

HYBRID CLUSTERING FRAMEWORK FOR LARGE-SCALE SCIENTIFIC LITERATURE STRUCTURING

Dmytro TEPLOV , Artur RADZIVIL  , Andrej BUGAJEV 

Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Vilnius, Lithuania 

Article History:

- received 4 March 2026
- accepted 18 June 2026

Abstract. This study addresses the challenge of organizing and interpreting large collections of scientific literature, focusing on a rapidly growing and heterogeneous AI research domain derived from object-detection-with-small-data literature. First, based on the set of keywords using Clarivate Analytics queries, over 18,500 research articles were retrieved, illustrating the problem of manual review infeasibility, leading to machine learning-based approaches. To begin literature filtration, a pipeline combining text preprocessing, TF-IDF feature extraction, and K-means++ clustering was implemented. Clusters were labelled using YAKE keyword extraction and thematically summarized with large language models (ChatGPT-4o and ChatGPT-o1), enabling the interpretation of research domains such as medical imaging, anomaly detection, etc. To compare clustering strategies, we employed neural autoencoder-based embeddings and deep embedded clustering architectures. The autoencoder latent space yielded both convergences and divergences relative to TF-IDF clusters, reflecting higher-level semantic structuring. Direct end-to-end DEC optimization produced unstable cluster assignments, thus we applied a three-phase IDEC training protocol using autoencoder pretraining, centroid initialization with K-means, and joint fine-tuning – this produced substantially more balanced and internally coherent clusters. Results demonstrate that integrating keyword-based and neural methods, supplemented by conversational AI summarization, offers a scalable and interpretable framework for structuring large scientific datasets.

This work highlights both methodological strengths and thematic overlaps across AI research, emphasizing the role of hybrid clustering approaches in managing the growing volume of scholarly publications.

Keywords: automated literature sorting, TF-IDF feature extraction, K-means++ clustering, large language models, autoencoder embeddings, deep embedded clustering, scientific text mining.

✉Corresponding author. E-mail: artur.radzivil@vilniustech.lt

1. Introduction

The accelerating pace of scientific discovery has led to a growth in scholarly publications, particularly in rapidly evolving fields such as computer science and artificial intelligence (AI). This abundance of information creates a significant challenge for researchers attempting to conduct systematic literature reviews, a structured method for reliably gathering research evidence through explicit and transparent procedures, uncovering research gaps, and tracking emerging trends (van Dinter et al., 2021). In specialized domains like computer vision, the enormous number of new articles makes manual review processes impractical and often nearly impossible. Reviewing thousands of publications requires significant time and effort, creating challenges for ongoing research tracking and maintaining an updated perspective on the field.

Traditional manual reviews, due to a human factor, are slow and limited in scale. Although some computational approaches are efficient, they often lack the semantic nuance required to interpret complex scientific texts (Kalibatiene & Miliauskaitė, 2026). This paper explores modern artificial intelligence, specifically the synergy between unsupervised machine learning and large language models (LLMs), as a way to automate and scale the process of scientific literature analysis. This work is focused on an emerging field of “AI for science”, where LLMs are increasingly applied to complex natural language processing (NLP) tasks in a scientific context, such as predicting citation counts and readership from abstracts (de Winter, 2024). The main research question in this study is:

How can a combination of advanced unsupervised clustering and conversational AI be applied to effectively map the topic landscape of a large and dynamic research domain?

To address this question, this paper presents a comparative study of different frameworks designed for automated literature analysis. Three distinct clustering pipelines are proposed and evaluated, each representing a different level of methodological depth:

- A lexical baseline method: A traditional approach using Term Frequency-Inverse Document Frequency (TF-IDF) for feature representation and *K*-Means++ for clustering. This method operates on keyword frequencies.
- A sequential semantic approach: An advanced method that utilizes pre-trained sentence embeddings to capture semantic meaning, then applies non-linear dimensionality reduction with a neural autoencoder before clustering.
- An integrated deep clustering approach: A state-of-the-art deep embedded clustering (DEC) model that jointly learns feature representations and cluster assignments in an end-to-end fashion.

A novel component of this framework is the use of conversational AI as a tool for thematic synthesis, specifically, different ChatGPT models. This final processing step converts the quantitative, machine-generated clusters into human-interpretable summaries, addressing the gap between raw data and knowledge.

The empirical evaluation uses a corpus assembled through a seed-plus-expansion keyword protocol rooted in the object detection with small data domain. The initial query returned approximately 4,000 articles. A frequency analysis of their keywords identified thirty high-frequency adjacent terms, and adding these terms expanded the corpus to nearly 18,500 articles covering topics such as medical image segmentation, fault diagnosis, few-shot learning, and graph neural networks. This breadth reflects the design of the expansion protocol: it identifies latent thematic neighbours of the seed domain and creates a heterogeneous test corpus for hybrid clustering methods rather than a closed single-domain collection.

This study makes several key contributions to the topic of AI-driven research analysis:

- A controlled comparison of three representation paradigms – lexical TF-IDF, sequential neural autoencoder, and joint deep embedded clustering – under a unified *K*-means partitioning scheme. Holding the clustering algorithm constant makes it possible to isolate the effect of representation choice on thematic structure.
- An empirical demonstration that the IDEC three-phase training protocol (autoencoder pretraining, *K*-means centroid initialisation, joint fine-tuning) mitigates cluster collapse in standard end-to-end DEC on a corpus of approximately 18,500 scientific abstracts.

- A qualitative comparison of GPT-4o and o1 as post-clustering interpretation models applied to the same unsupervised clustering outputs. This step uses LLMs for narrative synthesis of cluster themes rather than for screening or extraction.
- A reproducible keyword-driven corpus construction protocol focused on small-data object detection and expanded through documented keyword-frequency analysis, yielding a heterogeneous thematic map of nearly 18,500 abstracts.

The principal scientific contribution of this work is a controlled experimental framework that separates representation quality from the clustering mechanism in unsupervised scientific text analysis. The results show that IDEC stabilizes deep clustering on domain-specific abstract corpora, while LLM-based post-clustering synthesis improves the interpretability of machine-generated clusters. These findings provide reproducible methodological insights for AI-assisted systematic review.

2. Related work

The framework presented in this study spans multiple research areas: systematic literature review methodology, unsupervised text clustering, large language model applications, and the specific domain of object detection with small data. This section surveys the relevant prior work organized into four thematic groups, each contributing essential components to our approach while revealing gaps that motivate the present investigation.

2.1. Systematic literature reviews and their automation

Systematic literature review (SLR) has become the gold standard for evidence synthesis in multiple disciplines. The foundational methodologies established by Kitchenham (2007) for software engineering and Petticrew and Roberts (2008) for social sciences provide rigorous protocols for study identification, screening, quality assessment, and synthesis. The PRISMA 2020 statement (Page et al., 2021) updated reporting guidelines to ensure transparency and completeness in systematic reviews. These manual approaches, while thorough, face severe scalability challenges when confronted with exponentially growing scientific output. Kalibatiene and Miliauskaitė (2026) documents the transition from manual to automated systematic review processes, highlighting both opportunities and persistent methodological challenges.

The need for automation has driven substantial research into computational support for SLR workflows. O'Mara-Eves et al. (2015) conducted a systematic review of text mining approaches for study identification, finding that automation could reduce screening workload while maintaining acceptability. Marshall and Wallace (2019) provided a practical guide to applying machine learning tools in research synthesis, emphasizing active learning strategies where human reviewers iteratively train models on their screening decisions. The ASReview framework (van de Schoot et al., 2021) implements this active learning paradigm in an open-source platform that has demonstrated efficiency gains across multiple domains. A comprehensive survey by van Dinter et al. (2021) examined the state of SLR automation, identifying 41 primary studies and revealing that most automation efforts focus on the screening phase rather than synthesis.

Complementary to workflow automation, bibliometric tools offer data-driven approaches to mapping research landscapes. VOSviewer (van Eck & Waltman, 2010) enables visualization of co-citation and co-occurrence networks, revealing the intellectual structure of research fields. CiteSpace (Chen, 2006) detects emerging trends and transient patterns through temporal analysis of citation networks. While these tools excel at revealing structural relationships among publications, they typically produce network visualizations that require expert interpretation and do not generate textual thematic summaries.

A critical gap emerges from this literature: existing automation tools primarily address screening efficiency, reducing the human effort required to select relevant studies from large candidate sets. However, the synthesis phase: “organizing selected studies into coherent themes and narratives”, remains largely manual. The present study addresses this gap by automating not only clustering but also thematic interpretation through large language models.

2.2. Text clustering and representation learning

The effectiveness of any clustering-based analysis depends fundamentally on text representation. Classical approaches use term frequency–inverse document frequency (TF-IDF), grounded in the statistical interpretation of term specificity introduced Jones (1972) and formalized in information retrieval systems by Salton and McGill (1983). TF-IDF represents documents as sparse vectors in high-dimensional term spaces, capturing lexical overlap but not semantic similarity. Despite its simplicity, TF-IDF remains a competitive baseline for many document clustering tasks.

The K-means algorithm (Lloyd, 1982; MacQueen, 1967) partitions documents into clusters by iteratively refining cluster centroids to minimize within-cluster variance. Arthur and Vassilvitskii (2007) introduced *K*-means++, which improves initialization Vaswani through probabilistic seeding proportional to squared distance from existing centroids, yielding faster convergence and provably better approximations than random initialization. Determining the optimal number of clusters remains a central challenge, common approaches include the elbow method based on within-cluster sum of squares, the silhouette coefficient (Rousseeuw, 1987) measuring cluster cohesion relative to separation.

The advent of deep learning has fundamentally transformed text representation. Autoencoders learn compressed representations through reconstruction objectives; Hinton and Salakhutdinov (2006) demonstrated that deep autoencoders could discover lower-dimensional codes superior to principal component analysis. Vincent et al. (2010) introduced denoising autoencoders that learn robust features by reconstructing inputs from corrupted versions. These neural representations capture nonlinear patterns that linear methods like TF-IDF cannot model.

Transformer-based language models (Vaswani et al., 2017) have further advanced semantic text representation. BERT (Devlin et al., 2019) introduced bidirectional pre-training of deep transformers, learning contextualized word representations that can be fine-tuned for downstream tasks. Sentence-BERT (Reimers & Gurevych, 2019) adapts BERT using siamese network architectures to produce fixed-length sentence embeddings optimized for semantic similarity, enabling efficient clustering and retrieval. The MiniLM compression technique (Wang et al., 2020) applies deep self-attention distillation to reduce model size while retaining representa-

tional quality; its integration with the Sentence-Transformers library (e.g., all-MiniLM-L6-v2) provides a practical, lightweight embedding model suitable for processing large corpora such as the one analyzed in the present study.

Deep embedded clustering methods jointly optimize representation learning and cluster assignment. The Deep Embedded Clustering (DEC) algorithm (Xie et al., 2016) learns a feature space where cluster centers are simultaneously refined through a self-training objective that minimizes Kullback-Leibler divergence between soft cluster assignments and an auxiliary target distribution. Guo et al. (2017) extended this with Improved Deep Embedded Clustering (IDEC), adding a reconstruction loss to preserve local structure and prevent representation collapse. Surveys by Min et al. (2018) and Ren et al. (2025) systematically review deep clustering architectures, noting that joint optimization often outperforms two-stage approaches where representation learning and clustering are separated. For interpretability, keyword extraction algorithms like YAKE (Campos et al., 2020) identify salient terms using multiple local features including term frequency, casing, term position, and term relatedness to context.

An alternative family of methods for discovering thematic structure in document collections is probabilistic topic modeling. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) models documents as mixtures of latent topics, each defined by a distribution over words, and has become the standard baseline for unsupervised thematic analysis of scientific text. More recently, BERTopic (Grootendorst, 2022) combines transformer embeddings with HDBSCAN density-based clustering and class-based TF-IDF to produce interpretable topic representations. While topic models generate human-readable keyword lists per topic, they do not produce narrative thematic summaries. The present study adopts *K*-means-based clustering rather than topic models because the primary objective is to compare representation methods (lexical, sequential neural, and joint deep) under a controlled clustering framework, where *K*-means provides a consistent partitioning mechanism across all three pipelines.

Despite these advances, a fundamental limitation persists: clustering algorithms produce numeric partitions and keyword lists, but they do not generate human-readable thematic descriptions. Domain experts must manually inspect cluster contents to construct narrative summaries, partially negating the efficiency gains from automation. This interpretability gap motivates the integration of large language models in our framework.

We adopt *K*-means as a consistent partitioning mechanism across all three pipelines to enable controlled comparison of representation methods. BERTopic and LDA, while valuable for topic discovery, differ in their clustering mechanisms (HDBSCAN, probabilistic allocation), which would confound the comparison of embeddings. Future work should compare pipeline-level alternatives including BERTopic.

2.3. Large language models for scientific text analysis

The release of GPT-4 (OpenAI et al., 2024) marked a qualitative leap in large language model capabilities, demonstrating strong performance on reasoning, summarization, and domain-specific tasks without fine-tuning. The subsequent o1 model series (OpenAI et al., 2024) introduced reinforcement learning-trained chain-of-thought reasoning, in which the model generates internal reasoning tokens before producing its final response. This architectural distinction is relevant to the present study: GPT-4o prioritizes fluency and speed, while o1

invests additional computation in multi-step logical inference, potentially yielding more analytically structured thematic summaries.

Khraisha et al. (2024) evaluated GPT-4 across multiple systematic review stages, finding that performance varied substantially by task and prompt design: full-text screening with well-designed prompts approached human-level performance, while title-abstract screening on imbalanced datasets yielded only moderate agreement after adjusting for chance. Delgado-Chaves et al. (2025) tested 18 different LLMs for screening and demonstrated workload reductions of 33–93% with omission rates below 1%, comparable to human reviewers. Scherbakov et al. (2025) surveyed 172 studies on LLM applications in literature reviews, finding that GPT-based models dominated (73% of applications) and that most use cases concentrated on literature search and data extraction rather than thematic synthesis.

A critical concern in applying LLMs to scientific tasks is hallucination – the generation of plausible but factually incorrect content. Huang et al. (2025) provide a comprehensive taxonomy of hallucination types in large language models, identifying failure modes including factual inconsistencies, logical contradictions, and citation fabrication. For systematic review applications, where accuracy and verifiability are paramount, these risks necessitate human validation of LLM outputs. Evaluating the quality of LLM-generated summaries remains an open methodological challenge: automated metrics such as ROUGE and BERTScore measure lexical and semantic overlap with reference texts, but domain-specific factual correctness often requires expert assessment. The emerging PRISMA-trAlce checklist (Holst et al., 2025) addresses these concerns by extending PRISMA 2020 reporting guidelines specifically for AI-assisted systematic reviews, requiring documentation of model versions, prompt strategies, and validation procedures.

These studies reveal a consistent pattern: large language models offer substantial productivity gains in systematic review workflows, but most applications focus on the screening and extraction phases. The use of LLMs for post-clustering thematic synthesis like interpreting the output of unsupervised clustering algorithms and generating narrative summaries remains underexplored. The present study contributes to this emerging area by systematically comparing LLM-generated thematic summaries across multiple clustering pipelines and two model variants (GPT-4o and o1).

2.4. Object detection as a case study domain

Object detection serves as an ideal case study domain for evaluating automated literature review frameworks. The field spans over two decades of research, evolving from hand-crafted feature-based methods to the deep learning approaches that have dominated since AlexNet (Krizhevsky et al., 2012) demonstrated the effectiveness of convolutional neural networks for visual recognition. Zou et al. (2023) provide a comprehensive 20-year survey tracing this evolution from traditional methods through region-based, single-shot, and transformer-based detectors, documenting thousands of publications and continual architectural innovation. Liu et al. (2020) offer a similarly extensive survey of generic object detection, systematically categorizing detection frameworks, feature representations, and training strategies. Additional surveys by Jiao et al. (2019) and Zhao et al. (2019) further confirm the scale and velocity of research activity in this domain.

This rapid growth renders manual systematic review increasingly impractical. The sheer volume of publications exceeds the capacity of individual researchers to comprehensively survey the field, and expert-authored narrative reviews, while valuable, necessarily reflect subjective selection and emphasis. Asmussen and Møller (2019) demonstrate how topic modeling can provide data-driven mapping of research fields, revealing thematic structure through algorithmic analysis of large corpora. The availability of bibliographic databases such as Web of Science, whose coverage and limitations have been characterized by Pranckutė (2021), facilitates large-scale computational analysis.

Existing object detection surveys are predominantly expert-authored narratives that synthesize selected publications into coherent frameworks. While these surveys provide invaluable conceptual organization, they do not typically employ computational methods to discover thematic structure from the literature itself. The present study complements expert-driven reviews by providing a data-driven thematic map generated through unsupervised clustering and LLM-based interpretation.

2.5. Synthesis and research gap

The literature reviewed above reveals four intersecting gaps. First, systematic review automation has focused primarily on screening efficiency rather than thematic synthesis. Second, while advanced clustering methods produce sophisticated document partitions, they lack mechanisms for generating interpretable thematic descriptions. Third, large language models have demonstrated promise in systematic review workflows but remain underutilized for post-clustering interpretation. Fourth, object detection surveys rely on expert curation rather than data-driven thematic discovery. Moreover, we additionally focus on object detection with small data.

The present study addresses these gaps by constructing a multi-stage framework that combines unsupervised clustering of scientific abstracts with large language model-based thematic synthesis. By comparing three clustering pipelines (TF-IDF with K-means++, autoencoder with K-means, and deep embedded clustering) and two LLM variants (GPT-4o and o1) on approximately 18,500 object detection abstracts, we systematically evaluate the feasibility of automated thematic literature mapping.

Several recent systems combine multiple representation or retrieval strategies in ways that may appear similar to our pipeline. BERTopic (Grootendorst, 2022) combines transformer-based sentence embeddings with HDBSCAN density clustering and class-based TF-IDF topic representation, and recent extensions such as LLoM (Lam et al., 2024) incorporate LLM-generated concept labels on top of embedding clusters. ASReview (van de Schoot et al., 2021) integrates active-learning classifiers with transformer-based feature extractors to accelerate screening workflows. VOSviewer-style analyses (van Eck & Waltman, 2010) combine bibliometric co-citation networks with semantic similarity overlays to map research landscapes. These systems share our goal of scaling literature analysis, but they differ from our pipeline in two respects. First, they do not keep the clustering algorithm fixed across different representation strategies; as a result, making it difficult to attribute thematic differences to the representation rather than to the clustering mechanism. Second, they do not systematically compare LLM variants, such as GPT-4o and o1, as post-clustering interpreters on the same partitioning output. Our pipeline is designed explicitly as a controlled experimental framework rather than a production-oriented literature-review tool.

3. Methodology

The corpus used in this study consists of approximately 18,500 scientific abstracts retrieved from the Web of Science database. At this scale, manual thematic review becomes impractical, motivating the automated pipeline described here. The corpus spans object detection, medical imaging, fault diagnosis, few-shot learning, and related areas, resulting in a heterogeneous and methodologically demanding test corpus for clustering methods. The keyword-based retrieval protocol is fully reproducible, defined by an initial seed query, and a frequency threshold applied to an expanded keyword set. The resulting clusters are compared with existing expert-authored survey taxonomies in computer vision and machine learning, enabling qualitative validation that complements the quantitative metrics discussed later in the limitations section.

This section describes the three clustering pipelines evaluated in this study (see Figure 3). All three pipelines receive the same input: 18,500 scientific abstracts retrieved from the Web of Science database. Each pipeline produces cluster assignments for each abstract, per-cluster keyword sets extracted with YAKE, and thematic summaries generated by GPT-4o and ChatGPT-o1. The pipelines differ in how they represent abstracts before clustering. The first pipeline uses TF-IDF feature vectors and K-means++ (lexical representation). The second pipeline encodes abstracts with the all-MiniLM-L6-v2 sentence transformer, compresses the resulting 384-dimensional embeddings to 32 dimensions through a trained autoencoder, and then applies K-means++ (sequential semantic representation). The third pipeline uses an Improved Deep Embedded Clustering (IDEC) architecture that learns embeddings and cluster assignments jointly (joint deep representation). Together, the three pipelines form a progression from lexical to joint deep semantic representation.

Algorithm 1: Shared Workflow for the Three Clustering Pipelines

Input: corpus of scientific abstracts ($N \approx 18,500$)

Output: cluster labels L , cluster keywords K , thematic summaries S

Preprocessing

- a. Tokenize each abstract in C , convert to lowercase;
- b. Remove URLs, numbers, and punctuation;
- c. Store cleaned abstracts in C' ;

Feature extraction

- d. Pipeline 1: (TF-IDF and K-means++)
Compute TF-IDF matrix $X \in \mathbb{R}^{(N \times 5,000)}$ from C' ;
- e. Pipeline 2: (MiniLM+AE+K-means)
Encode C' with all-MiniLM-L6-v2 to obtain 384-dimensional embeddings;
Train autoencoder (384 $\rightarrow$ 128 $\rightarrow$ 32 $\rightarrow$ 128 $\rightarrow$ 384) on MSE loss;
Extract 32-dimensional latent vectors Z from encoder;
- f. Pipeline 3: (IDEC)
Phase 1: pretrain autoencoder on reconstruction loss (100 epochs);
Phase 2: extract latent vectors, initialize K centroids with K-means;
Phase 3: jointly fine-tune autoencoder and clustering head using loss (reconstruction MSE + KL divergence between q and p):

$$L = L_{rec} + \text{gamma} * L_{clust} \quad (1)$$

Cluster number selection

Compute within-cluster sum of squares (WCSS) for K in {2,...,20}

Compute mean silhouette score for K in {2,...,20}

g. Select candidate configurations: K = 8 and K = 11

Clustering

h. Apply K-means++ (Pipelines 1 and 2) or use IDEC cluster head (Pipeline 3) with the selected K values

i. Assign each abstract in C to a cluster; store labels in L

Keyword extraction

j. For each cluster c in L, concatenate all member abstracts

Apply YAKE to the concatenated text; retain top keywords

k. Store keyword set K[c] for each cluster c

LLM thematic summarization

l. For each cluster c, divide member abstracts into batches not exceeding 70% of the model token limit

Submit each batch to LLM with summarization prompt

m. Merge batch summaries into a single thematic summary S[c]

Repeat steps h and i

Cross-pipeline comparison

n. For each pair of pipelines (and for K = 8 and K = 11), construct a confusion matrix between the two label sets

o. Apply the Hungarian algorithm to find the optimal label permutation that minimizes total mismatch

p. Compute percentage-normalized confusion matrices after remapping

q. Record agreements and divergences for analysis.

3.1. Automated literature clustering method using standard tools

In this section, a method for automated literature filtration and analysis will be presented. The results of a developed method in this section will serve as a baseline to compare with AI-augmented methods described in the following sections.

Using the hand-picked keyword pool, we found close to 18,500 scientific articles, and given this number of articles, manually going through each is not feasible.

The filtration starts by retrieving abstracts for the entire literature pool and combining them into a single table to be processed further.

We begin clustering algorithm by preprocessing the abstract column imported from the table. Preprocessing stage includes converting text to lowercase, removing URLs, removing numbers, removing punctuation and tokenizing the text. Pre-processed abstracts are then used to create a multi-dimensional feature vector for each abstract. This is done by computing the TF-IDF score for each valuable term (max. 5000). The result of computing the TF-IDF of each valuable term for each abstract is a matrix of shape (number_of_documents,

number_of_features), where each row represents a document, and each column represents a term from the vocabulary:

$$X = \begin{pmatrix} x_1 & \cdots & z_1 \\ \vdots & \ddots & \vdots \\ x_n & \cdots & z_n \end{pmatrix}, \quad (2)$$

where $x_1, \dots, x_n$ represent the TF-IDF score for 1... n different terms that appear in abstract x . Each term may be found in all abstracts but may also be found only in some. The columns $x, \dots, z$ represent different abstracts.

Given this matrix X , it is now possible to launch the K-means algorithm, if we consider that $x_1, \dots, x_n$ is a point in a multi-dimensional space, and a distance between these points is a regular Euclidean distance. Euclidean distance is calculated by taking the square root of the sum of squared differences in the coordinates of the points:

$$D_{ij}^2 = \sum_{v=1}^n (x_{vi} - x_{vj})^2. \quad (3)$$

We used a modification to K-means called K-means++ which differs from the original in how the cluster centroids are initialized. Instead of randomly generating centroids, K-means++ is generating centroids sequentially and considers existing centroids when initiating a new one, ensuring better quality of clustering result (Bradley & Fayyad, 1998). The algorithm for generating a list of centroids is as follows.

1. Choose the very first centroid randomly. Each unchosen data point x has a probability of being selected as the next centroid:

$$P(x_i) = \frac{D(x_i)^2}{\sum_{j=1}^N D(x_j)^2}. \quad (4)$$

2. Where $D(x)$ is the distance from point x to its nearest initialized centroid. The distance here is squared to ensure that farther points are more likely to be selected but not guaranteed.
3. Once the probabilities are calculated, the random weighted selection is performed where points with higher $P(x)$ have a greater chance of being chosen.
4. Repeat steps 2 and 3 until enough cluster centroids are initiated.

Using this algorithm the centroids initiation is done much better compared to pure random selection as some clusters may start too close to each other when choosing randomly, leading to bad clustering results.

When the whole matrix is processed by K-means, it results in generated clusters with nearest articles assigned to each. Using these clusters, YAKE (Yet Another Keyword Extractor) python library, is used to generate cluster labels by extracting significant keywords from the textual content of each cluster. YAKE uses text statistical features extracted from single documents to select the most important keywords of a text (from YAKE documentation).

In order to run a clustering visualization algorithm, we first had to determine the actual number of clusters that we need. For this purpose, we turned to the elbow and silhouette score methods, which are frequently used in statistics.

The elbow method is usually used with K-means algorithm to determine best K by plotting within-cluster sum of squares (WCSS) to the picked number of clusters (1, 2, 3, ... and so on). WCSS represents the total squared distance between each data point in a cluster and the cluster's centroid:

$$WCSS = \sum_{i=1}^k \sum_{x \in C_i} \|x - \varepsilon_i\|^2, \quad (5)$$

where k – is number of clusters used in K-means to divide the abstracts, x – is an abstract (point) from cluster C_i , ε_i – cluster centroid for C_i .

We also decided to run a silhouette scores method and build a diagram using this method for further improvement in clarity for picking optimal number of clusters.

The silhouette scores method performs scoring of each clustering configuration by calculating a mean silhouette value over all the data points in the pool, in our case all the articles.

$$s = \sum_{i=1}^A \frac{s(i)}{A}, \quad (6)$$

$s(i)$ being silhouette value per data point, A – overall number of points (articles). Silhouette value per data point is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (7)$$

$$a(i) = \frac{1}{A_i - 1} \sum_{j \in C_i, i \neq j} d(i, j); \quad (8)$$

$$b(i) = \min_{J \neq I} \frac{1}{A_J} \sum_{j \in C_J} d(i, j), \quad (9)$$

$d(i, j)$ here represents a distance between i and j data points, C_i represents a cluster and A_i – number of articles in a cluster C_i .

As will be discussed in the results section, we have determined that the best configuration of clusters would be either **8** or **11 clusters** based on the obtained data.

3.2. Cluster summarization via large language models

To facilitate the interpretation of cluster content obtained from the K-means++ algorithm, we leveraged large language models (LLMs), specifically ChatGPT-4o and ChatGPT-o1, to generate representative summaries for each cluster. Abstracts from each cluster were compiled into plain text files, where each line contained a separate abstract. These were submitted to the LLMs with the prompt: *“Based on the following collection of abstracts, generate a single representative abstract that captures the main themes, topics, and findings across them. The result should read like a coherent scientific abstract, reflecting the overall content of the input.”*

The generated summaries revealed the thematic focus of each cluster. Although lexical formulations varied slightly between the two models, the generated content was conceptually stable across models.

During initial testing, we encountered token length limitations, which meant that only partial data was used per request. To address this, abstracts were tokenized and randomly shuffled within each cluster. We then divided the abstracts into batches not exceeding 70% of the LLM's maximum token capacity and submitted them in multiple requests. Each batch's summary was evaluated and merged into a final thematic synthesis per cluster.

3.3. Neural clustering with autoencoders

Following the K-means clustering approach, we explored a neural method for unsupervised clustering based on an autoencoder architecture trained on abstract embeddings. The goal was to evaluate whether a latent space learned from the abstract text could yield a meaningful thematic organization of scientific literature.

We used the all-MiniLM-L6-v2 SentenceTransformer model to embed each abstract into a 384-dimensional vector. These embeddings were then passed through an autoencoder composed of two fully connected neural blocks:

Input dimensions: 384

Encoder:

- Layers: Linear(384 $\rightarrow$ 128) $\rightarrow$ ReLU $\rightarrow$ Linear(128 $\rightarrow$ 32)
- Output: 32-dimensional latent representation

Decoder:

- Layers: Linear(32 $\rightarrow$ 128) $\rightarrow$ ReLU $\rightarrow$ Linear(128 $\rightarrow$ 384)
- Output: reconstructed embedding

The architecture of the standalone autoencoder is shown in Figure 1.

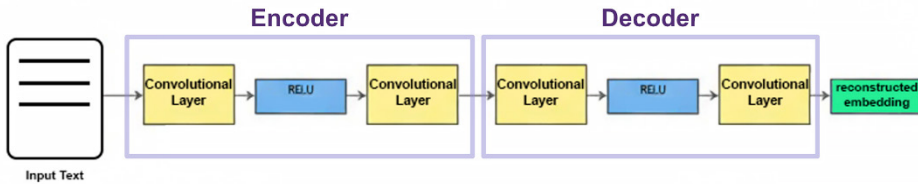


Figure 1. Architecture of the standalone autoencoder: encoder 384-128-32, decoder 32-128-384 with ReLU activations; the 32-d bottleneck feeds K-means clustering

The model was trained on mean squared error (MSE) between input and reconstructed vectors, using the Adam optimizer with a learning rate of 1e-3 and a batch size of 32. The training ran for 100 epochs, using GPU acceleration.

After training, the 32-dimensional latent vectors were extracted from the encoder and clustered using K-means with the same number of clusters (8 or 11) as in our baseline experiments. Dimensionality reduction (PCA) was then applied to visualize cluster distributions.

To assess correspondence between neural clustering and the original K-means results, we applied the trained autoencoder and clustering model to the same set of abstracts and computed a confusion matrix between true (baseline) and predicted (autoencoder-based) cluster labels. Results were visualized as a heatmap with percentage-normalized values.

The analysis revealed both agreements and divergences between the two clustering methods. Some neural clusters showed high alignment with their TF-IDF counterparts, while others merged or split existing themes, potentially reflecting more abstract groupings derived from semantic structure rather than keyword frequency.

We selected all-MiniLM-L6-v2 for its balance of computational efficiency and embedding quality on general-domain text (Wang et al., 2020). Domain-specific models such as SPECTER2 (Singh et al., 2023) represent a natural extension, however, our corpus spans multiple domains (medical imaging, fault diagnosis, graph networks) rather than a single scientific field, making a general-purpose encoder a reasonable default. Comparing embedding models is left for future work.

3.4. Deep embedded clustering with autoencoder-based latent space

To move beyond traditional clustering applied post-hoc to fixed embeddings, we developed a unified deep neural network that integrates both representation learning and clustering into a single architecture. This design is inspired by the Deep Embedded Clustering (DEC) framework and aims to leverage the full potential of neural networks for unsupervised document grouping.

The model consists of a symmetric autoencoder augmented with a clustering head. Its structure is as follows:

Encoder:

- Linear(384 → 128) → ReLU → Linear(128 → 32)

Decoder:

- Linear(32 → 128) → ReLU → Linear(128 → 384)

Clustering Head:

- A learnable set of cluster centroids of shape ($n_{\text{clusters}} \times 32$) that interacts with the latent vectors to compute soft assignments. These assignments are calculated using a Student's t-distribution over the squared Euclidean distance between each latent vector and each centroid. This results in a matrix q of soft cluster memberships for every abstract.

The model outputs both a reconstructed input and cluster assignment probabilities in a single forward pass. The complete IDEC architecture with the clustering head is shown in Figure 2.

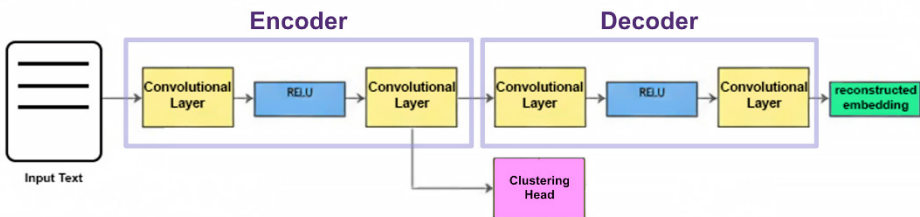


Figure 2. Architecture of the Improved Deep Embedded Clustering (IDEC) model: the autoencoder of Figure 1 extended with a learnable clustering head of 32-d centroids producing soft assignments

Initially, we trained the entire network – including the clustering head – from scratch using a combined objective:

$$\mathcal{L}_{total} = \mathcal{L}_{reconstruction} + \lambda \times \mathcal{L}_{KL}(q \parallel p), \quad (10)$$

where q represents the soft cluster assignments and p is a sharpened target distribution computed as in the IDEC framework. This formulation aimed to balance the autoencoder's reconstruction capabilities with a differentiable clustering objective.

However, this approach led to unstable training and poor clustering performance. Specifically, several clusters collapsed – i.e., they remained empty throughout training – while others dominated by absorbing most samples. This imbalance was caused by the randomly initialized centroids having no meaningful alignment with the latent structure, preventing convergence toward semantically valid clusters.

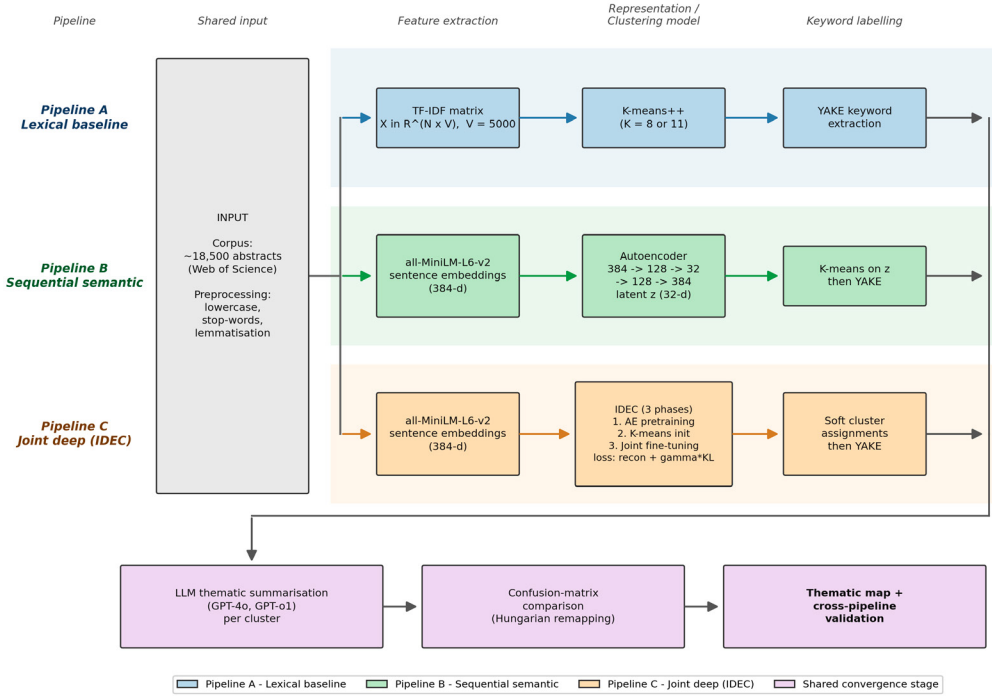


Figure 3. Shared methodology pipeline showing the three parallel clustering pipelines (A lexical, B sequential semantic, C joint deep IDEC) sharing input and output stages. See Algorithm 1 for the pseudocode counterpart

After clustering, we compare the three pipelines using confusion matrices for both the 8- and 11-cluster configurations as well as by computing cross cluster similarity metrics. Because cluster indices are arbitrary across pipelines, we apply the Hungarian algorithm to find the label permutation that minimizes total assignment cost and then compute percentage-overlap matrices to quantify how much of each reference cluster each predicted cluster

recovers. Separately, LLM-based thematic summaries are generated independently for each pipeline and each cluster configuration, producing six pipeline-configuration summary sets (three pipelines by two cluster counts), each generated by both GPT-4o and ChatGPT-o1.

4. Results

We developed a list of keywords potentially associated with the concept of small data and conducted a search for related articles in the Web of Science database. This initial search yielded over 4,000 articles. Subsequently, we performed a frequency analysis of the keywords used in these articles, filtering out irrelevant ones. Using the 30 most popular keywords from this analysis, we extended the search and identified nearly 18,500 articles.

Next, we organized the articles by publication year and sorted them based on their average number of citations per month since publication. This approach allowed us to prioritize the most impactful and relevant articles for further analysis.

4.1. Results of the automated literature clustering

The results of a developed method in this section will serve as a baseline to compare with AI-augmented methods described in the following sections. The results of the elbow method are presented in Figure 4.

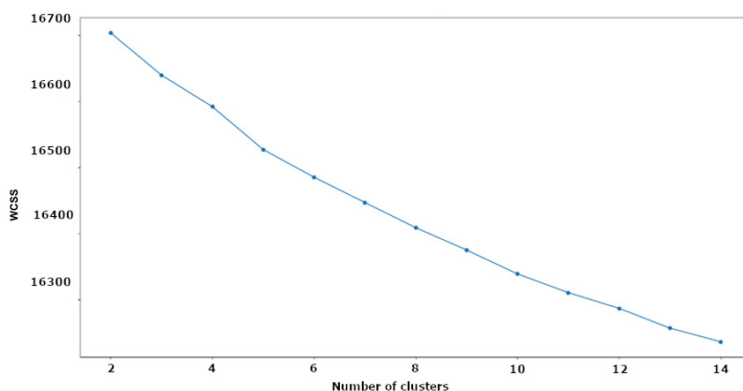


Figure 4. Within-cluster sum of squares (WCSS) versus K for the TF-IDF + K-means++ baseline on 18,500 abstracts

It is quite difficult to determine the number of clusters required judging only by this diagram, so it was decided to also run a silhouette scores method and build a diagram using this method.

The corresponding silhouette scores are presented in Figure 5.

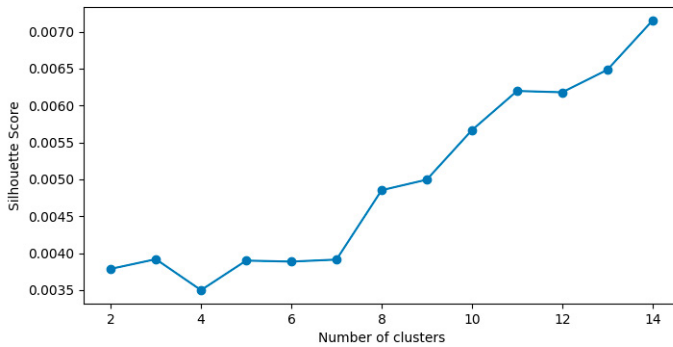


Figure 5. Mean silhouette coefficient versus K for the TF-IDF + K-means++ baseline on 18,500 abstracts

By looking at this visual representation of the silhouette scores (the larger the score the better), we determined that the best configuration of clusters would be either **8** or **11 clusters**.

The whole clustering process produces a CSV file, while Figure 6 presents a visual representation of the resulting clusters with generated labels.

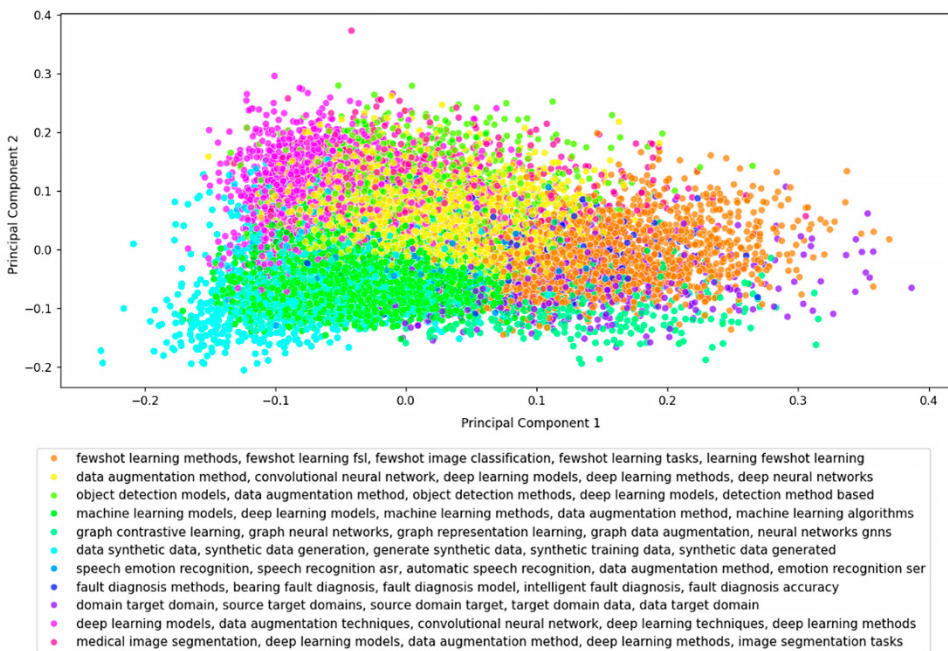


Figure 6. TF-IDF Clustering result, visual representation

4.2. AI summarization results

Using the configuration discussed previously, we ran the labelling task which has yielded a clearly defined thematic groups:

- **Cluster 0:** Object detection and anomaly recognition in industrial and visual contexts.
- **Cluster 1:** Medical image segmentation.
- **Cluster 2:** Synthetic data, federated learning, and privacy-preserving machine learning.
- **Cluster 3:** Few-shot learning, meta-learning, and small data applications.
- **Cluster 4:** Fault diagnosis in engineering and industrial systems.
- **Cluster 5:** Graph neural networks (GNNs) and graph-based data processing.
- **Cluster 6:** Broad AI applications (finance, healthcare, industrial) with focus on trust, explainability, and generalization.
- **Cluster 7:** Classification tasks, largely medical, using CNN-based architectures.

To examine the effects of finer partitioning, we ran the same labelling task on the 11-cluster configuration. While thematic resolution improved, methodological overlap between some clusters became more apparent:

- **Cluster 0:** Medical diagnostics and classification, focusing on disease recognition using deep learning models such as CNNs and GANs.
- **Cluster 1:** General AI and machine learning applications with emphasis on interpretability, ethical considerations, and societal impact.
- **Cluster 2:** Fault diagnosis in industrial and mechanical systems, focusing on condition monitoring and predictive maintenance.
- **Cluster 3:** Synthetic data generation, federated learning, and privacy-preserving approaches in machine learning.
- **Cluster 4:** Object detection and anomaly recognition across various domains, including industrial, visual, and security applications.
- **Cluster 5:** Medical image segmentation, involving pixel-level classification tasks in radiology, pathology, and related fields.
- **Cluster 6:** Domain adaptation and transfer learning, targeting model generalization across data distributions and use cases.
- **Cluster 7:** Speech-related applications including automatic speech recognition (ASR), emotion detection, and audio-based diagnostics.
- **Cluster 8:** General computer vision tasks focused on recognition, detection, and classification in diverse application areas.
- **Cluster 9:** Graph-based methods and Graph Neural Networks (GNNs), addressing structured data analysis in domains like medicine and IoT.
- **Cluster 10:** Few-shot learning and meta-learning, aimed at solving problems with limited labelled data and enhancing model adaptability.

These intersection metrics, while qualitative, were based on shared techniques, keyword co-occurrence, and LLM-generated descriptions. They reveal that although clustering achieves clear topical separation, real-world AI/ML research frequently spans multiple methodologies and application domains. For example, clusters focused on medical diagnostics (Cluster 0) and medical segmentation (Cluster 5) often employ similar deep learning techniques such as CNNs and UNet architectures. Cluster 10, centered on few-shot and meta-learning, shares methodological foundations with Clusters 3 and 6, which deal with synthetic data, privacy,

and domain adaptation. Likewise, Cluster 3 and Cluster 1 both explore federated learning and ethical considerations, albeit from different perspectives – technical and societal, respectively. Applications of GANs, domain transfer, and low-data learning appear across multiple clusters, including Clusters 0, 3, 6, and 10. Even more specialized domains such as speech and emotion recognition (Cluster 7) intersect methodologically with other clusters through the shared use of few-shot learning and convolutional architectures. This complex interrelation highlights the utility of conversational AI for navigating and synthesizing thematic overlaps in large-scale, multi-topic scientific corpora.

4.3. Neural clustering results

4.3.1. Neural clustering – autoencoder

Using the neural clustering method, the data was again sorted into 8 and 11 clusters as previously. Figure 7 presents the 8-cluster configuration projected onto a two-dimensional space using PCA.

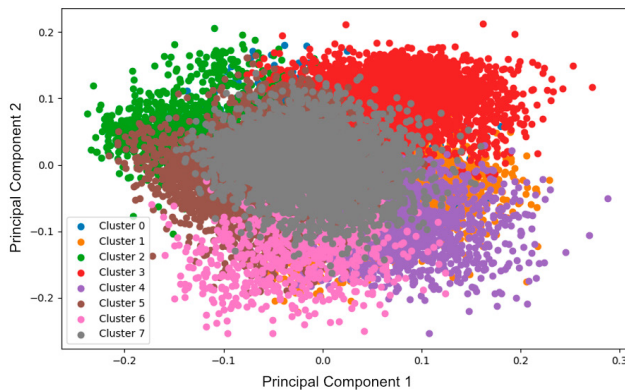


Figure 7. 2D PCA projection of autoencoder + K-means clusters ($K = 8$) on 18,500 abstracts; colours denote cluster assignments

The corresponding 11-cluster configuration is presented in Figure 8.

It is apparent that this clustering method produced clearly distinct groups of articles, however, to further investigate the robustness of the proposed autoencoder clustering pipeline, the experiments were repeated 5 times for both evaluated cluster configurations ($K = 8$ and $K = 11$) and results were then averaged. This evaluation was introduced to reduce the possibility that the observed clustering quality was influenced by initialization randomness or stochastic optimization effects.

Internal clustering quality was evaluated using the silhouette coefficient, Davies-Bouldin index, and Calinski-Harabasz score. The silhouette coefficient values for the autoencoder + K-Means approach were relatively low, reaching 0.063 for $K = 8$ and 0.066 for $K = 11$. These

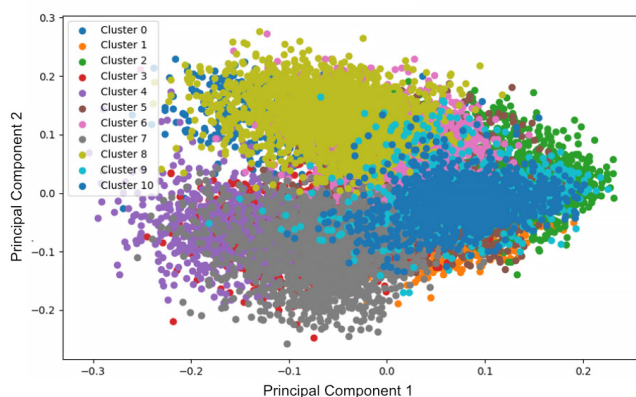


Figure 8. 2D PCA projection of autoencoder + K-means clusters ($K = 11$) on 18,500 abstracts

results suggest that although the latent representations contained some cluster structure but the boundaries between clusters are poorly separated. Similarly, the Davies-Bouldin index values were comparatively high (2.95 for $K = 8$ and 2.77 for $K = 11$), indicating increased overlap and lower separation between neighboring clusters. The Calinski-Harabasz scores reached approximately 645 for the 8-cluster configuration and 572 for the 11-cluster configuration, which further reflects a weaker overall partitioning structure within the latent space produced by the standalone autoencoder.

The calculated standard deviation values remained relatively low across all evaluated metrics, indicating stable clustering behavior between repeated experimental runs. The silhouette coefficient standard deviation remained below 0.008 for both cluster configurations, suggesting that the compactness and separation quality of the generated clusters changed only minimally across repetitions. Similarly, the Davies-Bouldin index and Calinski-Harabasz score exhibited low variation (~ 0.1 for both cluster configurations), demonstrating that the overall structural organization of the latent space remained consistent despite stochastic initialization effects during neural network training.

4.3.2. DEC with autoencoder-based latent space results

The results of the naive end-to-end DEC clustering for $K = 8$ and $K = 11$ are presented in Figures 9 and 10, respectively.

These results revealed that a purely end-to-end approach was insufficient in this context. Without prior organization in the latent space, the clustering mechanism failed to discover balanced groupings, especially on high-dimensional semantic data like research abstracts.

To overcome these limitations, we adopted a three-phase training protocol, which provided both stability and accuracy:

Autoencoder Pretraining: The autoencoder was trained independently on the reconstruction loss for 100 epochs, with the clustering head frozen.

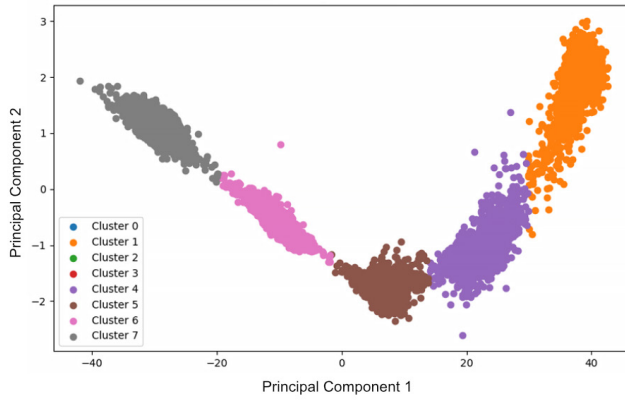


Figure 9. 2D PCA projection of naive end-to-end DEC clusters ($K = 8$) on 18,500 abstracts, showing cluster collapse

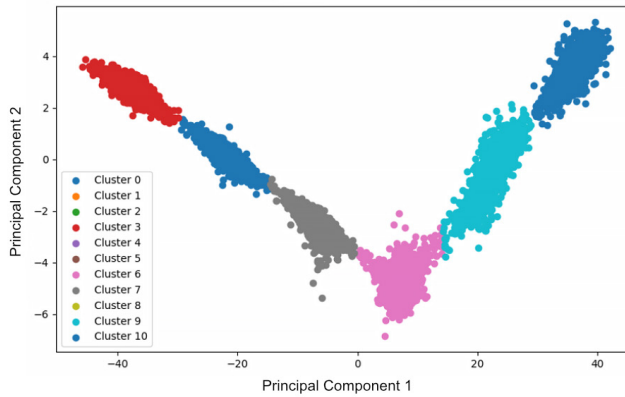


Figure 10. 2D PCA projection of naive end-to-end DEC clusters ($K = 11$) on 18,500 abstracts, showing cluster collapse

Centroid Initialization with K-Means: After pretraining, the encoder’s latent representations were extracted, and K-Means was applied to initialize the cluster centroids. These centroids were copied into the trainable clustering layer.

Joint Fine-Tuning: The entire model was then fine-tuned end-to-end using a weighted sum of reconstruction loss and KL divergence between the soft assignments q and the target distribution p . We used a weighting factor $\lambda = 0.1$ to moderate the influence of the clustering objective.

This method produced well-separated, balanced clusters and eliminated the earlier issue of collapsed centroids.

The total loss was computed as:

$$\mathcal{L}_{total} = \mathcal{L}_{reconstruction} + \lambda \times \mathcal{L}_{KL}, \quad (11)$$

where $\mathcal{L}_{reconstruction}$ is the mean squared error between input and reconstruction, and $\mathcal{L}_{KL}$ is the *Kullback-Leibler* divergence between the predicted cluster assignments q and the target distribution p . The weighting factor $\lambda = 0.1$ was empirically chosen to balance both objectives. The resulting IDEC cluster configurations for $K = 8$ and $K = 11$ are presented in Figures 11 and 12, respectively.

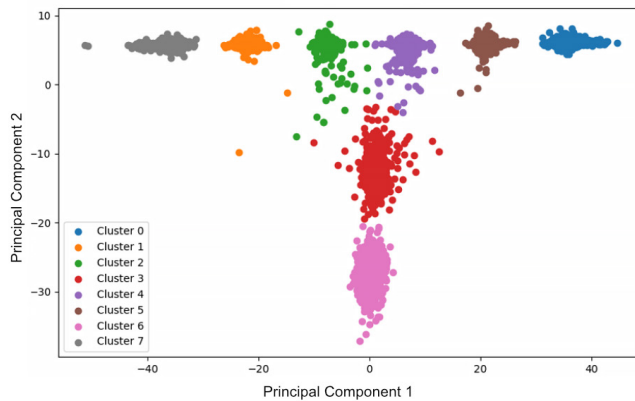


Figure 11. 2D PCA projection of IDEC (three-phase training) clusters ($K = 8$) on 18,500 abstracts

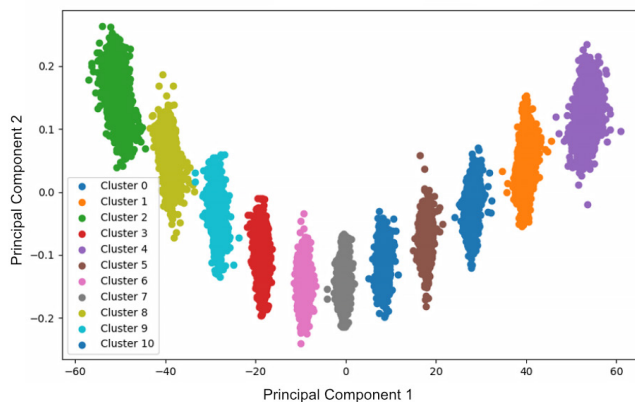


Figure 12. 2D PCA projection of IDEC (three-phase training) clusters ($K = 11$) on 18,500 abstracts

The structured approach proved critical for stable optimization and reliable clustering. Notably, the latent space produced by the pretrained autoencoder provided sufficient structure for the clustering head to initialize meaningfully and refine during fine-tuning.

For the $K = 8$ configuration, the IDEC model achieved a silhouette score of 0.855, while the $K = 11$ configuration produced a slightly lower but still strong value of 0.820. These results

indicate that the learned latent representations formed compact clusters with clear separation between neighboring groups. The Davies-Bouldin index remained low in both cases (0.191 for $K = 8$ and 0.227 for $K = 11$), further supporting the conclusion that the clusters were internally coherent and well separated. Similarly, the Calinski-Harabasz index produced exceptionally high values exceeding 477,000 for both configurations and reaching over 555,000 for $K = 8$, indicating a highly structured partitioning of the latent feature space.

In addition to internal validation metrics, external comparison metrics were computed against a reference clustering baseline generated using TF-IDF representations. For this purpose, Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) scores were calculated between TF-IDF clustering assignments and IDEC-generated assignments. The obtained ARI values (~ 0.048) and NMI values (~ 0.116 – 0.126) indicate that the neural clustering solution produced substantially different partitioning structures compared to the traditional TF-IDF approach. While the overlap between assignments remained limited, this divergence is expected due to the fundamentally different representation spaces used by the two methods. TF-IDF primarily captures sparse lexical similarity, whereas IDEC operates on dense semantic embeddings learned through nonlinear latent representations. Consequently, the IDEC model was able to form clusters based on deeper semantic relationships between abstracts rather than relying solely on term-frequency statistics.

The standard deviation values were consistently low just like in the previous section, further confirming the robustness of the proposed three-phase training strategy. The silhouette coefficient standard deviation remained close to 0.01 for both cluster configurations, while the Davies-Bouldin index variation remained below 0.03, indicating highly stable cluster compactness and separation properties across runs. Although the Calinski-Harabasz index exhibited larger absolute standard deviation values ($\sim 123,000$ for both configurations), this variation, although higher than desirable, remained proportionally small relative to the exceptionally high overall metric magnitude exceeding 477,000–555,000.

The repeated experiments demonstrated that the proposed training strategy not only stabilizes optimization but also produces highly consistent clustering behavior across multiple runs. The additional evaluation metrics provide quantitative evidence that the latent space learned by the IDEC architecture remains both structurally coherent and semantically discriminative, supporting its suitability for large-scale scientific abstract clustering tasks.

The following sections will focus on comparison of the data obtained from neural clustering with the data obtained initially with traditional methods.

4.4. Neural clustering comparison with keyword-based clustering

Based on the results of the two different clustering approaches, we constructed confusion matrices (for 8 cluster and 11 cluster configurations respectively) and cluster distribution diagram which might be very helpful for further analysis.

Figure 13 presents the original and adjusted confusion matrices for the 8-cluster configuration. Even though neural clustering, just like TF-IDF approach, results in a column containing the number of an assigned cluster for each article, the numbers are arbitrary, meaning that first cluster from TF-IDF clustering is not necessarily corresponding to the first cluster from neural clustering. This brings a new challenge of mapping matching clusters as

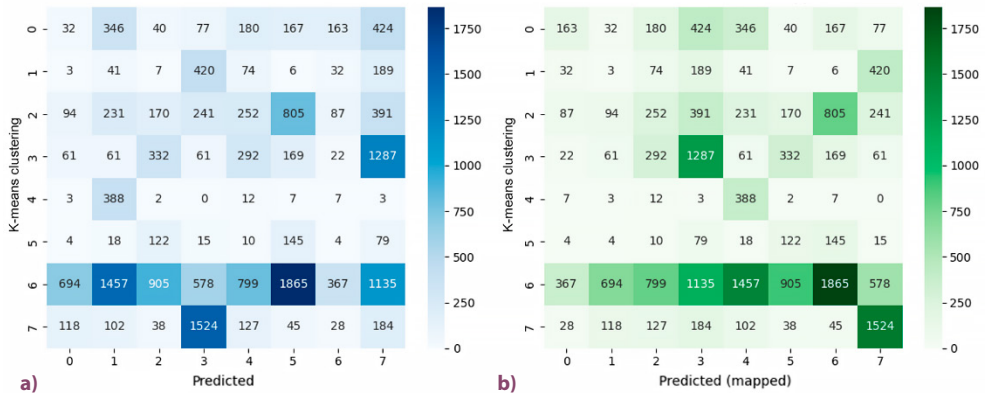


Figure 13. Confusion matrices for TF-IDF + K-means++ versus autoencoder + K-means at K = 8: a – raw counts; b – Hungarian-algorithm remapped

close as possible, and so Hungarian Algorithm was used for cluster mapping. The Hungarian Algorithm is a combinatorial optimization method used to solve assignment problems, and for our purposes the algorithm treats the problem of matching clusters from two different methods as a cost-minimization task: the “cost” represents the mismatch between items assigned to a given pair of clusters. As a result, we have an adjusted cluster matching set, which is presented on the right-hand side of the figure above. Ideally, we should be seeing most of the articles mapped on the diagonal squares of the diagram, however, a lot of articles happen to be assigned different clusters than what TF-IDF approach suggests. To examine the distribution of cluster overlap in percentage terms, the percentage-based confusion matrix shown in Figure 14 was constructed.

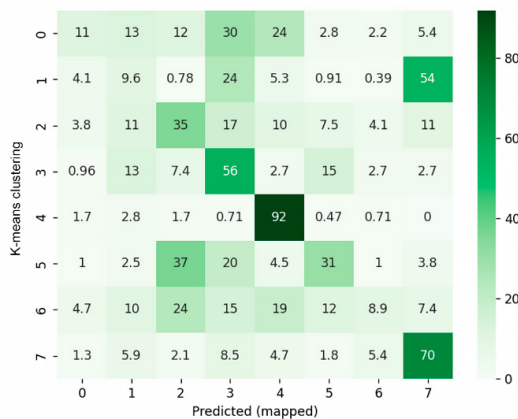


Figure 14. Percentage-based overlap confusion matrix for TF-IDF + K-means++ versus autoencoder + K-means at K = 8 (rows sum to 100%)

The mapping from Hungarian algorithm used in the previous figure doesn't necessarily translates to a good mapping for a percentage-based approach, so clusters were re-mapped again to reflect the transition to the percentage representation of cluster overlap. Based on this figure, cluster 4, labelled "intelligent bearing fault diagnosis", has the most matches out of all others, with cluster 7 and 3 falling behind with 70% and 56% of overlap respectively. The number here represents how much each predicted cluster contains the articles from clusters created with TF-IDF, numbers of a single row are adding up to 100% of the corresponding TF-IDF cluster. Interestingly, 7th cluster from predicted contains the articles from the 1st TF-IDF cluster, 54% of them, suggesting that clusters 1 and 7 from the TF-IDF clustering may be related to one another.

Same can be said about clusters 2, 5 and 6, considering that predicted cluster 2 contains 35%, 37% and 24% of each respectively.

The rest of clusters have unremarkable overlap, with 54% of cluster 0 being split between predicted clusters 3 and 4.

For the 11-cluster configuration the corresponding confusion matrices are presented in Figure 15.

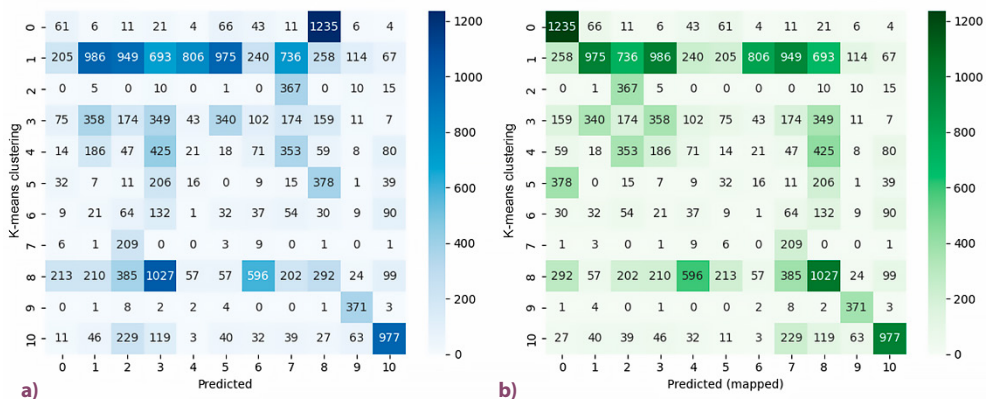


Figure 15. Confusion matrices for TF-IDF + K-means++ versus autoencoder + K-means at K = 11: a – raw counts; b – Hungarian-algorithm remapped

Same as with 8 clusters, the mapping algorithm was used to achieve adjusted confusion matrix presented on the right of this figure. From these diagrams it is visible that the overlap is as inconsistent as with previous configuration, but there are a few curious places. The overlap is similarly inconsistent in this configuration; therefore, the percentage-based overlap matrix shown in Figure 16 was calculated for further analysis.

Here, as with 8-cluster configuration, the clusters were re-mapped again to fit better the percentage-based representation, and there are a couple of interesting details that deserve attention. Predicted cluster 9 contains 95% of its TF-IDF counterpart, which can be regarded as an extremely good match. As well, clusters 0, 2 and 7 are almost fully matching, with 10th cluster falling behind at 62% – while still a good result it is noticeably worse than the others

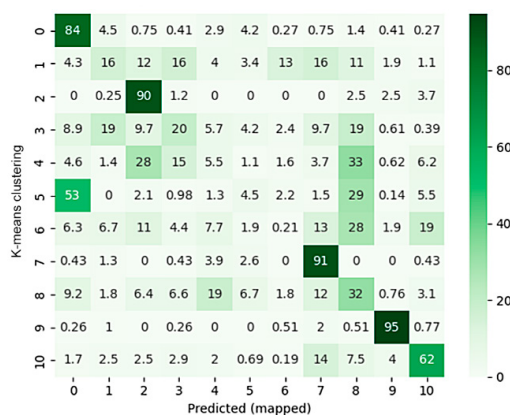


Figure 16. Percentage-based overlap confusion matrix for TF-IDF + K-means++ versus autoencoder + K-means at K = 11 (rows sum to 100%)

mentioned. Predicted cluster 0 also contains 53% of the 5th cluster from TF-IDF approach, which could be supported by the fact that proposed labels for the cluster 0 and 5 contain quite an overlap as visible on Figure 6 of this article. Otherwise, the articles from all other clusters are fairly dispersed over the predicted clusters, like 1st, 3rd, 4th, 6th, and 8th clusters.

Following the confusion matrix comparison, specific comparison metrics were calculated using TF-IDF-based cluster assignments as a reference baseline. The Adjusted Rand Index (ARI) values ranged from 0.089 for K = 8 to 0.118 for K = 11. ARI operates by comparing pairs of samples and checking whether both methods place them in the same or different clusters. Values close to 1 indicate highly similar cluster structures, values near 0 indicate random agreement, and negative values suggest disagreement worse than random assignment. These results show that the methods produced quite different cluster assignments. Following ARI, the corresponding Normalized Mutual Information (NMI) scores were calculated. NMI evaluates the amount of shared information between two clustering solutions and was included to measure the consistency of cluster structures independently of cluster label permutations. The metric is based on information theory and quantifies how much knowing one clustering assignment reduces uncertainty about the other. Values as well range from 0 to 1, where higher values indicate stronger structural similarity between clustering results. The NMI scores ranged from 0.202 to 0.241 supporting the ARI conclusion that the clustering using the two methods produces quite different results. These differences do not necessarily imply inferiority of either method. Rather, they indicate that lexical and semantic representations capture different forms of document similarity.

4.5. Improved deep embedded clustering comparison with keyword-based clustering

To begin the comparison of IDEC approach with the keyword-based clustering, below is presented a confusion matrix, featuring the 8-cluster configuration.

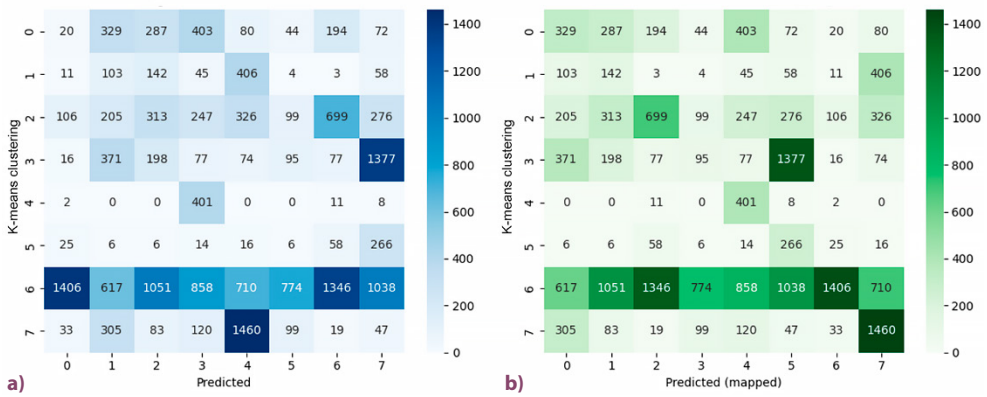


Figure 17. Confusion matrices for TF-IDF + K-means++ versus IDEC at K = 8: a – raw counts; b – Hungarian-algorithm remapped

Here, on the Figure 17 above it is visible that cluster 6 from TF-IDF approach is spread almost uniformly over all 8 clusters in the IDEC approach, quite similar to what happened in the previous comparison of Autoencoder approach with TF-IDF clustering, potentially meaning that 6th cluster was too broad in its definition to begin with. To further analyze the data, confusion matrix was constructed based on cluster overlap percentages.

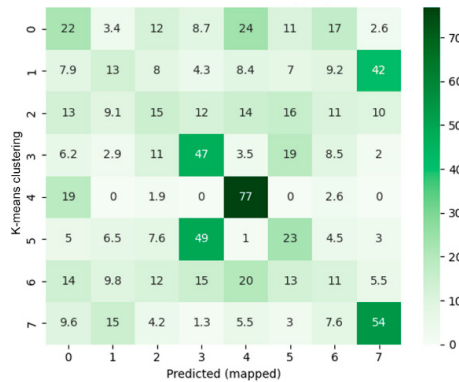


Figure 18. Percentage-based overlap confusion matrix for TF-IDF + K-means++ versus IDEC at K = 8 (rows sum to 100%)

On the Figure 18 it is clear that the best match happens to be cluster 4 with 77% overlap between approaches with cluster 7 falling behind at 54% overlap. Predicted cluster 3 contains almost half of cluster 3 and 5 from TF-IDF approach, suggesting they can be related. Another interesting feature is predicted cluster 7 containing 42% of the 1st cluster from the initial approach.

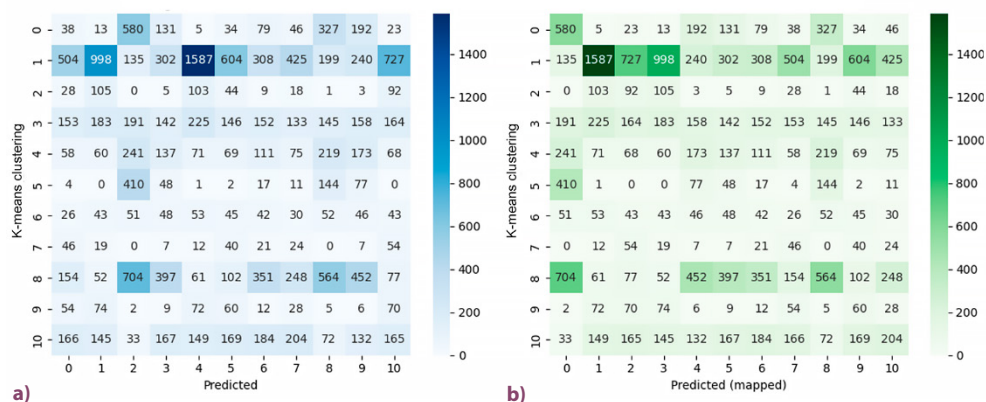


Figure 19. Confusion matrices for TF-IDF + K-means++ versus IDEC at K = 11: a – raw counts; b – Hungarian-algorithm remapped

In the 11-cluster configuration case, presented on Figure 19 above, the mapping is similar to all the previous cases, with some major overlap for handful of clusters and unremarkable results for others.

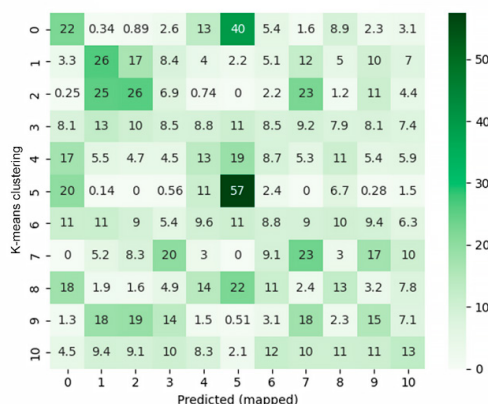


Figure 20. Percentage-based overlap confusion matrix for TF-IDF + K-means++ versus IDEC at K = 11 (rows sum to 100%)

Figure 20 presents the percentage-based overlap matrix for the 11-cluster TF-IDF versus IDEC comparison. The clusters were re-mapped to better align with the percentage-based representation, and several points are worth highlighting. Several clusters exhibited partial thematic convergence across pipelines. For example, the IDEC cluster aligned with 57% of the TF-IDF medical-imaging cluster while simultaneously absorbing approximately 40% of another medically oriented TF-IDF cluster, suggesting that semantic embeddings merged lexically distinct but conceptually related medical subdomains. Predicted clusters 0, 1, 2 and 7 are all sharing some overlap with their TF-IDF counterparts, although it is a debatable whether 20–30% of overlap range can be considered meaningful.

To further compare the IDEC clustering results with a traditional text-based approach, additional external comparison metrics were computed against cluster assignments generated using TF-IDF representations. As previously, the ARI and NMI were calculated between TF-IDF-based clusters and IDEC-generated clusters. The obtained ARI scores were approximately 0.048 for both cluster configurations, while NMI values ranged from 0.116 for $K = 8$ to 0.126 for $K = 11$. These relatively low agreement scores suggest that the IDEC model produced substantially different clustering structures compared to TF-IDF-based clustering. Such divergence is again expected because the two approaches rely on fundamentally different representation paradigms. TF-IDF captures sparse lexical co-occurrence patterns, whereas IDEC learns dense nonlinear semantic embeddings capable of grouping documents according to contextual and semantic similarity rather than surface-level vocabulary overlap.

4.6. Comparison metrics for the two neural clustering methods

To further evaluate the relationship between the two neural clustering approaches, additional comparison metrics were calculated directly between the cluster assignments produced by the standalone autoencoder + K-Means pipeline and the IDEC-based clustering method. In this case, the comparison focused on identifying how similarly both neural approaches partitioned the same dataset under identical cluster count configurations.

For the 8-cluster configuration, the comparison produced an ARI score of 0.245 and an NMI score of 0.329. When the number of clusters was increased to 11, the similarity between methods decreased slightly, resulting in an ARI value of 0.174 and an NMI value of 0.290. These results indicate that although both methods identified partially overlapping structural patterns in the dataset, the resulting cluster assignments remained substantially different. The observed divergence can be explained by the different optimization objectives used by the two methods. The standalone autoencoder primarily learns latent representations optimized for reconstruction quality, whereas IDEC additionally applies clustering-oriented optimization that explicitly reshapes the latent space during training. As a result, the IDEC method produces cluster structures that are more strongly influenced by cluster compactness and separability objectives, while the standalone autoencoder retains representations more closely tied to reconstruction fidelity.

The decrease in similarity metrics for the 11-cluster configuration further suggests that the differences between the two methods become more pronounced as clustering granularity increases. This indicates that finer-grained semantic partitioning amplifies the impact of clustering-aware optimization on the final latent space organization.

The comparison metrics between the autoencoder and IDEC clustering methods demonstrated relatively low variability across repeated runs. For the 8-cluster configuration, the ARI and NMI standard deviations were 0.045 and 0.052, while the 11-cluster configuration exhibited even lower variability of 0.028 ARI and 0.042 NMI. These results indicate that the structural differences observed between the two neural clustering approaches were not caused by random fluctuations during training but instead reflected systematic differences in the learned latent space organization.

5. Discussion

The results of comparative analysis provide a wide view of the object detection research field and offer relevant insights into methodologies for literature analysis. The following discussion interprets these findings, providing a context of contributions and discussion in the context of “AI for science”.

5.1. From lexical patterns to semantic structures

The three methodological approaches evaluated in this study represent a progression from lexical pattern matching to deep semantic abstraction. The TF-IDF with K-means ++ plays the role of a baseline which is highly interpretable (it is also computationally efficient, which is not important in context of current research). It is strong due to direct mapping of keyword frequencies to thematic groups leading to a creation of the dominant terminology in the field. However, this method is limited by shortcomings of bag-of-words models – it cannot take into account synonymy, polysemy and broader semantic context, a well-documented challenge in texts processing (Xu et al., 2023).

The differences between baseline and neural network methods do not indicate errors, instead they provide evidence of fundamental differences due to different abstraction principles:

- TF-IDF tend to identify scientific communities using the same specific terms (“fault diagnosis”, “GAN”).
- Neural embedding identifies conceptual communities.

Neural models group articles based on their ideas and problem statements. For example, “medical diagnostics” merges with “medical image segmentation” into single cluster by both, the autoencoder and DEC models – these topics have different keywords but are conceptually close as they address similar high-level challenges within medical imaging. Confusion matrices reveal where the terminology of the field aligns with its conceptual meaning or diverges from it. For full understanding a scientific domain requires mapping both terminology and conceptual structure, thus the comparative analysis presented here provides a powerful tool to achieve multi-layered presentation.

Out of three methods the IDEC model is theoretically the most potent one. Different from sequential approaches, IDEC optimizes the representation of learning and cluster assignment. This allows the model to learn feature representations that are suitable for clustering rather than merely descriptive.

Taken together, the results from the three pipelines raise several methodological points about representation choice, clustering stability, and the conditions under which deep clustering transfers to large scientific-abstract corpora. The following paragraphs address each of these points in turn.

- The three pipelines operate at qualitatively different abstraction levels. TF-IDF clusters at the lexical layer: groups form around shared technical vocabulary, so terms such as “fault diagnosis” or “generative adversarial network” act as natural separating signals. The autoencoder pipeline clusters at the distributional semantic layer: the all-MiniLM-L6-v2 sentence encoder assigns similar vectors to abstracts that describe similar problems, regardless of the specific words used. The IDEC pipeline clusters at the joint representation-geometry layer, where the encoder and the cluster boundaries

are optimized together. The practical consequence of these levels is visible in the confusion matrices. The TF-IDF medical-image-segmentation cluster (Cluster 1) and the CNN-based medical-classification cluster (Cluster 7) overlap by 54% in the autoencoder pipeline (Figure 14), because the sentence encoder captures shared problem structure despite the lexical difference between “segmentation” and “classification”. When all three pipelines agree on a theme, that theme is confirmed across abstraction levels. When they disagree, the discrepancy shows either a terminology-only community or a concept-only community that shares no common vocabulary.

- Naive end-to-end DEC training produced degenerate partitions on this corpus. In the 8-cluster and 11-cluster configurations, several clusters collapsed to zero members within the first training epochs, while one or two clusters absorbed the majority of the 18,500 documents (Figures 9 and 10). The root cause is random centroid initialization on a structureless latent space: without a pretrained encoder, gradient descent converges toward the trivial solution of concentrating all soft assignments in one cluster. The three-phase IDEC protocol eliminated this problem and produced balanced, well-separated partitions (Figures 11 and 12). This is an empirical demonstration that the IDEC approach transfers to scientific-abstract corpora at the scale studied here: approximately 18,500 documents with 384-dimensional input embeddings. The protocol succeeds because autoencoder pretraining constructs a latent geometry that already reflects the reconstruction structure of the data, so K-means initialization places centroids in meaningful regions before joint fine-tuning begins.
- Three implications follow from these results. First, because all three pipelines use the same K-means partitioning step, differences in cluster structure are attributable to the choice of representation, not to differences in the clustering algorithm. This controlled design isolates the effect of moving from lexical to semantic to joint-optimization representations and allows the comparison to be interpreted without confounding factors. Second, the framework is not specific to object detection literature. The input to each pipeline is a set of text embeddings and the clustering proceeds without domain-specific adjustments; the same approach can be applied to any sufficiently large collection of scientific abstracts from other fields. Third, the three-phase training protocol improves reproducibility. It separates the training process into deterministic stages (autoencoder pretraining and K-means centroid extraction) before the stochastic joint fine-tuning stage. Each stage has a clear objective and a natural stopping criterion, which makes the process easier to document and to replicate across different runs and different hardware environments.

The eight canonical clusters recovered by the TF-IDF pipeline fit the findings of thematic branches by four authoritative expert-authored surveys of object detection. Zou et al. (2023) split the field into traditional feature-based detectors, region-based deep detectors (R-CNN family), single-shot deep detectors (SSD and YOLO families), transformer-based detectors, detection datasets and evaluation metrics, and speedup and efficiency techniques; our Cluster 0 (object detection and anomaly recognition in industrial and visual contexts) corresponds to the single-shot and region-based detector branches, while Cluster 3 (few-shot learning, meta-learning, and small data applications) maps onto the low-data training sub-branch covered under their speedup and efficiency discussion. Liu et al. (2020) partition the literature into detection frameworks (one-stage, two-stage, and cascade architectures), object feature representation, object proposal generation, context modelling, training strategies, and evaluation

metrics; our Cluster 7 (classification tasks, largely medical, using CNN-based architectures) corresponds to the backbone and feature-representation common to both one-stage and two-stage families, while Cluster 3 fits with their training strategies branch through transfer learning. Jiao et al. (2019) further distinguish survey categories by application domain, separating industrial inspection, medical imaging, pedestrian detection, and face detection; our Cluster 4 (fault diagnosis in engineering and industrial systems) represents an application-domain cluster of this type, and Cluster 1 (medical image segmentation) corresponds to their medical imaging application branch. Köhler et al. (2024) survey few-shot object detection and organize the field into meta-learning approaches, transfer learning and fine-tuning, prototype-based methods, attention and feature enhancement, data augmentation and synthetic sampling, benchmark datasets, and open challenges; our Cluster 3 maps precisely onto their meta-learning, transfer learning, and prototype-based categories, and Cluster 2 (synthetic data, federated learning, and privacy-preserving machine learning) fits with their data augmentation and synthetic sampling branch. Regarding Cluster 2, the mapping table in Section 3 (Results) shows that this cluster connects well to all four reference surveys by appearing under data augmentation and training strategy sub-branches, however, its full scope, including federated learning and privacy-preserving machine learning, is absent from pre-2023 object-detection taxonomies and reflects a research direction that was formed after those surveys were written. Therefore, this divergence illustrates that the machine-discovered structure is not just the repeated existing expert categorizations, but it also captures emerging research directions which appeared after the reference surveys were completed. The broad correspondence between our unsupervised clusters and expert taxonomy branches provides qualitative validation of the clustering results, complementing the quantitative evaluation of metrics reported in Section 3 (Results).

5.2. Limitations

The first limitation is the absence of expert annotation and user studies. The most direct measure of cluster validity is agreement between machine-generated groupings and the judgement of domain specialists. Such annotation was outside the scope of this study. As partial compensation, the thematic clusters produced by all three pipelines were qualitatively compared against the findings in other four independent object-detection survey papers discussed in the Discussion section, where the main cluster labels showed good correspondence with established subdomains identified by those surveys. Nevertheless, the absence of expert-labelled ground truth limits the ability to directly evaluate whether the generated clusters correspond to semantically meaningful topics of domain. The reported internal and comparison metrics, including silhouette coefficient, ARI, NMI, Davies-Bouldin index, and Calinski-Harabasz score, evaluate structural clustering properties but not expert-defined semantic correctness.

Another limitation is about evaluation of LLM-generate thematic summaries. We acknowledge the hallucination risk associated with large language models in Section 1.2. The main mitigation applied in this study was cross-model consistency: GPT-4o and GPT-o1 were applied independently to the same cluster content, and the results were semantically consistent across models, with only minor lexical variation for some clusters. Despite the quantitative

evaluation of clusters using multiple internal and comparison metrics, the thematic summaries generated were not evaluated against expert-written references. Thus, the quality and completeness of the summaries generated remain dependent on qualitative inspection but not on formal reference-based evaluation.

The study was limited by evaluating only two predefined cluster configurations ($K = 8$ and $K = 11$), selected based on interpretability and exploration analysis of the dataset. While both configurations produced meaningful thematic separations, the selected cluster counts may not represent the globally optimal partitioning. More of different values of K could potentially produce alternative topic granularities and different clustering structures.

And the last limitation that will be discussed is technical, however, we find highlighting it to be useful for the reader. Although the neural clustering methods were evaluated using repeated runs and averaged metrics, the TF-IDF baseline was executed only once. This limits the strength of direct comparisons between traditional and neural clustering approaches, since variability estimates are unavailable for the TF-IDF results. Consequently, comparisons involving TF-IDF clustering should still be interpreted with some caution, as variability estimates are available only for the neural methods and randomized initialization of K-means could potentially modify the results from this method.

6. Conclusions

This study was dedicated to challenge big amount of scientific literature in rapidly evolving fields such as computer vision AI field. Thus, there is a necessity of scalable, automated and interpretable approaches for navigating and understanding the scientific landscape. Based on the analysis of nearly 18,500 scientific articles, the principal findings of this research consist of three parts:

- Deep semantic clustering methods successfully identify high-level conceptual structures within the literature complementing the patterns obtained by the traditional lexical methods.
- The proposed three-steps protocol, consisting of autoencoder pretraining, K-means centroid initialization and Joint fine-tuning, effectively stabilizes the training of Deep Embedded Clustering models. This allowed us to solve the common issue of collapsed clusters.
- The LLM integration for generating thematic summaries of machine-generated clusters provides a scalable and effective solution for interpretation. Thus, we addressed the gap between quantitative analysis and human-level insights – large-scale text mining becomes accessible and actionable.

The proposed methodology in this study lets automate the reading and understanding of large number of scientific papers. This study provides a detailed thematic map of the object detection research field and also offers a generalizable, hybrid framework for organizing and interpreting rapidly expanding scientific domains beyond computer vision.

Disclosure statement

Authors do not have any competing financial, professional, or personal interests from other parties.

References

- Arthur, D., & Vassilvitskii, S. (2007, January 7–9). K-means++: The advantages of careful seeding. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1027–1035). Society for Industrial and Applied Mathematics.
- Asmussen, C. B., & Møller, C. (2019). Smart literature review: a practical topic modelling approach to exploratory literature review. *Journal of Big Data*, 6(1), Article 93. <https://doi.org/10.1186/s40537-019-0255-7>
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(4–5). <https://doi.org/10.7551/mitpress/1120.003.0082>
- Bradley, P. S., & Fayyad, U. M. (1998). Refining initial points for K-means clustering. In *Proceedings of the Fifteenth International Conference on Machine Learning*.
- Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509, 257–289. <https://doi.org/10.1016/j.ins.2019.09.013>
- Chen, C. (2006). CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57(3), 359–377. <https://doi.org/10.1002/asi.20317>
- de Winter, J. (2024). Can ChatGPT be used to predict citation counts, readership, and social media interaction? An exploration among 2222 scientific abstracts. *Scientometrics*, 129(4), 2469–2487. <https://doi.org/10.1007/s11192-024-04939-y>
- Delgado-Chaves, F. M., Jennings, M. J., Atalaia, A., Wolff, J., Horvath, R., Mamdouh, Z. M., Baumbach, J., & Baumbach, L. (2025). Transforming literature screening: The emerging role of large language models in systematic reviews. *Proceedings of the National Academy of Sciences of the United States of America*, 122(2), Article e2411962122. <https://doi.org/10.1073/pnas.2411962122>
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *NAACL HLT 2019 – 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies – Proceedings of the Conference* (vol. 1, pp. 4171–4186). Association for Computational Linguistics.
- Grootendorst, M. (2022). *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*. arXiv. <http://arxiv.org/abs/2203.05794>
- Guo, X., Gao, L., Liu, X., & Yin, J. (2017). Improved deep embedded clustering with local structure preservation. *IJCAI International Joint Conference on Artificial Intelligence* (pp. 1753–1759). <https://doi.org/10.24963/ijcai.2017/243>
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507. <https://doi.org/10.1126/science.1127647>
- Holst, D., Moenck, K., Koch, J., Schmedemann, O., & Schüppstuhl, T. (2025). Transparent reporting of AI in systematic literature reviews: Development of the PRISMA-trAlce checklist. *JMIR AI*, 4, Article e80247. <https://doi.org/10.2196/80247>
- Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2), Article 2. <https://doi.org/10.1145/3703155>
- Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, 7, 128837–128868. <https://doi.org/10.1109/ACCESS.2019.2939201>
- Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1), 11–21. <https://doi.org/10.1108/eb026526>
- Kalibatiene, D., & Miliuskaite, J. (2026). From manual to automated systematic review: Key attributes influencing the duration of systematic reviews in software engineering. *Computer Standards & Interfaces*, 96, Article 104073. <https://doi.org/10.1016/J.CSI.2025.104073>
- Köhler, M., Eisenbach, M., & Gross, H.-M. (2024). Few-shot object detection: A comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9), 11958–11978. <https://doi.org/10.1109/TNNLS.2023.3265051>

- Khraisha, Q., Put, S., Kappenberg, J., Warraitch, A., & Hadfield, K. (2024). Can large language models replace humans in systematic reviews? Evaluating GPT-4's efficacy in screening and extracting data from peer-reviewed and grey literature in multiple languages. *Research Synthesis Methods*, 15(4), 616–626. <https://doi.org/10.1002/jrsm.1715>
- Kitchenham, B. (2007). *Guidelines for performing systematic literature reviews in software engineering* (Technical Report, Ver. 2.3). EBSE.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105.
- Lam, M. S., Teoh, J., Landay, J., Heer, J., Bernstein, M. S. (2024). Concept induction: analyzing unstructured text with high-level concepts using LLoOM. In *Proceedings of CHI 2024*. Association for Computing Machinery. <https://doi.org/10.1145/3613904.3642830>
- Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128, 261–318. <https://doi.org/10.1007/s11263-019-01247-4>
- Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129–137. <https://doi.org/10.1109/TIT.1982.1056489>
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281–297). University of California Press.
- Marshall, I. J., & Wallace, B. C. (2019). Toward systematic review automation: A practical guide to using machine learning tools in research synthesis. *Systematic Reviews*, 8, Article 163. <https://doi.org/10.1186/s13643-019-1074-9>
- Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., & Long, J. (2018). A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6, 39501–39514. <https://doi.org/10.1109/ACCESS.2018.2855437>
- O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., & Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: A systematic review of current approaches. *Systematic Reviews*, 4, Article 5. <https://doi.org/10.1186/2046-4053-4-5>
- OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., ... Zoph, B. (2024). *GPT-4 Technical Report*. arXiv. <http://arxiv.org/abs/2303.08774>
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, Article 71. <https://doi.org/10.1136/bmj.n71>
- Petticrew, M., & Roberts, H. (2008). *Systematic reviews in the social sciences: A practical guide*. Wiley. <https://doi.org/10.1002/9780470754887>
- Pranckutė, R. (2021). Web of Science (WoS) and Scopus: The titans of bibliographic information in today's academic world. *Publications*, 9(1), Article 12. <https://doi.org/10.3390/publications9010012>
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP-IJCNLP 2019 – 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing. Proceedings of the Conference* (pp. 3982–3992). Association for Computational Linguistics. <https://doi.org/10.18653/v1/D19-1410>
- Ren, Y., Pu, J., Yang, Z., Xu, J., Li, G., Pu, X., Yu, P. S., & He, L. (2025). Deep clustering: A comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4), 5858–5878. <https://doi.org/10.1109/TNNLS.2024.3403155>
- Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)
- Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill.

- Scherbakov, D., Hubig, N., Jansari, V., Bakumenko, A., & Lenert, L. A. (2025). The emergence of large language models as tools in literature reviews: A large language model-assisted systematic review. *Journal of the American Medical Informatics Association*, 32(6), 1071–1086. <https://doi.org/10.1093/jamia/ocaf063>
- Singh, A., D'Arcy, M., Cohan, A., Downey, D., & Feldman, S. (2023). SciRepEval: A Multi-format benchmark for scientific document representations. In *EMNLP 2023 – 2023 Conference on Empirical Methods in Natural Language Processing*. (pp. 5548–5566). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.emnlp-main.338>
- van de Schoot, R., de Bruin, J., Schram, R., Zahedi, P., de Boer, J., Weijdemans, F., Kramer, B., Huijts, M., Hoogerwerf, M., Ferdinands, G., Harkema, A., Willemsen, J., Ma, Y., Fang, Q., Hindriks, S., Tummers, L., & Oberski, D. L. (2021). An open source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence*, 3, 125–133. <https://doi.org/10.1038/s42256-020-00287-7>
- van Dinter, R., Tekinerdogan, B., & Catal, C. (2021). Automation of systematic literature reviews: A systematic literature review. *Information and Software Technology*, 136, Article 106589. <https://doi.org/10.1016/j.infsof.2021.106589>
- van Eck, N. J., & Waltman, L. (2010). Software survey: VOSviewer, a computer program for bibliometric mapping. *Scientometrics*, 84, 523–538. <https://doi.org/10.1007/s11192-009-0146-3>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017, December). Attention is all you need. In *Advances in Neural Information Processing Systems*.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11, 3371–3408.
- Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020, December). MINILM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)* (pp. 5776–5788). Curran Associates Inc.
- Wei, X., Zhang, Z., Huang, H., & Zhou, Y. (2024). An overview on deep clustering. *Neurocomputing*, 590, Article 127761. <https://doi.org/10.1016/j.neucom.2024.127761>
- Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *33rd International Conference on Machine Learning, ICML 2016*. (pp. 478–487). JMLR.org.
- Xu, Q., Gu, H., & Ji, S. W. (2023). Text clustering based on pre-trained models and autoencoders. *Frontiers in Computational Neuroscience*, 17, Article 1334436. <https://doi.org/10.3389/fncom.2023.1334436>
- Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232. <https://doi.org/10.1109/TNNLS.2018.2876865>
- Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3), 257–276. <https://doi.org/10.1109/JPROC.2023.3238524>