Topic Modeling arXiv Abstracts with BERTopic

Anirudha Bhaktharahalli Subramanya[1], Anvaya Chandrika Gudibanda Sreesha[1], Ekta Mulkalwar[1], David Camacho-Fernandez[1], Joseph Comeaux[1], Runyi Yang[1]

[1] Department of Computer Science, California State University, Los Angeles
Course Instructor: Dr. Jesus Armando Beltran Verdugo
Spring 2025

Abstract :
This paper investigates the use of BERTopic, a transformer-based unsupervised learning technique, to perform topic modeling on arXiv research abstracts. BERTopic integrates BERT embeddings, UMAP for dimensionality reduction, HDBScan clustering, and c-TF-IDF for topic representation. The project aims to enhance academic research discovery by enabling semantically grouped literature insights. We describe our methodology, evaluate clustering quality, discuss limitations, and present potential directions for improving automated topic modeling systems in scientific domains.
Index Terms— Topic Modeling, BERTopic, Natural Language Processing, arXiv, Unsupervised Learning.

1 Introduction

When working with large volumes of academic research, especially in fast-evolving fields like computer science or statistics, the sheer number of available papers can be overwhelming. Researchers often spend hours navigating through search engines, reading paper titles, scanning abstracts, and following citation trails, only to find that many of the results are not directly relevant to their area of interest. This inefficiency not only slows down research progress but also limits the opportunity for discovering connections between ideas that may be spread across different subfields or publication venues.

This project began with a shared frustration: how difficult it can be to find the right papers, even when using well-known platforms like arXiv. Each member of our team had experienced the cycle of clicking through dozens of abstracts, hoping that one would lead to a genuinely useful set of related works. We wondered if there might be a better way, not just to search more effectively, but to actually understand how research papers relate to each other thematically. That question led us to explore topic modeling, and specifically BERTopic, as a potential solution.

Topic modeling is a technique that helps uncover hidden themes in large sets of textual data. Instead of reading each document individually, a good topic model can summarize the dataset by clustering documents into coherent groups and representing each group with a set of descriptive keywords. BERTopic extends this idea by incorporating pretrained transformer models like BERT to better capture the context and semantics of each document. For our use case, working with abstracts from arXiv , this context is particularly valuable. Most abstracts in the dataset are well-written, with clear structure and enough length (often over 100 words) to allow for meaningful embeddings. This results in topic clusters that are both more coherent and more relevant than those produced by traditional models like Latent Dirichlet Allocation (LDA).

We chose BERTopic for two key reasons. First, it uses an unsupervised learning approach, which is critical when labeled data is either unavailable or subjective. Creating a training set where each abstract is matched with a predefined topic would require manual labeling at scale, which is not practical. Second, BERTopic has a modular architecture that allows for interpretation and visualization, which makes it easier to analyze results and reflect on how topics relate to one another.

The core problem we aimed to solve can be stated as follows: given a large set of research paper abstracts from arXiv, automatically cluster them into interpretable topics using an unsupervised pipeline, and produce visualizations and keyword summaries that help users explore and understand those topics.

Our goal was not only to group related abstracts together, but to evaluate whether those groupings would actually make sense to someone trying to use them. We wanted to see whether a topic cluster truly felt coherent, whether the keywords reflected a meaningful subfield, and whether users could reliably find related work based on a paper of interest.

Given the size and diversity of the dataset, we split our work across several notebooks. Each notebook focused on a different arXiv subtopic, such as computer science or statistics, to allow for more manageable experimentation and analysis.

Across all subtopics, we used a consistent modeling

pipeline:

- **BERT-based sentence embeddings** were used to capture the contextual meaning of each abstract.
- **UMAP** was applied for dimensionality reduction, making clustering more efficient and interpretable.
- **HDBSCAN** was used to discover natural groupings in the reduced embedding space.
- Class-based TF-IDF (c-TF-IDF) extracted keywords that best represented each topic cluster.

This architecture gave us flexibility and produced meaningful results across most subdomains. However, we also faced a few challenges along the way:

- Some runs required high memory, especially when working with larger subtopics.
- UMAP and HDBSCAN occasionally produced unstable or inconsistent clusters, depending on parameter tuning.
- Certain subtopics, such as cs.SE, generated too many small or overlapping clusters, making the visualizations cluttered and difficult to interpret.

Despite these challenges, the modularity of BERTopic allowed us to iterate on parameters and better understand how the model behaved across different research areas.

Still, the project helped us understand how advanced topic modeling tools like BERTopic can be used not just to organize information, but to create a more intuitive and semantically rich way of exploring academic content. We believe this work points toward a broader shift in how people interact with research repositories: away from linear search, and toward thematic discovery.

2 Methodology

We followed a **structured, repeatable process** using BERTopic to extract meaningful topics from arXiv research abstracts. The same steps and parameters were applied consistently across all subtopics, and each experiment was conducted in a separate notebook. No deviation was made from this methodology. We assumed that the abstracts contain sufficient thematic information to represent the core subject of each paper. This assumption allowed us to use abstracts as standalone inputs without needing full-text processing.

2.1 Incorporated Literature:

Our approach was based on and informed by these main sources:

1. **Maarten Grootendorst's BERTopic Implementation (Kaggle)**
   `https://www.kaggle.com/code/maartengr/topic-modeling-arxiv-abstract-with-bertopic`
   This notebook directly inspired our setup. It demonstrates how to apply BERTopic to arXiv abstracts and visualize relationships between topics. We followed a similar process and compared our topic outputs and cluster distribution behavior with those demonstrated in the notebook. Grootendorst's implementation used BERTopic with UMAP and HDBSCAN to explore topic clusters in arXiv abstracts. We analyzed how his cluster maps aligned with subcategories and replicated his preprocessing steps to benchmark consistency.

2. **BERTopic Documentation:**
   `https://maartengr.github.io/BERTopic/`
   The official documentation was used to understand the internal components of BERTopic, particularly the role of c-TF-IDF in topic representation and the use of transformer-based embeddings. It also helped in exploring built-in visualization tools and understanding parameter options for HDBSCAN and UMAP.

3. **BERTopic for Topic Modeling of Hindi Short Texts: A Comparative Study**
   Mutsaddi et al. (2025):
   `https://arxiv.org/pdf/2501.03843`
   This paper focused on topic modeling on Hindi short texts with BERTopic, different embedding models, and other topic modeling techniques like LDA. We took inspiration from their pre-processing methodology which involved removing stopwords, symbols, and much more in order to help reduce noise in the data. Mutsaddi et al. also evaluated their models using cluster purity metrics, which informed our own evaluation strategy, especially in subtopics with shorter or noisier abstracts.

4. **UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction**
   McInnes, L., Healy, J., Melville, J. (2018)
   `https://arxiv.org/abs/1802.03426`
   UMAP is the dimensionality reduction algorithm used in BERTopic. This paper helped us understand the stochastic behavior of UMAP, its sensitivity to input data distribution, and why the same input could lead to slightly different results across runs. It also confirmed its suitability for downstream clustering when working with semantic embeddings.

5. **Unveiling Ruby: Insights from Stack Overflow and Developer Survey**
   Akbarpour, N., Mirza, A. S., Raoofian, E., Fard, F.,

Rodríguez-Pérez, G. (2025)
`https://arxiv.org/pdf/2503.19238`
This paper's methodology consisted of data collection, data pre-processing, extracting only relevant parts of the dataset like Ruby topics using BERTopic, and extra additional steps not related to BERTopic. They specifically used e5-mistral-7b-instruct model for the embedding layer for many reasons including the 7 billion parameters. The researchers also used k-Means instead of HDBScan because of the more accurate results. This was helpful for us to better understand that some of the default settings were not always ideal or the most accurate results.

2.2 Data Collection

We used a subset of **arXiv** metadata focusing on the **abstract field**. This subset was in the form of JSON (JavaScript Object Notation) and contained a total of 2,725,401 articles. The fields included in this dataset include **submitter**, **authors**, **title**, **category**, **abstract**, **update_date**, and more. Only abstracts from certain categories like `statistics` and `computer science` were selected, and each notebook focused on a specific subtopic (e.g., `cs.CL`, `cs.LG`, `stat.ML`).

We assumed that these subdomains were diverse enough to test **BERTopic**'s ability to generalize and segment content. No additional labels or metadata were used during training. The data was pre-collected and structured, requiring no additional filtering.

2.3 Data Preparation

The abstracts were modified from their raw form. We made all of the abstracts lowercase and removed punctuation, symbols, stopwords, and numbers. We used the Natural Language Toolkit which already contained a list of stopwords to help us filter out keywords. This allowed BERTopic to output better clusters as opposed to keeping stopwords which occasionally yielded results where most abstracts were clustered into one group.

2.4 Model Selection

We selected BERTopic as our core topic modeling framework. It includes several layers which we used exactly as documented:

- **Embedding Layer:**
  A pre-trained BERT-based model (default used in BERTopic) was applied to tokenize and vectorize each abstract into a 384-dimensional embedding. This preserved sentence-level semantics.

- **Dimensionality Reduction:**
  UMAP was used to project the high-dimensional embeddings into a lower-dimensional space. This made the structure more suitable for clustering while maintaining relationships between documents.

- **Clustering:**
  HDBSCAN was used to identify clusters based on density in the reduced space. It automatically determined the number of topics and flagged outliers as noise. This was helpful when abstracts didn't strongly belong to any cluster.

- **Topic Representation:**
  BERTopic used class-based TF-IDF (c-TF-IDF) to extract keywords for each topic. This approach treats all documents in a cluster as a single text and identifies the most informative terms specific to that group.

2.5 Training

Each notebook trained a BERTopic model on a different subtopic. Training involved running the embedding, dimensionality reduction, clustering, and topic representation steps. Since the method is unsupervised, no labeled data or topic counts were required. The same pipeline was executed across all notebooks. We monitored silhouette scores throughout runs (ranging from -0.14 to 0.07) as a measure of cluster quality. Cluster diversity was approximated by keyword uniqueness ratios.

2.6 Evaluation Techniques

We evaluated our topics by inspecting the top keywords and visualizing clusters using built-in BERTopic plots. The goal was to assess whether the documents within each cluster shared a common theme, and whether the keywords reflected that theme accurately. In some cases, we manually read a few abstracts from a cluster to validate topic coherence.

2.7 Examples

In subtopics like cs.CL and stat.ML, topics were clearly separated, and keywords made sense contextually. For instance, one topic grouped abstracts focused on language modeling and machine translation, while another centered around probabilistic inference. However, subtopics like cs.SE resulted in a large number of small clusters, leading to cluttered visualizations and lower interpretability.

2.8 Challenges and Limitations

We faced several issues during implementation:

- Larger subtopics consumed more memory and sometimes caused runtime issues on Kaggle.
- HDBSCAN occasionally generated too many small or overlapping clusters, especially in subtopics with broader themes.

- UMAP produced different cluster patterns across runs due to its stochastic nature.

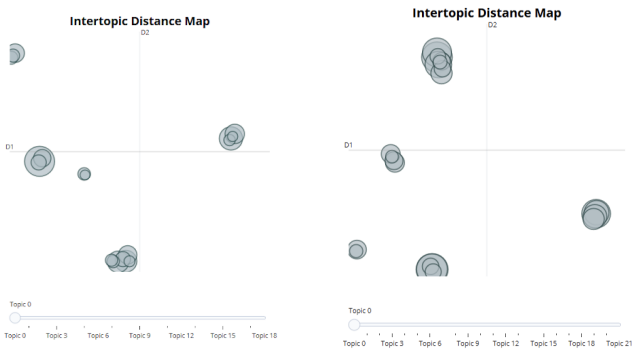

Figure 1. Comparison of two UMAP-based inter-topic distance maps on different runs. The left shows 18 topics while the right shows 21 topics. Despite using the same input data, UMAP scattered the points differently due to its stochastic nature.

This comparison demonstrates UMAP's sensitivity to initialization and sampling variance, underscoring the importance of careful interpretation when using stochastic dimensionality reduction.

- BERTopic occasionally produced results where a vast majority of the articles were grouped into one topic instead of many topics

Topic	Count	Name	Representation	Representative_Docs
0	0	973	0_data_model_models_algorithm	[data, model, models, algorithm, method, learn...	[propose novel method multiple clustering assu...
1	1	27	1_causal_variables_data_model	[causal, variables, data, model, acyclic, infe...	[many statistical methods proposed estimate ca...

Figure 2. BERTopic's `get_topic_info()` output when inputting 1000 articles from the `stat.ML` category. 973 out of 1000 were clustered into Topic 0, while only 27 were placed into Topic 1. This demonstrates BERTopic's tendency to group a large proportion of documents into a dominant topic in certain subdomains.

2.9 Expected Outcomes

We expected each subtopic to result in a manageable number of coherent topics with clear and interpretable keyword labels. For some subtopics, this expectation was met. Others, like cs.SE, did not produce high-quality topics due to either data sparsity or high intra-class variation.

All notebooks followed the same modeling architecture without deviation. These consistent conditions allowed us to isolate topic modeling performance by domain rather than pipeline variability, strengthening the validity of our comparisons across subfields.

3 Results

The BERTopic pipeline yielded varying outcomes depending on the subdomain's thematic consistency and the structural diversity of its abstracts. In certain categories like stat.ML and cs.CL, the topics produced were semantically coherent and matched expectations for that subdomain. In others, the clusters were noisy, difficult to interpret, or too fragmented to provide useful insight. These differences were closely tied to the distribution of content within each subtopic, as well as the behavior of UMAP and HDBSCAN under various conditions.

For categories such as machine learning statistics (stat.ML), the model performed particularly well. Clusters were well-separated, top keywords were meaningful, and the assigned documents shared consistent themes. Topics related to sparse, algorithms, clustering, kernels, optimisation, topics, and ensembles emerged naturally and were reflected both in the keyword labels and in the sample abstracts grouped under each cluster. This is illustrated in Figure 3, which shows the stat.ML projection with clear topic separation and minimal overlap.

Figure 3. UMAP projection of topic clusters for the stat.ML category. Each point represents an abstract, colored by assigned topic. Cluster separation is visible, with minimal overlap between dense regions.

In contrast, categories like cs.SE produced cluttered and less interpretable results. The model identified many small or overlapping clusters, and in several cases, HDBSCAN labeled a significant number of points as noise. The top keywords associated with these smaller clusters were often too generic or ambiguous, making it difficult to assign a meaningful label. This made the resulting topic visualizations dense and visually noisy, limiting their practical usefulness.

Figure 4. UMAP projection for the cs.SE category. A high number of overlapping clusters and noise points are visible, leading to visual clutter and interpretability issues. Total of 80+ topics

These inconsistencies were influenced in part by the size

and focus of each subtopic. Broader or more diverse categories led to more scattered embeddings and reduced topic coherence. We also noted that small datasets tended to result in over-fragmentation, where minor differences between abstracts led to unnecessary cluster splits.

A key limitation was high memory usage during embedding and UMAP transformation, particularly on Kaggle's limited-resource runtime. Runs with larger subtopics occasionally crashed or timed out. Larger subtopics, particularly those with several hundred abstracts, resulted in long runtimes and high memory usage. On Kaggle, these runs occasionally crashed or timed out during the embedding or dimensionality reduction steps. This constrained our ability to perform multiple parameter sweeps or rerun models with different configurations.

Additionally, UMAP's stochastic nature introduced variability across runs. Since UMAP is stochastic, the same input with the same parameters can yield slightly different projections on different runs. As a result, while the general structure of clusters remained stable, the specific shape, size, and placement of topics could shift, affecting keyword rankings and document assignments. Figure 5 shows us an example from cs.AI, where visual clutter and overlapping color assignments resulted in somewhat lacking clarity.

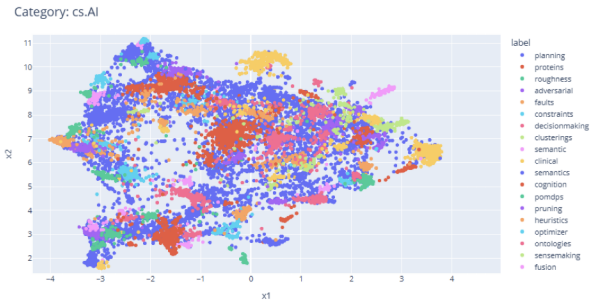

Figure 5. UMAP projection of topic clusters for the stat.ML category. Each point represents an abstract, colored by assigned topic. Cluster separation is visible, with minimal overlap between dense regions.

With cs.AI, the results produced were semi-cluttered and mostly interpretable. One issue is the colors having to be duplicated for multiple topics. For example, green was used for ai, satisfiability, graphs, and tasks. The most popular keyword used across all articles in the artificial intelligence category was planning which is represented by the color violet. This made the resulting topic visualizations less than ideal but still performed well by keeping similar topics near each other.

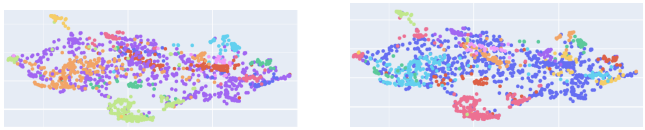

Figure 6. Comparison of two independent UMAP runs on stat.ML. Though core clusters are preserved, minor variations in shape and spacing are visible, particularly in edge regions.

Topic representation using class-based TF-IDF was effective in larger clusters with well-aligned documents. In those cases, the top keywords captured the central theme of the topic and helped differentiate it from others. However, in smaller clusters or those with semantic drift, the extracted keywords were less precise and sometimes repeated across multiple topics, reducing clarity.

Despite these limitations, the approach proved valuable in subdomains with well-defined research boundaries. When successful, the model grouped related abstracts accurately, exposed useful high-level structures in the research space, and provided interactive visual tools for further exploration.

Figure 7. Comparison of two different clustering models using cs.LG category. The left graph was produced using HDBScan and the right graph was produced using k-Means.

Upon further analysis, we realized that there is a tradeoff between choosing different sub-models for BERTopic. For example, when choosing a sub-model for the clustering portion, HDBScan will more often separate noisy data points into its own topic (e.g. Topic -1), whereas the k-Means clustering model will force those noisy data points into certain topics that are not always relevant. This highlighted a key tradeoff: HDBSCAN is better at isolating noise and organically determining the number of clusters, whereas k-Means forces assignment into fixed groups—potentially improving silhouette scores but at the cost of thematic purity.

We conducted multiple iterations using both HDBSCAN and k-Means. While we were not able to preserve all silhouette score logs due to time constraints (each evaluation required 30–60 minutes), we observed that k-Means tended to produce slightly higher average silhouette scores than HDBSCAN. For example, several HDBSCAN runs hovered around 0.01-0.05, whereas the k-means averaged around 0.03-0.07.

4 Conclusion

Working on this project gave us the opportunity to explore how topic modeling, specifically through BERTopic, can help organize and make sense of large volumes of academic writing. As we experimented with different subtopics, we began to observe how research papers naturally grouped into meaningful clusters, even without any predefined labels. This ability to uncover structure from raw text felt both powerful and rewarding.

In subtopics such as cs.CL and stat.ML, the results were clear and encouraging. The model generated distinct clusters, and the top keywords effectively summarized the key themes within each group. These outcomes made it easier to understand the direction of research in those areas and validated our expectations about what well-structured data could reveal.

At the same time, subtopics like cs.SE presented challenges. The results in these cases were less interpretable, with a high number of small or overlapping clusters and keywords that lacked specificity. Visualizations became cluttered, and it was difficult to draw strong conclusions. These cases reminded us that the quality of topic modeling depends not just on the model itself but also on the nature of the data it receives.

We also encountered technical constraints. Larger subtopics required significant memory and occasionally caused timeouts, especially when using free platforms like Kaggle. The behavior of UMAP introduced additional variability, sometimes changing the arrangement of clusters between runs. These limitations made it harder to reproduce results exactly, but they also highlighted the importance of careful evaluation and parameter control.

Despite these challenges, the overall experience was insightful. The project helped us see the strengths of unsupervised learning in action and gave us a better understanding of how tools like BERTopic can be applied to real academic data. If developed further, this approach could make it easier for researchers to identify relevant papers, understand research trends, and explore new areas more efficiently.

Our original goal was to find out whether BERTopic could offer a more intuitive way to explore scientific literature. In several subtopics, it did just that. Where it struggled, it showed us the boundaries of current methods and where more work is needed. That balance between discovery and limitation shaped the core learning from this project and made the process meaningful.

References

[1] Grootendorst, M. (2022). *Topic Modeling arXiv Abstract with BERTopic*. Kaggle. https://www.kaggle.com/code/maartengr/topic-modeling-arxiv-abstract-with-bertopic

[2] BERTopic Official Documentation. (n.d.). https://maartengr.github.io/BERTopic/

[3] McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. arXiv preprint arXiv:1802.03426. https://arxiv.org/abs/1802.03426

[4] Mutsaddi, A., Jamkhande, A., Thakre, A., & Haribhakta, Y. (2025). *BERTopic for Topic Modeling of Hindi Short Texts: A Comparative Study*. https://arxiv.org/pdf/2501.03843

[5] Akbarpour, N., Mirza, A. S., Raoofian, E., Fard, F., & Rodríguez-Pérez, G. (2025). *Unveiling Ruby: Insights from Stack Overflow and Developer Survey*. https://arxiv.org/pdf/2503.19238

[6] OpenAI. (2025). *ChatGPT (version GPT-4.0)*. Used for content organization, editing assistance, and explanation support throughout the project. https://chat.openai.com/