INTERNATIONAL
HELLENIC
UNIVERSITY

Using Data Mining to Analyze Temporal Trends in the Reporting of Method-Related Keywords

Konstantinos Stathakis

SID: 3301220002

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

JANUARY 2025

THESSALONIKI – GREECE

INTERNATIONAL HELLENIC UNIVERSITY

Using Data Mining to Analyze Temporal Trends in the Reporting of Method-Related Keywords

Konstantinos Stathakis

SID: 3301220002

Supervisor:	Prof. Christos Tjortjis
Supervising Committee Members:	Dr P. Koukaras Dr C. Berberidis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

JANUARY 2025
THESSALONIKI – GREECE

Abstract

Scientific documents, such as research articles, are rich sources of data for Information Retrieval (IR) and text mining tasks, allowing the extraction and analysis of specialized knowledge from extensive bodies of scholarly work. Along these lines, abstracts from scientific articles can be a valuable source of information for understanding the scope of a study.

This study attempted to identify inherent groupings of psychology abstracts and to analyze temporal trends in the reporting of method-related keywords examining a corpus of 85,452 articles. For this purpose, a glossary of 365 method-related terms was utilized, serving as a gold standard collection. To meet the first objective, the retrieved keywords were vectorized using SciBERT before being fed to a clustering model. Both an unsupervised and a semi-supervised model were utilised. The best clustering results produced six clusters for the unsupervised model and ten for the semi-supervised model. For the second objective, keyword frequencies and TF-IDF scores were examined independently of the formed clusters.

Neither clustering approach yielded clear grouping formations. Terms were found to overlap between clusters, even with the use of higher weights for those terms with high theoretical discriminative value. Moreover, the number of abstracts with no method-related terms appears to be facing a downward trend.

Overall, this study provides insights regarding temporal trends in the reporting of method-related keywords in psychology research abstracts. The increasing presence of these terms over the years reflects a shift towards greater methodological transparency and standardization in abstract reporting. To enhance the identification of method grouping trends, the use of a predefined glossary as a standalone gold standard for term retrieval should ensure the inclusion of terms with high discriminative value, facilitating clearer differentiation of abstracts based on method-related terms.

Acknowledgements

Sincere thanks to my supervisor, Professor Christos Tjortjis, and to Ph.D. candidate Georgios Papageorgiou for the time they devoted and their constructive feedback.

Konstantinos Stathakis

17/01/2025

Contents

List of Figures

List of Tables

1 Introduction

Scientific documents, such as research articles, are rich sources of data for IR and text mining tasks, allowing the extraction and analysis of specialized knowledge from extensive bodies of scholarly work [1]. In this framework, the extraction process can be conducted so much at the document level as at the section level, depending on the aim of the task.

Extraction that occurs at the level of the entire document is pertinent to tasks whose objective might be overall summarization, topic modeling, or full-text classification [2]. On the other hand, by focusing on specific sections within a document, the distinct content typically found in each of those sections of a scientific paper can be leveraged. Thereupon, information about the objectives of a study, the methods used, or the key findings could be extracted using text mining techniques for further analysis [2].

Along these lines, abstracts from scientific articles can be a valuable source of information for understanding the scope of a study. Although abstract formats may vary based on field and journal guidelines, they currently tend to follow a semi-structured format, that nonetheless includes key elements, such as the study background, objectives, methods, results, and conclusions.

According to the latest American Psychological Association (APA) publication manual [3], empirical article abstracts should adhere to specific content guidelines. Among other elements, essential features regarding the study method are requested to be included. These guidelines emphasize the inclusion of essential methodological details, such as data-gathering procedures, sample characteristics, research design, and materials used.

The APA's guidelines reflect methodological standards promoted by prestigious organizations in psychology. While adherence to these guidelines does not inherently ensure methodological rigor or replicability, it supports transparency and encourages robust reporting practices. Importantly, such an emphasis on methodological clarity in abstracts represents a shift towards greater transparency in psychological research, which was not always a standard practice in the field.

In 2011, the publication of Bem's paper [4], which claimed to provide experimental evidence for precognition[1], sparked considerable controversy within the scientific community.

[1] *Precognition* refers to the phenomenon where individuals can perceive or be aware of future events before they

His research was later found to have serious methodological flaws, whereas his findings, which ultimately could not be replicated, raised critical questions about the methodologies and standards employed generally in psychological research. Notably, six years prior to Bem's publication, Ioannidis had already voiced his concerns [5], highlighting that methodological issues in psychology were not a sudden emergence, but rather longstanding problems. This marked the inception of the *replication crisis* in the field.

In 2015, the Reproducibility Project culminated in a landmark study published in *Science*, which succeeded in replicating only 36% of the 100 studies analyzed from leading psychology journals [6]. Other large-scale replication initiatives, such as Many Labs 1 and 2 [7], further corroborated these findings, indicating that a significant number of psychological studies, particularly within social psychology, exhibited low replicability.

The concepts of *publication bias*, p-hacking, and inadequate methodological reporting[2] became central to the discourse surrounding the replication crisis and were thought of as factors intertwined with it [8]. In response to these challenges, psychology journals and research institutions have initiated reforms aimed at improving research integrity. For instance, the abovementioned APA reporting guidelines exemplify efforts to enhance methodological transparency and rigor in psychological research.

Fourteen years after his initial publication raising concerns, Ioannidis reiterated the importance for vigilance against selection biases that skew reported p-values and contribute to the broader misuse of statistical inference [9]. Enhancing the quality of psychological research depends on robust methodological practices and, crucially, a commitment to methodological transparency. This transparency can be fostered through systematic adoption of practices, such as registered reports, data sharing, and pre-registration, all of which ensure comprehensive method reporting and thereby support robust replication testing [8].

Considering the low reproducibility rate in psychological research, the current study applies Natural Language Processing (NLP), text mining, and Machine Learning (ML) methods to examine how methodological reporting in psychology article abstracts has evolved over the last 30 years. More specifically, it employs a predefined glossary to retrieve method-related keywords from the examined abstracts. The retrieved keywords are then vectorized using

happen.

[2] *Publication bias* refers to the journals' tendency to publish statistically significant results, leading to a distorted understanding of research findings (e.g. overestimation of the effect size of a treatment or intervention). *P-hacking* is defined as the practice of selectively reporting or manipulating data to achieve statistical significance, often by re-analyzing data multiple times until a statistically significant result is obtained. *Poor methodological reporting* is inadequate or unclear documentation of research methods, including sample size, data collection, and analysis procedures, which hinders reproducibility and compromises transparency.

SciBERT, and the generated vectors feed both an unsupervised and a semi-supervised clustering model. In parallel to this process, and irrespective of the clustering method, frequency and trend analyses are performed on the retrieved keywords. Hence, the objectives of the study can be summarized as follows:

- To identify inherent method groupings of psychology abstracts, based on the presence of method-related keywords.
- To analyze the prevalence of the retrieved keywords and their temporal trends.

The results of trend analysis indicate that there is an increasing presence of method-related keywords in the examined abstracts over the years. With respect to the cluster analyses, generic keywords dominate the cluster formation even with the semi-supervised approach in which higher weights are assigned to keywords with theoretically high discriminative value.

Overall, through the effective investigation of these topics, research and method trends can be illuminated, contributing to the transparency and rigor in research communication. Moreover, abstracts with clear method-related information provide a better snapshot of the research design and approach, contributing this way to the replicabililty of studies.

2 Theoretical Background

Text mining is a critical process in NLP that encompasses several steps, ranging from retrieving relevant information to analyzing text data to uncover latent knowledge and structures. This pipeline integrates a variety of components, including core NLP techniques, preprocessing methods, and ML applications. The following section delves into the theoretical foundations of these components and highlights some of the most prevalent techniques used in text mining.

2.1 NLP Analytical Axes

There are five main axes of analysis, namely morphological and lexical, syntactic, semantic, pragmatic and discourse integration, each of which focuses on a different aspect of language [10]. Figure 1 presents a schematic of the said axes.

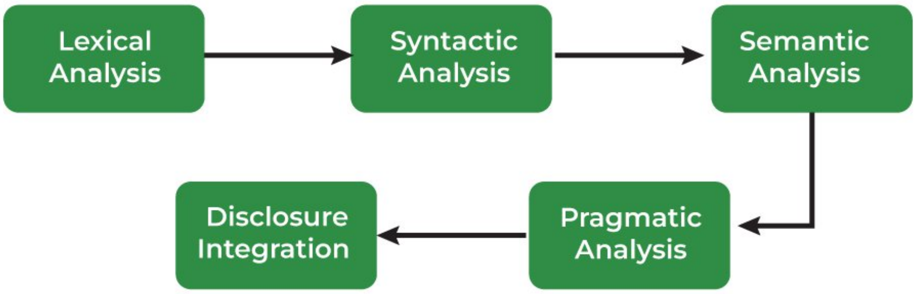


Figure 1: Axes of NLP analysis

To accommodate and facilitate these different layers of analysis, a variety of statistical, ML, and deep learning techniques are employed, enabling more nuanced and effective interpretations of language across each axis [11].

The morphological and lexical analysis axis is centered on breaking text into smaller units, a process known as tokenization. Tokenization produces individual characters, words, or even phrases based on the specific requirements of the NLP task. For these units to be interpretable by a computer, several preprocessing steps are applied. The text is "decluttered" by removing irrelevant elements, such as special characters, HTML tags, and stopwords (e.g., "a," "the," "and"), resulting in a more normalized form. Sentences are separated by punctuation or other defined linguistic cues, and individual tokens are identified, often using whitespace or punctuation as delimiters. Tokens may be further standardized by lowercasing, stemming, or lemmatization [10].

Stemming and lemmatization are NLP preprocessing techniques used to reduce words to their base or root forms, providing several benefits. First, they reduce vocabulary size, thereby simplifying the complexity and computational load on subsequent NLP models. Search and retrieval accuracy is also improved, as variations of a word are grouped together and considered in searches. Moreover, by converting words with different tenses or derivations to a common form, these techniques maintain consistency, allowing models to recognize related words as the same concept. However, it needs to be highlighted that stemming and lemmatization achieve this reduction differently.

Generally, stemming removes prefixes or suffixes from words, which can sometimes produce truncated forms that are not valid words (e.g., "happiness" might be reduced to "happi"). Over the years, various stemming methods have been developed to address issues of under-stemming, over-stemming, or mis-stemming, leading to categories such as linguistic knowledge-based, statistical, corpus-based, context-sensitive, and hybrid stemmers [12]. Linguistic knowledge-based techniques, for example, report higher accuracy for highly inflected languages like Arabic and Hindi [12], underscoring the language-specific requirements in NLP and the limitations of a one-size-fits-all approach. In contrast, lemmatization is a more context-aware, linguistically-driven approach that reduces words to their canonical dictionary form, known as lemma.

Unlike stemming, which mechanically truncates words, lemmatization leverages vocabulary, morphological analysis, and part-of-speech (POS) information to ensure that the output is a valid word. For instance, lemmatization converts "better" to "good" and transforms variations like "am," "are," and "is" to "be". In general, lemmatization is preferred when accuracy and preservation of meaning are critical, such as in tasks involving sentence similarity. In comparison, stemming is useful for simpler or more performance-sensitive applications where speed optimization is prioritized [13].

The second axis of analysis pertains to the syntax of the sentence. In this phase, the tokens derived from the previous analysis are assigned grammatical tags, such as noun, verb, adjective etc., through a process known as POS tagging. At this point it is important to clarify that this assignment is often an integral component of lemmatization; thus, the chronological order of these steps is not distinct and may vary depending on the employed techniques.

In general, POS tagging aids in understanding the grammatical structure of the sentence, providing crucial information about how words relate to one another and serving as a precursor for parsing. Over the years, the methods used for tagging have evolved from rule-based and statistical taggers to more advanced ML and deep learning models [14]. The latter, often referred to as stochastic taggers, can be broadly classified into two categories: supervised and unsupervised, based on whether their training involves a dependent variable (i.e., they are trained with pairs of input and defined output).

Common models in this framework include Hidden Markov Models (HMMs), Conditional Random Fields (CRFs), and Neural Networks (NN). Additionally, there is a third kind, the hybrid ones which aim to combine the strengths of both rule-based and ML approaches. These hybrid systems employ rules to capture specific, known patterns while also incorporating ML elements to generalize to unknown or ambiguous cases. As a general note it is worth noting that the performance of taggers is highly dependent on the application domain and the specific task at hand. However, stochastic and hybrid taggers consistently outperform rule-based methods across nearly all comparison criteria [14].

POS tagging is intertwined with parsing, as the former elicits the information required for the latter. As mentioned above, taggers clarify the role of each word, setting the stage for parsing to analyze the syntactic structure of a sentence and reveal the relationships between words and phrases. As with tagging, parsing approaches vary in purpose and methodology, with some for example focusing on understanding syntax while others on resolving ambiguities within sentences. Constituency and dependency parsing are two primary approaches for elucidating the syntactic constituents of a sentence [11]. These approaches can be categorized as either top-down or bottom-up.

Top-down approaches, such as constituency parsing, start with the overall structure of the sentence and break it down into smaller units, providing a broad overview of the sentence's organization. In contrast, bottom-up approaches, such as dependency parsing, begin with individual words and build up the sentence structure from there, allowing for a more detailed analysis of word relationships. That is achieved through the construction of parse trees that illustrate the hierarchical and syntactic relationships between words. The constituency tree, on one hand, depicts a sentence’s hierarchical structure by delineating the bigger grammatical units into which words are arranged. On the other hand, the dependency tree, as its name suggests, depicts the grammatical links between the words, highlighting their interdependencies.

Semantic parsing goes beyond syntax to interpret meaning, directly addressing ambiguities, by selecting the most contextually appropriate interpretations [10]. Building on syntactic parsing, in this third axis of analysis a sentence is mapped to its meaning using a structured, machine-readable format. This structured format enables advanced NLP processes, such as determining the correct sense of a word based on context, resolving coreferences, and labeling words according to their roles in an action or event. Key components in this process include identifying entities and their respective relationships as well as recognizing basic intent, which is particularly important for commands and requests.

Three main types of those structured representations that are commonly used to express sentence meaning are logical forms (such as lambda calculus), knowledge graphs (where entities are represented as nodes and their relationships as edges), and SQL queries (where natural language is translated into a database query to enable specific information extraction). These structured, rich, representations form the basis of semantic-level analysis and along with the aforementioned processes applied on them set the foundation for the axis of pragmatics.

Analysis on the pragmatic level extends on the direct, dictionary meanings of words and sentence structures that the semantic level provides [10]. Speaker's intent and tone as well as the targeted audience are elements which are taken into account. Surrounding context is examined to identify presuppositions within a sentence or phrase, allowing for the derivation of implicatures and enabling the inference of meanings that are implied rather than directly stated. At this level, language interpretation aligns with real-world usage, where meaning can be inferred beyond literal content.

An example of language's contextual understanding could be elucidated through the pragmatic analysis of the sentence "It is cold outside". Considering the speaker's intention as well as the respective context, this sentence could be interpreted as an implied request to close the window, instead of a mere statement about the temperature. ML algorithms, such as Support Vector Machines (SVMs), random forests and NN are frequently employed at this stage to classify text based on its sentiment, intent and tone. By doing so, NLP models can generate more nuanced and contextually relevant responses, enhancing their effectiveness in real-world applications such as automated customer service.

Finally, discourse integration represents the highest and most complex level of NLP analysis. At this level, the focus is on understanding relationships between sentences, paragraphs, and larger units of text, aiming to derive the text's overall meaning [11]. Discourse-level analysis builds on meanings from the pragmatic level to enable holistic interpretation, essential for a range of NLP tasks such as text segmentation, classification, summarization, and coherence assessment. In each of these tasks, text units are analyzed and skillfully combined, acknowledging their interdependencies and the way they contribute to the overarching narrative or argument. For instance, in text summarization, the discourse level helps identify the most important information and organize it in a coherent and meaningful way. In general, this level of analysis is particularly challenging due to the nuanced interplay between units, as discourse often involves maintaining themes, tracking references, and managing shifts in tone or topic across multiple sentences or paragraphs.

In conclusion, the levels of analysis in NLP are increasingly complex, from the lexical level to the discourse level, with each level building upon the previous one to provide a deeper understanding of the text. While this theoretical framework suggests a gradual increase in complexity, the actual complexity of each level can vary depending on the specific task and the nuances of the language being analyzed.

2.2 NLP Techniques: From Preprocessing to Machine Learning Applications

At each analytical level presented in the previous segment, there is a wide range of available techniques and tools to accommodate the diversity of NLP tasks. Generally, while the goal of each task shapes the methodological pipeline used, there is often considerable variation even among similar tasks. In the following section, and within the framework of the current study, a selection of relevant techniques is outlined.

2.2.1 Text Preprocessing

In general, text preprocessing encompasses several methods designed to restructure and organize text into a format more suitable for analysis. Figure 2, drawn from a recent 2022 review on preprocessing methods [15], presents a bar chart summarizing the overall usage percentages of these methods across the examined studies.

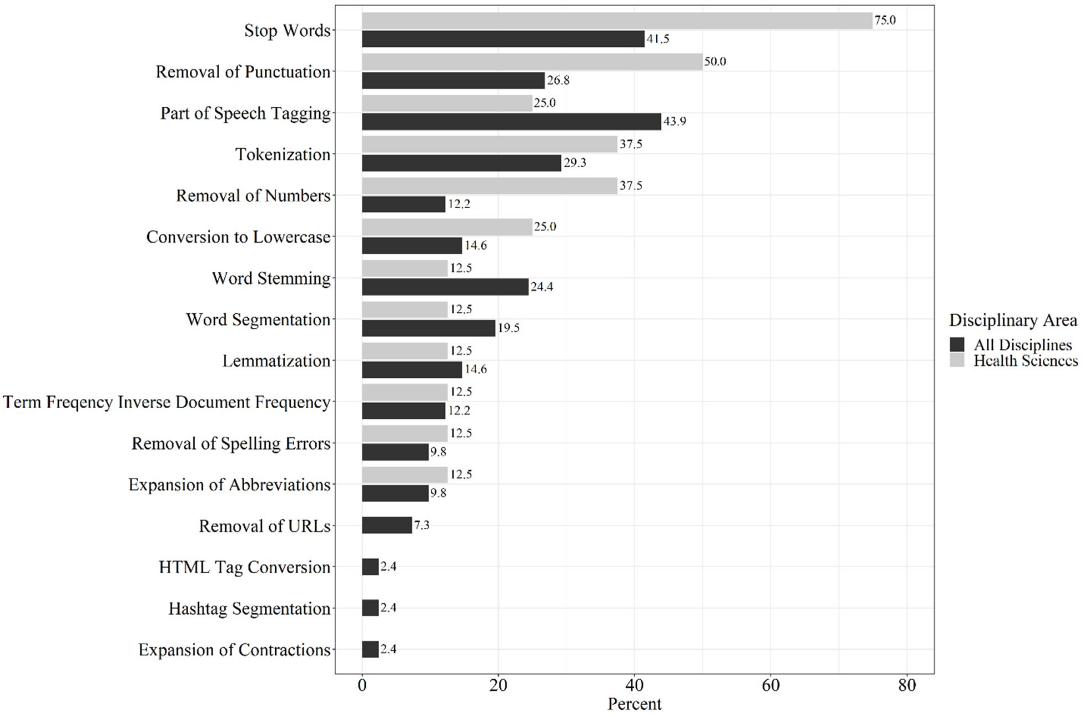


Figure 2: Overall Percentage Usage of Preprocessing Techniques on UT

As shown in the chart, although preferred preprocessing techniques vary by discipline area, stopword and punctuation removal, POS tagging, and tokenization are among the most popular. For health science articles specifically, removal of numbers and conversion to lowercase are also among the top five methods, while stemming is commonly preferred for articles from other disciplinary areas. However, it needs to be noted that only articles utilizing Unstructured Text (UT) were included in that review.

Moreover, its findings are partially corroborated by a more recent survey, conducted in 2024 [16]. The authors of this survey reported the prevalence of each preprocessing technique across the relevant articles they examined. They specifically identified studies that, alongside applying preprocessing techniques for their main Text Classification (TC) tasks, conducted comparative evaluations of those techniques. Among the most commonly applied methods were stopword removal (RSW), stemming (STM), lemmatization (LEM), conversion to lowercase (LOW), punctuation removal (RPT), and noise replacement (RNS)[3]. Figure 3 displays the frequency distribution of these techniques, as reported in the 2024 survey. For reference, Table 1 in the Appendix provides definitions of the acronyms used in Figure 3, along with an example of each technique.

[3] *Noise replacement* as a preprocessing technique does not refer to a single, distinct method. Instead, it encompasses a variety of approaches, such as removing HTML tags, boilerplate content, and Unicode artifacts through regular expressions, to ensure the text under analysis is free from uninformative elements.

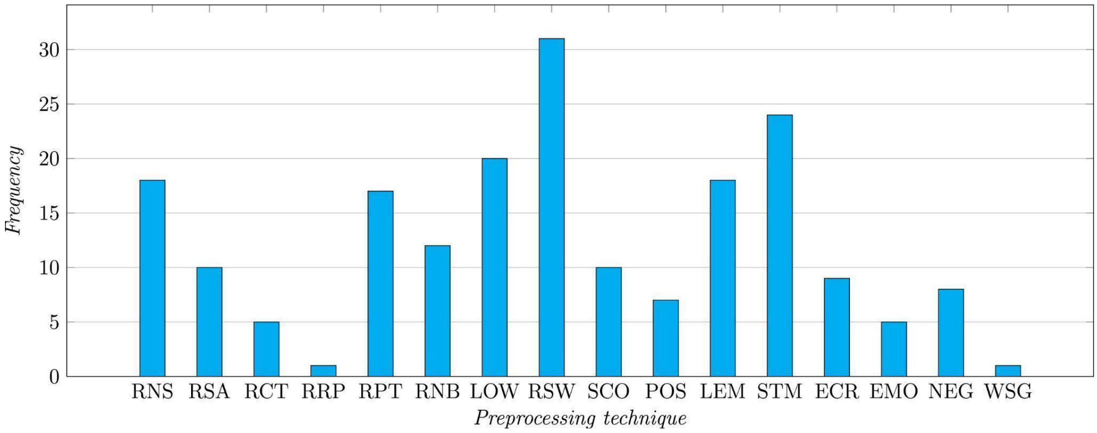


Figure 3: Frequencies of Preprocessing Techniques

The relevant articles identified in the survey served as benchmarks for assessing the main objective of the study: evaluating the performance of nine modern Transformers (RoBERTa, XLNet, ELECTRA, ANN, CNN, BiLSTM, NB, SVM, and LR) on four text classification tasks. The analysis was based on whether employing the most prevalent preprocessing techniques, as identified in the literature, led to improved performance. The results confirmed that text preprocessing strategies can indeed impact the performance of modern classifiers, including recent Transformers. However, it is important to note that the characteristics of the dataset play a significant role in determining the extent to which preprocessing techniques enhance model performance. Notably, Transformers appear to be less sensitive to preprocessing compared to simpler classification models, particularly when applied to large datasets.

Nevertheless, different techniques, and their combinations, still have varying effects on Transformer performance. For instance, while lowercasing is beneficial for ELECTRA, stopword removal tends to improve performance with RoBERTa and XLNet. For simpler classification methods, the takeaway is more straightforward as the choice of preprocessing techniques plays a crucial role in classification performance. Overall, given their varied impact on model performance, understanding and selecting appropriate preprocessing techniques, such as lowercasing, tokenization, stopword and punctuation removal are essential steps for optimizing a range of information extraction tasks.

2.2.2 Lowercasing

Lowercasing is among the most common preprocessing techniques in NLP applications and involves converting all characters in a text to lowercase. This step helps ensure consistency, as it removes case distinctions between words, so "Apple" and "apple" are treated as the same word. Thereupon, a word embedding layer would not assign separate vectors based on case differences. In general, lowercasing is particularly useful in tasks such as Information Retrieval (IR), where the goal is to focus on the meaning of the text rather than its formatting. By standardizing the case, lowercasing reduces the complexity of the text and helps improve the performance of ML models [16].

2.2.3 Tokenization

Tokenization is a fundamental preprocessing step in NLP that is essential for converting raw text data into a format that machines can process and understand effectively. In that raw text data, characters other than well-defined words can be found, such as numerical digits, symbols and punctuation marks, as well as whitespace characters. This blend imposes a demanding challenge for computers which are designed to process structured data. Thereupon, tokenization becomes a factor as it acts as a bridge between the complexity of human language and the structured, numerical input required by computational models. In simple terms, without tokenization, computers would lack the capacity to grasp the meaning and structure of natural language. Moreover, they would fall short in their capability to distinguish individual words, identify where sentences begin and end, or understand the relationships between different linguistic elements [17]. Figure 4 illustrates the essence of the tokenization functionality.

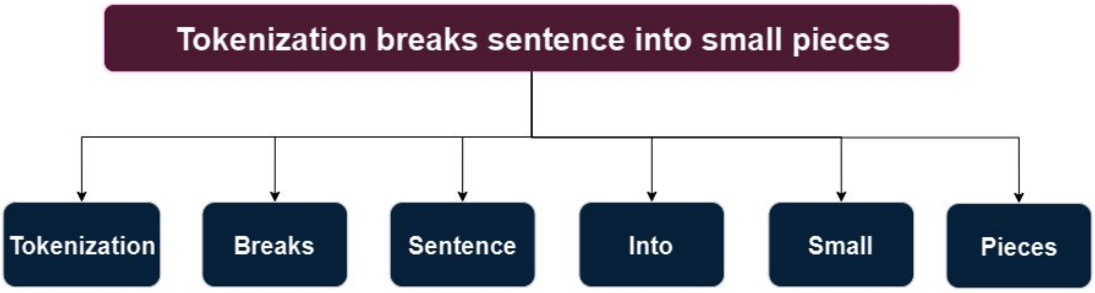


Figure 4: Essence of Tokenization

There are three primary categories of tokenization methods, which differ based on the level of granularity of the resulting tokens: character-level, subword-level, and word-level tokenization [17]. Character-level tokenization breaks down text into individual characters. This fine-grained approach is particularly useful for applications such as spelling correction, language identification, and language modeling. It is also advantageous for languages like

Japanese or Chinese, where words are not separated by spaces. Subword-level tokenization falls between character-level and word-level tokenization. It splits words into smaller, meaningful subword units using techniques like Byte Pair Encoding (BPE), WordPiece, and SentencePiece.

This approach is particularly effective for handling rare or out-of-vocabulary (OOV) words, improving performance on tasks involving diverse vocabularies. Finally, word-level tokenization, the most straightforward method, decomposes text into individual words, typically using whitespace or punctuation as delimiters. Word-level tokenization underpins many traditional and modern NLP techniques, such as Bag of Words (BoW), TF-IDF, Latent Dirichlet Allocation (LDA), and word embeddings. Figure 5 provides a schematic representation of these tokenization methods.

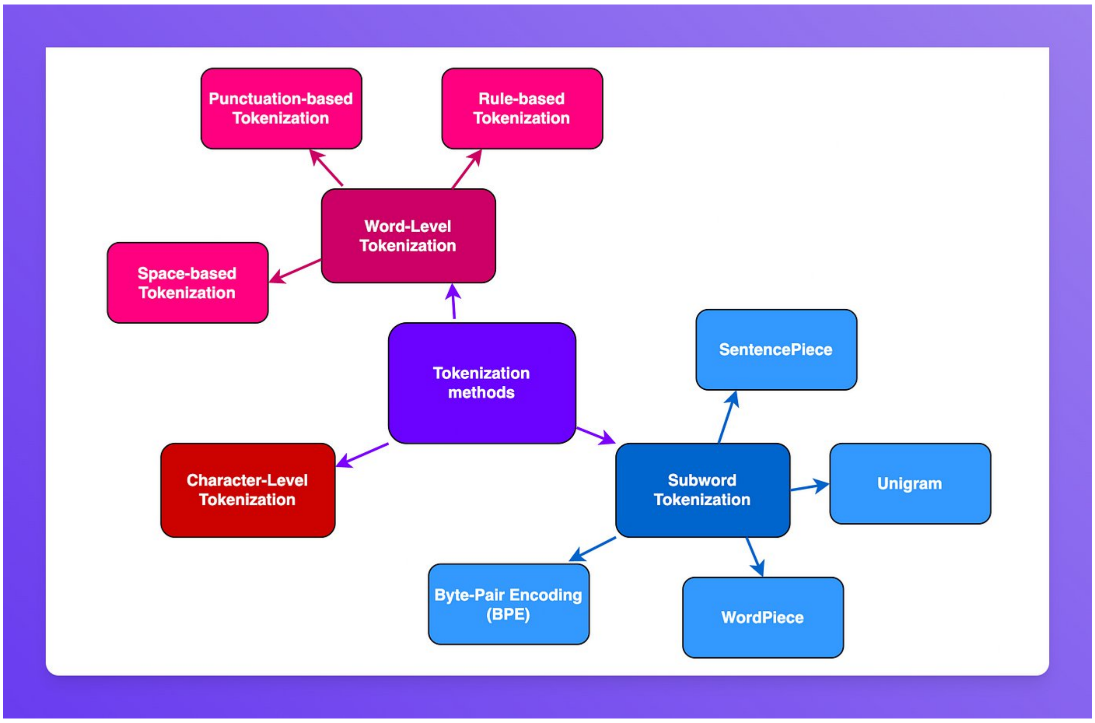


Figure 5: Schematic representation of the tokenization methods

Regarding the implementation of these methods, several well-established libraries within the NLP ecosystem offer robust tokenization capabilities. Among the most prominent ones are Natural Language Toolkit (NLTK), spaCy, Keras, and HuggingFace which stand out for their comprehensive tokenization tools.

Overall, the selection of a tokenization method, and the relevant tools thereof, is contingent upon the specific needs of the NLP project. While word-level tokenization may suffice for numerous tasks, subword and character-level tokenization offer greater effectiveness when dealing with uncommon words, varied vocabularies, and intricate language structures. Moreover, in a study published in 2022, Xue et al. [18] proposed a variant of the T5 architecture, referred to as ByT5, in the context of pre-trained language models. In their paper, they advocate for processing byte sequences, i.e., raw text, rather than tokenized text, arguing that byte-level models are more resilient to noise and demonstrate superior performance on tasks sensitive to spelling and pronunciation. If we are indeed moving towards a token-free NLP framework, this remains to be resolved. Nevertheless, for the time being, and regardless of the method chosen, tokenization continues as an integral component of most NLP pipelines.

2.2.4 Stopwords Removal

Stopwords are words with low discriminative power, meaning they contribute minimally to distinguishing between categories in text analysis tasks. They are common words, such as "a", "the", "and", and "so" in English, primarily serving grammatical or connective purposes without adding significant semantic value. Stopwords are frequently encountered in texts and vary by language and, at times, by domain. They generally carry little informative or predictive value. Consequently, stopword removal is a widely used preprocessing technique in text mining and IR to streamline text data and improve processing efficiency.

One primary benefit of stopword removal is that it reduces the volume and size of the dataset, which can enhance the effectiveness and accuracy of text mining applications [19]. However, in certain tasks, such as machine translation, text summarization, and sentiment analysis, removing stopwords can lead to a substantial loss of meaning, potentially degrading the quality of analysis. Therefore, stopword removal is not universally recommended and should be evaluated based on the specific task requirements [19].

Two main approaches to stopword identification are static and dynamic methods. The static, or traditional, approach relies on predefined lists of stopwords, which serve as a reference to identify uninformative tokens in text. While straightforward, this method may not capture domain-specific stopwords, necessitating adjustments for specialized text, such as scientific or legal documents. Nonetheless, a wide array of established stopword resources, like those provided by NLTK and SpaCy, are readily available and cover a broad range of general language needs. The dynamic approach, in contrast, uses statistical or data-driven techniques to identify stopwords based on their behavior within a specific dataset. This method does not depend on a pre-existing list but instead examines the dataset to determine which terms should be treated as stopwords.

Common techniques within this approach include methods based on Zipf's Law, mutual information, and random sampling of data chunks [16]. Zipf's Law identifies terms with low frequency for removal, assuming these terms are unlikely to carry meaningful information and lack discriminative power. Mutual Information measures the information content of each word by assessing its contribution to distinguishing specific categories or classes in the corpus. Thereupon, words with low mutual information values are evenly distributed across categories and thus offer minimal discriminative value.

Finally, random sampling uses Kullback–Leibler (KL) divergence to evaluate segments of the dataset, ordering terms within each chunk by their informativeness. These dynamic techniques allow for tailored stopword identification that can better capture relevant terms in specific contexts. However, they are more computationally demanding and often require fine-tuning to set appropriate frequency or significance thresholds. In practice, a hybrid approach can be employed, where a standard stopword list provides a baseline, and additional context-relevant stopwords are identified dynamically within the dataset. This combination can offer improved accuracy and relevance for specialized text mining applications.

2.2.5 Keyword Extraction

In the field of text mining and IR, keyword extraction is the process of identifying and extracting key terms or phrases from a document that convey essential information relevant to the specific task. By isolating these relevant terms, the main elements of a document can be recognized without the need to analyze its full content [20]. For instance, in the context of scientific articles, particularly their abstracts, such key elements may refer to various sections of a study, including methods, results, and conclusions. Once extracted and appropriately processed, i.e. vectorized, these elements can serve as input for NLP applications in ML, such as text clustering and classification, and for association rule mining to identify co-occurrence patterns [21], [22].

The diverse range of data types as well as the multitude of contexts and applications have led to the development of numerous approaches, techniques and tools for keyword extraction.

2.2.5.1 Direct & Dictionary-Based Matching

Direct matching shares features with dictionary-based methods, as both approaches rely on predefined information that guides the extraction process. In exact matching, this involves a list of terms that must match precisely, whereas a dictionary-based method may allow for synonyms, homophones, as well as stemmed or lemmatized forms to be detected.

In this framework, the quality and comprehensiveness of the list or dictionary plays a critical role in determining the value of the extraction process. Examples of predefined resources used in these methods include domain-specific lexicons, such as those containing biomedical terms [23].

Although these methods generally lack the adaptability to contextual variations offered by more advanced techniques, such as ML and deep learning, they can still be effective in scenarios where keyword and phrase consistency is expected. This can be encountered in specialized domains with well-defined terminologies [24] or in targeted search applications [25], such as e-commerce platforms or recommender systems.

Moreover, dictionary-based approaches have been employed throughout the years, both as standalone solutions [26], [27] and in combination with more sophisticated, context-aware methods [28]. In a recent work [26], a tailored dictionary was developed to remove stopwords in the Khmer language, while in [28], a geoscience dictionary was constructed and employed for matching and identifying geoscience concepts as part of a layered process for classifying geoscience reports and documents.. Additionally, while these approaches can be effective independently, hybrid methods may provide enhanced flexibility. For example, exact matching can be used to identify specific keywords, which could then be further refined using regular expressions (RegEx), fuzzy matching, or even pre-trained LLMs, depending on the specific requirements of the task.

2.2.5.2 Rule-Based & RegEx

Following the basic techniques of direct and dictionary-based matching, rule-based and RegEx are two additional traditional approaches commonly employed for keyword extraction.

Rule-based leverage predefined linguistic rules and domain-specific knowledge to identify the relevant keywords. Those rules can be crafted based on syntactic, semantic or statistical features of the text [20]. In general, although ML approaches often outperform rule-based models in terms of flexibility and scalability, the latter still hold significance due to their simplicity and their interpretability [20].

On the other hand, RegEx utilize formal language to describe and capture patterns in text through specific string-matching sequences. These sequences can represent characters, words, or even phrases, and are often applied to structured patterns such as dates or phone numbers. Over the years, RegEx techniques have been extensively used and highly optimized [27]. Within standard RegEx engines, the "leftmost-longest" matching approach is typically applied where the engine identifies the leftmost match that is also the longest, and subsequent searches begin at the rightmost position of the previous match.

In their study, Riveros et al. [29] extend this standard approach by introducing REQL, a modified query language capable of addressing all-match semantics[4], thereby enabling the capture of overlapping patterns. Similarly, Chida et al. [30] enhance the functionality of standard RegEx by incorporating real-world extensions, such as backreferences and lookarounds. These extensions transcend basic membership testing by enabling substring extraction, significantly broadening the practical applications of RegEx. These examples are indications of how advanced RegEx variants continue to offer solid solutions for data extraction challenges.

2.2.5.3 Fuzzy Matching

Fuzzy matching is another approach that can be used for keyword extraction, particularly when the predefined resources used in direct and dictionary-based matching are not sufficient or when the context is too complex for rule-based and regular expressions.

Fuzzy matching builds upon the mathematical principles of fuzzy logic, accommodating uncertainty and imprecision by capturing a continuum between a clear-cut match and a mismatch [31]. In text mining, this involves comparing input text to a set of predefined keywords or phrases, while allowing for flexibility in the matching process. This is achieved through algorithms that compute similarity scores between strings, such as edit distance, token-based similarity, or phonetic matching. Fuzzy matching proves particularly valuable in handling noisy or ambiguous data, where exact matches are impractical. A recent review on fuzzy techniques in text mining highlights that these methods are frequently combined with statistical approaches, as well as machine and deep learning models, to address primarily feature extraction tasks [31]. Moreover, Figure 6 illustrates the trajectory of publications and citations related to the use of fuzzy techniques in text mining on Web of Science (WOS) since 1993 [32].

[4] *All-match semantics* in the context of RegEx engines refers to a behavior where the RegEx engine attempts to find all possible matches of a pattern within a given input string, even if these matches overlap. This contrasts with the more common first-match or longest-match semantics, where the engine prioritizes specific types of matches based on its configuration (e.g., greedy or lazy matching). In engines supporting all-match semantics, the RegEx does not stop processing once it finds a match or skip over overlapping potential matches. Instead, it explores and captures every occurrence of the pattern that satisfies the given constraints, including overlapping instances.

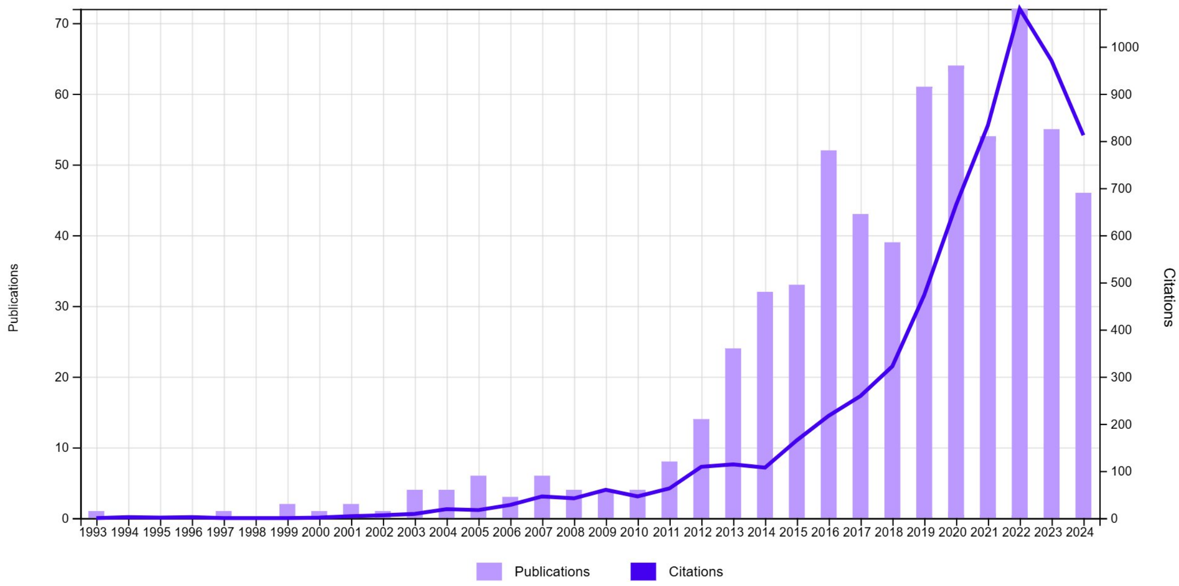


Figure 6: Co-occurrences of *fuzzy* & *text mining* on WOS

Although the data reveals a decline in the number of publications over the past two years, the high co-occurrence of fuzzy matching and text mining remains noteworthy. This decrease may reflect the growing adoption of LLMs, whose advanced capabilities increasingly address similar challenges in feature extraction and keyword analysis. Nevertheless, fuzzy matching continues to hold relevance as a complementary approach, particularly in applications requiring flexible, or domain-specific solutions.

2.2.5.4 Context-aware & Learning-based

The techniques presented so far, such as fuzzy matching and dictionary-based approaches, provide valuable tools for keyword extraction by addressing surface-level variations and exact matching challenges. However, these methods lack the ability to acknowledge the broader linguistic and contextual nuances in which keywords appear. Context-aware techniques, driven by advancements in deep learning, bridge this gap by leveraging semantic understanding to identify keywords based on their relevance and role within the text.

A cornerstone of these methods is the use of embeddings, which form the foundation for transitioning from traditional, surface-level approaches to semantically rich, context-aware systems [33]. Embeddings are dense, multi-dimensional representations of words or phrases, capturing both their syntactic and semantic properties. Unlike static embeddings from earlier models like Word2Vec [34] and GloVe [35], modern LLMs generate contextual embeddings, where the representation of a word dynamically changes depending on its surrounding text. This allows context-aware methods to discern subtle differences in meaning, such as distinguishing "mean" as a measure of central tendency from "mean" as an adjective which characterizes someone as unkind or unfair.

Furthermore, the role of embeddings in a keyword extraction process differs based on the type of method employed [36]. Within supervised frameworks, embeddings are utilized as input features for training deep learning models. This way, the keyword detection task is converted into a classification or regression problem [36]. In this setup, the model is trained to determine whether a candidate word in a text is a keyword or non-keyword, utilizing the local and global relationships captured by the embeddings. In contrast, unsupervised methods leverage embeddings to identify patterns and relationships without labeled data [36]. For example, embeddings enable clustering by grouping semantically similar words or phrases based on their proximity in the embedding space. Additionally, they support similarity-based techniques that rank potential keywords by comparing their embeddings to the broader context of the text.

In the presented context, both supervised and unsupervised methods refer to a class of approaches which aim at classificatory keyword extraction (CKE). Approaches that differ from generative keyword extraction [37]. CKE approaches operate by scanning contiguous text portions to ascertain whether a word is to be included in a pool of keywords, while models for generative keyword extraction focus, in addition to the extraction, on predicting - generating - keywords which are not present in the text [37]. Figure 7 [38] depicts a schematic representation of the generic architecture of a two-stage, supervised and usupervised, keyphrase extraction framework.

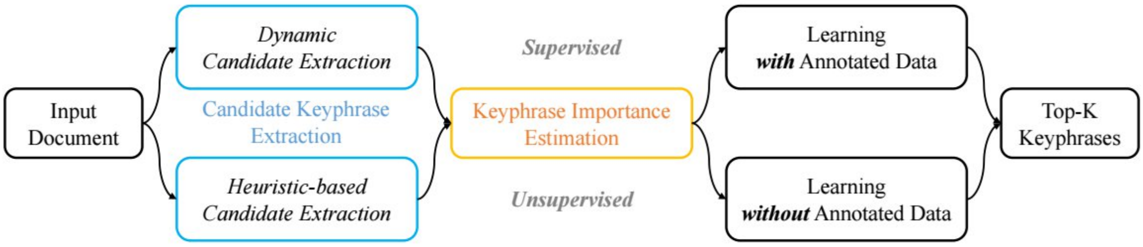


Figure 7: Two-stage generic pipeline for keyphrase extraction

The stages of this generic framework comprise the extraction of candidate phrases as a first step and as a second, the application of supervised or unsupervised methods to determine which of those phrases are actual keyphrases.

2.2.5.5 Extraction Evaluation

The most common evaluation metrics for a keyword extraction task are precision, recall and F1-score [36].

Precision is a metric for evaluating the accuracy of a model's prediction and it is defined as the ratio of the true keywords/phrases to the total number of predicted keywords/phrases made by the model:

$$Precision = \frac{True\frac{Keywords}{phrases}}{True\frac{Keywords}{phrases} + False\frac{Keywords}{phrases}}$$

Recall is a metric for evaluating the effectiveness of a model and is defined as the ratio of the true keywords/phrases to the total number of actual keywords/phrases in the text:

$$Recall = \frac{True\frac{Keywords}{phrases}}{True\frac{Keywords}{phrases} + False\frac{NonKeywords}{phrases}} F1 - Score{:}2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

[36].

To elaborate, execution time is a crucial parameter, especially as it influences the scalability of the method. This factor becomes particularly critical when extraction is performed in real-time applications. Another important consideration is data type, as the length of the input document affects the search space. Longer documents, such as full scientific papers or meeting transcripts, present more challenges during the extraction process, compared to shorter texts like abstracts or emails. Furthermore, the structural consistency of the document is also significant, as more standardized formats tend to facilitate the extraction process. Additionally, author-assigned keywords are often used as a gold standard for evaluation, particularly in datasets involving scientific articles or abstracts, where domain-specific expertise is crucial [36]. However, this approach can be double-edged. While a gold-standard collection of keywords helps assess the effectiveness of an extraction method, it can also bias the evaluation by excluding potentially relevant terms that were not included by the author.

Overall, the final evaluation of a keyword extraction model is a delicate process that depends on the application context. Nevertheless, it is also important to note that there are viewpoints which advocate for the use of semantic-based metrics, as opposed to traditional direct-matching ones, to better capture the nuanced meaning of keywords [38], [39].

2.2.6 Keyword Vectorization

After successfully extracting meaningful keywords from a text, the next crucial step in order to proceed with other advanced text analytics tasks is to convert these keywords into a numerical format that can be utilized by ML algorithms. This process, known as vectorization, is essential for transforming textual data into a structured form that computers can understand. Furthermore, through this process semantic relationships between words are captured, allowing algorithms to account for semantic differences instead of solely syntactic. Moreover, vectorized text can potentially be integrated with structured data, enabling more comprehensive analyses that combine textual and non-textual insights. In this section, some key vectorization techniques are outlined, ranging from traditional, statistical methods to more sophisticated, context-aware, and deep-learning-based ones like word embeddings and Transformer-based models [40].

2.2.6.1 BoW & TF-IDF

BoW and TF-IDF are two foundational methods used for text vectorization in NLP.

BoW, originally developed as a multi-purpose feature - variable - extraction method, has been extensively applied in computer vision for tasks such as classification, detection, and recognition [41]. Thereupon, the underlying principle of BoW remains consistent across disciplines; it converts features into numerical vectors.

In the context of NLP, BoW represents each unique word in the text as a feature and constructs a feature, numerical, vector for each sentence, with each dimension corresponding to the frequency of a specific word in the sentence. Despite its simplicity and ease of implementation, BoW has notable limitations. It fails to account for word order or the context in which words appear, leading to sparse vector representations that lack the ability to capture semantic relationships between words [40].

TF-IDF builds upon the BoW model by incorporating a weighted scoring system. Unlike BoW, which simply counts the frequency of each word in a document, TF-IDF adjusts the raw frequency by assigning higher weights to words that occur less frequently and lower weights to those that appear more commonly across the corpus [40]. This is basically its IDF component. Although this technique addresses the issue with the word importance, just like

BoW it falls short in the representation of the word relationships. This lack is what prompted the generation of the word embeddings.

2.2.6.2 Word Embeddings

Word embeddings is a more sophisticated text vectorization technique, created to account for the deficiencies of the previous methods to capture semantic word relationships. The differentiating element is that they generate high-dimensional vector spaces, with each word being represented by an array of values, instead of the basic, binary, representation they are given by traditional older methods. Various models have been developed for creating word embeddings, with Word2Vec and GloVe being among the most widely used and well-known approaches [42].

Word2Vec [34] is a neural network-based model designed to produce dense vector representations of words. Its core principle involves training a neural network to predict context words from a target word, or vice versa, thereby learning vector representations that encode the semantic meaning of words. The model captures semantic relationships between words through two primary architectures, Continuous Bag of Words (CBOW) and Skip-gram [42]. CBOW predicts a word based on its surrounding context whereas Skip-gram predicts surrounding context words from a given target word. Word2Vec has demonstrated efficacy in numerous applications, including recommendation systems and knowledge discovery tasks [42]. It has nonetheless its own drawbacks, as the handling of OOV words is done in a suboptimal manner by assigning random vectors to them. Moreover, using Word2Vec for learning new languages can be convoluted and tedious, as its parameterization in one language cannot be used in another; requiring that the training begins from scratch. [42].

GloVe [35] combines two methodologies to generate word embeddings. It leverages word co-occurrence statistics, capturing how frequently words appear together across a corpus, to derive broader semantic meanings. That materialized through the construction of a large matrix table in which the frequencies of the word pairings are recorded. Subsequently, through the application of matrix factorization techniques, that matrix is simplified, and numerical vectors are generated for individual words. Those vectors encode both the semantic meaning of the words as well as the relationships between them. GloVe extends the Word2Vec functionality as it exhibits comparable performance but with faster training and even with a small corpus [43]. However, like Word2Vec, it faces limitations in handling OOV words effectively.

2.2.6.3 Contextualized Embeddings

The introduction of Transformers marked a paradigm shift with a profound impact on text vectorization. Unlike earlier models that processed text sequentially, Transformers introduced the ability to process entire sentences simultaneously. This innovation enabled the resolution of long-standing challenges, such as polysemy (multiple meanings of a single word) and homonymy (words that are spelled and pronounced the same but have different meanings). Transformer-based models, including BERT [44], RoBERTA [45], ELMo [46], and UMLFiT [47] allow the generation of contextualized word embeddings, accounting for the surrounding context of a word. With this capability, a two-fold objective is achieved. Different embeddings are created not only for a polysemous word, but also for words with a unique meaning which take on different nuances based on the particular context. These are functionalities that could not materialize only with the local semantic meaning derived from the traditional and static methods. Figure 8 [42] presents a diagrammatic overview of central word embedding techniques.

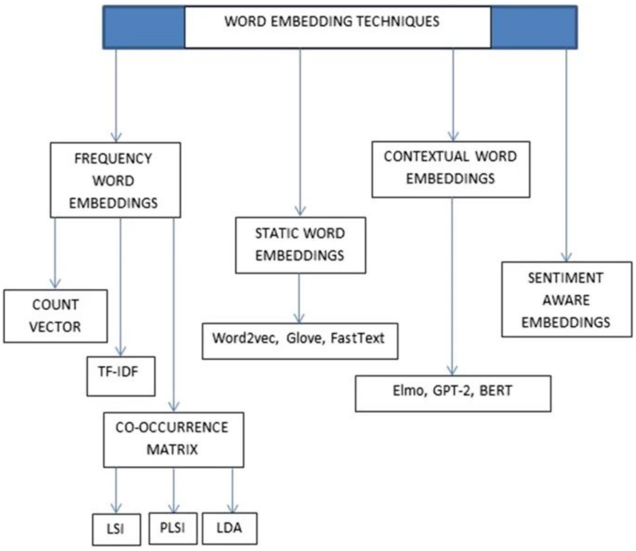

WORD EMBEDDING TECHNIQUES
FREQUENCY WORD EMBEDDINGS
STATIC WORD EMBEDDINGS
CONTEXTUAL WORD EMBEDDINGS
SENTIMENT AWARE EMBEDDINGS
COUNT VECTOR
TF-IDF
CO-OCCURRENCE MATRIX
LSI
PLSI
LDA
Word2vec, Glove, FastText
Elmo, GPT-2, BERT

Figure 8: Diagrammatic overview of embedding techniques

2.2.7 Clustering

Clustering is a fundamental data mining technique that groups data points based on measures of similarity. By identifying similarities between data points, clustering facilitates the discovery of patterns, insights, as well as the overall organization and summarization of data [48].

Unlike classification and regression, clustering is an unsupervised method, as it usually operates without labeled data. When there is no ground truth for the algorithm to match, the clusters are derived intrinsically through an exploratory process. Nonetheless, there are cases of semi-supervised learning, where some predefined categories guide the clustering process. Constrained K-Means and Semi-Supervised Fuzzy C-Means are examples of algorithms used in such scenarios [49]. Figure 9 presents a generic framework for a clustering operation [50].

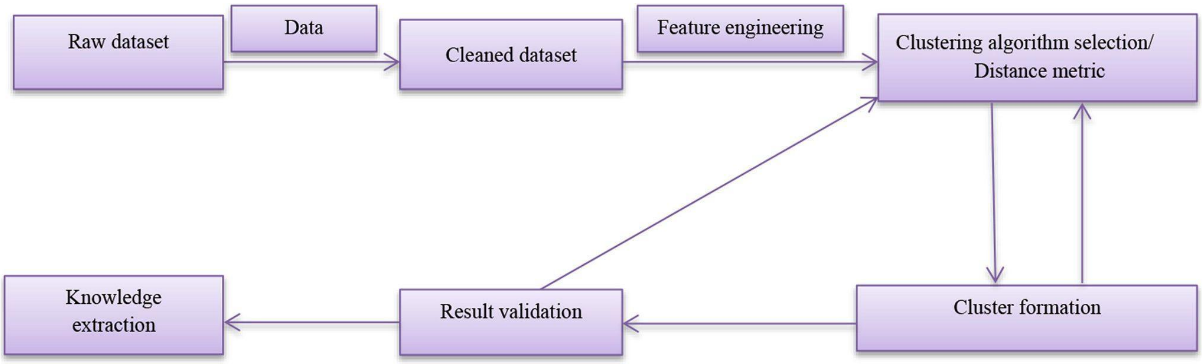


Figure 9: Diagrammatic representation of a clustering analysis workflow

Clustering algorithms are employed across a wide range of applications in daily life and can process various data types, including vectorized text. Figure 10 [50] illustrates the different data types used in clustering, while Figure 11, taken from a recent review [50], depicts the frequency of studies by clustering application domain, with text mining leading the list.

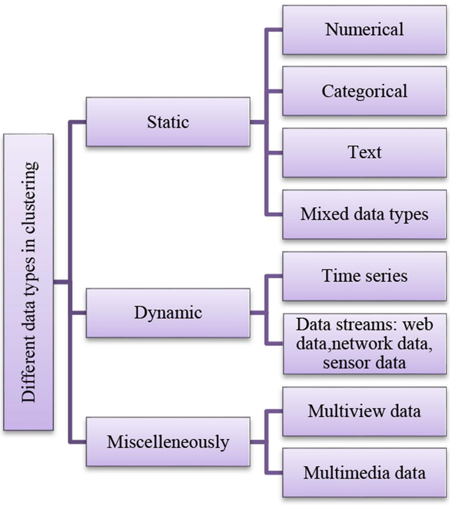


Figure 10: Data types in clustering

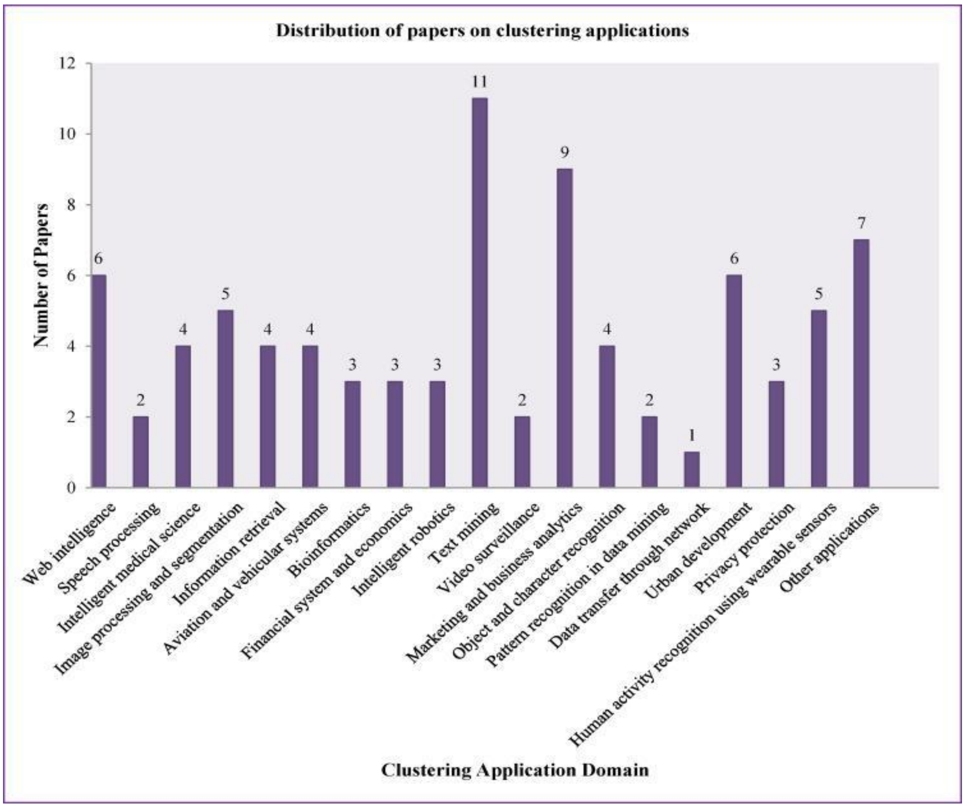


Figure 11: Number of studies per application domain

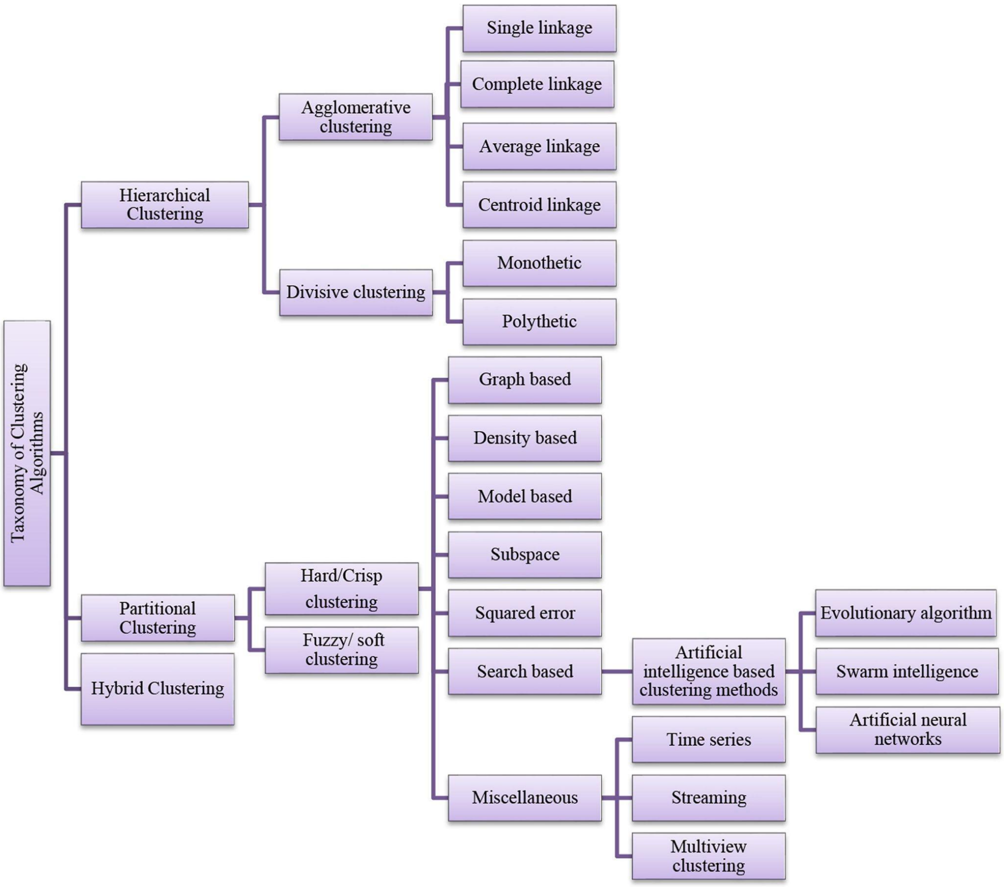
This variety, coupled with the rise of big data and increasingly complex data structures, drives the continuous advancement of clustering algorithms. Consequently, an extensive range of clustering algorithms has been developed to address the diverse demands arising from these complexities. The way clustering algorithms operate, the types of data they handle, the underlying assumptions about cluster structures, and their optimization criteria are central factors used to classify them into different categories.

The taxonomy of clustering algorithms is constantly evolving, with new leaves being added to the respective dendrogram. However, the parent branches remain consistent, encompassing hierarchical and partition-based clustering.

Hierarchical clustering operates on the principle of partitioning data objects into hierarchical levels [48]. These hierarchical levels are formulated in a bottom-up or top-down approach, or agglomerative and divisive respectively. The bottom-up method builds clusters from single objects. These objects are iteratively merged until a single cluster is formed or a stopping criterion is met. Conversely, the top-down method begins with all objects grouped in a single cluster, which is repeatedly divided into smaller clusters until each object forms its own cluster or a stopping criterion is reached. In this framework, the measure of similarity is crucial in determining how clusters are merged or split.

Partition-based clustering does not produce a hierarchical structure; instead, it partitions the dataset directly into distinct, homogeneous groups. The group formation process seeks to uncover the inherent groupings within the dataset, guided by an objective criterion function, most commonly the squared error[5]. In this framework, the dataset is iteratively partitioned and the measure of similarity determines the assignment of data points to clusters in a way that could minimize intra-cluster or maximize inter-cluster distance [48].

Partition-based clustering can be categorized into hard (crisp) and soft (fuzzy) approaches, depending on the nature of cluster membership assignment. In hard clustering, each data point is assigned exclusively to a single cluster, resulting in binary membership. In contrast, soft clustering assigns fractional degrees of membership to each data point, allowing it to belong to multiple clusters simultaneously. Figure 12 [50] provides an overview of the taxonomy of contemporary clustering techniques.



[5] The *squared error* as a criterion function evaluates the quality of clustering by measuring the total squared distance between each data point and its corresponding cluster representative (e.g. centroid, medoid, mode).

Figure 12: Taxonomy of clustering techniques

According to Al-Jabery et al. [51], there are nine key properties for evaluating the performance of a clustering algorithm. These properties overlap with those proposed by Singh et al. [50] in their recent review, with some minor differences. Common elements between the two proposals include scalability, robustness, high dimensionality, mixed data types, and parameter reliance. However, Al-Jabery et al. highlight the order of input patterns and the number of clusters as separate properties that affect clustering performance, while Singh et al. focus on computational complexity and convergence speed, treating them as distinct factors that require separate consideration. Furthermore, Singh et al. categorize these properties into two groups; *Dataset* and *Computational Methods*. Figure 13 provides a diagrammatic representation of the parametric characteristics they proposed, which influence the performance of a clustering algorithm.

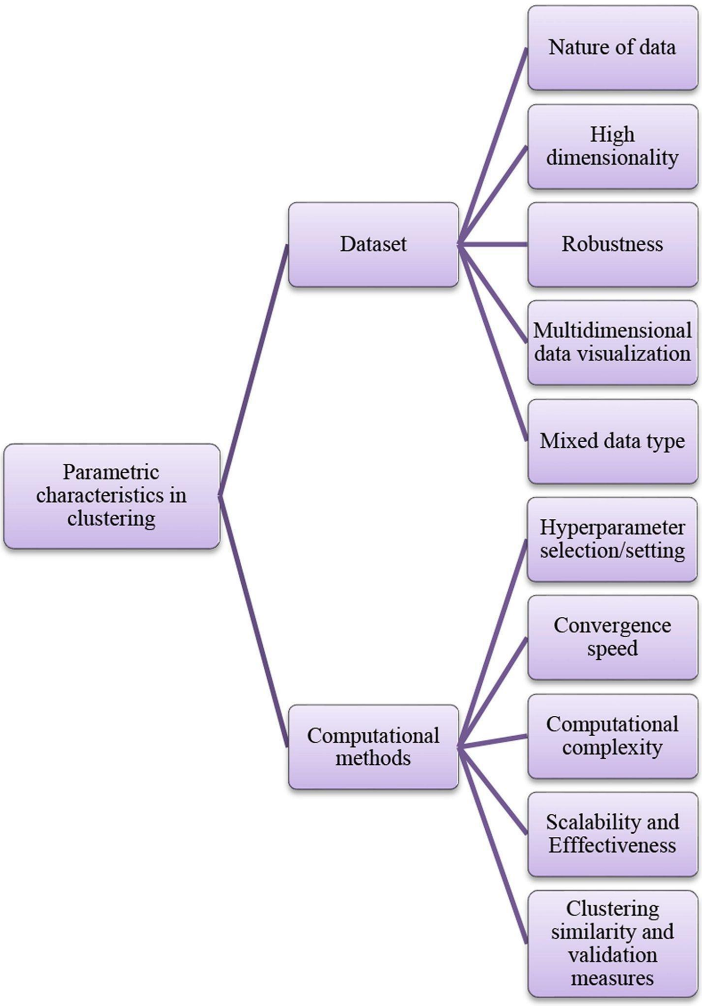

Parametric characteristics in clustering
Dataset
Nature of data
High dimensionality
Robustness
Multidimensional data visualization
Mixed data type
Computational methods
Hyperparameter selection/setting
Convergence speed
Computational complexity
Scalability and Efffectiveness
Clustering similarity and validation measures

Figure 13: Clustering parametric characteristics

Overall, these characteristics essentially constitute the open challenges in clustering evaluation. As a result, a wide range of evaluation metrics has been developed to assess each of these parameters, with their optimization indicating better clustering performance.

3 Related Work

Over the years, the rapid growth in scientific publications has created a vast corpus of information, offering immense potential for extracting meaningful insights. However, manually analyzing such large volumes of text is both impractical and inefficient. As a result, text mining techniques have become indispensable for analyzing scientific literature, enabling the discovery of patterns and knowledge inherent in these texts. To illustrate these applications, this section presents exemplary studies from the past five years that employ modern text mining, NLP, and ML techniques to extract valuable insights from scientific documents.

3.1 Psychology

Schiekiera et al. [21] tackled the issue of positive result reporting in psychotherapy studies by evaluating the classification performance of a fine-tuned SciBERT model and a pre-trained Random Forest model. Their analysis was conducted on a corpus of 1,978 in-domain and 300 out-of-domain abstracts. Building on earlier work by De Winter & Rodou [52], they compared these models against rule-based benchmarks using n-grams derived from phrases like "significant difference" and statistical parameters such as "p<.05." Abstracts were classified into two categories: "positive results only" or "mixed/negative results", where the distinction depended on the presence or absence of statistically significant results. A single negative result within an abstract categorized it as negative. To address potential bias from this sensitivity, a logistic regression classifier was employed as an additional benchmark, factoring in the abstract's length.

The SciBERT model achieved the best performance, with an accuracy of 0.86 on in-domain data and 0.85–0.88 on out-of-domain data. The Random Forest model showed solid but slightly lower performance, achieving accuracies of 0.80 and 0.79–0.83, respectively. Following validation, the SciBERT model was applied to infer patterns of positive result reporting in a corpus of 20,212 randomized control trial abstracts from PubMed. Logistic regression analysis revealed an initial rise in positive result reporting starting in the early 1990s, peaking in the early 2010s. This was followed by a modest decline and a subsequent moderate increase in the early 2020s.

For prediction tasks, linear regression analysis was conducted on the inference data, split into two periods: 1990–2005 and 2005–2022. A breakpoint around 2011 indicated a significant shift in reporting trends. Despite limitations, such as relying on abstracts instead of full texts and potential oversimplification in categorization, the study demonstrated that ML models are valuable tools for identifying and analyzing trends in positive result reporting.

Within similar research lines, Sokolova et al. [22] aimed to identify emerging trends and prominent topics within the fields of clinical psychology and psychotherapy. For this purpose, they utilized metadata from scientific papers available in the Microsoft Academic Graph database, focusing on publications from 2000 to 2019.

To meet their objective, they utilized a text mining system developed by Gokhberg et al. [53], which incorporates the Python Gensim package to enable a layered text mining approach alongside NLP and ML techniques. Their methodology consisted of two key steps: first, identifying a relevant set of terms for the subject area, and second, formulating thematic clusters based on these terms and creating trend maps.

For term identification, a two-fold approach was implemented, combining quantitative and qualitative methods. Initially, noun phrases related to clinical psychology and psychotherapy were extracted from the database using n-grams and skip-grams. A 200-dimensional vectorized space was then created using Word2Vec, grounded in an initial set of keywords curated by experts. The extracted phrases were subsequently filtered through expert evaluation and by assessing their cosine similarity to the *psychotherapy* vector. Terms with a cosine similarity greater than 0.5 were retained, forming the finalized list of terms.

The significance and dynamics[6] of each term were then calculated. After additional filtering to remove overly general terms, a final list of 234 terms was established, which included only those with a dynamics value exceeding 118%. Significance was defined as a measure of a term's importance, adjusted for variations in document volume across years, while dynamics reflected whether a term was gaining or losing prominence annually, thereby indicating its popularity.

To present their findings, the authors organized the terms into seven thematic clusters. Notably, terms could belong to more than one cluster, reflecting the conceptual overlap across topics. They also created trend maps showcasing the top 10 most significant terms within each cluster. Among the findings:

- In the *mental disorders* cluster, *gambling disorder* ranked as the most significant term.

[6] *Significance* calculates the overall normalized frequency of a term over multiple years by summing its yearly occurrence rates, weighted by the total number of documents each year. *Dynamics* measures the average annual percentage change in a term's relative significance over a specified time period.

- In the *symptoms, negative mental factors, and experiences* cluster, *moral injury* was the leading term.
- For the *psychotherapy* cluster, *mindfulness therapy* emerged as the most rapidly growing term.
- In the *interventions, methods, and factors of treatment* cluster, *medication-assisted treatment* held the highest absolute significance.
- In the *pharmacology and neurobiology* cluster, *computational psychiatry* was the top term.
- In the *skills, positive factors, and personal growth* cluster, *self-compassion* had the highest absolute significance.
- Finally, in the *general and other terms* cluster, *common mental health disorders* ranked highest.

In conclusion, this study is the first to systematically capture research trends in psychotherapy and clinical psychology, providing a strong foundation for future investigations. However, the study underscores the necessity of domain expertise, particularly in determining the relevance of keywords.

3.2 Urban Planning

In [54], Weng et al. investigated key topics in scientific publications at the intersection of AI, ML, and urban planning. Their dataset comprised 593 articles sourced from Scopus, and their methodological approach was divided into three phases.

Phase 1 involved clustering abstracts. The authors utilized the pretrained GPT-3 model to generate similarity embeddings for the abstracts. To reduce the dimensionality of these embeddings, they applied Uniform Manifold Approximation and Projection (UMAP), the results of which were then used as input for the HDBSCAN clustering algorithm. Since HDBSCAN requires the dimensionality of the input to be smaller than the number of data points (in this case, 593 abstracts), the authors reduced the dimensions to a range of 20 to 500. Additionally, they optimized the minimal cluster size parameter by experimenting with values between 10 and 50. The resulting clusters were evaluated using the Silhouette score to determine their quality.

Phase 2 focused on identifying keywords to represent abstracts in each cluster. The authors used the Stanford CoreNLP toolkit for preprocessing and extracted candidate keywords based on syntactic patterns. For each abstract and its candidate keywords, pairwise cosine similarity was computed, and the keywords were ranked according to their relevance to the corresponding abstract. To finalize the keyword list for each abstract, they applied the Maximal Marginal Relevance (MMR) algorithm, which prioritizes keywords with high similarity to the abstract while minimizing redundancy with other keywords.

Phase 3 involved grouping the extracted keywords. The same techniques used in Phase 1 for abstract clustering were applied to the keyword grouping process. Specifically, similarity embeddings were generated for the keywords, UMAP was used for dimensionality reduction, and HDBSCAN was employed for clustering, with the Silhouette score as the evaluation metric. However, unlike the abstract clustering process, where outliers were iteratively treated as separate clusters until all clusters contained fewer than 50 abstracts, only one iteration was performed for the keyword grouping. Outliers in this step were discarded, leaving only representative keywords for the cluster topics. The representativeness of each keyword group was assessed using the coverage[7] metric.

Regarding the abstract clustering, 24 clusters were identified, each containing between 10 and 46 abstracts. These clusters were distributed across three well-defined bands, with 17 clusters achieving good Silhouette scores (>0.5). Only two clusters had negative Silhouette scores, representing 8% of the total abstracts. In the keyword grouping, a total of 2,965 keywords were identified, which were organized into 98 keyword groups. On average, each cluster contained three keyword groups, and every cluster had at least one keyword group with good coverage (>50%).

Complementing the numerical results presented in the paper, the authors also developed a web-based visualization tool to provide a comprehensive view of the outcomes from each phase of the study.

In conclusion, this study presents a comprehensive approach to extracting keywords and identifying relevant topics, incorporating modern techniques such as the use of GPT-3 for generating embeddings.

[7] *Coverage* was calculated as the proportion of abstracts in the cluster that include at least one keyword from the group. Higher coverage means that the keyword group captures a larger portion of the abstracts in the cluster, making it more representative.

3.3 mHealth

In a systematic review conducted in 2019, Park et al. utilized text mining techniques on a corpus of 5,600 journal articles retrieved from WoS to examine trends in the field of mHealth research [55]. The review covered studies published over a decade, specifically from 2008 to 2018. Specifically, they analyzed the titles and abstracts of the retrieved studies using KH Coder 3, an open-source software package designed for quantitative content analysis and text mining. After removing stopwords, the authors extracted terms related to medical conditions, types of interventions, and study populations. The list of terms was finalized through consensus among researchers to ensure that only meaningful and relevant terms were retained. Subsequently, the relationships between the identified terms were evaluated using Document Frequency (DF) and a co-occurrence network. To measure the similarity between paired terms, Jaccard's coefficient and betweenness centrality were employed.

The analysis identified 48 medical terms, with mood disorders (9.7%), diabetes (9.4%), and infections (8.5%) being the most prevalent. Regarding intervention-related terms, 30 unique terms were identified. Among these, cell phone was the most frequently mentioned intervention (DF = 18.7%), followed by SMS (14.7%) and internet-based interventions (13%). A notable shift in trends was observed after 2012, with the use of mobile application interventions surpassing those of cell phone, SMS, and internet-based interventions. In terms of study populations, the term female appeared most frequently (DF = 20%), followed by healthcare workers (13.4%) and children (10.8%).

In analyzing the strength of relationships between paired terms, the strongest association was observed between female populations and pregnancy issues, with a Jaccard value of 0.22. Additionally, cell phone emerged as the central node in the co-occurrence network. Expectedly, it was linked to other intervention types such as SMS and mobile applications, as well as to a range of medical conditions, including mood disorders, anxiety, and diabetes.

In conclusion, this review applied straightforward text mining techniques to quantitatively assess trends in mHealth research, offering a foundational perspective for trend analysis in the field. However, the reliance on DF as a text processing measure introduces inherent biases. To address this limitation, incorporating TF-IDF could have helped mitigate these biases, thereby enhancing the robustness of the findings.

3.4 AI & Healthcare

In [56], Shin et al. examined research trends at the intersection of AI and healthcare technology. To achieve their study objectives, they collected 15,260 studies from Elsevier's Scopus database spanning a period of 55 years (1863–2018). They specifically utilized the abstracts of these papers for analysis, applying text and temporal frequency analysis techniques as a first step. For preprocessing, they employed the integrating functionalities of the OpenNLP morpheme analyzer to extract relevant terms from the abstracts. Out of 11,008 extracted words, 3,949 terms were retained based on criteria that included a TF-IDF value higher than 0.3 and occurrence in more than eight papers. These retained words were subsequently input into an LDA-based topic modeling process.

In the topic modeling step, the parameters α and β were set at 7.0 and 0.1, respectively, and 1000 iterations of Gibbs sampling[8] were performed. The authors also reported a similarity value of 0.05 between research subjects, although the specific similarity metric used was not disclosed. To supplement the topic modeling results, they conducted a two-mode network analysis using the top seven words assigned to each topic to demonstrate the lack of overlap between research topics.

The topic modeling identified seven research topics, with three major ones emerging: AI for Clinical Decision Support Systems (CDSS), AI for Medical Imaging, and Internet of Healthcare Things (IoHT), listed in order of prominence. In the final step, the authors employed word cloud analysis and ego network analysis. The word cloud targeted the top 100 allocation probabilities, while the ego network analysis focused on the top five words associated with each of the three major topics. For the first topic (AI for CDSS), the word Medical Doctor had the highest connection and mediation centrality and was linked to terms such as Medical Prescription, Hospital, and Decision Making. For the second topic (AI for Medical Imaging), key terms included Medical Image, Brain, Segmentation, Human Tissue, and Computed Tomography (CT), highlighting research themes related to augmented reality. For the third topic (IoHT), the top five words were Sensor, Body Area Network (BAN), Monitoring, Real-Time, and Safety. Among these, Sensor exhibited the highest centrality and mediating centrality, and was closely connected to terms like IoT, Accelerometers, and Human Activity

[8] The *alpha (α), beta (β),* and *Gibbs sampling* values are important parameters and techniques associated with LDA and influence the quality and efficiency of the topic modeling process. The *α* value controls how concentrated the topics are in documents. The *β* controls how concentrated words are in topics, while the *Gibbs sampling* refers to the number of iterations - cycles - before convergence.

Recognition (HAR), indicating a research trend towards personalized healthcare.

In conclusion, although written in a convoluted way, this paper provides a comprehensive overview of the research trends in AI and healthcare. A notable limitation that could be attributed to it, is that it lacks some methodological details regarding the performed analyses, resulting to a not so clear reporting.

Overall, these studies exemplify contemporary techniques for retrieving information from scientific documents, particularly abstracts. In the field of psychology, the studies by Schiekiera et al. [21] and Sokolova et al. [22] assert their pioneering roles in addressing their respective challenges through the analysis of information extracted from scientific abstracts. Although the authors of the presented studies acknowledge the use of abstracts instead of full documents as a limitation, this approach could gain traction due to its accessibility and the increasingly standardized reporting practices. Furthermore, a key insight is that information found in abstracts is effectively utilized to identify trends in research and reporting within the domain under examination.

4 Materials & Methods

This section outlines the methodology and materials used to achieve the study's objectives. More specifically, we provide a detailed description of the following processes: data collection, data cleaning and preprocessing, terms retrieval, terms vectorization, clustering models, as well as the statistical analyses.

4.1 Data Collection

Data were collected using 3 highly regarded databases in the fields of health, psychology and behavioral sciences, i.e. Elsevier's Scopus, PubMed's MEDLINE and Ovid's PsycINFO. For the purposes of the current study the abstract as well as basic metadata of psychology articles published in the mentioned databases between 1995 and 2024 were retrieved. Basic metadata comprised *"Title", "Journal", "Publication_Year", "Authors", "Abstract", "DOI", "ISSN", "Volume", "Issue", "Pages", "Keywords", "Publication_Type" and "MeSH_Terms"*.

For retrieving articles from PubMed, its publicly available API, Entrez Programming Utilities (E-utilities), was employed. The specific search term applied was: `'(Psychology[MeSH Major Topic]) AND ("1995/01/01"[Date - Publication] : "2024/08/15"[Date - Publication]) AND (English[Language]) AND (Clinical Trial[pt] OR Randomized Controlled Trial[pt] OR Observational Study[pt] OR Case Report[pt] OR Comparative Study[pt] OR Meta-Analysis[pt])'`. This search initially yielded 1,578 results. For the complete script, see "*dataPubMed.py*" in the provided Github repository [57].

For Elsevier's Scopus, the non-commercial SCOPUS Search API was used. The query targeted specific psychology journals: *`'((EXACTSRCTITLE("Psychological Medicine") OR EXACTSRCTITLE("Clinical Psychology Review") OR EXACTSRCTITLE("Cognitive Therapy and Research") OR EXACTSRCTITLE("Psychological Science") OR EXACTSRCTITLE("Journal of Clinical Psychology in Medical Settings") OR EXACTSRCTITLE("Behavior Research and Therapy") AND DOCTYPE(ar))'`*. This query initially returned 30,483 results. For the complete script, see "*dataElsevier.py*" in the provided Github repository [57].

For Ovid’s PsycINFO, the Advanced Search feature was employed. 'Psychology' was set as a major keyword, and additional filters were applied, including 'Articles with Abstracts,' 'APA PsycArticles,' and 'Original Articles'. The initial query returned 195,192 results.

Moreover, the date range was specified as a separate parameter in Elsevier’s Scopus retrieval script and applied as an additional limit in Ovid’s PsycINFO search. The selection of Elsevier’s Scopus journals and the application of the 'APA PsycArticles' and 'Original Articles' limits in PsycINFO are aligned with the study’s objectives. The chosen journals are known for publishing empirical studies, and limiting them to 'Original Articles' ensured the exclusion of publication types like editorials, book reviews, corrections, and reports.

4.2 Cleaning & Preprocessing

The initial corpus consisted of a total of 227,253 records. After removing duplicates, as well as records without available abstracts, and outliers such as editorials and commentaries, the final dataset amounted to 85,452 articles. Due to the qualitative and contextual nature of the data, substituting missing values with generic placeholders or statistical measures would not preserve the integrity of the information. Therefore, instead of attempting imputation, missing values were consistently labeled as “n/a” to indicate the absence of a value without distorting the analysis. Moreover, to further ensure data integrity and consistency, redundant information such as author ranking numbers and boilerplate texts was removed and data was coded in a consistent format for each feature. Additionally, publication years were validated to confirm they fell within the targeted date range.

For the keyword extraction, three glossaries were drawn from relevant, publicly available, psychology textbooks [58], [59], [60]. Each glossary includes method-related and statistical single and multi-word entries (hereafter jointly referred to as 'terms') commonly encountered in psychological research. Before merging the glossaries, duplicates were removed as well as those terms that refer to ethical guidelines, such as “anonymity” and “APA ethics code” and generic methodological concepts, such as “heuristics” and “deductive reasoning”. The final glossary consisted of 365 terms, serving as the gold standard collection for the extraction process. For the complete glossary refer to the *glossary_Hyphenated.json* and *glossarySpaced.json* in the provided GitHub repository [57].

After finalizing the glossary, the first preprocessing step involved converting all glossary terms and the corpus of abstracts to lowercase. Multiword terms from the glossary appearing in the abstracts were hyphenated using Python's regular expressions module to preserve their integrity during tokenization. Additionally, Python's inflect library was used to handle both singular and plural forms of the terms. For the complete script, see "*prepro_Hyphen.py*" in the provided GitHub repository [57].

The next preprocessing steps involved stopword removal and tokenization. The prior hyphenated terms were replaced temporarily by placeholders to ensure they were treated as single units during tokenization. They were converted back to their hyphenated form after the completion of tokenization. Moreover, numbers, decimals, and percentages embedded in the text were processed using custom regular expressions. This ensured that numeric values (e.g., 75%, 0.8, =3.14) were correctly identified, extracted, and handled separately from other tokens. The NLTK package was used for the preprocessing steps, along with the *FreqDist* function to count occurrences of glossary terms in their exact form. For the complete script, refer to "*prepro_NLTK.py"* in the provided GitHub repository [57].

4.3 Terms Retrieval

For the retrieval of the terms from the abstracts, two approaches were combined on the preprocessed text. Specifically, two variants of the final glossary were created, one with hyphenated terms and another with spaced terms. Direct string matching was first applied using the hyphenated glossary, aiming to retrieve terms that had been previously hyphenated. For terms that were not directly matched, fuzzy string matching was applied using the spaced version of the glossary. To account for minor variations, the threshold for fuzzy matching was set at 90%. The *RapidFuzz* library was used for this purpose, specifically its *fuzz.partial_ratio* function.

For the complete retrieval script and glossaries, refer to *direct&fuzzy_NLTK.py*, *glossaryHyphenated.json*, and *glossarySpaced.json* in the provided GitHub repository [57].

4.4 Clustering Analysis

To identify inherent method groupings of the examined abstracts the retrieved terms were vectorized and we used their vectors to feed both an unsupervised a semi-supervised clustering model. A detailed description of these processes is presented in this section.

4.4.1 Terms Vectorization

The terms extracted from the abstracts were vectorized before being used in downstream analysis. For this purpose, the SciBERT model was employed. SciBERT is a pre-trained language model designed specifically for scientific text, making it a suitable choice given the nature of the dataset in the current study. CLS embeddings were generated for the retrieved terms within the context of their corresponding abstracts. The CLS token is used to represent the entire input sequence (the abstract plus the key term). The embedding corresponding to this token is then used as the representation of the key term within the context of the abstract. For terms that appear multiple times within the same abstract, the average of their embeddings is computed. Additionally, to obtain a single embedding for each abstract, the embeddings of all the terms within that abstract are aggregated using mean-pooling. For the complete script, refer to *embeddings.py* in the provided GitHub repository [57].

4.4.2 Unsupervised Clustering

For the identification of inherent groupings within the examined abstracts, the keyword embeddings generated from the process previously described were used. Although the embeddings for each keyword were averaged, SciBERT inherently retains 768 dimensions for each individual embedding. To address this, UMAP was applied. UMAP's functionality aids not only in reducing the dimensionality of the embeddings but also in preserving the local and global structure of the data. Additionally, UMAP visualizations provide an initial exploration of the data, which can guide the clustering analysis more efficiently.

UMAP allows for a set of parameter settings that influence the clustering results. Before settling on a final UMAP configuration, several parameters were examined. Specifically, a range of values for the number of neighbors, the minimum distance, the distance metric, and the number of components was tested. Table 2 summarizes the different tested parameters. Configurations that appeared most promising through the UMAP visual exploration were selected for testing with the clustering algorithms. For the complete script, refer to *umapReductionUnsupervised.py* in the provided GitHub repository [57].

Table 2: UMAP Parameter Testing

Parameter	Values
n_neighbors	5, 10, 15, 30, 50, 100
min_dist	0.1 - 0.5
distance metric	cosine, correlation,

	euclidean
n_components	2, 3, 5, 10, 50

K-Means and DBSCAN were the two clustering algorithms that were tested. As with UMAP, their parameterization affects the clustering performance. For DBSCAN, the different parameters tested can be found in Table 3. For the complete script, refer to *cluster_DBSCAN.py* in the provided GitHub repository [57].

Table 3: DBSCAN Parameter Testing

Parameter	**Values**
eps_values	0.2 - 0.8
min_samples_values	2 - 8
distance metric	cosine, correlation, euclidean

For k-means, the different parameters tested can be found in Table 4. For the complete script, refer to *cluster_k-means.py* in the provided GitHub repository [57].

Table 4: K-Means Parameter Testing

Parameter	**Values**
n_clusters	3 - 9
random_state	42
max_iter	300
init_method	k-means++

For the evaluation of the clustering algorithms, the *Silhouette score* was used as a quantitative metric. Furthermore, the t-distributed Stochastic Neighbor Embedding (t-SNE) technique was employed for visual inspection of the clustering results in high-dimensional data. t-SNE includes parameters that affect the visualization quality but do not influence the clustering performance itself. To ensure efficient exploration, t-SNE settings were tested only on the clustering models with the highest Silhouette scores. Table 5 presents the tested t-SNE settings.

Table 5: T-sne Parameter Testing

Parameter	**Values**
n_components	2
random_state	42
perplexity	20 - 60 (increments of 10)
learning_rate	200 - 600 (increments of 100)
n_iter	1000 and 2000
metric	cosine

Outliers were identified for the chosen clustering model based on two specific criteria: a Silhouette score lower than 0.2 or a distance greater than 2.0 from the cluster centroid. Since the number of outliers (115) was small relative to the dataset's size, and no discernible pattern emerged upon manual inspection of their corresponding abstracts, it was decided not to apply specific treatments for these points. For the complete script, refer to *clusterOutliers.py* in the provided GitHub repository [57].

4.4.2.1 Cluster Analysis

To further analyze the composition of the clusters, descriptive statistics were calculated. Specifically, the following metrics were computed to gain insights into the clusters' characteristics: *Silhouette coefficients, homogeneity indices, cluster size, total terms, and unique terms* per cluster. For the full script used to calculate these metrics, refer to *clusterDescriptivesUnsupervised.py* in the provided GitHub repository [57].

Additionally, to assess the importance of each cluster's terms, TF-IDF scores were computed. A key difference from typical TF-IDF calculations is the use of binary counts instead of word frequencies for the TF component. This means that the TF value is 1 if a term is present in an abstract, and 0 if it is absent. The frequency of the term's appearance is not considered, so the focus is on whether a term is "present" or "absent" rather than how often it appears. For the full script, refer to *tf-idf_Binary.py* in the provided GitHub repository [57].

Furthermore, Jaccard similarity scores were calculated based on the TF-IDF term scores to measure pairwise similarity between the clusters. For the full script, refer to *jaccardSimilarityUnsupervised.py* in the provided GitHub repository [57].

4.4.3 Semi-supervised Clustering

In addition to the unsupervised clustering approach utilized in this study, a semi-supervised technique was explored to potentially enhance the clustering outcomes. For this purpose, a weighting scheme was developed, assigning varying weights to the glossary terms. The goal of this weighting approach was to guide the clustering algorithm in emphasizing key concepts with high theoretical discriminative value. Figure 14 in the Appendix depicts the employed weighting scheme for the emphasized terms. The rest of the glossary terms were assigned a weight value of 1.

The weight assignment was applied on the embedding level. More specifically, as a first step, the average term embeddings were calculated. Given the usage of SciBERT for the generation of the embeddings, a single term had initially different embeddings across different abstracts. Thereupon, those different term embeddings were averaged by summing them and dividing them by the respective total term count. After the calculation of the averaged term embeddings for each term found in an abstract, weights were applied according to the mentioned weighting scheme, and the abstract embeddings were updated based on the weighted, averaged, term embeddings. For the full scripts with which the averaged term embeddings were produced, the weights were assigned, and the abstract embeddings were updated refer to *avgTermEmbed.py* and *weightedAbstractEmbed.py* in the provided GitHub repository [57].

The clustering method closely followed the unsupervised approach, with some modifications to tested parameters. UMAP reduced embedding dimensionality, excluding n_neighbors=50, 100 and n_components=50 while cosine was the sole distance metric examined. For the complete script of the UMAP technique, refer to the *umapReductionSemiSupervised.py* in the provided GitHub repository [57].

Regarding the clustering algorithm, only k-means was tested, with the same parameter values as in the unsupervised approach. Furthermore, the *Silhouette score* and *t-SNE* were used for evaluation purposes in the same manner as in the unsupervised approach. The decision to test a subset of the parameters was based on the results derived from the unsupervised approach. For the complete script of the clustering refer to *cluster_k-means_SemiSupervised.py* in the provided GitHub repository [57].

Finally, cluster analysis comprising descriptive statistics and TF-IDF scores was conducted to provide insights on the clusters' composition. For the full script used to calculate these metrics, refer to *clusterDescriptivesSemiSupervised.py*, and *jaccardSimilaritySemiSupervised.py* in the provided GitHub repository [57].

4.5 Frequency & Trend Analyses

To gain an overall perspective of the dataset regarding term occurrences, the corpus of abstracts was analyzed independently of cluster formation. More specifically, we calculated descriptive statistics to summarize the presence and distribution of extracted terms across the abstracts, including the proportion of abstracts containing terms, the average number of terms per abstract, the median, and the standard deviation. For the complete script used to calculate these descriptives, refer to *termsStats.py* in the provided GitHub repository [57].

For the assessment of the overall importance of the retrieved terms and their evolution over time, frequency and trend analyses were conducted. This analysis included calculating the frequencies of individual terms and term pairs, as well as their respective TF-IDF scores. The TF-IDF calculation follows the same binary logic as described for the clusters above; however, the TF term is adjusted based on the total number of abstracts in each year. For the complete scripts used to calculate the weighted frequencies and the interactive dashboards, refer to *tf-idf_AllAbstractsByYear.py, interactiveTF-IDF_AllAbstractsByYear.py* and *co-occurrences.py*, *interactive_co-occurrences.py* in the provided GitHub repository [57].

Finally, a trend analysis was conducted to explore temporal variations in term occurrences. Specifically, we analyzed the proportion of abstracts without term presence for each year, calculated as the annual number of abstracts without terms divided by the total number of abstracts for that year. For the complete script used to calculate this, refer to *noKeys_DualAxis.py* in the provided GitHub repository [57].

Figure 15 shows a schematic representation of the employed methodology.

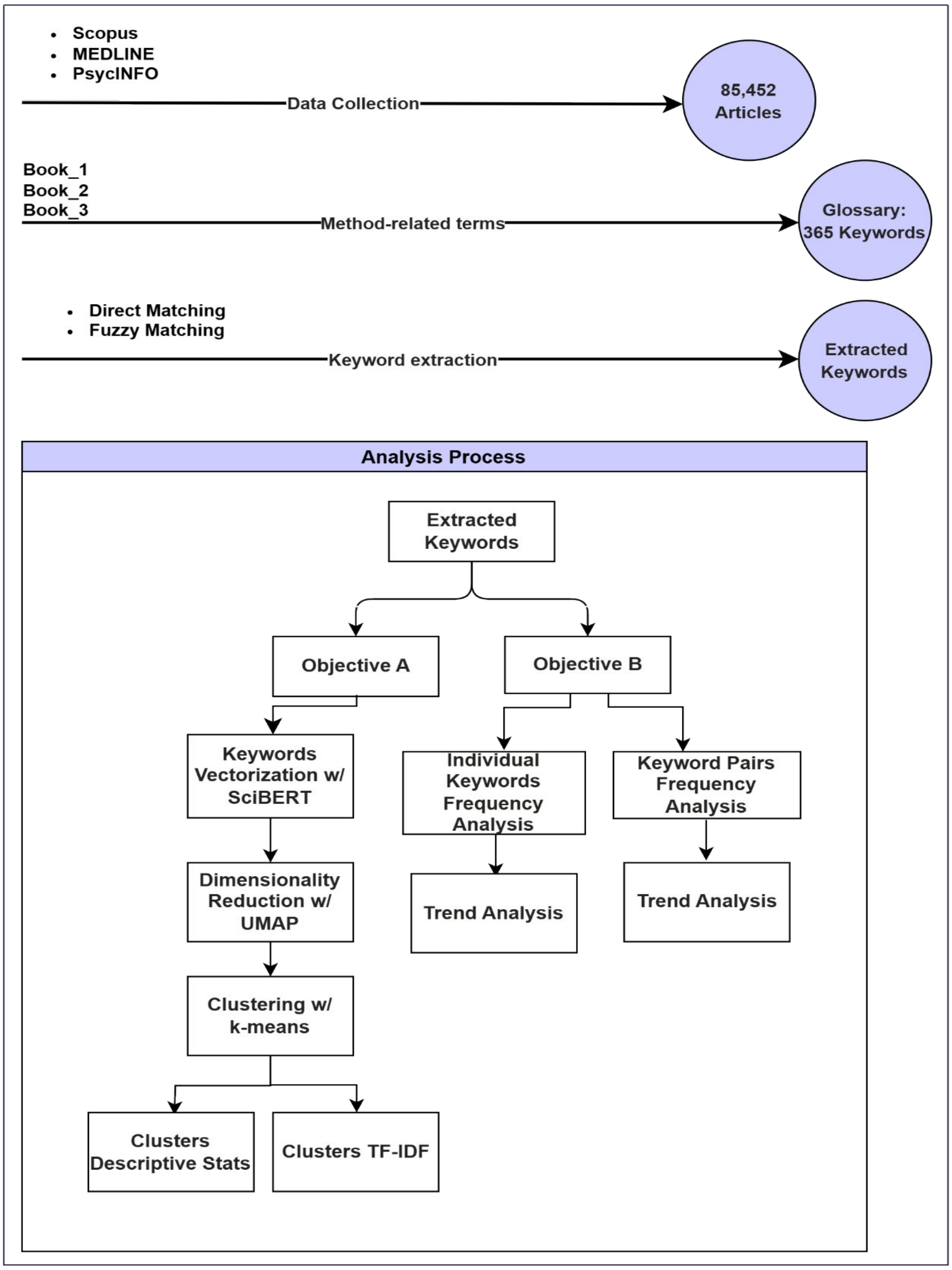


Figure 15: Schematic representation of the employed methodology

5 Results

This section presents the results of our research. For the clustering analysis, we report the tested UMAP configurations, descriptive statistics of the formed clusters, and evaluation metrics, including Silhouette scores and t-SNE visualizations, for both the unsupervised and semi-supervised approaches. Regarding the frequency and trend analyses, we provide descriptive statistics on term occurrences, irrespective of cluster formation, along with the corresponding TF-IDF scores. Additionally, we examine temporal trends, including the annual proportion of abstracts without any terms.

5.1 Clustering Analysis

5.1.1 Unsupervised Clustering

The UMAP configuration that yielded the best clustering results, based on the Silhouette score and the inspected UMAP and t-SNE visualizations, had the following parameter settings: 10 neighbors (n_neighbors), 0.1 minimum distance (min_dist), cosine distance metric, and 50 components (n_components).

The best clustering model was based on the k-means algorithm, which produced 6 clusters. The parameter settings for k-means were: random_state set to 42, max_iter at 300, and init_method set to k-means++.

For t-SNE, the parameter settings were: random_state also set to 42, perplexity at 40, learning_rate at 300, n_iter at 2000, and early_exaggeration at 18, with cosine as the metric. This combination of parameters resulted in a Silhouette score of 0.765. Figure 16 depicts the respective t-sne visualization.

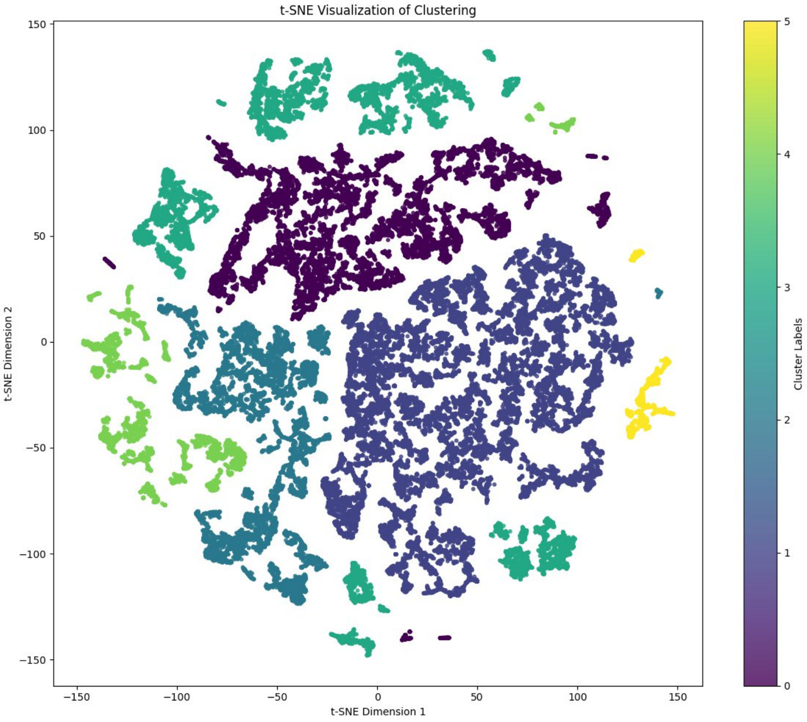


Figure 16: t-sne Visualization of Unsupervised Clustering

As observed in the t-SNE visualization, the resulting clusters varied in size. The number of records in each cluster is presented in Table 6.

Table 6: No. of records per cluster

	Records
Cluster 0	15078
Cluster 1	24890
Cluster 2	10301
Cluster 3	10712
Cluster 4	4716
Cluster 5	1095

Additionally, the proportional composition of the clusters relative to the overall dataset is illustrated in Figure 17, where the size of each slice corresponds to the proportion of records in each cluster.

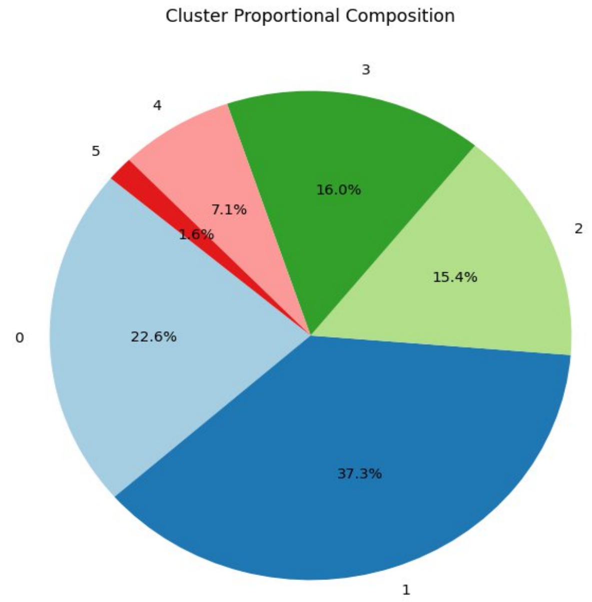


Figure 17: Proportional Cluster Composition

Four out of the six clusters appear to be more balanced in size, while clusters 4 and 5 are significantly smaller. The respective Silhouette coefficients, mean and standard deviation (sd), for each cluster can be found in Figure 18.

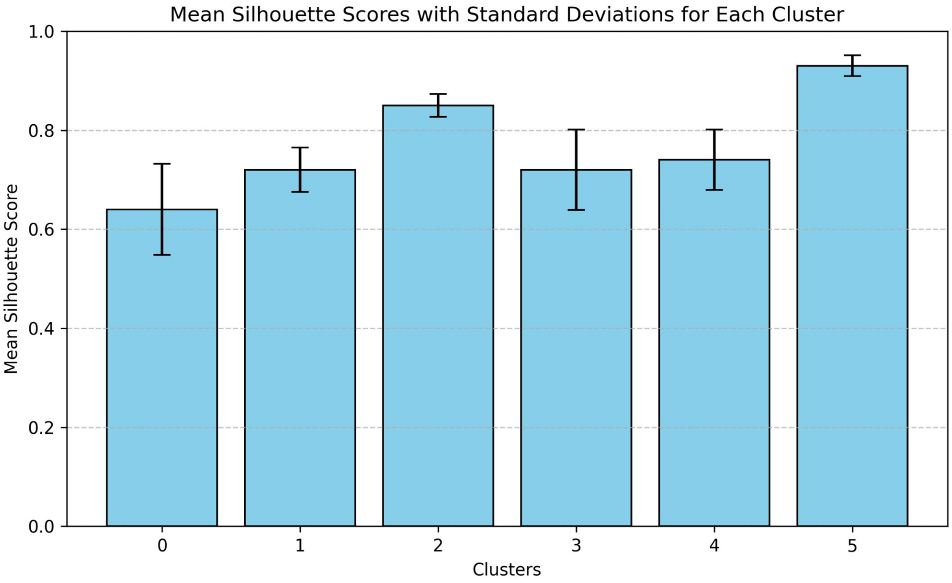


Figure 18: Mean Silhouette Scores with SD per Cluster

All clusters have a Silhouette score above 0.6, with cluster 0 having the lowest (0.64), and cluster 5 the highest (0.93). Furthermore, SD, represented by the respective error bars, is low for all clusters, with cluster 0 having the highest (0.09).

Figure 19 depicts the compactness of each cluster in relation to its separation from other clusters. Compactness is represented by the average distance of points in a cluster to their centroid, while separation as the smallest distance between the centroid of a cluster and any other cluster centroid.

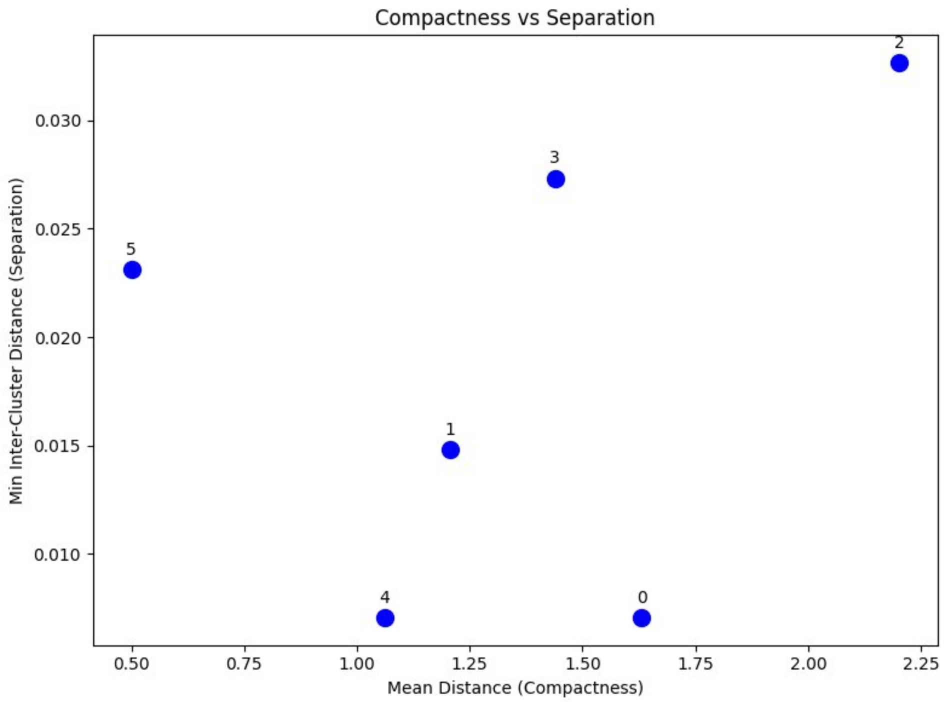


Figure 19: Compactness vs. Separation

Higher values in the x-axis indicate that points are more loosely grouped, while higher values in the y-axis suggest that the cluster is well-separated from others.

To illustrate the composition of terms within the clusters, Table 7 presents the total number of terms in each cluster, along with the unique terms identified in each one. Additionally, the TF-IDF scores are provided to account for cluster size in relation to these term counts. Click **here** [61] to view the interactive plot for the top 30 terms per cluster. Use the legend on the right to toggle between clusters, and hover over the bars to see the normalized frequency of each term. The plot displays bars for the top 30 terms based on their TF-IDF scores, and the tooltips provide the normalized frequency for each term.

Table 7: Sum of terms & unique terms for each cluster

	Total Terms	Unique Terms
Cluster 0	34697	215
Cluster 1	56118	223
Cluster 2	20263	203
Cluster 3	31085	203
Cluster 4	11289	177
Cluster 5	1927	102

Certain terms are prevalent across all clusters. Notably, *participants*, *experiment*, and *sample* consistently rank among the top terms in every cluster. In addition to this, a heatmap was generated to illustrate the overlap in keywords between clusters, using their respective Jaccard similarity scores. Figure 20 displays this heatmap.

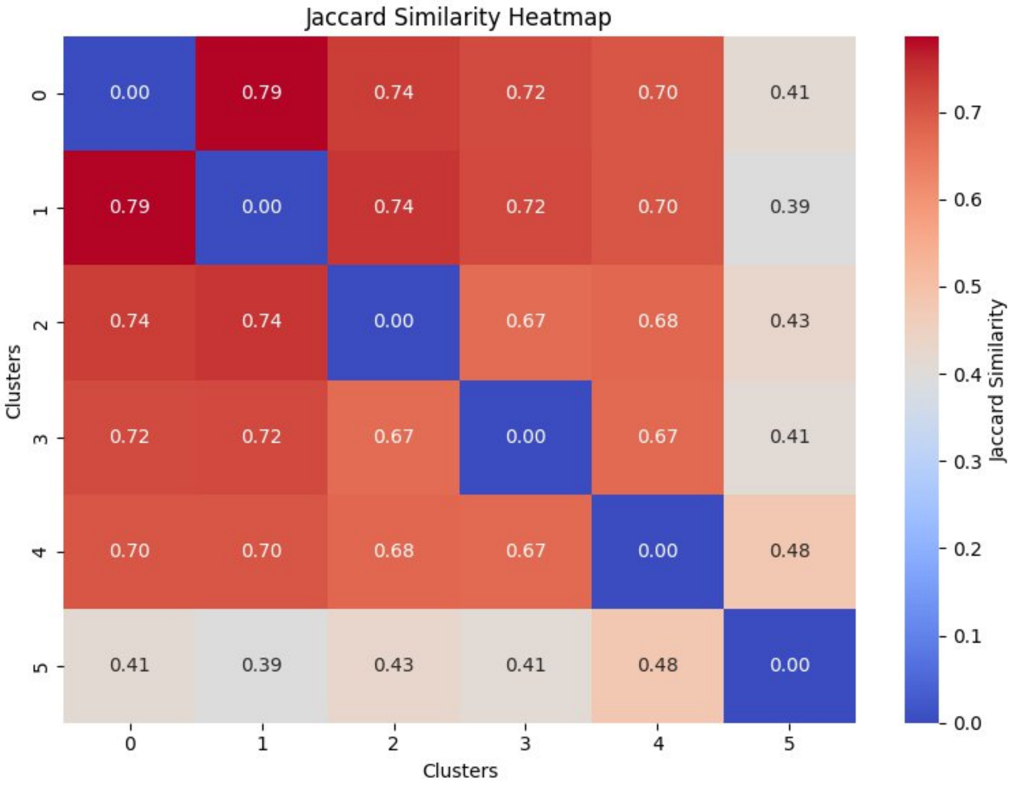


Figure 20: Jaccard similarity scores between clusters

All clusters, except cluster 5, show strong similarity, with the highest score (0.79) observed between cluster 1 and cluster 0.

5.1.2 Semi-supervised Clustering

The UMAP configuration that yielded the best clustering results, based on the Silhouette score, had the following parameter settings: 30 n_neighbors, 0.5 min_dist and 10 n_components. The resulting Silhouette score was 0.4598. Figure 21 depicts the respective t-sne visualization.

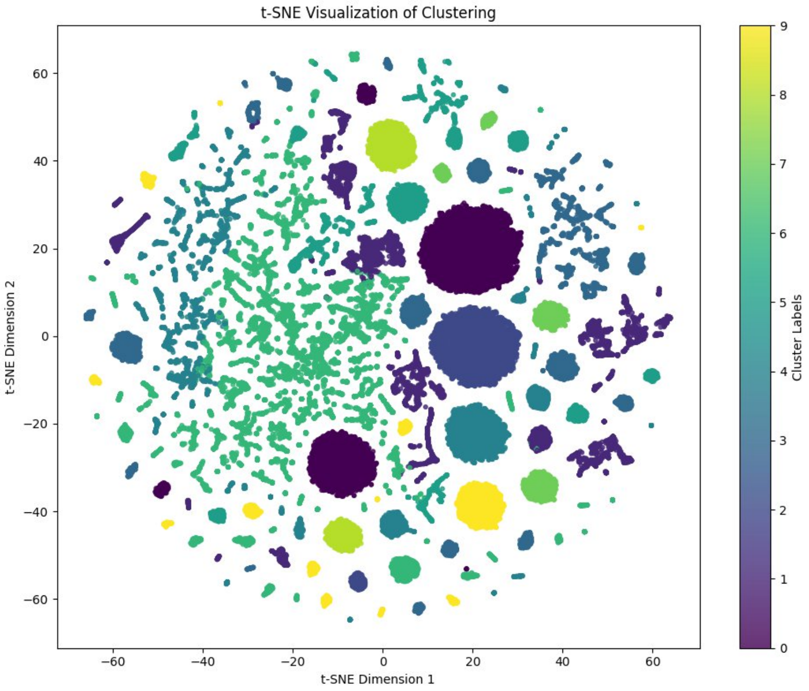


Figure 21: t-sne Visualization of Semi-supervised Clustering

The number of clusters that were identified was 10 and they varied in size. The number of records in each cluster is presented in Table 8.

Table 8: No. of records per cluster

	Records
Cluster 0	7504
Cluster 1	7714
Cluster 2	3960
Cluster 3	7129
Cluster 4	9249
Cluster 5	5709
Cluster 6	18045
Cluster 7	1987
Cluster 8	2338
Cluster 9	2973

The proportional composition of the clusters relative to the overall dataset is illustrated in Figure 22, where the size of each slice corresponds to the proportion of records in each cluster.

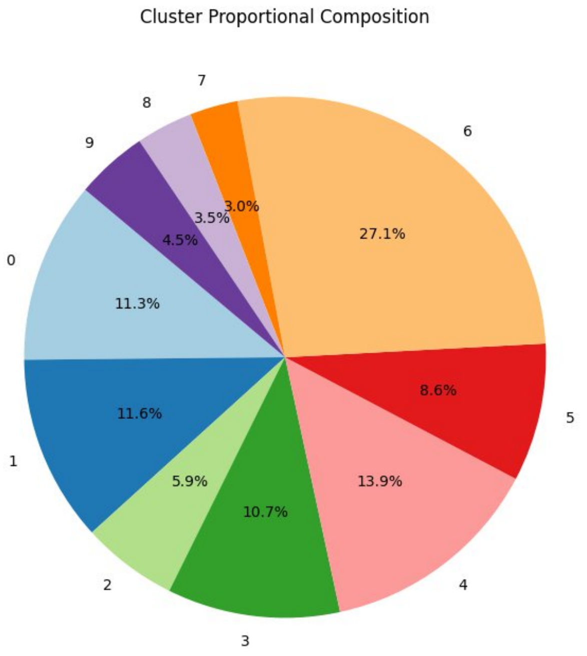


Figure 22: Proportional Cluster Composition Semi-supervised

Six out of 10 clusters, namely from 0 to 5, are comparable in size. Moreover, almost one third of the abstracts are placed in cluster 6. The respective Silhouette coefficients, mean and SD, for each cluster can be found in Figure 23.

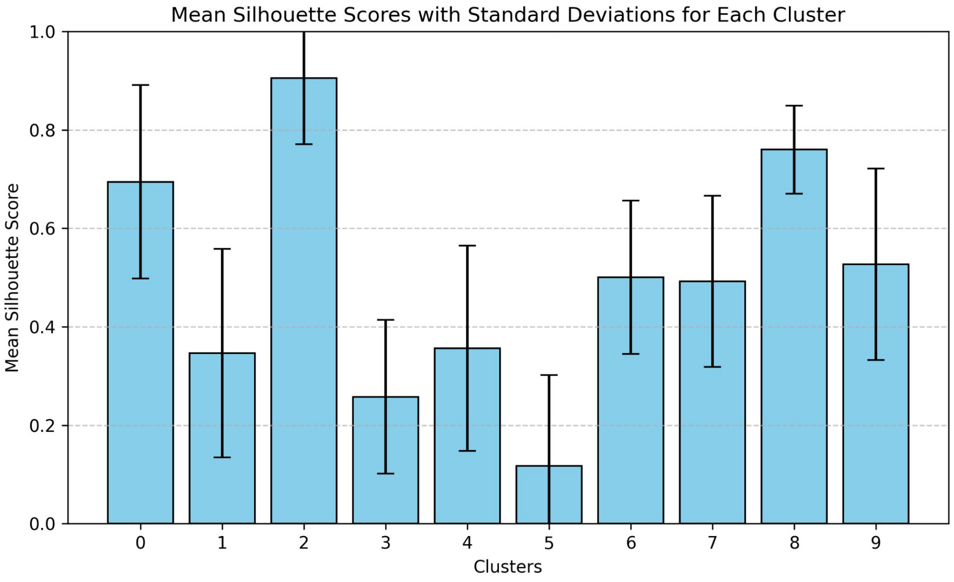


Figure 23: Mean Silhouette Scores with SD per Cluster

Only cluster 8 shows a sufficient Silhouette score, with a small SD.

To illustrate the composition of terms within the clusters, Table 9 presents the total number of terms in each cluster, along with the unique terms identified in each one. Additionally, the TF-IDF scores are provided to account for cluster size in relation to these term counts. Click **here** [62] to view the interactive plot for the top 30 terms per cluster. Use the legend on the right to toggle between clusters, and hover over the bars to see the normalized frequency of each term. The plot displays bars for the top 30 terms based on their TF-IDF scores, and the tooltips provide the normalized frequency for each term.

Table 9: Sum of terms & unique terms for each cluster

	Total Terms	**Unique Terms**
Cluster 0	8205	15
Cluster 1	30022	208
Cluster 2	4302	12
Cluster 3	16852	149
Cluster 4	24580	157
Cluster 5	11063	132
Cluster 6	50827	205
Cluster 7	3048	18
Cluster 8	2372	9
Cluster 9	3921	22

In addition to this, a heatmap was generated to illustrate the overlap in keywords between clusters, using their respective Jaccard similarity scores. Figure 24 displays this heatmap.

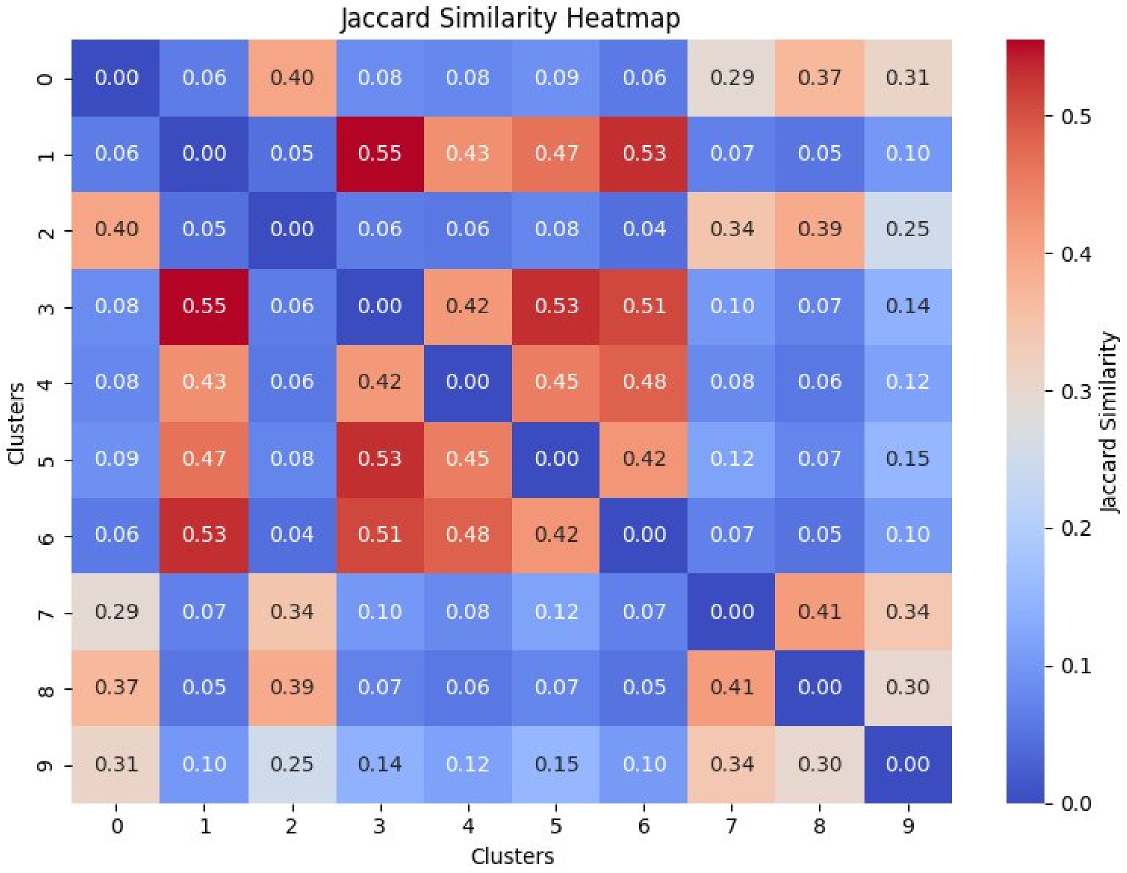


Figure 24: Jaccard similarity scores between clusters

5.2 Frequency & Trend Analyses

Out of 85,452 abstracts, 66,792 (78.16%) contained extracted terms. On average, each abstract had 1.82 terms, with a median of 1 term per abstract and an SD of 1.62. Figure 25 presents the top 10 detected terms.

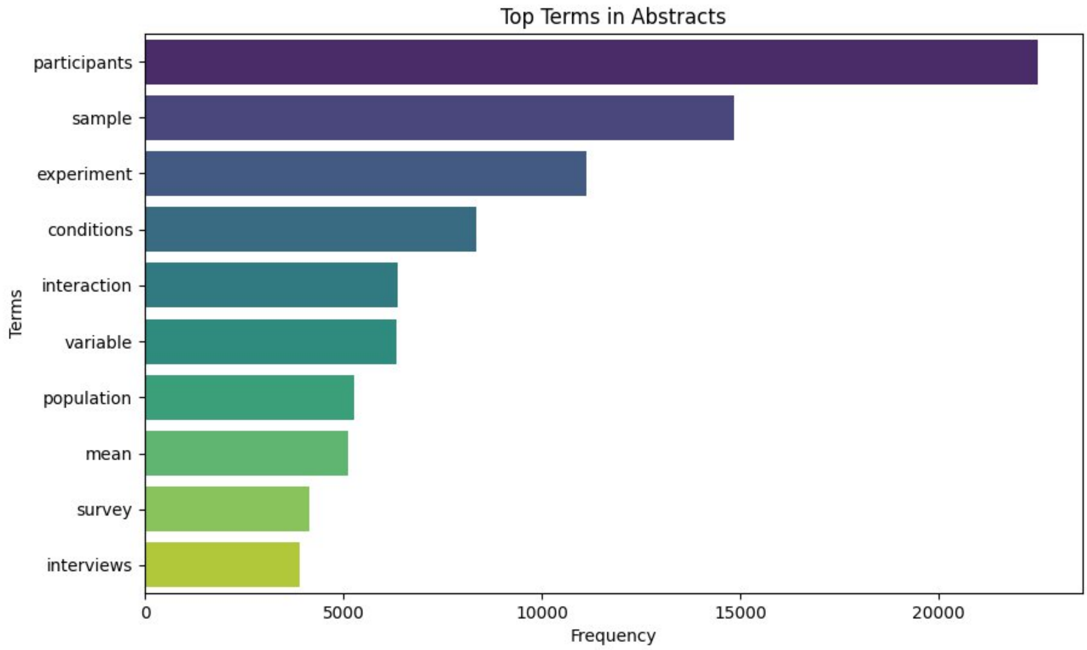


Figure 25: Top 10 Terms in abstracts

For a weighted term presentation, click **here** [63] to view the interactive plot for the top 30 terms per year. Use the legend on the right to toggle between years, and hover over the bars to see the TF-IDF scores for each term normalized by the total number of abstracts per year.

By the same token, click **here** [64] to get the interactive plot with the normalized term co-occurrence frequencies per year. Use the legend on the right to toggle between years, and hover over the bars to see the normalized frequencies. *No Terms* indicate the number of abstracts with no extracted terms for each particular year. Finally, in Figure 26 the proportion of abstracts with no terms over the years is presented.

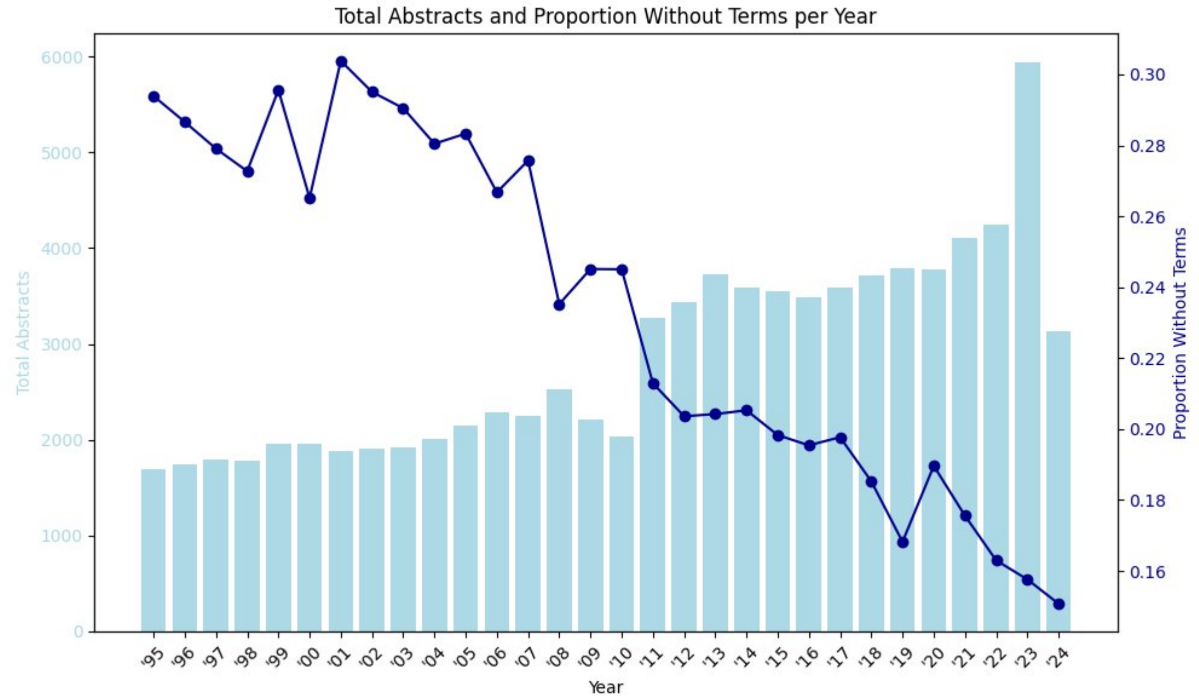


Figure 26: Proportion of abstracts with no terms

A downward trend is observed over the years, indicating that the number of abstracts without method-related terms has been dropping.

6 Discussion

This study examined methodological terminology in psychology research by analyzing a corpus of psychology abstracts. More specifically, we researched how terms are grouped together and whether there are patterns in the way psychological terminology is used.

Results from both unsupervised and semi-supervised clustering approaches did not yield clear inherent groupings. In the unsupervised approach, both K-Means and DBSCAN were tested to leverage their complementary characteristics, as K-Means assumes spherical clusters and requires the number of clusters to be predefined, while DBSCAN detects arbitrary-shaped clusters and handles noise explicitly. However, neither approach produced distinct clusters, as overlapping terms were observed across clusters. That was the case also for the weighted approach in which more generic terms had been assigned lower weights. This element was partially mitigated with the semi-supervised approach, yet by examining the respective TF-IDF scores of the terms found in each cluster, no clear thematic groupings could be derived. Moreover, although the produced Jaccard similarity heatmap implied no significant overlapping between certain clusters, the terms found in those clusters were not differentiating to the extent that a clear theme could be drawn. Nevertheless, an elucidating finding was that the method-related research terms comprising the glossary tend to appear more over the years, as the number of abstracts with no terms is following a downward trend.

In the context of identifying topics and emerging trends from scientific abstracts, the findings of related work as presented in the relevant section are not directly comparable, as those studies dealt with positive reporting [21] and the niche of clinical psychology and psychotherapy [22]. Hence, the considerations that can be made pertain mostly to the methods employed by those studies and how these methods compare with the methods utilized here.

A major difference lies in how terms were retrieved from the abstracts. In those studies, more dynamic approaches were employed to extract key terms, as opposed to the predefined gold standard collection used here. While those studies also emphasize the contribution of domain experts in refining the keyword list, their approach begins with terms drawn directly from the dataset, which are then filtered by experts. Consequently, the machine learning methods applied in those studies benefited from a better-refined input.

Furthermore, method-related terminology might not be clearly reported in psychology abstracts. Although the reporting of method-related terms appears to increase over the years, those terms with high discriminative value, which would naturally place an abstract in a certain cluster might be lacking from abstracts. For example, a combination of terms like *repeated measures anova, attrition, dependent t-test, line graph, trend analysis, multiple time points* could point with a high degree of confidence to a longitudinal study. Nevertheless, this kind of informative pattern seems to be lacking from the examined abstracts.

Overall, the use of a predefined glossary may contribute to the poor clustering results. While the glossary consists of terms frequently used in psychology research to describe methodologies, it runs the risk of lacking context-specificity. Additionally, the application of SciBERT for vectorizing the terms does not appear to have provided the intended benefit. The rationale behind using SciBERT was to generate contextualized embeddings that could capture the nuanced meanings of terms, thereby negating the need for topic modeling. This is also why the average of the term embeddings across different abstracts was calculated. However, the clustering results did not support this approach. This suggests that, although SciBERT is optimized for scientific text, psychological terminology may require more domain-specific embedding techniques. Thereupon, a lighter vectorizer, such as Word2Vec, might have been more effective.

Despite the mentioned limitations, the current study has some strong elements, offering some extra value to the field. More specifically, it offers an updated view regarding the use of certain method-related terms in psychology abstracts. The lack of terms with discriminative power from psychology abstracts could affect how psychological research is translated into practice and guide the way of future reporting by leading to clearer ways of mapping psychological concepts. These findings could have practical implications for the development of more standardized abstract writing guidelines and the enhancement of methodological transparency in psychological research reporting. Future research might benefit from exploring alternative clustering approaches specifically designed for psychological terminology, or from developing more sophisticated term discrimination techniques that account for the interconnected nature of psychological concepts.

7 Conclusion & Future Work

7.1 Conclusion

This study provides valuable insights into the temporal trends in reporting method-related keywords in psychology research abstracts, highlighting both the growing presence of such terms over time and the challenges in their clear categorization.

A key takeaway is that using a predefined glossary of method-related terms as a gold standard for keyword retrieval proved challenging, as evidenced by the clustering results. Neither the unsupervised nor semi-supervised models produced distinct clusters, with terms overlapping between them. Additionally, the use of SciBERT for term vectorization may not be suitable for fixed-term applications like those employed in this study.

Overall, while the clustering approaches did not yield distinct groupings, these findings highlight the need for more standardized reporting of methodological details in abstracts and suggest opportunities for improving how psychological research methods are documented and classified.

7.2 Future Work

Based on the findings of our study, several suggestions can be made to enhance future research in this area.

First, the glossary used could be refined through contextualized filtering, dynamically adjusting to the dataset at hand. This could be followed by a manual labeling approach based on the refined extracted terms, allowing for the training of a corresponding machine learning model to yield more targeted results.

Additionally, given the high number of overlapping terms observed with K-Means and DBSCAN, a hierarchical clustering algorithm might offer better differentiation. A two-level approach could be employed, where first-level clustering groups similar abstracts based on broader patterns, and second-level clustering within each group captures finer distinctions.

Lastly, clustering could be based on significant co-occurrences rather than single-term vector representations. A term co-occurrence graph, where nodes represent terms and edges represent their co-occurrence in abstracts, could be constructed. Graph-based clustering algorithms could then be applied to identify method groupings based on structural patterns in the network.

References

[1] Z. Nasar, S. W. Jaffry, and M. K. Malik, “Information extraction from scientific articles: a survey,” *Scientometrics*, vol. 117, no. 3, pp. 1931–1990, Dec. 2018, doi: 10.1007/s11192-018-2921-5.

[2] Z. Hong, L. Ward, K. Chard, B. Blaiszik, and I. Foster, “Challenges and Advances in Information Extraction from Scientific Literature: a Review,” *JOM*, vol. 73, no. 11, pp. 3383–3400, Nov. 2021, doi: 10.1007/s11837-021-04902-9.

[3] American psychological association, Ed., *Publication manual of the American psychological association: the official guide to APA style*, 7th ed. Washington (D.C.): American psychological association, 2020.

[4] D. J. Bem, “Feeling the future: Experimental evidence for anomalous retroactive influences on cognition and affect.,” *J. Pers. Soc. Psychol.*, vol. 100, no. 3, pp. 407–425, 2011, doi: 10.1037/a0021524.

[5] J. P. A. Ioannidis, “Why Most Published Research Findings Are False,” *PLoS Med.*, vol. 2, no. 8, p. e124, Aug. 2005, doi: 10.1371/journal.pmed.0020124.

[6] Open Science Collaboration, “Estimating the reproducibility of psychological science,” *Science*, vol. 349, no. 6251, p. aac4716, Aug. 2015, doi: 10.1126/science.aac4716.

[7] R. A. Klein *et al.*, “Investigating Variation in Replicability: A ‘Many Labs’ Replication Project,” *Soc. Psychol.*, vol. 45, no. 3, pp. 142–152, May 2014, doi: 10.1027/1864-9335/a000178.

[8] Z. Kekecs *et al.*, “Raising the value of research studies in psychological science by increasing the credibility of research reports: the transparent Psi project,” *R. Soc. Open Sci.*, vol. 10, no. 2, p. 191375, Feb. 2023, doi: 10.1098/rsos.191375.

[9] J. P. A. Ioannidis, “What Have We (Not) Learnt from Millions of Scientific Papers with *P* Values?,” *Am. Stat.*, vol. 73, no. sup1, pp. 20–25, Mar. 2019, doi: 10.1080/00031305.2018.1447512.

[10] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: state of the art, current trends and challenges,” *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[11] L. Zhao, W. Alhoshan, A. Ferrari, and K. J. Letsholo, “Classification of Natural Language

Processing Techniques for Requirements Engineering," 2022, *arXiv*. doi: 10.48550/ARXIV.2204.04282.

[12] A. Jabbar, S. Iqbal, M. I. Tamimy, A. Rehman, S. A. Bahaj, and T. Saba, "An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems," *IEEE Access*, vol. 11, pp. 133681–133702, 2023, doi: 10.1109/ACCESS.2023.3332710.

[13] R. Pramana, Debora, J. J. Subroto, A. A. S. Gunawan, and Anderies, "Systematic Literature Review of Stemming and Lemmatization Performance for Sentence Similarity," in *2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)*, Yogyakarta, Indonesia: IEEE, Nov. 2022, pp. 1–6. doi: 10.1109/ICITDA55840.2022.9971451.

[14] R. V. Siva Balan, K. Walia, and K. Gupta, "A Systematic Review on POS Tagging," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India: IEEE, Apr. 2022, pp. 1531–1536. doi: 10.1109/ICACITE53722.2022.9823658.

[15] M. Nesca, A. Katz, C. Leung, and L. Lix, "A scoping review of preprocessing methods for unstructured text data to assess data quality," *Int. J. Popul. Data Sci.*, vol. 7, no. 1, Oct. 2022, doi: 10.23889/ijpds.v7i1.1757.

[16] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers," *Inf. Syst.*, vol. 121, p. 102342, Mar. 2024, doi: 10.1016/j.is.2023.102342.

[17] S. J. Mielke *et al.*, "Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP," Dec. 20, 2021, *arXiv*: arXiv:2112.10508. Accessed: Nov. 12, 2024. [Online]. Available: http://arxiv.org/abs/2112.10508

[18] L. Xue *et al.*, "ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models," *Trans. Assoc. Comput. Linguist.*, vol. 10, pp. 291–306, Mar. 2022, doi: 10.1162/tacl_a_00461.

[19] D. J. Ladani and N. P. Desai, "Stopword Identification and Removal Techniques on TC and IR applications: A Survey," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India: IEEE, Mar. 2020, pp. 466–472. doi: 10.1109/ICACCS48705.2020.9074166.

[20] R. R. Dhala, A. V. S. P. Kumar, and S. P. Panda, "A Comparative Study on Keyword Extraction and Generation of Synonyms in Natural Language Processing," in *2023*

International Conference in Advances in Power, Signal, and Information Technology (APSIT), Bhubaneswar, India: IEEE, Jun. 2023, pp. 330–337. doi: 10.1109/APSIT58554.2023.10201796.

[21] L. Schiekiera, J. Diederichs, and H. Niemeyer, “Classifying Positive Results in Clinical Psychology Using Natural Language Processing,” *Z. Für Psychol.*, vol. 232, no. 3, pp. 147–159, Jul. 2024, doi: 10.1027/2151-2604/a000563.

[22] A. Sokolova, P. Lobanova, and I. Kuzminov, “Identifying emerging trends and hot topics through intelligent data mining: the case of clinical psychology and psychotherapy,” *foresight*, vol. 26, no. 1, pp. 155–180, Jan. 2024, doi: 10.1108/FS-02-2023-0026.

[23] National Library of Medicine, “Unified Medical Language System (UMLS).” U.S. National Library of Medicine, 2024. Accessed: Nov. 15, 2024. [Online]. Available: https://www.nlm.nih.gov/research/umls

[24] M. Y. Landolsi, L. Hlaoua, and L. Ben Romdhane, “Information extraction from electronic medical documents: state of the art and future research directions,” *Knowl. Inf. Syst.*, vol. 65, no. 2, pp. 463–516, Feb. 2023, doi: 10.1007/s10115-022-01779-1.

[25] S. Yang, J. Pancras, and Y. (Amy) Song, “Broad or exact? Search Ad matching decisions with keyword specificity and position,” *Decis. Support Syst.*, vol. 143, p. 113491, Apr. 2021, doi: 10.1016/j.dss.2021.113491.

[26] N. Thuon, W. Zhang, and S. Thuon, “KSW: Khmer Stop Word based Dictionary for Keyword Extraction,” 2024, *arXiv*. doi: 10.48550/ARXIV.2405.17390.

[27] A. P. Quimbaya *et al.*, “Named Entity Recognition Over Electronic Health Records Through a Combined Dictionary-based Approach,” *Procedia Comput. Sci.*, vol. 100, pp. 55–61, 2016, doi: 10.1016/j.procs.2016.09.123.

[28] Q. Qiu, Z. Xie, L. Wu, and L. Tao, “Dictionary-Based Automated Information Extraction From Geological Documents Using a Deep Learning Algorithm,” *Earth Space Sci.*, vol. 7, no. 3, p. e2019EA000993, Mar. 2020, doi: 10.1029/2019EA000993.

[29] C. Riveros, N. Van Sint Jan, and D. Vrgoč, “REmatch: A Novel Regex Engine for Finding All Matches,” *Proc. VLDB Endow.*, vol. 16, no. 11, pp. 2792–2804, Jul. 2023, doi: 10.14778/3611479.3611488.

[30] N. Chida and T. Terauchi, “Repairing Regular Expressions for Extraction,” *Proc. ACM Program. Lang.*, vol. 7, no. PLDI, pp. 1633–1656, Jun. 2023, doi: 10.1145/3591287.

[31] Y.-W. Lai and M.-Y. Chen, “Review of Survey Research in Fuzzy Approach for Text Mining,” *IEEE Access*, vol. 11, pp. 39635–39649, 2023, doi: 10.1109/ACCESS.2023.3268165.

[32] Clarivate Analytics, “Web of Science Citation Report For Fuzzy and Text Mining Co-occurrences,” Nov. 2024. [Online]. Available: https://www.webofscience.com/wos/woscc/citation-report/015495cd-3b7c-46e5-b912-65ca5241bc54-64afb73e

[33] L. Ajallouda, F. Z. Fagroud, A. Zellou, and E. H. Benlahmar, “A Systematic Literature Review of Keyphrases Extraction Approaches,” *Int. J. Interact. Mob. Technol. IJIM*, vol. 16, no. 16, pp. 31–58, Aug. 2022, doi: 10.3991/ijim.v16i16.33081.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” 2013, *arXiv*. doi: 10.48550/ARXIV.1301.3781.

[35] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[36] M. Nadim, D. Akopian, and A. Matamoros, “A Comparative Assessment of Unsupervised Keyword Extraction Tools,” *IEEE Access*, vol. 11, pp. 144778–144798, 2023, doi: 10.1109/ACCESS.2023.3344032.

[37] T. Nomoto, “Keyword Extraction: A Modern Perspective,” *SN Comput. Sci.*, vol. 4, no. 1, p. 92, Dec. 2022, doi: 10.1007/s42979-022-01481-7.

[38] M. Song, Y. Feng, and L. Jing, “A Survey on Recent Advances in Keyphrase Extraction from Pre-trained Language Models,” in *Findings of the Association for Computational Linguistics: EACL 2023*, Dubrovnik, Croatia: Association for Computational Linguistics, 2023, pp. 2153–2164. doi: 10.18653/v1/2023.findings-eacl.161.

[39] M. Kölbl, Y. Kyogoku, J. N. Philipp, M. Richter, C. Rietdorf, and T. Yousef, “Beyond the Failure of Direct-Matching in Keyword Evaluation: A Sketch of a Graph Based Solution,” *Front. Artif. Intell.*, vol. 5, p. 801564, Mar. 2022, doi: 10.3389/frai.2022.801564.

[40] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad, “A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models,” Oct. 28, 2020, *arXiv*: arXiv:2010.15036. Accessed: Nov. 21, 2024. [Online]. Available: http://arxiv.org/abs/2010.15036

[41] W. A. Qader, M. M. Ameen, and B. I. Ahmed, “An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges,” in *2019 International Engineering Conference (IEC)*, Erbil, Iraq: IEEE, Jun. 2019, pp. 200–204. doi: 10.1109/IEC47844.2019.8950616.

[42] S. J. Johnson, M. R. Murty, and I. Navakanth, “A detailed review on word embedding techniques with emphasis on word2vec,” *Multimed. Tools Appl.*, vol. 83, no. 13, pp. 37979–38007, Oct. 2023, doi: 10.1007/s11042-023-17007-z.

[43] J. Noh and R. Kavuluru, “Improved biomedical word embeddings in the transformer era,” *J. Biomed. Inform.*, vol. 120, p. 103867, Aug. 2021, doi: 10.1016/j.jbi.2021.103867.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” May 24, 2019, *arXiv*: arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.

[45] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” Jul. 26, 2019, *arXiv*: arXiv:1907.11692. doi: 10.48550/arXiv.1907.11692.

[46] M. E. Peters *et al.*, “Deep contextualized word representations,” Mar. 22, 2018, *arXiv*: arXiv:1802.05365. doi: 10.48550/arXiv.1802.05365.

[47] J. Howard and S. Ruder, “Universal Language Model Fine-tuning for Text Classification,” May 23, 2018, *arXiv*: arXiv:1801.06146. doi: 10.48550/arXiv.1801.06146.

[48] A. E. Ezugwu *et al.*, “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects,” *Eng. Appl. Artif. Intell.*, vol. 110, p. 104743, Apr. 2022, doi: 10.1016/j.engappai.2022.104743.

[49] H. Yin, A. Aryani, S. Petrie, A. Nambissan, A. Astudillo, and S. Cao, “A Rapid Review of Clustering Algorithms,” Jan. 14, 2024, *arXiv*: arXiv:2401.07389. doi: 10.48550/arXiv.2401.07389.

[50] J. Singh and D. Singh, “A comprehensive review of clustering techniques in artificial intelligence for knowledge discovery: Taxonomy, challenges, applications and future prospects,” *Adv. Eng. Inform.*, vol. 62, p. 102799, Oct. 2024, doi: 10.1016/j.aei.2024.102799.

[51] *Computational learning approaches to data analytics in biomedical applications*. London: Academic press, 2020.

[52] J. C. De Winter and D. Dodou, “A surge of p -values between 0.041 and 0.049 in recent decades (but negative results are increasing rapidly too),” *PeerJ*, vol. 3, p. e733, Jan. 2015, doi: 10.7717/peerj.733.

[53] L. Gokhberg, I. Kuzminov, E. Khabirova, and T. Thurner, “Advanced text-mining for trend analysis of Russia’s Extractive Industries,” *Futures*, vol. 115, p. 102476, Jan. 2020, doi: 10.1016/j.futures.2019.102476.

[54] M.-H. Weng, S. Wu, and M. Dyer, “Identification and Visualization of Key Topics in

Scientific Publications with Transformer-Based Language Models and Document Clustering Methods," *Appl. Sci.*, vol. 12, no. 21, p. 11220, Nov. 2022, doi: 10.3390/app122111220.

[55] H. Park and M. S. Park, "Capturing the trend of mHealth research using text mining," *mHealth*, vol. 5, pp. 48–48, Oct. 2019, doi: 10.21037/mhealth.2019.09.06.

[56] "Analysis of Research Trends in Artificial Intelligence and Healthcare Convergence using Text Mining Techniques," in *Proceedings of the International Conference on Industrial Engineering and Operations Management*, Sydney, Australia: IEOM Society International, Dec. 2022, pp. 1657–1665. doi: 10.46254/AU01.20220369.

[57] K. Stathakis, "Dissertation Repository." GitHub. Accessed: Jan. 16, 2025. [Online]. Available: https://github.com/KosStath/Dissertation/Appendix

[58] "Glossary," in *Research Methods in Psychology, 4th Edition*, Kwantlen Polytechnic University. Accessed: Dec. 01, 2025. [Online]. Available: https://kpu.pressbooks.pub/psychmethods4e/back-matter/glossary/

[59] "Glossary of Terms," in *Introduction to Statistics for Psychology*, Maricopa Open Press. Accessed: Dec. 01, 2025. [Online]. Available: https://open.maricopa.edu/psy230mm/chapter/glossary-of-terms/

[60] "Key Terms for Psychological Research," in *Introduction to Psychology & Neuroscience*, Dalhousie University. Accessed: Dec. 01, 2025. [Online]. Available: https://digitaleditions.library.dal.ca/intropsychneuro/chapter/key-terms-for-psychological-research/

[61] K. Stathakis, "Interactive Bar Plot - Unsupervised Clusters." https://github.com/KosStath/Dissertation. Accessed: Jan. 16, 2025. [Online]. Available: https://kosstath.github.io/Dissertation/Interactive_BarPlot_TopTerms_Clusters.html

[62] K. Stathakis, "Interactive Bar Plot - Semi-Supervised Clusters." https://github.com/KosStath/Dissertation. Accessed: Jan. 16, 2025. [Online]. Available: https://kosstath.github.io/Dissertation/Interactive_Top_Terms_BarPlot_SemiSupervised.html

[63] K. Stathakis, "Interactive Bar Plot - Overall Top Terms." https://github.com/KosStath/Dissertation. Accessed: Jan. 16, 2025. [Online]. Available: https://kosstath.github.io/Dissertation/Interactive_BarPlot_OverallTopTerms_byYear.html

[64] K. Stathakis, "Interactive Bar Plot - Cooccurrence Frequencies." https://github.com/KosStath/Dissertation. Accessed: Jan. 16, 2025. [Online]. Available:

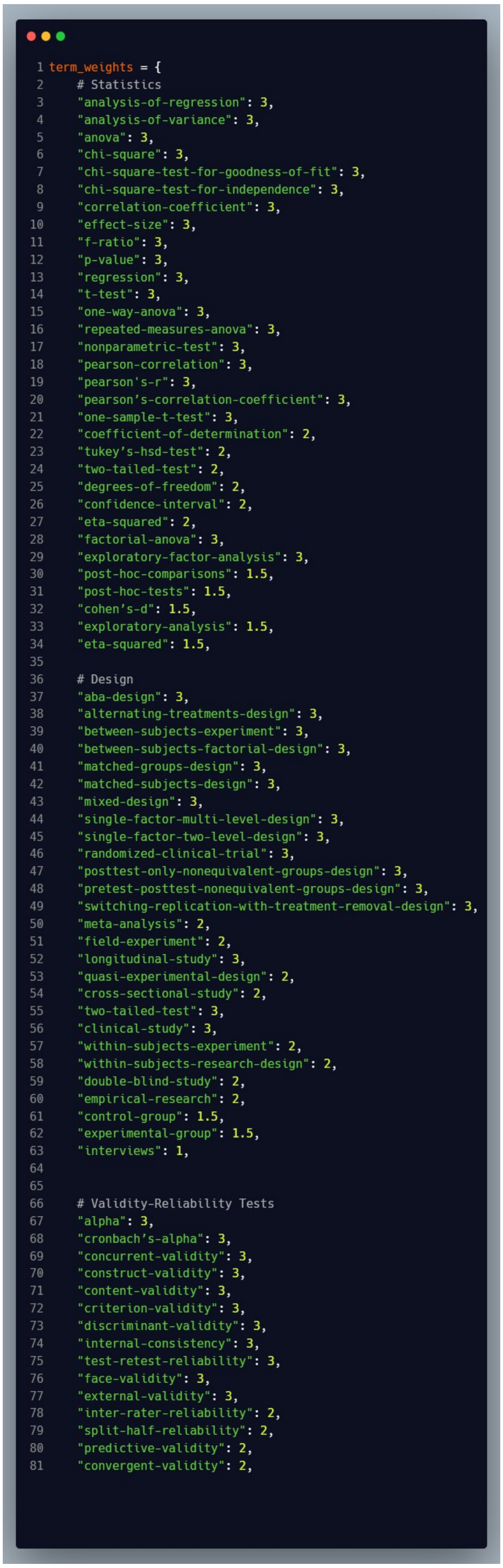
https://kosstath.github.io/Dissertation/Interactive_BarPlot_Co-occurrenceFreqs_Top30_byYear.html

Appendix

Table 1: Acronym Definitions

Acronym	Technique	Raw	Processed
DON	Do Nothing	“Like a Rolling Stone”	“Like a Rolling Stone”
RNS	Replace Noise	“@Obama tells #metoo! bit.ly/—”	“USER tells HASHTAG! URL”
RSA	Replace Slang/Abbreviations	“omg you are so nice!”	“Oh my God you are so nice!”
RCT	Replace Contraction	“I don’t like butterflies.”	“I do not like butterflies.”
RRP	Remove Repeated Punctuation	“I like her!!!”	“I like her multiExclamation”
RPT	Removing Punctuation	“You. are. cool.”	“You are cool”
RNB	Remove Numbers	“You are gr8.”	“You are gr.”
LOW	Lowercasing	“You Rock! YEAH!”	“you rock! yeah!”
RSW	Remove Stop Words	“This is nice”	“is nice”
SCO	Spelling Correction	“TIlenia is so kind!”	“lenia is so kind!”
POS	Part-of-Speech Tagging	“Kim likes you”	“Kim (PN) likes (VB) you (N)”
LEM	Lemmatization	“I am going to shopping”	“I be go to shop”
STM	Stemming	“Girl's shirt with different colors”	“Girl shirt with differ color”
ECR	Remove Elongation	“You are cooool!”	“You are cool!”
EMO	Emoticon Handling	“)”	“happy”

NEG	Negation Handling	“I am not happy today!”	“I am sad today!”
WSG	Word Segmentation	“#sometrendingtopic”	“some+trending+topic ”



```python
term_weights = {
    # Statistics
    "analysis-of-regression": 3,
    "analysis-of-variance": 3,
    "anova": 3,
    "chi-square": 3,
    "chi-square-test-for-goodness-of-fit": 3,
    "chi-square-test-for-independence": 3,
    "correlation-coefficient": 3,
    "effect-size": 3,
    "f-ratio": 3,
    "p-value": 3,
    "regression": 3,
    "t-test": 3,
    "one-way-anova": 3,
    "repeated-measures-anova": 3,
    "nonparametric-test": 3,
    "pearson-correlation": 3,
    "pearson's-r": 3,
    "pearson’s-correlation-coefficient": 3,
    "one-sample-t-test": 3,
    "coefficient-of-determination": 2,
    "tukey’s-hsd-test": 2,
    "two-tailed-test": 2,
    "degrees-of-freedom": 2,
    "confidence-interval": 2,
    "eta-squared": 2,
    "factorial-anova": 3,
    "exploratory-factor-analysis": 3,
    "post-hoc-comparisons": 1.5,
    "post-hoc-tests": 1.5,
    "cohen’s-d": 1.5,
    "exploratory-analysis": 1.5,
    "eta-squared": 1.5,

    # Design
    "aba-design": 3,
    "alternating-treatments-design": 3,
    "between-subjects-experiment": 3,
    "between-subjects-factorial-design": 3,
    "matched-groups-design": 3,
    "matched-subjects-design": 3,
    "mixed-design": 3,
    "single-factor-multi-level-design": 3,
    "single-factor-two-level-design": 3,
    "randomized-clinical-trial": 3,
    "posttest-only-nonequivalent-groups-design": 3,
    "pretest-posttest-nonequivalent-groups-design": 3,
    "switching-replication-with-treatment-removal-design": 3,
    "meta-analysis": 2,
    "field-experiment": 2,
    "longitudinal-study": 3,
    "quasi-experimental-design": 2,
    "cross-sectional-study": 2,
    "two-tailed-test": 3,
    "clinical-study": 3,
    "within-subjects-experiment": 2,
    "within-subjects-research-design": 2,
    "double-blind-study": 2,
    "empirical-research": 2,
    "control-group": 1.5,
    "experimental-group": 1.5,
    "interviews": 1,


    # Validity-Reliability Tests
    "alpha": 3,
    "cronbach’s-alpha": 3,
    "concurrent-validity": 3,
    "construct-validity": 3,
    "content-validity": 3,
    "criterion-validity": 3,
    "discriminant-validity": 3,
    "internal-consistency": 3,
    "test-retest-reliability": 3,
    "face-validity": 3,
    "external-validity": 3,
    "inter-rater-reliability": 2,
    "split-half-reliability": 2,
    "predictive-validity": 2,
    "convergent-validity": 2,
```

Figure 14: Weighting scheme for the emphasized terms