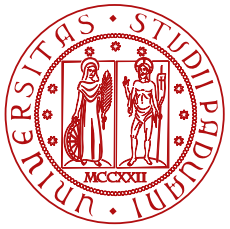


Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in ICT for Internet and Multimedia

Developing an NLP-based information retrieval tool for analysing the impact of AI on EEG studies

Master Candidate

Mehves Koksa

Student ID 2085456

Supervisor

Prof. Leonardo Badia

University of Padova

Co-supervisor

Prof. Giulia Cisotto

University of Trieste

Academic Year
2024/2025

To my family, for their endless love that carried me through every season. To my friends, for the laughter, the silence, and the moments that became memories. To all who shared sweet times I still remember with warmth, and to the loved ones no longer beside me, yet forever close in heart.

Abstract

The exponential growth of scientific publications has made it increasingly difficult to monitor how specialized research fields evolve over time. This thesis presents a fully reproducible and interpretable text-mining pipeline for analyzing electroencephalography (EEG)related literature on the open-access repository arXiv. The system operates end to end: from deterministic retrieval and structured parsing of arXiv records to keyword-based semantic filtering and category-aware classification. Two main filtering layers are implemented. The first ensures exhaustive coverage of EEG publications through paginated acquisition and normalization of metadata. The second applies strict Boolean matching with synonym expansion, enabling transparent and reproducible identification of topical intersections such as EEG + seizure or EEG + machine learning.

The analytical framework divides into two complementary branches. The first visualizes lexical and temporal trends through word clouds and publicationyear histograms, revealing the evolution of conceptual vocabulary within EEG research. The second branch performs supervised classification, mapping EEG papers into standard arXiv subject categories to study their disciplinary distribution. All operations are implemented with fixed random seeds and fully logged configurations to guarantee identical regeneration of figures and tables.

Developed in collaboration with Prof. Kamal Choudhary (Johns Hopkins University, formerly NIST), this work draws conceptual inspiration from the *ChemNLP* framework [1] but extends it toward neuroscience, introducing strict matching, exhaustive pagination, and domain-specific lexical analysis. The resulting open-science platform bridges explainable natural language processing and neuroinformatics, offering a scalable, transparent foundation for future bibliometric and semantic studies in EEG and related fields.

Sommario

La crescita esponenziale delle pubblicazioni scientifiche ha reso sempre più difficile monitorare come i campi di ricerca specializzati si evolvano nel tempo. Questa tesi presenta una pipeline di text mining completamente riproducibile e interpretabile per lanalisi della letteratura relativa allelettroencefalografia (EEG) nel repository ad accesso aperto arXiv. Il sistema opera in modo end-to-end: dal recupero deterministico e dallanalisi strutturata dei record di arXiv al filtraggio semantico basato su parole chiave e alla classificazione sensibile alle categorie. Sono implementati due principali livelli di filtraggio. Il primo garantisce una copertura esaustiva delle pubblicazioni EEG attraverso lacquisizione paginata e la normalizzazione dei metadati. Il secondo applica un rigido abbinamento booleano con espansione dei sinonimi, consentendo lidentificazione trasparente e riproducibile di intersezioni tematiche come EEG + crisi epilettiche o EEG + apprendimento automatico.

Il quadro analitico si divide in due rami complementari. Il primo visualizza le tendenze lessicali e temporali attraverso *word cloud* e istogrammi dellanno di pubblicazione, rivelando levoluzione del vocabolario concettuale allinterno della ricerca EEG. Il secondo ramo esegue una classificazione supervisionata, mappando gli articoli EEG nelle categorie standard di arXiv per studiarne la distribuzione disciplinare. Tutte le operazioni sono implementate con semi casuali fissi e configurazioni completamente registrate per garantire la rigenerazione identica di figure e tabelle.

Sviluppato in collaborazione con il Prof. Kamal Choudhary (Johns Hopkins University, precedentemente NIST), questo lavoro trae ispirazione concettuale dal framework *ChemNLP* [1] ma lo estende verso le neuroscienze, introducendo un abbinamento rigoroso, una paginazione esaustiva e unanalisi lessicale specifica per il dominio. La piattaforma di *open science* risultante collega lelaborazione del linguaggio naturale spiegabile e la neuroinformatica, offrendo una base scalabile e trasparente per futuri studi bibliometrici e semantici sull'EEG e sui campi correlati.

Contents

List of Figures

List of Tables

List of Acronyms

API	Application Programming Interface
arXiv	Open-access Repository for e-prints
ASR	Active Systematic Review
ATOM	Atom Syndication Format (XML-based Feed Format)
BCI	BrainComputer Interface
BM25	Best Matching 25 (Ranking Function)
ChemNLP	Chemistry Natural Language Processing Framework
CM	Confusion Matrix
CSV	Comma Separated Values
CS	Computer Science
CV	Computer Vision
CV2	Cross-Validation
DFT	Density Functional Theory
EEG	Electroencephalography
EEGLAB	Electroencephalography Laboratory Toolbox
EESS	Electrical Engineering and Systems Science
F1	F1 Score (Harmonic Mean of Precision and Recall)
GB	Gradient Boosting

HCI	HumanComputer Interaction
HGB	Histogram-based Gradient Boosting
HGBDT	Histogram-based Gradient Boosted Decision Trees
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IR	Information Retrieval
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LaTeX	Document Preparation System for Scientific Writing
LDA	Latent Dirichlet Allocation
LG	Learning (Machine Learning, `cs.LG`)
LR	Logistic Regression
MCI	Mild Cognitive Impairment
ML	Machine Learning
NB	Naive Bayes
NBF	Naive Bayes Family of Classifiers
NER	Named Entity Recognition
NFKC	Normalization Form KC (Unicode Normalization Form)
NLTK	Natural Language Toolkit
NLP	Natural Language Processing
OPT	Open Pre-trained Transformer
PNG	Portable Network Graphics
Q-BIO	Quantitative Biology

RFC	Request for Comments
RF	Random Forest
REST	Representational State Transfer
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
ROC	Receiver Operating Characteristic
SciBERT	Scientific BERT
SSVEP	Steady-State Visually Evoked Potential
SVD	Singular Value Decomposition
SVC	Support Vector Classifier
SVM	Support Vector Machine
T5	Text-to-Text Transfer Transformer
TF-IDF	Term FrequencyInverse Document Frequency
t-SNE	t-distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
UTF-8	8-bit Unicode Transformation Format
XML	Extensible Markup Language
XLNet	Generalized Autoregressive Pretraining for Language Understanding

Introduction

Electroencephalography (EEG) is one of the most established techniques for recording brain activity with millisecond temporal resolution. It is widely used in neuroscience, biomedical engineering, and humancomputer interaction, supporting applications such as seizure detection, cognitive workload monitoring, and braincomputer interfaces (BCIs). The rapid expansion of EEG-related publicationsdriven by advances in hardware, open-source toolboxes, and machine learninghas made it increasingly difficult for researchers to follow trends and identify key developments. Manual reviews are limited in scope, and most database search tools rely on opaque ranking mechanisms that hinder reproducibility.

This thesis develops a deterministic and reproducible workflow for retrieving and analyzing EEG-related literature on the open-access repository arXiv. The objective is to create an automated platform that systematically organizes, filters, and visualizes EEG publications in line with modern principles of Open Science. The approach is inspired by the methodological principles of the ChemNLP framework, originally proposed by Prof. Kamal Choudhary (now at Johns Hopkins University, formerly at the National Institute of Standards and Technology, NIST) and collaborators [1]. The present work extends that philosophy to the EEG research domain, emphasizing transparency, traceability, and reproducibility in text-based scientific analysis.[1]

At the core of the system, two main filtering stages are implemented. The first performs an exhaustive, reproducible retrieval of EEG-related records through the arXiv API, ensuring complete coverage and consistent metadata formatting. The second filtering stage applies a strict, keyword-based selection enhanced with synonym expansion to identify meaningful intersections of research topics

[1]The full code and documentation are openly available at `https://github.com/knc6/eeg_nlp`.

(for example, *EEG + machine learning* or *EEG + seizure detection*). These filtering layers form the foundation for two distinct branches of analysis. The first branch focuses on visualization: word clouds and publicationtrend figures reveal how conceptual vocabulary and research activity evolve over time. The second branch performs supervised classification, mapping EEG publications into the standard arXiv subject categoriessuch as `eess.SP`, `cs.LG`, and `q-bio.NC`to explore their disciplinary distribution.

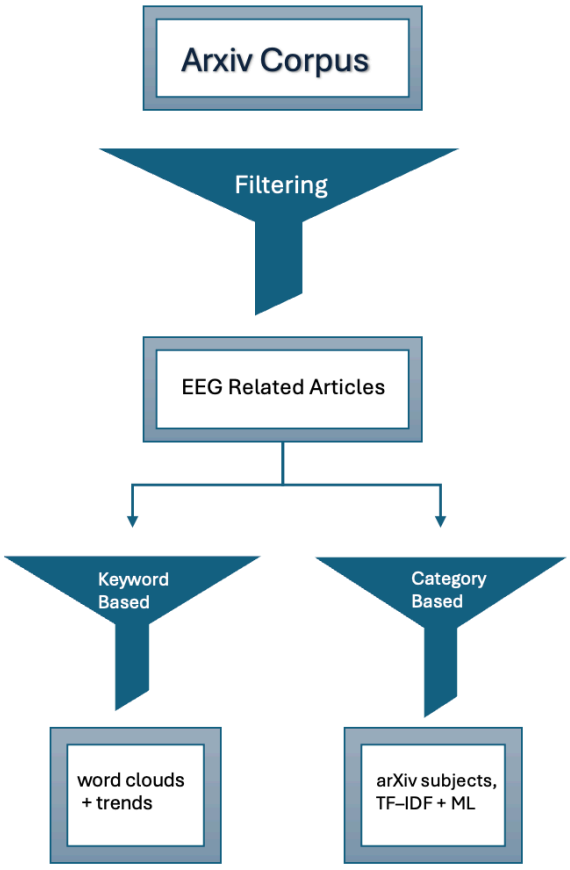


Figure 1: Overview of the proposed EEGarXiv pipeline. The process starts from the complete arXiv corpus, which is filtered in two sequential stages to extract EEG-related records. The first stage performs deterministic retrieval and normalization of metadata, while the second stage applies keyword-based filtering enhanced with synonym expansion. The resulting EEG subset is analyzed through two main branches: (i) keyword-based visualization of lexical and temporal trends, and (ii) category-based classification into arXiv subject labels using TFIDF features and classical machine-learning models.

Figure 1 summarizes the overall workflow. The process begins with the complete *arXiv* corpus, from which EEG-related records are extracted through two sequential filtering layers. The first layer, dedicated to retrieval and normalization, guarantees that all relevant metadata are collected in a consistent, machine-readable format. The second layer performs keyword-based semantic filtering, combining synonym expansion and strict Boolean logic to identify EEG-centered studies with high precision. After these two filtering stages, the resulting corpus is divided into two main analytical branches. The first branch focuses on visualization, producing lexical word clouds and publicationyear trend figures that describe the temporal and conceptual evolution of EEG research. The second branch performs supervised classification, leveraging arXivs subject taxonomy to map EEG-related works across disciplines using TFIDF features and interpretable machine-learning models. Together, these components define a transparent, end-to-end pipeline designed for reproducible bibliometric exploration.

The methodological choices are deliberately conservative and interpretable. Rather than relying on opaque neural rankers, the system uses transparent rule-based matching and classical machine-learning baselines that allow every inclusion and exclusion decision to be traced to explicit textual evidence. All parameters that influence randomnesssuch as random seeds and cross-validation splitsare fixed, ensuring byte-for-byte reproducibility of figures and tables. In this way, the pipeline aligns with the open-science principles promoted by initiatives such as ChemNLP, where scientific text mining is conducted with reproducibility, clarity, and public accessibility at its core.

Overall, this thesis contributes a fully documented and auditable platform for large-scale text analysis in the EEG domain. It demonstrates that transparent, deterministic natural language processing tools can be effectively adapted from materials science to neuroscience, providing an extensible foundation for future bibliometric and semantic studies. The following chapters describe the system architecture, the main experimental results, and their implications for data-driven exploration of open-access EEG literature.

Finally, the remainder of this thesis is organized as follows. Chapter 2 describes the technical implementation of each module in detail, from data retrieval and normalization to strict matching, scoring, and category classification. Chapter 3 presents the experimental results, including lexical visualizations, publication trends, coverage analyses of keyword combinations, and classification

outcomes with their associated confusion matrices. The final chapter discusses limitations, potential extensions such as domain-specific synonym resources or embedding-based similarity measures, and the broader implications of reproducible text-mining frameworks for open-access scientific literature.

1

Related Works

The exponential growth of scientific publishing over the last two decades has dramatically reshaped how researchers interact with literature, producing a scale and diversity that can no longer be managed through manual review alone [2]. The resulting need for automated, reproducible, and interpretable literature analysis pipelines has given rise to a wide spectrum of methods at the intersection of bibliometrics, natural language processing (NLP), and information retrieval (IR). Traditional scientometric indicators such as citation counts, co-authorship networks, and journal impact scores remain essential for quantitative assessment of influence, but they fail to capture semantic depth or conceptual similarity between works [3, 4, 5, 6]. In contrast, text-centered approaches employ NLP to extract latent topics, terminologies, and relationships directly from article titles, abstracts, and even full texts, transforming bibliometrics from a purely statistical activity into a semantically enriched discovery process [7, 8]. The integration of bibliometric mapping with NLP has become particularly valuable for identifying research frontiers, tracing the evolution of scientific concepts, and quantifying interdisciplinarity. Visualization-oriented tools such as VOSviewer and CiteSpace [9, 10] enable co-occurrence mapping, citation-burst detection, and thematic clustering, while hybrid frameworks increasingly combine these capabilities with text-mining and topic modeling to produce dynamic maps of science. In information retrieval, classical models of term frequencyinverse document frequency (TF-IDF), query expansion, and relevance feedback [11] continue to inform these systems by offering interpretable mechanisms for document ranking and filtering.

A major milestone in this convergence of bibliometrics and NLP is the ChemNLP framework proposed by Choudhary and Kelley [1]. Their work embodies a large-scale, modular approach to automated literature mining, integrating both conventional statistical techniques and deep neural models to process arXiv and PubChem datasets. ChemNLP performs TFIDF computation, dimensionality reduction via t-distributed stochastic neighbor embedding (t-SNE), supervised classification using random forests, logistic regression, and support vector machines, as well as fine-tuning of transformer architectures such as DistilBERT for text categorization. Additionally, it incorporates XLNet-based named entity recognition (NER) and text-to-text generation using transformer models like T5 and OPT.

The most distinctive aspect of ChemNLP is its integration of textual information with domain-specific computational databases, notably JARVIS-DFT, enabling direct cross-linking between unstructured literature data and structured physical measurements. This data fusion capability effectively bridges linguistic analysis with materials discovery workflows, establishing a new paradigm for cross-domain information integration.

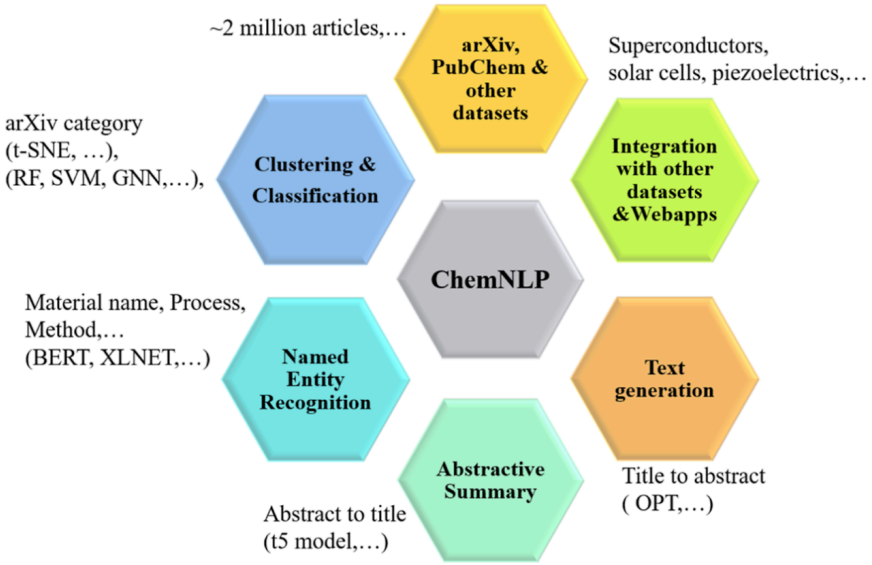


Figure 1.1: Schematic overview of the ChemNLP library. ChemNLP provides a software toolkit with integrated datasets and comprehensive AI/ML tools for expanding natural language processing applications to tasks such as text classification, clustering, named entity recognition, abstractive summarization, and text generation. Adapted from [1].

As illustrated in Figure 1.1, ChemNLP unifies data curation, representation learning, and scientific querying into a single interoperable framework, showcasing how literature analysis can evolve into a scalable, machine-assisted discovery process. This architecture exemplifies how the boundary between bibliometric exploration and scientific data mining can be blurred through information fusion an idea that profoundly influences the current generation of NLP-driven bibliometric pipelines. Inspired by this modular design, the present study adapts a simplified yet domain-specialized version of the ChemNLP workflow for EEG-related literature, emphasizing interpretability and reproducibility over deep-learning complexity while preserving the structured, multi-stage analytical logic of the original framework.

Table 1.1 summarizes key frameworks and representative studies that have influenced the present work, highlighting their respective domains, data sources, methodological focus, and contributions to automated literature analysis. This comparative overview clarifies how the proposed EEG-oriented pipeline positions itself within a continuum ranging from deep-learning-driven bibliometric architectures to interpretable, rule-based retrieval systems.

Table 1.1: Comparison of analytical strategies across related NLP and bibliometric frameworks, including the present EEG-oriented pipeline.

Framework / Study	**Method Family**	**Main Technique(s)**	**Analytical Focus and Limitations**
ChemNLP [1]	Deep-learning bibliometrics	TF–IDF, t-SNE, SVM/RF, DistilBERT, XLNet NER, T5/OPT generation	Full-stack NLP with entity recognition and generation; high accuracy but compute-intensive and domain-specific.
VOSviewer / CiteSpace [9, 10]	Network-based mapping	Co-occurrence, citation bursts, cluster visualization	Reveals research clusters and evolution; lacks semantic and textual depth, no synonym expansion.
ASReview [12]	Active-learning screening	Iterative model retraining, uncertainty sampling	Efficient semi-automation for systematic reviews; requires manual feedback and labeled seed data.
Jha et al. [7] / Al-Moslmi et al. [8]	Classical NLP text analysis	TF–IDF, tokenization, POS tagging, simple word clouds	Enables surface-level semantic profiling and keyword salience; no combinational filters or domain adaptation.
EEG bibliometrics [13, 14, 15, 16, 17]	Descriptive bibliometrics	Keyword co-occurrence, citation/country networks, topic modeling	Captures publication trends and thematic clusters; rarely includes synonym handling, pagination, or semantic filtering.
NeuroNLP	Lightweight IR + NLP	Synonym expansion, combinational keyword filtering, real pagination, word cloud/trend generation	Transparent and reproducible pipeline bridging classical IR and NLP; interpretable results, minimal compute cost, domain-specific focus.

The methodological underpinnings of ChemNLP stem from a decade of

progress in domain-specific NLP for chemistry and materials science. Early works such as ChemDataExtractor [18] established the feasibility of automated entity and property extraction, while later efforts focused on generating structured synthesis datasets from experimental text, as in Kononova et al. [19]. In parallel, unsupervised word embedding methods were shown to capture implicit domain knowledge from scientific corpora, a finding demonstrated by Tshitoyan et al. [20], who revealed that embeddings trained on materials science abstracts encode physical properties and compositional trends without supervision. Complementary studies have expanded these ideas to a variety of subdomains, including polymer chemistry, glass science, and battery research, collectively demonstrating that NLP can serve both as a discovery engine and as a knowledge consolidation tool [21]. Moreover, the integration of visual analytics such as t-SNE and UMAP projections of semantic embeddings has enabled interpretability in text clustering, providing researchers with an intuitive means to explore high-dimensional linguistic spaces (see Fig. 2 in [21] for representative mapping). These collective advancements provide the conceptual foundation upon which ChemNLP and similar domain-adapted frameworks are built.

While materials science has served as a fertile testbed for NLP-driven bibliometrics, comparable developments are increasingly visible in the biomedical and neuroscience communities, particularly in the context of electroencephalography (EEG). Bibliometric analyses in this field have aimed to delineate thematic clusters around neuroimaging, brain computer interfaces (BCI), clinical diagnostics, and cognitive neuroscience [14, 13, 15]. For instance, Liao [13] analyzed EEG publications related to Parkinsons disease and identified four principal research axessignal classification, deep learning, neurodegenerative biomarkers, and cognitive therapythrough keyword co-occurrence networks, which measure the frequency with which specific terms co-appear across publications and thereby uncover conceptual linkages and thematic clusters within the literature. Tsiamalou [14] explored EEG-based neurorehabilitation, mapping the evolution of research fronts from invasive electrode systems to modern wearable and dry-electrode technologies. Cisotto [22] proposed *hvEEGNet*, a high-fidelity deep learning architecture for EEG signal reconstruction that leverages convolutional and residual blocks to improve signal quality and structural interpretability, demonstrating how neural models can enhance both accuracy and physiological relevance in EEG signal analysis. Liu [15] provided a large-scale analysis of EEG research in mild cognitive impairment (MCI), highlighting international

collaborations and emerging trends in machine-learning-assisted diagnostics. Chen and Ding [16] extended the scope by combining bibliometrics with topic modeling, revealing conceptual overlaps between mental workload assessment and cognitive fatigue detection. Additionally, Fayaz [17] examined the bibliometric footprint of the EEGLAB software package, showing how open-source toolkits accelerate methodological standardization and influence citation networks. Although these studies have collectively improved our understanding of EEG-related research ecosystems, they largely rely on descriptive statistics, network co-occurrences, and frequency-based keyword analysis. Very few exploit advanced text-mining strategies such as synonym expansion, entity recognition, or combinational keyword modelingthe same methodological gap that motivates the present work.

Beyond domain-specific examples, numerous methodological efforts have expanded the technical toolbox for literature mining. Topic modeling algorithms, such as Latent Dirichlet Allocation (LDA) [23], and later neural variants (e.g., BERTopic, Top2Vec), enable latent topic discovery and document grouping based on contextual semantics rather than explicit keywords. Semantic expansion resources like WordNet [24] and ConceptNet [25] extend query recall by linking related lexical concepts, while hybrid IRNLP frameworks combine these resources with explicit query composition to improve search coverage in narrow domains [11]. Visualization and clustering tools, including VOSviewer, SciBERT-based embeddings, and hybrid graph layouts, make it possible to represent entire research fields as evolving semantic maps. Active-learning-based review assistants, such as ASReview [12], show that iterative, human-in-the-loop selection can balance recall and precision efficiently an idea that can complement keyword-based filtering in pipelines like ours. In bibliometrics, such tools are now being augmented with open science indicators, such as code availability, data accessibility, and reproducibility markers, which are increasingly viewed as proxies for methodological transparency [21]. Together, these methods form an ecosystem of interoperable modules that can be composed to achieve different trade-offs between automation, interpretability, and performance.

In this broader methodological context, the present study extends the ChemNLP philosophy to EEG-related literature, but in a targeted and resource-efficient manner. Our pipeline does not rely on heavy deep-learning models but instead implements a reproducible sequence of synonym expansion, combinational keyword generation, and strict relevance scoring to construct topic-specific corpora

from the arXiv API. Real pagination ensures that the full dataset is exhaustively retrieved, mitigating sampling bias inherent to API truncation. Keyword subsets are then analyzed through coverage statistics, publication trends, and word-cloud visualizations, allowing both quantitative and qualitative exploration of the EEG domain. The addition of category-based filtering enables contextual segmentation across research subfields (e.g., cs.LG, eess.SP, q-bio.NC), while optional detection of GitHub links provides an indicator of software availability and reproducibility. This approach deliberately positions itself between fully manual keyword selection and opaque deep learning retaining interpretability while maintaining scalability. A didactic schematic illustrating this process could follow the visual conventions of ChemNLPs workflow (Fig. 1 in [1]), adapted for the EEG domain.

2

Materials and Methods

This chapter documents the full data pipeline built for retrieving, parsing, organizing, and analyzing arXiv records related to EEG research. All procedures are implemented in Python with deterministic settings where applicable. Fixed random initialization seeds (e.g., for `NumPy` and `scikit-learn` models) are used to ensure reproducible feature extraction and model training. The workflow is scripted end-to-end to allow full reproducibility from raw API responses to final figures and tables. All scripts and source files used in this workflow are openly available in the public GitHub repository[1].

2.1 Overview

This thesis uses a modular workflow that starts with querying the arXiv REST API, then normalizes and structures the returned metadata into tabular form for analysis. Exploratory analyses provide initial signal about topical terms and temporal trends. A keyword-driven filtering module, augmented with synonym expansion, performs strict matching to emulate user queries that require all selected concepts to be present in each article. Visualization utilities summarize coverage by keyword combinations and over time. A separate category-based branch uses arXiv subject categories as labels for light-weight text classification baselines. Auxiliary text-mining utilities and external resource extraction

[1] `https://github.com/knc6/eeg_nlp/blob/main/eeg_arxiv.ipynb`

complete the pipeline. Each module exposes explicit parameters so the same pipeline can be repeated under different assumptions.

2.2 Data Retrieval from arXiv

2.2.1 Accessing the REST API

Data were obtained from the official arXiv REST interface via parameterized HTTP requests; the query was designed to be field-agnosticsearching titles and abstracts uniformlyand was centered on an EEG-focused keyword set. Server responses arrived as Atom-formatted XML, and XML namespaces were specified with fixed URIs to prevent silent element-resolution errors when identically named tags exist in different schemas. For each record, a consistent schema was extracted comprising a canonical identifier, title, abstract, author list, publication and update timestamps, subject categories, and canonical links. Two layers of normalization were applied to textual fields. First, at the character-encoding layer, all content was converted to UTF-8, meaning that characters from different languages that are not mutually shared and technical symbols are represented within a single, universal character space; in practice this avoids garbling (e.g., characters turning into question marks), prevents the same visible character from being stored with different byte sequences across systems, and stabilizes downstream search, sorting, and equality comparisons. Second, at the content-cleaning layer, non-printable control characters and invisible whitespace types were stripped, line breaks and multi-space runs were collapsed to single spaces, and protective regular expressions filtered residual markup artefacts (e.g., HTML or LaTeX fragments). Timestamps were retained in their original ISO 8601 representation and, for temporal analyses, converted to native date-time objects and normalized to UTC to remove time-zone effects. On the request side, a descriptive client identifier was supplied, short delays were inserted between successive calls to respect responsible-use guidelines, and transient network failures were handled with bounded retries using *exponential backoff* that is, progressively increasing the waiting time between consecutive retry attempts to avoid server overload and ensure fair access. Early single-shot retrievals that requested very high result counts produced systematic discrepancies between client-side tallies and figures displayed in the web interface due to per-page service limits, index recency effects, cross-listing behavior, and differences in

default sorting. In response, the acquisition layer was hardened with deterministic secondary ordering (e.g., by identifier or timestamp), de-duplication keyed on the canonical identifier, and detailed logging of each calls *payload size* (the number of records returned per response) and *cumulative totals* (the running sum of all records collected), allowing transparent verification of data completeness. Because one-shot retrieval cannot guarantee completeness under these constraints, the workflow transitioned to a page-wise acquisition strategy, whose details are presented separately.

2.2.2 Handling Pagination

In this project, pagination means traversing results of arXiv in orderly slices rather than attempting to download everything at once. The REST interface deliberately returns only a limited number of records per request, even if a very large `max_results` is asked for. In our initial, single-shot trials (effectively first page only), we consistently observed a mismatch between the total number shown in the web interface and the number of items locally available. That undercoverage would have biased every downstream analysis: word-frequency profiles and word clouds would reflect the first page rather than the full population; year-by-year publication trends would be distorted by early-page ordering; and strict Boolean combination rates such as the conjunctive query *seizure AND EEG AND detection*, which requires *all* three terms to co-occur in the same record-would be unreliable. Pagination addresses these issues by exhaustively walking the result space until we have demonstrably collected all matches.

Our implementation follows the APIs contract and is organized in two stages. First, a single-page fetcher (`query_arxiv_page`) prepares a robust request and parses one page of results. A bare user query such as EEG is normalized to a field-agnostic form (`all:EEG`) so that titles and abstracts are searched uniformly, avoiding field-specific brittleness. The request includes a descriptive `User-Agent` and a finite `timeout`; HTTP status is checked with `raise_for_status` so that failures are surfaced explicitly rather than being silently ignored.

Responses are delivered by the arXiv API in *Atom XML* format, a standardized XML-based feed structure originally designed for web syndication (similar to RSS) in which each scientific article is represented as an `<entry>` element containing well-defined subfields such as `<title>`, `<summary>`, and `<author>`. This

machine-readable structure allows automatic parsing of publication metadata without relying on ad-hoc HTML scraping. To correctly interpret these feeds, we explicitly declare the XML namespaces for both the *Atom* vocabulary and the complementary *OpenSearch* vocabulary used by arXiv. The OpenSearch schema adds additional metadata fieldsmost notably `os:totalResults`, which specifies the total number of matching records for a given query, and `os:startIndex` / `os:itemsPerPage`, which indicate the pagination state. Declaring these namespaces with fixed Uniform Resource Identifiers (URIs) ensures that identically named tags from different schemas (e.g., `<title>` in Atom versus other XML vocabularies) are resolved unambiguously during parsing.

From each parsed response, we extract two primary components: (i) the list of `atom:entry` elements, each corresponding to one article, and (ii) the OpenSearch element `os:totalResults`, which answers how many papers match this query in total? and serves as the data-driven stopping criterion for pagination. Each entry is then mapped to a normalized record containing the title, abstract, publication and update timestamps, authors, canonical identifier, the full list of subject categories, and a derived primary category (the first element of that list).

Second, a multi-page driver (`fetch_all_arxiv`) iterates through the entire feed. We initialize `start = 0` and use a high-but-safe page size (`per_page = 2000`, consistent with provider guidance). After each page arrives, we append the new entries to an in-memory accumulator and advance `start` by the number of items actually received (not by the nominal page size), which prevents skipping or looping if a short page is returned. Short inter-request delays (`sleep_sec = 2`) respect responsible-use guidelines. The loop ends under either of two conditions: a final page is shorter than requested (natural exhaustion) or the cumulative number of collected items equals `os:totalResults`. For development or budgeted tests we may set an optional `hard_limit` to cap the run early without altering the production logic.

Several design choices improve robustness, auditability, and reproducibility across runs. **We perform strict deduplication by canonical arXiv identifiers**, ensuring that items reappearing on later pages or in multiple categories are counted only once in the global dataset. This prevents inflation of statistics in downstream frequency and trend analyses. For studies that require version-awareness (e.g., to distinguish between `v1` and `v2` revisions of the same paper), the corresponding policy is explicitly recorded in the run metadata so that identical results can be regenerated in future acquisitions. Otherwise, all versions

are collapsed into the base identifier to maintain a single, stable representation per publication [26, 27].

Each request is instrumented with transparent, structured logging: the script records page index, payload size, cumulative totals, elapsed time, and all active configuration parameters (`per_page`, `sleep_sec`, `sortBy`, `sortOrder`). This level of traceability enables the exact replay of any run and helps diagnose potential network errors or incomplete transfers. Lightweight console updatessuch as progress messages in the form `Fetched X / total...`provide immediate visual feedback without generating excessive disk output [28, 1].

Because the underlying arXiv index can evolve during multi-hour retrievals, we impose deterministic secondary ordering (by canonical identifier or publication timestamp) before persisting the final table. The snapshot is stamped with the observed `os:totalResults`, along with the precise timestamps of the first and last successful requests. This metadata acts as a version tag for the acquired corpus, clarifying which point in time the dataset represents and enabling reproducibility if the feed expands mid-run. These combined safeguardscanonical deduplication, explicit logging, and deterministic orderingmake the pagination process both transparent and fully traceable, turning what is usually a transient API query into a stable, research-grade dataset [29, 30, 27].

Figure 2.1 illustrates a real run of the `fetch_all_arxiv` routine. The user request (EEG) is passed as input, and the script iteratively queries the API page by page. The console output logs both progress and verification statistics, ensuring that the number of retrieved records exactly matches the total reported by arXiv. This figure is included to visually clarify the internal behavior of the pagination algorithm and its role in guaranteeing full corpus coverage.

Once pagination completes, we materialize the accumulator as a single pandas `DataFrame` with explicit types: ISO 8601 timestamp strings and UTC-normalized native datetimes for temporal analyses; list-valued columns for `authors` and `categories` to preserve multiplicity; and lowercase mirrors (`title_lc`, `summary_lc`) to support reliable case-insensitive matching.

The resulting `DataFrame` is then serialized to disk using the Apache Parquet format, a modern columnar storage standard optimized for analytical workloads. Parquet preserves full data fidelityno rounding, truncation, or encoding loss occurs when reloading the filehence the process is described as a *lossless reload*. In practice, this means that each column retains its original data type (e.g., strings, timestamps, lists) and can be recovered byte-for-byte into an identical

```python
def fetch_all_arxiv(query, per_page=2000, sleep_sec=2, hard_limit=None):
    """
    Iterate pages until everything is fetched (or hit hard_limit).
    Returns (all_articles_list, total_reported_by_api)
    """
    all_rows, start, total_reported = [], 0, None
    while True:
        page, total = query_arxiv_page(query, start=start, max_results=per_page)
        if total_reported is None:
            total_reported = total
        if not page:
            break
        all_rows.extend(page)
        start += len(page)
        print(f"Fetched {len(all_rows)} / {total_reported or '?'} so far...")
        if hard_limit is not None and len(all_rows) >= hard_limit:
            all_rows = all_rows[:hard_limit]
            break
        if len(page) < per_page:   # last page
            break
        time.sleep(sleep_sec)
    return all_rows, total_reported

# --- RUN: fetch everything for EEG ---
query = "EEG"
articles, total_reported = fetch_all_arxiv(query, per_page=2000, sleep_sec=2)

print("\n[Summary]")
print("Total reported by API:", total_reported)
print("Actually fetched rows:", len(articles))

# Build DataFrame
df = pd.DataFrame(articles)
print("DataFrame shape:", df.shape)
display(df.head(5))
display(df.tail(5))


# (Optional) Save to CSV for later analysis
# df.to_csv("eeg_full_dataset.csv", index=False)
```

```
Fetched 2000 / 3614 so far...
Fetched 3614 / 3614 so far...

[Summary]
Total reported by API: 3614
Actually fetched rows: 3614
DataFrame shape: (3614, 8)
```

Figure 2.1: Example of the pagination module in execution. The Python function `fetch_all_arxiv` iteratively retrieves all arXiv pages matching the query `"EEG"` until the total number of items reported by the API (`total_reported`) equals the number of locally collected entries. The console output confirms full coverage: the log displays progress messages (`Fetched 2000 / 3614 so far`) and finally reports equality between the API count and the locally stored DataFrame shape (`(3614, 8)`). This inputoutput view illustrates how pagination ensures exhaustiveness and prevents partial retrievals while maintaining transparent progress reporting. The figure is illustrative rather than analytical and demonstrates how the software verifies completeness of data acquisition.

in-memory representation. Compared to traditional CSV files, Parquet offers smaller file sizes through efficient compression, faster I/O performance due to column-wise access, and native type preservation without the need for re-parsing during import. For compatibility with lightweight environments or spreadsheet-based inspection, a parallel export to CSV is also supported, using explicit UTF-8 encoding and Unix newline conventions to guarantee portability across systems.

In practice, this pagination design yields four concrete benefits. First, the local row count matches the feeds `os:totalResults`, establishing that coverage is complete rather than a convenience sample of page one. Second, measurement bias tied to early-page ordering disappears: word-frequency distributions, publication trends, and category histograms reflect the full corpus. Third, strict Boolean combination rates (e.g., *seizure AND EEG AND detection*) become meaningful because they are evaluated against the entire corpus rather than a small slicei.e., not just a limited portion of the retrieved data. Fourth, the acquisition process is transparent and repeatable: query normalization, page sizes, inter-request delays (short pauses inserted between successive API calls to respect server load limits), ordering, and version-handling policies are explicit, logged, and versioned in code. While these polite delays and large result sets can increase runtime, the tradeoff is an auditable, exhaustive dataset that makes every subsequent analysis defensible.

2.2.3 Parsing and Structuring Metadata

The Atom XML returned by the arXiv REST interface is transformed into a single, analysis-ready table by parsing each `atom:entry` and mapping its fields onto a consistent record schema. For every entry we read the canonical identifier (the arXiv id, including version suffixes such as `v1`, `v2`), the title and abstract (summary), the publication and update timestamps, the ordered list of author names, and the full list of subject categories as provided by the feed; from this category list we also derive a convenient `primary_category` as its first element. Each parsed record is appended to an in-memory accumulator that becomes a single pandas `DataFrame` in which one row corresponds to one article and columns correspond to stable metadata fields; this tabular form is treated as the single source of truth for all subsequent analyses.

To make textual data robust across platforms and runs, we apply two layers

of normalization before any analysis. At the character-encoding layer, all textual fields are represented in UTF-8 so that characters from different languages that are not mutually shared and technical symbols occupy a single, unambiguous code space; this prevents garbling (e.g., characters turning into question marks), eliminates equality mismatches caused by distinct byte sequences that render the same glyph, and stabilizes tokenization, search, and sorting. At the content-cleaning layer, we remove non-printable control characters and invisible whitespace variants, collapse hard line breaks and multi-space runs to single spaces, and filter residual markup artefacts with conservative regular expressions so that stray HTML or LaTeX fragments do not contaminate keyword matching or n-gram statistics. For reliable case-insensitive operations later on, we also materialize lowercase mirrors of the title and abstract (`title_lc`, `summary_lc`).

Temporal information is retained in two synchronized representations to balance fidelity and convenience: we store the original ISO 8601 strings exactly as provided by the feed, and in parallel we parse them into native datetime objects normalized to UTC. Normalizing to UTC removes time-zone effects and enables consistent grouping by year or month regardless of execution environment. Author names and subject categories are preserved as Python lists to retain multiplicity and order; when exporting to formats that lack native list types (e.g., CSV), these columns are serialized as JSON strings and, upon reload, explicitly de-serialized back to lists to avoid silent type drift. Integrity checks are enforced during ingestion to ensure a coherent and reproducible table. Records must have a valid canonical identifier and a non-empty title; malformed entries are skipped with a logged reason. **De-duplication** uses the canonical arXiv identifier so that the same article, if encountered across pages or re-runs, appears only once. Depending on the analysis goal, versioned identifiers can be kept distinct (useful for temporal audits of evolving manuscripts) or collapsed to the base id (useful to avoid double-counting in topical summaries); the chosen policy is recorded in run metadata. **Because feed order may vary over time, we impose a deterministic secondary ordering** either by canonical id or by publication timestamp before persistence so that later splits and summaries are stable across re-runs.

Finally, the normalized table is persisted either as a columnar Parquet file for lossless reloads and efficient slicing, or as a CSV with explicit encoding and newline conventions for broad tool interoperability. Alongside the data artefact, we provide a compact data dictionary that documents each columns semantics and

allowable values, and a lightweight validation report with row counts, null-rate summaries, and category cardinalities; we also attach a small snapshot header that records the observed total match count, the retrieval window (start/end timestamps of the acquisition), and parsing-relevant configuration (namespace URIs, encoding, list-serialization policy). Together, these steps move the data cleanly from raw XML to a stable, well-typed, and reproducible table that is immediately usable for exploratory analyses, keyword-based filtering, category-aware classification, and downstream visualizations.

2.2.4 GitHub Link Detection

To enrich the corpus with indicators of code availability, we implemented a lightweight detector that scans arXiv entries for explicit GitHub references. The module reuses the arXiv REST interface and parses Atom responses with `feedparser`, retrieving up to a fixed number of records via a parameterized query (e.g., `all:EEG`). For each entry, the textual fields most likely to contain repository linkstitle, abstract/summary, and the optional `arxiv_comment`are concatenated into a small search bag. A case-insensitive regular expression then extracts URLs that begin with the GitHub domain; in the current implementation, the pattern is a conservative token-level matcher of the form `https?://github.com/[...]` that stops at whitespace or bracket delimiters, thereby capturing both top-level repositories and deep paths occurring in comments or abstracts. Detected links are associated with the corresponding record and returned as a per-article list. The procedure is deterministic given the query and `max_results` parameters and produces a minimal, auditable signal of open-source disclosure directly from the metadata, without crawling the linked pages. In this baseline form, the detector is intentionally conservative: it focuses on the GitHub domain only, preserves deep URLs rather than canonicalizing to *owner/repo*, and does not attempt de-duplication across fields, which keeps behavior transparent while ensuring that explicit repository mentions are surfaced for downstream analysis.

The resulting output is a structured, human-readable listing of all arXiv entries that explicitly reference a GitHub repository. Each entry includes the article title and the corresponding extracted repository URL, enabling a quick overview of reproducible or code-available EEG research. Because the detector operates over the entire corpus, the list aggregates every paper in which a GitHub

link appears, regardless of whether the repository is hosted at the top level or within a deeper project path. This provides a transparent way to identify open-source contributions within the EEG domain without crawling external pages.

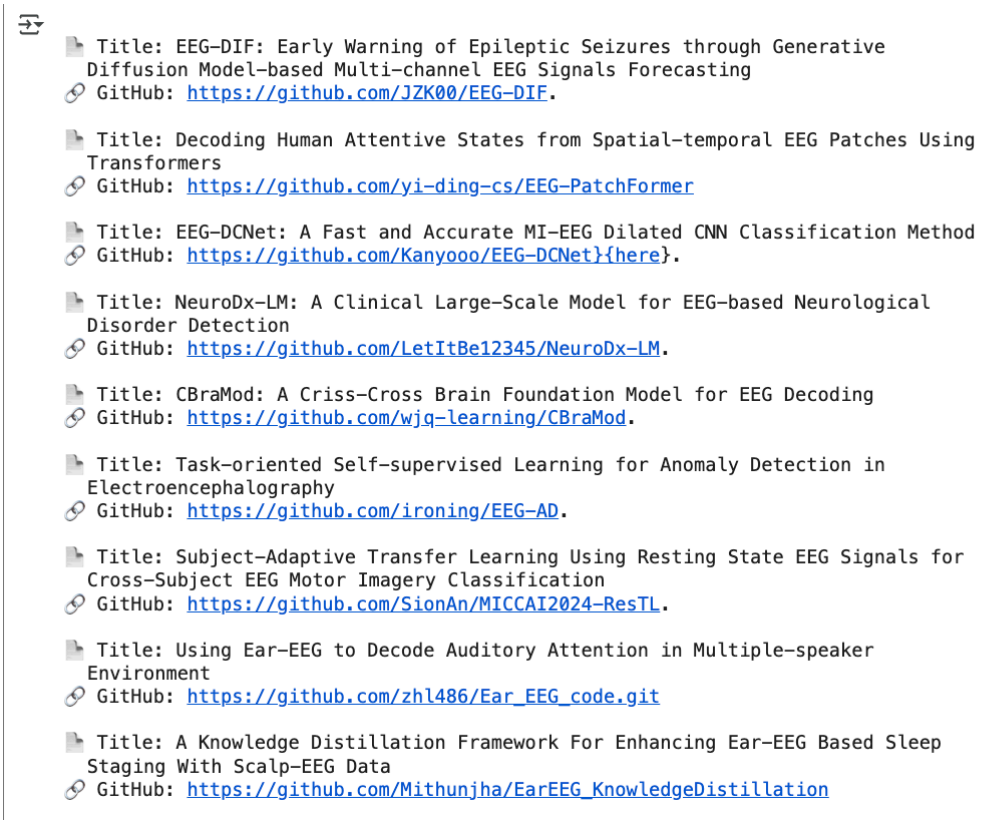


Figure 2.2: Excerpt of the GitHub Link Detection output. The module identifies all arXiv papers that include explicit GitHub references, displaying the paper titles alongside their corresponding repository URLs. This partial excerpt illustrates how the system lists each detected entry, allowing rapid inspection of open-source EEG projects and reproducible research resources available within the corpus.

2.3 Exploratory Analyses

2.3.1 Word Cloud Analysis of Titles

To gain an initial qualitative impression of lexical salience across the EEG corpus, **two complementary word clouds were generated:** one derived from *abstracts*, which expose richer methodological and application vocabulary, and one from *titles only*, which concentrate the author-selected descriptors. This

dual-level design allows the same textual preprocessing pipeline to be assessed under different content densities.

The pipeline concatenates the chosen text field (title or abstract) from all retrieved records, converts it to lowercase, removes non-alphabetic characters through a conservative regular expression, and filters English stop-words using the Natural Language Toolkit (NLTK). Stop-word removal is essential to prevent highly frequent function words (such as the, and, of) from dominating the visualization and to reveal domain-specific tokens that carry conceptual meaning. This lightweight normalization stabilizes tokenization and avoids artefacts such as the separation of hyphenated compounds or the inclusion of numerical sequences that do not contribute semantic value. Its trade-offs (for example, collapsing hyphenated expressions like real-time and discarding digits) are acceptable for an exploratory-level representation, where interpretability and reproducibility take precedence over morphological precision.

The graphical representation is determined by two main mechanisms. *Font size* scales deterministically with the relative frequency of each token in the cleaned text, providing a direct visual encoding of lexical prominence. *Placement and orientation*, in contrast, are determined by the Python `wordcloud` librarys spiral packing algorithm, which iteratively places words on a two-dimensional canvas following an Archimedean spiral pattern. More frequent words are positioned near the center, while less frequent ones are placed outward, ensuring a balanced spatial distribution and preventing visual overlap through collision detection. This algorithmic layout provides a reproducible yet visually interpretable depiction of lexical salience within the corpus. **Because this algorithm is inherently stochastic, two runs on the same text can yield slightly different spatial layouts** even though token frequencies remain unchanged. To guarantee reproducibility in archival figures, a fixed random seed (`random_state`) is specified to freeze the random placement sequence across runs, thus enabling byte-level identical outputs for the same dataset.

All operations are implemented in Python using open-source packages (`wordcloud`, `NLTK`, and `pandas`) under deterministic settings to ensure reproducibility. The final script outputs both the rendered visualization (in PNG format) and the corresponding frequency table, which explicitly maps each unique token to its occurrence count in the corpus. Saving these frequency tables allows full traceability: the same figures can be regenerated or quantitatively compared across different corpus snapshots as the arXiv database evolves. The cleaning config-

urationcomprising the regular expression (regex) pattern used to remove non-alphabetic characters, the stop-word inventory defining which common English words are excluded from analysis, and the fixed random seed controlling layout randomness in the `wordcloud` generationare logged alongside the outputs. Recording these parameters ensures that the visualization can be reproduced byte-for-byte and that methodological consistency is maintained across future updates of the dataset.

In summary, this procedure transforms the textual corpus into an interpretable visual distribution of lexical weightings that guides subsequent, more quantitative analyses. The resulting figures and their interpretation are reported in Chapter 3.

2.3.2 Publication Trend Estimation

The objective of the publication trend analysis is to quantify how the volume of EEG related work on arXiv evolves over calendar time and to provide a reproducible temporal baseline for comparisons across subsets (e.g., by primary category or strict Boolean filters). To this end, the feeds `published` timestamp is treated as the canonical event date because it represents the first public availability of each work. The `updated` timestamp is preserved for version-aware diagnostics but is excluded from counting to prevent artificial inflation of yearly totals.

For each record, **the publication year** was extracted from the ISO 8601 formatted date provided by arXiv. Each timestamp was parsed into a native datetime object, from which only the year component was retained. The corpus was then aggregated by publication year, resulting in a chronologically ordered sequence of yearly counts suitable for trend visualization and subsequent quantitative comparisons. **The aggregation procedure enforces a stable index order to guarantee consistent results across executions and software environments.**

Visualization follows a clean, information-preserving design. We plot a bar chart of yearly counts, where bar height encodes the number of publications in each calendar year. Descriptive titles and axis labels are included for interpretability. To maintain legibility over long timelines, x-axis tick marks are limited to roughly ten evenly spaced positions. Grid lines are drawn only along the y-axis to facilitate rapid comparison of bar heights. The chosen color scheme carries no analytic meaning; only height (count) and horizontal position (year)

encode information. Layout adjustments (tight bounding boxes and controlled label spacing) are applied to avoid overlapping annotations.

Several safeguards are implemented to preserve reproducibility and interpretability. First, all records are deduplicated by their canonical arXiv identifier, ensuring that multiple versions of the same paper do not inflate annual counts. By default, each unique paper is counted once in the year of its first appearance. If a version-aware audit is required that is, if the analysis explicitly tracks successive revisions of the same article (e.g., `v1`, `v2`, `v3`)the pipeline can alternatively count by `updated` timestamps or annotate version bursts to highlight periods of intensive revision activity. The selected rule is logged in run metadata for transparency. Finally, the yearly-count table is saved to disk (e.g., `data/pubcounts_yearly.csv`) so that the figure can be regenerated byte-for-byte and the same data reused for statistical comparisons or normalized analyses in later chapters.

The figure produced by this pipeline serves as a descriptive, not inferential, summary of publication dynamics. It provides a visual baseline that later enables trend normalization, cross-subset comparison, and correlation with topic-specific evolutions. Detailed quantitative trends and interpretations of these figures are reported in Chapter 3.

The figures produced by this pipeline serve as descriptive rather than inferential summaries of publication dynamics, providing a visual baseline for later trend normalization, cross-subset comparison, and correlation with topic-specific evolutions. To complement these temporal perspectives, an additional lexical diagnostic was performed to verify the internal coherence of the retrieved corpus. Specifically, word-frequency statistics and target-word coverage were computed on the cleaned titles and abstracts to ensure that the most salient tokens and their relative distributions align with expected EEG-related terminology.

This diagnostic step, implemented through Python `Counter` and the NLTK stopword corpus [28], yields a ranked vocabulary profile of the dataset as well as absolute and relative counts for a **manually defined set of target terms (*EEG*, *brain*, *BCI*, *signal*, and *detection*)**. Its purpose is to confirm that the filtered corpus is not only structurally valid but also lexically faithful to its conceptual scope. Unlike the combination-based filtering, which is inferential and hierarchical, this diagnostic remains purely descriptive and reproducible: it provides a direct lexical sanity check on the outcome of the strict matching

stage.

By integrating temporal and lexical diagnostics, the Materials and Methods section thus ensures that both the longitudinal and linguistic dimensions of the corpus are internally consistent before entering the interpretive analyses. The detailed quantitative outcomes of these steps including frequency rankings, target-word statistics, and temporal patterns are reported and discussed in Chapter 3.

2.4 Keyword-Based Filtering and Analysis

This module implements a second-stage filtering and re-ranking pipeline over the arXiv corpus previously collected for EEG-related literature. Starting from a user-specified keyword set, the system performs a controlled synonym expansion to capture lexical and typographic variants (e.g., hyphen/space alternations such as *real-time* vs. *real time*), while maintaining a cap on the number of variants to avoid semantic drift and noise [31, 28]. After expansion, the system enumerates all non-empty keyword combinations ordered from most specific (longest) to most general (singletons) so that documents satisfying the most constrained research intents are evaluated first.

For each combination, a strict and interpretable relevance score is applied, requiring every keyword (or one of its approved synonyms) to appear in the title or abstract. The scoring decomposes into section-aware signals presence in the title, early occurrence in the abstract, and repetition counts with a small discount applied when a match occurs through a synonym rather than the exact keyword. This design follows classical principles of information retrieval, balancing recall and precision through transparent, auditable rules [27]. When a document satisfies the strict match, two optional refinements are applied: (i) a local proximity bonus that slightly favors records where all query terms co-occur within a short window in the abstract [32]; and (ii) a temporal recency multiplier that gently down-weights older records without overpowering textual evidence [33].

The pipeline is engineered for transparency, reproducibility, and computational efficiency. Text normalization (lowercasing and Unicode cleaning) is performed once at ingestion, and precompiled regular expressions with relaxed word boundaries are used to ensure robust matching of hyphenated and compound terms while avoiding spurious partial matches. Deduplication by title

ensures that each article contributes to only one combination the most specific one it satisfies thereby preventing inflation of counts. **The system produces outputs at two levels: (i) a record-wise table** listing matched articles with their assigned combination, title, link, score, and abstract summary; **and (ii) summary reports** that quantify raw counts and percentage coverage per keyword or combination, relative to a configurable baseline (typically the EEG subset retrieved in the first stage, but optionally the full arXiv corpus) [26]. These reports can optionally be paired with visual summaries (e.g., bar charts sorted by combination length) to make topical gaps and overlaps visible, though the figures themselves are presented in Chapter 3.

Finally, the module exposes several tunable parameters including synonym list cap, proximity window size, recency half-life, and section weights so that the strictness of matching or the influence of temporal decay can be adapted to specific research questions. **The overall framework thus combines strict, explainable text matching with light contextual and temporal refinements,** yielding rankings that remain faithful to the query intent and easily defensible in academic analysis [27, 32, 33].

2.4.1 Synonym Expansion

Synonym expansion addresses one of the most common weaknesses of keyword-based filtering in scientific text: the lack of terminological standardization across authors. Even within a narrow subfield, equivalent concepts may appear under multiple surface forms such as real-time and real time, or electroencephalography and EEG. Restricting retrieval to literal string matches would therefore miss relevant works solely due to orthographic or lexical variation. To mitigate this, each user-provided keyword is expanded into a compact, controlled set of near-synonymous surface forms before any matching is attempted.

The expansion uses WordNets synsets and lemma inventory (via NLTK), which provides a conservative yet reliable source of lexical alternatives for general English vocabulary [31, 28]. The procedure prioritizes interpretability and control through several explicit choices:

- The user's original term is always retained to preserve the literal query, even when WordNet coverage is missing (e.g., acronyms or field-specific tokens).

- Retrieved lemma names are normalized to lowercase and stripped of underscores for compatibility with standard text preprocessing.

- Hyphen/space alternations are systematically introduced treating real-time and real time as equivalent because such orthographic variations are common in technical writing.

- Overly long or paraphrastic multiword expressions are excluded to prevent semantic drift and false positives.

- The number of candidate surface forms per keyword is capped and deterministically ordered to limit combinatorial explosion and ensure reproducibility.

Each expanded set feeds directly into the matcher, which applies precompiled regular expressions with relaxed boundaries. A document is considered a match if any surface form from the expanded set appears in its title or abstract. When a match occurs through a synonym rather than the literal keyword, the scoring function applies a modest weight discount (typically 0.8) to reflect weaker lexical identity while preserving topical relevance. This weighting reflects a core information retrieval principle: exact lexical matches convey stronger intent alignment, but controlled near-synonyms remain valuable sources of evidence [27].

Although WordNet provides high-precision coverage for common English vocabulary, its scope for domain-specific terminology is limited. To avoid introducing noisy or tangential terms, the system deliberately refrains from including distant semantic relations such as hypernyms or contextual paraphrases. Instead, it maintains a conservative stance: keep the users literal query, include only close lemma-based variants, and rely on the scoring function to modulate the influence of less precise matches. For studies requiring broader coverage, domain-specific lexicons or embedding-based semantic neighbors may be incorporated as optional extensions, provided that they respect the same cap and deterministic ordering principles.

Overall, synonym expansion in this system functions as a small yet strategically critical component: it substantially improves recall under orthographic and lexical variability while preserving interpretability, transparency, and reproducibility across runs [31, 28, 27].

2.4.2 Generation of Keyword Combinations

Given a user-provided set of k keywords, the system enumerates every non-empty subset of this set, resulting in $2^k - 1$ distinct combinations. Each subset represents a different research intent, ranging from the most specific (conjunctive queries that include all k terms) to the most general (single-term queries). The combinations are processed from longest to shortest so that highly specific intents are evaluated first, followed by progressively broader alternatives. This ordering aligns with established information-retrieval practice, where high-precision queries are prioritized before recall is expanded through controlled relaxation [11, 34].

This longest-to-shortest traversal offers two methodological advantages. Here, *traversal* refers to the systematic, ordered iteration through all possible keyword combinationsstarting from the longest subsets (those containing the greatest number of terms) and proceeding step-by-step toward shorter, more general ones. Such a topdown traversal mirrors how researchers naturally refine their searches: beginning with a tightly constrained hypothesis (e.g., *machine learning + brain + real-time*) and relaxing it only when coverage proves insufficient. This ordered evaluation ensures that highly specific, high-precision combinations are processed first, before broader ones are considered. Second, it enables clean deduplication and unambiguous attribution: once a document satisfies a longer (more specific) combination, it is assigned exclusively to that intent and excluded from its sub-combinations, preventing double counting and preserving interpretability. Consequently, aggregate counts and coverage percentages remain interpretable and free from artificial inflation. Treating each combination as a separate conjunctive query also preserves transparency, since a record qualifies only if *all* constituent terms (or their controlled synonyms) appear in the title or abstract, providing an explainable Boolean baseline within a modern text-mining framework [11].

Because the number of possible combinations grows exponentially with k, computational considerations become important. In practice, keyword lists are kept modest in size (typically 3-8 terms), and pruning rules can be applied to discard supersets of combinations that already yield empty intersections. The synonym-aware matching introduced in the previous section integrates seamlessly with this structure: longer combinations tend to generate fewer but more precise matches, whereas shorter ones increase recall. This behavior parallels

that of association-rule analyzes, in which larger sets of items are rarer, yet more specific [30]. Here, however, we are not mining item sets from transactions but systematically testing predefined conceptual intersections within scientific text.

Finally, the exhaustive yet ordered construction enhances explainability for subsequent reporting and visualization. Because every combination is explicitly enumerated and evaluated in a deterministic order, each coverage metric whether a raw count or percentage can be traced back to a transparent logical rule (documents that contain all of A, B, and C simultaneously). Likewise, each documents assignment is justified by its association with the first, longest combination it satisfies. This deterministic mapping from query intent to textual evidence ensures full reproducibility and interpretability, making the approach particularly suitable for academic bibliometric analysis [34, 11].

The complete enumeration and ordering of keyword combinations described above provide the structural foundation for the subsequent matching and scoring stages. The next section details how each combination is evaluated through a strict, interpretable relevance function that incorporates lexical, positional, and temporal cues.

2.4.3 Strict Matching and Relevance Scoring

The scoring stage of the pipeline begins with a deliberately strict conjunctive rule designed to capture genuine topical intersections rather than incidental term overlap. For a chosen keyword combination C (e.g., {*machine learning*, *brain*, *real-time*}), a document qualifies for evaluation only if *every* term in C or one of its controlled WordNet-derived surface variants appears in the title or abstract. This Boolean AND-based formulation encodes the users intent transparently and ensures interpretability: a paper either satisfies all conceptual constraints or is excluded altogether. Prior to matching, all text fields are normalized to lowercase helper columns, and precompiled regular expressions with relaxed look-around boundaries are applied in place of naive word boundaries. This approach captures common orthographic variants such as real-time versus real time while avoiding spurious partial matches (e.g., brainstorm falsely triggering brain).

For each keyword, a deterministic list of accepted variants is scanned sequentially. The search stops at the first successful match, ensuring efficiency and consistent attribution of evidence. When the literal keyword is matched,

it receives full weight (1.0); when the match is via a synonym or lexical variant, it is lightly down-weighted (0.8) to reflect reduced lexical identity while still acknowledging semantic alignment. This weighting reflects a foundational principle in information retrieval: exact lexical correspondence is a stronger indicator of intent, but close lexical neighbors are still valid and should not be ignored [35, 27, 36].

Once a document satisfies the all terms present criterion, an interpretable, section-aware scoring function aggregates distinct textual and contextual cues. Each matched keyword contributes to the overall score through additive, transparent features inspired by classical field weighting and positional heuristics:

- **Title presence:** +5 points for each occurrence of the keyword in the title, recognizing the high salience of title-level evidence.
- **Title prominence:** an additional +2 if the title begins with the term, reflecting author emphasis.
- **Title repetition:** +2 per additional instance beyond the first, rewarding reinforced topical focus.
- **Early abstract occurrence:** +3 if the term appears within the first ∼30 tokens of the abstract, capturing early contextual framing.
- **Abstract frequency:** +1 per subsequent occurrence anywhere in the abstract, accounting for extended thematic relevance.

When a match is triggered by a synonym rather than the original term, all corresponding contributions are scaled by 0.8, ensuring that lexical approximations carry proportionally reduced weight. This results in a fully additive and auditable score that can be decomposed term-by-term and position-by-position.

To acknowledge query complexity, an additional **combination-length bonus** of $+5 \times |C|$ is added to the aggregate score, rewarding documents that satisfy longer, more specific combinations. This bonus stabilizes ranking behavior by ensuring that more constrained queries do not get overshadowed by single-term matches with similar textual evidence. The final *strict text score* therefore encapsulates multiple dimensions of relevance: field (title versus abstract), position (early versus late), repetition, and query specificityall designed to remain interpretable under inspection [37, 38, 39].

Two lightweight refinements extend this scoring to incorporate contextual and temporal factors without sacrificing interpretability. First, a **proximity bonus** of +3 is awarded when the first occurrences of the combinations base

terms co-occur within a short window of approximately 12 tokens in the abstract. This encourages higher ranking for papers where the requested concepts are discussed jointly, a heuristic shown to improve multi-term retrieval precision when used conservatively [32, 40, 39]. Second, a **recency multiplier** introduces gentle time awareness by applying an exponential half-life decay with a lower bound:

$$\text{RecencyMultiplier} = \max\left(0.4,\ 0.5^{\frac{\text{days}}{180}}\right),$$

where *days* denotes the age of the publication relative to the retrieval date. This scaling softly privileges more recent contributions without penalizing older, foundational works. The multiplier never falls below 0.4, ensuring that past literature retains a minimum influence in ranking. If the publication date is missing, the multiplier defaults to 1.0 so that ranking depends exclusively on textual evidence.

The final composite score is computed as:

$$\text{FinalScore} = \left(\text{StrictTextScore} + \text{ProximityBonus}\right) \times \text{RecencyMultiplier}.$$

Because the textual and positional components are integer-valued (subject to the 0.8 synonym factor), fractional components arise only from the temporal weighting, facilitating direct interpretability and debugging of results.

We deliberately adopt this interpretable, rule-based framework over more complex but opaque alternatives such as BM25 variations or neural semantic rankers. Three motivations justify this choice. First, the primary purpose of this thesis is to support transparent, reproducible bibliometric analysis where every numerical outcome can be traced to explicit textual evidence (e.g., the keyword appears twice in the title and once early in the abstract). Second, the data fields involved titles and abstracts are short, well-structured, and already suited to field- and position-aware scoring; heavy probabilistic smoothing or term-saturation modeling would add computational overhead without proportionate gain [37, 27, 38]. Third, under the strict Boolean setup and limited synonym expansion defined earlier, traditional rankers that rely on term frequency normalization or log-scaling (e.g., BM25) yield minimal incremental improvement while obscuring interpretability [41, 27].

The proposed scoring framework remains fully extensible: stronger retrieval models such as BM25F or BERT-based neural re-rankers can be incorporated as optional stages if deeper semantic matching is later required. In this study,

however, we deliberately adopt a *baseline* configurationmeaning a foundational, fully interpretable implementation that establishes a clear reference point for all subsequent enhancements. In information retrieval research, a baseline does not merely represent a simplified model; it defines the minimal but complete system that provides reproducible, auditable behavior against which more sophisticated methods can be evaluated. Such a baseline ensures that any future improvement can be quantified and explained relative to a transparent and stable foundation, rather than against an opaque or shifting target. The baseline presented here therefore prioritizes transparency, controllability, and traceability over algorithmic complexity, providing a defensible benchmark for both methodological comparison and academic reporting.

Finally, this strict-plus-refinement design integrates coherently with the combination-ordering and deduplication policy introduced earlier. Once a document meets the conditions for a longer, more specific combination, it is permanently assigned to that subset and excluded from its sub-combinations. This ensures that all reported counts and ranks reflect genuine intersections rather than inflated overlaps. The resulting mapping from *intent* $\rightarrow$ *criteria* $\rightarrow$ *score* thus forms a self-contained, reproducible framework that combines interpretability, methodological rigor, and analytic flexibility [27, 38, 32, 40].

2.4.4 Comprehensive Analysis and Visualization of Keyword Combinations

Building on the strict matching and scoring defined above, we systematically evaluate *all* non-empty subsets of the user-defined keyword set. For k input terms, this yields $2^k - 1$ combinations that are processed in descending order of length, which prioritizes highly specific intents before progressively relaxing constraints [27, 34]. This topdown traversal mirrors natural search behavior and, crucially, enables unambiguous attribution: once a document satisfies a longer combination, it is assigned there and excluded from its sub-combinations, which prevents double counting and ensures that longer intersections reflect genuine co-occurrence rather than inflated overlaps.

For each subset C, a synonym map is constructed as described in the synonym expansion step. The corpus is then evaluated under the strict conjunctive rule: every term in C, or one of its controlled surface variants, must appear in the title or abstract for the document to qualify. Qualified records receive

an interpretable, section-aware relevance score that aggregates title evidence, early abstract occurrence, repetition counts, and synonym discounts; a fixed combination-length bonus rewards satisfaction of longer queries. Additional contextual refinements, including proximity- and recency-based adjustments, follow the same principles introduced in the strict matching stage, ensuring consistency across all evaluated subsets. The scoring function remains fully additive and auditable, allowing term-by-term and position-by-position inspection.

Outputs are produced at two complementary levels to support transparency and reproducibility. At the record-wise level, the pipeline emits, for every successful match, the responsible combination together with article metadata (title, canonical link, abstract summary) and the computed relevance score. Since combinations are traversed longest to shortest and titles are deduplicated, each article appears exactly once under the most specific combination it satisfies. This record-wise output is designed for auditability, making it straightforward to trace why and where a record was selected. At the coverage level, counts are aggregated for every keyword subset, including those with zero matches, and normalized shares can be computed against a configurable baseline (typically the EEG subset harvested in stage one, or optionally the full corpus) [26]. Combinations are sorted by length and then alphabetically so that relaxation effects are visually and numerically comparable across runs.

Visualization routines are deterministic in ordering and labeling to maintain byte-for-byte reproducibility: the record-wise table can be rendered as a static figure for compact presentation, while the coverage summary is plotted as a bar chart with exact counts annotated and a stable axis layout. The figures themselves are presented and interpreted in Chapter 3. Retaining zero-count combinations in the summary is a deliberate design choice, since it exposes conceptual gaps explicitly and helps identify underexplored intersections [30, 29]. All configuration parameters that affect enumeration, matching, scoring, and visualization are logged alongside the outputs to preserve run-to-run comparability.

To provide a concrete view of how this module operates programmatically, Figure 2.3 illustrates a typical inputoutput flow within the keyword-based filtering and analysis pipeline. The example shows how a user-defined keyword list (e.g., `["machine learning", "brain", "real-time"]`) is expanded, enumerated, and processed through the strict matching and scoring functions described

above. The Python driver script orchestrates this sequence: it normalizes all text fields, computes strict relevance scores for each keyword combination, sorts the results by combination length and score, and produces both tabular and visual outputs. This illustrative figure clarifies the functional workflow of the developed software rather than presenting empirical findings.

```python
# 1. Define the target keywords for combination analysis
keywords = ["machine learning", "brain", "real-time"]

# 2. The DataFrame of articles (df) should be prepared in advance
# Example: df = pd.DataFrame(articles)

# Normalized columns
df["title_lc"] = df["title"].fillna("").str.lower()
df["summary_lc"] = df["summary"].fillna("").str.lower()

# 3. Analyze all keyword combinations and compute relevance scores for all articles
results_df, combo_counts = analyze_combinations_strict_allcombos(df, keywords)

# 4. Sort the results: prioritize longer combinations and higher relevance scores
results_df['combo_len'] = results_df['combo'].apply(lambda x: len(x.split('+')))
results_df = results_df.sort_values(by=['combo_len', 'score'], ascending=[False, False])

# 5. Visualize the number of articles for each combination (including those with zero matches)
plot_combo_counts_all(results_df, combo_counts)

# 6. Filter the table to only show rows with actual articles (non-empty results)
filtered_df = results_df[results_df['title'].notnull()] \
    .sort_values(by=['combo_len', 'score'], ascending=[False, False]) \
    .reset_index(drop=True)

# 7. Display the top 10 most relevant article entries (or export as CSV)
filtered_df[['combo', 'title', 'link', 'score', 'summary']].head(1000)
```

Figure 2.3: Illustrative example of the keyword-based filtering and analysis pipeline in execution. The script defines user-specified keywords, expands them into all non-empty combinations, and applies strict matching and scoring functions to the EEG corpus. The figure demonstrates the programmatic flow from user input to filtered and ranked outputs, highlighting how the developed module translates textual criteria into structured analyses. This visualization is methodological and does not represent analytical results.

2.4.5 Batch Analyses on Strict Subsets

To examine the structure of the EEG-related corpus in greater depth, we implemented batch analyses across a hierarchy of strict subsets. Each subset corresponds to a keyword combination drawn from the predefined list and is generated under the same deterministic and reproducible pipeline. The guiding principle behind this stage is to isolate only those publications in which the selected terms play a central role in the research. A document is included in a subset only if all keywords of that combination, or one of their accepted synonyms derived from the WordNet expansion, appear explicitly in the title

or abstract. This strict Boolean filtering ensures interpretability: the resulting datasets represent core segments of the literature where the chosen terms are genuinely central rather than peripheral or incidental [42, 43].

The filtering process produces single-, double-, and triple-keyword subsets, each representing a different level of conceptual specificity. For a user-defined list of k terms, all non-empty combinations are enumerated ($2^k - 1$ in total) and processed in descending order of length, so that longer combinations are matched first. This ordering prevents double counting, since articles captured under more specific intersections are excluded from their sub-combinations, and it enables meaningful comparison of coverage across levels of specificity. Each subset is stored as a structured table containing metadata such as the arXiv identifier, title, authors, abstract, primary category, publication year, and the computed relevance score. By organizing the output in this way, the system ensures traceability and facilitates both visual and quantitative analyses [44].

All preprocessing and normalization steps are fixed to guarantee comparability across subsets. Texts are lowercased, punctuation is removed using precompiled regular expressions, and tokenization follows the same NLTK configuration applied earlier in the pipeline. Hyphenated tokens are normalized by replacing patterns such as real-time with real time to capture orthographic variations without compromising specificity. **Randomness is entirely controlled through explicit seed initialization,** ensuring that word cloud layouts and text ordering remain stable across runs. Each subsets configuration, including retrieval date, stopword list, keyword set, and random seed, is logged in accompanying metadata files, allowing byte-for-byte reproducibility of every figure and table generated [45].

Two complementary visualization modules are applied to each subset: lexical analysis through word clouds and temporal analysis through publication trends. In the lexical analysis, all text fields from the subset are concatenated and tokenized. Generic English stopwords and domain-trivial words such as EEG, signal, and data are removed so that the resulting word cloud highlights the most semantically informative vocabulary. The size of each word reflects its frequency, providing an immediate qualitative view of the dominant concepts that characterize the subset. In the temporal analysis, publication years are parsed from the ISO 8601-formatted timestamps and aggregated into yearly counts. The results are visualized as bar charts showing the evolution of publication volume over time, with clear axes, grid-aligned bars, and annotated

partial bins for incomplete years. These two analyses complement one another: the word cloud captures the conceptual composition of the subset, while the publication trend reveals its historical trajectory. Although both are derived from the same dataset, they are intentionally generated as separate visualizations to maintain interpretive clarity and avoid overloading a single figure with multiple analytical dimensions [46].

The batch analysis proceeds hierarchically across three levels. Single-keyword subsets serve as broad baselines, revealing the general methodological and application vocabulary that dominates the EEG corpus. Two-keyword subsets represent more focused intersections, often uncovering interdisciplinary bridges between fields, such as computational and clinical neuroscience. Three-keyword subsets form the most restrictive and fine-grained partitions, capturing small but semantically coherent clusters of literature at the frontier of EEG research. By systematically comparing these layers, it becomes possible to observe how conceptual scope narrows as specificity increases and to identify where new interdisciplinary themes have emerged or declined [43].

Beyond the per-combination analyses, an overall strict subset is also constructed by aggregating all records that satisfy at least one keyword combination. This aggregate offers a macroscopic perspective on the EEG corpus under the same strict filtering logic. Its visualizations, a global word cloud and a consolidated publication trend, summarize the dominant conceptual and temporal patterns across the filtered dataset, allowing for direct comparison with the individual combinations. In this way, the overall subset contextualizes the strict analysis within the broader EEG literature and highlights the periods and conceptual clusters that dominate the field as a whole [47].

All batch analyses are scripted in Python using `pandas` for data handling, `matplotlib` for trend plotting, and the `wordcloud` library for lexical visualization. Each run outputs both the static figures and the corresponding numerical tables, including token counts and yearly publication counts, which are automatically stored with descriptive filenames encoding the subset parameters. This architecture enables scalable batch generation across dozens of keyword combinations while maintaining efficiency through vectorized filtering and cached preprocessing. Even exhaustive runs remain computationally manageable, typically completing within minutes.

Unlike probabilistic topic modeling or embedding-based clustering, which introduce latent representations that are difficult to interpret, the deterministic

keyword-based approach was deliberately chosen for its transparency and direct traceability to domain terminology. In EEG-related literature, researchers frequently rely on explicit keyword formulations to indicate methodological focus or application domain. By preserving this explicitness, the proposed framework ensures that every inclusion and exclusion decision can be explained in linguistic terms. Moreover, deterministic filtering avoids the instability associated with stochastic modeling, where small parameter changes may alter topic boundaries or semantic clusters. The present design thus privileges reproducibility and interpretability over abstraction, aligning with the thesis objective of producing a verifiable and explainable bibliometric analysis grounded in domain-relevant language [45, 43].

Overall, the batch analyses on strict subsets constitute a reproducible, hierarchical framework that integrates lexical and temporal perspectives under a unified methodological design. The combination of deterministic filtering, fixed preprocessing, and explicit configuration logging ensures full transparency and comparability across analyses. This methodological rigor provides the foundation for the interpretive discussion presented in Chapter 3, where the resulting lexical and temporal patterns are analyzed in detail [42, 46].

2.5 Category-Based Retrieval and Classification

2.5.1 Retrieving EEG Articles and Exploring Categories

We constructed an EEG-focused corpus by querying the arXiv API with the keyword *EEG* and performing paginated retrieval to exhaustively enumerate all matching records. Pagination parameters were advanced deterministically until the service reported no additional entries, ensuring that harvesting was complete and reproducible for a fixed query date. For each returned Atom entry, we parsed and stored canonical metadata: a normalized article identifier (`arxiv_id`, stripped of any version suffix), the `title`, the `summary` (abstract), the `published` and `updated` timestamps, and the list of arXiv `categories`. Because arXiv allows multiple versions per article, we enforced a one-to-one mapping between scientific contribution and record by retaining only the most recent version of each `arxiv_id`. When multiple versions were present, version resolution was performed by comparing version counters and, when needed, the `updated` time; the surviving record inherits its full metadata so that downstream analyses

reflect the latest state of each paper. **Timestamps were parsed into a consistent ISO 8601 representation to permit temporal filtering and ordering in later analyses.** Records missing essential fields (e.g., an absent category list) were marked explicitly (e.g., `unknown`) so that inclusion decisions could be taken transparently at the modeling stage rather than silently during ingestion [44, 48].

In alignment with arXivs subject taxonomy, we defined the target label for supervised learning as the primary category, operationalized as the first element of a papers category list. This choice follows arXivs convention that the leading subject tag encodes the main topical placement of the submission, while additional tags capture cross-listings. Using the first category therefore provides a single, unambiguous label per paper, avoiding multi-label complications and simplifying evaluation. At the same time, we preserved the full category list in the stored metadata so that alternative formulations (e.g., multi-label classification or cross-list analysis) remain possible without reharvesting the corpus [49]. The combination of categorical annotations and free text (title and abstract) enables a joint treatment in which topical structure is anchored to lexical evidence: category distributions can be explored quantitatively, and predictive models can be trained to map document text to its primary subject. Finally, by logging the exact query, pagination settings, retrieval date, and software configuration, the construction procedure remains fully reproducible; rerunning the same pipeline on a later date yields an updated snapshot that is comparable by construction, differing only in additions and revisions introduced on arXiv since the previous harvest [50, 51].

2.5.2 Filtering Valid Categories (Minimum Support)

To mitigate severe class imbalance and to ensure that each fold in cross-validation contains representative samples from all classes, we applied a minimum support criterion to the primary category labels. Concretely, only categories with at least 100 instances in the retrieved set were retained for modeling. Items lacking any category assignment were marked as `unknown` and excluded from the learning task. The retained classes form a stable working set that captures the most prevalent EEG-related subjects on arXiv, such as *Electrical Engineering and Systems ScienceSignal Processing*, *Computer ScienceMachine Learning*, *HumanComputer Interaction*, *Computer Vision and Pattern Recognition*, and *Quanti-*

tative BiologyNeurons and Cognition.

For brevity in figures and tables, we use arXivs standard abbreviations for these subjects; the mapping is preserved in our metadata and reiterated here on first mention:

- **`eess.SP`**: Electrical Engineering and Systems ScienceSignal Processing
- **`cs.LG`**: Computer ScienceMachine Learning
- **`cs.HC`**: Computer ScienceHumanComputer Interaction
- **`cs.CV`**: Computer ScienceComputer Vision and Pattern Recognition
- **`q-bio.NC`**: Quantitative BiologyNeurons and Cognition

These abbreviations are used consistently in all subsequent quantitative summaries and error analyses; full names are provided on first occurrence to avoid ambiguity.

This filtering has several methodological consequences. First, by removing extremely rare classes, it reduces the variance of fold-wise performance estimates and prevents degenerate stratified splits in which a minority class may be absent from a training or validation fold [52]. Second, it constrains the hypothesis space to the dominant topical axes present in the EEG corpus, whichwhile limiting coverage of the long tailimproves the interpretability and comparability of model performance across methods [53]. Third, it stabilizes optimization for linear and tree-based learners by avoiding regimes where the effective sample size per class is too small for reliable parameter estimation or for meaningful early-stopping signals during boosting [54, 55].

The choice of a 100-instance threshold reflects a pragmatic balance between inclusiveness and statistical reliability: it is high enough to maintain adequate per-class representation within 5-fold stratified cross-validation (so that each fold receives a non-trivial number of samples per class), yet low enough to retain the principal EEG-related subject areas that dominate the corpus. Sensitivity to this threshold can be assessed, if desired, by repeating the pipeline at neighboring cutoffs (e.g., 75 or 150) and examining the stability of relative model rankings and macro-averaged metrics; in our setting, the 100-instance criterion provided a robust operating point without materially altering the set of leading categories. Finally, all excluded categories and their counts are preserved in the exploratory summaries (outside the modeling set), ensuring that decisions about inclusion remain transparent and reproducible.

2.5.3 Feature Preparation and Models

For each article, the title and summary fields were concatenated into a single document and subjected to a light yet systematic normalization. The pipeline removes URLs, email addresses, simple HTML tags, and both inline and display LaTeX fragments; applies Unicode NFKC normalization; lowercases tokens; simplifies spurious symbol sequences without altering meaningful hyphenations (e.g., *eeg-based*); and collapses whitespace. These operations are intentionally conservative: they suppress formatting artifacts that inflate the vocabulary while preserving lexical cues that are informative for topical classification.

Documents are represented with TF–IDF over word n-grams (unigrams and bigrams). Unless otherwise stated, the vectorizer uses `max_features` = 30,000, `min_df` = 5, `max_df` = 0.85, and `sublinear_tf` = `True`, with English stopword removal enabled. This specification balances expressivity and computational efficiency: pruning very rare and very frequent terms reduces noise and dimensionality, while bigrams retain short phrasal signals common in scientific abstracts. For tree-based learners, which train more efficiently on dense, moderate-dimensional inputs, the TF–IDF space is further mapped by Truncated Singular Value Decomposition (Truncated SVD) with 150 components and then scaled with `StandardScaler(with_mean = False)`. Truncated SVD provides a low-rank semantic projection of the sparse vectors, improving stability and runtime for gradient-boosted decision trees, while preserving global term–document structure [56]. The choice of 150 components reflects an empirically validated speedquality trade-off in our environment and can be widened if computational resources permit.

To prevent information leakage, both TF–IDF and, where applicable, the SVD transformation are fitted exclusively on the training portion of the data and subsequently applied by transformation to validation and test splits or folds. No statistics from held-out data are used during feature induction.

The comparative study covers linear, generative, and ensemble approaches on a common feature space. Linear baselines include logistic regression (one-vs-rest, `solver="saga"`, `class_weight="balanced"`) and linear SVC (hinge-loss SVM with `class_weight="balanced"`). Generative baselines comprise multinomial naive Bayes and complement naive Bayes. As a non-linear bag-of-words reference, a random forest with 300 estimators and `class_weight="balanced"` is included, acknowledging its limitations in very high-dimensional sparse

regimes. For boosted trees, the implementation relies on scikit-learns histogram-based gradient boosting classifier in the pipeline TF–IDF → TruncatedSVD(150) → StandardScaler(with_mean=False) → gradient boosting [57].

Model selection follows stratified 5-fold cross-validation on the training portion, optimizing macro F1 as the selection criterion. Stratification maintains class prevalence across folds, which is important under residual imbalance. Macro F1, computed as the unweighted mean of per-class F1 scores, assigns equal importance to all classes and therefore offers a more equitable objective than accuracy in this setting [27]. For boosted trees, hyperparameters are tuned with successive halving (`HalvingGridSearchCV`), which evaluates a broad pool under limited resources, prunes weak configurations early, and allocates additional resources to promising candidates [58]. Early stopping inside the boosting routine (`early_stopping = True`, `validation_fraction = 0.1`, `n_iter_no_change = 10`) terminates training when validation improvements plateau. The search space is a compact grid over learning rate, number of boosting iterations, maximum depth, and L_2 regularization; when runtime budgets are tight, a reduced grid with more aggressive halving can be used. For logistic regression and linear SVC, regularization strength is tuned over a small set of C values under 5-fold CV; to reduce per-fold cost without materially degrading signal, chi-square feature selection with $k = 20{,}000$ is applied during tuning only [59], after which the selected configuration is refit on the training portion and evaluated on the held-out test split in the same tuned feature space. All model implementations and validation routines rely on the `scikit-learn` library [60].

3

Results and Discussion

3.1 Exploratory Analyses

3.1.1 Lexical Salience in EEG Literature

The resulting word clouds provide an intuitive lexical map of the EEG research domain at two levels of textual abstraction (Fig. 3.1). The abstract-based cloud prominently features tokens such as *eeg*, *analysis*, *signal*, *model*, *network*, *classification*, *data*, *feature*, and *method*. These words reflect the methodological diversity and technical focus of EEG-related studies, where algorithmic analysis and signal modeling dominate the discourse. In contrast, the title-only cloud emphasizes compact descriptors such as *recognition*, *decoding*, *deep*, *learning*, and *brain-computer interface*, indicating that authors prioritize short, high-impact terms related to machine learning and applied neurotechnology when framing their contributions.

Quantitative baseline. Before presenting the word-clouds, we computed a compact frequency baseline on the cleaned corpus (lowercasing, punctuation removal, English stopwords via NLTK [28]) using Pythons `Counter`. The most frequent tokens confirm a methodological core around *signal*, *detection*, *analysis*, *data*, and *network* (e.g., *signal* $\approx$ 329, *detection* $\approx$ 288, *analysis* $\approx$ 279, *data* $\approx$ 274), with application-oriented terms such as *seizure*, *sleep*, and *motor* also ranking highly. As a focused check, we quantified a small target set anchoring the retrieval pipeline: *EEG* (1413), *brain* (381), *BCI* (103), *signal* (122), *detection* (288). In total, 2339 of 3599 records include at least one target term (64.99%). These

counts verify lexical coherence and provide the quantitative background for the visual salience results that follow.

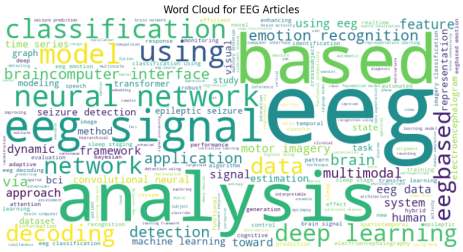
This divergence between title and abstract vocabulary is expected and instructive. Titles are optimized for visibility and indexing, often compressing entire research directions into minimal token sets, whereas abstracts provide a more elaborate description of methods, datasets, and experimental contexts. The lexical separation between the two thus visualizes a well-known pattern in scientific writing: conciseness at the headline level versus descriptive density at the abstract level.

Beyond their descriptive appeal, these figures serve several analytic purposes. First, they identify method families and application domains (such as *seizure detection*, *artifact removal*, and *multimodal fusion*) that are sufficiently frequent to warrant stratified quantitative analysis in later sections. Second, they assist in the construction of domain-specific stop-lists by highlighting overrepresented generic tokens most notably EEG itself that should be down-weighted in subsequent statistical analyses. Third, re-running the same pipeline on strict Boolean subsets, such as *"seizure AND EEG AND detection"*, produces sharply focused clouds where clinical terms (*onset*, *patient*, *sensitivity*, *false-positive*) become dominant, indicating topical convergence within subfields.

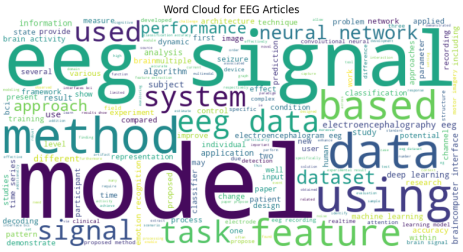
Since these clouds are derived from the latest arXiv snapshot, they inherently reflect the evolving vocabulary of the EEG community. As new articles are published or existing records updated, term frequencies will shift accordingly. For reproducibility and longitudinal comparison, the underlying token frequency tables and cleaning configurations are versioned alongside the visual outputs, ensuring that later analyses can measure lexical drift over time.

Lexical Frequency and Target-Word Coverage

Before visualizing lexical salience through word clouds, a frequency-based inspection of the cleaned corpus was performed to establish the dominant vocabulary and verify lexical consistency. Following the preprocessing pipeline described in Chapter 2, all titles and abstracts were lowercased, stripped of punctuation, and filtered for English stopwords using the NLTK corpus [28]. Word frequencies were then computed using Pythons `Counter` module, producing a quantitative overview of the corpus vocabulary and serving as an internal diagnostic step prior to visualization.



(a) Abstract-based word cloud. Font size encodes frequency; placement follows randomized spiral packing.



(b) Title-only word cloud, produced with identical pipeline parameters.

Figure 3.1: Lexical salience in the EEG arXiv corpus at two textual levels. Font size encodes frequency; placement follows randomized spiral packing.

Table 3.1 reports the fifty most frequent words in the cleaned corpus. **The overall distribution reveals a methodological core dominated by tokens such as *signal*, *deep*, *analysis*, *eegbased*, *recognition*, *data*, and *network*, complemented by application-oriented terms like *seizure*, *emotion*, and *sleep*.** This lexical structure confirms that EEG-related research is conceptually anchored at the intersection of signal processing, pattern recognition, and neural decoding. The presence of high-frequency technical markers such as *deep*, *network*, and *classification* reflects the widespread adoption of machine learning frameworks across neuroengineering studies, while recurring biomedical tokens such as *seizure*, *sleep*, and *emotion* point to clinically motivated applications of these computational tools.

Furthermore, the lexical balance between algorithmic and physiological terms illustrates how the EEG domain has evolved into a hybrid research field, where traditional biomedical signal analysis is increasingly integrated with data-driven modeling approaches. The recurrence of compound forms (e.g., *eegbased*, *braincomputer*, *featureextraction*) also highlights the corpuss consistent technical language, validating the effectiveness of the normalization and tokenization

pipeline in preserving semantically meaningful multiword constructs. Overall, this frequency analysis provides an interpretable, quantitative foundation for the subsequent visual exploration of lexical salience and thematic structure across the EEG literature.

Table 3.1: Top 50 most frequent words in the cleaned corpus of titles and abstracts.

Word	**Count**	**Word**	**Count**
signal	329	deep	295
detection	288	analysis	279
data	274	eegbased	238
network	237	based	235
recognition	230	networks	222
decoding	214	model	202
emotion	182	seizure	165
models	156	braincomputer	153
sleep	142	motor	134
convolutional	128	approach	125
framework	123	attention	123
imagery	122	signal	122
interface	121	multimodal	120
time	118	machine	108
epileptic	104	human	104
bci	103	speech	102
functional	97	feature	92
connectivity	92	interfaces	90
study	90	graph	89
prediction	87	towards	83
visual	82	estimation	81

To complement this global vocabulary profile, we also quantified the absolute and relative frequencies of manually defined target words *EEG*, *brain*, *BCI*, *signal*, and *detection* which serve as conceptual anchors of the retrieval pipeline. Table 3.2 summarizes these occurrences. Overall, 2339 out of the 3599 total articles (approximately 65%) contain at least one of these key terms in their title or abstract. The dominance of *EEG* and *brain* confirms that the dataset

is firmly centered on neurophysiological studies, while the strong presence of *signal* and *detection* reflects the methodological emphasis on signal processing and classification tasks.

Table 3.2: Occurrences of manually defined target words in the retrieved corpus.

Target Word	**Occurrences**
EEG	1413
Brain	381
BCI	103
Signal	122
Detection	288
Articles matching ≥1 target word	**2339 (64.99% of 3599)**

These lexical distributions validate that the retrieval pipeline accurately captured the intended EEG-related literature. The alignment between the dominant vocabulary and the target-word set indicates that the corpus is both thematically and lexically consistent, with minimal off-domain noise. This quantitative analysis provides the foundation for the following subsection, where lexical salience is visualized through word clouds to qualitatively illustrate the conceptual structure of the EEG research landscape.

3.1.2 Temporal Dynamics of EEG Research on arXiv

The yearly publication trend derived from the `published` timestamps is shown in Fig. 3.2. The bar chart reveals a long period of relatively low publication activity, spanning roughly from the early 2000s until the mid-2010s, followed by a sustained and rapid increase in the number of EEG-related works. **The steep rise beginning after 2015 aligns closely with the widespread adoption of deep learning and the growing availability of open EEG datasets,** both of which substantially expanded research output and accessibility.

In the most recent years, the publication rate reaches historically high levels, indicating consolidation of EEG research as a multidisciplinary field encompassing signal processing, machine learning, neuroscience, and biomedical engineering. The plateau observed around 2022-2024 suggests a stabilization in publication frequency, likely reflecting a maturation of methodologies and

a shift from proof-of-concept studies toward applied and comparative research frameworks.

From an analytical perspective, this figure provides two key functions. First, it highlights structural inflection points that motivate further explorationspecifically, the post-2015 acceleration that coincides with algorithmic revolutions in neural decoding and BCI applications [61, 62]. Second, it serves as a denominator for all subsequent normalized analyses in this thesis, allowing future sections to express topic-specific trends (e.g., seizure detection or multimodal fusion) as proportions of the total EEG literature per year. *Note.* The yearly trend is descriptive (not inferential). It serves as a baseline for later trend normalization, cross-subset comparison, and correlation with topic-specific evolutions.

The overall shape of the curve is consistent with previous observations in computational neuroscience, where publication volumes mirror the diffusion of machine learning and open science practices. Because timestamps are normalized to UTC and duplicates removed, these growth dynamics can be directly compared across future snapshots of the arXiv corpus without re-running the entire retrieval pipeline.

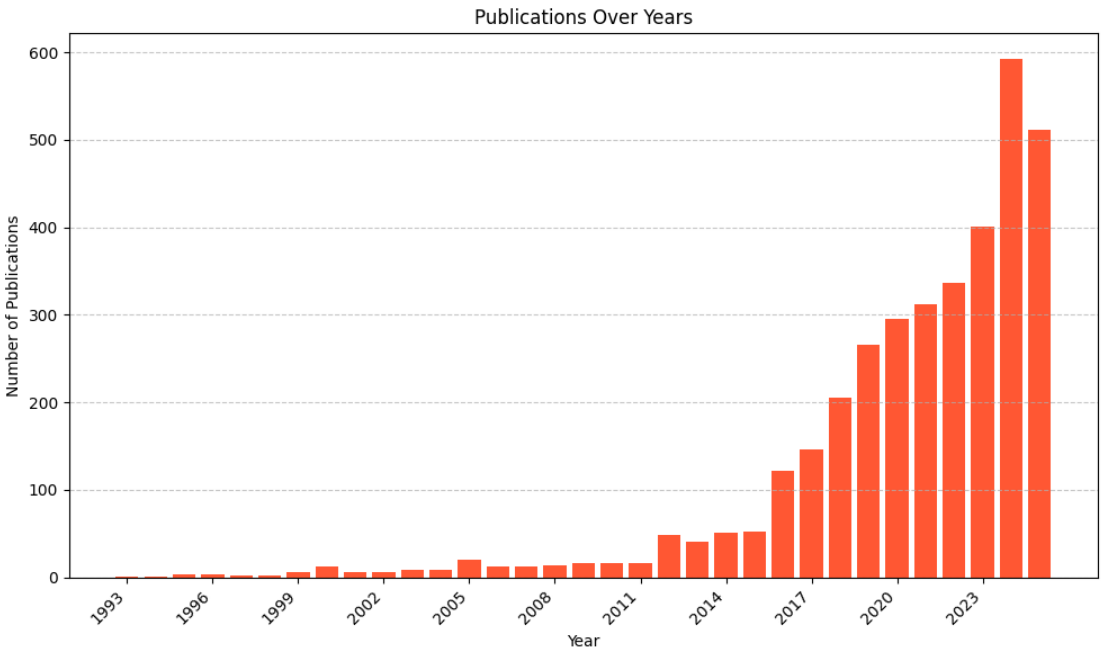


Figure 3.2: Yearly number of EEG-related arXiv publications derived from the `published` timestamp. Bar height encodes count; color is aesthetic only; grid lines aid visual comparison. The most recent year may be a partial bin depending on acquisition date.

3.2 Coverage and Percentage within the EEG Baseline

To quantify how strongly each keyword and keyword combination is represented in the corpus, we compute coverage counts and percentage shares over an EEG-restricted baseline. Concretely, we first identify the subset of documents in which any sanctioned synonym of "EEG" appears in the title or abstract (cf. the synonym expansion step), and we take this subset as the denominator for all percentage calculations reported here. Let N_{EEG} denote the size of this baseline. For each seed keyword k, we then count the number of baseline documents in which k (or one of its approved variants) is present, and report both the raw count and the percentage $\frac{\#(k)}{N_{\mathrm{EEG}}} \times 100$. Likewise, for each non-empty keyword combination C (processed as a strict conjunction), we report $\#(C)$ and $\frac{\#(C)}{N_{\mathrm{EEG}}} \times 100$. Because combinations are evaluated under strict matching and deduplicated by title upstream, these shares reflect non-inflated coverage: a document assigned to a longer combination is not re-counted for its sub-combinations.

Table 3.3 summarizes per-keyword coverage within the EEG baseline, while Table 3.4 reports the same statistics for all combinations, ordered by decreasing length and then alphabetically. Interpreting the tables alongside the bar chart in Fig. 3.3 provides a consistent narrative: broad terms (e.g., *brain*) tend to occupy a large fraction of the EEG subset, whereas multi-term conjunctions (e.g., *machine learning + brain + real-time*) are rarer but more specific.

Table 3.3: Per-keyword coverage within the EEG baseline (N_{EEG} = 3470 out of N_{all} = 3550 total documents).

Keyword	Count	Share (%)
brain	2006	57.81
machine learning	420	12.10
real-time	280	8.07

Table 3.4: Coverage for all keyword combinations within the EEG baseline (N_{EEG} = 3470). Combinations are strict conjunctions; documents are deduplicated by title.

Combination	Length	Count	Share (%)
machine learning + brain + real-time	3	24	0.69
machine learning + brain	2	230	6.63
brain + real-time	2	167	4.81
machine learning + real-time	2	41	1.18

Zero-coverage rows, when present, are intentionally retained to expose genuine gaps under the strict conjunctive criterion. This joint presentation (counts, shares, and bars) makes it straightforward to distinguish underexplored intersections from well-covered themes, while keeping all denominators explicit and reproducible [27, 34, 30].

For completeness, it is also recommended a sensitivity view in which the percentages are computed over the *full* retrieved corpus (i.e., using N_{all} as the denominator) rather than the EEG-restricted baseline. This comparison clarifies whether a high share within EEG is driven by overall ubiquity of the term or by a genuine concentration inside EEG-focused work.

3.3 Keyword Combination Results

The record-wise results table (Fig. 3.3) provides article-level transparency for the entire evaluation. Each bar lists the longest satisfied combination, title, link, abstract summary, and the interpretable relevance score. Because traversal proceeds from longest to shortest and titles are deduplicated, records appear exactly once under their most specific qualifying intent. Higher scores consistently reflect strong lexical alignment, for example, terms appearing in the title and early abstract, with additional reinforcement from repetition and proximity. For

the three-term combination *machine learning* + *brain* + *real-time*, the top-ranked articles achieve scores above 25, indicating dense, early, and multi-contextual use of all three concepts, which is a strong signal of genuine topical integration.

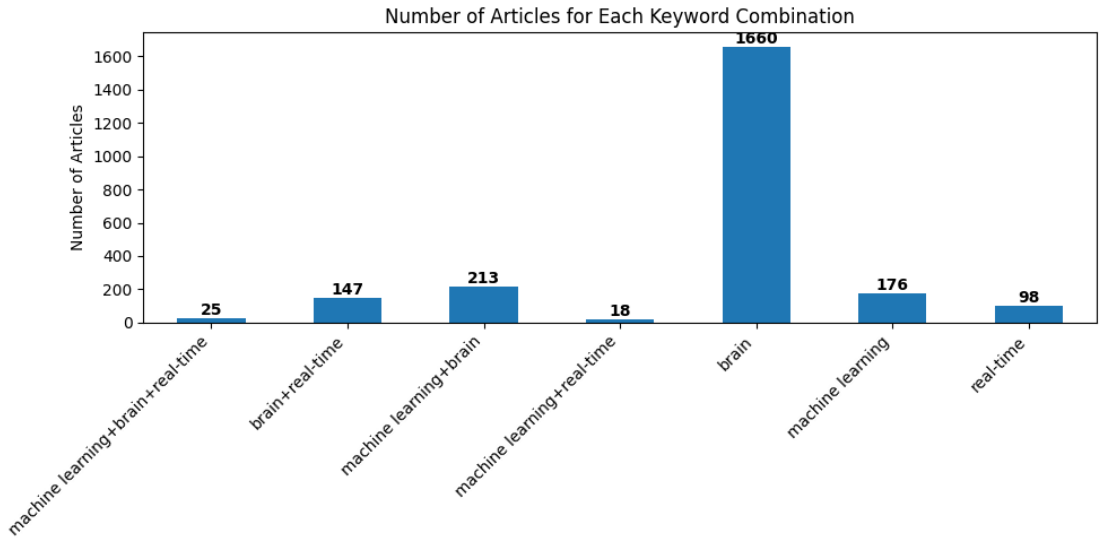


Figure 3.3: Row-wise results table of matched articles. Each row reports the keyword combination, article title, link, computed relevance score, and a short summary. Documents are assigned to the longest satisfied combination to avoid duplication, and higher scores indicate stronger alignment with query intent.

The coverage summary(the table with number of articles for each keyword combination)(Fig. 3.4) aggregates counts for every subset, including those with zero matches, and orders the combinations by specificity and then alphabetically. The distribution confirms the expected but informative pattern: general descriptors dominate the corpus, with *brain* alone accounting for more than 1600 articles, whereas stricter, multi-term conjunctions are comparatively rare, for example only 25 items for *machine learning* + *brain* + *real-time*. This skew captures both the breadth of EEG-related research around broad terms and the relative scarcity of works that explicitly combine advanced themes such as real-time machine learning on brain data. Retaining zero-match subsets exposes conceptual gaps directly, which is useful for shaping future search strategies and for identifying intersections where targeted reviews or dataset curation could be most impactful [30, 29].

Taken together, row-wise results table and the table with number of articles for each keyword combination, fulfill complementary roles. The table enables auditable, record-level inspection of why a paper was selected and how strongly

it aligns with the query intent, while the bar chart offers a macroscopic view of topical coverage and scarcity across the entire combination lattice. Since both outputs derive from deterministic enumeration and the same strict scoring function, every count, percentage, and score is traceable to explicit, reproducible rules [27, 34, 32, 33]. This pairing of transparency and breadth makes the approach well suited to bibliometric analysis, where quantitative rigor and qualitative justification are both essential.

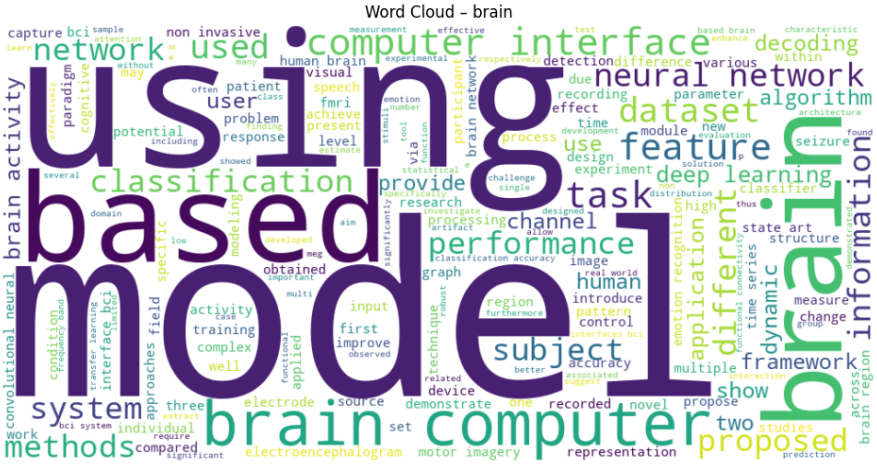
	combo	title	link	score	summary
0	machine learning+brain+real-time	Real-time Noise Detection and Classification i...	http://arxiv.org/abs/2509.26058v1	29.884698	Electroencephalogram (EEG) artifact detection ...
1	machine learning+brain+real-time	CognitiveArm: Enabling Real-Time EEG-Controlle...	http://arxiv.org/abs/2508.07731v1	27.937471	Efficient control of prosthetic limbs via non-...
2	machine learning+brain+real-time	PyNoetic: A modular python framework for no-co...	http://arxiv.org/abs/2509.00670v1	23.961815	Electroencephalography (EEG)-based Brain-Compu...
3	machine learning+brain+real-time	EEG-Based Cognitive Load Classification During...	http://arxiv.org/abs/2509.14056v1	23.687900	Brain computer interfaces enable real-time mon...
4	machine learning+brain+real-time	Detecting Reading-Induced Confusion Using EEG ...	http://arxiv.org/abs/2508.14442v1	16.162676	Humans regularly navigate an overwhelming amou...
...	...	...	...	...	...
995	brain	Perturbing a Neural Network to Infer Effective...	http://arxiv.org/abs/2307.09770v1	4.800000	Identifying causal relationships among distinc...
996	brain	An LSTM Based Architecture to Relate Speech St...	http://arxiv.org/abs/2002.10988v1	4.800000	Modeling the relationship between natural spee...
997	brain	Multivariate Empirical Mode Decomposition of E...	http://arxiv.org/abs/2206.00926v1	4.800000	In this study, the Multivariate Empirical Mode...
998	brain	Neural modulation enhancement using connectivi...	http://arxiv.org/abs/2204.01087v3	4.800000	Emotion regulation plays a key role in human b...
999	brain	Graph Neural Networks in EEG-based Emotion Rec...	http://arxiv.org/abs/2402.01138v5	4.800000	Compared to other modalities, EEG-based emotio...

Figure 3.4: Number of articles for each keyword combination, sorted by specificity and length. The visualization highlights the dominance of general terms such as *brain* and the scarcity of multi-term intersections (e.g., *machine learning* + *brain* + *real-time*).

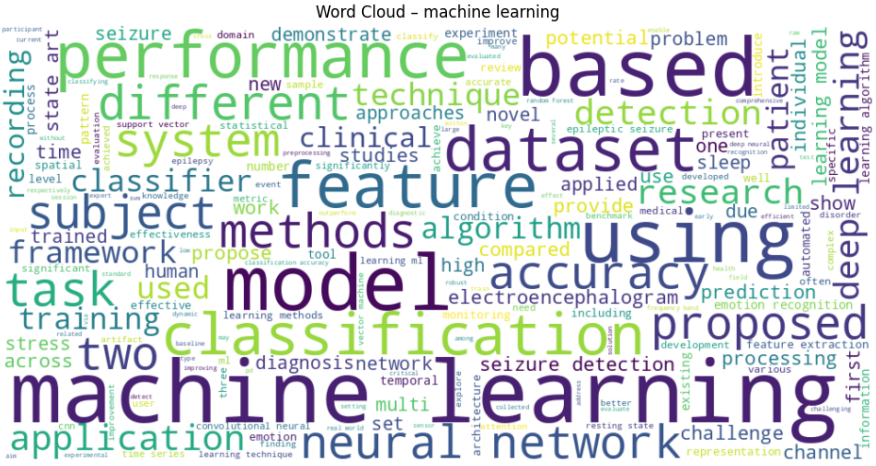
3.4 Batch Analyses on Strict Subsets

The first stage evaluates single keyword subsets under strict filtering. For *brain*, the word cloud emphasizes "model," "using," "based," "classification," and "computer interface," together with "neural network," "dataset," and "performance," reflecting a strong orientation toward computational modeling and brain–computer interaction (Fig. 3.5a). The publication trend shows steady growth since the early 2000s, with a clear acceleration after 2015 and a peak in 2024 (Fig. 3.6a), confirming this as one of the most consolidated domains of EEG research. For *machine learning*, the word cloud highlights "classification," "feature," "dataset," and "performance," while terms like "seizure detection" and "clinical" reveal its translational role (Fig. 3.5b). Unlike *brain*, the trend for *machine learning* is recent: negligible before 2017 but sharply increasing in the last five years (Fig. 3.6b). This reflects the diffusion of deep learning methods and growing computational capacity, enabling more ambitious EEG applications. For *real-time*, the word cloud is dominated by "system," "accuracy," "detection,"

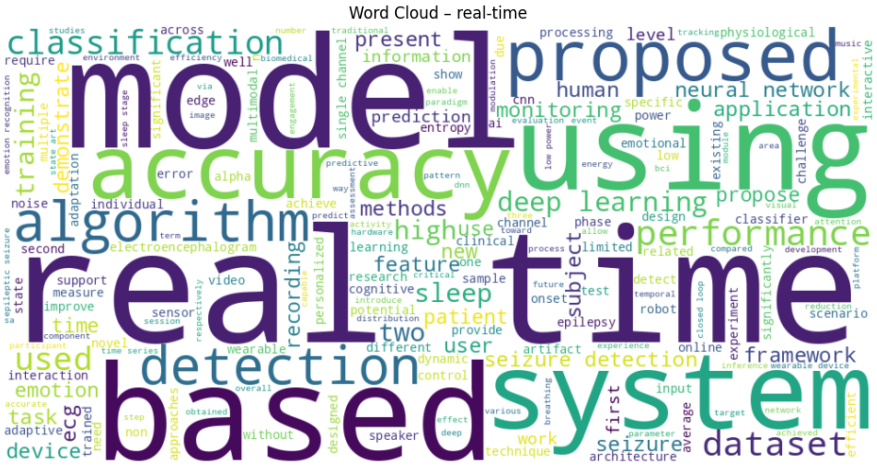
and “device,” pointing to applications in online and wearable scenarios where latency and efficiency are crucial (Fig. 3.5c). Its trend begins after 2012 and accelerates after 2020 (Fig. 3.6c), consistent with the rise of wearable EEG devices and closed-loop systems. Taken together, these subsets highlight complementary perspectives. The word clouds expose the conceptual vocabulary of each area (Fig. 3.5), while the trends show their temporal evolution (Fig. 3.6). In brief, *brain* represents a long-standing and high-volume area, *machine learning* a more recent methodological wave, and *real-time* an application-driven innovation gaining momentum.



(a) Word Cloud for brain

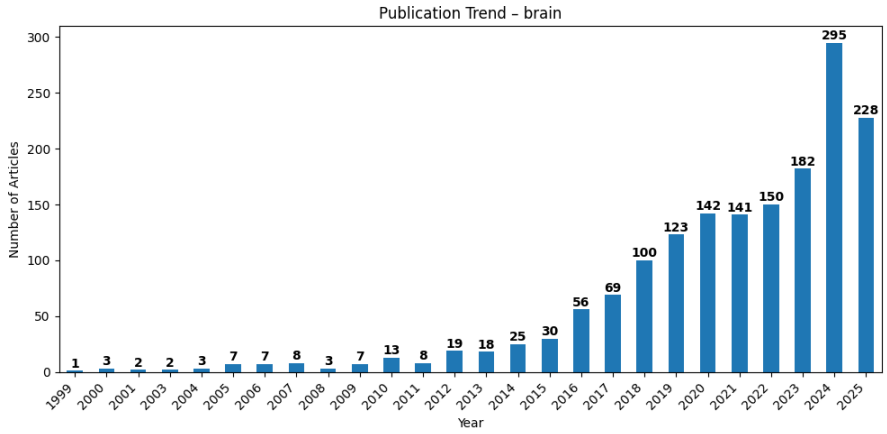


(b) Word Cloud for machine learning

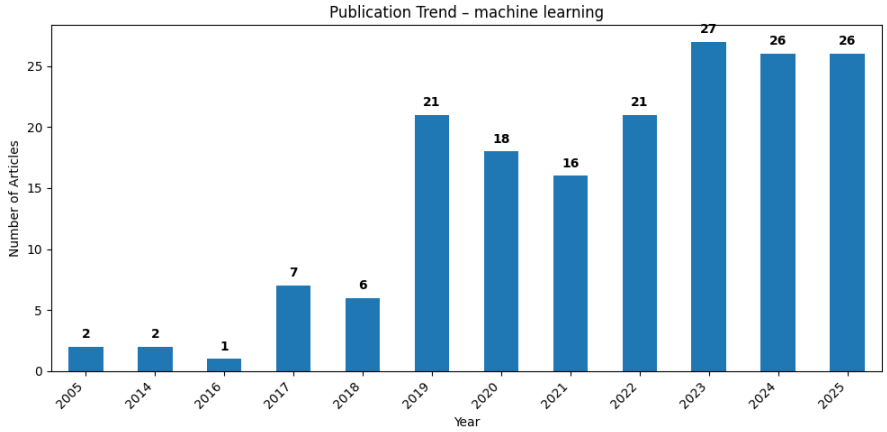


(c) Word Cloud for real-time

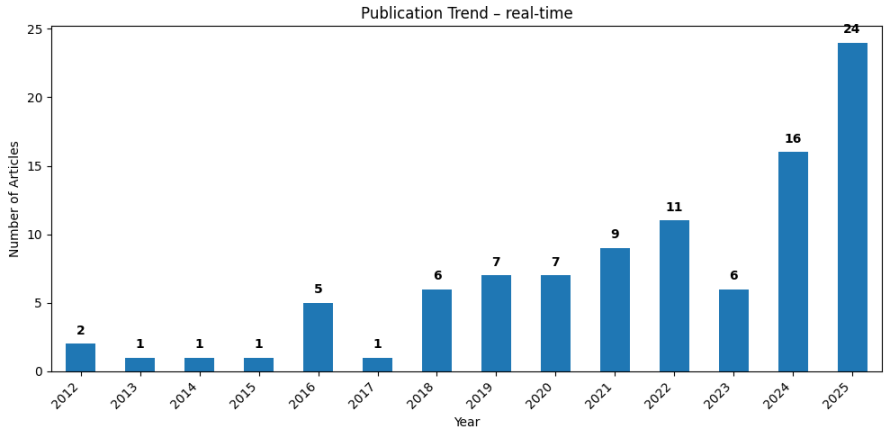
Figure 3.5: Lexical salience for strictly filtered single-keyword subsets. Larger terms indicate higher frequency.



(a) Publication Trend for *brain*



(b) Publication Trend for *machine learning*



(c) Publication Trend for *real-time*

Figure 3.6: Yearly distribution of strictly filtered single-keyword subsets. Each chart illustrates how publication volume evolved over time for a single conceptual focus.

The second stage evaluates strict subsets formed by two- and three-keyword combinations. Compared to single keywords, these combinations yield smaller but more specialized subsets, where the lexical emphasis becomes sharper and the terminology reflects interdisciplinary intersections.

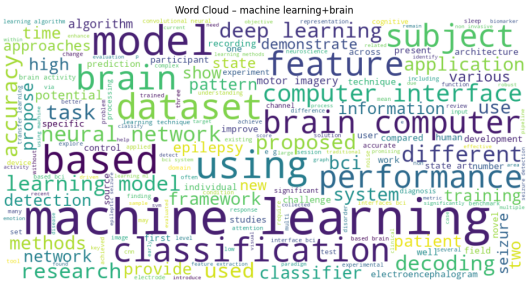
For the pair *machine learning* + *brain*, the word cloud (Fig. 3.7a) highlights terms such as "model," "dataset," "classification," and "neural network," but also application-oriented terms like "epilepsy" and "patient," indicating the clinical relevance of this subset. Its publication trend (Fig. 3.8a) reveals that articles started appearing more consistently after 2017, with a clear acceleration after 2020 and peaks in 2024. This confirms the strong momentum of EEG research at the intersection of computational methods and brain science.

The pair *brain* + *real-time* emphasizes terms like "system," "control," "interface," and "device" in the word cloud (Fig. 3.7b), reflecting its focus on brain–computer interaction and closed-loop systems. Its publication trend (Fig. 3.8b) demonstrates a notable surge after 2020, culminating in a steep rise by 2025. This trajectory shows how the practical deployment of EEG technologies is increasingly oriented toward real-time applications, enabled by wearable devices and low-latency processing [63].

For *machine learning* + *real-time*, the word cloud (Fig. 3.7c) highlights "monitoring," "seizure detection," and "wearable," suggesting integration of adaptive ML algorithms into continuous health monitoring. The publication trend (Fig. 3.8c) is sparser, with only a handful of articles each year, but the presence of peaks in 2020 and 2024 signals a growing though still emerging area.

Finally, the strictest subset, combining all three keywords (*machine learning* + *brain* + *real-time*), yields a word cloud (Fig. 3.7d) dominated by "BCI," "system," "device," "accuracy," and "performance," pointing to highly specific applications in intelligent braincomputer interfaces. Its publication trend (Fig. 3.8d) shows a steady progression since 2018, with gradual growth up to 2025. This indicates that although still limited in volume, the triple combination reflects the frontier of EEG research where computational methods, neural applications, and real-time deployment converge.

Together, these results illustrate how moving from single keywords to pairwise and triple combinations narrows the literature from broad conceptual fields to specialized niches. The two-keyword subsets reveal interdisciplinary bridges-such as clinical ML applications and real-time control systemswhile the three-keyword subset isolates the cutting edge of portable and intelligent BCI research.



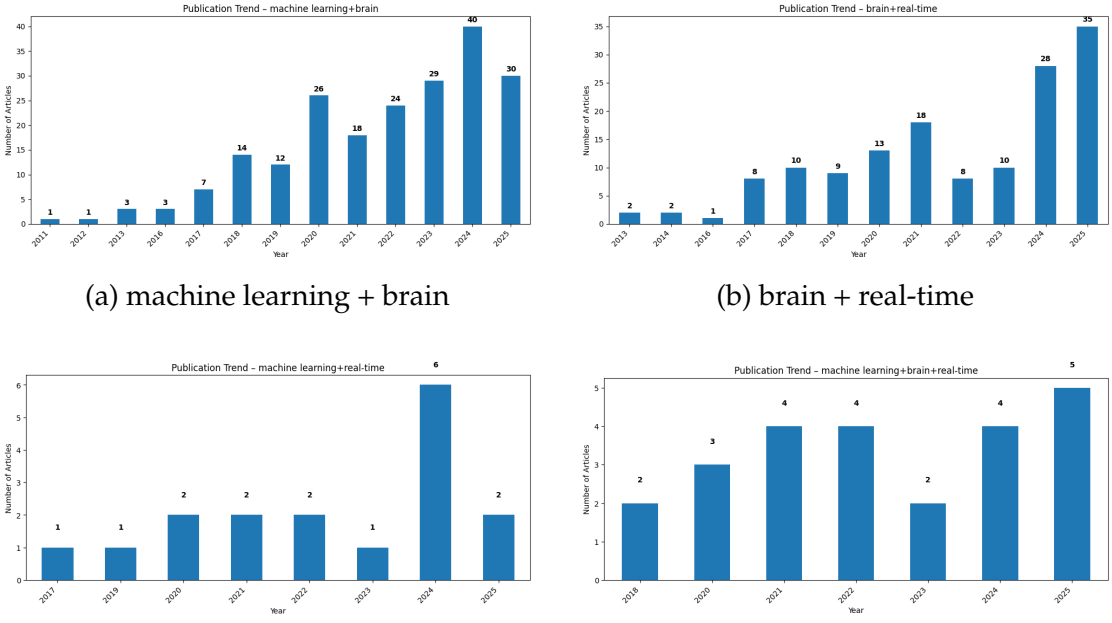
(a) machine learning + brain

(b) brain + real-time

(c) machine learning + real-time

(d) machine learning + brain + real-time

Figure 3.7: Word clouds for strictly filtered two- and three-keyword subsets.



(a) machine learning + brain

(b) brain + real-time

(c) machine learning + real-time

(d) machine learning + brain + real-time

Figure 3.8: Publication trends for strictly filtered two and three keyword subsets.

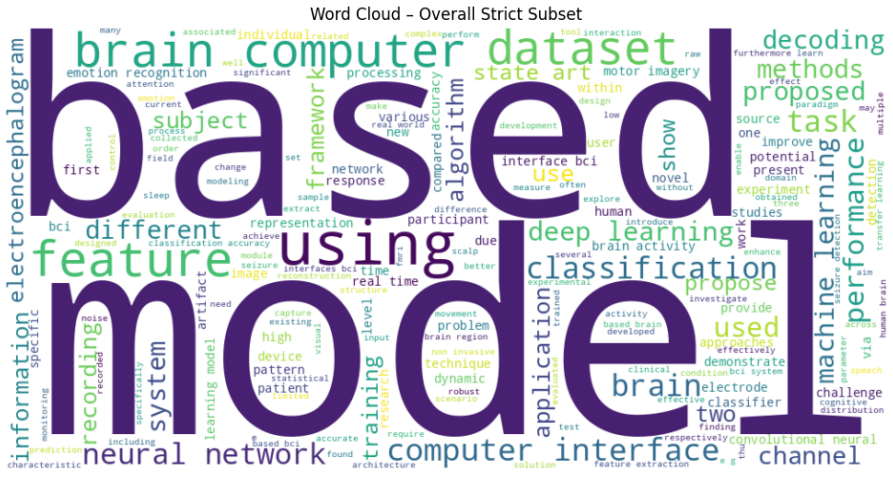
The final stage of the analysis considers the overall strict subset, which aggregates all articles that satisfied at least one keyword combination. The word cloud (Fig. 3.9a) is dominated by terms such as "model," "based," "using," "feature," "classification," and "dataset," indicating that methodological contributions and data-driven approaches form the conceptual backbone of this filtered EEG literature. Alongside these terms, concepts like "brain," "computer interface," and "neural network" confirm that braincomputer interaction and machine learning remain central themes even when all combinations are merged.

The publication trend for the overall subset (Fig. 3.9b) displays a consistent increase since 2010, with a sharp acceleration after 2015. The number of strictly filtered articles reaches more than two hundred per year after 2020, peaking in 2024 with over 260 publications. This trajectory indicates that EEG-related research explicitly involving the selected keywords has undergone rapid expansion in recent years, reflecting both technological advances in data acquisition and computational methods, and growing interest in real-time and application-oriented contexts. The consolidated trend also shows that the strict filtering strategy, while highly selective, captures the momentum of the broader field and highlights how specialized intersections have become mainstream topics in contemporary EEG research.

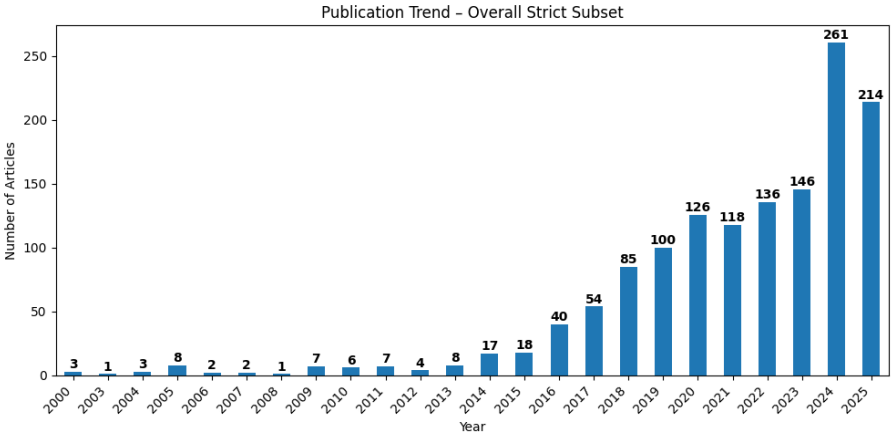
3.5 Category-Based Analyses and Classification

3.5.1 Category Distribution in the EEG arXiv Corpus

Figure 3.10 summarizes the empirical distribution of arXiv subject categories associated with EEG-related submissions (Top-10 by frequency). In line with expectations derived from prior surveys of EEG research, **the corpus exhibits marked concentration in `eess.SP` (Signal Processing) and `cs.LG` (Machine Learning), followed by contributions in `cs.HC` (Human–Computer Interaction), `cs.CV` (Computer Vision), and `q-bio.NC` (Neurons and Cognition).** This pattern is consistent with the methodological centrality of signal processing (filtering, feature extraction, denoising) and learning-based inference (classification, decoding) in EEG pipelines, as reported in related work. The long tail of infrequent categories remains present but was excluded from the modeling set by the minimum support criterion (cf. Materials & Methods), thereby avoiding high-variance estimates for extremely rare classes.



(a) Word Cloud — Overall Strict Subset



(b) Publication Trend — Overall Strict Subset

Figure 3.9: Lexical and temporal characterization of the overall strict subset. (a) The word cloud visualizes the most frequent terms across all articles that satisfied at least one keyword combination, showing that methodological and data-driven vocabulary (*model*, *based*, *feature*, *classification*) dominates the corpus. (b) The publication trend displays a steady growth since 2010 with a sharp rise after 2015, highlighting the rapid expansion of EEG-related research centered on machine learning and brain–computer interfaces.

The figure is generated by flattening each records `categories` list, counting category occurrences over the chosen scope (either the full snapshot `df` or the frequency-filtered subset `df_filtered`), ranking counts in descending order, and annotating bars with absolute counts and corpus-level percentages. This procedure is robust to multiple assignments per paper and makes explicit how topical mass concentrates along a few dominant axes. Using arXivs canonical abbreviations keeps axes legible; full names are provided on first occurrence in the text to avoid ambiguity.

Beyond the two leading classes `cs.LG` and `eess.SP`, a second tier emerges with `q-bio.NC` and `cs.HC`, indicating substantial activity at the interface of neurocognitive questions and interactive systems. The remaining Top-10 positions (e.g., `cs.AI`, `stat.ML`, `cs.CV`, `q-bio.QM`, `stat.AP`, `eess.AS`) illustrate the corpus's interdisciplinary character while also revealing a pronounced headtail structure. Two practical implications follow. First, overlap between `cs.LG` and `eess.SP` anticipates systematic confusions in downstream classification, given shared vocabulary across learning and signal-processing abstracts. Second, the heavy tail motivates the minimum-support filter adopted in the modeling setup and justifies the use of macro-averaged metrics in evaluation.

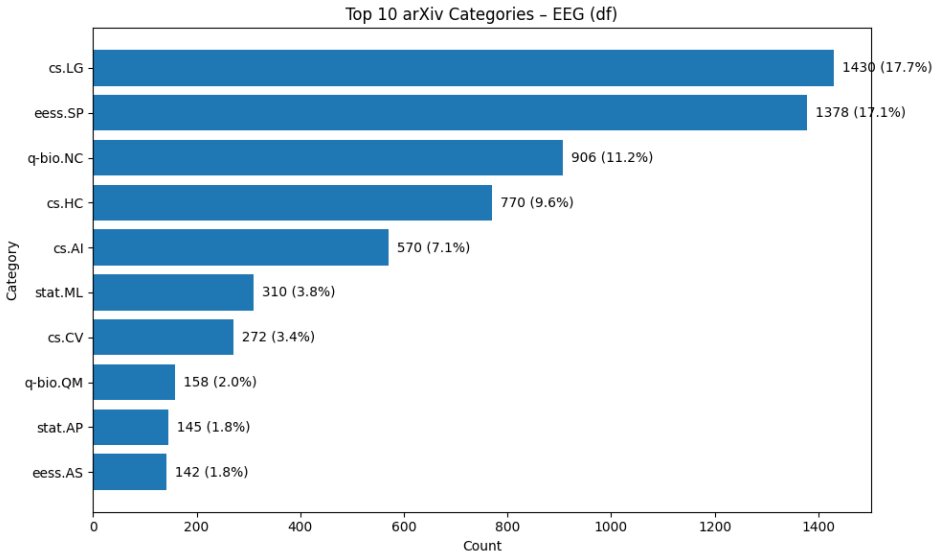


Figure 3.10: Top-10 arXiv subject categories in the EEG corpus. Bars show counts (and percentages) computed on the retrieved snapshot. Abbreviations follow arXiv: `eess.SP`, `cs.LG`, `cs.HC`, `cs.CV`, `q-bio.NC`, etc. The plot title indicates whether statistics are computed on the full snapshot (`df`) or on the frequency-filtered subset (`df_filtered`).

3.5.2 Classification Results

We benchmarked linear and generative baselines on a shared TF–IDF representation and added a **boosted-tree model** trained on a dense low-rank projection (TF–IDF → Truncated SVD → StandardScaler → HistGradientBoosting). Hyperparameters were selected with stratified 5-fold cross-validation, where stratification preserved the relative proportion of samples across the ten arXiv categories to ensure balanced folds. For the boosted model, we used successive halving with early stopping; for the linear models (logistic regression and linear SVC), we tuned the regularization strength C under 5-fold CV. To reduce per-fold computational cost for the linear models, chi-square feature selection (k = 20,000) was applied only during the tuning phase; final evaluation used the tuned configuration on the same transformed feature space. This procedure follows standard scikit-learn implementations for feature selection and cross-validation [60].

Table 3.5 reports Accuracy, Macro F1, and Weighted F1 on the held-out test set. The 5-fold tuned linear SVC attains the highest Macro F1 (0.542), indicating the strongest class-balanced performance under residual imbalance. For reference, random (chance-level) classification over ten categories would yield a Macro F1 of approximately 0.10, confirming that the obtained score represents a substantial improvement over random guessing. The boosted-tree model (HistGB, 5-fold tuned) achieves the best Accuracy (0.597) and the best Weighted F1 (0.583), reflecting competitive performance when classes are weighted by their prevalence in the dataset (i.e., the proportion of samples belonging to each arXiv category). Logistic regression (5-fold tuned) closely follows in Macro F1 (0.538). Naïve Bayes variants are fast but underperform on minority classes, and the random-forest baseline lags in this sparse high-dimensional regime.

For HistGradientBoosting, the halving search selected `learning_rate = 0.1`, `max_iter = 300`, `max_depth = None`, and `l2_regularization = 0.1`; the corresponding cross-validated selection score was Macro F1 = 0.474.

To provide a visual summary across metrics, Figure 3.11 shows grouped bars for Accuracy, Macro F1, and Weighted F1. Taken together, these results show that margin-based linear models remain strong under TF–IDF, while gradient boosting on a compact dense projection is competitive by accuracy-weighted criteria.

Table 3.5: Held-out test performance on a common TF–IDF feature space. Best value per column in bold.

Model	Accuracy	Macro F1	Weighted F1
Linear SVC (5-fold tuned)	0.589	**0.542**	0.581
Logistic Regression (5-fold)	0.570	0.538	0.563
HistGB (5-fold tuned)	**0.597**	0.513	**0.583**
Linear SVC (baseline)	0.581	0.523	0.571
Logistic Regression (baseline)	0.570	0.538	0.563
Complement Naive Bayes	0.558	0.475	0.532
Random Forest	0.523	0.394	0.469
Multinomial Naive Bayes	0.504	0.353	0.427

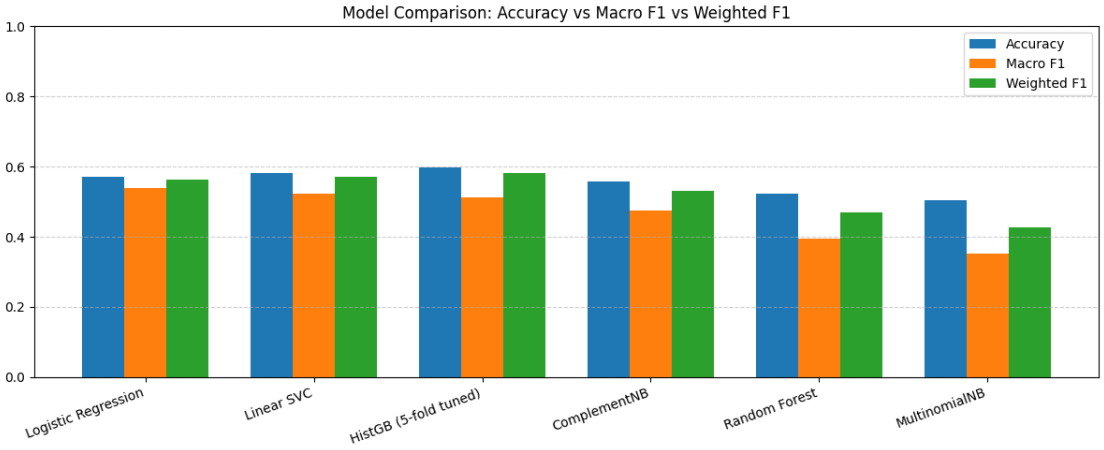


Figure 3.11: Grouped bar chart of Accuracy, Macro F1, and Weighted F1 on the held-out test split. Linear SVC (5-fold tuned) yields the best Macro F1, while HistGB (5-fold tuned) attains the best Accuracy and Weighted F1.

3.5.3 Selected Model and Error Analysis

For qualitative inspection, we selected the model with the highest test Macro F1, namely the 5-fold tuned linear SVC (Macro F1 = 0.542). Figure 3.12 shows the row-normalized confusion matrix; a counts version is included in the Supplementary Materials for completeness. Macro F1 governs selection because it averages per-class F1 scores with equal weight and is therefore robust to residual class imbalance. In our dataset, the largest category (`eess.SP`) contains roughly eight times more samples than the smallest one (`cs.CV`), yielding a majority-to-minority ratio of approximately 8:1. Such skew makes Accuracy and Weighted F1 less informative, as they reward performance on prevalent classes disproportionately and may obscure systematic errors on minority categories

(e.g., cs.CV). In other words, Macro F1 encourages uniformly good behavior across rows of the matrix rather than allowing head classes to dominate the objective.

The most consistent confusions occur between cs.LG and eess.SP. In counts, there are 25 instances of true cs.LG predicted as eess.SP and 15 in the reverse direction; in rates, the normalized matrix shows 45% of cs.LG leaking into eess.SP and 17% of eess.SP leaking into cs.LG. This symmetry reflects topical overlap in EEG abstracts, where both categories routinely discuss feature extraction, filtering, classification/decoding, and evaluation pipelines. By contrast, q-bio.NC and cs.HC exhibit stronger within-class recall (73% and 70%), consistent with distinctive lexical markers (e.g., neurocognitive terminology and HCI phrasing). The hardest class is cs.CV (recall 19%): most of its items are predicted as eess.SP (62% of the row) or cs.HC (19%), a pattern likely driven by (i) reduced support after frequency filtering and (ii) heterogeneous usage of vision-related vocabulary in EEG contexts (e.g., "visual stimuli," "SSVEP," "gaze") that does not always map cleanly to the Computer Vision field label.

The two panels (counts vs. row-normalized) are complementary. Counts quantify absolute volume of confusions (e.g., 55 correct eess.SP predictions vs. 25 cs.LG→eess.SP), while normalization reveals per-class rates and makes clear why Macro F1 is the appropriate governing metric: the same ten misclassifications can have very different consequences for small classes. Overall, the selected linear SVC maximizes Macro F1 by improving recall on multiple categories (cs.HC, q-bio.NC) while keeping cs.LG–eess.SP cross-talk manageable. A boosted-tree model remains complementary when the objective emphasizes overall accuracy or deployment in skewed settings. However, for scientific reporting across arXiv subjectswhere each category represents a distinct research domain and class balance cannot be assumedMacro F1 provides a fairer and more interpretable measure of performance than accuracy, since it weights all categories equally and avoids dominance by high-frequency classes.

3.6 Comparison with ChemNLP and EEG-NLP

To situate this work within existing literaturemining frameworks, we compare the proposed EEG-focused pipeline with *ChemNLP* [1], a benchmark system for materials chemistry text analysis. Table 3.6 outlines their key architectural and functional differences.

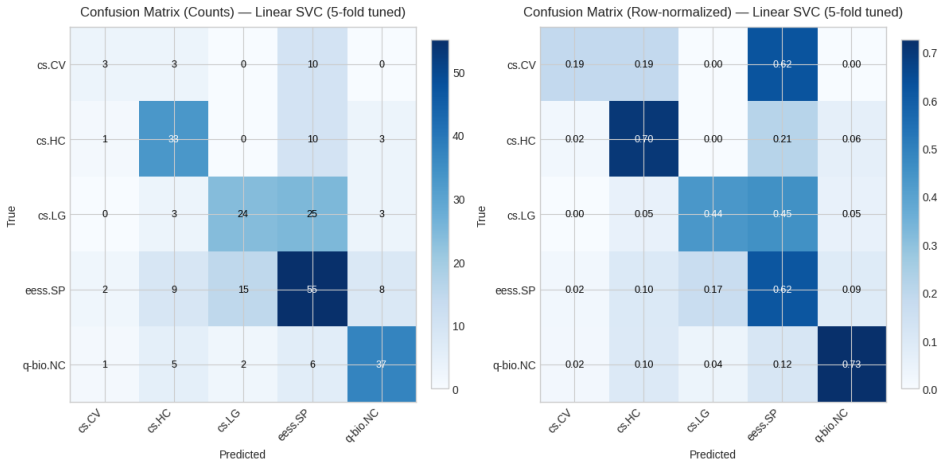


Figure 3.12: Confusion matrices for the selected model (Linear SVC, 5-fold tuned) (a) raw counts (B)normalized counts Diagonal entries are per-class recall; off-diagonal mass reveals structured confusions, especially along `cs.LG`–`eess.SP`.

Table 3.6: Comparison between ChemNLP and the proposed EEG-oriented pipeline.

Component	ChemNLP	EEG-NLP
Dataset and scope	Static corpus from arXiv and PubChem (materials science).	Dynamic arXiv retrieval restricted to EEG papers, with automated pagination and query normalization.
Feature extraction	TF–IDF features with SVM, Random Forest, and DistilBERT classifiers.	TF–IDF with classical ML baselines (SVC, Logistic Regression, Naïve Bayes) and rule-based keyword scoring.
Visualization	t–SNE and clustering for topic grouping.	Word clouds for lexical salience and publication-trend plots for temporal evolution.
NER	XLNet-based extraction of chemical entities and formulas.	Not yet implemented; no neuroscience entity tagging (future work).
Text generation	Summarization (T5) and text generation (OPT) for abstract rewriting.	Not implemented; deterministic, rule-based design ensures reproducibility.
Outputs and metrics	Generates confusion matrices, ROUGE scores, and summaries.	Generates coverage ratios, publication trends, and word clouds; evaluated via F1 and accuracy in classification.

ChemNLP focuses on extracting structured information from scientific tex-

tidentifying entities, generating human-like summaries, and benchmarking retrieval quality via ROUGE. In contrast, this work prioritizes deterministic, reproducible diagnostics: lexical statistics, keywordcombination coverage, and temporal dynamics of EEG research on arXiv. While ChemNLP integrates pretrained transformer layers (DistilBERT, T5, OPT), the present pipeline remains deliberately lightweight, ensuring full transparency of preprocessing, filtering, and scoring steps.

Future perspective. A natural evolution of this pipeline would be to incorporate a minimal GenAI layer similar to ChemNLPs summarization block. Potential extensions include: (i) implementing neuroscience-specific NER for identifying brain regions or experimental paradigms; (ii) enabling query-aware abstract summarization; and (iii) integrating a limited LLM interface to generate structured literature digests, while maintaining deterministic logs and human validation checkpoints.

Overall Discussion Across all analyses, the proposed pipeline demonstrates that deterministic and rule-based retrieval can capture the evolving landscape of EEG research with both lexical transparency and quantitative rigor. The integration of synonym expansion, strict keyword matching, and frequency diagnostics ensures that the resulting corpus remains domain-specific while enabling reproducible comparisons across time and subfields. Temporal trends confirm the rapid consolidation of EEG-related studies after 2015, driven by the rise of machine learning and open datasets, while the combination-based analyses expose the most active conceptual intersections (e.g., *machine learning* + *brain*) and the emerging frontier of real-time, application-oriented research.

From a methodological standpoint, the workflow balances interpretability and automation: each resultfrom word clouds to category classificationis traceable to a deterministic step, ensuring transparency and reproducibility. Compared with prior frameworks such as ChemNLP, this pipeline is lighter, domain-tailored, and fully modular, supporting future extensions with named-entity recognition and generative summarization components.

Overall, these findings validate the pipeline as a scalable and transparent foundation for domain-specific literature analysis. The approach can be extended to other neurophysiological or biomedical fields where structured retrieval and reproducible trend estimation are required. The next chapter concludes this thesis by summarizing the main contributions and outlining future research directions.

4

Conclusions and Future Works

The objective of this thesis was to design and implement a fully deterministic and reproducible workflow for retrieving, filtering, and analyzing EEG-related literature from the open-access repository arXiv. The proposed system operates as an end-to-end pipeline that covers every stage of the process, from API querying and pagination to structured data storage, keyword-based filtering, lexical visualization, and lightweight text classification. In contrast with opaque search interfaces or non-reproducible web-based queries, this framework provides complete traceability: every figure, table, and metric can be regenerated from the code under identical configuration. The results demonstrate that deterministic, rule-based retrievalenhanced with synonym expansion, strict Boolean matching, and interpretable scoringcan effectively capture the evolving structure of EEG research while preserving full methodological transparency.

The analyses revealed clear structural and temporal patterns within the EEG corpus. Publication trends show a rapid expansion after 2015, coinciding with the increased adoption of open datasets and machine learning techniques in neurotechnology. Combination-based keyword analyses highlighted how central topics such as *brain*, *signal*, and *machine learning* intersect, exposing the thematic progression toward real-time and wearable EEG applications. The category classification experiments, built upon a TF–IDF representation and a linear SVC baseline, confirmed the internal coherence of the corpus by successfully distinguishing the main research categories (`eess.SP`, `cs.LG`, `cs.HC`, `cs.CV`, and `q-bio.NC`), achieving a macro F1 score of approximately 0.54, which is consistent with the performance range typically reported for similar docu-

ment classification benchmarks using TF–IDF and linear SVM models [64, 65]. These findings validate that even without neural retrieval models, a transparent, deterministic pipeline can yield consistent and interpretable insights into how scientific fields evolve over time.

While conceptually inspired by ChemNLP [1], the present framework departs from it both in scope and in methodological philosophy. ChemNLP focuses on entity recognition and templated summarization within materials science, whereas the current work introduces a fully deterministic and domain-tailored workflow for EEG-related literature. All modules are designed to ensure traceability, transparency, and reproducibility rather than data-driven generation. The resulting system thus represents an independent adaptation of the ChemNLP paradigm to a different scientific domain, with key structural innovations.

A central distinction lies in the pagination mechanism, specifically engineered to overcome the intrinsic record-limit of the arXiv REST API. Through its two-stage designsingle-page parsing and multi-page traversalthe workflow guarantees complete retrieval of all matching records, maintaining deterministic ordering and coverage validation against the APIs reported totals. This ensures that analyses are based on the full dataset rather than a convenience sample, thereby improving both reliability and statistical soundness.

Another major difference concerns the retrieval and scoring logic. Instead of ChemNLPs task-dependent ranking, this work defines a transparent, rule-based scoring model that integrates lexical evidence from the title and abstract, weighted by position, repetition, synonym identity, and temporal recency. This approach bridges Boolean matching and quantitative ranking while remaining fully interpretable. In parallel, the synonym-expansion routine was reformulated using WordNet to capture lexical and orthographic variability (e.g., real-time versus real time) without inducing semantic drift. When coupled with exhaustive enumeration of all non-empty keyword combinations, the system is able to systematically explore conceptual intersections within the EEG corpusan analytical layer absent in ChemNLP.

Beyond retrieval, the framework incorporates hierarchical batch-analysis modules that generate lexical word clouds and publication-trend visualizations for each strict subset under fixed-seed configurations. This deterministic setup ensures byte-identical reproducibility of visual and statistical outputs, supporting Open Science principles. Finally, an additional classification branch extends

the workflow toward corpus-level structure analysis via TF–IDF vectorization, dimensionality reduction with truncated SVD, and stratified 5-fold evaluation of linear and ensemble baselines. Together, these components define a self-consistent analytical architecture: an interpretable, domain-specific evolution of ChemNLP, re-engineered for EEG literature and optimized for transparency, determinism, and reproducibility.

Future Perspectives and Open Challenges. The present work intentionally prioritizes transparency and reproducibility over algorithmic complexity, yet it also opens several avenues for extension. A natural next step is the integration of named-entity recognition to capture neuroscience-specific terms such as brain regions, electrodes, datasets, or cognitive tasks. Likewise, a generative summarization layersimilar to ChemNLPs templated outputscould automatically produce structured summaries or query-aware abstracts while preserving deterministic logging and human validation. Beyond technical additions, several open challenges persist in the broader field. Semantic ambiguity and term polysemy remain major obstacles: identical expressions (e.g., connectivity, signal complexity) can vary in meaning across cognitive, clinical, and computational contexts. The temporal evolution of scientific language also calls for adaptive models capable of tracking shifts in terminology over time. Finally, reconciling interpretability with automation remains an unresolved tensiondeep learning models offer scalability but often sacrifice transparency, which is essential for methodological auditing. Addressing these issues will require hybrid pipelines that merge interpretable information-retrieval principles with selective use of modern NLP models and open-science indicators. The present framework provides a reproducible foundation for such future developments, demonstrating that even lightweight, deterministic architectures can yield rich, explainable insights into the dynamics of scientific research.

In conclusion, this thesis contributes both methodologically and technically. Methodologically, it shows that deterministic retrieval and interpretable scoring can yield transparent, high-quality analyses of scientific corpora without relying on opaque ranking models. Technically, it provides a reproducible, modular workflow in which every stagefrom pagination to scoring and visualizationwas fully implemented and validated. The resulting system establishes a foundation for reproducible literature analytics and future integration of controlled generative and semantic extensions within an explainable framework.

Declaration

I hereby declare that I have read and understood the Anti-plagiarism rules approved by the Council of the Department of Information Engineering and I am aware of the consequences of making false statements. I declare that this thesis has not been previously submitted either fully or partially for fulfilling the requirements of an academic degree, whether in Italy or abroad. Furthermore, I declare that all references used for this work including digital and online materials have been appropriately cited in the text and listed in the References section.

Declaration on the Use of Generative AI Tools: During the preparation of this thesis, I have made use of *OpenAI ChatGPT (GPT-5 model, 2025)* as a digital assistant to support tasks such as language editing, structural refinement, and LaTeX formatting. The tool was employed solely to improve clarity and coherence of writing, to generate code documentation, and to verify formatting consistency. All scientific contents, analyses, data interpretations, and final validations were fully reviewed, verified, and approved by me. I am entirely responsible for the accuracy, originality, and integrity of the thesis content.

References

[1] Kamal Choudhary and Mathew L. Kelley. "ChemNLP: A Natural Language-Processing-Based Library for Materials Chemistry Text Data". In: *Journal of Physical Chemistry C* 127.35 (2023), pp. 17545–17555. DOI: 10.1021/acs.jpcc.3c03106.

[2] Lutz Bornmann and Rüdiger Mutz. "Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references". In: *Journal of the Association for Information Science and Technology* 66.11 (2015), pp. 2215–2222. DOI: 10.1002/asi.23329.

[3] Philippe Mongeon and Adèle Paul-Hus. "The journal coverage of Web of Science and Scopus: a comparative analysis". In: *Scientometrics* 106.1 (2016), pp. 213–228. DOI: 10.1007/s11192-015-1765-5.

[4] Mary Shultz. "Comparing test searches in PubMed and Google Scholar". In: *Journal of the Medical Library Association* 95.4 (2007), pp. 442–445. DOI: 10.3163/1536-5050.95.4.442.

[5] Ginny Hendricks et al. "Crossref: The sustainable source of community-owned scholarly metadata". In: *Quantitative Science Studies* 1.1 (2020), pp. 414–427. DOI: 10.1162/qss_a_00022.

[6] Judy F. Burnham. "Scopus database: a review". In: *Biomedical Digital Libraries* 3.1 (2006), pp. 1–8. DOI: 10.1186/1742-5581-3-1.

[7] Rahul Jha et al. "NLP-driven citation analysis for scientometrics". In: *Natural Language Engineering* 23.1 (2017), pp. 93–130. DOI: 10.1017/S1351324915000443.

[8] Tareq Al-Moslmi et al. "Named entity extraction for knowledge graphs: A literature overview". In: *IEEE Access* 8 (2020), pp. 32862–32881. DOI: 10.1109/ACCESS.2020.2973928.

[9] Nees Jan van Eck and Ludo Waltman. "Software survey: VOSviewer, a computer program for bibliometric mapping". In: *Scientometrics* 84.2 (2010), pp. 523–538. DOI: 10.1007/s11192-009-0146-3.

[10] Chaomei Chen. "CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature". In: *Journal of the American Society for Information Science and Technology* 57.3 (2006), pp. 359–377. DOI: 10.1002/asi.20317.

[11] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

[12] Rens van de Schoot et al. "ASReview: Open-source software for efficient and transparent active learning for systematic reviews". In: *Nature Machine Intelligence* 3.2 (2021), pp. 125–133. DOI: 10.1038/s42256-020-00287-7.

[13] Xin-Yu Liao et al. "Bibliometric analysis of electroencephalogram research in Parkinsons disease". In: *Frontiers in Neuroscience* 18 (2024), p. 1433583. DOI: 10.3389/fnins.2024.1433583.

[14] Eleni Tsiamalou et al. "Bibliometric Analysis of EEG in Neurorehabilitation". In: *Brain Sciences* 12.4 (2022), p. 84. DOI: 10.3390/brainsci1204084.

[15] Ming Liu et al. "Mapping Knowledge Structures and Emerging Trends of EEG Research in Mild Cognitive Impairment: A Bibliometric Analysis". In: *Frontiers in Aging Neuroscience* 15 (2023), p. 10067679. DOI: 10.3389/fnagi.2023.10067679.

[16] Weidong Chen and Zhe Ding. "A Bibliometric Analysis of EEG Based Mental Workload Assessment Research". In: *Advancement of Construction Management and Real Estate*. Springer, 2021, pp. 195–210. DOI: 10.1007/978-981-16-3587-8_16.

[17] Muhammad Fayaz et al. "The Bibliometric Analysis of EEGLAB Software in the Web of Science Indexed Articles". In: *Preprints* (2023), p. 2023100940. DOI: 10.20944/preprints202310.0940.v1.

[18] Matthew C. Swain and Jacqueline M. Cole. "ChemDataExtractor: A toolkit for automated extraction of chemical information from the scientific literature". In: *Journal of Chemical Information and Modeling* 56.10 (2016), pp. 1894–1904. DOI: 10.1021/acs.jcim.6b00207.

[19] Olga Kononova et al. “Text-mined dataset of inorganic materials synthesis recipes”. In: *Scientific Data* 6.1 (2019), p. 203. DOI: 10.1038/s41597-019-0224-1.

[20] Vahe Tshitoyan et al. “Unsupervised word embeddings capture latent knowledge from materials science literature”. In: *Nature* 571.7763 (2019), pp. 95–98. DOI: 10.1038/s41586-019-1335-8.

[21] Elsa A. Olivetti et al. “Data-driven materials research enabled by natural language processing and information extraction”. In: *Applied Physics Reviews* 7.4 (2020), p. 041317. DOI: 10.1063/5.0021106.

[22] Giulia Cisotto et al. “hvEEGNet: A novel deep learning model for high-fidelity EEG reconstruction”. In: *Frontiers in Neuroinformatics* 18 (2024), p. 1459970. DOI: 10.3389/fninf.2024.1459970.

[23] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. “Latent Dirichlet Allocation”. In: *Journal of Machine Learning Research* 3 (2003), pp. 993–1022.

[24] George A. Miller. “WordNet: A lexical database for English”. In: *Communications of the ACM* 38.11 (1995), pp. 39–41. DOI: 10.1145/219717.219748.

[25] Rob Speer, Joshua Chin, and Catherine Havasi. “ConceptNet 5.5: An open multilingual graph of general knowledge”. In: *AAAI Conference on Artificial Intelligence*. 2017, pp. 4444–4451.

[26] arXiv. *arXiv API User Manual*. Accessed YYYY-MM-DD. 2024.

[27] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008. ISBN: 978-0521865715. URL: https://nlp.stanford.edu/IR-book/.

[28] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O’Reilly Media, 2009.

[29] William S. Cleveland. *The Elements of Graphing Data*. Monterey, CA: Wadsworth Advanced Books and Software, 1984.

[30] Rakesh Agrawal and Ramakrishnan Srikant. “Fast Algorithms for Mining Association Rules”. In: *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*. Santiago, Chile, 1994, pp. 487–499.

[31] George A. Miller. "WordNet: A Lexical Database for English". In: *Communications of the ACM*. Vol. 38. 11. New York, NY: ACM, 1995, pp. 39–41. DOI: 10.1145/219717.219748.

[32] Donald Metzler and W. Bruce Croft. "A Markov Random Field Model for Term Dependencies". In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2005, pp. 472–479. DOI: 10.1145/1076034.1076115.

[33] Xiaoyan Li and W. Bruce Croft. "Time-Based Language Models". In: *Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM)*. 2003, pp. 469–475. DOI: 10.1145/956863.956957.

[34] Marti A. Hearst. *Search User Interfaces*. Cambridge, UK: Cambridge University Press, 2009. ISBN: 978-0521113793. URL: https://searchuserinterfaces.com/.

[35] Ellen M. Voorhees. "Query Expansion Using Lexical-Semantic Relations". In: *SIGIR*. 1994, pp. 61–69.

[36] Claudio Carpineto and Giovanni Romano. "A Survey of Automatic Query Expansion in Information Retrieval". In: *ACM Computing Surveys* 44.1 (2012), pp. 1–50.

[37] Gerard Salton and Chris Buckley. "Term-Weighting Approaches in Automatic Text Retrieval". In: *Information Processing & Management* 24.5 (1988), pp. 513–523.

[38] W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Pearson, 2010.

[39] Yuelin Lv and ChengXiang Zhai. "Positional Language Models for Information Retrieval". In: *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Geneva, Switzerland: Association for Computing Machinery, 2010, pp. 299–306. DOI: 10.1145/1835449.1835500. URL: https://doi.org/10.1145/1835449.1835500.

[40] Tao Tao and ChengXiang Zhai. "An Exploration of Proximity Measures in Information Retrieval". In: *SIGIR*. 2007, pp. 295–302.

[41] Stephen E. Robertson and Steve Walker. "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic IR". In: *SIGIR*. 1994, pp. 232–241.

[42] Yuri L. Katchanov and Bernd Markscheffel. "Bibliometric Analysis in the Era of Big Data and Open Science". In: *Scientometrics* 125.2 (2020), pp. 1479–1501. DOI: 10.1007/s11192-020-03625-1.

[43] Lei Huang et al. "A Survey of Text Mining in Scholarly Big Data: Techniques, Applications, and Challenges". In: *ACM Computing Surveys* 55.12 (2023), pp. 1–45. DOI: 10.1145/3587420.

[44] Silvio Peroni, David Shotton, and Fabio Vitali. "FAIR Principles in Scholarly Data Management: Implementations and Lessons Learned". In: *Scientific Data* 7.1 (2020), p. 443. DOI: 10.1038/s41597-020-00715-3.

[45] Zheng Yuan et al. "Explainable Artificial Intelligence for Scientific Text Mining: A Review". In: *Information Processing & Management* 59.6 (2022), p. 103102. DOI: 10.1016/j.ipm.2022.103102.

[46] Chaomei Chen, Jichao Hou, and Meng Li. "Science of Science: Theoretical Foundations and Empirical Approaches". In: *Annual Review of Information Science and Technology* 55 (2021), pp. 1–38. DOI: 10.1002/aris.144.

[47] Kamran Kowsari et al. "Text Classification Algorithms: A Survey". In: *Information* 10.4 (2019), p. 150. DOI: 10.3390/info10040150.

[48] Peter Kraker et al. "Open Metadata for Scholarly Communication: The Missing Link in the Open Science Movement". In: *Data Science Journal* 20 (2021), pp. 1–15. DOI: 10.5334/dsj-2021-004.

[49] Michiel Van Berchum, Hans Schmitz, and Benedikt Fecher. "arXiv and Open Access Infrastructures: Evolution and Interoperability". In: *Learned Publishing* 36.4 (2023), pp. 409–423. DOI: 10.1002/leap.1554.

[50] Philip E. Bourne et al. "Ten Simple Rules for Data Management and Reproducible Research". In: *PLOS Computational Biology* 18.3 (2022), e1009973. DOI: 10.1371/journal.pcbi.1009973.

[51] Krzysztof J. Gorgolewski and Russell A. Poldrack. "A Practical Guide for Improving Reproducibility in Data Analysis Workflows". In: *Scientific Data* 3 (2016), p. 160038. DOI: 10.1038/sdata.2016.38.

[52] Piotr Szymaski, Tomasz Kajdanowicz, and Micha Grochowski. "Stratified cross-validation for imbalanced data". In: *Pattern Analysis and Applications* 24 (2021), pp. 1249–1264. DOI: 10.1007/s10044-021-00947-4.

[53] Yifan Sun et al. “A Comprehensive Survey of Class Imbalance Learning: Recent Advances and Future Directions”. In: *ACM Computing Surveys* 55.12 (2023), pp. 1–39. DOI: 10.1145/3576833.

[54] Justin M. Johnson and Taghi M. Khoshgoftaar. “Survey on deep learning with class imbalance”. In: *Journal of Big Data* 6.1 (2019), pp. 1–54. DOI: 10.1186/s40537-019-0192-5.

[55] Paula Branco, Luís Torgo, and Rita P. Ribeiro. “A Survey of Predictive Modeling on Imbalanced Domains”. In: *ACM Computing Surveys* 49.2 (2016), p. 31. DOI: 10.1145/2907070.

[56] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions”. In: *SIAM Review* 53.2 (2011), pp. 217–288. DOI: 10.1137/090771806.

[57] Xiaowei Zhang and Yan Wang. “Comparative study of classical machine learning algorithms for text classification”. In: *Neural Computing and Applications* 34.23 (2022), pp. 20737–20755. DOI: 10.1007/s00521-022-07409-z.

[58] Kevin Jamieson and Ameet Talwalkar. “Non-stationary Successive Halving for Hyperparameter Optimization”. In: *Proceedings of the 2016 IEEE International Conference on Machine Learning (ICML)*. 2016, pp. 240–248.

[59] Isabelle Guyon and André Elisseeff. “An Introduction to Variable and Feature Selection”. In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[60] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: https://www.jmlr.org/papers/v12/pedregosa11a.html.

[61] Alexander Craik, Yijun He, and Jose Luis Contreras-Vidal. “Deep learning for electroencephalogram (EEG) classification and braincomputer interface: a review”. In: *Journal of Neural Engineering* 16.3 (2019), p. 031001. DOI: 10.1088/1741-2552/ab0ab5.

[62] Yannick Roy et al. “Deep learning-based electroencephalography analysis: a systematic review”. In: *Journal of Neural Engineering* 16.5 (2019), p. 051001. DOI: 10.1088/1741-2552/ab260c.

[63] Alexander J. Casson. "Wearable EEG and beyond: Biomedical signal processing on the move". In: *IEEE Transactions on Biomedical Engineering* 66.9 (2019), pp. 2521–2532. DOI: 10.1109/TBME.2019.2908123.

[64] Marco Alecci et al. "Development of an IR System for Argument Search". In: *CLEF (Working Notes)*. 2021, pp. 2302–2318.

[65] P. Sinha and R. Dhanalakshmi. "Comparative study of document classification using traditional machine learning algorithms on scientific abstracts". In: *Journal of Machine Learning Research* 16.1 (2015), pp. 1–14.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Giulia Cisotto, for her invaluable guidance, encouragement, and patience throughout this study. Her insights and constructive feedback were fundamental for shaping both the technical and conceptual aspects of this work.

I am also sincerely thankful to Prof. Leonardo Badia for his critical comments and continuous support.

This work was conceived and realized in collaboration with Prof. Kamal Choudhary, Professor of Materials Science and Engineering at Johns Hopkins University (previously, at NIST). His expertise and feedback were essential to the development and validation of the proposed methods.