CASE: Large Scale Topic Exploitation for Decision Support Systems

Lorena Calvo-Bartolomé
Universidad Carlos III
`lcalvo@pa.uc3m.es`

Jerónimo Arenas-García
Universidad Carlos III
`jarenas@ing.uc3m.es`

David Pérez-Fernández
Universidad Autónoma de Madrid
`david.perez@inv.uam.es`

Abstract

In recent years, there has been growing interest in using NLP tools for decision support systems, particularly in Science, Technology, and Innovation (STI). Among these, topic modeling has been widely used for analyzing large document collections, such as scientific articles, research projects, or patents, yet its integration into decision-making systems remains limited. This paper introduces CASE, a tool for exploiting topic information for semantic analysis of large corpora. The core of CASE is a Solr engine with a customized indexing strategy to represent information from Bayesian and Neural topic models that allow efficient topic-enriched searches. Through *ad-hoc* plug-ins, CASE enables topic inference on new texts and semantic search. We demonstrate the versatility and scalability of CASE through two use cases: the calculation of aggregated STI indicators and the implementation of a web service to help evaluate research projects.

1 Introduction

How can public administration officials efficiently manage thousands of grant proposals? How to find the most appropriate researchers for specific funding calls? Topic models, beyond their traditional uses in information retrieval and summarization (Boyd-Graber et al., 2017), can play a major role in these tasks tailored to Science, Technology, and Innovation (STI) (Zhang et al., 2016).

Despite the recent dominance of large language models (LLMs) in topic modeling research (Lam et al., 2024; Pham et al., 2024; Reuter et al., 2024), Bayesian and Neural topic models (Blei et al., 2003; Bianchi et al., 2021a; Dieng et al., 2020) remain pertinent, particularly in their interpretability and ability to preserve document-topic and word-topic distributions –which most of these new algorithms disregard. This approach tailors applications such as thematic trend analysis, topic-based document retrieval, or similarity search.

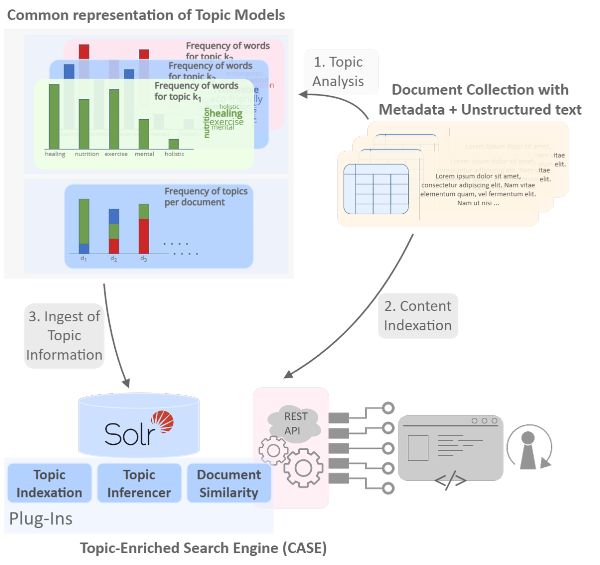

Common representation of Topic Models
Frequency of words for topic k₂
Frequency of words for topic k₂
Frequency of words for topic k₁
healing
nutrition
exercise
mental
holistic
Frequency of topics per document
1. Topic Analysis
Document Collection with Metadata + Unstructured text
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam vitae elementum quam, vel fermentum elit. Nam ut nisi ...
2. Content Indexation
3. Ingest of Topic Information
Solr
REST API
Topic Indexation
Topic Inferencer
Document Similarity
Plug-Ins
Topic-Enriched Search Engine (CASE)

Figure 1: System Overview: A Solr-powered search engine enhanced with custom plug-ins for indexing topic model, and providing real-time topic inference capabilities. A REST API provides access for web services or direct user interaction.

Though LLM-based ranking models (Khattab and Zaharia, 2020; Santhanam et al., 2022) excel in information retrieval (IR) tasks requiring deep contextual understanding, they rely on dense vectors (*embeddings*) that behave as black-box models, offering high-dimensional representations that lack interpretability for answering specific STI-related questions. Furthermore, the growing use of dedicated vector stores (e.g., FAISS, Pinecone) for efficient retrieval introduces new approaches to IR architectures, but their capabilities largely overlap with those of established systems like Lucene (Yang et al., 2018; Xian et al., 2024).

Several studies have underscored the potential of integrating topic information with Lucene (Hassan et al., 2011; George et al., 2014; Chen and Xu, 2016; Rajapaksha and Silva, 2019). There is also considerable work on developing topic modeling tools focused on training and visualization (Terragni et al., 2021; Babb et al., 2021). However, these efforts do not directly address the exploitation of topic models, particularly their integration into more complex IR systems. This gap highlights the necessity for tools that enhance the practical application of topic models within broader information systems, leveraging their interpretability and thematic analysis capabilities.

To address this, we propose CASE, a tool designed for leveraging topic models and semantic analysis in large-scale, thematically diverse datasets. CASE's core is a Solr search engine optimized for search and retrieval tasks involving large datasets. Using a novel approach for indexing topic information and integrating several plugins for topic-based similarity searches, CASE enables:

- **Searching and filtering** by document metadata and topic information;
- **Efficient aggregation** of search results with partial document-topic assignments; and
- **Semantic similarity searches** based on topic-based representations.

CASE aims to support STI managers and public administration officials by simplifying tasks such as expert identification, project proposals classification, similarity checks against previously funded projects, and alignment of researchers with relevant funding opportunities. The tool includes a RESTful API for seamless indexing and retrieval, compatible with standalone usage or a user-friendly frontend, packaged in Docker for easy deployment.

CASE is available on GitHub under an MIT license,[1] with a demo showcasing its functionalities.[2]

2 CASE

This section describes CASE's solution for indexing corpus and topic model information. It also covers its architectural and functional descriptions.

2.1 Common representation of topic models

CASE employs a Solr engine to jointly index document metadata and topic model information (see Fig. 1). It supports any topic model adhering to the common representation described in this section, ensuring compatibility with various algorithms. Let c_d $(d = 1, \ldots, D)$ denote each document in corpus C with a vocabulary size V, with $w_v,\ v = 1, \ldots V$

[1]https://github.com/Nemesis1303/CASE

[2]https://youtu.be/Iv366i9lEXc

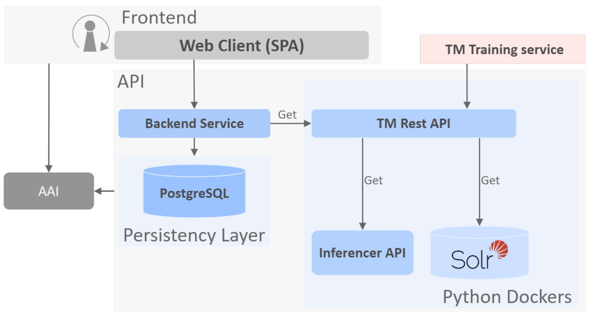


Figure 2: CASE architecture: A Python-based REST API interacts with the Solr engine, the user or frontend, and the Inferencer service. In this setup, CASE is integrated into a complete exploitation system.

being each of the words in the vocabulary. We assume that any topic model trained on C with K topics provides:

Word-to-topic assignments. Each topic t_k $(k = 1, \ldots, K)$ is characterized by an over-the-words distribution, where $\beta_{k,v}$ is the probability of observing word w_v given topic t_k:

$$\boldsymbol{\beta}_k = [\beta_{k,v}, v = 1, \ldots, V], \quad (1)$$

$$\beta_{k,v} = P(w_v \mid t_k), \quad (2)$$

Document-to-topic assignments. Each document is represented as a mixture of topics:

$$\boldsymbol{\theta}_d = [\theta_{d,k}, k = 1, \ldots, K], \quad (3)$$

$$\theta_{d,k} = P(t_k \mid c_d), \quad (4)$$

where $\boldsymbol{\theta}_d$ and $\theta_{d,k}$ are the topic-based vector and probability of topic t_k for document c_d.

For training, we use TARS,[3] a software that supports interactive, user-in-the-loop topic modeling within CASE's common representation. TARS includes algorithms such as LDA-MALLET (McCallum, 2002) (Bayesian-based), CTM (Bianchi et al., 2021c,b), and BERTopic (Grootendorst, 2022) (neural-based). Models can also be trained using other implementations and subsequently mapped to CASE's structure. We exclude newer LLM-based implementations (LLMs) (Lam et al., 2024; Pham et al., 2024; Reuter et al., 2024) because they either disregard or poorly approximate word-topic and document-topic distributions, which are essential for CASE's indexed information.

2.2 Selection of most relevant documents

Selecting the most representative documents for a given topic is crucial for the correct interpretation

[3]https://github.com/IntelCompH2020/topicmodeler

of the topics. This is usually done based on the documents with the highest θ values for the topic. However, CASE implements a newly proposed criterion:

$$S3_{d,k} = \frac{1}{n_d} \sum_{i=1}^{n_d} \beta_{k,w_{d,i}}, w_{d,i} \in \{w_v\}_{v=1}^{V}. \quad (5)$$

Here, $S3_{d,k}$ represents the average of the topic-word distribution values for all words in document d for topic k, normalized to account for document length. $S3_{d,k}$ is thus large for documents containing many words that fit well with topic k. We expect the proposed method to provide more representative documents than the θ criterion, as the latter may select documents containing words that are not very representative of the target topic (small $\beta_{k,v}$), as long as no other topic fits better these words, whereas $S3_{d,k}$ is robust to this situation.

2.3 Architecture

CASE is a multi-container application (see Fig. 2, *"Python Dockers"*) that builds on an Apache Solr search engine in *SolrCloud*[4] mode for data storage and retrieval. A Python-based RESTful API serves as intermediate between the Solr engine and the user (or frontend service) to provide a series of endpoints for indexation and exploitation of both collections and topic models. The Inferencer API computes topic representations for new texts.

2.4 Corpus and topic information indexation

SolrCloud mode utilizes *collections* of shards/cores to create logical indexes. A *Document* (the basic unit stored and indexed within a collection) contains one or more *Fields*, akin to rows and columns in a traditional database. For CASE, we define three collection types (see Fig. 3):

Corpus. Each dataset of text documents is associated with one *Corpus* collection, and contains corpus-related metadata, as well as topical document representations, and pair-to-pair similarities for each model associated to the corpus.

Model. Each topic model is stored in a different *Model* collection, representing its topics as Solr documents. Each model is associated with a *Corpus* collection (its training dataset).

[4]SolrCloud is chosen over User-Managed mode to automate shard creation for large document corpora, removing the need for manual configuration by the end user.

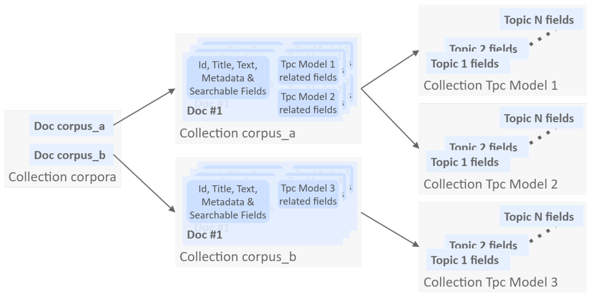


Figure 3: Organization of Solr Collections: *Corpora* contains one document per ingested corpus; each *Corpus* collection stores information on the documents text and metadata, as well as document-related topic information; each topic model requires also its own collection and is associated to just one *Corpus* collection.

Endpoint	Description
`getModelInfo`	Retrieve statistics for all topics in a model
`getBetasTopicById`	Retrieve the β_k for the indicated model and topic
`getMostCorrelatedTopics`	Retrieve the list of most similar topics (and their similarity) to a given topic
`getThetasDocById`	Retrieve the topic representation for a particular document
`inferDoc`	Infer the topic representation for a provided text fragment
`getDocsWithHighSim`	Get the most similar documents to the one provided based on topic representation
`getTopicsLabels`	Retrieve the labels for all topics in a model
`getTopicTopDocs`	Retrieve the ids of the most representative documents for a given topic

Table 1: Selection of endpoints available at the system RESTful APIs. The endpoints included in the table are associated to queries related to topic model exploitation.

Corpora. Each element of the collection represents a distinct *corpus*, including its associated models and metadata. Only one *Corpora* collection can exist per CASE deployment.

To store document-topic representations, we add a new field to each document in the corpus collection for every trained model. This field utilizes Lucene's payloads –custom binary data associated with token positions and encoded in base64– to store the representations, for which inverted indexes are automatically created. Using this document representation, along with Solr's built-in plugins for payload-based operations, enables efficient storage, filtering, and retrieval of topic scores associated with documents.

2.5 Functional description

To support information management and exploitation from Solr and connected web services, CASE includes two RESTful APIs. The first API handles basic Solr operations, including collection management (creation, indexing, deletion) and topic-based operations, such as filtering documents relevant for a topic, or aggregating results, which can be efficiently carried out benefiting from Solr inverted indexes and caching implementations.

In addition to this, there are scenarios where deriving the document-topic distribution for a user-provided document not included in the dataset is necessary. For this purpose, the second API (Inferencer API) provides endpoints using a wrapper that supports topic inference using LDA-MALLET and can be easily extended to other technologies, such as CTM or BERTopic. While this endpoint can function independently, it is primarily used as an internal server within the TM API ecosystem.

Finally, we have developed two Java plug-ins for Solr to enable semantic search based on the representation of documents in the topic space. The first plug-in allows to calculate the documents most similar to a given text, while the second plug-in exploits the pairwise distances among all indexed documents, allowing to identify pairs of documents very similar to each other. This process involves calculating the complete similarity matrix during the indexing phase, which scales quadratically with the number of documents in the collection. To address this, an approximation has been implemented to improve scalability and efficiency.

Table 1 lists selected endpoints for queries using topic model information. Appendix §A shows the complete lists of available endpoints at CASE's backend.

3 Case Studies

We demonstrate how CASE supports decision support systems in the STI domain through two real-world use cases.

3.1 Topic-enriched aggregated indicators

Analyzing STI data is crucial for policy-making, especially during the agenda-setting phase. Key data sources include scientific abstracts, research project summaries, and patents. This subsection presents the analysis of scientific production in Artificial Intelligence (Gago and Barroso, 2024) and Cancer (Levi, 2024), two living labs (LLs) from the H2020 IntelComp project.[5]

These living labs (LLs) validated IntelComp's tools, including CASE, by co-creating STI policies within their domains. Identifying policy questions is the first step toward determining STI policy needs (Markianidou et al., 2021). Using CASE, the LLs addressed analysis of the research-related datasets, providing relevant indicators for agenda setting. For example, Fig. 4 represents the evolution of the number of papers per year and topic in the Cancer domain. Based on this graph, stakeholders can identify prevailing research trends (e.g., "Transcription and Molecular Mutations"), emerging or declining topics (e.g., the decline in publications from 2009 to 2010 and after 2019, probably due to the financial crisis and the COVID-19 irruption, respectively), and potential research gaps or areas of oversaturation to make decisions about future funding or policy priorities.

Each living lab employed various datasets, including one derived from OpenAIRE,[6] which includes metadata such as publication date, authors, affiliations, and funding entities. The AI and Cancer datasets comprise $574,346$ and $2,329,760$ publications, respectively, with AI publications identified using expert-validated keywords and Cancer publications selected by domain experts.

Topic models were trained on these datasets using LDA-MALLET via TARS[3]. The number of topics (K) was set to 25 after two domain experts evaluated models with $K \in [10, 40]$ for AI and $K \in [25, 50]$ for Cancer (see §3.2). LDA was chosen over other TARS models for its superior topic quality, as judged by project experts. The final models were indexed in CASE's Solr engine, along with document metadata and topic information, enabling retrieval of aggregated indicators, such as:

I1. Number of publications per topic and year, assigning each publication to all active topics.

I2. Same as I1, but with fractional counting, i.e., each publication's contribution to its topics is weighted by $\beta_{d,k}$.

I3. Number of publications per topic containing terms *deep learning* (AI) / *metastasis* (cancer).

I4. Similar to I3, but using fractional counting.

Table 2 presents the calculation times and resource consumption (CPU and memory) for indicators I1-I4 across the AI and Cancer collections. We compare Pandas filtering as a baseline for direct computation with Solr queries to measure the advantage of Solr's built-in functionalities over

[5]https://intelcomp.eu/

[6]https://graph.openaire.eu/

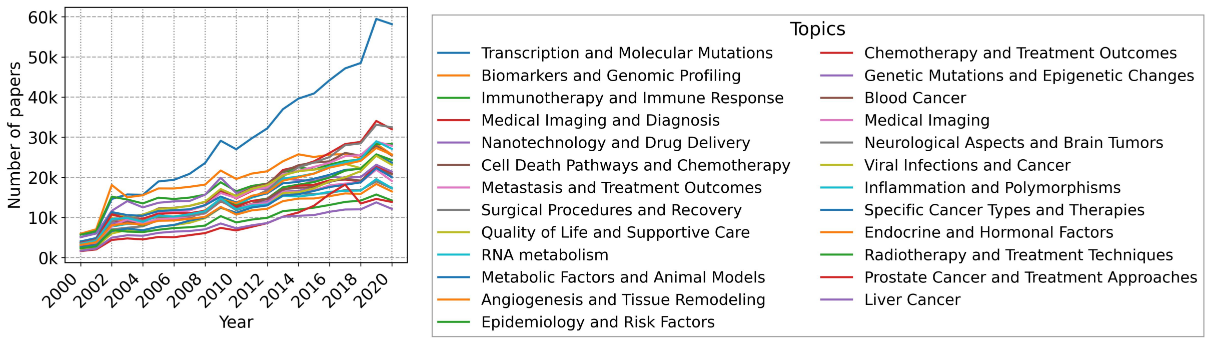


Figure 4: Evolution of publication shares across topics from 2000 to 2020 in the Cancer domain. The topic model was calculated with TARS, using LDA-MALLET for training. Labels were provided by ChatGPT and validated by Cancer LL users.

Query	**AI - Pandas**			**AI - Solr**		
	Time (s)	**CPU (%)**	**Memory (MB)**	**Time (s)**	**CPU (%)**	**Memory (MB)**
I1	0.82 ± 0.09	81.87 ± 7.25	$4\,773.7 \pm 93.1$	0.08 ± 0.015	118.95 ± 66.39	$1\,149.3 \pm 4.9$
I2	1.95 ± 0.10	97.48 ± 13.27	$5\,548.9 \pm 146.5$	0.49 ± 0.009	102.99 ± 13.94	$1\,208.4 \pm 4.7$
I3	0.75 ± 0.05	65.32 ± 5.29	$4\,404 \pm 0.00$	0.009 ± 0.001	82.04 ± 40.8	$1\,271.9 \pm 42.9$
I4	0.79 ± 0.07	79.94 ± 6.54	$4\,404 \pm 0.00$	0.013 ± 0.001	68.94 ± 13.76	$1\,308.1 \pm 3.2$
Query	**Cancer - Pandas**			**Cancer - Solr**		
	Time (s)	**CPU (%)**	**Memory (MB)**	**Time (s)**	**CPU (%)**	**Memory (MB)**
I1	2.18 ± 0.16	55.73 ± 48.11	$19\,444.8 \pm 168.6$	0.20 ± 0.013	102.4 ± 25.08	$1\,319.6 \pm 1$
I2	7.06 ± 0.22	96.29 ± 10.14	$20\,866.2 \pm 494.2$	1.35 ± 0.19	104.12 ± 7.05	$1\,324.3 \pm 2.2$
I3	1.95 ± 0.16	97.69 ± 15.32	$18\,473.66 \pm 0.00$	0.008 ± 0.001	60.22 ± 21.64	$1\,332.8 \pm 0.2$
I4	2.74 ± 0.33	99.91 ± 4.24	$18\,627 \pm 27.5$	0.02 ± 0.001	82.53 ± 14.74	$1\,337.8 \pm 4.2$

Table 2: Computation Time (s), CPU usage (%), and memory usage (MB) for indicators I1-I4 using Pandas vs. Solr-based Service across AI and Cancer datasets. Pandas statistics are derived by averaging the execution times across 1000 runs of each query, with resource consumption measured at 1-second intervals. For Solr, the calculations are based on 10 000 executions, aggregating the resource usage of all Docker containers involved in the CASE process and monitoring also computational resource consumption every second.

Python data structures. To obtain comparable measurements, all calculations were carried out on a server with 96 Intel Xeon @ 2.10GHz CPUs. For Pandas, the execution times are reported as the mean and standard deviation over 1000 runs, with CPU usage and RAM consumption monitored at 1-second intervals throughout the process. Similarly, for Solr, the same monitoring approach is used; however, the number of runs is increased to 10 000 to obtain a reliable estimate of resource consumption, considering the shorter execution times.

In all cases, Lucene's inverted indexes allow for a much faster filtering operation than Python. The results show particularly significant gains for CASE in the calculation of indicators I3 and I4, thanks to Solr's efficient term filtering capabilities. Notably, CASE demonstrates excellent scalability for these indicators, with execution times remaining virtually constant as the dataset size increases. Indeed, scalability stands out as one of the key advantages of the Solr-based service, as CPU and memory consumption remain almost unchanged when querying the AI and Cancer datasets.

When comparing Solr to the Pandas implementation in terms of resource usage, Solr exhibits slightly higher CPU consumption but significantly lower memory requirements, which appear to be mostly independent of the dataset size. Additionally, fractional counting (I2, I4) introduces a noticeable CPU overhead in Pandas, whereas Solr handles these calculations with minimal impact on CPU or memory usage. This confirms the efficiency of Solr's payload-based approach for indexing document topics.

3.2 *Ex-ante* evaluation of research projects

This use case demonstrates the integration of CASE within a complete system exploiting topic models for the *ex-ante* evaluation of research projects, the Evaluation Workbench (EWB). The service was co-designed with Hcéres[7] in the context of Intel-Comp[5]. While this study focuses on a dataset of projects funded by Hcéres, the service can be easily extended for semantic exploration and search of other text collections.

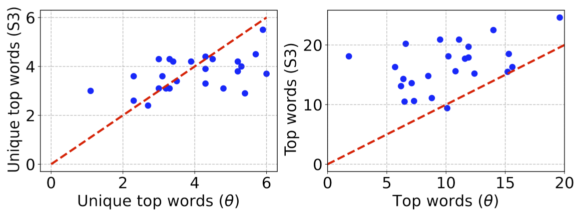
With the corpus and model information indexed in the Solr database, deploying dashboards meeting specific user requirements is straightforward (see Figs. 6-12, and also the EWB demo video[11] in Appendix §B). The general view showcases an interactive dashboard displaying various topics within the model. Top keywords and a label are displayed for each topic. Clicking on a topic provides detailed information, including top defining words, word weights within the topic, topic statistics (size, active document count, relevance, and coherence), and a list with the most representative documents for the topic. Similar actions are available in the temporal view, which is the direct result of indicator I2 described in §3.1. All these operations rely on CASE's functionalities, in particular they activate the endpoints `getModelInfo`, `getBetasTopicById`, `getTopicsLabels`.

Using the plugins discussed in §2.4, we can also create similarity-based services. For example, users can input a text fragment, and the service displays the most similar projects. This service uses the `InferDoc` endpoint to calculate the topic representation of the text and the `getDocsWithHighSim` plugin for searching similar documents.

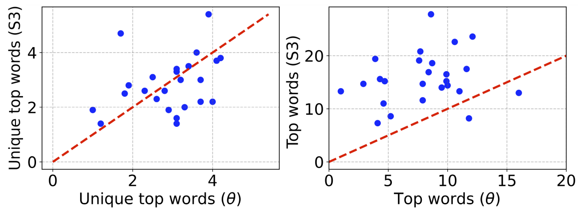
We conclude this subsection with an analysis of the performance of the document selection criterion S3 (see §2.2). Fig. 5 illustrates the performance comparison between the θ and $S3$ criteria for selecting documents relevant to the identified topics. For each topic, we selected 10 documents based on each criterion and calculated the average number of unique terms (left column) and total terms (right column) from the topics' top-10 relevant words that appear in the selected documents.

It can be seen that the documents retrieved using $S3$ consistently contain a similar number of unique terms as those retrieved using θ, but a significantly higher number of topic-relevant terms (as indicated by the majority of topics lying above the diagonal).

[7]https://www.hceres.fr/en



(a) Artificial Intelligence



(b) Cancer

Figure 5: Average occurrences of the top 10 terms in topics for the 10 most representative documents selected according to larger $\theta_{d,k}$ vs $S3_{d,k}$. Each dot represents a different topic. Left: average number of unique terms. Right: average top-word occurrences.

In summary, $S3$ enables the identification of documents that align more closely with the top words of each topic.

4 Future work

We plan to extend CASE functionality to support additional features, such as:

Model updates. Currently, the system supports updates to collections by either adding new documents using a pre-trained model or introducing new topic models for an existing document collection. However, enabling functionality to update an already indexed topic model when adding new documents could be valuable. This would require implementing queries that facilitate the training or updating of the existing model and re-indexing the document representations accordingly. When deploying such functionality, the availability of GPUs must be considered, as training or updating neural models may become impractical without adequate computational resources.

Contextual embeddings-based similarity. Recent advances in language models and contextual embeddings provide potentially more accurate methods for calculating semantic similarity compared to the current approach in the service, which relies on the presence of topics in documents. Preliminary implementations of this functionality are already in place.

Personalized dashboards. Solr can be used as a backend to implement dashboards for navigating and filtering topic-related information and other metadata in its indexed datasets. While Solr provides efficient metadata filtering out-of-the-box, filtering by topic information requires fractional counting and specific plugins.

Natural language request processing with LLMs. With the rise of conversational systems based on LLMs, a natural extension would be to allow user queries in natural language. This approach offers a more intuitive and flexible information retrieval mechanism, ensuring the system's responses are based on controlled data and methodologies.

5 Related Work

Several tools focus on visualizing topic models. *pyLDAvis*[8] is widely used for representing topics in 2D space, allowing analysis of inter-topic distances, topic content, and relevance. The *Topic Model Visualization Engine (TMVE)*[9] facilitates document navigation and provides a corpus summary with links to individual documents. *stmBrowser* enables web-based exploration of Structural Topic Models (STM) (Roberts et al., 2019), and David Mimno's *jsLDA* offers a similar platform for LDA models. *tsLDA* builds upon *jsLDA* introducing new features like hyperparameter optimization, intuitive visualizations, and streamlined workflows. *dfr-browser*[10] integrates topics, documents, words, and metadata into a comprehensive visual field.

Despite their user-friendly designs, these tools are not easy to deploy for users without programming skills, and, apart frompyLDAvis, they are tied to specific topic modeling libraries, limiting their flexibility compared to an API-driven solution.

There is also some work on integrating topic models with search engines to enhance information retrieval. Hassan et al. 2011 applied LDA with Lucene for indexing OCR-extracted documents, representing topics as numeric fields in Solr documents, and using numeric range queries for topic vectors. George et al. 2014 used LDA and latent semantic indexing (LSI) to represent documents in a topic space, improving retrieval by finding document similarities in this space with various ranking methods, but did not integrate topic-based similarity within Lucene. Chen and Xu 2016 developed the Educational Resource Retrieval Mechanism (ERRM) using Lucene and LDA-based topic indexing, demonstrating improved retrieval performance. Rajapaksha and Silva 2019 proposed a hybrid semantic retrieval approach combining LDA, community preferences, and collaborative filtering, leveraging Lucene's payloads for topic indexing and re-ranking results based on topic content after initial Lucene retrieval.

[8]https://github.com/bmabey/pyLDAvis
[9]https://github.com/ajbc/tmv
[10]https://github.com/agoldst/dfr-browser?tab=readme-ov-file

6 Conclusion

In this paper we have presented CASE, a Solr-based system based for the joint indexing of metadata and information from topic models. CASE allows efficient implementation of functions such as search or filtering by metadata and/or topics. Available plugins provide semantic search based on text or topic representations, including the possibility of performing inference for texts provided by the user.

Two use cases in the field of STI demonstrated the system's versatility in supporting various services, providing good scalability to work with large document collections.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101004870 (IntelComp project) and from Grant TED2021-132366B-I00 funded by MICIU/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTICIU.

References

Simon Babb, Mia Celeste, Dana Harris, Ingrid Wu, Theo Bayard de Volo, Alfredo Gomez, Tatsuki Kuze, Taeyun Lee, David Mimno, and Alexandra Schofield. 2021. Introducing tslda: A workflow-oriented topic modeling tool. *WeCNLP (West Coast NLP) Summit*.

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2021a. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proc. 59th Annual Meeting of the ACL and 11th Intl. Joint conf. on NLP (Volume 2: Short Papers)*, pages 759–766.

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2021b. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th Intl. Joint*

Conf. Natural Language Processing (Volume 2: Short Papers), pages 759–766.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021c. Cross-lingual contextualized topic models with zero-shot learning. In *Proc. 16th Conf. European Chapter Association for Computational Linguistics: Main Volume*, pages 1676–1683.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Jordan Boyd-Graber, Yuening Hu, David Mimno, et al. 2017. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 11(2-3):143–296.

Xiaomei Chen and Lizhen Xu. 2016. An educational resource retrieval mechanism based on lucene and topic index. In *2016 13th Web Information Systems and Applications Conf. (WISA)*, pages 125–128. IEEE.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Trans. ACL*, 8:439–453.

D. Gago and M. J. Barroso. 2024. Report on the artificial intelligence living lab. Zenodo.

Clint Pazhayidam George, Sahil Puri, Daisy Zhe Wang, Joseph N Wilson, and William F Hamilton. 2014. Smart electronic legal discovery via topic modeling. In *Proc. 27th Intl. Flairs Conf.*

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Ehtesham Hassan, Vikram Garg, SK Mirajul Haque, Santanu Chaudhury, and Madan Gopal. 2011. Searching OCR'ed text: An LDA based approach. In *2011 Intl. Conf. on Document Analysis and Recognition*, pages 1210–1214. IEEE.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proc. the 43rd Intl. ACM SIGIR Conf. on research and development in Information Retrieval*, pages 39–48.

Michelle S Lam, Janice Teoh, James Landay, Jeffrey Heer, and Michael S Bernstein. 2024. Concept induction: Analyzing unstructured text with high-level concepts using lloom. *arXiv preprint arXiv:2404.12259*.

T. Enock Levi. 2024. Report on the health living lab. Zenodo.

P. Markianidou, L. Tsipouri, A. Kossack, and J. Sanmartin. 2021. Report on the needs of science, technology and innovation policy in artificial intelligence, blue economy and cancer (1.1). https://doi.org/10.5281/zenodo.5704976.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. *http://mallet. cs. umass. edu*.

Chau Minh Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. 2024. Topicgpt: A prompt-based topic modeling framework. *Preprint*, arXiv:2311.01449.

Minuri Rajapaksha and Thushari Silva. 2019. Semantic information retrieval based on topic modeling and community interests mining. In *2019 Moratuwa Eng. Research Conf. (MERCon)*, pages 60–65. IEEE.

Arik Reuter, Anton Thielmann, Christoph Weisser, Sebastian Fischer, and Benjamin Säfken. 2024. Gptopic: Dynamic and interactive topic representations. *Preprint*, arXiv:2403.03628.

Margaret E Roberts, Brandon M Stewart, and Dustin Tingley. 2019. Stm: An r package for structural topic models. *Journal of statistical software*, 91:1–40.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.

Silvia Terragni, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano, and Antonio Candelieri. 2021. OCTIS: Comparing and optimizing topic models is simple! In *Proc. 16th Conf. EACL: System Demonstrations*, pages 263–270.

Jasper Xian, Tommaso Teofili, Ronak Pradeep, and Jimmy Lin. 2024. Vector search with openai embeddings: Lucene is all you need. In *Proc. 17th ACM Intl. Conf. on Web Search and Data Mining*, pages 1090–1093.

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using lucene. *Journal of Data and Information Quality (JDIQ)*, 10(4):1–20.

Yi Zhang, Guangquan Zhang, Hongshu Chen, Alan L Porter, Donghua Zhu, and Jie Lu. 2016. Topic analysis and forecasting for science, technology and innovation: Methodology with a case study focusing on big data research. *Technological forecasting and social change*, 105:179–191.

Appendix

A Complete list of available endpoints in TM RESTful API

The TM REST API is equipped with the following functionalities, as shown in Figs. 7 and 8.

Generic Solr operations. Create/delete collection, list available collections, and generic queries.

Corpora operations. Actions at the corpus level, including creating and indexing corpora, deleting and listing corpora, and managing fields for textual searches. This encompasses tasks such as searching for documents with specific strings in their titles, listing document collections (i.e., collections storing corpus information for a specific data collection), displaying metadata in a frontend (if developed) for that corpus, and listing models associated with a corpus.

Models operations. Actions at the model level, including creating and indexing models, deleting, and listing.

Queries to retrieve information from Solr.

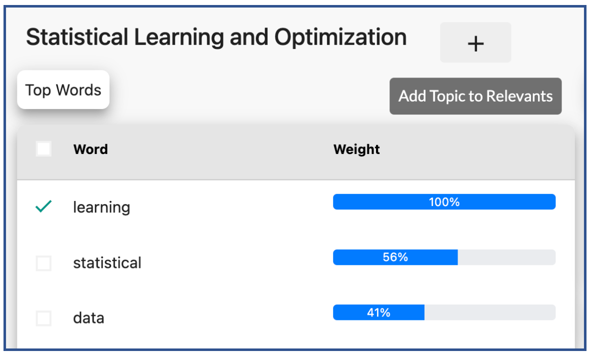
The Inferencer API provides endpoints for inference, as well as for managing the inference models created during the process (see Fig. 9).

B More on the *ex-ante* evaluation of research projects

Figs. 10a and 10b display the interactive dashboard illustrating the general and temporal evolution of the topics identified by the selected model. By clicking on any topic within these views, users can access detailed information, including top defining words, word weights within the topic, topic statistics (size, active document count, relevance, and coherence), and a list of the most representative documents for the topic.

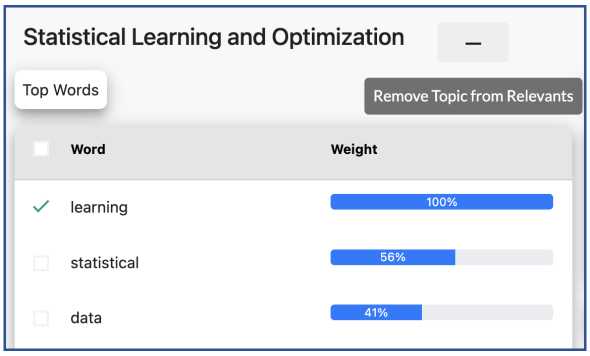
User studies revealed that users value the ability to perform an initial evaluation of topics and mark those deemed valuable for a more thorough evaluation later, as this saves time. To address this, we incorporated user-specific information into the Solr collections. Each user is assigned a unique identifier, allowing us to index relevant topics for individual users through the *Model* collection. Based on their selections, users can choose to visualize either all topics or only those they have marked as relevant (see Fig. 6), as showcased in Fig. 11 for the General View.

A video showcasing CASE's full integration with a frontend for the *ex-ante* evaluation of STI funding proposals in the H2020 IntelComp project[5] (§3.2) through the Evaluation Workbench (EWB) is also available from the project's YouTube channel.[11]



	Word	Weight
✓	learning	100%
	statistical	56%
	data	41%

(a) Add to relevant topics

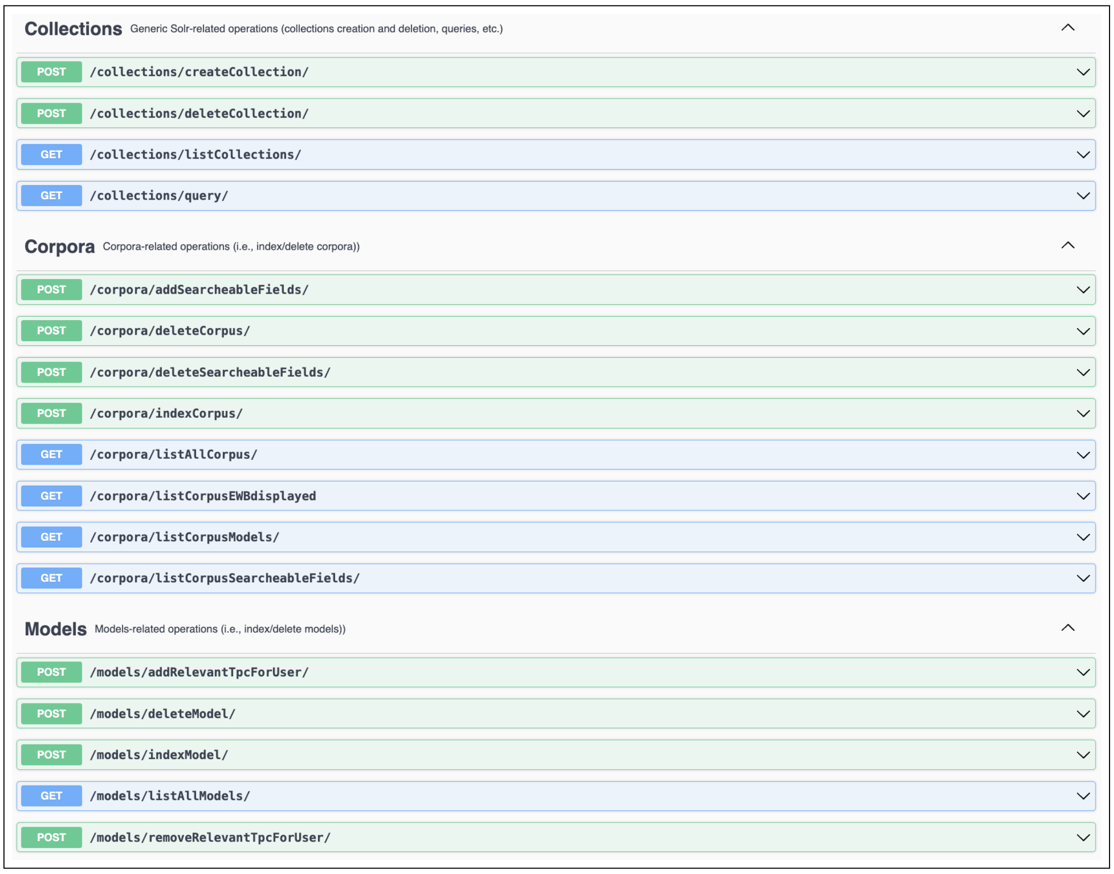

Statistical Learning and Optimization
–
Top Words
Remove Topic from Relevants

	Word	Weight
✓	learning	100%
	statistical	56%
	data	41%

(b) Remove from relevant topics

Figure 6: Users can manage the relevance of topics from the detailed topic information view.

[11]https://www.youtube.com/watch?v=wIjwIsrJmFo&ab_channel=FECYTciencia


Collections Generic Solr-related operations (collections creation and deletion, queries, etc.)

POST /collections/createCollection/
POST /collections/deleteCollection/
GET /collections/listCollections/
GET /collections/query/

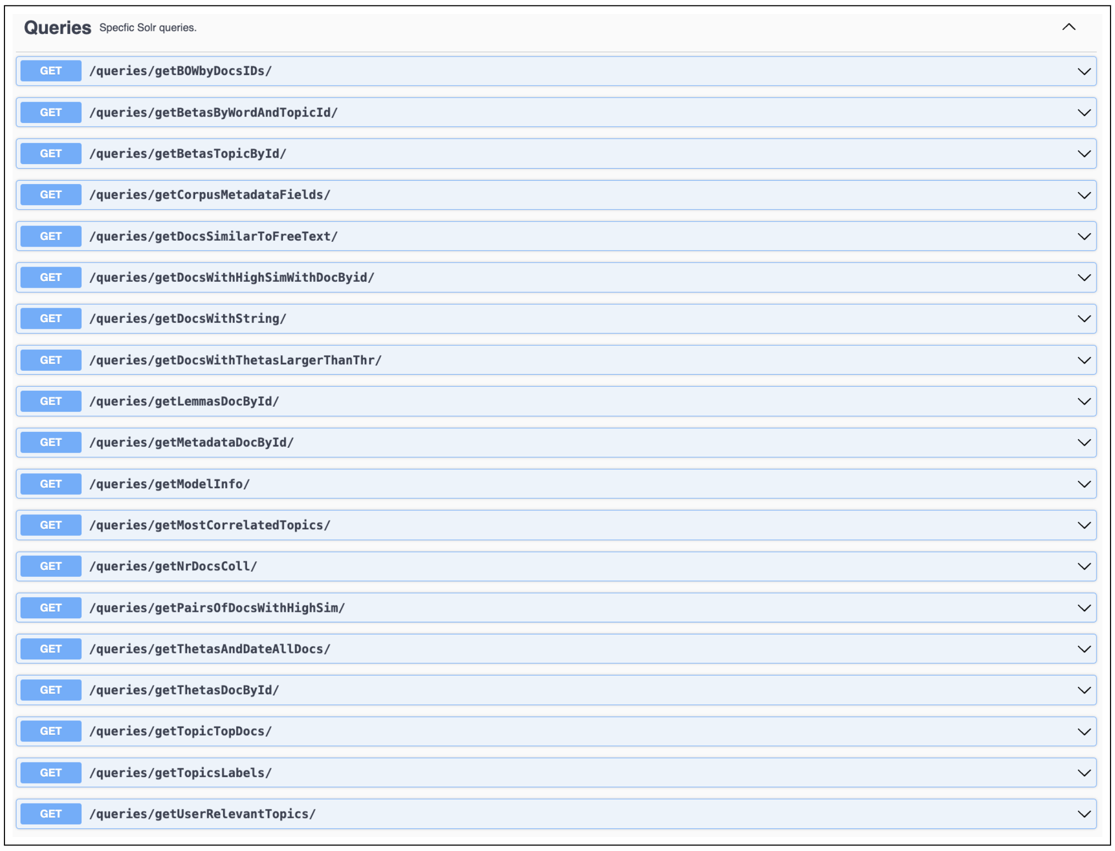
Corpora Corpora-related operations (i.e., index/delete corpora))

POST /corpora/addSearcheableFields/
POST /corpora/deleteCorpus/
POST /corpora/deleteSearcheableFields/
POST /corpora/indexCorpus/
GET /corpora/listAllCorpus/
GET /corpora/listCorpusEWBdisplayed
GET /corpora/listCorpusModels/
GET /corpora/listCorpusSearcheableFields/

Models Models-related operations (i.e., index/delete models))

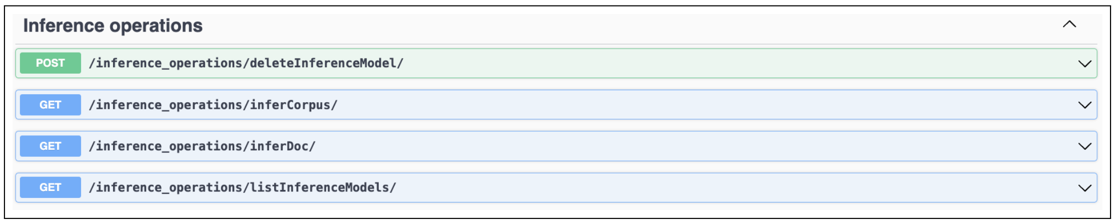
POST /models/addRelevantTpcForUser/
POST /models/deleteModel/
POST /models/indexModel/
GET /models/listAllModels/
POST /models/removeRelevantTpcForUser/

Figure 7: Complete list of endpoints used for managing generic Solr collections, along with specific endpoints for handling our *Corpus* and *Model* collections, as displayed on the Swagger interface.


Queries Specfic Solr queries.

GET /queries/getBOWbyDocsIDs/
GET /queries/getBetasByWordAndTopicId/
GET /queries/getBetasTopicById/
GET /queries/getCorpusMetadataFields/
GET /queries/getDocsSimilarToFreeText/
GET /queries/getDocsWithHighSimWithDocByid/
GET /queries/getDocsWithString/
GET /queries/getDocsWithThetasLargerThanThr/
GET /queries/getLemmasDocById/
GET /queries/getMetadataDocById/
GET /queries/getModelInfo/
GET /queries/getMostCorrelatedTopics/
GET /queries/getNrDocsColl/
GET /queries/getPairsOfDocsWithHighSim/
GET /queries/getThetasAndDateAllDocs/
GET /queries/getThetasDocById/
GET /queries/getTopicTopDocs/
GET /queries/getTopicsLabels/
GET /queries/getUserRelevantTopics/

Figure 8: List of query endpoints for retrieving information from the Solr collections (Swagger interface).



Inference operations

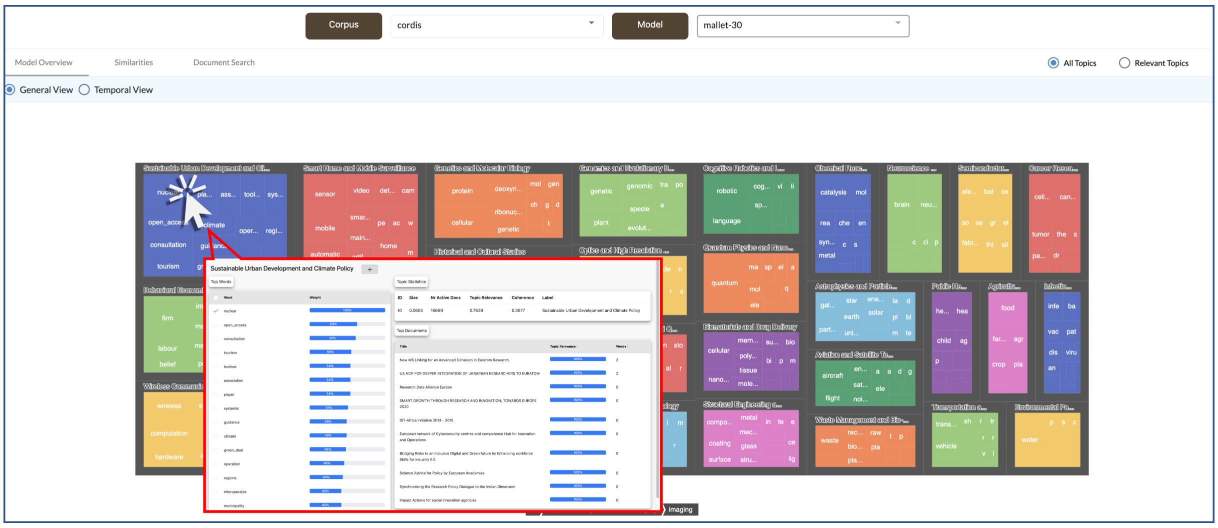
POST /inference_operations/deleteInferenceModel/

GET /inference_operations/inferCorpus/

GET /inference_operations/inferDoc/

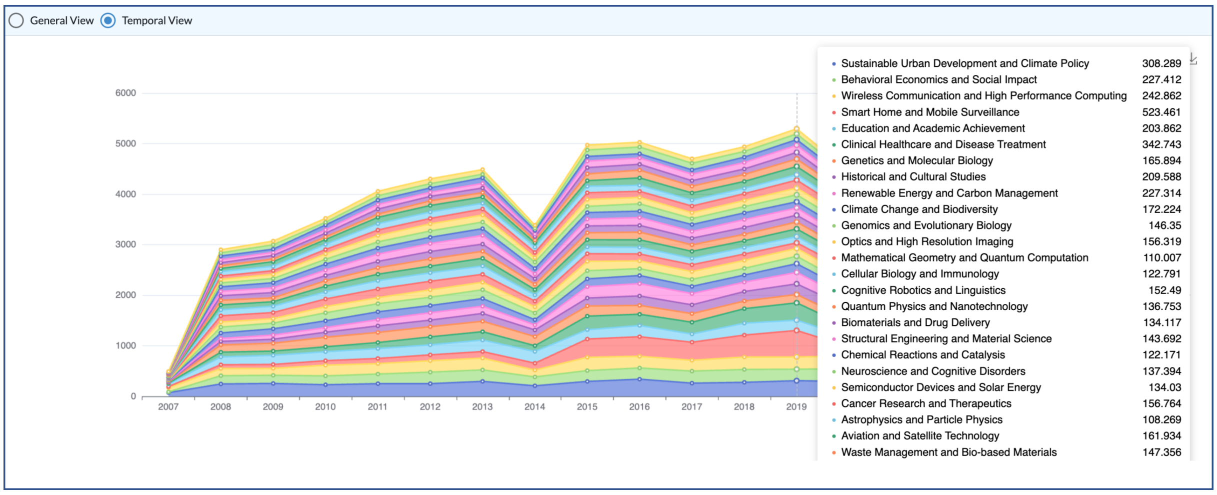
GET /inference_operations/listInferenceModels/

Figure 9: List of endpoints available in the Inference API, as displayed on the Swagger interface.



ID	Size	Nr Active Docs	Topic Relevance	Coherence	Label
0	0.0655	16699	0.7639	0.3577	Sustainable Urban Development and Climate Policy

(a) General View



Sustainable Urban Development and Climate Policy	308.289
Behavioral Economics and Social Impact	227.412
Wireless Communication and High Performance Computing	242.862
Smart Home and Mobile Surveillance	523.461
Education and Academic Achievement	203.862
Clinical Healthcare and Disease Treatment	342.743
Genetics and Molecular Biology	165.894
Historical and Cultural Studies	209.588
Renewable Energy and Carbon Management	227.314
Climate Change and Biodiversity	172.224
Genomics and Evolutionary Biology	146.35
Optics and High Resolution Imaging	156.319
Mathematical Geometry and Quantum Computation	110.007
Cellular Biology and Immunology	122.791
Cognitive Robotics and Linguistics	152.49
Quantum Physics and Nanotechnology	136.753
Biomaterials and Drug Delivery	134.117
Structural Engineering and Material Science	143.692
Chemical Reactions and Catalysis	122.171
Neuroscience and Cognitive Disorders	137.394
Semiconductor Devices and Solar Energy	134.03
Cancer Research and Therapeutics	156.764
Astrophysics and Particle Physics	108.269
Aviation and Satellite Technology	161.934
Waste Management and Bio-based Materials	147.356

(b) Temporal View

Figure 10: Dashboards offering an overview of the topics in the HFRI collection. Each topic is depicted by a differently colored square, with detailed information accessible by clicking on it. (a) General overview. Detailed topic information can be accessed by clicking on it. The pop-up window shows topic statistics, the top defining words, and its most representative projects. (b) Temporal evolution of the topics in the collection.

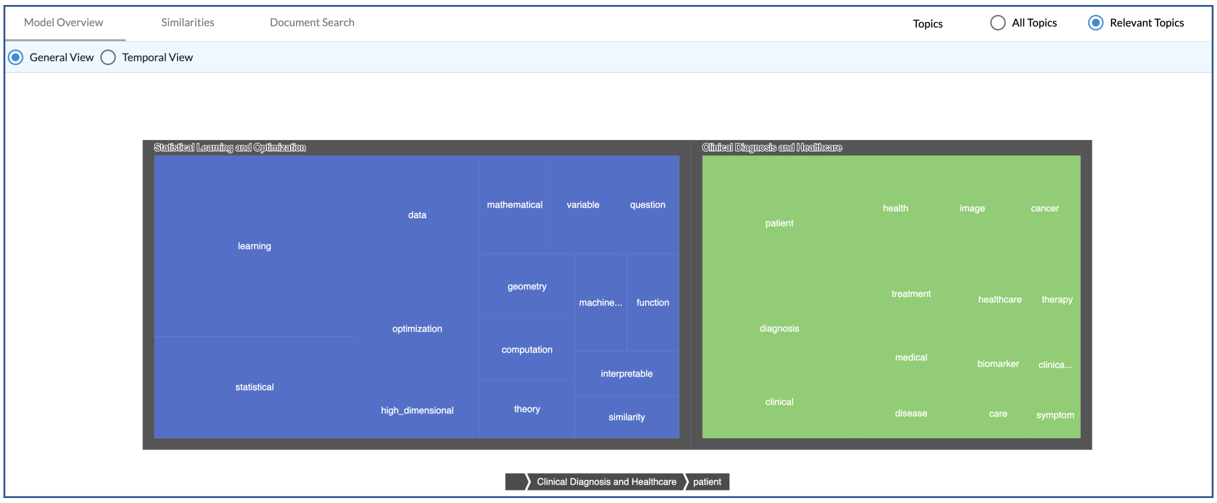
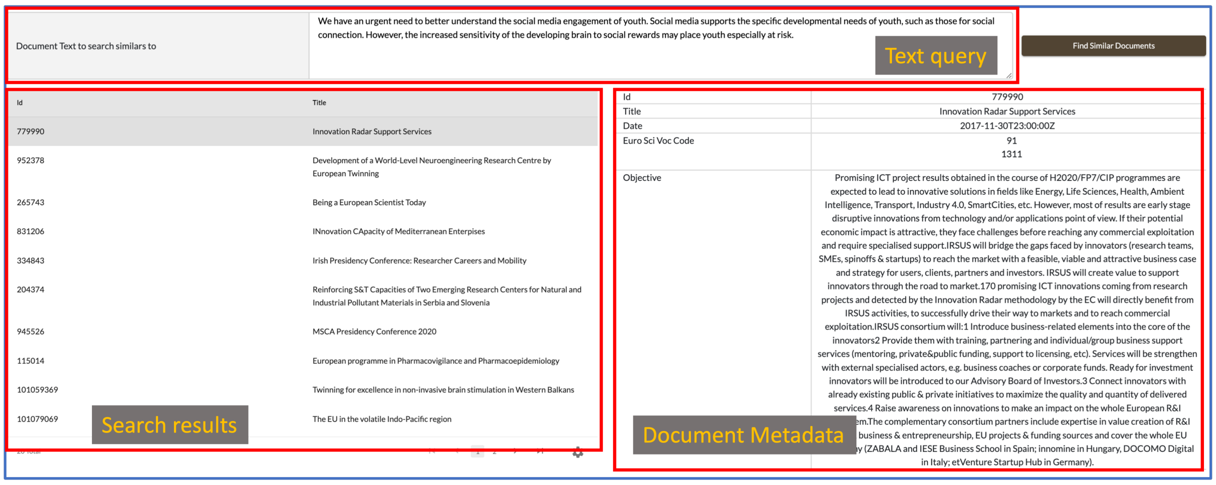


Figure 11: Dashboard providing an overview of user-relevant topics within a specified collection. All functionalities from the "All Topics" view (see Fig. 10a) are available.



Document Text to search similars to: We have an urgent need to better understand the social media engagement of youth. Social media supports the specific developmental needs of youth, such as those for social connection. However, the increased sensitivity of the developing brain to social rewards may place youth especially at risk.

Id	Title
779990	Innovation Radar Support Services
952378	Development of a World-Level Neuroengineering Research Centre by European Twinning
265743	Being a European Scientist Today
831206	INnovation CApacity of Mediterranean Enterpises
334843	Irish Presidency Conference: Researcher Careers and Mobility
204374	Reinforcing S&T Capacities of Two Emerging Research Centers for Natural and Industrial Pollutant Materials in Serbia and Slovenia
945526	MSCA Presidency Conference 2020
115014	European programme in Pharmacovigilance and Pharmacoepidemiology
101059369	Twinning for excellence in non-invasive brain stimulation in Western Balkans
101079069	The EU in the volatile Indo-Pacific region

Id	779990
Title	Innovation Radar Support Services
Date	2017-11-30T23:00:00Z
Euro Sci Voc Code	91 1311
Objective	Promising ICT project results obtained in the course of H2020/FP7/CIP programmes are expected to lead to innovative solutions in fields like Energy, Life Sciences, Health, Ambient Intelligence, Transport, Industry 4.0, SmartCities, etc. However, most of results are early stage disruptive innovations from technology and/or applications point of view. If their potential economic impact is attractive, they face challenges before reaching any commercial exploitation and require specialised support.IRSUS will bridge the gaps faced by innovators (research teams, SMEs, spinoffs & startups) to reach the market with a feasible, viable and attractive business case and strategy for users, clients, partners and investors. IRSUS will create value to support innovators through the road to market.170 promising ICT innovations coming from research projects and detected by the Innovation Radar methodology by the EC will directly benefit from IRSUS activities, to successfully drive their way to markets and to reach commercial exploitation.IRSUS consortium will:1 Introduce business-related elements into the core of the innovators2 Provide them with training, partnering and individual/group business support services (mentoring, private&public funding, support to licensing, etc). Services will be strengthen with external specialised actors, e.g. business coaches or corporate funds. Ready for investment innovators will be introduced to our Advisory Board of Investors.3 Connect innovators with already existing public & private initiatives to maximize the quality and quantity of delivered services.4 Raise awareness on innovations to make an impact on the whole European R&I em.The complementary consortium partners include expertise in value creation of R&I business & entrepreneurship, EU projects & funding sources and cover the whole EU y (ZABALA and IESE Business School in Spain; innomine in Hungary, DOCOMO Digital in Italy; etVenture Startup Hub in Germany).

Figure 12: Service to retrieve documents indexed in Solr similar to a user-inputted document based on a trained topic model. Users can then inspect the specific metadata of the retrieved documents.