Data Science in Action x Deloitte
Technical Report
NLP-Powered Analysis of Academic Publications

Gabriele Goglia - Group Delegate gabriele.goglia@studenti.luiss.it
Simone Moroni moroni.simone@studenti.luiss.it
Matteo Piccirilli matteo.piccirilli@studenti.luiss.it

May 12, 2025

Company: DELOITTE

1 Introduction

Have you ever found yourself at a dead-end in your research activity, unable to find related articles and identify similar topics? This is a problem that many researchers face multiple times in their careers, with studies [1] showing that academics use over 20% of their research time just searching for relevant architecture. Such procedure is inevitably time-consuming and very inefficient; our project addresses these problems by implementing a model that exploit NLP analysis. It combines three different NLP techniques: Topic recognition using BERTopic modeling, Semantic Recommendation system using Sentence Transformers and Intelligent Text Summarization using FalconsAi model.
The data are retrieved using free APIs from Open Alex website, and they focus on the macro area of "AI for pricing and promotion for GDO in the last 10 years". Therefore we created a comprehensive dataset of academic publications that we used to train the model.
The final product consists of a interactive interface where users can enter an article title and retrieve recommended articles, topic recognised and many more options. Our solution increases research efficiency and improve discovery potential of papers that otherwise may remain hidden in the vast world of academic publications.

2 Methods

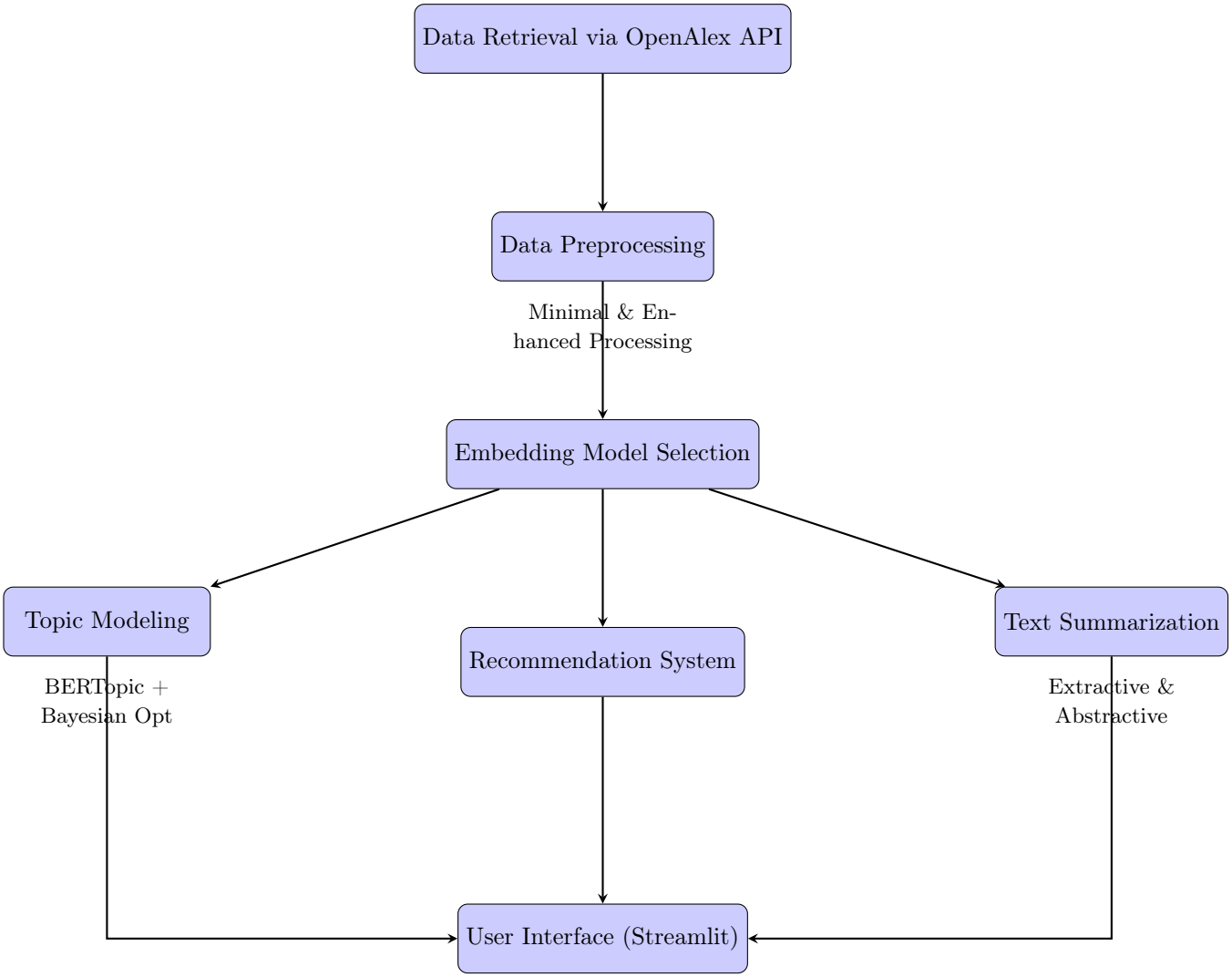

Figure 1: System architecture flow diagram for the Academic Research Analyzer.

2.1 Data Retrieval

The collection process focused on publication related to the macro area of "AI for pricing and promotion for GDO". We leveraged the OpenAlex API, filtering for articles with publication year greater than 2014. For each article we retrieved title, abstract, authors, year and DOI. We decided to settle for these parameters because

they are functional to the way we structured our project. Moreover a function has been implemented to reconstruct the abstract from the inverted index method.
To ensure model robustness and more viable topic recognition, we implemented a wider data acquisition approach. We expanded the article search by increasing the number of pages uploaded from OpenAlex (45 pages), using a descending relevance score pagination. This kind of approach allows us to retrieve all the most relevant articles (with respect to the relevant score) at first, but then as the score decreases, we gradually populate the dataset also with articles less relevant and belonging to the theme's contextual surroundings. Such approach ensures better topic diversity and more balanced distribution.

2.2 Data Preprocessing

The initial data collection process retrieved 9000 publications, which then are filtered to remove articles with missing values; almost 30% of all articles were missing their abstract, which is a pivotal parameter for our search. The final dataset consists of 6432 articles with complete data including title, abstract, authors, publication year, and DOI references.
Our analysis showed that just a single preprocessing approach would not be optimal to support all our NLP tasks. So we decided to distinguish two different strategies: the first one, that we called *Minimal Processing*, is implemented to create embeddings used in recommendation tasks and embedding models comparison. It consists in maintaining the original text structure and features but removing special characters and multiple spaces, in order to preserve more the semantic similarity and the overall meaning of the original text.
The second method, *Enhanced Preprocessing*, is a more aggressive approach that we used to create embeddings for topic modeling tasks. We perform stopword removal and lemmatization to the original text to remove all noisy words (such as I, and, to ...) that were contaminating the topic recognition process and affecting the topic's coherence.
Each approach leverages texts that are a combination of title and abstract, because most of the times titles contain highly contextual words that give an idea of the topic of the article.

2.3 Embedding Models comparison

Before implementing the 3 NLP techniques, we tested several embedding models to determine which best captures the semantic structure of scientific articles.
The compared models are four different variants of the Sentence Transformer model, which are: all-MiniLM-L6-v2, all-mpnet-base-v2, distiluse-base-multilingual-cased-v1 and paraphrase-albert-small-v2. The model selection was driven by practical and theoretical reasons. The evaluation of each embedding model was conducted implementing an hybrid evaluation approach, that evaluates both topic model performance and embedding quality on a subset of 1000 articles (which allows us to test the embedding models more quickly and on a more representative dataset.), to assess its capabilities across different NLP tasks.

1. **Topic discovery evaluation**: we generate new embeddings from highly preprocessed texts and used BERTopic model with standard parameters for UMAP and HDBSCAN (more later) to identify new topics. Then we compute some evaluation metrics as topic diversity, topic coherence and outlier percentage, in the end obtaining a BERTopic score:

   $$\text{Score}_{BERTopic} = 0.35 \cdot \text{topic diversity} + 0.20 \cdot (1 - \text{outlier percentage}) + 0.20 \cdot \text{mean coherence} + 0.25 \cdot \text{cosine coherence} \quad (1)$$

   .

2. **Intrinsic Embedding Quality**: we produce new embeddings from minimally preprocessed text to preserve linguistic structure and contextual information: Then we evaluate the model using silhouette score and Davies-Bouldin index (using k-means for clustering), obtaining a score:

   $$\text{Score}_{Intrinsic} = 0.5 \cdot \text{silhouette} + 0.5 \cdot \text{Davies Bouldin Index} \quad (2)$$

   .

For each model we generate both embeddings types and produced an overall score of:

$$\text{Score}_{Overall} = 0.7 \cdot \text{Score}_{BERTopic} + 0.3 \cdot \text{Score}_{Intrinsic} \quad (3)$$

.We assigned an higher weight to the BERTopic score because we assessed that the topic recognition system needed more prominence than the recommendation system. To avoid overestimating models that produced too few topics, a penalty (factor 0.5) was applied when fewer than five topics were discovered.
Below is shown a table that represents the final output of the function that contains all these statistics, allowing for easy comparison across multiple embedding approaches.

Metric	all-MiniLM-L6-v2	all-mpnet-base-v2	dis-base-mult-cased-v1	paraphrase-albert-small-v2
N. Topics	9	17	10	13
Outliers	0.240	0.126	0.292	0.307
Topic Diversity	0.980	0.985	0.989	0.979
Topic Coherence	0.382	0.370	0.504	0.415
Cosine Coherence	0.433	0.462	0.446	0.457
Silhouette	0.231	0.239	0.229	0.211
Davies Bouldin	1.357	1.347	1.331	1.440
$\text{Score}_{BERTopic}$	0.681	0.709	0.700	0.679
$\text{Score}_{Intrinsic}$	0.520	0.523	0.522	0.508
$\mathbf{Score}_{Overall}$	0.632	0.653	0.647	0.627

Table 1: Evaluation metrics of embedding models

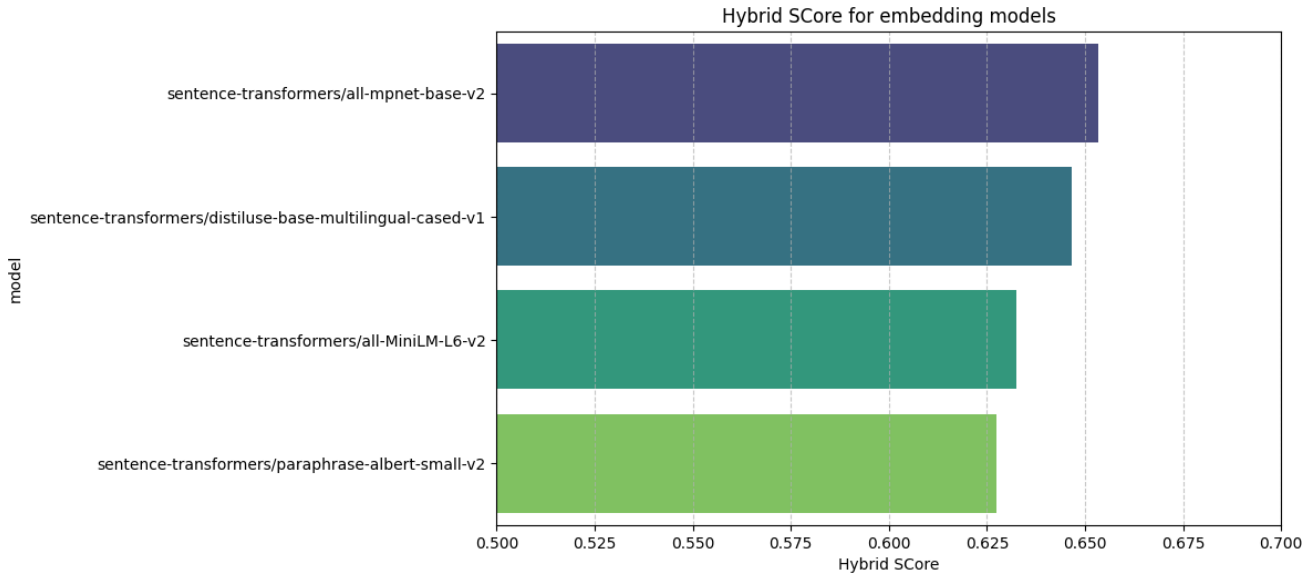

Figure 2: Embedding models Hybrid score

The four models did not show significant differences in terms of performance quality for the topic recognition task. However, the best embedding model turned out to be **sentence-transformers/all-mpnet-base-v2**, with a score of 0.653. Once the best model was identified, we used it to generate embeddings for both topic recognition and recommendation system tasks (more later).

2.4 Topic Recognition

A critical factor to achieve good topic clusters, as we experimented ourself, is to generate high quality document embeddings. We used the best embedding model computed in the previous section to generate both minimal processed embeddings and enhanced processed embeddings for the entire dataset, which then are stored. Enhanced preprocessing steps are fundamental to eliminate linguistic noise that would otherwise dominate the topic representations. Moreover storing the embeddings is a necessary procedure that enables faster experimentations without repeatedly generating embeddings.
Our initial study focused on understanding which topic model to use for the analysis; the choice was between the Latent Dirichlet Allocation (LDA) technique and the BERTopic model. The latter is a simpler version without HDBSCAN and UMAP function, used just compare with LDA; later in this section we define the complete BERt model parameters. Both methods are evaluated on a subset of 20% the total articles (for efficiency reasons) using as evaluation metrics topic coherence and topic diversity. The results of the evaluation are shown below:

Metric	BERTopic	LDA
Topic Coherence	0.516	0.433
Topic Diversity	0.991	0.909
Training Time (s)	47.21	19.87

Table 2: Evaluation metrics for BERTopic and LDA models

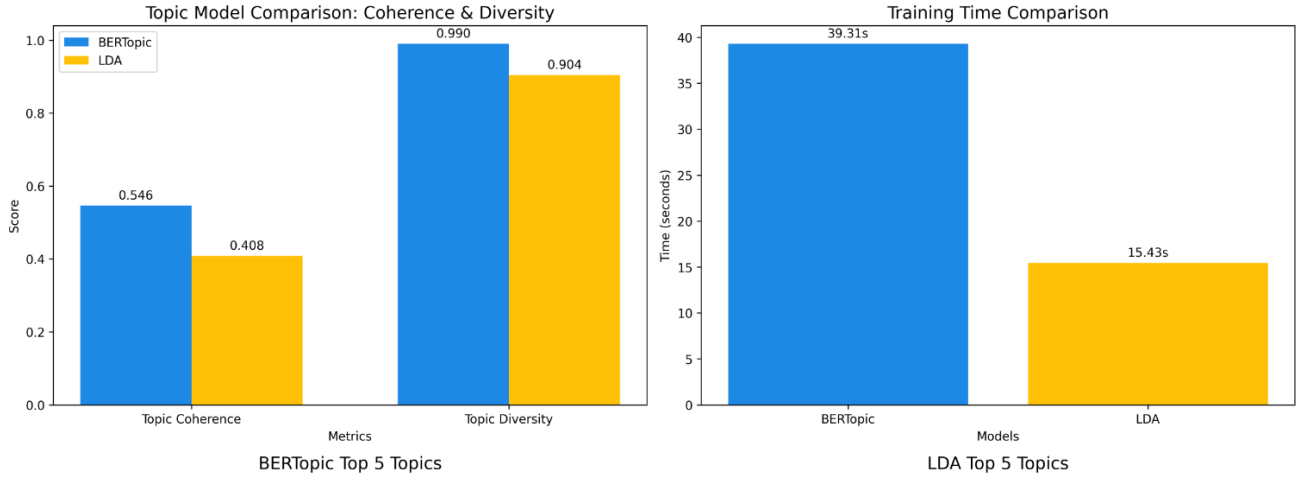

Figure 3: Comparison metrics between BERTopic and LDA and top 5 topics found

Overall BERTopic model is much better than LDA, despite its higher computational cost. This superiority can be attributed to the fact that BERTopic utilizes contextual embeddings that helps with complex sentence structures common in academic writings. Therefore we select BERTopic as our topic modeling framework.
Once assessed the best embedding model and the best topic recognition model, we can begin our topic modeling pipeline. A critical factor to achieve high quality topic modeling, is the Hyperparameter tuning phase. We compared two optimization approaches to find the best configuration: Grid Search and Bayesian Optimization. The first evaluates all the combinations on the predefined parameters grid, the latter uses a probabilistic approach to sample the parameter space. The parameters are shown in table below: We create the BERTopic model using

Parameter	Bayesian Opt	Grid Search
min topic size	[5,30]	10,20
Topic number	[10,25]	10,20
Neighbors number	[5,40]	10,30
Components number	[5,30]	10,20
min distance	[0.01,0.4]	0.1,0.3
min cluster size	[15,30]	15
min samples	[5,20]	10

Table 3: Parameters space for Bayesian Optimization and Grid Search

UMAP, that allows to transform highly dimensional vectors into a more manageable form while preserving semantic similarities; we also use HDBSCAN for clustering tasks to identify natural groupings in the transformed space.
Both approaches are evaluated over a subset of 1000 documents to decrease computational costs. The Grid Search produces 64 different iterations (evaluates all possible parameters combinations), while for the Bayesian Optimization we set the number of iteration to 30 to have a fair comparison. As already discussed, the evaluation metrics consist in topic diversity (which measures the distinctiveness between topics using Jaccard dissimilarity between topic keyword sets), mean coherence (quantifies the internal consistency of topics based on keyword relevance scores), cosine coherence (measures the semantic similarity between documents within the same topic) and outlier percentage (Coverage ; rewards models that assign more documents to meaningful topics rather than the outlier category) with a penalty factor given to configurations with less than 5 topics to discourage configu-

rations with few clusters. The final score is composed as following:

$$\text{score} = (0.30 \cdot \text{topic diversity} + 0.20 \cdot (1 - \text{outlier percentage}) + 0.25 \cdot \text{mean coherence} + 0.25 \cdot \text{cosine coherence}) \cdot \text{penalty}_{\{0.5,1\}} \tag{4}$$

Each metric is weighted accordingly to what we believed most important for the analysis. So we give high importance to coherence (two different metrics considered accounting for 50%) and to topic diversity. We decided to keep in consideration also outlier percentage to favour models with less outliers, so that new articles have higher possibilities of being assigned to an existing topic. The results are shown below:

-	Bayesian Opt	Grid Search
Best Score	0.625	0.644
Topics	8	14
Outliers (%)	24.40	24.00
Diversity	0.949	0.981
Coherence	0.390	0.390

Table 4: Evaluation metrics of Bayesian and Grid Search methods

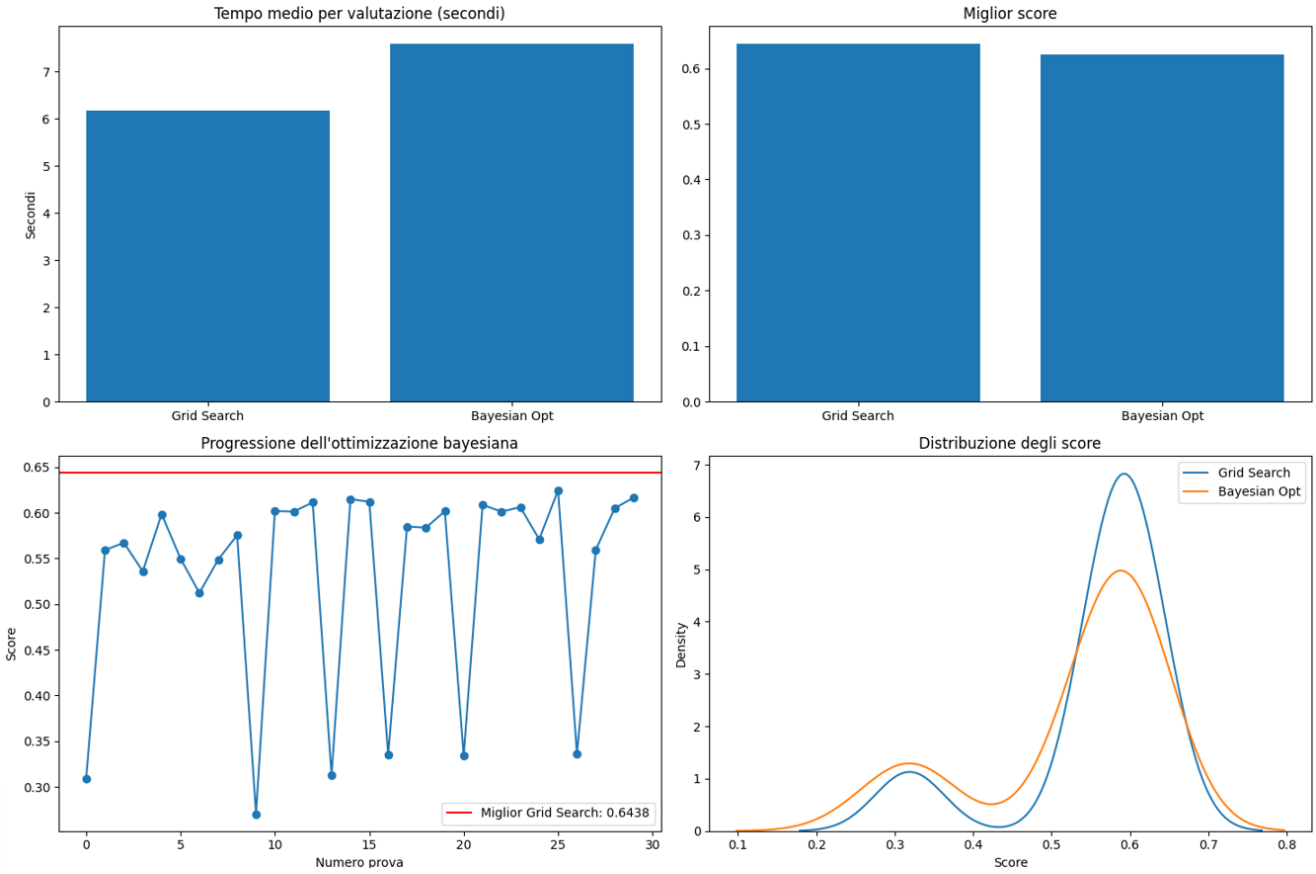

Figure 4: Results of the Bayesian Optimization and Grid Search comparison

Our comparison showed that in metrics-pure evaluation, Grid Search is slightly better. However it needs over 60 iterations over a limited discrete parameters space to achieve these results. On the other hand the Bayesian Optimization achieved similar results with just 30 iterations while also exploring a wider and continuos parameters space. So we decided to implement the second approach for its superior exploration efficiency in high-dimensional parameter spaces and also its ability to handle continuous parameters and wider ranges.
The final Bayesian Optimization implementation leverages the Optuna framework. The optimization process consists of 100 different trials on a subset of 2000 articles (for efficiency reason) using a wider and increased parameter space. The best trial (with respect to the score computed using equation (4)) results are shown below:

Parameter	Best value
min topic size	29
topics number	20
Neighbors number	5
components number	27
min distance	0.058
min cluster size	10
min samples	3

Table 5: Parameters of the best Bayesian trial

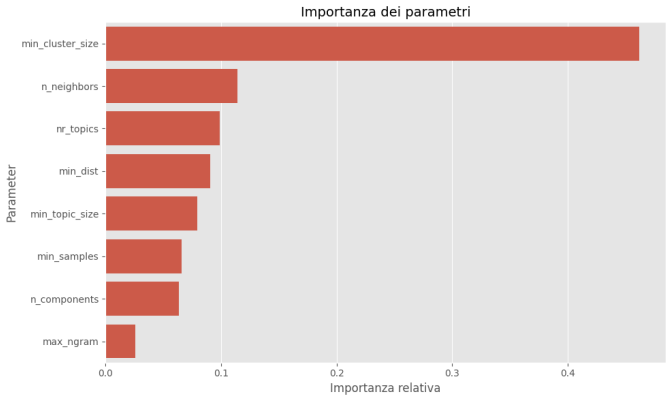

Figure 5: Bayesian most important parameters

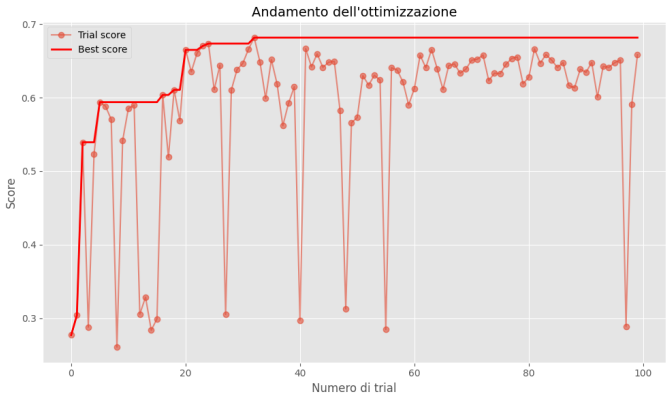

Figure 6: Optimization steps

Then we set the BERTopic model using these parameters. At first we trained it on a large batch of 4000 documents to define topics (BERT function fit-transform). For the remaining documents we applied the transform function of BERT just to assign the topics. But in the end we preferred to fit the model on the entire dataset to let the BERTopic model have a wider and more complete understanding of the dataset. We used two more metrics to evaluate the robustness of the approach: PMI-based coherence evaluation and topic stability using k-fold cross validation. The results:

Metrics	Value
Topic coherence	0.68
topics diversity	0.98
PMI coherence	1.11

Table 6: Evaluation metrics of BERTopic model

N° topic	N° Articles	Topic labels
1	1578	Business, digital, industry, ai, marketing
2	712	forecasting, price, learning, neural, stock
3	574	energy, grid, power, smart,electric
4	571	health, artificial,care, human, intelligence height

Table 7: Top 4 topics recognised and their toplabels

2.5 Recommendation system

For the design of our recommendation system, our goal was to create a tool capable of suggesting relevant scientific articles to the user based on a given input text, such as the title of an article or a generic topic.
The model leverages the Sentence Transformer embedding model to represent the semantic content of texts, and the topic modeling model to introduce a thematic layer to the recommendation process.
Let's go into more detail about how the system was developed. The embedding model is the **semantic component** of our recommendation system and it performs half the work as its function is to transform the articles (abstract and title) into vectors, creating an N-dimensional vector space capable of capturing the semantic content of scientific articles.

To measure the similarity of meaning between articles, we used **cosine similarity**, which is a measure that indicates how much two vectors point in the same direction in a multidimensional space. The higher the cosine similarity value, the more similar in meaning the compared items will be, and therefore recommendable. We considered it to be the best metric since it compares the orientation of the vectors and not their length, performing very effectively in our case where the abstracts to be compared have different lengths.
Therefore, in an initial phase, the model takes as input a query such as the title of an article, generates the embedding of its abstract (plus title), and calculates the cosine similarity between this embedding and those of all other articles within the dataset, which were previously generated.
To further improve the quality of the recommendations and make them more sensitive to the thematic context, we integrated a **topic modeling component**. Specifically, we used the previously trained BERTopic. First, the system calculates the most likely topic to which the query articles belongs by selecting the ten articles most similar to the query in terms of semantic similarity (via cosine similarity) and identifies the most frequent topic among them. This topic is assumed to be the representative topic of the query article. Once this "query topic" is determined, the system compares it with the pre-calculated topics of each article in the dataset, to assess whether they deal with similar or different themes, and based on this comparison a thematic similarity score is assigned. Then, we implemented a penalty-and-reward mechanism that favours articles thematically consistent with the query and penalizes those that deal with different topics.
Clearly, this thematic approach to measuring similarity between articles does not replace the semantic approach, which must remain the preponderant factor. In fact, these two scores are combined linearly into an aggregate score where more weight is given to the semantic component. In this way, the system is able to favour articles that are not only semantically similar to the query but also thematically consistent, improving the accuracy and practical usefulness of the recommendations for the user. We believe this hybrid approach is particularly effective in avoiding misleading recommendations, where articles that are similar in form or language actually address very different topics.
Finally, the system ranks all articles based on the combined score and returns the top 7 results to the user.
The demonstration results of the recommendation system on a sample article are shown below:

Score	**Recommended Article Title**
0.5668	IBM's smart city as techno-utopian policy mobility
0.5584	Beyond the smart city: a typology of platform urbanism
0.5444	Exposing smart cities and eco-cities: Frankenstein urbanism and the sustainability challenges of the experimental city
0.5333	Urbanism and Neoliberal Order: The Development and Redevelopment of Amman
0.5316	Future Trends and Current State of Smart City Concepts: A Survey
0.5313	The emerging data–driven Smart City and its innovative applied solutions for sustainability: the cases of London and Barcelona
0.5301	The digital skin of cities: urban theory and research in the age of the sensored and metered city, ubiquitous computing and big data

Table 8: Recommended articles for the sample article titled *The city as innovation machine.*

Each recommended article is also accompanied by a brief extractive summary, which provides a quick overview of its content. The following paragraph explains how the text summarization model was implemented.

2.6 Text Summarization

We begin by comparing two different text summarization methods on a subset of 30 articles: generative summarization leveraging FalconsAI model, which generates entirely new text to capture semantic meanings and extractive summarization using TD-IDF vectorization and cosine similarity; it basically splits the abstract into sentences, creates TD-IDF representation and computes semantic similarity. In the end it selects the highest scoring sentences maintaining logical order.
The metrics to evaluate the two models are Rouge-1, Rouge-2 and Rouge-L, all of which measure the content overlap, phrase structure, and sequence similarity between generated summaries and reference texts. The results are shown below:

Metric	Extractive	Abstractive
Rouge-1	0.629	0.490
Rouge-2	0.577	0.376
Rouge-L	0.629	0.490
Compression	0.435	0.255
Time (ms)	202.33	6328.61

Table 9: Evaluation metrics of Extractive and Abstractive models

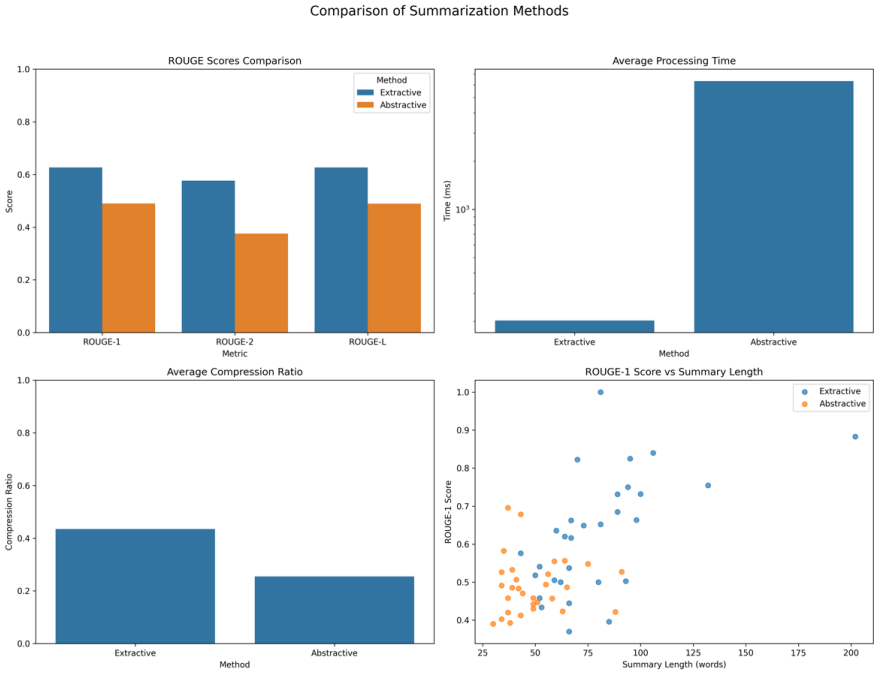

Figure 7: Comparison of summarization methods

Our quantitative evaluation reveals that extractive model achieved higher ROUGE scores while also being 60 times faster than the abstractive method. However such evaluation may be misleading, because ROUGE metrics primarily measure overlap between generated summary and reference text, naturally favouring extractive methods that reuse original phrasing. Manually evaluating some of the summaries generated, we found that abstractive approach shows better coherence and readability; however its computational cost is significantly higher to be deployed for the entire dataset.
Therefore the hybrid strategy deployed consist of producing summaries for the entire dataset using extractive model, accepting the trade-off between quality and time, while producing an abstractive summary for the article given in input, to provide the user with higher quality and readability.

2.7 User Interface

The interactive user interface represents the final product of the project, delivering the NLP techniques in a user-friendly manner. We implemented a web application using Streamlit, programmed via python codes.
The APP design consists of three main areas, each accessible through a dedicated tab :

1. **Article Analysis**: The main feature for analysing academic papers. The user can input an article's title and the system initially search through the internal dataset to seek correspondence; if negative it queries the Open Alex API to retrieve the article via the web. Once obtained and displayed both the title and the abstract, the system produces a generative summary using the model FalconsAI. Then it starts the topic recognition and displays the topic retrieved, with key topic terms and their relevance score. If an article is not recognised and ends up in the outliers group, the system still shows its most similar topic, found using max topics probabilities. For the recommendation system, it computes the embeddings of the new articles and computes the semantic similarities with the documents embeddings, retrieving the most similar.

2. **Browse Dataset**: in this section the user can independently scan into the dataset, searching for specific words in the title or even in the abstract.

3. **Topic Explorer**: here the user can discover all the topics found by BERTopic model and look more deeply into details and statistics(metrics as such as topic coherence, topic diversity, Precision@k) and interactive visualizations (heatmap and barchart).

3 Results and Discussion

3.1 Technical Results

Our implementation of an NLP-based analysis system of academic research produced some interesting results. Firstly, the BERTopic model optimized using Bayesian Optimization is able to identify valid topic clusters in our dataset, achieving a very high topic diversity score, which indicates distinct topics with minimal overlap. Moreover the topic coherence demonstrates that the topic are consistent and semantically robust. The evaluation of the topic model using k-fold cross-validation also confirmed the high stability of the model.
The recommendation system demonstrates also great performance in suggesting relevant articles, It has been evaluated using Precision@K metric, achieving a value of 0.80.
The text summarization system revealed great performance while using the extractive approach, obtaining score of Rouge-1 = 0.629, Rouge-2 = 0.577 and Rouge-L 0.629. However a more deep assessment of the summaries revealed that the generative model using FalconAI produced more coherent and readable summaries. Because of its high computational cost, we used it only for the summarization of the user query article.

3.2 Business Impact

The business value of our system is there to grasp. Our NLP-based model provides significant business value not only for research-oriented organizations. It automates the analysis of publications, addressing the main issue of time and resource efficiency, and potentially redefining the frontiers of research activities.
The topic discovery component enables users to rapidly retrieve informations about contexts, emerging trends and for organizations this means making more informed decisions regarding priorities and resource allocation. The recommendation system provides to the user the possibility to discover new relevant works that may otherwise remain hidden. The summarization component enhances productivity by allowing researchers to rapidly assess the relevance of an article just by giving a look to the summaries. Finally, the user interface integrates all these capabilities into a cohesive platform where the user can thrive, making NLP techniques accessible to everyone.

4 Conclusions

4.1 Final thoughts

This project has successfully demonstrated the potential of NLP techniques to transform academic research processes by automating literature analysis. Our integrated approach that combines topic modeling, semantic recommendation and text summarization might potentially saving researchers considerable time and expanding their awareness of relevant literature.
The key findings from our work highlight several important contributions. First, we found that BERTopic, when optimized using Bayesian Optimization provides a robust framework for identifying meaningful themes in academic literature, that outperforms traditional approaches like LDA.
Additionally, our hybrid recommendation system, wich combines semantic similarity with topic awareness provides better article suggestions than purely semantic similarity-based methods.
We also observed that while extractive summarization methods produce higher ROUGE scores, abstractive methods using transformer-based models produces higher readability and coherence, suggesting that quantitative measures alone might not be fully representative of summary quality.
Finally, we discovered that tailoring preprocessing methods for particular NLP tasks (using light-weight processing for semantic similarity and heavier processing for topic modeling) has a significant positive impact on system performance overall.

4.2 Future Improvements

Even though our system presents very good performance, there exist some restrictions and future directions of work. First, our project currently deals only with English-language publications. This considerably limits the number of articles processed, not considering so many valuable articles that are written in other languages. Allowing the system to conduct multilingual analysis would make it even more beneficial for international research communities. Second, the present topic model presupposes that every article is associated with one topic, which

may not best capture the multidisciplinary nature of most research publications. To still be able to capture multidisciplinary we preferred not to push too hard on coherence score improvement so as to have different themes (words) within each topic. However, this resulted in a slightly high outlier rate, since an article must be coherent with a larger number of themes (words within the topic) in order to be assigned a topic. Having a soft clustering mechanism where articles can be associated with multiple topics but with different weights could better reflect research landscapes. Third, while our current recommendation engine does include semantic similarity and topic coherence, it may also be helped by including additional signals like citation networks, publication channels, and co-authorship relationships. These enhancements are natural next steps for the project and could further increase its value to research organizations and scholarly researchers. Overall, our research indicates that advanced NLP techniques have the ability to significantly increase research productivity when appropriately integrated into researcher-centric interfaces that are easy to use.

References

[1] Saleh AA, Ratajeski MA, Bertolet M. Grey Literature Searching for Health Sciences Systematic Reviews: A Prospective Study of Time Spent and Resources Utilized. Evid Based Libr Inf Pract. 2014;9(3):28-50. doi: 10.18438/b8dw3k. PMID: 25914722; PMCID: PMC4405801.

5 Appendix

5.1 Appendix A: Code Description

Our implementation follows a modular architecture with distinct components for data retrieval, preprocessing, model training, and user interface development. The high-level structure of our codebase is organized as follows:
1. Data Acquisition Module:
- Implements API calls to OpenAlex for retrieving academic publications
- Handles pagination and result filtering
- Reconstructs abstracts from inverted index format
- Stores retrieved data in structured format
2. Data Preprocessing Module:
- Implements two preprocessing pipelines: Minimal Processing for recommendation system and enhanced processing for topic modeling
- Handles missing value detection and filtering
- Performs text normalization and cleaning
3. Model Training Pipeline:
- Embedding Model Selection: Compares and evaluates four Sentence Transformer variants
- Topic Modeling: Evaluates BERTopic model and LDA clustering. Evaluates Bayesian optimization and Grid Search. Implements BERTopic with Bayesian optimization for Hyperparameter tuning
- Recommendation System: Combines semantic similarity with topic awareness
- Text Summarization: Implements both extractive and abstractive approaches
4. User Interface (Streamlit App):
- Three main tabs: Article Analysis, Topic Explorer, and Browse Dataset
- Article search functionality (local dataset and OpenAlex API)
- Topic visualization with interactive charts
- Article recommendation display
- Text summarization presentation
The key algorithms implemented include:
1. BERTopic with UMAP dimensionality reduction and HDBSCAN clustering
2. Hybrid recommendation system combining cosine similarity and topic awareness
3. Extractive summarization using TF-IDF vectorization and sentence scoring
4. Abstractive summarization using FalconsAI transformer-based model
5. Bayesian hyperparameter optimization using Optuna framework
The system is designed for extensibility, allowing for easy addition of new embedding models, summarization techniques, or recommendation approaches. Error handling is implemented throughout the codebase to ensure robustness when dealing with missing data, API failures, or unexpected input formats.

5.2 Appendix B: Author Contribution

Simone Moroni : Conceptualization, Methods, Data Retrieval, Topic Modeling, Embedding Model, Streamlit Interface, Writing Report.
Gabriele Goglia: Conceptualization, Methods, Recommendation System, Embedding Model, PowerPoint presentation, Writing Report.
Matteo Piccirilli: Conceptualization, Methods, Text Summarization, Visualization, Writing Report.
This was, broadly speaking, the division of tasks. However, all members oversaw the entire project, also reviewing sections outside their individual responsibilities and making changes when agreed upon by the whole team.