electronics

MDPI

Article

Exploring and Visualizing Multilingual Cultural Heritage Data Using Multi-Layer Semantic Graphs and Transformers

Isabella Gagliardi * and Maria Teresa Artese

IMATI-MI National Research Council, Via A. Corti 12, 20133 Milan, Italy; teresa@mi.imati.cnr.it

* Correspondence: gagliardi@mi.imati.cnr.it

Abstract: The effectiveness of archives, particularly those related to cultural heritage, depends on their accessibility and navigability. An intuitive interface is essential for improving accessibility and inclusivity, enabling users with diverse backgrounds and expertise to interact with archival content effortlessly. This paper introduces a new method for visualizing and navigating dataset information through the creation of semantic graphs. By leveraging pre-trained large language models, this approach groups data and generates semantic graphs. The development of multi-layer maps facilitates deep exploration of datasets, and the capability to handle multilingual datasets makes it ideal for archives containing documents in various languages. These features combine to create a user-friendly tool adaptable to various contexts, offering even non-expert users a new way to interact with and navigate the data. This enhances their overall experience, promoting a greater understanding and appreciation of the content. The paper presents experiments conducted on diverse datasets across different languages and topics employing various algorithms and methods. It provides a thorough discussion of the results obtained from these experiments.

Keywords: multi-layer semantic graph; QueryLab intangible cultural heritage portal; clustering; data visualization; transformers; large language models; unsupervised pipeline; intuitive interactions

check for updates

Citation: Gagliardi, I.; Artese, M.T. Exploring and Visualizing Multilingual Cultural Heritage Data Using Multi-Layer Semantic Graphs and Transformers. *Electronics* **2024**, *13*, 3741. https://doi.org/10.3390/electronics13183741

Academic Editor: Domenico Ursino

Received: 19 July 2024
Revised: 5 September 2024
Accepted: 18 September 2024
Published: 20 September 2024

1. Introduction

In the current digital age, cultural heritage institutions are increasingly digitizing their collections to increase public accessibility, a trend accelerated by the pandemic. However, the effectiveness of these digital archives depends on their usability and navigability. Users need to find the information they need efficiently, without having to navigate through complex menus and options.

An intuitive user interface greatly influences how users interact with digital archives. A logical hierarchy of information and user-friendly design can make navigating the archive a seamless and engaging experience. In addition, simplicity enhances inclusivity and accessibility by accommodating users with varying levels of digital literacy and cognitive abilities, as well as those with disabilities such as visual impairments or limited mobility.

This paper presents an innovative method for visualizing and navigating information in cultural heritage archives and datasets. The proposed approach employs pre-trained language models to group data and create semantic graphs. The creation of multi-layer maps enables deep exploration of archives with large datasets, while the ability to handle multilingual datasets makes it suitable for archives with documents in various languages. A semantic graph visually represents words, items, or concepts and their interrelationships, facilitating the exploration of semantic similarities and connections between related concepts.

Key features of this approach that improve information retrieval and user engagement include the following:

- An unsupervised pipeline that works efficiently once hyperparameters and transformation models are optimized.

- Layered semantic graphs for large archives that allow users to explore a manageable number of items at a time, encouraging in-depth exploration.
- Applicability to small archives, using pre-trained linguistic models for datasets as small as a few hundred items.
- Multilingual support, effectively handling archives with documents in multiple languages (e.g., English, French, Italian).

Multi-layer maps, knowledge [1,2], and semantic graphs are powerful tools often used to visualize and model complex systems. They are widely used in GIS for telecommunications, smart cities, and environmental monitoring [3]. Meanwhile, semantic graphs are commonly applied in NLP, healthcare [4], and robotics. In this paper, we explore the use of these techniques for the visualization and interaction of archives and textual datasets. To the best of our knowledge, this is the first experiment in the field of cultural heritage.

The paper aims to define a simple-yet-effective method that can be easily applied to various cultural heritage archives and datasets, yielding satisfactory results without the need for sophisticated techniques to adapt to different cases. To achieve this, we have set the following objectives:

1. Design the pipeline to handle datasets of different sizes and in different languages, from small collections to large-scale archives, efficiently.
2. Ensure that the method can be easily integrated with existing digital tools and platforms used in cultural heritage management.
3. Develop the method in a way that allows for easy replication of results across different datasets and archives.
4. Create an intuitive and accessible interface that can be easily utilized by users with varying levels of technical expertise.

The pipeline has been tested with diverse datasets encompassing various languages and topics employing a range of algorithms and methodologies. The paper discusses the results of these experiments, accompanied by a comprehensive discussion of the results obtained.

The paper is organized as follows: First, there is a brief review of related work on graph usage in cultural heritage and a detailed description of the pipeline with technical insights, followed by experimental results showing clustering and semantic graph results. The paper concludes with a discussion of the results and future research directions.

2. Related Works

2.1. Semantic Graph Creation

In this paper, we introduce a method for unsupervised semantic graph creation, where the graphs are constructed based on the semantic similarity of their nodes. Knowledge graphs have been extensively studied and discussed in the literature. For instance, the journal *Heritage* published a special issue in 2021 on “Knowledge Graphs for Cultural Heritage”.

Significant research has been conducted on the use of linked open data and ontologies for knowledge graph (KG) creation. Ryen et al. [5] delve into the creation and publication of knowledge graphs within the Semantic Web domain. Another notable example is the study by Arco [6], which facilitates the construction of knowledge graphs based on linked open data (LOD).

The Semantic Web for Cultural Heritage (SW4CH) project and its associated workshops [7] seek to leverage Semantic Web technologies to provide access to cultural heritage data, including the development of ontologies, vocabularies, and tools for publishing and querying these data.

In [8], the authors propose a new approach for indirect access to heterogeneous data sources to simplify the creation of knowledge graphs. The approach is based on a unified meta-model as a content carrier for different representations.

Kokash et al. [9] designed a pipeline to extract structured information from bibliographies and index lists of existing scholarly publications, as well as to disambiguate and export it as linked data. They applied the pipeline to a corpus of books in the

Arts, Humanities, and Social Sciences provided by the publisher Brill to create the Brill Knowledge Graph.

In the educational context, the use of knowledge graphs or data clustering is achieving great success. For example, Jhajj [10] demonstrates the potential of large language models (LLMs) in improving the process of building EduKG, a specialized KG, particularly for course modeling.

2.2. UMAP, Clustering Algorithms, and LLMs

Recently, UMAP and clustering algorithms have been used to extract information from texts such as open-ended questions and legal documents. The effectiveness of BERT for automatic coding of open-ended questions has been investigated and compared with traditional methods [11], and integrated into a pipeline with LDA [12]. DistilBERT and UMAP have demonstrated superior performance in legal document analysis [13]. In addition, UMAP has improved clustering results on time series data [14]. LLMs and clustering algorithms have been investigated, in [15], to assess how embeddings influence clustering results, the role played by dimensionality reduction through summarization, and model size adjustment.

Recent advances in large language models (LLMs), particularly in transformer architectures, have further revolutionized semantic graph generation and knowledge representation. Models such as GPT-3 and BERT have shown remarkable capabilities in generating semantically rich embeddings that improve clustering performance by capturing deeper contextual relationships in textual data in various domains, including cultural heritage. These models utilize self-attention mechanisms to model broad dependencies, making them highly effective for tasks such as topic modeling and semantic clustering [16,17]. In addition, the application of LLMs has been particularly impactful in multilingual settings, where they significantly improve cross-lingual transfer learning and semantic similarity tasks [18]. Mixture of Experts (MoE) architectures, which dynamically activate only a subset of model parameters during inference, have been explored to further improve the efficiency of LLMs, allowing scaling to larger model sizes while maintaining computational feasibility and enhancing performance on specialized tasks [19,20].

Topic modeling has been applied in various domains, including course evaluations [21], information extraction [22], banking [23], tourism, cultural policy formulation, and crisis intervention [24]. These methods have also facilitated the study of cultural data for policy-making [25]. In addition, Twitter, with its extensive user base and real-time data stream, has become a valuable source for evaluating and comparing different topic modeling models [26]. Short texts from Twitter have served as a testbed for improved text clustering approaches [27]. To the best of our knowledge, this is one of the first experiments to create navigable semantic graphs using transformers.

3. Materials and Methods

The goal of this research is to establish a pipeline that facilitates the unsupervised creation of a semantic graph for content navigation from textual datasets. The innovative aspects of this pipeline include a set of interrelated processes and techniques that collectively enable the automated generation of meaningful relationships and connections between disparate data. These processes involve several critical steps, including dimensionality reduction, data clustering, selection of pre-trained language models, methods for defining a single vector representation per element, and keyword extraction to facilitate the understanding of topics. Figure 1 and Table 1 provide a comprehensive overview of the stages of the pipeline, which will be discussed in greater detail in the subsequent sections of this paper.

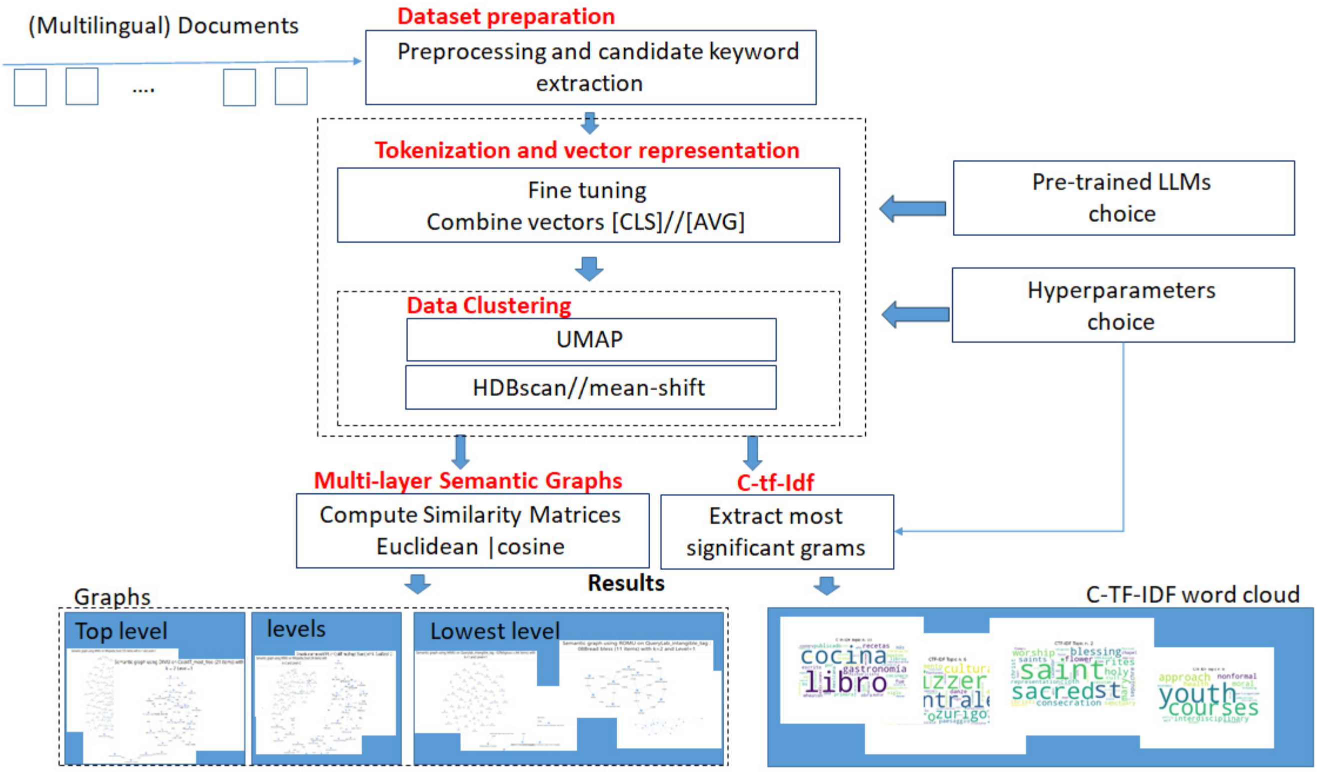


Figure 1. The proposed pipeline. The steps are highlighted in red, the solid rectangles represent actions, and the dashed boxes represent steps that produce outputs. The rectangles on the right show configuration choices, such as selecting pre-trained models (LLMs) and setting hyperparameters. The rectangles at the bottom show the final results, including graphs and a word cloud generated by the C-TF-IDF method. Some graphs generated from the pipeline can be accessed via the following URL address: http://arm.mi.imati.cnr.it/papers/Kgraphs/html/index.html, 17 Septembrer 2024.

Table 1. Steps of the proposed approach.

Task 1: Dataset preparation
- Preprocessing (possibly strip stopwords, accents, . . .) - Process data to extract items to be used - Output: items of interest
Task 2: Tokenization and vector representation
- Choice of transformers and pre-trained models - Tokenization and vector representation - Fine-tuning of pre-trained Bert-like models to obtain the vectors
Task 3: Data clustering
- Choice of hyperparameters for UMAP and HDBSCAN/Mean Shift - Iterative clustering to obtain multi-layer clustering - Evaluation of clustering results using Calinski–Harabasz Index - Output: centroids of clustered items, and elements of each cluster
Task 4: Multi-layer semantic graph and other data visualization
- Choice of transformers and pre-trained models, both on raw data and on clustered items and fine-tuning - Creation of similarity matrix using [AVG] or [CLS] tokens - Output: semantic graphs with k most similar items, with k = 1. . .4 and word cloud
Task 5: c-TF-IDF scores
- c-TF-IDF scores computation for each cluster and their visualization as word cloud - Preliminary qualitative evaluation of the results with domain experts and web users

3.1. Innovative Elements of the Pipeline

Dimensionality reduction: Advanced techniques, such as UMAP, are employed to effectively reduce the dimensionality of high-dimensional text embeddings while preserving local data structures. The use of dimensionality reduction techniques is necessary to manage the high-dimensional nature of our dataset (results of the vectorization step), to mitigate the curse of dimensionality, and to improve computational efficiency while preserving essential information.

Uniform Manifold Approximation and Projection (UMAP) [28] is a state-of-the-art dimensionality reduction technique widely used in machine learning and data analysis. Competing with PCA and t-SNE, UMAP is characterized by its ability to map high-dimensional data to lower-dimensional spaces while preserving their local structure. UMAP's effectiveness stems from its use of manifold learning, which integrates topological and geometric methods to handle nonlinear relationships and high-dimensional datasets. The algorithm needs two main parameters: the number of dimensions n and the distance metric d, typically tuned to optimize performance.

$$\text{UMAP} : \mathbb{R}^{\boldsymbol{n}} \to \mathbb{R}^{\boldsymbol{m}} \tag{1}$$

where $m \ll n$

Data clustering: Clustering algorithms such as HDBSCAN and Mean Shift are implemented to identify clusters of different shapes, densities, and sizes, facilitating the discovery of intrinsic data patterns without the need for predefined cluster numbers. Data clustering has been integrated to uncover underlying patterns and groupings within the data, facilitating a more granular understanding of the relationships between data points.

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [29] is a robust clustering algorithm that can identify clusters of different shapes, densities, and sizes. Unlike traditional clustering algorithms, HDBSCAN automatically determines the number of clusters and can identify noise points without needing a preset number of clusters or neighborhood size. Key parameters for HDBSCAN include minimum cluster size and the minimum samples to form a dense region.

$$\boldsymbol{HDBSCAN}(\boldsymbol{X}) = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_k\} \cup \{\boldsymbol{noise}\} \tag{2}$$

where X is the dataset, C_i are the clusters.

Mean Shift, another clustering technique we use, is effective at identifying clusters without specifying their number. It iteratively shifts data points toward the mean of the points within a defined bandwidth. This method is particularly effective for image segmentation and clustering high-dimensional data, handling nonlinear data shapes with minimal parameter tuning. The primary parameter for Mean Shift is the bandwidth which defines the neighborhood size. Mean Shift is also robust to outliers, making it a versatile tool for various clustering applications [30–32].

Pre-trained large language models: State-of-the-art transformer-based models such as BERT, RoBERTa, and paraphrase-multilingual-MiniLM-L12-v2 are leveraged to generate high-quality text embeddings that capture deep semantic relationships. Transformer-based models have been chosen for their outstanding ability to capture large-scale dependencies and contextual relationships within the data that are critical to the tasks at hand.

Transformers, a breakthrough in NLP, have significantly advanced the field with their self-attention mechanisms [33]. These mechanisms allow transformers to focus on different parts of the input text, efficiently capturing long-term dependencies and contextual information.

$$\boldsymbol{Self} - \boldsymbol{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{softmax}\left(\frac{\boldsymbol{QK}^{\boldsymbol{T}}}{\sqrt{\boldsymbol{d}_{\boldsymbol{k}}}}\right)\boldsymbol{V} \tag{3}$$

where Q (query), K (key), and V(value) are matrices derived from the input embeddings.

BERT (Bidirectional Encoder Representations from Transformers) [16] is the first pre-trained language model to use bidirectional language modeling to understand context by looking at the preceding and succeeding words in a sentence. Other notable models include GPT (Generative Pre-Training Transformer) 3.5 or 4 by OpenAI, which uses a similar architecture; RoBERTa (Robustly Optimized BERT Approach) by Facebook AI, trained on larger, more diverse corpora; and ALBERT (A Lite BERT), a more efficient variant from Google.

Selecting the appropriate pre-trained model depends on several factors, including the specific task, the size of the dataset, and computational resources. BERT and its variants provide robust performance for tasks that require deep contextual understanding. In our experiments, we focus on paraphrase detection and semantic similarity. The languages involved are Italian, English, Spanish, and French. For multilingual tasks, models such as paraphrase-multilingual-MiniLM-L12-v2 are preferred because of their ability to handle multiple languages with reduced computational overhead. In this paper, we test different pre-trained models as described below.

Tokenization and vector representation: Methods to split texts into tokens and compress complex textual information into single, representative vectors for each element are being developed, facilitating efficient data processing and clustering.

- Tokenization is a basic preprocessing step in natural language processing (NLP) where text is broken down into smaller units called tokens. These tokens can be words, phrases, or characters. Transformers such as BERT and GPT use specific tokenization strategies to convert raw text into a format that can be efficiently processed by the model.
- WordPiece Tokenization [16]: This method, used by models such as BERT, breaks down words into subwords or smaller units. It allows the model to handle out-of-vocabulary words by breaking them down into known subword units. For example, the word "playing" could be tokenized to ["play", "##ing"], where "##" denotes a subword prefix.
- Byte Pair Encoding (BPE) [34]: Used by models such as GPT, BPE is a data compression technique adapted for tokenization. It iteratively combines the most frequent pairs of bytes (or characters) in a corpus. The phrase "low lying" could be tokenized to ["low", "lying"], but, if "ly" is a frequent pair, it could become ["low", "ly", "ing"].
- SentencePiece: This is a language-independent tokenization method that treats the input as a sequence of Unicode characters and uses BPE or Unigram language models to tokenize the text. In the following example, "Tokenization" could be segmented into ["T", "oke", "n", "iz", "ation"].

Transformers handle multi-word phrases and sentences by embedding the tokenized text into high-dimensional vectors that capture semantic relationships.

Multi-layer semantic graph: We create semantic graphs that visualize the relationships and connections between items based on their semantic similarity, thereby improving content navigation and exploration. Based on the clustering results, we again used large language models (LLM) to construct semantic graphs, applying text vectorization to generate dense vector representations of text entities. Cosine similarity is the method used to construct a similarity matrix, which facilitates the construction of a detailed semantic graph. To handle large datasets, we implement a multi-level approach that iteratively constructs graphs at multiple levels of abstraction.

c-TF-IDF (class-based TF-IDF) score: For the terms in each cluster, the weight is calculated using the c-TF-IDF algorithm [35]. This algorithm calculates TF-IDF scores by treating all documents within a cluster as a single, combined document. This approach helps determine the importance of terms in the context of the entire cluster.

TF-IDF (Term Frequency–Inverse Document Frequency) is a formula widely used in information retrieval (IR) [36] to assess the significance of terms within a document relative to the entire dataset. It assigns higher weights to terms that are specific to a document and lower weights to terms that are common across many documents. TF-IDF consists of two main components, Term Frequency (TF) and Inverse Document Frequency (IDF). TF measures how frequently a term *t* appears within a specific document *d* and highlights the relative importance of a term within a document. IDF measures the rarity of a term across all documents in the dataset; IDF decreases the weight of terms that occur frequently across all documents and emphasizes terms that are rare but significant.

Assessing the quality of the pipeline is an essential step in research activities. In this case, two types of assessments are performed. The first is an objective evaluation that focuses on measuring the quality of the clusters using quantitative metrics. The second, still preliminary, is a subjective evaluation that measured user acceptance and satisfaction. This dual approach ensures a comprehensive understanding of the pipeline's performance from both a technical and a user perspective.

3.2. The Pipeline

Dataset preparation encompasses all activities conducted on the data before their use in the algorithms. Removing stopwords, removing accents, reducing to normal form, and other changes can be necessary at this stage. In this instance, automatically identifying tags or keywords may also be part of preprocessing. The outcomes of this activity lead to the identification of items of interest that will be used to test algorithms and pipelines.

An initial clustering phase is necessary when there are too many components to create a single semantic graph. In this work, the vectors produced by fine-tuned transformers are clustered using UMAP, HDBSCAN, or Mean Shift algorithms. Several pre-trained models are evaluated and fine-tuned on the items to determine which model is best suited for the task. Additionally, this work involves selecting the UMAP, HDBSCAN, and Mean Shift hyperparameters to obtain better results tailored to each dataset. The item clustering, including the centroids and the collection of items within each cluster at any level, is the result of the task.

For semantic graph creation, after the items are grouped in a multi-layer mode (when necessary), each individual item is again vectorized using pre-trained BERT-like transformers, fine-tuned, and then averaged (or a comparable method is performed) to produce a single vector for each item. Semantic similarity is applied to these vectors to create a similarity matrix. The k most similar items are connected to create a semantic graph based on this similarity matrix. The values of k in the experiment presented here range from 1 to 4. The cosine similarity of the resulting vectors is used to compute the similarity of the items. The output of the task is an unsupervised semantic graph. The items are grouped and n + 1 graphs are generated, i.e., the centroid graph plus the n clusters. If multi-layer graphs are required, then n + m + 1, where n is the number of clusters of the lowest level and m the number of intermediate clusters plus the centroid graph.

Using the c-TF-IDF (class-based TF-IDF) score, the next step is to analyze the clusters found by HDBSCAN/Mean Shift to extract the most important topics and words based on their frequency or weight [35]. The c-TF-IDF approach captures the importance of terms within clusters in relation to the entire corpus of text.

Evaluating the quality of clustering results is crucial to ensure that the algorithm effectively groups similar data points while separating dissimilar ones. For objective evaluation, several metrics can be used to evaluate clustering performance, each providing different insights into the clustering structure. Here, we discuss four widely used clustering evaluation metrics: the Silhouette Score, Calinski–Harabasz Index, Davies–Bouldin Index, and Dunn Index.

The Silhouette Score measures [37] how similar a data point is to its own cluster compared to other clusters. It provides a value between -1 and 1, where a higher value indicates better-defined clusters.

The Calinski–Harabasz Index [38], also known as the Variance Ratio Criterion, evaluates the ratio of the sum of between-cluster dispersion to within-cluster dispersion. Higher values indicate a model with dense and well-separated clusters.

The Davies–Bouldin Index [39] measures the average similarity ratio of each cluster with its most similar cluster. Interpretation: lower values indicate that clusters are compact and well separated from each other.

The Dunn Index [40] measures the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance. A higher value indicates better clustering performance.

Moreover, since the goal of the work is to develop tools for easier navigation in archives or textual datasets, we integrate a preliminary qualitative evaluation based on informal feedback and judgments from experts and users in a real-world context.

4. The Experimentation

4.1. Datasets Used

The pipeline was tested on several datasets related to cultural heritage. The data come from QueryLab [41,42], a portal created specifically for the management of intangible cultural heritage, CookIT [43,44], a site that collects traditional Italian recipes, and Wikipedia, with data extracted from food-related categories. The datasets consist of plain texts of varying lengths, from brief entries of one or two words, such as tags, to extended passages containing several sentences. These include Italian recipes, intangible heritage assets, and Wikipedia entries. The texts are presented in one or more languages across different datasets. Furthermore, each element may serve as a link to an archive item or as a query performed on the archive, particularly in the case of tags, as follows:

QueryLab_descr: This dataset includes titles and brief descriptions of inventoried assets. The text describes intangible cultural items, such as a dance, ritual, or knowledge, among others, with the intent to protect the asset and preserve information for future generations. The lengths of the descriptions range from a few lines to many paragraphs. The languages are Italian, English, French, and German.

QueryLab_tags: Expert-defined tags connected with archive documents provide additional contextual information. This study employed two separate datasets. The first dataset, from the Ethnography and Social History Archive [45], includes tags chosen by expert ethnographers. These tags were originally defined in Italian and later translated into English, French, and German. Here, we use the English version of tags. The second dataset, from UNESCO's Intangible Heritage and Cultural Asset Management [46], comprises simple or compound tags, affording significant insights into the nature and characteristics of cultural heritage assets.

Wikipedia_food: We scraped data from Wikipedia starting from the root of the food category and iteratively expanding in three different languages: Italian, English, and Spanish. The texts are extensive and cover a wide range of food-related topics, including traditional dishes, beverages, utensils, and recipes. For each language, we considered 500 items, resulting in entries that cover the same concepts in different languages as well as unique and overlapping entries.

CookIT: It is an archive of traditional Italian recipes, sourced from reputable websites. Each recipe's origin is clearly indicated, along with a link to the original source. For every recipe, the archive includes the name of the dish, a list of ingredients, detailed cooking instructions, and, where available, trivia and additional information. The texts, mainly in Italian, provided for each recipe are notably comprehensive.

The main challenges with the datasets used arise from working with real data about intangible cultural heritage, which is created by communities to preserve and pass down their traditions, sayings, dialect expressions, local object names, masks, and other cultural elements. Because this heritage is so unique and specialized, the terms and concepts involved are rarely represented in pre-trained models.

4.2. The Proposed Approach

In this paper, we employ transformers in combination with UMAP and HDBSCAN (or Mean Shift) for clustering data. Specifically, transformers and BERT-like models are used (and fine-tuned) to transform text data into high-dimensional vectors that capture semantic meaning. We then apply UMAP to these vectors to reduce the dimensionality, creating a lower-dimensional space that serves as the input for the chosen clustering algorithm. We conducted numerous tests to determine the optimal hyperparameters for UMAP, HDBSCAN, Mean Shift, and the pre-trained models for various datasets.

For UMAP, we tested the n_neighbors parameter, which controls the balance between the local and global data structure. The default value in the Python implementation is 15, and we evaluated the following values: 15, 10, 5, and 3.

For HDBSCAN, we tested different values for min_cluster_size and min_samples. The min_cluster_size parameter sets the smallest cluster size to be considered a cluster, while min_samples provides a measure of clustering conservatism. The values tested for min_cluster_size were 15, 10, and 5, and, for min_samples, we tested 5 and 1.

In addition, we also explored the use of the Mean Shift clustering algorithm, testing various bandwidths to determine the optimal clustering performance. It determines the radius of the region the algorithm uses to search for neighbors and affects the number of clusters detected. The bandwidth values tested were 1.0, 0.9, 0.7, 0.6, and 0.5.

When the number of clusters exceeded a certain threshold, we implemented a recursive procedure to create multi-level graphs. This approach is designed to facilitate user navigation of the dataset by organizing the data into a hierarchical structure. The multi-level graphs help users to easily explore and understand the relationships and patterns within the dataset, improving both usability and comprehension.

We utilized several Italian, English, and multilingual large language models (LLMs) to process and analyze the text data.

- tgsc/sentence-transformers_paraphrase-multilingual-mpnet-base-v2 ('tgmu'): It is designed to generate high-quality multilingual paraphrases that effectively capture semantic similarities across languages while maintaining contextual integrity.
- paraphrase-multilingual-MiniLM-L12-v2 ('mimu'): A multilingual model designed to handle multiple languages simultaneously. It is based on the MiniLM architecture, which is a smaller and faster version of BERT.
- xlm-roberta-base-multilingual-en-ar-fr-de-es-tr-it ('romu'): A robust model trained on a diverse multilingual corpus, including languages such as English, Arabic, French, German, Spanish, Turkish, and Italian. XLM-RoBERTa is known for its strong performance across various languages.
- all_datasets_v3_mpnet-base ('flax'): This model combines the capabilities of MPNet and BERT, providing enhanced contextual embeddings. It has been fine-tuned on various datasets to improve its versatility.
- Bert-base-Wikipedia-sections-mean-tokens ('bewi'): A BERT model fine-tuned on Wikipedia sections, providing embeddings based on the mean of token embeddings. This model is particularly good at capturing the semantic structure of Wikipedia articles.
- distiluse-base-multilingual-cased ('dimu'): A smaller, faster, and cheaper version of BERT, maintaining 97% of BERT's performance while being 60% faster, with multilingual capabilities for diverse text data.

The abbreviations in parentheses are used to identify the models listed in the tables in the Results section.

The problem of terms not present in the pre-trained model is overcome by the use of BERT-like transformers and their tokenizers. This solution has had much better results than the use of single-word-level tokenizers or models with Word2Vec [47] or GloVe [48].

Tokenization and vector representation enable machines to understand and process textual data. We used native tokenizers designed specifically for the LLMs used, taking advantage of the tokenization utilities available in PyTorch and the Hugging Face Transformers library. In this way, we ensured that the textual inputs are processed in a way that is fully compatible with the LLM's architecture and training regime.

In the vector representation, each token is typically represented as a numerical vector through embedding techniques. These vectors encode semantic meaning and relationships between words or tokens in a high-dimensional space. Models like Word2Vec, GloVe, and BERT utilize different strategies to generate these embeddings; Word2Vec and GloVe produce static embeddings based on co-occurrence statistics, while BERT and the other LLMs' transformer-based models generate contextual embeddings that consider the surrounding context of each token within a sentence or document.

When working with word embeddings such as Word2Vec or GloVe, the typical approach involves averaging the vectors, possibly weighting them based on word frequency or importance. In contrast, BERT-like models utilize not only the average (AVG) token but also the [CLS] token, which encapsulates the entire sentence [16,33].

To create a similarity matrix, pairwise comparisons are made between the [CLS] or [AVG] tokens using metrics like cosine similarity. The resulting scores indicate the similarity relationships among the [CLS] or [AVG] tokens of the input text, forming the basis of the similarity matrix.

Keywords are extracted from the entire dataset and each individual cluster uses class-based TF-IDF scoring. This technique computes the importance of terms by considering their frequency within documents and across clusters. The extracted keywords are then visualized as word clouds, offering users a graphical representation that highlights the most significant terms in the dataset and within each cluster. These word clouds serve as intuitive tools that allow users to quickly grasp the prominent themes and topics present in the analyzed data.

4.3. The Results

We tested our pipeline on the datasets previously presented. Rather than evaluating each innovative element individually, we chose to assess the multi-layer semantic graphs using standard quantitative measures, such as Silhouette Scores and the others detailed above. Some of the results and graphs generated from the pipeline can be accessed via the following URL address: http://arm.mi.imati.cnr.it/papers/Kgraphs/html/index.html (accessed on 17 September 2024).

4.3.1. Clustering

Since one of the key features of our system is its completely unsupervised nature, we determined that the clustering configuration where the parameter value of the Calinski–Harabasz score is maximum is automatically selected. This selection is made at the first level of clustering, and, if necessary, is iterated to produce a second-level clustering.

The provided tables—two for each dataset corresponding to the two clustering algorithms tested (HDBSCAN and Mean Shift)—contain rows that correspond to the parameter values optimizing the results. In the Supplementary Materials of the paper, there are two Excel files that include all results obtained as hyperparameters vary. These tables are designed to track both the current clustering level (denoted as "I") and the previous clustering level (denoted as "minus_1") for the second level of clustering. The "I" column (and "minus_1" if applicable) corresponds to the pipeline configuration used to obtain the respective results. The string in these columns is composed of the clustering algorithm, the clustering level, the hyperparameters, and the large language model used. For UMAP+HDBSCAN, the hyperparameters include the number of neighbors (n_n), minimum cluster size (min_clu_s), and minimum samples (min_samp), while, for UMAP+Mean Shift, they include the number of neighbors (n_n) and bandwidth (bandw). These hyperparameters are also displayed in the corresponding columns. The number of elements (docs) to be clustered is always indicated for clarity, along with the number of clusters obtained (clus) and the level (level). For both HDBSCAN and Mean Shift algorithms, four evaluation metrics are consistently provided: Silhouette Score and the Calinski–Harabasz, Davies–Bouldin, and Dunn Indices. The results are sorted in descending order according to the Calinski–Harabasz Index value. The highest values, at the top, indicate better clustering. To ensure that the clusters were well separated and not overlapping, similarity matrices were created. These matrices were constructed using the terms from each cluster to evaluate their similarities. The similarity between clusters is visually represented by their coloration; clusters with higher similarity have a darker coloration, while those with lower similarity appear lighter. The similarity is computed using the cosine similarity measure on the resulting vectors, providing a quantification of how closely related the clusters are based on their feature vectors.

QueryLab_descr

QueryLab_descr is a dataset consisting of data extracted from the QueryLab portal. It includes inventoried assets of intangible goods in Lombardy, Bolzano, Trentino Alto Adige, and Switzerland. The data are written in several languages. The pre-trained models used are multilingual. In addition to MIMU, which was used in all experiments, the model "xlm-roberta-base-multilingual-en-ar-fr-de-es-tr-it" was also used on this dataset.

It was observed that HDBSCAN detects fewer clusters, which is a general trend in all datasets. One-level graphs were generated. Mean Shift, on the other hand, generates multi-level graphs for both models. As shown in Tables 2 and 3, the best results (maximum Calinski–Harabasz Index) for both algorithms are achieved with a low value of n_neighbor. For HDBSCAN, the optimal performance is obtained with a min_cluster_size of 10 for both LLMs. Mean Shift achieves the best results at level 1 with the minimum bandwidth value for both models.

Table 2. Best results of the UMAP+HDBSCAN on QueryLab_descr dataset.

I	Docs	Level	N_N	Min_Clu_S	Min_Sam	Clus	Silhouette	Calinski	Davies	Dunn
_hL1_3_10_5_romu	463	1	3	10	5	13	0.48	554.39	1.65	0.01
_hL1_3_10_1_mimu	463	1	3	10	1	15	0.47	471.50	1.26	0.06

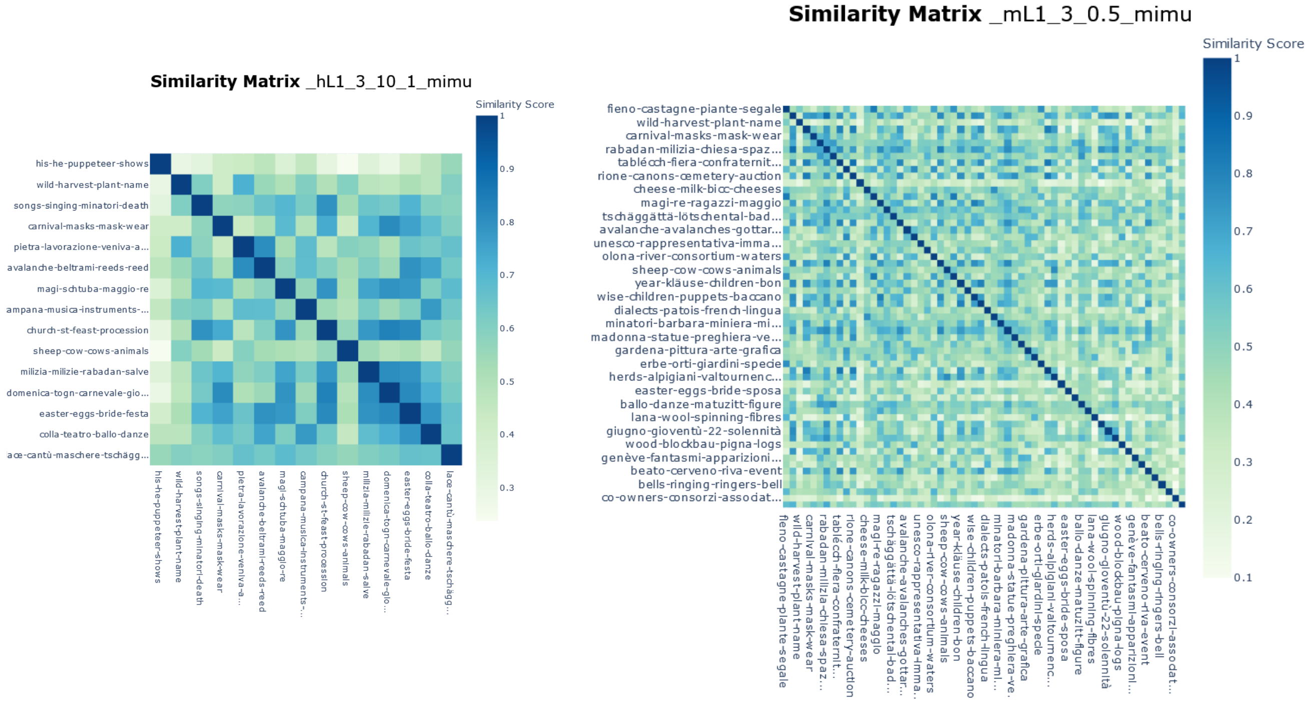
Table 3. Best results of the UMAP+Mean Shift on QueryLab_descr dataset.

I	Minus_1	Docs	Level	N_N	Bandw	Clus	Silhouette	Calinski	Davies	Dunn
_mL1_3_0.5_mimu		463	1	3	0.5	60	0.53	1413.30	0.57	0.09
_mL1_3_0.5_romu		463	1	3	0.5	62	0.53	1387.72	0.58	0.02
_mL2_3_0.5_romu	_mL1_3_0.5_romu	62	2	3	0.5	20	0.51	167.58	0.48	0.46
_mL2_3_0.7_romu	_mL1_3_0.5_romu	62	2	3	0.7	15	0.50	145.04	0.61	0.24
_mL2_3_0.6_romu	_mL1_3_0.5_romu	62	2	3	0.6	17	0.49	141.56	0.48	0.24
_mL2_3_0.5_mimu	_mL1_3_0.5_mimu	60	2	3	0.5	23	0.46	138.10	0.51	0.50
_mL2_3_0.6_mimu	_mL1_3_0.5_mimu	60	2	3	0.6	18	0.45	121.94	0.55	0.30
_mL2_3_0.7_mimu	_mL1_3_0.5_mimu	60	2	3	0.7	14	0.48	117.36	0.59	0.28
_mL2_3_0.9_romu	_mL1_3_0.5_romu	62	2	3	0.9	11	0.51	113.96	0.57	0.20
_mL2_3_0.9_mimu	_mL1_3_0.5_mimu	60	2	3	0.9	11	0.43	100.70	0.67	0.13
_mL2_3_1.0_romu	_mL1_3_0.5_romu	62	2	3	1	9	0.42	93.42	0.73	0.14
_mL2_3_1.0_mimu	_mL1_3_0.5_mimu	60	2	3	1	7	0.42	87.46	0.77	0.18

Figure 2 shows the level 1 similarity matrices for two different clustering methods. On the left is the similarity matrix generated by HDBSCAN, while, on the right, is the similarity matrix generated by the Mean Shift algorithm. Both matrices use the MIMU LLM and are based on the QueryLab_descr dataset.

QueryLab_intangible_tags

The QueryLab tags dataset consists of 260 items, most of which are single words. As shown in Tables 4 and 5, the number of clusters identified by HDBSCAN is much smaller than that identified by Mean Shift. HDBSCAN (with UMAP) uses three parameters, which are 15, 15, and 5, for both models.



(**a**) (**b**)

Figure 2. Level 1 similarity matrix for HDBSCAN (**a**) and Mean Shift (**b**) with MIMU LLM on QueryLab_descr dataset.

Table 4. Best results of the UMAP+HDBSCAN on QueryLab_intangible_tags dataset.

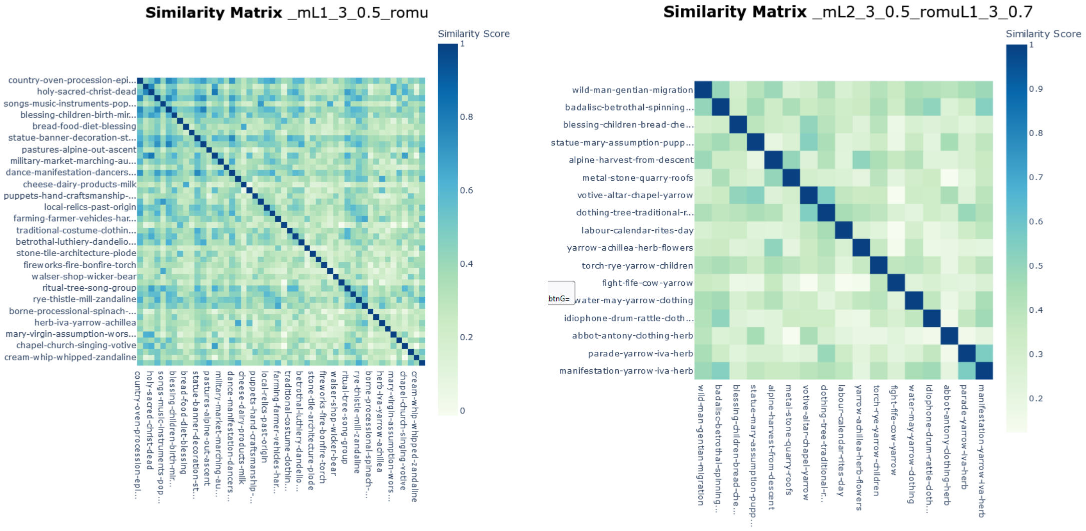
I	Docs	Level	N_N	Min_Clu_S	Min_Sam	Clus	Silhouette	Calinski	Davies	Dunn
_hL1_15_15_5_mimu	260	1	15	15	5	5	0.37	147.84	1.26	0.06
_hL1_15_15_5_romu	260	1	15	15	5	5	0.35	135.43	1.23	0.04

Table 5. Best results of the UMAP+Mean Shift on QueryLab_intangible_tags dataset.

I	Minus_1	Docs	Level	N_N	Bandw	Clus	Silhouette	Calinski	Davies	Dunn
_mL1_3_0.5_romu		260	1	3	0.5	50	0.63	1270.73	0.43	0.11
_mL1_3_0.5_mimu		260	1	3	0.5	50	0.60	1208.45	0.47	0.19
_mL2_3_0.5_romu	_mL1_3_0.5_romu	50	2	3	0.5	22	0.54	323.48	0.46	0.58
_mL2_3_0.6_romu	_mL1_3_0.5_romu	50	2	3	0.6	17	0.51	215.59	0.59	0.40
_mL2_3_1.0_romu	_mL1_3_0.5_romu	50	2	3	1	4	0.57	198.54	0.60	0.23
_mL2_3_0.5_mimu	_mL1_3_0.5_mimu	50	2	3	0.5	18	0.45	181.18	0.47	0.50
_mL2_3_0.6_mimu	_mL1_3_0.5_mimu	50	2	3	0.6	16	0.48	178.63	0.48	0.34
_mL2_3_0.7_romu	_mL1_3_0.5_romu	50	2	3	0.7	12	0.45	163.93	0.66	0.26
_mL2_3_0.9_romu	_mL1_3_0.5_romu	50	2	3	0.9	6	0.53	163.00	0.66	0.16
_mL2_3_0.7_mimu	_mL1_3_0.5_mimu	50	2	3	0.7	13	0.49	161.10	0.53	0.23
_mL2_3_0.9_mimu	_mL1_3_0.5_mimu	50	2	3	0.9	9	0.56	120.06	0.53	0.24
_mL2_3_1.0_mimu	_mL1_3_0.5_mimu	50	2	3	1	9	0.55	119.03	0.55	0.19

The distribution of items, excluding outliers, is Cluster 0: 22 items, Cluster 1: 106 items, Cluster 2: 44 items, Cluster 3: 23 items, Cluster 4: 38 items. Unlike the procedure where clustering iterates when the number of clusters exceeds a given threshold, in this case, iteration should occur when the number of items in a cluster exceeds the same threshold. In the Mean Shift case, the best results are achieved with n_neighbor = 3 and bandwidth = 0.5, producing 50 clusters with both LLMs. At the second level, varying the parameters yields between 4 and 22 clusters. Here, we report all values related to the first-level hyperparameters to evaluate which solutions are the best, also considering feedback from web users.

Figure 3 shows similarity matrices illustrating the results of the Mean Shift algorithm. The left side shows the level 1 similarity matrix, while the right side shows the level 2 similarity matrix. Both matrices were generated using the ROMU LLM on the QueryLab_intangible_tag dataset.



(**a**) (**b**)

Figure 3. Similarity matrix for Mean Shift level 1 (**a**) and level 2 (**b**) with ROMU LLM on QueryLab_intangible_tags dataset.

QueryLab UNESCO Tags

UNESCO's Intangible Cultural Heritage (ICH) Tags are specific labels or categories used to identify and classify elements of intangible cultural heritage. They have been integrated into the QueryLab portal to be used together with intangible tags. There are 828 items in English languages. Here, we tested the pipeline on three LLMs, TGMU, FLAX, and MIMU, the only one used on all datasets. The results are shown in Tables 6 and 7.

Table 6. Best results of the UMAP+HDBSCAN on QueryLab_UNESCO_tags dataset.

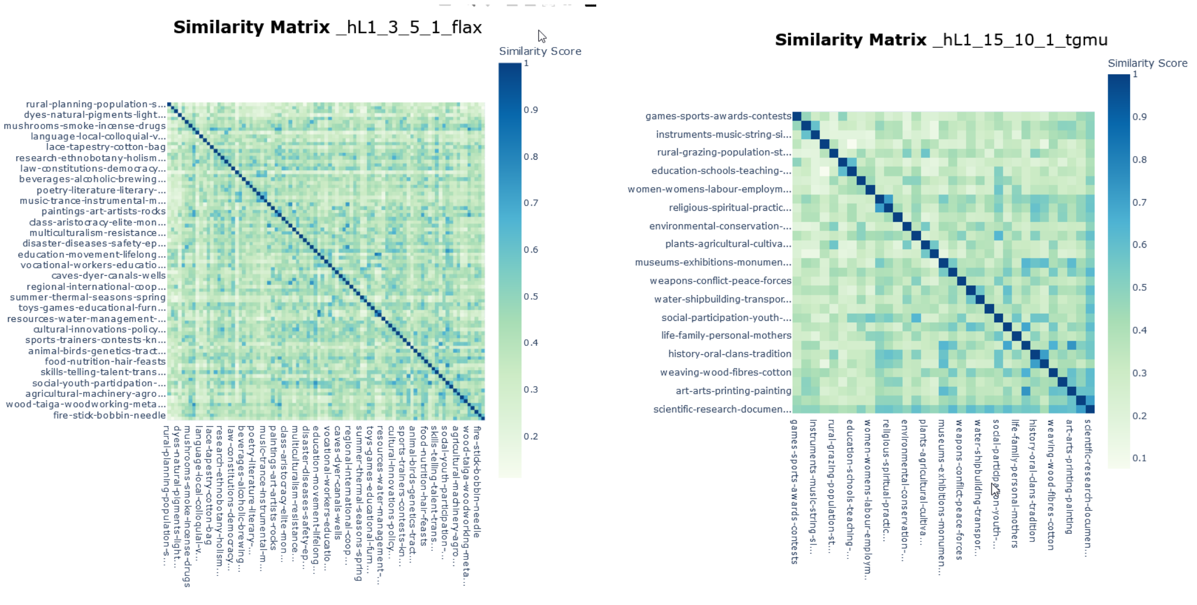
I	Minus_1	Docs	Level	N_N	Min_Clu_S	Min_Sam	Clus	Silhouette	Calinski	Davies	Dunn
_hL2_3_5_5_tgmu	_hL1_3_15_1_tgmu	24	2	3	5	5	2	0.71	380.87	0.35	0.90
_hL1_10_15_1_mimu		828	1	10	15	1	22	0.33	227.04	1.14	0.07
_hL1_15_10_1_tgmu		828	1	15	10	1	33	0.37	209.90	1.06	0.05
_hL1_3_5_1_flax		828	1	3	5	1	89	0.54	208.26	0.97	0.03
_hL2_3_5_1_flax	_hL1_3_5_1_flax	89	2	3	5	1	10	0.57	175.67	0.58	0.24
_hL2_3_5_5_flax	_hL1_3_5_1_flax	89	2	3	5	5	6	0.46	82.79	1.15	0.07
_hL2_3_15_1_flax	_hL1_3_5_1_flax	89	2	3	15	1	3	0.47	81.24	0.76	0.28
_hL2_3_15_5_flax	_hL1_3_5_1_flax	89	2	3	15	5	3	0.42	60.36	1.24	0.05
_hL2_3_10_5_flax	_hL1_3_5_1_flax	89	2	3	10	5	3	0.42	60.36	1.24	0.05
_hL2_3_10_1_flax	_hL1_3_5_1_flax	89	2	3	10	1	4	0.31	25.78	2.91	0.10
_hL2_15_10_1_tgmu	_hL1_15_10_1_tgmu	33	2	15	10	1	2	0.27	14.66	1.22	0.46
_hL2_15_5_1_tgmu	_hL1_15_10_1_tgmu	33	2	15	5	1	3	0.27	14.66	1.22	0.46
_hL2_10_5_1_mimu	_hL1_10_15_1_mimu	22	2	10	5	1	2	0.25	12.73	1.33	0.38
_hL2_10_5_5_mimu	_hL1_10_15_1_mimu	22	2	10	5	5	2	0.21	12.35	1.67	0.25
_hL2_15_5_5_tgmu	_hL1_15_10_1_tgmu	33	2	15	5	5	2	0.05	5.71	2.39	0.19

Table 7. Best results of the UMAP+Mean Shift on QueryLab_UNESCO_tags dataset.

I	Minus_1	Docs	Level	N_N	Bandw	Clus	Silhouette	Calinski	Davies	Dunn
_mL1_3_0.5_mimu		828	1	3	0.5	102	0.65	3385.96	0.43	0.05
_mL1_3_0.5_tgmu		828	1	3	0.5	102	0.64	2639.38	0.43	0.02
_mL1_3_0.5_flax		828	1	3	0.5	106	0.61	1305.95	0.48	0.06
_mL2_3_0.5_mimu	_mL1_3_0.5_mimu	102	2	3	0.5	28	0.55	1048.99	0.51	0.27
_mL2_3_0.6_mimu	_mL1_3_0.5_mimu	102	2	3	0.6	26	0.54	960.69	0.55	0.26
_mL2_3_0.5_flax	_mL1_3_0.5_flax	106	2	3	0.5	26	0.58	762.09	0.48	0.24
_mL2_3_0.7_mimu	_mL1_3_0.5_mimu	102	2	3	0.7	19	0.51	715.05	0.57	0.13
_mL2_3_0.9_mimu	_mL1_3_0.5_mimu	102	2	3	0.9	15	0.53	694.39	0.58	0.10
_mL2_3_1.0_mimu	_mL1_3_0.5_mimu	102	2	3	1	13	0.56	683.80	0.56	0.15
_mL2_3_0.6_flax	_mL1_3_0.5_flax	106	2	3	0.6	22	0.57	663.88	0.52	0.32
_mL2_3_0.7_flax	_mL1_3_0.5_flax	106	2	3	0.7	20	0.55	608.69	0.54	0.12
_mL2_3_0.5_tgmu	_mL1_3_0.5_tgmu	102	2	3	0.5	28	0.57	587.24	0.48	0.27
_mL2_3_0.6_tgmu	_mL1_3_0.5_tgmu	102	2	3	0.6	22	0.55	465.31	0.56	0.26
_mL2_3_0.9_flax	_mL1_3_0.5_flax	106	2	3	0.9	11	0.59	428.74	0.59	0.12
_mL2_3_1.0_flax	_mL1_3_0.5_flax	106	2	3	1	11	0.58	407.88	0.53	0.12
_mL2_3_0.7_tgmu	_mL1_3_0.5_tgmu	102	2	3	0.7	20	0.56	406.89	0.51	0.11
_mL2_3_0.9_tgmu	_mL1_3_0.5_tgmu	102	2	3	0.9	16	0.51	347.97	0.57	0.15
_mL2_3_1.0_tgmu	_mL1_3_0.5_tgmu	102	2	3	1	11	0.55	333.78	0.59	0.14

Both algorithms generate a high number of clusters, especially at low hyperparameter values. For HDBSCAN, the optimal configurations produce 22, 33, and 89 clusters at level 1 for MIMU, TGMU, and FLAX, respectively. Unlike the other two models, FLAX achieves the best result with all three parameters set to their minimum values, resulting in a larger number of clusters. At the highest level, these configurations produce between 2 and 10 clusters. Mean Shift, which consistently produces a larger number of clusters, produces 102 and 106 clusters at level 1, which are further subdivided into 11 to 28 clusters.

Figure 4 illustrates the level 1 similarity matrix for HDBSCAN, showing optimal results for two different language models. On the left is the similarity matrix for the FLAX LLM, while, on the right, is the similarity matrix for the TGMU model. Both matrices are derived from the QueryLab_UNESCO_tag dataset.



(**a**) (**b**)

Figure 4. Level 1 similarity matrix for HDBSCAN for optima results for FLAX LLM (**a**) and TGMU (**b**) on QueryLab_UNESCO_tag dataset.

CookIT Most Frequent

CookIT Most Frequent is a dataset consisting of traditional culinary recipes written in Italian. Alongside MIMU, a multilingual model was also used because recipes and food vocabulary often include terms from other languages, such as French, requiring a broader linguistic understanding for accurate processing. HDBSCAN produces a relatively limited number of clusters. To address this, we considered not only the configuration that optimized the Calinski score but also the next maximum value due to the small number of clusters in the initial analysis.

Using Mean Shift, both LLMs generate a comparable number of elements at the first level. At the second level, the number of clusters varies from 9 to 27 for the DIMU model, while, for MIMU, it ranges from 7 to 20. The results are reported in Tables 8 and 9.

Table 8. Best results of the UMAP+HDBSCAN on CookIT Most Frequent dataset.

I	Docs	Level	N_N	Min_Clu_S	Min_Sam	Clus	Silhouette	Calinski	Davies	Dunn
_hL1_3_10_5_dimu	488	1	3	10	5	2	0.63	146.21	0.29	0.93
_hL1_3_10_1_dimu	488	1	3	10	1	21	0.27	107.95	1.20	0.08
_hL1_15_15_1_mimu	488	1	15	15	1	9	0.12	96.14	1.54	0.11

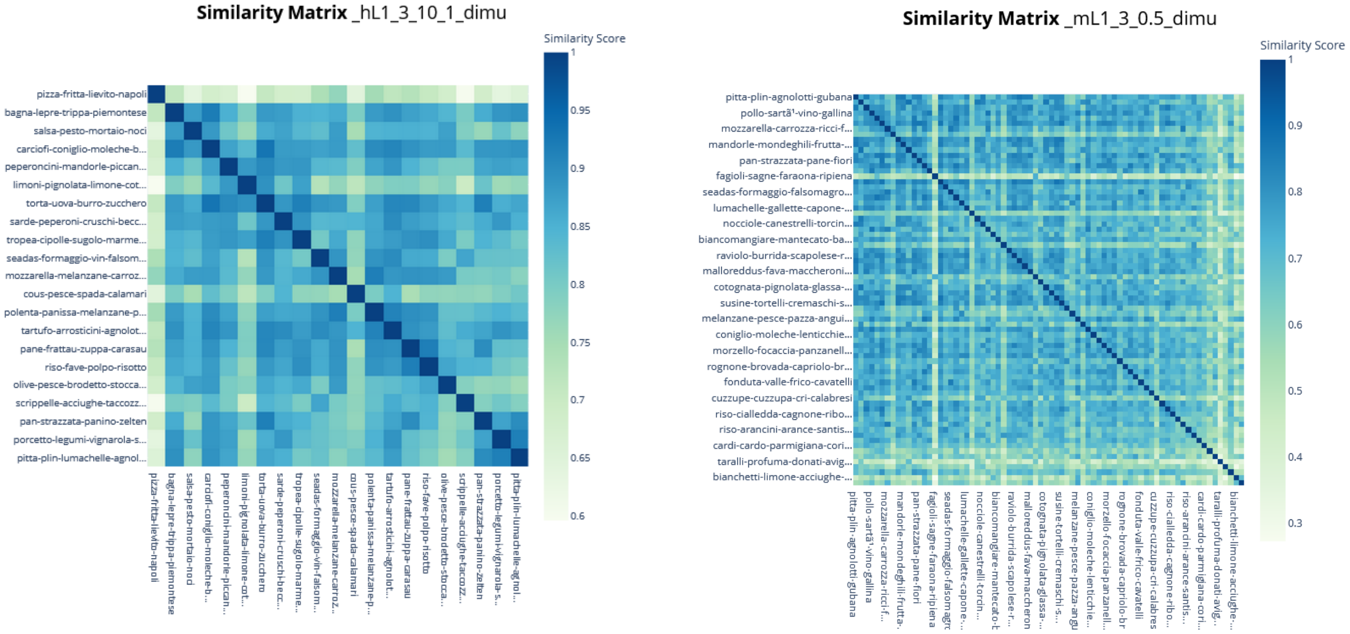
Table 9. Best results of the UMAP+Mean Shift on CookIT Most Frequent dataset.

I	minus_1	Docs	Level	n_n	Bandw	Clus	Silhouette	Calinski	Davies	Dunn
_mL1_3_0.5_mimu		488	1	3	0.5	60	0.53	612.58	0.59	0.05
_mL1_3_0.5_dimu		488	1	3	0.5	74	0.53	444.09	0.60	0.07
_mL2_3_0.6_mimu	_mL1_3_0.5_mimu	60	2	3	0.6	18	0.54	85.05	0.53	0.45
_mL2_3_0.5_mimu	_mL1_3_0.5_mimu	60	2	3	0.5	20	0.49	78.56	0.55	0.31
_mL2_3_0.7_mimu	_mL1_3_0.5_mimu	60	2	3	0.7	16	0.51	76.91	0.59	0.31
_mL2_3_0.7_dimu	_mL1_3_0.5_dimu	74	2	3	0.7	13	0.44	70.72	0.67	0.22
_mL2_3_0.5_dimu	_mL1_3_0.5_dimu	74	2	3	0.5	27	0.38	70.66	0.55	0.28
_mL2_3_0.6_dimu	_mL1_3_0.5_dimu	74	2	3	0.6	18	0.43	67.82	0.59	0.28
_mL2_3_0.9_dimu	_mL1_3_0.5_dimu	74	2	3	0.9	10	0.43	66.91	0.76	0.19
_mL2_3_1.0_dimu	_mL1_3_0.5_dimu	74	2	3	1	9	0.42	60.38	0.77	0.15
_mL2_3_0.9_mimu	_mL1_3_0.5_mimu	60	2	3	0.9	11	0.48	48.23	0.60	0.16
_mL2_3_1.0_mimu	_mL1_3_0.5_mimu	60	2	3	1	7	0.37	45.19	0.90	0.13

Figure 5 represents the similarity matrix of the clustering results using the two algorithms with the same LLM (DIMU). HDBSCAN, at the first (and only) level, produces 21 clusters, whereas Mean Shift at the first level generates 74 clusters, necessitating another iteration.

Wikipedia_food

This dataset is the largest collection, with approximately 2000 items. It was included because its texts are long, written in several languages, and contain many significant terms. Due to the large number of items, two-level graphs are generated for both clustering algorithms, as shown in Tables 10 and 11. With all hyperparameter values, HDBSCAN produces two top-level clusters. In contrast, Mean Shift produces more diverse results, generating over 150 clusters at the first level, which are then reduced to a range of 20 to 30 clusters at the second level for both LLMs. Interestingly, the highest scores are achieved with elevated hyperparameter values, resulting in a relatively small number of first-level clusters, while Mean Shift performs best with lower hyperparameter values, producing over 150 first-level clusters.



(**a**) (**b**)

Figure 5. Similarity matrix for HDBSCAN (**a**) and Mean Shift (**b**) with DIMU LLM on CookIT Most Frequent dataset.

Table 10. Best results of the UMAP+HDBSCAN on Wikipedia_food dataset.

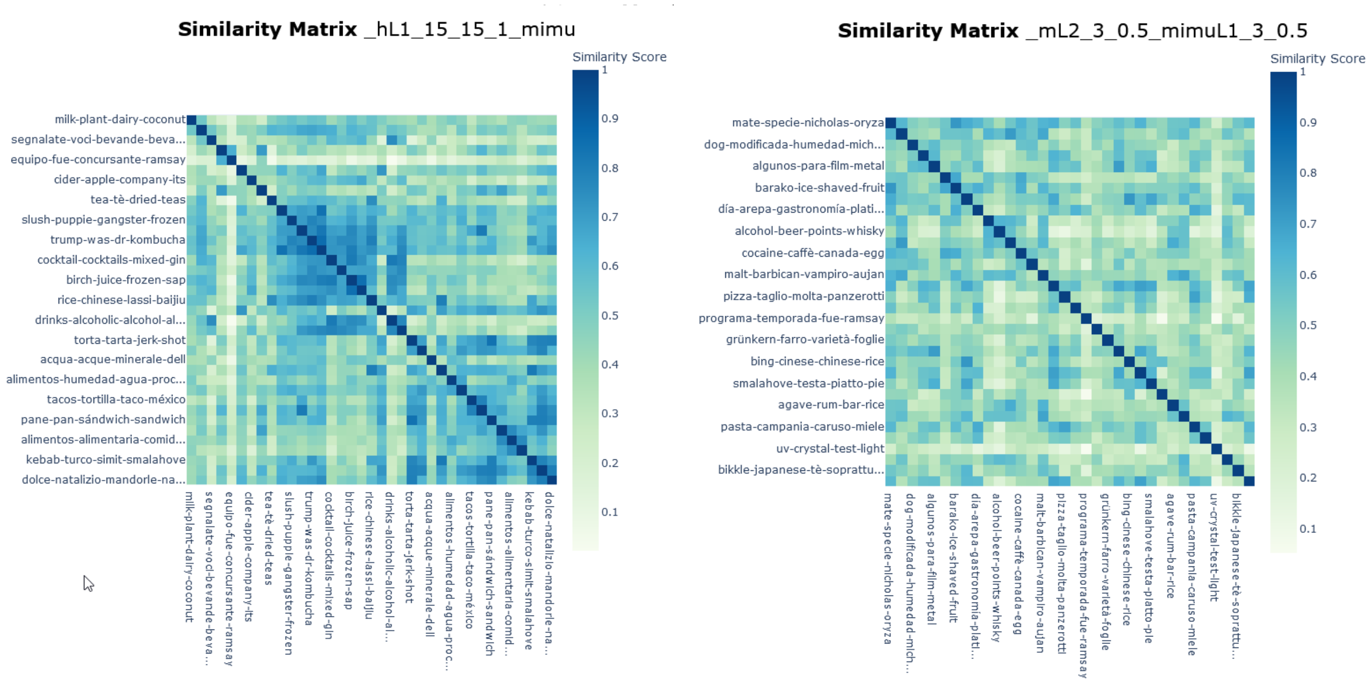
I	minus_1	Docs	Level	n_n	Min_clu_s	Min_sam	Clus	Silhouette	Calinski	Davies	Dunn
_hL2_3_15_5_mimu	_hL1_3_5_5_mimu	109	2	3	15	5	2	0.75	396.63	0.29	1.17
_hL1_15_15_1_mimu		1998	1	15	15	1	37	0.31	337.53	1.19	0.04
_hL1_10_15_1_romu		1998	1	10	15	1	36	0.25	307.27	1.09	0.03
_hL2_10_15_1_romu	_hL1_10_15_1_romu	36	2	10	15	1	2	0.49	50.24	0.78	0.43
_hL2_10_10_1_romu	_hL1_10_15_1_romu	36	2	10	10	1	2	0.49	50.24	0.78	0.43
_hL2_10_5_1_romu	_hL1_10_15_1_romu	36	2	10	5	1	2	0.49	50.24	0.78	0.43
_hL2_15_15_1_mimu	_hL1_15_15_1_mimu	37	2	15	15	1	2	0.17	20.63	0.86	0.41
_hL2_15_10_1_mimu	_hL1_15_15_1_mimu	37	2	15	10	1	2	0.17	20.63	0.86	0.41
_hL2_15_5_1_mimu	_hL1_15_15_1_mimu	37	2	15	5	1	2	0.17	20.63	0.86	0.41
_hL2_10_10_5_romu	_hL1_10_15_1_romu	36	2	10	10	5	2	0.25	19.42	1.82	0.14
_hL2_10_5_5_romu	_hL1_10_15_1_romu	36	2	10	5	5	2	0.25	19.42	1.82	0.14
_hL2_15_5_5_mimu	_hL1_15_15_1_mimu	37	2	15	5	5	2	0.19	18.51	1.69	0.17

Table 11. Best results of the UMAP+Mean Shift on Wikipedia_food dataset.

I	minus_1	Docs	Level	n_n	Bandw	Clus	Silhouette	Calinski	Davies	Dunn
_mL1_3_0.5_mimu		1998	1	3	0.5	155	0.59	3630.89	0.49	0.02
_mL1_3_0.5_romu		1998	1	3	0.5	162	0.59	3609.27	0.51	0.00
_mL2_3_0.5_mimu	_mL1_3_0.5_mimu	155	2	3	0.5	34	0.54	571.38	0.57	0.21
_mL2_3_0.6_mimu	_mL1_3_0.5_mimu	155	2	3	0.6	26	0.51	475.13	0.59	0.12
_mL2_3_0.7_mimu	_mL1_3_0.5_mimu	155	2	3	0.7	21	0.52	433.22	0.58	0.03
_mL2_3_0.9_mimu	_mL1_3_0.5_mimu	155	2	3	0.9	14	0.54	347.64	0.59	0.16
_mL2_3_1.0_mimu	_mL1_3_0.5_mimu	155	2	3	1	13	0.50	292.86	0.58	0.08
_mL2_3_0.5_romu	_mL1_3_0.5_romu	162	2	3	0.5	36	0.50	192.95	0.60	0.13
_mL2_3_0.6_romu	_mL1_3_0.5_romu	162	2	3	0.6	31	0.50	180.01	0.62	0.12
_mL2_3_0.7_romu	_mL1_3_0.5_romu	162	2	3	0.7	26	0.51	163.24	0.63	0.24
_mL2_3_0.9_romu	_mL1_3_0.5_romu	162	2	3	0.9	16	0.43	118.18	0.78	0.11
_mL2_3_1.0_romu	_mL1_3_0.5_romu	162	2	3	1	13	0.42	116.74	0.81	0.07

Figure 6 presents the similarity matrices for two clustering methods applied to the Wikipedia_food dataset. On the left is the level 1 similarity matrix generated by the

HDBSCAN algorithm, while, on the right, is the level 2 similarity matrix generated by the Mean Shift algorithm. It is noteworthy that both methods yield approximately the same number of clusters. The MIMU LLM was used for both analyses, further emphasizing the consistency of clustering results across methods.



(**a**) (**b**)

Figure 6. Similarity matrix for HDBSCAN level 1 (**a**) and Mean Shift level 2 (**b**) with MIMU LLM on Wikipedia_food dataset. The number of clusters is approximately the same.

4.3.2. Semantic Graph Creation and Visualizations

The visualization and creation of semantic graphs are the last and most practical parts of the whole pipeline. Leveraging the clustering results, we create multi-level graphs that allow users to effortlessly navigate the entire dataset and database, ultimately allowing direct access to individual items. These graphs are constructed using a similarity matrix derived from individual elements or clusters at the lowest level.

To build the similarity matrix, we use a single vector for each element. Using transformers, we compute the average of the [AVG] tokens or the [CLS] tokens that encapsulate whole sentences. The model generates a contextualized vector for each element in the input; it can be a single item at the lowest level or the top elements in the case of level 1 or level 2 clustering. To build the similarity matrix, we compare [CLS] or [AVG] tokens in pairs using a distance metric, in this case, cosine similarity. The resulting scores indicate the similarity relationships between these tokens and are used to construct the matrix. The graphs shown in the figures were created using [CLS] tokens to generate a single vector for each item.

The similarity metric was then used to construct the semantic graphs by selecting the k most similar items for each element, where k ranges from 1 to 4. When k = 1, disjoint graphs may be formed, resulting in some nodes being connected to each other but not to others. In contrast, when k $\geq$ 2, the graphs are fully connected. However, if k = 4, almost all nodes are connected, which can make the graphs difficult to read.

Figure 7 shows the semantic graph constructed from the Wikipedia_food dataset, consisting of 155 interconnected nodes. In clustering, each node is named after the most similar element within its corresponding cluster. Due to the high density of the graph, it presents significant challenges for navigation, as the high volume of connections can lead to information overload. This structure highlights the complexity of the relationships within the dataset, making it difficult to navigate the graph, hence the need for iterative clustering.

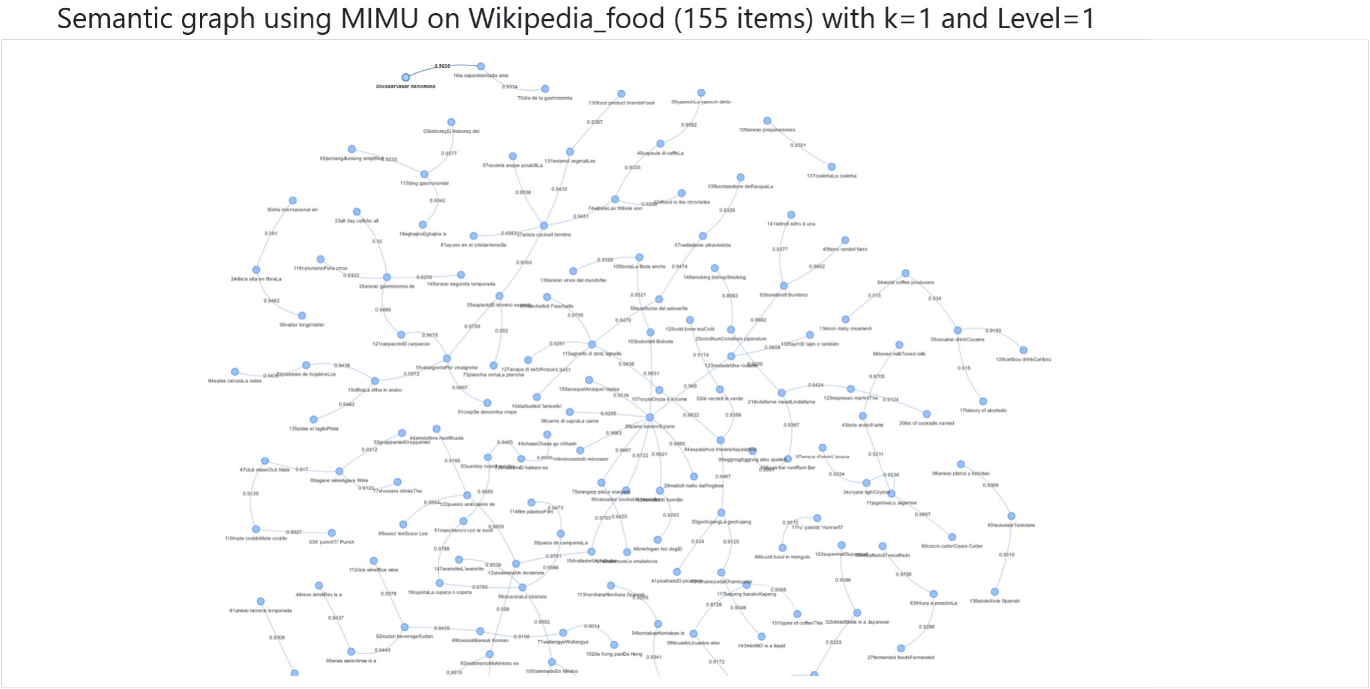


Figure 7. The semantic graph for whole Wikipedia_food dataset consisting of 155 nodes, accessible via the following URL address http://arm.mi.imati.cnr.it/papers/Kgraphs/graphs/wiki_mysql_food_compl_it_v1_bert__mL1_3_0.5_mimu_centroid_best_n1_pyvis2.html, 17 September 2024.

Figure 8 shows the semantic graph for level 2 (L2) of the Wikipedia_food dataset, consisting of 34 nodes. The initial items for this analysis were the 155 items identified in the first clustering. The hyperparameters used for both clustering processes were set to n_neighbor = 3 and bandwidth = 0.5. The clustering algorithm used was Mean Shift.

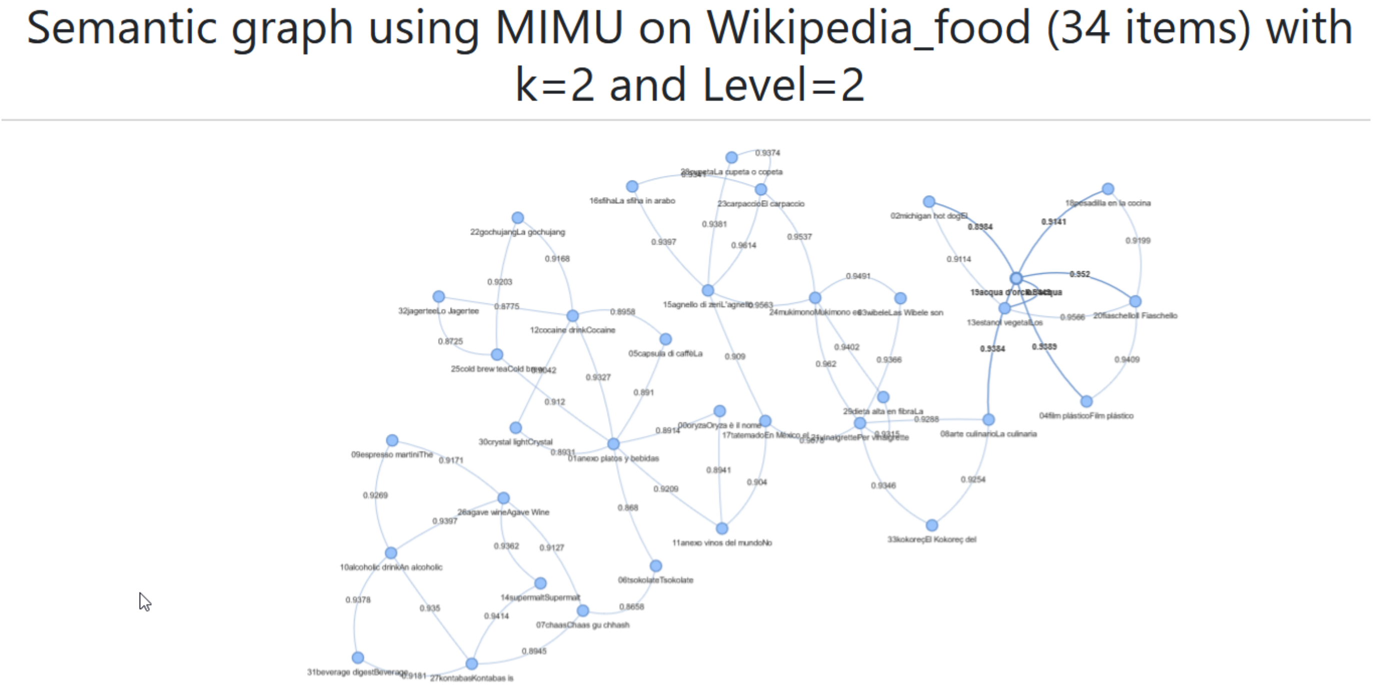


Figure 8. The semantic graph for L2 of Wikipedia_food dataset consisting of 34 nodes, accessible via the following URL address http://arm.mi.imati.cnr.it/papers/Kgraphs/graphs/wiki_mysql_food_compl_it_v1_bert__mL2_3_0.5_mimuL1_3_0.5_centroid_best_n2_pyvis2.html, 17 September 2024.

Figure 9 shows the semantic graph for the entire CookIT dataset, which consists of 21 nodes. Including the two best connections between these nodes (k = 2) increases the complexity of the graph, ensuring that all parts are connected, with no isolated segments. This increased connectivity means that every node is part of the graph, providing a complete view of the relationships within the dataset. The interconnected structure facilitates a deeper examination of how different nodes relate to each other, providing deeper insights. However, the added connections also increase complexity, potentially making the graph more difficult to interpret. In addition, because all nodes are connected, the graph can become cluttered, leading to potential information overload and making it difficult to focus on individual nodes or specific connections without additional filtering or visualization techniques.

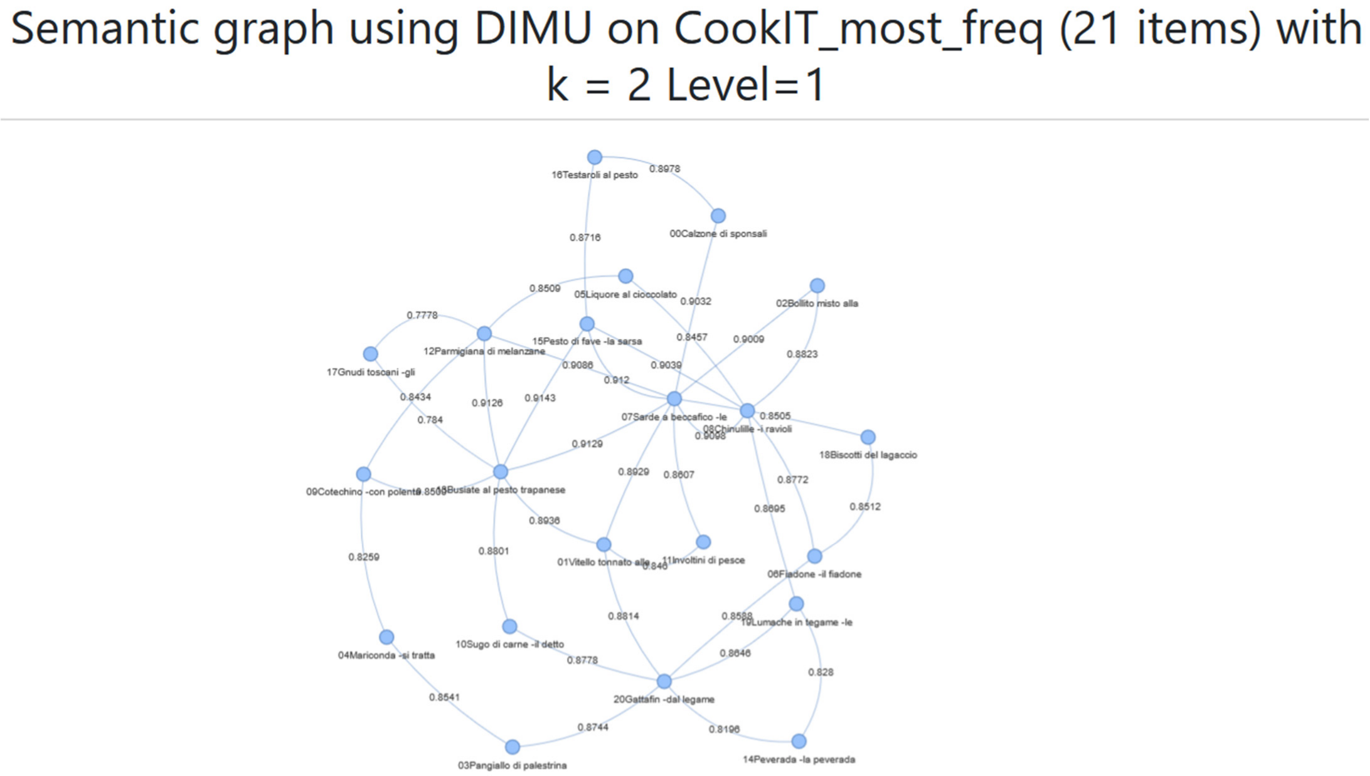
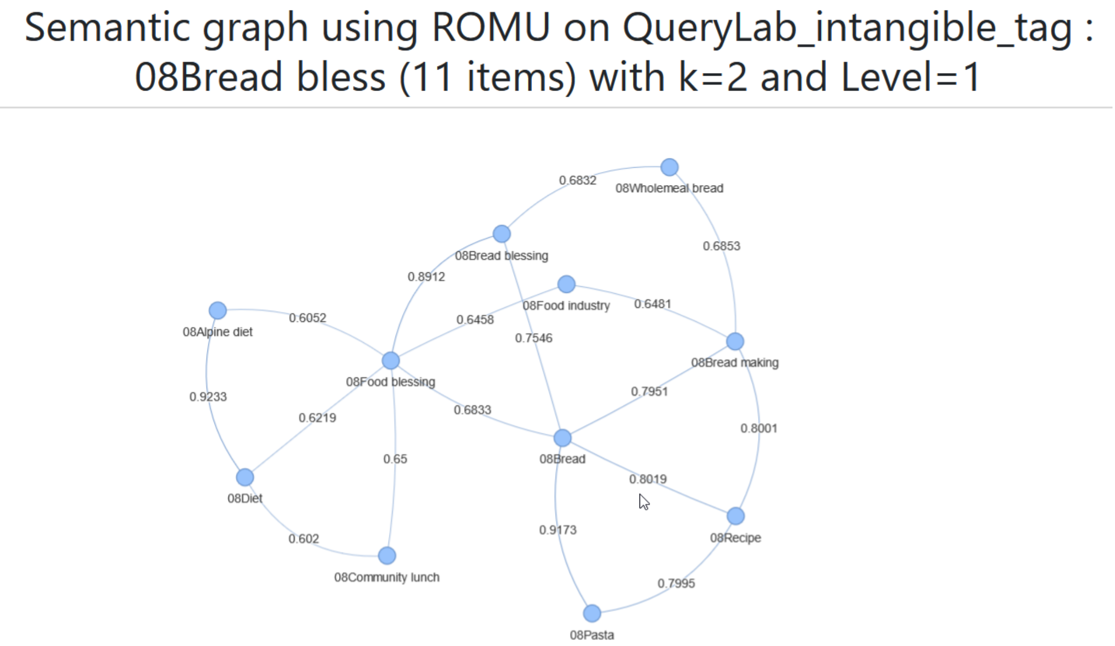


Figure 9. The semantic graph for whole CookIT dataset consisting of 21 nodes. The 2 best connection among nodes make the graph more complex and there are no pieces of graph unconnected. The graph is accessible via the following URL address http://arm.mi.imati.cnr.it/papers/Kgraphs/graphs/cookIT_mostfreq_it_v1_bert__hL1_3_10_1_dimu_centroid_best_n2_pyvis2.html, 17 September 2024.

Figure 10 shows the elements of Cluster 08 related to Bread Blessing in the QueryLab_intangible_tags dataset. All elements within this cluster are related to either food or blessings. The analysis was performed using the ROMU language model with hyperparameters set to n_neighbor = 3 and bandwidth = 0.5. The clustering algorithm used was Mean Shift. This clustering highlights the specific thematic focus on food and blessings within the dataset.



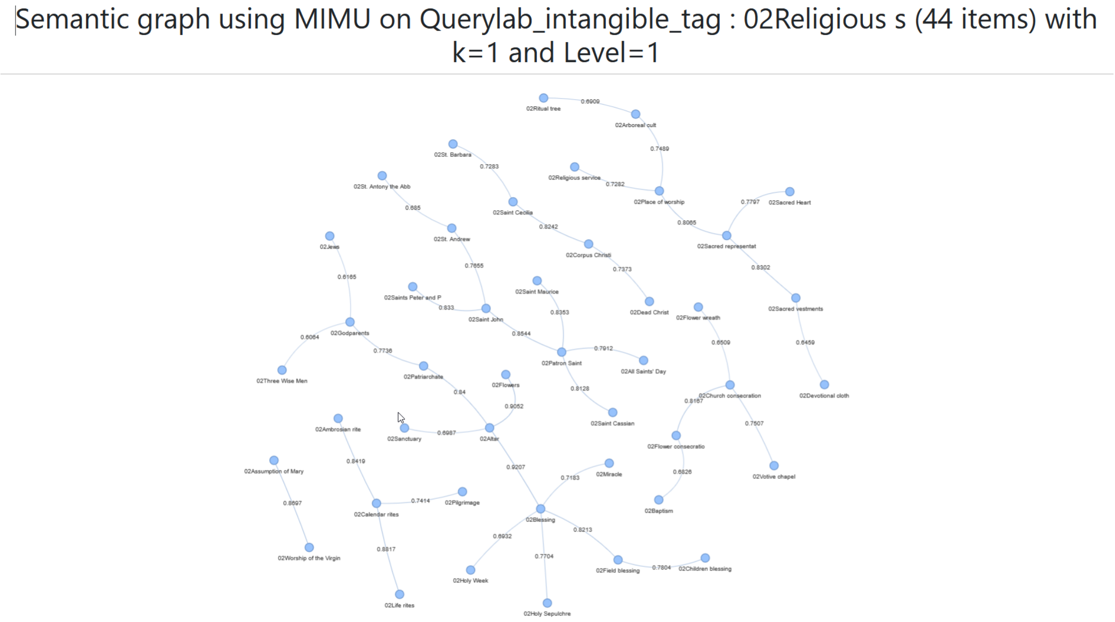
Semantic graph using ROMU on QueryLab_intangible_tag : 08Bread bless (11 items) with k=2 and Level=1

Figure 10. Elements of Cluster 08, related to Bread Blessing, on the QueryLab_intangible_tags. The LLM is ROMU and the hyperparameters are n_neighbor 3 and bandwidth 0.5. The algorithm is Mean Shift. The graph is accessible via the following URL address http://arm.mi.imati.cnr.it/papers/Kgraphs/graphs/querylab_intangible_tag_it_v1_bert__mL1_3_0.5_romu_cluster_08Bread%20bl_best_n2_pyvis2.html, 17 September 2024.

Figure 11 shows the elements of Cluster 02, related to religion, in the QueryLab _intangible_tags dataset. All 44 elements within this cluster are related to religion, saints, and church. The analysis was performed using the MIMU language model with hyperparameters set to n_neighbor = 15, min_cluster_size = 15, and min_samples = 5. The applied clustering algorithm was HDBSCAN. It is noteworthy that setting k = 1 results in some nodes remaining unconnected.

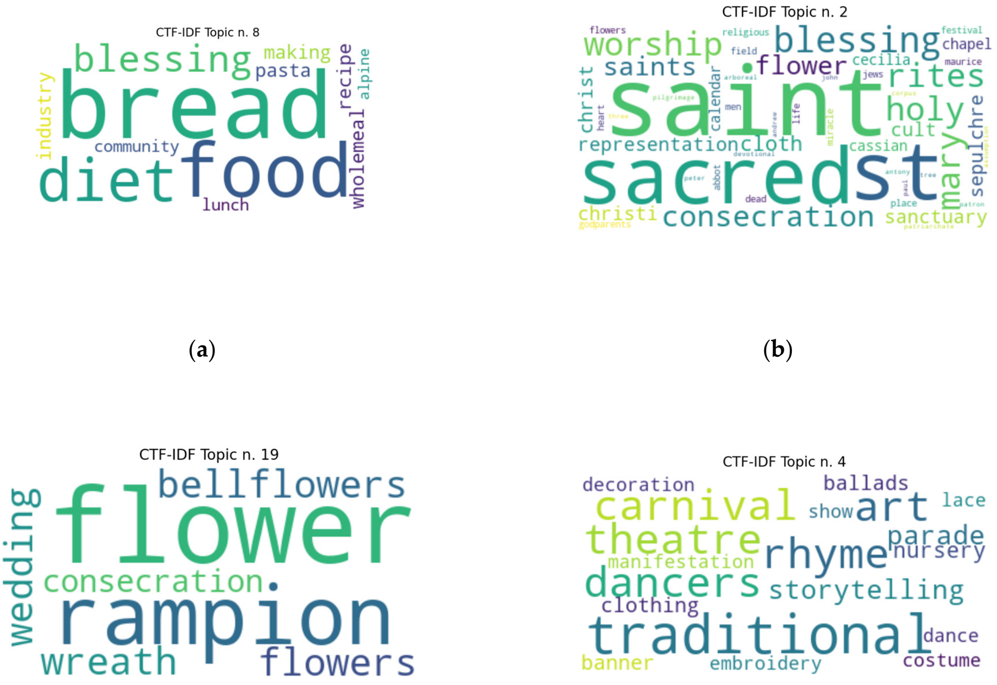
After clustering the documents, we extract keywords for each cluster and generate word cloud visualizations. These visualizations employ the c-TF-IDF algorithm, which gives higher weight to terms that are more prominent within a specific cluster compared to the entire document set. This method highlights the most significant terms for each cluster. Additionally, it is evident that each cluster contains items in various languages. Visualizing word clouds for the clusters enables us to easily assess cluster diversity, similar to the matrices shown in Figures 2–6.

Figure 12 illustrates the c-TF-IDF terms of clusters generated by either the HDBSCAN or Mean Shift algorithms, providing insight into the distinctive terms associated with each cluster. Figure 12a shows the c-TF-IDF terms for the same cluster as Figure 10, focusing on elements related to the blessing of bread. Figure 12b illustrates the terms for Cluster 2, all of which are associated with religion, corresponding to the elements discussed in Figure 11. Figure 12c shows Cluster 19, providing a comparison to Figure 12a and demonstrating the differences between clusters under the same configuration. Figure 12d shows the terms for Cluster 4, which can be inferred to be related to art, theater, and traditional dance.



Semantic graph using MIMU on Querylab_intangible_tag : 02Religious s (44 items) with k=1 and Level=1

Figure 11. Elements of Cluster 02, related to religion, on the QueryLab_intangible_tags. The LLM is MIMU and the hyperparameters are n_neighbors 15, min_cluster_size 15, and min_samples 5. The algorithm is HDBSCAN. The graph is accessible via the following URL address http://arm.mi.imati.cnr.it/papers/Kgraphs/graphs/querylab_intangible_tag_it_v1_bert__hL1_15_15_5_mimu_cluster_02Religiou_best_n1_pyvis2.html, 17 September 2024.



(**a**) (**b**)

(**c**) (**d**)

Figure 12. c-TF-IDF terms of clusters using HDBSCAN or Mean Shift. (**a**) c-TF-IDF terms of Cluster 8 using Mean Shift (same as Figure 10). (**b**) c-TF-IDF terms of Cluster 2 using HDBSCAN (same as Figure 11). (**c**) c-TF-IDF terms of Cluster 19 using Mean Shift. (**d**) c-TF-IDF terms of Cluster 4 using HDBSCAN.

The difference in the number of words between the figures on the left (Figure 12a,c) and those on the right (Figure 12b,d) is due to the number of items within each cluster. Clusters 8 and 19 contain fewer than 10 items, resulting in fewer terms, whereas the clusters on the right have a significantly larger number of items, resulting in a richer set of associated terms. This illustrates how the density and size of clusters affect the breadth of their respective c-TF-IDF terms.

In this prototype phase, we focused on two main aspects for subjective evaluation: (1) whether users found navigating the archive through the graph useful and engaging, and (2) whether multi-level graphs were effective, especially considering the presence of semantically distant elements within the same cluster at the higher level, which could be difficult for web users to interpret. Initial qualitative evaluations showed a positive response to both aspects, albeit to varying degrees. Feedback from heritage professionals and web users highlighted the simplicity and usability of the graph visualization as key strengths. However, the multi-level visualization was sometimes difficult for users to understand. In addition, even low-level clusters sometimes contained unrelated items, while related items were sometimes spread across multiple clusters, particularly when the number of clusters was not well suited.

5. Discussion

This study describes a prototype for the unsupervised construction of semantic graphs. The primary factors influencing the results are discussed below.

5.1. Visual Representation of Data

The proposed method provides users with a visual representation of archives or datasets, facilitating engaging and intuitive exploration. These graphs visually represent relationships and connections between various elements, making the exploration and understanding of intangible heritage information easier for researchers and the web users alike [49,50].

Another advantage is the dynamic representation these graphs offer. Unlike static lists or databases, navigation network graphs are interactive, allowing users to explore different nodes and edges. This interaction helps uncover hidden connections and provides deeper insights into the cultural and historical contexts of intangible heritage items. This interactivity also increases user engagement, as individuals can actively participate in the discovery process, fostering a greater interest in preserving and promoting intangible heritage [51,52].

User feedback confirmed these results, emphasizing that the playful and interactive aspects are highly appreciated after an initial adjustment period, regardless of age or background.

However, while comprehensive data visualization is beneficial, it can lead to information overload. Users might find it challenging to navigate large and dense graphs, potentially missing critical information or feeling overwhelmed by the volume of data [53]. To address information overload, we structured the clusters hierarchically to simplify navigation. This layered approach allows users to start with a high-level overview and zoom in on specific information as needed. Users can view these clusters as single entities that can be expanded or collapsed. This reduces visual clutter and allows users to focus on one cluster at a time, simplifying navigation and improving comprehension.

Moreover, accessing and interacting with navigation network graphs may require specific technical knowledge, limiting accessibility for users without the necessary resources or expertise, especially in regions with limited technological advancement [54].

The decision to create simple HTML pages or PNG images (such as the current word clouds) that can be easily integrated into web portals such as QueryLab or CookIT addresses both technical and cognitive limitations, thus broadening the potential target audience. These choices are in line with the method's philosophy, which emphasizes the use of state-of-the-art tools and techniques in a way that is easy to implement and use, ultimately improving the user experience.

5.2. Optimal Size of Elements in Graphs

The optimal size of elements in a knowledge graph for cultural heritage applications typically ranges from a few elements to less than a few hundred. This range ensures clear visualization and manageable navigation, which is critical for users exploring complex relationships within cultural heritage datasets. Knowledge graphs within this node range maintain the clarity of relationships between nodes, avoiding overwhelming users with too much complexity [55,56]. However, the specific optimal size can vary based on the analysis objectives and the granularity of the data.

- Effective visualization tools are crucial in cultural heritage contexts to ensure that users can easily navigate and understand the data.
- According to Das et al. [57], balancing visual clarity with the volume of data is essential. Interactive features like zooming, filtering, and subgraph exploration enhance user engagement and comprehension, which is particularly important for audiences who may not be experts in data analysis.
- The level of detail, or granularity, represented in the knowledge graph impacts its size. Coarser granularity might simplify the graph, resulting in fewer nodes, while finer granularity increases the number of nodes to capture detailed relationships [58].

Taking these considerations into account and testing graphs with different densities, we empirically found that a number of elements between 20 and 30 allows for easy interaction for our contexts, users, and purposes. We have also observed that, in the case of homogeneous graphs (such as those at the lowest level), the number of elements can be higher, as long as the number of edges is limited. In cases with larger numbers of elements with multiple links, we opted for iterative clustering solutions, resulting in multi-level graphs.

5.3. Selection of the LLM

The proposed system leverages a large language model to construct knowledge graphs, specifically employing models like Bert due to their state-of-the-art performance in natural language understanding and generation. The selection of the LLM is challenging and has been guided by the following criteria:

- Accuracy: BERT and similar models have demonstrated exceptional accuracy against various benchmarks, ensuring precise and reliable knowledge extraction from large datasets. High accuracy is vital for generating meaningful and correct relationships within the knowledge graph [17].
- Multilingual Support: The chosen models should support multiple languages, which is crucial for datasets that span different linguistic contexts. This capability ensures that the system can handle and integrate data from diverse sources without losing contextual integrity [18].
- Open-Source Nature: Utilizing open-source models like those provided by the Hugging Face Transformers library allows for greater transparency, flexibility in customization, and the ability to modify the underlying code to better fit specific use cases. This openness also fosters community collaboration and peer review, enhancing the model's robustness [59].
- Community and Vendor Support: Strong support from both the community and vendors ensures that the model remains up to date with the latest advancements and improvements. It also provides users with access to a wealth of resources, tutorials, and troubleshooting assistance, which is critical for maintaining and scaling the system effectively [60].

These criteria were identified to ensure that the pipeline remains reproducible and accessible, allowing other researchers and practitioners to effectively apply this solution to their own data.

With respect to the number of languages supported by the model, while both multilingual and monolingual models perform well in their respective domains, there are important structural differences between the two that influence their performance. Multilingual models are designed to handle multiple languages simultaneously, enabling cross-lingual transfer learning and flexibility in multilingual contexts. These models are trained on a variety of languages, allowing them to perform well on tasks involving multiple languages.

Monolingual models, on the other hand, are optimized for a single language and can capture more intricate linguistic structures and patterns specific to that language. This results in superior performance on tasks that are strictly within a single language. The structural divergence between multilingual and monolingual models can be attributed to the increased complexity of multilingual models, which often exhibit less faithful explanatory accuracy compared to monolingual models, because larger multilingual models must generalize across multiple languages, leading to different representations and potential reductions in model performance fidelity for individual languages [61].

The selection of LLMs used in this experiment was made in line with the considerations mentioned above, starting from the Hugging Face website and opting for solutions provided by Microsoft or Facebook (e.g., MIMU) or particularly high-performing models such as ROMU.

6. Conclusions and Future Works

In this paper, we presented a prototype for creating semantic graphs in an unsupervised manner. The ultimate goal is to integrate into websites or portals this new way of searching and browsing data in cultural heritage archives, using semantic graphs as a layered map with different granularity.

Users are presented with the contents of the archive in a simple visual representation, enabling them to explore and navigate the data in an engaging and intuitive way. Navigation network graphs offer enhanced accessibility to data. Users, experts in the field, and web users gave an initial positive qualitative assessment of the prototype, judging this overall view of the archive and each node positively. A more quantitative evaluation of the whole pipeline is being studied. As more relevant datasets and benchmarks will become available, we plan to conduct an extensive comparative analysis with existing methods, allowing for a broader evaluation of the effectiveness of our approach.

The complexity of creating and maintaining these graphs is a significant challenge. This process requires specialized expertise in data visualization and data management, which may not be readily available in all cultural institutions. The proposed pipeline seeks to mitigate this complexity by providing user-friendly tools that are adaptable to different environments.

Future work will focus on developing tools that facilitate user-driven navigation within the graphs, allowing seamless movement from one point to another. In addition, the implementation of fish-eye views will address graph density issues and improve visibility and comprehension.

In the future, we intend to integrate more evaluation measures to provide a more comprehensive assessment of the clustering quality and to guide the selection of appropriate algorithms and parameter settings.

Additionally, navigation network graphs play a critical role in data integration by bringing together heterogeneous datasets, regardless of their origin, language, or granularity. Future experiments will explore the effectiveness of this prototype using multilingual datasets to assess its robustness.

The integration of feedback mechanisms will allow users to report problems and suggest improvements, thereby identifying common usability challenges. These strategies can significantly improve navigation through large, complex graphs, minimizing the risk of information overload and enriching the overall user experience.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics13183741/s1, Complete results of HDBSCAN and Mean Shift algorithms on the datasets.

Author Contributions: Conceptualization I.G.; methodology, I.G.; software, I.G. and M.T.A.; writing—original draft preparation, I.G.; writing—review and editing, I.G. and M.T.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data publicly available. The data will be made available upon request to isabella.gagliardi@cnr.it.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hogan, A.; Blomqvist, E.; Cochez, M.; D'amato, C.; De Melo, G.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge Graphs. *ACM Comput. Surv.* **2022**, *54*, 1–37. [CrossRef]
2. Ehrlinger, L.; Wöß, W. Towards a Definition of Knowledge Graphs. *SEMANTiCS* **2016**, *48*, 2.
3. Hao, X.; Ji, Z.; Li, X.; Yin, L.; Liu, L.; Sun, M.; Liu, Q.; Yang, R. Construction and Application of a Knowledge Graph. *Remote Sens.* **2021**, *13*, 2511. Available online: https://www.mdpi.com/2072-4292/13/13/2511 (accessed on 17 September 2024). [CrossRef]
4. Achour, A.; Al-Assaad, H.; Dupuis, Y.; El Zaher, M. Collaborative Mobile Robotics for Semantic Mapping: A Survey. *Appl. Sci.* **2022**, *13*, 10316. [CrossRef]
5. Ryen, V.; Soylu, A.; Roman, D. Building Semantic Knowledge Graphs from (Semi-)Structured Data: A Review. *Future Internet* **2022**, *14*, 129. [CrossRef]
6. Carriero, V.A.; Gangemi, A.; Mancinelli, M.L.; Marinucci, L.; Nuzzolese, A.G.; Presutti, V. ArCo: The Italian Cultural Heritage Knowledge Graph. In *The Semantic Web—ISWC 2019*; Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 36–52. [CrossRef]
7. SWODCH 2022. Semantic Web and Ontology Design for Cultural Heritage. Available online: https://swodch2022.inf.unibz.it/ (accessed on 17 July 2024).
8. Asprino, L.; Daga, E.; Gangemi, A.; Mulholland, P. Knowledge Graph Construction with a Façade: A Unified Method to Access Heterogeneous Data Sources on the Web. *ACM Trans. Internet Technol.* **2023**, *23*, 131. [CrossRef]
9. Kokash, N.; Romanello, M.; Suyver, E.; Colavizza, G. The Brill Knowledge Graph: A database of bibliographic references and index terms extracted from books in humanities and social sciences. *Res. Data J. Humanit. Soc. Sci.* **2024**, *1*, 1–21. Available online: https://brill.com/view/journals/rdj/aop/article-10.1163-24523666-bja10036/article-10.1163-24523666-bja10036.xml (accessed on 17 July 2024). [CrossRef]
10. Jhajj, G.; Zhang, X.; Gustafson, J.R.; Lin, F.; Lin, M.P.-C. Educational Knowledge Graph Creation and Augmentation via LLMs. In *Generative Intelligence and Intelligent Tutoring Systems*; Sifaleras, A., Lin, F., Eds.; Springer Nature: Cham, Switzerland, 2024; pp. 292–304. [CrossRef]
11. Gweon, H.; Schonlau, M. Automated classification for open-ended questions with BERT. *J. Surv. Stat. Methodol.* **2024**, *12*, 493–504. [CrossRef]
12. George, L.; Sumathy, P. An integrated clustering and BERT framework for improved topic modeling. *Int. J. Inf. Tecnol.* **2023**, *15*, 2187–2195. [CrossRef]
13. Rao, A.; Halgekar, A.; Khankhoje, D.; Khetan, I.; Bhowmick, K. Legal Document Clustering and Summarization. In Proceedings of the 2022 6th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 26–27 August 2022; pp. 1–4. [CrossRef]
14. Clément, P.; Bouleux, G.; Cheutet, V. Improved Time-Series Clustering with UMAP dimension reduction method. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 5658–5665. [CrossRef]
15. Petukhova, A.; Matos-Carvalho, J.P.; Fachada, N. Text clustering with LLM embeddings. *arXiv* **2024**, arXiv:2403.15112. [CrossRef]
16. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805. Available online: http://arxiv.org/abs/1810.04805 (accessed on 6 May 2024).
17. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: New York, NY, USA,

2020; pp. 1877–1901. Available online: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html (accessed on 17 July 2024).

18. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv* **2020**, arXiv:1911.02116. [CrossRef]
19. Lepikhin, D.; Lee, H.J.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; Chen, Z. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. *arXiv* **2020**, arXiv:2006.16668. [CrossRef]
20. Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *arXiv* **2017**, arXiv:1701.06538. [CrossRef]
21. Koufakou, A. Deep learning for opinion mining and topic classification of course reviews. *Educ. Inf. Technol.* **2024**, *29*, 2973–2997. [CrossRef]
22. Parfenova, A. Automating the Information Extraction from Semi-Structured Interview Transcripts. In *Proceedings of the WWW '24: Companion Proceedings of the ACM Web Conference 2024*; Singapore, 13–17 May 2024, Association for Computing Machinery: New York, NY, USA, 2024; pp. 983–986. [CrossRef]
23. Ogunleye, B.; Maswera, T.; Hirsch, L.; Gaudoin, J.; Brunsdon, T. Comparison of topic modelling approaches in the banking context. *Appl. Sci.* **2023**, *13*, 797. [CrossRef]
24. Sprenkamp, K.; Zavolokina, L.; Angst, M.; Dolata, M. Data-Driven Governance in Crises: Topic Modelling for the Identification of Refugee Needs. In Proceedings of the 24th Annual International Conference on Digital Government Research, Gdansk, Poland, 11–14 July 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1–11. [CrossRef]
25. Wojciechowska, J.; Sypniewski, M.; Śmigielska, M.; Kamiński, I.; Wiśnios, E.; Schreiber, H.; Pieliński, B. Deep Dive into the Language of International Relations: NLP-based Analysis of UNESCO's Summary Records. *arXiv* **2023**, arXiv:2307.16573. Available online: http://arxiv.org/abs/2307.16573 (accessed on 6 May 2024).
26. Egger, R.; Yu, J. A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. *Front. Sociol.* **2022**, *7*, 886498. [CrossRef]
27. Asyaky, M.S.; Mandala, R. Improving the Performance of HDBSCAN on Short Text Clustering by Using Word Embedding and UMAP. In Proceedings of the 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Bandung, Indonesia, 29–30 September 2021; pp. 1–6. [CrossRef]
28. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* **2020**, arXiv:1802.03426. Available online: http://arxiv.org/abs/1802.03426 (accessed on 6 May 2024).
29. McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*, 205. [CrossRef]
30. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799. [CrossRef]
31. Jain, A.K.; Murty, M.N. Data Clustering: A Review. *ACM Comput. Surv.* **1999**, *31*, 264–323. [CrossRef]
32. Aggarwal, C.C. *Data Mining: The Textbook*; Springer International Publishing: Cham, Switzerland, 2015. [CrossRef]
33. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30, Available online: https://proceedings.neurips.cc/paper/7181-attention-is-all (accessed on 6 May 2024).
34. Bostrom, K.; Durrett, G. Byte Pair Encoding is Suboptimal for Language Model Pretraining. *arXiv* **2020**, arXiv:2004.03720. [CrossRef]
35. Grootendorst, M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv* **2022**, arXiv:2203.05794. Available online: http://arxiv.org/abs/2203.05794 (accessed on 6 May 2024).
36. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; Available online: https://nlp.stanford.edu/IR-book/ (accessed on 18 July 2024).
37. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
38. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat.* **1974**, *3*, 1–27. [CrossRef]
39. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227. [CrossRef]
40. Dunn, J.C. Well-Separated Clusters and Optimal Fuzzy Partitions. *J. Cybern.* **1974**, *4*, 95–104. [CrossRef]
41. QueryLab Portal: Explore the Intangible With Us! Available online: https://querylab.imati.cnr.it/home_page.php?status=start (accessed on 18 July 2024).
42. Artese, M.T.; Gagliardi, I. Integrating, Indexing and Querying the Tangible and Intangible Cultural Heritage Available Online: The QueryLab Portal. *Information* **2022**, *13*, 260. [CrossRef]
43. Artese, M.T.; Ciocca, G.; Gagliardi, I. CookIT: A Web Portal for the Preservation and Dissemination of Traditional Italian Recipes. *Int. J. Humanit. Soc. Sci.* **2019**, *13*, 171–176.
44. CookIT Online Archive—Ricette e Immagini Della Cucina Tradizionale Italiana. Available online: https://arm.mi.imati.cnr.it/cookIT/open_home_page.php (accessed on 18 July 2024).
45. Teresa, A.M.; Isabella, G. Inventorying intangible cultural heritage on the web: A life-cycle approach. *Int. J. Intang. Herit.* **2017**, *12*, 112–138. Available online: https://scholar.google.com/citations?view_op=view_citation&hl=it&user=Sb6ZHAEAAAAJ&cstart=20&pagesize=80&sortby=pubdate&citation_for_view=Sb6ZHAEAAAAJ:fbc8zXXH2BUC (accessed on 18 July 2024).

46. UNESCO—Identifying and Inventoring Intangible Cultural Heritage. Available online: https://ich.unesco.org/doc/src/01856-EN.pdf (accessed on 18 September 2024).
47. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems. NIPS'13, Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013*; Curran Associates Inc.: Red Hook, NY, USA, 2013; pp. 3111–3119.
48. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
49. Marchand, E.; Gagnon, M.; Zouaq, A. Extraction of a Knowledge Graph from French Cultural Heritage Documents. In Proceedings of the ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, Lyon, France, 25–27 August 2020; Bellatreche, L., Bieliková, M., Boussaïd, O., Catania, B., Darmont, J., Demidova, E., Duchateau, F., Hall, M., Merčun, T., Novikov, B., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 23–35. [CrossRef]
50. Ranjgar, B.; Sadeghi-Niaraki, A.; Shakeri, M.; Rahimi, F.; Choi, S.-M. Cultural Heritage Information Retrieval: Past, Present, and Future Trends. *IEEE Access* **2024**, *12*, 42992–43026. [CrossRef]
51. Dimoulas, C.A. Cultural Heritage Storytelling, Engagement and Management in the Era of Big Data and the Semantic Web. *Sustainability* **2022**, *14*, 812. [CrossRef]
52. Bikakis, A.; Hyvönen, E.; Jean, S.; Markhoff, B.; Mosca, A. Editorial: Special issue on Semantic Web for Cultural Heritage. *Semant. Web* **2021**, *12*, 163–167. [CrossRef]
53. Enhancing the Functionality of Augmented Reality Using Deep Learning, Semantic Web and Knowledge Graphs: A Review—ScienceDirect. Available online: https://www.sciencedirect.com/science/article/pii/S2468502X20300012 (accessed on 17 July 2024).
54. Calzolari, N.; Kan, M.-Y.; Hoste, V.; Lenci, A.; Sakti, S.; Xue, N. Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). ELRA and ICCL, Turin, Italy, 20–25 May 2024; Available online: http://hdl.handle.net/1854/LU-01HZSKHKPD07MY5AFTEZTGBY5N (accessed on 17 July 2024).
55. Schulz, H.-J.; Schumann, H. Visualizing Graphs—A Generalized View. In Proceedings of the Tenth International Conference on Information Visualisation (IV'06), London, UK, 5–7 July 2006; pp. 166–173. [CrossRef]
56. Khemani, B.; Patil, S.; Kotecha, K.; Tanwar, S. A review of graph neural networks: Concepts, architectures, techniques, challenges, datasets, applications, and future directions. *J. Big Data* **2024**, *11*, 18. [CrossRef]
57. A Key Review on Graph Data Science: The Power of Graphs in Scientific Studies—ScienceDirect. Available online: https://www.sciencedirect.com/science/article/pii/S0169743923001466 (accessed on 17 July 2024).
58. Zhang, L.; Liu, P.; Gulla, J.A. Recommending on graphs: A comprehensive review from a data perspective. *User Model User-Adap. Inter.* **2023**, *33*, 803–888. [CrossRef]
59. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; Liu, Q., Schlangen, D., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 38–45. [CrossRef]
60. Alec, R. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* **2019**, *1*, 9.
61. Ruzzetti, E.; Ranaldi, F.; Logozzo, F.; Mastromattei, M.; Ranaldi, L.; Zanzotto, F. Exploring Linguistic Properties of Monolingual BERTs with Typological Classification among Languages. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP Singapore, Singapore, 6–10 December 2023; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 14447–14461. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.