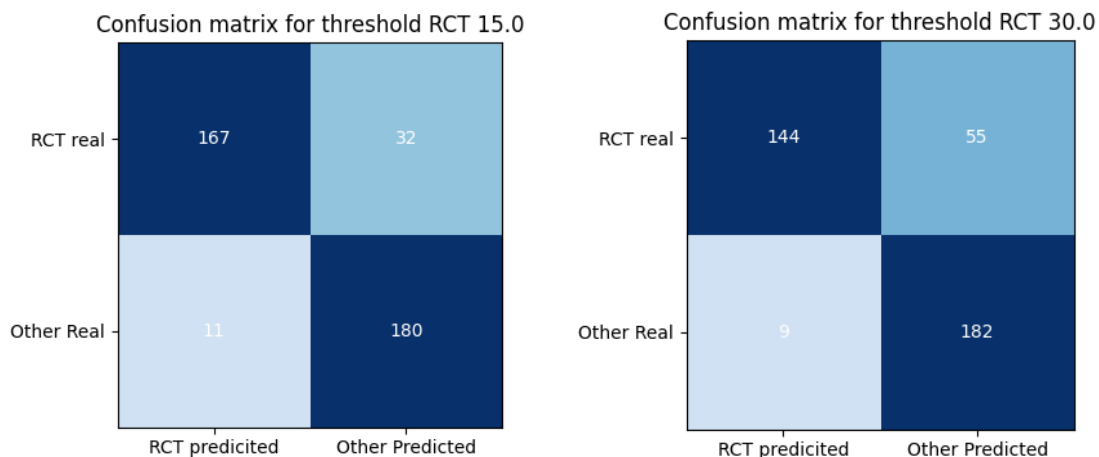


Figure 3-15 - Confusion Matrices of RCT vs Others with thresholds 15.0 and 30.0

to 15.0 in terms of Sensitivity and Accuracy, at expenses of a lower Precision.

Threshold	15.0	30.0
Sensitivity	0.84	0.72
False Positive Rate	0.06	0.05
Accuracy	0.89	0.836
Precision	0.938	0.941

Table3-7 - Values of Sensitivity, FPR, Accuracy, Precision for thresholds 15.0 and 30.0

3.2.3 Test phase

In this phase, three databases were created with the queries “pacemaker”, “artificial pancreas”, and “telemedicine”. In particular, 100 records for each database were randomly extracted and classified, manually screening afterwards their content to verify the correctness of the automated classification.

Figure 3.16a shows the final results of the classification of 100 records taken from the database “pacemaker”: 2 records were correctly classified as SRMA (they had the wrong PubMed classification) and 7 records were correctly classified as RCT (only 2/7 had the correct PubMed classification), with 2 records as FN that were RCTs.

Figure 3.16b shows the final results of the classification of 100 records taken from the database “artificial pancreas”: 1 record was correctly classified as SRMA (with wrong PubMed classification) and 9 records as RCT (4/9 had the correct PubMed classification), no FP and 5 FN.

TP 2 SRMA 7 RCT	FN 2 RCT	TP 1 SRMA 9 RCT	FN 4 RCT 1 META	TP 5 SRMA 8 RCT	FN 1 RCT 2 META
FP 0	TN 89	FP 0	TN 85	FP 2 RCT	TN 82

Figure 3-16 - Confusion Matrices during Test. Figure 3-16a: database "pacemaker". Figure 3-16b: database "artificial pancreas". Figure 3-16c: database "telemedicine".

Figure 3.16c shows the final results of the classification of 100 records taken from the database "telemedicine": 5 records were correctly classified as SRMA (all with the wrong PubMed classification) and 8 records were correctly classified as RCT (all with the wrong PubMed classification). In this case there were 2 FP RCTs and 3 FN.

3.3 Web Interface

This section describes the final Web Interface, through which the user can generate the queries to create the database and then intuitively visualize its content.

3.3.1 Tab1: Web Scraper Tool

The first Tab allows the user to perform the research. Figure 3-17 illustrates the organization of the page:

- At the top, a dropdown menu can be selected to see the already existing databases, relevant to previous queries. By selecting "Old databases", the list of the existing files will be visualized (Figure 3-19) thanks to a direct connection with their local folder. Each item is a '.csv' file named with the query string and the creation date.

- An input box is present to insert the new query, using any alphanumeric character.
- “Filter by dates” allows filtering the research in a specific range of publication years from 2000 to 2022; if the user does not make any selection, the research will start backward from the most recent published article.
- By clicking the button “Search”, the research is performed automatically both in PubMed and Google Scholar, as described in chapter *Materials and Methods*.
- Once the research is completed, a text message appears at the end of the page with the number of results obtained for both websites (Figure 3-18).

The image shows a search interface with a light blue background. At the top, there is a section titled "Please select a folder to see old researches" containing a dropdown menu with the text "Select a folder...". Below this is a section titled "START THE SEARCHING" with a text input field labeled "Insert a string...". Underneath is a section titled "FILTER BY DATES" with a small note: "Optional values: if you do not insert this value the research will start from the most recent article". This section contains two dropdown menus labeled "From" and "To", and a "Search" button. To the right of the main interface, a zoomed-in view of the "From" dropdown menu is shown, displaying a list of years from 2000 to 2005, with 2000 selected and highlighted in light blue.

Figure 3-17 - Organization of first Tab, with a zoom on the 'From' dropdown menu



Figure 3-19 - Dropdown menu of Tab2 clicked

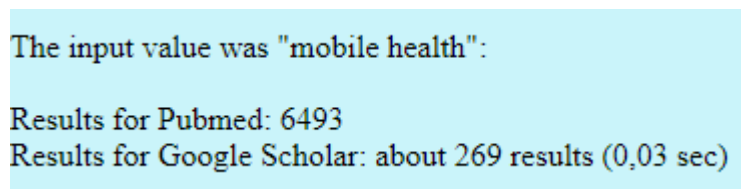


Figure 3-18 - Example of results shown at the bottom of the page

3.3.2 Tab2: Visualization of Results

In the second tab, at the beginning, only a dropdown menu and a button “Update list” are present, as shown in Figure 3-20.

If the user clicks on the dropdown menu, the list of all the already existing databases appears (as in Tab 1), but without the newly created one. By clicking on the “Update list” button, the list is updated with the new file added in the local folder. In this way, it is possible to select both an old database (switching to this Tab without doing any new query) or the new one, and then visualize the related graphs, updated in real-time according to the selected option.

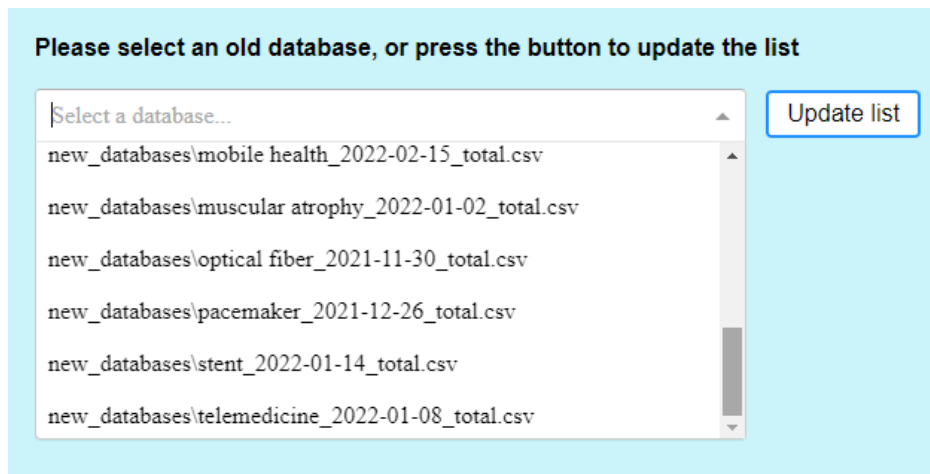


Figure 3-20 - List of old databases after clicking the dropdown menu

The remaining part of the page was divided into four sections:

- Database Overview

After selecting the database, the general information about its content appears. Figure 3-21 shows in order: the total number of included records, a pie plot with the percentage of how many articles were found in PubMed and Google Scholar, and the result of the classification in a pie plot.

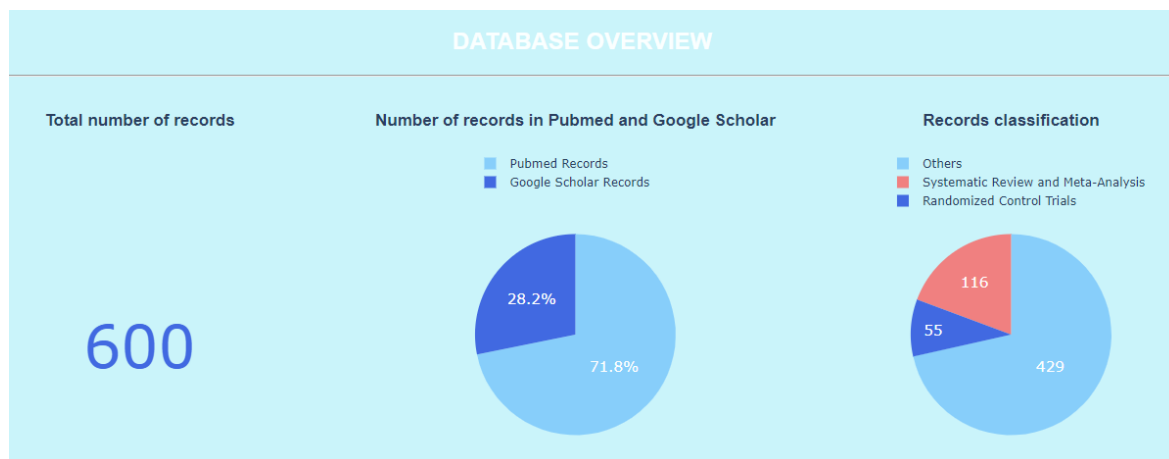


Figure 3-21 - Section 'Database Overview' of Tab2: information taken from database created with the string "atrial fibrillation"

- Information on Journals

This section shows the frequencies of papers published in a specific journal. Figure 3-22 illustrates the list of the Journals contained in the database (left) with the

relative frequency expressed as percentage considering only the items selected (center), and the absolute frequency in the database (right). The Journal items were sorted alphabetically, and only the first fifteen items were selected at the beginning. The pie plot on the center shows the percentage of papers only in the selected Journals (in this case the selected Journals have 20 papers published among them, so the percentage are $2/20*100 = 10\%$ and $1/20*100 = 5\%$), while the bar plot represents the number of records that were published in a particular Journal, whose name is visible by over imposing the mouse.

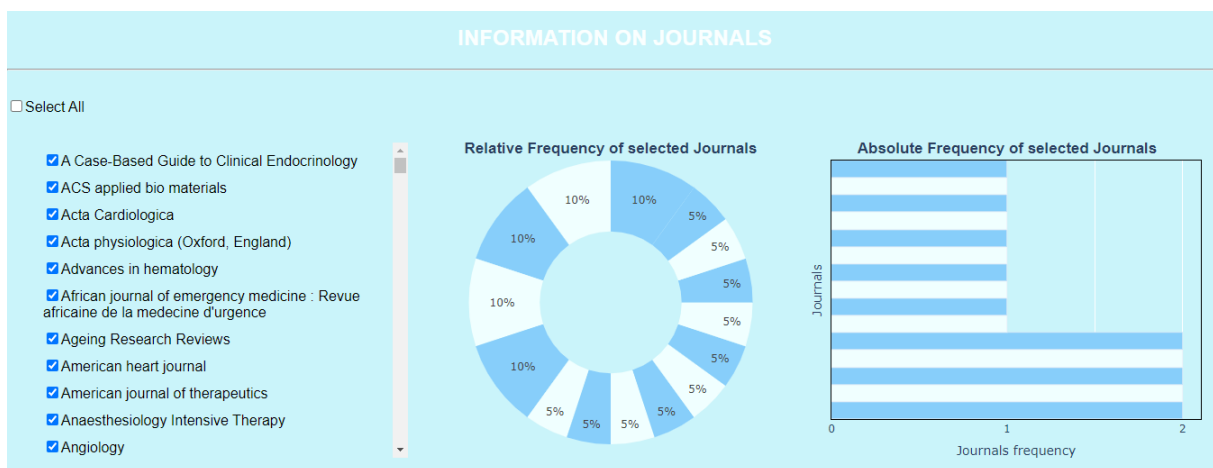


Figure 3-22 - Section 'Information on Journals' of Tab2: information taken from database created with the string "atrial fibrillation"

- Information on Years

The third part consists in a timeline chart, in which the records' publication years were shown (Figure 3-24). On the x-axis, there are the labels indicated as year and month, while on the y-axis there is the corresponding frequency in the database.

Four lines were created to represent singularly SRMA, RCT and Others, in addition to the gray line, representing the frequency for all the records. The items in the legend can be selected or deselected to visualize only the desired lines.

On the bottom, a range slider was inserted to give the possibility to filter the plot within the time desired period. By changing the range on the slider, the plot will automatically update with the selected dates, as shown in Figure 3-23.

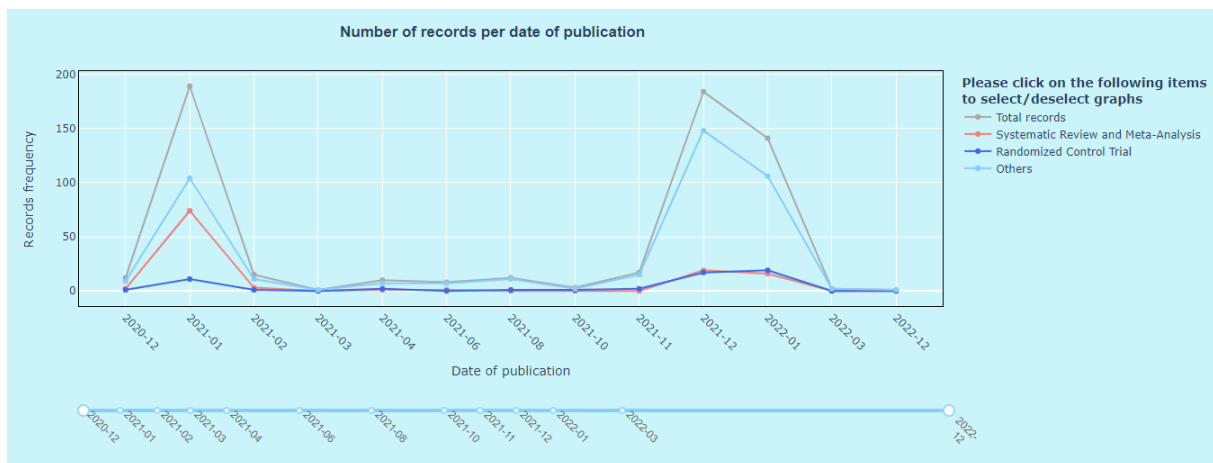


Figure 3-24 - Section 'Information on Years' of Tab2: information taken from database created with the string "atrial fibrillation"

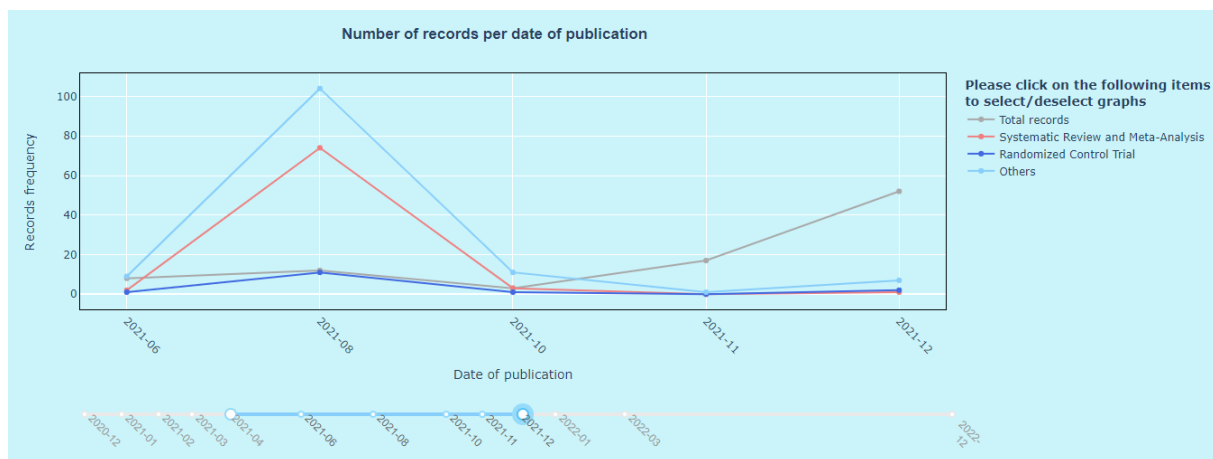


Figure 3-23 - Section 'Information on Years' of Tab2 with the range slider selected: information taken from database created with the string "atrial fibrillation"

- Database visualization and filtering

Figure 3-25 shows the DataTable with the record information present in the database. At the top, there is a checklist that the user can select to visualize only particular types of records in the table below. It is possible to select only one item or more items simultaneously, and the articles will be automatically updated.

The columns with a symbol inside have a direct link: in the column 'DOI', it opens a link composed by the DOI string, while the one in the column 'Link' opens the PubMed link if the record was retrieved from PubMed, or the link extracted through Web Scraping if the record was retrieved from Google Scholar.

Please select the type of record to show in the table

Systematic Review AND Meta-Analysis Meta-Analysis Randomized Controlled Trial Others

Cells are clickable, if you click cells in last column after download ClinicalTrials.gov data you will see all the structured data

DOI	Title	Abstract	Link	Date	Journal	SecondarySource
✓	Diagnostic Utility of Smartwatch Te...	Background: Smartphone technolo...	✓	2021-04	Journal of atrial fibrillation	
✓	Temperature-Controlled Catheter A...	Background: A novel QDOT MICR...	✓	2021-04	Journal of atrial fibrillation	Q QDOT MICRO
✓	Rate Control Versus Rhythm Contr...	Background: Atrial fibrillation (AF) i...	✓	2021-04	Journal of atrial fibrillation	
✓	Adjunctive Vein of Marshall Ethano...	Introduction: Catheter ablation (CA...	✓	2021-06	Journal of atrial fibrillation	
✓	Incidence and Prognostic Impact of...	Background: Corona virus disease ...	✓	2021-08	Journal of atrial fibrillation	
✓	Discharge heart rate and 1-year cli...	BACKGROUND: The association b...	✓	2021-10	Chinese medical journal	Q [ClinicalTrials.gov/NCT02878811]
✓	Robotic-assisted thoracic surgery r...	Background: Our previous study d...	✓	2021-11	Translational lung cancer research	
✓	Anticoagulation Management and ...	BACKGROUND: Therapeutic dose...	✓	2021-11	American journal of therapeutics	
✓	Cardiovascular Burden and Advers...	Objectives: The aim of this study w...	✓	2021-12	JACC. CardioOncology	Q [ClinicalTrials.gov/NCT03619317]

Figure 3-25 - Section 'DataTable' of Tab2: information taken from database created with the string "atrial fibrillation"

The range slider is connected to the DataTable, so if a range is selected only the articles published in the desired period will be visualized. Some cells do not show all the content inside, but if the user, for example, wants to read the Abstract without opening the paper link, he can move the cursor on the corresponding cell to have a preview of the whole text, or directly click the cell, and the content will appear on the bottom of the table.

In the 'Secondary Source' column two information can be visualized: the NCT code or a set of uppercase letters. These letters were extracted from the record Title in case the NCT code was not retrieved, to potentially obtain the acronym of the corresponding RCT. By clicking on the cell with the NCT code, the corresponding link in ClinicalTrials.gov will be opened, while by clicking on the cell with the acronym, the same website will be opened but inserting such string in the URL, to find all the studies with that acronym.

At the bottom of the DataTable two input boxes with buttons (Figure 3-26) allow the user to download the database (applying the filtering options selected) after typing whatever filename in the input box, or downloading all the RCTs that have a NCT code into another database: in this case, additional information are retrieved from ClinicalTrials.gov: *Official title, Study type, Allocation, Intervention model, Primary purpose, Masking, Enrollment, Condition, Minimum Age, Maximum Age, Gender, Healthy Volunteers, Phase, Primary Outcome, Secondary Outcome, Number of arms, Intervention Name, Intervention Type, Arm Group Type.*

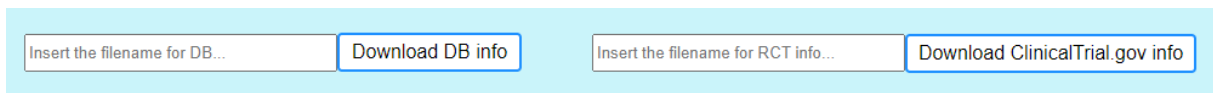


Figure 3-26 - Input boxes and buttons at the bottom of the DataTable

After downloading all these info, if the user clicks again on the cell with the NCT code, he could visualize the structured information as shown in Figure 3-27.

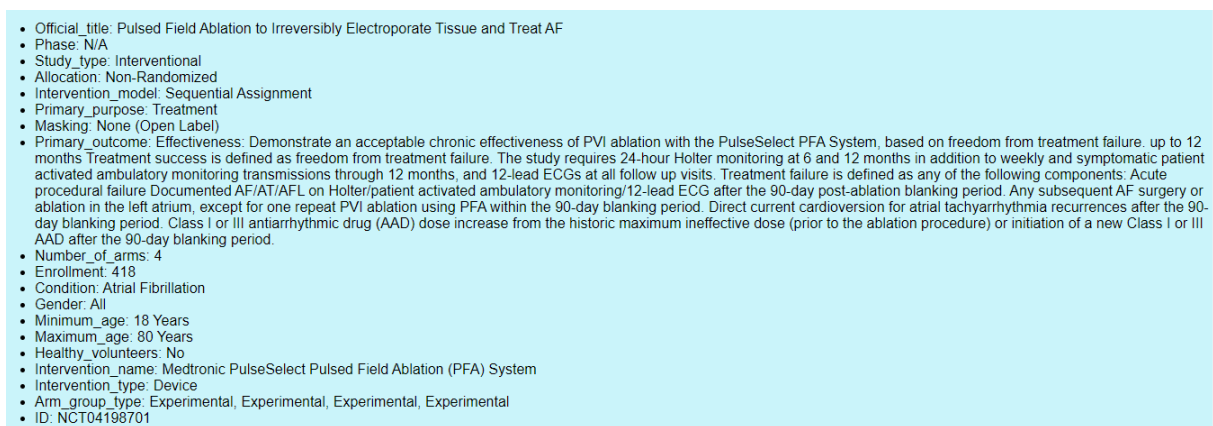


Figure 3-27 - Example of structured data after clicking the 'Secodnary Source' cell

4. Discussion and Conclusion

This project aimed to build the basis for a framework able to unify the necessary phases for conducting clinical literature research, from the automated retrieval of the information from websites, to the organization of the results, up to the final visualization for users, including the classification of the reported studies into SRMA, RCT, or Other, in order to evaluate the level of clinical evidence, to be applied to queries referring to a specific class of devices, a type of disease, or whatever topic relevant to the biomedical research field.

To achieve this purpose, *Python* programming language was used to implement the entire platform. The steps involved: 1) the creation of a database formed by scientific papers; 2) the classification of papers according to the type of study; 3) the visualization of the search results and study characteristics through a Web Interface.

4.1 Database creation

Among all the possible databases used for clinical research, only PubMed and Google Scholar were chosen for retrieving information. This choice was made because PubMed is one of the most important online resources providing clinical data and a simple search engine, called Entrez, is provided to extract information from the Internet and export them in local folders through code. Entrez allows searching in 38 different databases and this aspect guarantees a lot of flexibility and

gives the possibility to choose what kind of research the user intends to perform, from a specific and focused topic to a more general one. In this project, Entrez was set to search only in PubMed or PubMed Central, because the other databases have a search strategy completely different from the one implemented. Google Scholar, instead, was chosen because it has free access concerning other databases, and it is considered a valid search engine for academic research, because it contains information about different topics and fields. In the Google Scholar webpages, papers are organized in a structured way, and this made easier the retrieval of data through Web Scraping techniques.

At the end of the research implemented in both websites, the final database was created by concatenating the retrieved information, with the final duplicates removal. Since PubMed and Google Scholar are both used to collect papers about biomedical literature, they have differences but also similarities: they display results in a different order, they provide different filters for searching, but sometimes results can still be equal, and for this reason, it was necessary to remove duplicate records from the total database. The DOI and the abstract represented the most important fields for this work, because the first one was necessary for duplicates removal (as explained in Section 2.3 the DOI string was used as a comparator between records if available, otherwise the title was used) while the abstract was fundamental for the classification process; for this reason, during the development of the work, lots of adjustments were made to maximize the algorithm efficiency for their extraction. For records downloaded from Google Scholar, abstracts and DOIs were retrieved first through HTML pages, and for missing values, the CrossRef API was used.

The total database was saved into *.csv* format and *.json* format after its creation, to give the possibility to visualize it in the way the user prefers. All the records were ordered chronologically, according to the publication date, from the oldest to the newest, to have a clear overview of the downloaded papers, to easily identify a particular record if the publication date is known, but also for the database updating process: as described in Chapter 2, during the creation phase, there was the computation of difference in days between the date of database creation and the date of the most recent record, to count if 30 days passed. The chronological order allows the extraction of this data simply from the "Publication Date" field of the last line in the database.

One of the strengths of this phase was represented by the adjustments made on the query string inserted by the user to start the searching process: if such string was composed of two or more words, double quotation marks were added in order to search on PubMed and Google Scholar the exact expression written by the user, to maximize the retrieval of relevant articles and to avoid downloading information that could be irrelevant.

Another advantage of the implemented process for database creation is the possibility to have structured information inside a file and simply look for a specific datum of a record. Other existing platforms based on web interfaces do not give the possibility to download a complete database with records from different resources, but they allow only the download of text files for offline use [9] or the analysis of the record directly on the interface, without giving the possibility to build locally a structured database [10].

4.2 Classification algorithm

The classification process is one of the most important steps performed in clinical literature research. It helps researchers to categorize papers and divide them according to specific criteria. Usually, the classification is based on the type of study, reflecting different levels of clinical evidence: this process is often done manually, due to the intrinsic difficulty to automatically classify the proper type for a publication, as there are many ways of writing an article and reporting its information. The PT tag provided by PubMed offers a support in providing a paper category, but in many cases, information is not precise, as shown in our tests. Moreover, the tag is manually applied by PubMed staff around 250 days after the article publication, so recent articles may not yet be indexed [15]. Google Scholar, instead, does not classify articles per type of study.

For these reasons, an automatic classification was implemented without using ML techniques, as many studies already performed have done, but analyzing the titles and abstracts lexicon through strings comparison, to create a simple algorithm able

to categorize articles as RCT, SRMA or Other, that can also be applied to databases provided by external sources including titles and abstracts.

The string comparison was performed between records and two manually created dictionaries, one for categorizing RCTs and the other one for SRMA. The scope of such dictionaries was to obtain a sort of collection of the most common words and expressions used in such types of studies. To do so, each downloaded paper was analyzed to verify if the dictionaries items were present in the title or abstract and to eventually label the paper in a specific category. In particular, a score was computed for every record according to the number of occurrences found and compared with a threshold to verify if the article could be inserted in a category or not.

The total workflow was constructed to first classify the SRMA, removing them from the total dataset after classification, and then the RCTs. The not identified records will be considered as other types of studies. This strategy allowed the reduction of misclassified items, because it was noticed that in many Meta-Analyses papers there are lots of terms belonging to the RCT dictionary, so an approach aiming at classifying simultaneously SRMA and RCT would result in many classification errors.

In this work, during the training phase, the algorithm developed for the classification was iterated among different values, in order to choose the threshold (that is compared with the score computed for each record) that allows obtaining the best results in terms of accuracy and sensitivity of the classification. At the end, T_{SRMA} was set to 30, and T_{RCT} was set to 15 because it was demonstrated that such values were the best for recognizing the maximum number of TP and for considering less FN. Another reason for choosing these values was due to the comparison with PubMed classification: among all the selected papers, the records with a wrong PT tag were all correctly identified with the implemented algorithm and with $T_{SRMA} = 30$ and $T_{RCT} = 15$. The thresholds choice was one of the most important points during the development of the algorithm since all the final results depended on it. As regards the SRMA classification, it was easy to recognize that setting $T_{SRMA} = 30$ gave the maximum number of TP and TN, while the choice of T_{RCT} was more challenging, because it was necessary to carry out a more detailed analysis, choosing two different ranges of values. At the end, it was demonstrated that the value 15 was the best for obtaining a large number of TP and for

minimizing the number of FN. For this reason, it was decided to accept a value such that to certainly have FP, but to avoid losing items that are RCTs but not properly recognized. In any case, at the end of the classification, it would be possible to manually analyze the obtained list of RCTs to exclude wrong items, which is much simpler than analyzing all the rejected articles that constitute the majority of the records.

During the validation phases, the accuracy values were higher than 90%, both for SRMA and RCT, and it was demonstrated that the implemented process achieved better results than the PubMed classification also in this phase. Optimal results were reached in the first validation because 35 SRMA and 49 RCTs not correctly classified by PubMed were all correctly identified by the developed classification procedure.

In the test phase, three examples of databases were selected (created with the strings “pacemaker”, “artificial pancreas”, and “telemedicine” respectively), and 100 records from each of them were selected and manually screened to verify the algorithm correctness. For every database, the accuracy was higher than 90%, proving that the classification process can be also applied to real data downloaded from the Internet without a manual selection.

The choice of basing the classification process on strings comparison was made to speed up the categorization of papers still obtaining good results: the already developed frameworks presented in Chapter 1 used ML approaches (SVM or CNN) to classify articles, but it was demonstrated that some records cannot be used as input for such algorithms, because they do not contain enough information, and this involves more work because some items need to be screened manually to complete the classification [16].

It is necessary to underline that methods based on ML techniques achieved very high results, but using different evaluation metrics. Marshall et al. [15] used the AUROC to demonstrate the validity of the research, obtaining a value of 0.987, while Thomas et al. [16] used the *recall*, arriving to a value of 93.8% on the final analysis on a database of 58.283 studies. Bulla et al [18], instead, used NLP techniques to extract PICO information from papers obtaining an accuracy of the classifier of 76%. Besides these optimal results, existing studies concentrated only on the development of a classification process, focusing exclusively on the extraction of a type of study from a collection of scientific articles, to speed up a process that

normally requires a lot of time and resources. Furthermore, they focused only on the categorization of articles in RCT or non-RCT [15], from the analysis of the text in the entire document downloaded locally [14]. This means that to properly operate those tools it is not enough to have a database with only the most important data (title and abstract) from a paper, but there is the need to have all the desired records entirely downloaded. The strength of this project is the possibility to classify scientific articles in RCT and SRMA (so two different types of studies), starting from a database where only the text in the title and the abstract represents the input parameter for the classification function.

In addition, the classification process proposed in this work represents only a part of the entire framework; in fact, the results were integrated into the final interface, to connect all the various phases of the project and present a complete workflow to the user. This aspect allows to perfectly integrate the classification process within the project, which aims not only to categorize the articles, but to automatize all the steps necessary for the clinical literature research, and to provide a complete framework that could support both professionals and researchers.

4.3 Web Interface

The Web Interface was created to provide an intuitive way of using the entire framework. Two tabs were created to make it user-friendly: the first one implements the searching process (with the possibility to filter the research by dates) and allows the user to create the database, and the second one displays all the obtained results, with the possibility to visualize either the newly created database, or one of the existing ones.

One of the strengths of this interface is the simplicity of the organization: the two tabs were divided to make the search process faster and more efficient, and all the elements inserted within the pages were clearly described in order to be used in an intuitive way. In fact, in the first tab, there is only the input box in which the user can enter the search string, and the two dropdowns to select the desired dates: in

this way, it is very easy to understand the interface functionalities, even opening it for the first time. On the second page, however, the results appear only if the user selects the database on the dropdown, in order to make the presentation clear and not overlap the information.

Another advantage is the ability to view the newly created database at the end of the second page, with the most important information, such as the title, abstract, or reference link. This certainly allows to have an idea of the information just downloaded, and to view only the most interesting data thanks to the features that have been implemented. As already mentioned above, this work does not involve the download of the articles in PDF locally, but only the insertion of some fields within a database; however, thanks to the interface, the user can directly access the link of a particular article, and eventually download it to read the entire text. This aspect represents a plus compared to other developed platforms, which do not allow to have a preview of the articles downloaded, but only to see the number of results in the various databases [11], or a list of articles without organizing the information in a structured way [10].

Another interesting feature is the possibility to have direct access to the *ClinicalTrials.gov* website for RCTs. As already mentioned in previous chapters, when a paper had the NCT code within the abstract, it was directly recognized as RCT, because it is officially registered as such. For this reason, for all the records with an NCT code, there is a direct link that accesses that trial page, giving the possibility to read more specific information about the study. In addition, the button "Download ClinicalTrials.gov info" gives the opportunity to download information taken from this website in another database, to help the user organizing data in local folders and to well-order them for further analyses. Furthermore, when the record is classified as RCT but does not have the NCT code, a function was created to try to extract the acronym of the study from the title, and to search on *ClinicalTrials.gov* such acronym. This implemented feature represents a very innovative point among all the available platforms, thus providing a direct connection with this very important clinical trial website, allowing direct access to specific details of the trial.

4.4 Limitations

Despite the good results obtained both in the database creation phase and during the classification, this work has some limitations that need to be discussed.

First, the online resources used to search and download articles from the Internet were only two, while in a normal process of clinical literature research more different archives containing biomedical literature are queried, thus possibly resulting in a non-completeness of the obtained results. However, this choice was made as it was too difficult to implement a search strategy that could be used for all databases, given that MEDLINE, Embase, or Scopus have different characteristics, and also some access problems that precluded the easy web scraping of their content. The addition of other databases will constitute the work for further development of this thesis.

Regarding the data downloading process, the main limit to be faced was the computational time necessary to extract all the information from the Internet, and to create the final database. It was estimated that on a personal computer (Processor Intel Core i5, RAM 8GB and Wi-Fi connection) about 10 minutes were required to download around 400 articles (depending on the connection stability), so creating larger databases could probably take hours.

In addition, there are still less than 10% of the total records retrieved in which the abstracts and DOI automatically extracted from Google Scholar from the single HTML pages could be missing. In these cases, a manual identification of the correct tag with this information should be performed to avoid null values in the database.

About the classification, the results obtained were very good, the only limitation could be that the articles considered both in the training, validation, and test phase are few to affirm that the algorithm can classify papers in any database with a level of accuracy higher than 90%.

4.5 Future Developments

One of the improvements that can be performed for this project could be the integration of the searching process in other databases. Since the framework has been developed in separate blocks for PubMed and Google Scholar, that are only at the end unified to create a complete database, other scripts could be developed to extract information from different resources, using different Web Scraping techniques, to obtain a more complete number of collected papers.

The classification process could be improved by adding new words to dictionaries or creating new dictionaries. In this way, the classification could include not only SRMA and RCTs but also other types of studies, allowing the division of articles into more categories.

The graphical interface could be evaluated through a usability study proposed to groups of final users with different background (more clinical or more biomedical) to collect different inputs to improve it and make it more appealing, also including additional features and information to be displayed.

It is important to underline that all the phases were created separately so any improvements can be simply added to the entire framework or substitute an already existing process, in a modular way.

4.6 Conclusion

This project was developed to provide a framework for automating scientific literature review, to support researchers in retrieving information on the Internet in clinical literature research. The tool was implemented with the *Python* programming language, using different Web Scraping techniques, to reduce the manual work and make all the phases automatic. The presented results show the different functionalities implemented during the work, highlighting the advantages, and recognizing the limitations. The final implementation represents certainly a good

starting point to develop a more powerful platform that could be very useful for healthcare professionals and researchers, as well as for supporting assessment of regulatory requirements, for example in the medical device context.

Appendix A

Table 0-1 - RCT dictionary and Meta-Analysis dictionary

RCT dictionary	Meta-Analysis dictionary
Randomized clinical trial, Treatment group, Standard group, targeted group, Controlled randomized, double-blinded, Blinded, intervention group, Controlled trial, Random sample, randomized study, Randomized controlled study, Control group, control groups, randomized control-group, Randomly assigned, Randomized Controlled Trial, Randomised controlled trial, Clinical Trial, Clinical Trial, Phase I, Clinical Trial, Phase II, Clinical Trial, Phase III, Clinical Trial, Phase IV, Clinical Trial Protocol, Clinical Trial, Veterinary, Placebo, Multicentre Study, multicenter trial, randomized multicenter study, Equivalence Trial, Controlled Clinical Trial, Randomized Control Trial, Randomization, Cluster randomized trial, Randomized controlled, random control trial, Scientific Experiment, Control Trial, Randomized Trial, randomised trial,	engagement index Cross-classification analysis pooled SMD 4-level Kirkpatrick model metaprop random effects analysis metaprop fix effect analysis meta-analytic review review of published studies meta-ethnography meta-analysis Meta Syntheses Meta Description meta-regression meta regression Meta-Analyses Meta-Analytic Meta Analyses

Randomized Equivalence Trial, Clinical Trials, Controlled Clinical Trials, Control Function, Randomized Comparative Trial, Randomized Comparison, Randomized Experiment, Pilot Trial, allocation concealment, controlled study, active-controlled trial, randomized prospective trial, placebo-controlled trial, randomly divided, randomly chosen, stochastically divided	Meta Analytic Meta-Description Meta-Syntheses Meta-Evaluation Meta Evaluation Meta Analysis, qualitative meta-analysis, qualitative meta analysis, Hypothesis test, Statistical test, evaluation study, statistical methods, synthesize data, evaluate data, forest plot, cumulative meta-analysis, funnel plot, PRISMA
--	--

Table 0-2 - Regular Expressions for RCT dictionary

Group of words	Regular Expression
['randomized comparative trial', 'randomised controlled trial', 'randomized controlled trial', 'randomised trial', 'randomized control trial', 'cluster randomized trial ', 'randomized prospective trial', 'randomized equivalence trial', 'randomized trial']	(?<=randomi.ed).*?(?=\btrial)
['randomized study ', 'randomized controlled study', 'randomized multicenter study']	(?<=randomi.ed).*?(?=\bstudy)
['clinical trial, phase i', 'clinical trial, phase ii', 'clinical trial, phase iii', 'clinical trial, phase iv', 'clinical trial', 'clinical trials', 'controlled clinical trial', 'clinical trial protocol']	(clinical \btrial)

['placebo-controlled trial', 'controlled trial', 'random control trial', 'control trial', 'active-controlled trial']	(\bcontrol\S* \btrial)
['control groups', 'randomized control-group', 'control group']	(\bcontrol\S* group)
['intervention group', 'intervention groups']	(intervention group)
['randomized controlled', 'controlled randomized']	^(?=.*\bcontrol\S*\b)(?=.*\brandomized\b).*\$

Table 0-3 - Regular Expressions for Meta-Analysis dictionary

Group of words	Regular Expression
['meta analytic', 'meta analysis', 'meta-analysis', 'meta-analyses', 'meta analyses', 'meta-analytic']	
['qualitative meta analysis', 'qualitative meta-analysis']	(?<=\bmeta\b).*(?=\banaly\S*)
['cumulative meta-analysis', 'cumulative meta analysis']	
['meta-description', 'meta description']	(?<=\bmeta\b).*(?=\bdescription)
['meta-regression', 'meta regression']	(?<=\bmeta\b).*(?=\bregression)
['synthesize data', 'meta-syntheses', 'meta syntheses']	(?<=\bmeta\b).*(?=\bsynth)
['meta evaluation', 'evaluation study', 'meta-evaluation']	(?<=\bmeta\b).*(?=\bevaluation)

List of Abbreviations

AI Artificial Intelligence

API Application Program Interface

AUC Area Under Curve

CEP Clinical Evaluation Plan

CER Clinical Evaluation Report

CNN Convolutional Neural Network

DOI Digital Object Identifier

FDA Food and Drug Administration

FN False Negative

FP False Positive

FPR False Positive Rate

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

ICASR International Collaboration for the Automation of Systematic Reviews

ISSN International Standard Serial Number

JSON JavaScript Object Notation

MDD Medical Device Directive

MDR Medical Device Regulation

ML Machine Learning

NCBI National Center for Biotechnology Information

NCT National Clinical Trial

NIH National Institutes of Health
NLP Natural Language Processing
OS Observational Study
PICO Population, Intervention, Comparison, Outcome
PII Personally Identifiable Information
PMCF Post-Market Clinical Follow-up
PMID PubMed Identifier
PRISMA Preferred Reporting Items for Systematic Reviews and Meta-Analyses
PT Publication Type
RCT Randomized Controlled Trial
REGEX Regular Expression
ROC Receiver Operating Characteristic
SR Systematic Review
SRMA Systematic Review and Meta-Analysis
SVM Support Vector Machine
TN True Negative
TP True Positive
TPR True Positive Rate
UDI Unique Device Identification
UID Unique Identifier
URL Uniform Resource Locator
XML eXtensible Markup Language

Bibliography

- [1] Barton, S. (2001). Using clinical evidence. *British Medical Journal*, 322(7285), 503–504. <https://doi.org/10.1136/bmj.322.7285.503>
- [2] Oakes, G. (2020). The Importance of Evidence. *Project Reviews, Assurance and Governance*, 137–170. <https://doi.org/10.4324/9781315602462-13>
- [3] Horton, R. (2015). Offline: What is medicine's 5 sigma? *The Lancet*, 385(9976), 1380. [https://doi.org/10.1016/S0140-6736\(15\)60696-1](https://doi.org/10.1016/S0140-6736(15)60696-1)
- [4] Szajewska, H. (2018). Evidence-Based Medicine and Clinical Research: Both Are Needed, Neither Is Perfect. *Annals of Nutrition and Metabolism*, 72(suppl 3), 13–23. <https://doi.org/10.1159/000487375>
- [5] Geddes, J., & Carney, S. (2003). Systematic reviews and meta-analyses. *Evidence in Mental Health Care, February*, 53–59. <https://doi.org/10.1016/b978-0-443-06367-1.50015-6>
- [6] Ganeshkumar, P., & Gopalakrishnan, S. (2013). Systematic reviews and meta-analysis: Understanding the best evidence in primary healthcare. *Journal of Family Medicine and Primary Care*, 2(1), 9. <https://doi.org/10.4103/2249-4863.109934>
- [7] Atkinson, L. Z., & Cipriani, A. (2018). How to carry out a literature search for a systematic review: a practical guide. *BJPsych Advances*, 24(2), 74–82. <https://doi.org/10.1192/bja.2017.3>
- [8] Beller, E., Clark, J., Tsafnat, G., Adams, C., Diehl, H., Lund, H., Ouzzani, M., Thayer, K., Thomas, J., Turner, T., Xia, J., Robinson, K., Glasziou, P., Ahtirski, O., Christensen, R., Elliott, J., Graziosi, S., Kuiper, J., Moustgaard, R., ... Wedel-Heinen, I. (2018). Making progress with the automation of systematic reviews: Principles of the International Collaboration for the

- Automation of Systematic Reviews (ICASR). *Systematic Reviews*, 7(1), 1–7. <https://doi.org/10.1186/s13643-018-0740-7>
- [9] Smalheiser, N. R., Lin, C., Jia, L., Jiang, Y., Cohen, A. M., Yu, C., Davis, J. M., Adams, C. E., McDonagh, M. S., & Meng, W. (2014). Design and implementation of Metta, a metasearch engine for biomedical literature retrieval intended for systematic reviewers. *Health Information Science and Systems*, 2(1), 1–9. <https://doi.org/10.1186/2047-2501-2-1>
- [10] Soto, A. J., Przybyła, P., & Ananiadou, S. (2019). Thalia: Semantic search engine for biomedical abstracts. *Bioinformatics*, 35(10), 1799–1801. <https://doi.org/10.1093/bioinformatics/bty871>
- [11] Ouzzani, M., Hammady, H., Fedorowicz, Z., & Elmagarmid, A. (2016). Rayyan—a web and mobile app for systematic reviews. *Systematic Reviews*, 5(1), 1–10. <https://doi.org/10.1186/s13643-016-0384-4>
- [12] Torres Torres, M., & Adams, C. E. (2017). RevManHAL: Towards automatic text generation in systematic reviews. *Systematic Reviews*, 6(1), 1–7. <https://doi.org/10.1186/s13643-017-0421-y>
- [13] Marshall, C., & Brereton, P. (2015). Systematic review toolbox: A catalogue of tools to support systematic reviews. *ACM International Conference Proceeding Series*, 27-29-April-2015(May). <https://doi.org/10.1145/2745802.2745824>
- [14] Marshall, I. J., & Wallace, B. C. (2019). Toward systematic review automation: A practical guide to using machine learning tools in research synthesis. *Systematic Reviews*, 8(1), 1–10. <https://doi.org/10.1186/s13643-019-1074-9>
- [15] Marshall, I. J., Noel-Storr, A., Kuiper, J., Thomas, J., & Wallace, B. C. (2018). Machine learning for identifying Randomized Controlled Trials: An evaluation and practitioner’s guide. *Research Synthesis Methods*, 9(4), 602–614. <https://doi.org/10.1002/jrsm.1287>
- [16] Thomas, J., McDonald, S., Noel-Storr, A., Shemilt, I., Elliott, J., Mavergames, C., & Marshall, I. J. (2021). Machine learning reduced workload with minimal risk of missing studies: development and evaluation of a randomized controlled trial classifier for Cochrane Reviews. *Journal of Clinical Epidemiology*, 133, 140–151. <https://doi.org/10.1016/j.jclinepi.2020.11.003>
- [17] Kiritchenko, S., De Bruijn, B., Carini, S., Martin, J., & Sim, I. (2010). ExaCT: Automatic extraction of clinical trial characteristics from journal publications. *BMC Medical Informatics and Decision Making*, 10(1). <https://doi.org/10.1186/1472-6947-10-56>

- [18] Bulla, L., Gangemi, A., Golinelli, D., Mongiovì, M., Nuzzolese, A. G., Rucci, P., Sanmarchi, F., Catania, R., & Bologna, U. (n.d.). *Toward AI-assisted Systematic Literature Reviews for Evidence-Based Medicine. I.*
- [19] Central, P. (2006). Entrez Help. *DNA Sequence, Md*, 1–39.
- [20] Sayers, E. (2010). 4. The E-utilities In-Depth : Parameters , Syntax and More. *Entrez Programming Utilities Help, January 2019*, 1–18.
- [21] Notes, I. (2021). *The Insider ' s Guide to Accessing NLM Data The 9 E-utilities and Associated Parameters.* 1–8.
- [22] Shariff, S. Z., Bejaimal, S. A. D., Sontrop, J. M., Iansavichus, A. V., Haynes, R. B., Weir, M. A., & Garg, A. X. (2013). Retrieving clinical evidence: A comparison of PubMed and Google scholar for quick clinical searches. *Journal of Medical Internet Research*, 15(8), 1–13. <https://doi.org/10.2196/jmir.2624>
- [23] Howells, M. (2007). *Editors ' Bulletin CrossRef: an overview CrossRef: an overview.* 1742(2006), 12–16. <https://doi.org/10.1101/gr.10.12.1841>

Sitography

- [I] https://www.researchgate.net/figure/Hierarchy-of-evidence-pyramid-The-pyramidal-shape-qualitatively-integrates-the-amount-of_fig1_311504831
- [II] <https://training.cochrane.org/handbook/current>
- [III] <https://www.evidence.it/articolodettaglio/209/it/376/consort-2010-spiegazione-ed-elaborazione-linee-guida-aggiornate/articolo>
- [IV] https://kemh.libguides.com/library/search_tips/faqs/difference_between_pubmed_medline_embase
- [V] <https://blog.apify.com/how-to-use-web-scraping-for-online-research/>
- [VI] <https://www.zyte.com/learn/what-is-web-scraping/>
- [VII] <https://www.webharvy.com/articles/what-is-web-scraping.html>
- [VIII] <https://www.imq.it/en/eu-directives/regulation-medical-devices-mdr>
- [IX] [https://www.emergobyul.com/resources/cer-quick-answers#:~:text=A%20Clinical%20Evaluation%20Report%20\(CER\)%20documents%20the%20conclusions%20of%20a,studies%20on%20substantiall y%20equivalent%20devices](https://www.emergobyul.com/resources/cer-quick-answers#:~:text=A%20Clinical%20Evaluation%20Report%20(CER)%20documents%20the%20conclusions%20of%20a,studies%20on%20substantiall y%20equivalent%20devices)
- [X] <https://www.linkedin.com/pulse/analyzing-differences-between-cers-mdr-vs-mdd-ethan-drower/>
- [XI] <https://www.netguru.com/blog/python-pros-and-cons>
- [XII] <https://www.ncbi.nlm.nih.gov/books/NBK25497/6>
- [XIII] <https://docs.python-requests.org/en/latest/>

- [XIV] <https://stuyhsdesign.wordpress.com/basic-html/structure-html-document/>
- [XV] <https://oak-tree.tech/blog/python-web-scraping-selenium>
- [XVI] <https://www.geeksforgeeks.org/difference-between-static-and-dynamic-web-pages/>
- [XVII] <https://www.packetlabs.net/posts/dynamic-pages/>
- [XVIII] <https://www.sciencedirect.com/science/article/pii/S0140673606680379>
- [XIX] <https://www.analyticsvidhya.com/blog/2020/08/web-scraping-selenium-with-python/>
- [XX] <https://www.crossref.org/blog/open-abstracts-where-are-we/>
- [XXI] <https://www.bibtex.com/g/bibtex-format/>
- [XXII] <https://machinelearningmastery.com/types-of-classification-in-machine-learning/>
- [XXIII] https://en.wikipedia.org/wiki/Levenshtein_distance#:~:text=Informally%20C%20the%20Levenshtein%20distance%20between,considered%20this%20distance%20in%201965
- [XXIV] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions?retiredLocale=it
- [XXV] <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>
- [XXVI] <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
- [XXVII] <https://bmcneurol.biomedcentral.com/articles/10.1186/s12883-019-1442-z/figures/1>

Ringraziamenti

Vorrei ringraziare prima di tutto il mio relatore, il Professor Enrico Gianluca Caiani, per tutto il supporto che mi ha dato durante lo svolgimento della tesi, per avermi sempre aiutato e guidato in questo percorso, ma anche per l'enorme disponibilità che ha sempre dimostrato nei miei confronti. Un sincero grazie va anche alla Professoressa Alessia Paglialonga, per tutti i consigli che ho ricevuto durante i vari incontri. È stato un vero piacere poter lavorare con voi.

Un grazie alla mia famiglia, mamma, papà e Fede, per avermi sempre incoraggiato in questa avventura e tifato per me in tutti questi anni, spingendomi sempre a credere in me stessa.

Un enorme grazie a Giovanni, conosciuto per caso il primo anno di università, diventato una delle persone più importanti alla fine di questo percorso. Grazie per aver sempre sopportato le mie lamentele, per avermi sempre spinto a dare il meglio, per essermi stato sempre accanto dall'inizio di questo lungo percorso, per tutto quello che abbiamo condiviso e vissuto insieme dentro e fuori l'università, senza il tuo supporto probabilmente non sarei nemmeno arrivata fino a questo punto.

Grazie a Irene, coinquilina e amica da una vita, per avermi sempre aiutato nel momento del bisogno, per avermi reso tutte le giornate post università in questi ultimi anni più spensierate, per supportarmi e sopportarmi tutti i giorni, per essere un punto di riferimento nella mia vita e per spronarmi sempre a raggiungere gli obiettivi.

Grazie ad Aurora che mi ha sempre fatto vivere tutto con spensieratezza, per avermi regalato infiniti momenti di pazzia, e per aver condiviso con me praticamente tutta la vita universitaria. Ci siamo sempre aiutate a vicenda per superare tutte le difficoltà incontrate, supportate nei momenti più difficili e sempre spronate per andare avanti senza mai cadere.

Grazie a Fabiola, mio braccio destro di sempre, quella persona che in qualche modo strano riesce sempre a dirti la cosa giusta per farti sentire speciale e per non abbatterti mai, spingendoti sempre a credere in te stessa.

Grazie a Michela, per avermi sempre ascoltata e capita in qualsiasi occasione (universitaria e non), per avermi sempre spinto in qualche modo a dare il massimo e ad essere fiera di me e del mio percorso.

Grazie a Elisabetta, una delle persone più buone che io abbia mai conosciuto, sempre pronta ad esserci per me, e per avermi incoraggiato ad andare avanti e arrivare soddisfatta alla fine di questa avventura.

Grazie a Giacomo, Davide, Alessandro e Damiano, per aver reso le giornate in università meno pesanti, per avermi regalato migliaia di bei momenti, per esserci sempre stati per me. L'università ci ha fatto incontrare e ci ha fatto vivere esperienze indimenticabili che ci hanno legato tanto, e che ci hanno permesso di arrivare alla fine portandoci dietro un bagaglio bellissimo. Vi ringrazio veramente di cuore, perché senza di voi probabilmente avrei affrontato questo percorso con molte più difficoltà invece ci siamo sempre spronati a vicenda e supportati nel momento del bisogno.

Ringrazio comunque tutti coloro che mi hanno accompagnato e aiutato in questo cammino, porterò sempre con me dei ricordi bellissimi.

