applied sciences

MDPI

Article

Testing Pretrained Large Language Models to Set Up a Knowledge Hub of Heterogeneous Multisource Environmental Documents

Paolo Tagliolato Acquaviva d'Aragona [1,*,†], Gloria Bordogna [1,†], Lorenza Babbini [2], Alessandro Lotti [2], Annalisa Minelli [2], Martina Zilioli [1] and Alessandro Oggioni [1]

[1] Institute for Remote Sensing of Environment (IREA), National Research Council (CNR), Via A. Corti 12, 20133 Milano, Italy; martina.zilioli@cnr.it (M.Z.); alessandro.oggioni@cnr.it (A.O.)
[2] INFO/RAC UNEP/MAP c/o ISPRA, DG-SINA, Via Vitaliano Brancati 48, 00144 Roma, Italy; lorenza.babbini@isprambiente.it (L.B.); annalisa.minelli@isprambiente.it (A.M.)
* Correspondence: tagliolato.p@irea.cnr.it
† These authors contributed equally to this work.

Abstract: This contribution outlines the design of a Knowledge Hub of heterogeneous documents related to the UNEP/MAP Barcelona Convention system. The Knowledge Hub is intended to serve as a resource to assist public authorities and users with different backgrounds and needs in accessing information efficiently; users should be able to either formulate natural language queries or to navigate a knowledge graph that is automatically generated to find relevant documents. The ad hoc retrieval task and the Knowledge Hub creation are defined based on state-of-the-art Large Language Models (LLMs). Specifically, this contribution focuses on a user-evaluation experiment that tested publicly available pretrained foundation Large Language Models (LLMs) for retrieving a subset of documents with varying lengths and topics.

Keywords: knowledge hub; heterogeneous documents with highly variable length; foundation large language models; natural language queries; knowledge graph

Academic Editor: Jose María Alvarez Rodríguez

Received: 4 April 2025
Revised: 3 May 2025
Accepted: 6 May 2025
Published: 12 May 2025

Citation: Tagliolato Acquaviva d'Aragona, P.; Bordogna, G.; Babbini, L.; Lotti, A.; Minelli, A.; Zilioli, M.; Oggioni, A. Testing Pretrained Large Language Models to Set Up a Knowledge Hub of Heterogeneous Multisource Environmental Documents. *Appl. Sci.* **2025**, *15*, 5415. https://doi.org/10.3390/app15105415

1. Introduction

This contribution describes an approach proposed to design a Knowledge Hub (KH) within the framework of the Knowledge Management Platform (KMaP) for the Mediterranean Action Plan of the United Nations Environmental Programme (UNEP/MAP) [1,2]. The present work, by expanding [3], will focus on experiments with Large Language Models to cope with specific needs of the KH construction. The KH should serve as a centralized access point for a diverse range of multimedia documents related to marine studies, political and economic directives, environmental studies, and other protocols and activities under the UNEP/MAP [2]. Given the diverse nature of these documents, which include meeting minutes, scientific reports, and documents of varying lengths and formats (such as PDF, HTML, and JPG), as well as multiple languages, the creation of a KH intended to function as a knowledge repository for stakeholders of the Mediterranean Action Plan is challenging. These stakeholders encompass policymakers, administrators, environmental scientists, project leaders, and citizens, each with distinct knowledge backgrounds and information needs. They require the ability to not only search but also navigate through this distributed archive [4], particularly by developing and exploiting a fuzzy classification of documents into seven predefined "themes" derived from the UNEP/MAP keywords and definitions that are regarded as "topics" to which each document should be attributed.

During the use case analysis [5], it became evident that natural language queries and structured navigation through document topics should be supported by the KH to assist users in accessing information resources [4]. To address these requirements, our approach is grounded in an Information Retrieval system [6] based on Large Language Models (LLMs), specifically leveraging open-source pretrained foundation LLMs [7]. In this direction, the first step for the KH creation consists of the identification of the best performing foundation LLM for the ad hoc retrieval task using natural language queries. Once this has been identified, in order to organize documents into the preexisting topics, the second step is to classify the documents into the topics by exploiting natural language descriptions of each topic to be used as distinct queries to retrieve the ranked lists of documents relevant to the topics.

This approach would serve to create a kind of knowledge graph, where each node represents a ranked list of documents related to a specific topic, and edges represent intersections between topics, thus supporting an organized structure to facilitate user navigation through the document collection [8]. There are several challenges of designing and implementing such a KH:

- First, the application of pretrained foundation LLMs for the ad hoc retrieval task of heterogeneous documents with highly variable length is still an open issue [9];
- Second, the following need to be investigated: the identification of the most effective combination of pretrained LLMs and the settings of both the document segments, named chunks, that are represented by embedding vectors, and the similarity matching function.
- Ultimately, the idea of constructing a kind of knowledge graph in which nodes represent concepts (topic meanings in this case) pertinent to a specific application, as described in various documents, mirror the fusion of different instances of these concepts discussed in each document. This last aspect is only proposed herein at the theoretical level and is not yet implemented.

This contribution focuses on the first step of the KH creation, that is, on the user evaluation of publicly available pretrained foundation LLMs, without modifying the default parameters of these models, but rather by varying the settings of the post hoc pooling strategy used to combine the chunks' relevance scores to determine the documents' relevance. By considering various combinations of the chunks' definitions, similarity metrics, and pooling function, we tested the accuracy of the various strategies in retrieving a subset of documents of the collection. The aim is to identify the most effective LLM for the ad hoc retrieval task supporting natural language queries on the collection, which would then be used for implementing the fuzzy classification of documents by topic.

The next section describes the model and document collection, and the results detail the user-evaluation experiment, emphasizing the selection of the optimal model for future fuzzy classification of documents into topics.

2. Materials and Methods

In this section, the characteristics of the documents' collection and the methods applied to harvest them (Section 2.1) so as to enable their search (Section 2.2) and fuzzy classification into topics (Section 2.3) are described.

2.1. Harvesting Documents' Collection

The documents of the collection originated from several sources, that is, websites and archives, with potentially interesting documents dealing with environmental information [8]. More specifically, the sources are the libraries of Regional Activity Centres (RACs) composing of the UNEP/MAP ecosystem, namely the Regional Marine Pollution Emergency Response Centre for the Mediterranean Sea (REMPEC) [10], the Regional Ac-

tivity Centre for Specially Protected Areas (SPA/RAC) [11], the Regional Activity Centre for Sustainable Consumption and Production (SCP/RAC) [12], the Priority Actions Programme/Regional Activity Centre (PAP/RAC) [13], the UNEP/MAP library [14], and the UNEP library [15]. From these sources, by applying website scraping with a deep-first strategy, all documents were harvested. To this end, CNR-IREA developed programs in R language and INFO-RAC in Python 3 language [16,17], both freely available under the GNU GPL license. After document harvesting, document characterization was performed. Most of the considered sources (18 out of 24) contain textual documents, 3 also have images and tables, while the remaining 3 provide geographical layers. The total number of documents from these sources is more than 12,000, mainly files, most of which are in PDF format dealing with 3 themes: (i) law, (ii) regulation, and management of the sea (13 out of 24), (iii) pollution (7) and biodiversity (2). Finally, 21 of the classified sources are open to the public, while the remaining 3 are private or have restricted access.

To share the files collected through the harvesting process, a GitHub repository was created [17]. The GitHub "scraping" folder contains the R and Python scripts developed for harvesting, while the harvested files can be found in the "results" folder.

2.2. Enabling Document Search

Once the collection was available, the first task was to enable the document search. This first implies indexing the content of documents by applying an off-line procedure to generate the documents' representation, which can be made available to a subsequent on-line query evaluation process that computes a list of the relevant documents for each user query. It was decided to experiment with the most up-to-date indexing methods using a latent "semantic" representation of documents in an embedding space the continuous bag-of-words paradigm [7]. The upper part of Figure 1 sketches the main phases of two processes: off-line document indexing via the creation of document embedding vectors and on-line query evaluation, which exploits the same components for the creation of query embeddings.

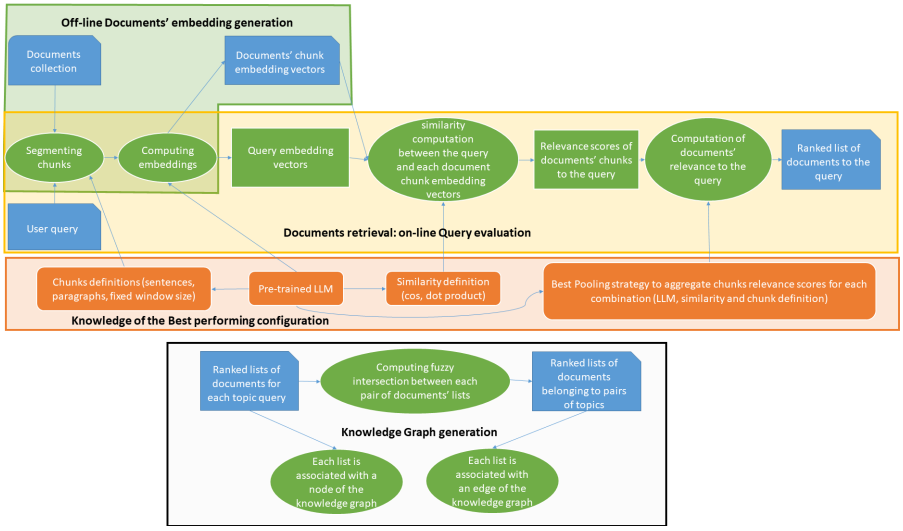

Figure 1. Sketch of the three processes to generate the Knowledge Hub: at the top, the off-line indexing that creates the documents' representation consisting of the document embedding vectors and the query evaluation process; below is the knowledge that is exploited to apply the best combination of LLM, chunk definitions, and the similarity function to implement the document retrieval; at the bottom, the process to generate the knowledge graph to aid users in browsing the collection.

In fact, in this context, both natural language queries and keyword queries are also presented in the same embedding space in order to implement the ad hoc retrieval task. It consists of retrieving documents if their embedding vectors, that is, vectors of continuous

numeric values in the latent semantic space, are close to the query vectors. Here, the concept of "closeness" resembles a "semantic" similarity. In order to avoid training a Large Language Model (LLM) from scratch to create the embedding space, which is costly, several pretrained foundation LLMs publicly available on the Hugging face library were experimented [18]. LLMs are generated by self-training a deep neural network requiring both huge computational resources and a big textual corpus, which has been provided as a training set. It should be pointed out that the term "semantics" is improperly used here, as LLMs identify regular patterns in texts based on heuristic statistical inference; thus, instead of "semantics", the term "relatedness" or "co-occurrency" would be more appropriate.

During the training phase, the deep neural network learns the characteristics of a natural language so as to be able to predict missing words in a sentence, continue a sentence, or answer a query, and, finally, to retrieve relevant documents in an ad hoc retrieval task activated by a user query. Such "semantic" models are the most effective in the case where one wants a natural language query interaction since documents which do not contain the specific query words can be retrieved, using only synonymous terms or concepts related with the query concepts.

In the context of the UNEP/MAP project, the adoption of a pretrained foundation LLM was the most feasible solution to implement since we did not have the available thesauri for expanding the meaning of terms in the documents, which are heterogeneous in both their themes and genre. We also lacked the huge computational resources needed to perform the training of an LLM from scratch.

Pretrained LLMs based on the evolution of BERT (Bidirectional Encoder Representations from Transformers [19,20]) were selected for the experiment. BERT is a state-of-the-art model by Google that uses a transformer architecture [19], i.e., a deep neural network with self-attention mechanisms, which allows keeping the context of words into account when creating their representation as embedding vectors.

This allows for coping with polysemous terms, which the model represents by distinct embedding vectors. Nevertheless, the applications of BERT to deal with long documents are not straightforward. In fact, while BERT and its extensions have been successful in dealing with short text-based applications (usually limited to 512 tokens), their applications to long documents have been addressed in the literature based on two categories of pooling strategies [9]. The first strategy segments the documents into chunks, defined as either sentences or passages, and computes the final document score by aggregating the relevance scores retrieved by BERT for the chunks, where the aggregation function is fixed as either the maximum, the top three scores, the sum, or the average of all relevance scores as the aggregation function [9]. The second strategy selects a subset of document chunks, and the final document score is computed as the maximum of the scores for the selected chunks [21]. We have decided to define a novel approach that mixes the two previous methods by testing different combinations of chunk definitions, similarity functions, and aggregation functions in order to identify the one that yields the best performance. These tests are described in the user-evaluation experiments below and are aimed at identifying the knowledge sketched in Figure 1 that best supports document retrieval.

To this end, once the LLMs were selected, the pre-processing operations that the corpus of documents should undergo to become a readable input to the selected models were defined and implemented. Such models accept the following as input: simple text with punctuation marks, allowing for the identification of single words, i.e., tokens; sentences ending with punctuation marks, for instance, full stop or semicolon; and paragraphs, starting with a new line. Moreover, the input text shall have a fixed limited length, typically of 512 tokens. Thus, the first operation was to transform the non-conforming documents consisting of PDF files into text, and then to define a segmentation of the documents' text

so as to represent each document by a given number of chunks of text, where each chunk could be represented by a distinct embedding vector using a pretrained LLM. To this end, hybridized techniques were implemented, for example, the contents of queries and documents were represented by applying different embedding methods and chunk definitions, and different similarity measures were experimented for computing the documents' ranks when matching the vectors of the documents and the queries. These processing steps have implied the selection of the implementation libraries and environment in order to code the whole process, including the indexing, the retrieval, and the fuzzy classification components of the KH. Considering that there is a number of open-source IR libraries, after a review, the SentenceTransformer Python framework [22] was selected as it makes several Hugging face pretrained models available for sentence embeddings. Moreover, the Python library NLTK (Natural Language Toolkit [23]) was exploited for splitting the documents into chunks, particularly sentences, paragraphs, or n-gram windows. For the purposes of future KH implementation, different combinations of pretrained LLMs with default parameter settings for the network architecture (as defined in the Hugging phase library) and document representations based on different chunk definitions were considered, as well as two similarity matching functions, the dot product, and the cosine. Since documents may contain varying numbers of chunks depending on their length, several aggregation functions of the chunks' relevance scores were experimented to compute the overall document relevance score, i.e., the document ranking score. Specifically, a KNN algorithm aggregation function was applied for computing the documents' relevance scores RSV(d); this was performed by experimenting aggregations of the K highest chunks' relevance scores chunk_score, with increasing values of the parameter K, and by using as metrics the fuzzy document cardinality measure [24], which is defined as follows:

$$\text{RSV(d)} = \sum_{i=1}^{K} \text{chunk\_score}_i$$

with

$$\text{chunk\_score}_i \geq \text{chunk\_score}_j \forall i \leq j$$

Using this function, it is possible to investigate if there exists a unique value for the K parameter so that the ranking function provides the best results independently of the LLM used.

We have selected the following pretrained LLMs based on sentence-transformer architectures:

(a) **msmarco-distilbert-cos-v5** [22,25]: It maps sentences and paragraphs to a 768-dimensional dense vector space and was designed for "semantic" search. It has been trained on 500k (query, answer) pairs from the MS MARCO Passages dataset (Microsoft Machine Reading Comprehension), which is a large-scale dataset focused on machine reading comprehension, question answering, and passage ranking.

(b) **all-MiniLM-L6-v2** [26]: It maps sentences and paragraphs to a 384-dimensional dense vector space and can be used for tasks like clustering or "semantic" search.

(c) **msmarco-roberta-base-ance-firstp** [27,28]: This is a port of the ANCE FirstP model, which uses a training mechanism to select more realistic negative training instances to the sentence-transformer model. It maps sentences and paragraphs to a 768-dimensional dense vector space and can be used for tasks like clustering or "semantic" search.

(d) **msmarco-bert-base-dot-v5** [29]: It maps sentences and paragraphs to a 768-dimensional dense vector space and was designed for "semantic" search. It has been trained on 500K (query, answer) pairs from the MS MARCO dataset.

(e) **msmarco-distilbert-base-tas-b** [30]: It is a port of the DistilBert TAS-B model to the sentence-transformer model. It maps sentences and paragraphs to a 768-dimensional dense vector space and is optimized for the task of "semantic" search.

2.3. Approach for the Fuzzy Classification of Documents into Topics

With regard to the fuzzy classification of the document corpus into topics—which, in the present paper, is proposed but not yet implemented—the topics were identified during the use case analysis based on the predefined keywords of the seven UNEP/MAP-related themes listed in the UNESCO thesaurus [31], an RDF SKOS concept scheme without definitions, as reported in Table 1. Then, we identified "definitions" of each topic keyword, having the form of textual abstracts, in renowned and authoritative sources, as reported in Table 1, i.e., open domain websites. The pre-existing thesauri were enriched by adding those definitions in the web of data. The results are available both as linked data [32] and through a SPARQL endpoint [33].

Table 1. Topics and their source of definitions.

Topic Keyword	Definition Source
Climate change	United Nations (UN)
Marine biodiversity	UN
Sustainability and blue economy	UN
Pollution	National Geographic
Marine spatial planning	EU Commission
Fishery and aquaculture	FAO
Governance	UN Dev. Progr.

After choosing the best performing combination of LLMs, similarity function, and chunk definitions, evaluated as explained in the next section, we are able to apply it to classify the whole collection into the distinct topics by considering the topics' definitions as queries. This way, for each topic, we will obtain a distinct ranked list of documents, and thus, in principle, a document will be assigned to multiple topics to a different extent, where the extent is the document's relevance score with respect to a topic. The more similar two ranked lists of two distinct topics are—that is, they have the same order and relevance score for the documents—the more the two topics are considered "similar" with regard to their descriptions in the collection.

The fuzzy intersection of a pair of ranked lists (named A and B), which will be returned by querying the model with two topics, is the ranked list (AB) of documents at the cross-road of both the two topics. It is such that for each document d_i in the two ranked lists A and B, the rank of document d_i is computed as the minimum of the relevance scores of the document in the two lists A and B:

$$RSV_{AB}(d_i) = min(RSV_A(d_i), RSV_B(d_i)$$

This representation of the contents of documents resembles a knowledge graph in which the nodes correspond to the different topics, the relevant concepts of the KH, and their corresponding ranked lists are the manifestations of the concepts in the documents of the collection. On the other hand, the edges of the knowledge graph represent the similarity relationships between pairs of connected concepts at the nodes, i.e., concepts at the cross-road of pairs of topics, the manifestation of which is the ranked list of documents obtained by computing the fuzzy intersection of the ranked lists of the connected nodes.

The bottom part of Figure 1 reports the operations necessary to generate the knowledge graph. By conceiving such a kind of "knowledge graph", we aim to mirror the fusion of the manifestations of the relevant concepts dealt with in the collection. Figure 2 sketches the visual representation of the UNEP-MAP knowledge graph.

Creating such a knowledge graph can be costly in terms of time elapsed, since it involves evaluating the seven queries that represent the topics in the entire collection and then computing the fuzzy intersections of all pairs of ranked lists. Nevertheless, we do not need quick execution since this is performed once and for all in a preliminary off-line phase in order to prepare the KH for subsequent browsing by final users. This is the equivalent of an "indexing" process of a collection of documents to aid in their retrieval by topics. To manage memory consumption to compute the fuzzy intersection of the ranked lists, optimization strategies, such as disk-based pagination, could possibly be exploited.

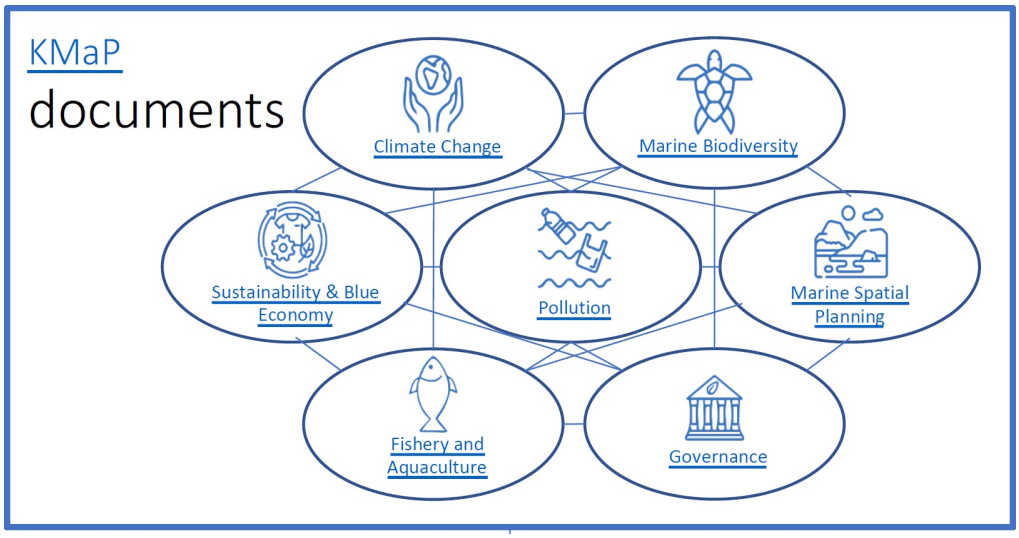

Figure 2. Visual representation of the UNEP-MAP knowledge graph.

3. Results: The User-Evaluation Experiment of the Pretrained LLMs

An evaluation experiment of the different LLMs was set up by first defining the ground truth for the comparison of the models' results with human judgments. To this end, 50 documents of the collection were randomly selected, and three users, with three different backgrounds (a physicist, an environmental scientist, and a biologist) were identified to create the ground truth. The users first read the documents; then, each of them formulated 10–30 queries, mimicking potential users of the system with their background knowledge; and finally, for each of their own queries, they identified the list of their relevant documents among the 50 samples. We noticed that the three users formulated their queries in such similar styles that it was difficult to distinguish who had authored each one. Most of the prompt was formulated as natural language sentences of varying lengths such as "remedies, directives and reports on oil spills and chemical spills pollution in Mediterranean sea" or "sea accidents and lesson learnt", or "pollution due to systems on ships, chemical substances and related accidents, plastics, heavy metals and their effects on lives health". Other queries were more in the style of keyword-based queries of up to three or four words, e.g., "tourism; industry; agriculture" or "waste management; dumping; dredging; ballast waters". The mean Average Precision (mAP) was deemed relevant to compute [34] as a relevant metric of the retrieval's effectiveness. The results obtained by applying several combinations of the five pretrained LLMs, i.e., from (a) to (e), document representations based on different chunk definitions, i.e., sentence, fixed window size, and paragraphs. Additionally, two matching functions (cosine and dot product) were evaluated. Figure 3 reports the average number of the chunks with

distinct type definitions in the 50 documents and their variability. It can be seen that the greatest variability of the number of chunks occurs for the chunk types defined as sentences and paragraphs, and then for chunks of window type ordered by increasing fixed length size. This means that sentences and paragraphs are generally shorter than the considered windows' sizes. Furthermore, regardless of the tokenization strategy, it is visible that the document length distribution is skewed to the bottom of the boxes (that is, the largest part of the collection is shorter than the mean length). Each box of the plot contains 50% of the documents (those from the 25th to the 75th percentile in length); e.g., in the first distribution on the left of the figure, 50% of the collection is between approximately 20 and 500 sentences, while the mean length of documents is around 100 sentences. The outliers appear as isolated points, representing very long, yet less frequent documents.

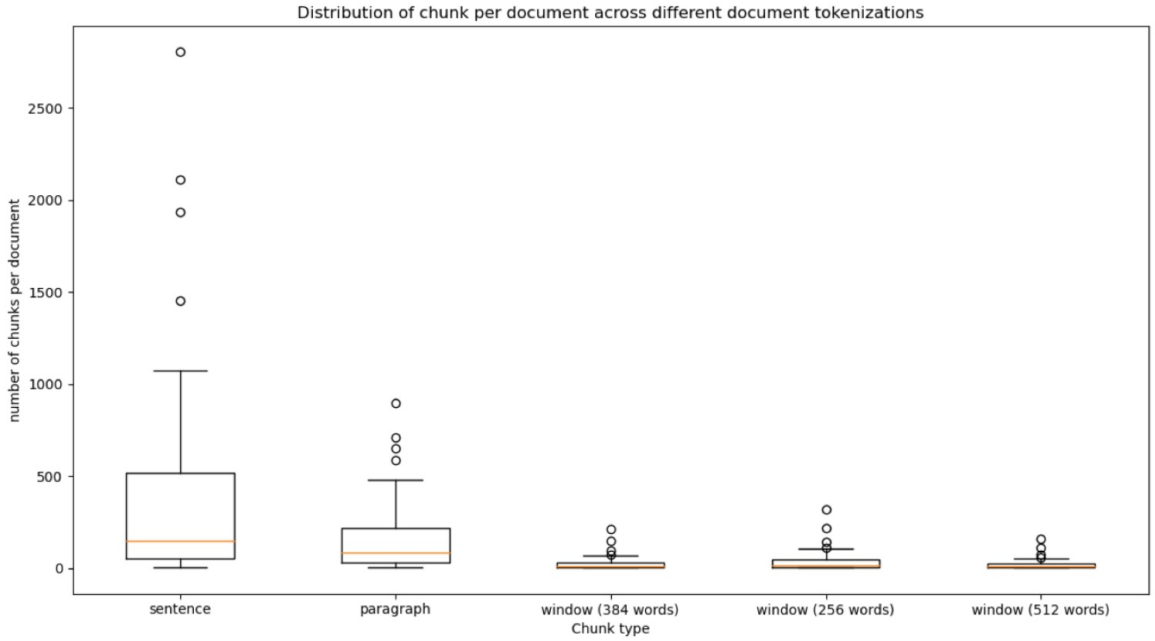

Figure 3. This graph reports the average number of the chunks and their variability for distinct definitions of chunk types (sentence, paragraph, and fixed window size with three distinct lengths) in the 50 documents of the ground truth set.

The results of the mAP for the tests are reported in the table shown in Figures 4 and 5. They differ in the computation of similarity matching function. The results shown in Figure 4 were obtained by computing the cosine similarity between the query vector and the vectors of the document chunks, whereas those reported in Figure 5 were obtained by computing the dot product.

In both tables of Figures 4 and 5, the first column contains the pretrained LLM used (indicated by the letter used in Section 2.2). The second column indicates the chunk type used, either sentence, window/n-gram, or paragraph. The following column reports the size of the input to the model in number of tokens. The subsequent columns report the mAP averaged over all users and all queries by considering different aggregation functions of the chunks' relevance scores. The names of these columns indicate the parameters passed to the aggregation function.

model	chunk type	window size	#ch: 1	#ch: 2 (sum)	#ch: 2 (avg)	#ch: 3 (sum)	#ch: 3 (avg)	#ch: 4 (sum)	#ch: 4 (avg)	#ch: 5 (sum)	#ch: 5 (avg)	#ch: 6 (sum)	#ch: 6 (avg)	#ch: 7 (sum)	#ch: 7 (avg)	#ch: 8 (sum)	#ch: 8 (avg)	#ch: 9 (sum)	#ch: 9 (avg)	#ch: 10 (sum)	#ch: 10 (avg)	#ch: all (sum)	#ch: all (avg)	virtual doc	max
(a)	sentence	384	0.60	0.61	0.61	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.36	0.48	0.54	0.64
	window		0.60	0.61	0.61	0.60	0.61	0.59	0.59	0.57	0.59	0.56	0.58	0.55	0.57	0.55	0.58	0.54	0.57	0.54	0.57	0.38	0.47	0.55	0.61
	paragraph		0.61	0.62	0.61	0.62	0.62	0.61	0.61	0.61	0.61	0.61	0.61	0.60	0.60	0.61	0.61	0.60	0.60	0.60	0.60	0.36	0.47	0.52	0.62
(b)	sentence	256	0.61	0.61	0.61	0.62	0.62	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.35	0.53	0.57	0.63
	window		0.61	0.64	0.64	0.62	0.63	0.61	0.62	0.60	0.61	0.59	0.60	0.59	0.60	0.58	0.61	0.58	0.61	0.58	0.62	0.37	0.51	0.60	0.64
	paragraph		0.62	0.64	0.64	0.63	0.63	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.35	0.52	0.55	0.64
(c)	sentence	512	0.58	0.59	0.59	0.60	0.60	0.59	0.59	0.59	0.59	0.58	0.58	0.57	0.57	0.56	0.56	0.56	0.56	0.55	0.55	0.28	0.46	0.53	0.60
	window		0.57	0.56	0.57	0.54	0.55	0.52	0.54	0.51	0.55	0.51	0.55	0.50	0.54	0.50	0.54	0.49	0.54	0.49	0.54	0.28	0.51	0.53	0.57
	paragraph		0.58	0.59	0.59	0.60	0.60	0.59	0.59	0.58	0.58	0.57	0.58	0.57	0.57	0.56	0.56	0.56	0.56	0.56	0.56	0.26	0.42	0.49	0.60
(d)	sentence	512	0.56	0.58	0.58	0.58	0.58	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.55	0.55	0.54	0.54	0.28	0.48	0.53	0.58
	window		0.52	0.55	0.56	0.56	0.55	0.54	0.55	0.52	0.54	0.53	0.55	0.52	0.55	0.52	0.54	0.50	0.54	0.50	0.54	0.28	0.48	0.54	0.56
	paragraph		0.57	0.58	0.58	0.57	0.57	0.57	0.57	0.57	0.57	0.56	0.56	0.55	0.55	0.55	0.55	0.54	0.54	0.54	0.54	0.26	0.49	0.49	0.58
(e)	sentence	512	0.60	0.61	0.61	0.62	0.62	0.64	0.64	0.63	0.63	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.28	0.47	0.57	0.64
	window		0.56	0.54	0.54	0.54	0.53	0.51	0.53	0.50	0.53	0.50	0.53	0.49	0.52	0.49	0.52	0.48	0.52	0.49	0.52	0.29	0.47	0.55	0.56
	paragraph		0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.61	0.61	0.61	0.61	0.60	0.60	0.60	0.60	0.59	0.59	0.26	0.45	0.55	0.61
			0.62	0.64	0.64	0.64	0.64	0.64	0.64	0.63	0.63	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.38	0.53	0.60	0.64

Figure 4. mAP for different combinations of pretrained LLMs, chunk definitions, and cosine similarity. The overall best mAP values across all combinations are highlighted in green.

model	chunk type	window size	#ch: 1	#ch: 2 (sum)	#ch: 2 (avg)	#ch: 3 (sum)	#ch: 3 (avg)	#ch: 4 (sum)	#ch: 4 (avg)	#ch: 5 (sum)	#ch: 5 (avg)	#ch: 6 (sum)	#ch: 6 (avg)	#ch: 7 (sum)	#ch: 7 (avg)	#ch: 8 (sum)	#ch: 8 (avg)	#ch: 9 (sum)	#ch: 9 (avg)	#ch: 10 (sum)	#ch: 10 (avg)	#ch: all (sum)	#ch: all (avg)	max
(a)	sentence	384	0.60	0.61	0.61	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.36	0.48	0.64
	window		0.60	0.61	0.61	0.60	0.61	0.59	0.59	0.57	0.59	0.56	0.58	0.55	0.57	0.55	0.58	0.54	0.57	0.54	0.57	0.38	0.47	0.61
	paragraph		0.61	0.62	0.61	0.62	0.62	0.61	0.61	0.61	0.61	0.61	0.61	0.60	0.60	0.61	0.61	0.60	0.60	0.60	0.60	0.36	0.47	0.62
(b)	sentence	256	0.61	0.61	0.61	0.62	0.62	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.35	0.53	0.63
	window		0.61	0.64	0.64	0.62	0.63	0.61	0.62	0.60	0.61	0.59	0.60	0.59	0.60	0.58	0.61	0.58	0.61	0.58	0.62	0.37	0.51	0.64
	paragraph		0.62	0.64	0.64	0.63	0.63	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.35	0.52	0.64
(c)	sentence	512	0.57	0.59	0.59	0.61	0.61	0.60	0.60	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.58	0.59	0.28	0.44	0.61
	window		0.60	0.59	0.60	0.57	0.58	0.55	0.57	0.53	0.57	0.53	0.57	0.51	0.55	0.51	0.54	0.50	0.54	0.49	0.53	0.28	0.49	0.60
	paragraph		0.60	0.62	0.62	0.62	0.62	0.61	0.61	0.60	0.60	0.60	0.60	0.60	0.60	0.59	0.59	0.58	0.58	0.58	0.58	0.26	0.42	0.62
(d)	sentence	512	0.63	0.64	0.64	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.28	0.47	0.64
	window		0.61	0.62	0.63	0.60	0.60	0.58	0.59	0.58	0.60	0.57	0.59	0.54	0.57	0.54	0.57	0.52	0.55	0.51	0.55	0.28	0.48	0.63
	paragraph		0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.61	0.61	0.60	0.60	0.26	0.46	0.64
(e)	sentence	512	0.61	0.63	0.63	0.64	0.64	0.65	0.65	0.65	0.65	0.65	0.65	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.28	0.49	0.65
	window		0.63	0.61	0.62	0.60	0.61	0.58	0.61	0.56	0.59	0.55	0.59	0.55	0.59	0.54	0.59	0.53	0.58	0.53	0.58	0.29	0.53	0.63
	paragraph		0.61	0.63	0.63	0.63	0.63	0.64	0.64	0.63	0.63	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.26	0.50	0.64
			0.64	0.64	0.64	0.64	0.64	0.65	0.65	0.65	0.65	0.65	0.65	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.64	0.38	0.53	**0.65**

Figure 5. mAP for different combinations of pretrained LLMs, chunk definitions, and dot product matching function. Higher mAP values within each row are highlighted in green. The overall best mAP values across all combinations are highlighted in blue.

The first parameter "*#ch: <number>*" controls the number of the best chunks considered for computing the document ranking score. When *<number> = All*, it means that all chunks are taken into account. The second parameter controls if the relevance score is defined as either an average of the chunks' scores (when the value "*avg*" is indicated), or a sum of the chunks' scores (when the value "*sum*" is indicated). More details are shown below:

- "*#ch: N (sum)*" indicates that the sum of the first N best chunk scores of each document was computed;
- "*#ch: N (avg)*" indicates that the average of the first N best chunk scores of each document was computed.

When *N = All*, this means that all the chunks in the documents are considered. Since documents generally consist of long texts with many chunks, an approach was also tested in which the document is represented by a single virtual embedding vector computed as the average of all the chunks' vectors. In this case, the results of mAP are reported in the column named "*Virtual Doc*" of the table in Figure 4. The last column named "*max*" reports the best mAP obtained by any of the tested definitions of the documents' chunks for the given setting in the row. Notice that the elapsed time for retrieval is negligible, as the aggregation of the chunks' relevance scores is instantaneous.

4. Discussion

By looking at the results in Figure 4, it can be easily noticed that three distinct models produce a maximum mAP = 0.64 for different settings by using cosine similarity between pairs of embedding vectors and by considering the aggregation of the best chunks' scores instead of using the virtual document vector. In this last case, the obtained mAP is sensibly lower. The most stable model with different input settings of the chunks' definitions, both window and paragraph, and when aggregating the first two best chunk's scores, either by computing their average or their sum, is **(b) all-MiniLM-L6-v2.**

This model also achieves the same mAP = 0.64 by considering the aggregation of the first best four chunks' scores when using the paragraph as the chunks' definition. Models **(a)** and **(e)** also achieve a maximum mAP = 0.64 when using sentence as the chunks' definition and when aggregating three and six best chunk scores using both sum and average, respectively. Nevertheless, these two models, when aggregating all the chunk scores, produce a lower value of mAP with respect to the model **(b) all-MiniLM-L6-v2.**

Figure 5 reports the mAP values using the dot product as the matching function. In this case, the best performing model is **(e) msmarco-distilbert-base-tas-b** that, when fed with chunks defined by sentences, reaches mAP = 0.65, taking into account from four to six best chunk scores using either their sum or average. The fact that this model produces the best results when using sentences as the definition of the chunks, whose number in the documents of the collection has the maximum variability (as can be seen in Figure 3) with respect to the other definitions (both paragraphs and windows with different sizes), makes this model the most stable.

Finally, a significance test was performed to exclude the null hypothesis. To this end, we considered as baseline the model *ch_per_doc_None_avg_0* that computes the RSV of documents by considering the dot product as the matching function, and by determining the final score as the maximum of the chunks' relevance scores.To evaluate the significance, we implemented the permutation test defined in [35].

Figure 6 reports 73 out of 345 tested cases in which the significance test is positive when the tested model yields results with respect to the baseline that are both improved and worsened. What can be noticed is that there are more cases of significantly worse results (59 out of 73 significant cases) with respect to the baseline, with a high decrease in mAP (down to −56%), with respect to significant cases of improved results (14 case out of 73), which also exhibit a small improvement (the maximum improvement reaches 6%). By analysing the results more closely, in the case of both the models **msmarco-distilbert-cos-v5** and **msmarco-roberta-base-ance-firstp**, the decrease in mAP in significant cases is proportional to the number of the chunk relevance scores considered for determining the document relevance; in such cases, the greater the number of considered chunks, the greater the decrease in mAP. This finding suggests that the users' relevance judgments were determined by considering the relevance of one or a few documents, up to four sentences, and not many chunks.

This finding can also be confirmed for both the models **all-MiniLM-L6-v2** and **msmarco-bert-base-dot-v5**, even if for these two models we do not have many significant cases when the number of considered chunks is high. A different behaviour can be observed for the last model **msmarco-distilbert-base-tas-b**; in this case, improved results are obtained by aggregating four or five sentences' relevance scores, while for the other types of chunks, the results are worse.

model	sentence	combination chunks/avg	map	map_baseline	stdev_ap	'2-sided_p-value'	improvement rate
msmarco-distilbert-cos-v5	sentence	ch_per_doc_3_avg_0	0.638235862	0.600093768	0.215023481	0.02492	6%
		ch_per_doc_3_avg_1	0.638189587	0.600093768	0.215019993	0.02497	6%
		ch_per_doc_4_avg_0	0.634607087	0.600093768	0.211517642	0.0496	6%
		ch_per_doc_4_avg_1	0.634607087	0.600093768	0.211517642	0.04932	6%
		ch_per_doc_None_avg_0	0.358772532	0.600093768	0.201809802	0	-40%
		ch_per_doc_None_avg_1	0.476321382	0.600093768	0.227160297	0.00275	-21%
	window	ch_per_doc_6_avg_0	0.562188361	0.603040109	0.247133172	0.03945	-7%
		ch_per_doc_7_avg_0	0.546297105	0.603040109	0.242898336	0.01228	-9%
		ch_per_doc_8_avg_0	0.545273232	0.603040109	0.246066575	0.01375	-10%
		ch_per_doc_9_avg_0	0.537798345	0.603040109	0.238630755	0.0057	-11%
		ch_per_doc_10_avg_0	0.535524667	0.603040109	0.240176224	0.00503	-11%
		ch_per_doc_None_avg_0	0.381825473	0.603040109	0.206792419	0	-37%
		ch_per_doc_None_avg_1	0.474609046	0.603040109	0.220858965	0	-21%
		rank_doc_embeddings	0.546523362	0.603040109	0.214225852	0.01228	-9%
	paragraph	ch_per_doc_None_avg_0	0.361209903	0.605824311	0.192404436	0	-40%
		ch_per_doc_None_avg_1	0.468254173	0.605824311	0.211933533	0.00002	-23%
		rank_doc_embeddings	0.524182009	0.605824311	0.202776523	0.00484	-13%
all-MiniLM-L6-v2	sentence	ch_per_doc_None_avg_0	0.346124524	0.605616912	0.190807977	0	-43%
		ch_per_doc_None_avg_1	0.530250795	0.605616912	0.233806499	0.01737	-12%
	window	ch_per_doc_None_avg_0	0.368567841	0.614097353	0.194725032	0	-40%
		ch_per_doc_None_avg_1	0.506413173	0.614097353	0.237648357	0.00008	-18%
	paragraph	ch_per_doc_2_avg_0	0.638288784	0.622562988	0.227354609	0.04536	3%
		ch_per_doc_None_avg_0	0.345945743	0.622562988	0.187564569	0	-44%
		ch_per_doc_None_avg_1	0.52329545	0.622562988	0.217194096	0.0003	-16%
		rank_doc_embeddings	0.553578553	0.622562988	0.223097	0.01284	-11%
msmarco-roberta-base-ance-firstp	sentence	ch_per_doc_None_avg_0	0.281526207	0.576636641	0.155485191	0	-51%
		ch_per_doc_None_avg_1	0.460320301	0.576636641	0.229267365	0.00327	-20%
	window	ch_per_doc_3_avg_0	0.543013496	0.568222426	0.212400459	0.03899	-4%
		ch_per_doc_4_avg_0	0.518230344	0.568222426	0.226106763	0.00411	-9%
		ch_per_doc_5_avg_0	0.508009484	0.568222426	0.231439854	0.00248	-11%
		ch_per_doc_6_avg_0	0.508900937	0.568222426	0.231615011	0.00556	-10%
		ch_per_doc_7_avg_0	0.501020091	0.568222426	0.233300732	0.00187	-12%
		ch_per_doc_8_avg_0	0.49755954	0.568222426	0.234018526	0.00116	-12%
		ch_per_doc_9_avg_0	0.489032655	0.568222426	0.234974692	0.00133	-14%
		ch_per_doc_10_avg_0	0.485604281	0.568222426	0.235887999	0.00114	-15%
		ch_per_doc_None_avg_0	0.28367026	0.568222426	0.151879281	0	-50%
		ch_per_doc_None_avg_1	0.513410791	0.568222426	0.215655962	0.04251	-10%
	paragraph	ch_per_doc_None_avg_0	0.255372247	0.575704624	0.143579289	0	-56%
		ch_per_doc_None_avg_1	0.419727198	0.575704624	0.195200221	0	-27%
		rank_doc_embeddings	0.489042685	0.575704624	0.207379167	0.00085	-15%
msmarco-bert-base-dot-v5	sentence	ch_per_doc_2_avg_0	0.580312555	0.554975623	0.223445153	0.01835	5%
		ch_per_doc_2_avg_1	0.580453343	0.554975623	0.224058514	0.01619	5%
		ch_per_doc_None_avg_0	0.281756554	0.554975623	0.155666241	0	-49%
		ch_per_doc_None_avg_1	0.483119026	0.554975623	0.204967508	0.03453	-13%
	window	ch_per_doc_2_avg_1	0.555060306	0.524196102	0.192683682	0.02415	6%
		ch_per_doc_None_avg_0	0.283827993	0.524196102	0.152086467	0	-46%
		ch_per_doc_None_avg_1	0.475244182	0.524196102	0.200466459	0.04989	-9%
	paragraph	ch_per_doc_None_avg_0	0.255806921	0.573744472	0.144020268	0	-55%
		ch_per_doc_None_avg_1	0.488240669	0.573744472	0.219934993	0.007	-15%
		rank_doc_embeddings	0.494578778	0.573744472	0.19360459	0.01085	-14%
msmarco-distilbert-base-tas-b	sentence	ch_per_doc_4_avg_0	0.637166787	0.603191182	0.215347909	0.01963	6%
		ch_per_doc_4_avg_1	0.637204504	0.603191182	0.215308845	0.02003	6%
		ch_per_doc_5_avg_0	0.634928718	0.603191182	0.21692179	0.03751	5%
		ch_per_doc_5_avg_1	0.634928718	0.603191182	0.21692179	0.03819	5%
		ch_per_doc_6_avg_0	0.638184408	0.603191182	0.221339929	0.04683	6%
		ch_per_doc_6_avg_1	0.638184408	0.603191182	0.221339929	0.04612	6%
		ch_per_doc_None_avg_0	0.282652987	0.603191182	0.15580007	0	-53%
		ch_per_doc_None_avg_1	0.465311214	0.603191182	0.218245156	0.00052	-23%
	window	ch_per_doc_4_avg_1	0.525744505	0.555644781	0.207108604	0.02978	-5%
		ch_per_doc_5_avg_0	0.498843971	0.555644781	0.238999867	0.02863	-10%
		ch_per_doc_6_avg_0	0.495768333	0.555644781	0.237324878	0.02009	-11%
		ch_per_doc_7_avg_0	0.491095686	0.555644781	0.238198012	0.01405	-12%
		ch_per_doc_7_avg_1	0.520442101	0.555644781	0.217011192	0.02045	-6%
		ch_per_doc_8_avg_0	0.486853852	0.555644781	0.23750149	0.00841	-12%
		ch_per_doc_8_avg_1	0.518523719	0.555644781	0.215852327	0.01265	-7%
		ch_per_doc_9_avg_0	0.482177669	0.555644781	0.237166538	0.00615	-13%
		ch_per_doc_9_avg_1	0.518724826	0.555644781	0.214813275	0.01462	-7%
		ch_per_doc_10_avg_0	0.492877595	0.555644781	0.24590586	0.03278	-11%
		ch_per_doc_10_avg_1	0.519332404	0.555644781	0.215428614	0.01758	-7%
		ch_per_doc_None_avg_0	0.285030617	0.555644781	0.151931048	0	-49%
		ch_per_doc_None_avg_1	0.474358828	0.555644781	0.204578292	0.00037	-15%
	paragraph	ch_per_doc_None_avg_0	0.256282202	0.596527317	0.144030691	0	-57%
		ch_per_doc_None_avg_1	0.448606899	0.596527317	0.221099584	0.00001	-25%

Figure 6. Mean AP (mAP) and corresponding standard deviation (st_dev_ap) for different combinations of pretrained LLMs, chunk definitions, and dot product matching function, along with a significance test against the baseline (*"ch_per_doc_1_avg_0"*). Rows highlighted in green indicate performance improvements over the baseline (positive variation in the last column); red highlights indicate a performance drop (negative variation).

5. Conclusions

The described experience presents several original contributions. First is the design of a KH to aid users in navigating a collection of documents by single topics and overlapping topics based on up-to-date LLM techniques and a fuzzy classification approach, allowing us to associate a document to more topics at the same time with distinct relevance. To this aim, the contribution focuses on the identification of the best performing LLM by carrying out a user-evaluation experiment. The second original contribution is the evaluation of different combinations of LLMs, pooling strategies, and similarity matching for the ad hoc retrieval task. As far as we know, a comparison of several "semantic" pretrained foundation LLMs, based on transformer architecture, to index and retrieve, in an ad hoc retrieval task, a highly heterogeneous collection of documents, with both highly variable length and variable nature and genre, was never performed before. The evaluation experiment investigated several aspects such as considering different chunk definitions and similarity metrics, and last but not least, different aggregation strategies of a varying number of the best chunks' relevance scores to compute the overall rank of documents. Determining the portion of the documents that is relevant to a query is important in the case the documents can be long, consisting of many chunks. This aspect can be dependent of personal user preferences, and indeed, the carried out experiments revealed that the engaged users considered documents as relevant when finding

just a few relevant sentences containing the search keywords. Nevertheless, depending on the tested foundation models, we could observe that the significance tests reveal that each model yields best results with a different number of considered chunks, meaning that relevance of documents does not solely depend on users' judgments but is also influenced by the model's characteristics. In this respect, a promising research direction could be to design a post hoc adaptive optimization mechanism to choose the best aggregation of the chunk relevant scores (i.e., find the best pooling strategy) for each user and collection. This application will be the objective of an evolution of the current proposal.

Author Contributions: Conceptualization, G.B. and P.T.A.d.; Data curation, P.T.A.d., A.M. and A.O.; Formal analysis, P.T.A.d.; Funding acquisition, L.B. and A.O.; Methodology, P.T.A.d. and G.B.; Project administration, L.B. and A.O.; Software (experiments), P.T.A.d.; Software (scraping), A.L., A.M., A.O. and P.T.A.d.; Writing—original draft, P.T.A.d. and G.B.; Writing—review and editing, P.T.A.d., G.B., L.B., A.L., A.M. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by ISPRA Istituto Superiore per la Protezione e la Ricerca Ambientale, grant number prot. CNR-IREA 0003070/2022.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data used for the user-evaluation study are freely available at the url cited in [17].

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

Abbreviations

The following abbreviations are used in this manuscript:

UNEP/MAP	Mediterranean Action Plan of the United Nations Environmental Programme
KH	Knowledge Hub
KMaP	Knowledge Management Platform
PDF	Portable Document Format
HTML	HyperText Markup Language
JPG	Joint Photographic Experts Group image compression format
IR	Information Retrieval
LLM	Large Language Model
RAC	Regional Activty Centre
REMPEC	Regional Marine Pollution Emergency Response Centre for the Mediterranean Sea
SPA/RAC	Regional Activity Centre for Specially Protected Areas
SCP/RAC	Regional Activity Centre for Sustainable Consumption and Production
PAP/RAC	Priority Actions Programme/Regional Activity Centre
CNR-IREA	National Research Council of Italy Institute for the Electromagnetic Sensing of the Environment
GNU GPL	GNU General Public License
BERT	Bidirectional Encoder Representations from Transformers
NLTK	Natural Language Toolkit
KNN	K-Nearest Neighbor
RSV	Relevance Score Value
RDF	Resource Description Framework
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language
mAP	Mean Average Precision

References

1. UNEP. Mediterranean Action Plan (MAP)—Barcelona Convention. Available online: https://www.unep.org/mediterranean-action-plan-map-barcelona-convention (accessed on 1 March 2025).
2. Bordogna, G.; Tagliolato, P.; Lotti, A.; Minelli, A.; Oggioni, A.; Babbini, L. Report 2—Semantic Information Retrieval–Knowledge Hub. *Zenodo* **2023**. [CrossRef]
3. Tagliolato Acquaviva d'Aragona, P.; Babbini, L.; Bordogna, G.; Lotti, A.; Minelli, A.; Oggioni, A. Design of a Knowledge Hub of Heterogeneous Multisource Documents to support Public Authorities. In Proceedings of the Ital-IA Intelligenza Artificiale—Thematic Workshops Co-Located with the 4th CINI National Lab AIIS Conference on Artificial Intelligence (Ital-IA 2024), Naples, Italy, 29–30 May 2024; Martino, S.D., Sansone, C., Masciari, E., Rossi, S., Gravina, M., Eds.; CEUR-WS.org, 2024, *CEUR Workshop Proceedings*, Volume 3762, pp. 458–463.
4. Kadhim, A.I. Survey on supervised machine learning techniques for automatic text classification. *Artif. Intell. Rev.* **2019**, *52*, 273–292. [CrossRef]
5. Oggioni, A.; Bordogna, G.; Lotti, A.; Minelli, A.; Tagliolato, P.; Babbini, L. Types of Users and Requirements—Report 0. *Zenodo*, **2023**. [CrossRef]
6. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval, Online Edition*; Cambridge University Press: Cambridge, UK, 2009.
7. Zhou, C.; Li, Q.; Li, C.; Yu, J.; Liu, Y.; Wang, G.; Zhang, K.; Ji, C.; Yan, Q.; He, L.; et al. A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT. *arXiv* **2023**, arXiv:2302.09419. [CrossRef]
8. Kraft, D.H.; Bordogna, G.; Pasi, G. Fuzzy Set Techniques in Information Retrieval. In *Fuzzy Sets in Approximate Reasoning and Information Systems*; Bezdek, J.C., Dubois, D., Prade, H., Eds.; Springer: Boston, MA, USA, 1999; pp. 469–510. [CrossRef]
9. Wang, J.; Huang, J.X.; Tu, X.; Wang, J.; Huang, A.J.; Laskar, M.T.R.; Bhuiyan, A. Utilizing BERT for Information Retrieval: Survey, Applications, Resources, and Challenges. *ACM Comput. Surv.* **2024**, *56*, 1–33. [CrossRef]
10. REMPEC. Regional Marine Pollution Emergency Response Centre for the Mediterranean Sea (REMPEC). Available online: https://www.rempec.org (accessed on 1 March 2025).
11. SPA/RAC. Regional Activity Centre for Specially Protected Areas. Available online: https://www.rac-spa.org (accessed on 1 March 2025).
12. SCP/RAC. Regional Activity Centre for Sustainable Consumption and Production. Available online: http://www.cprac.org (accessed on 1 March 2025).
13. PAP/RAC. Priority Actions Programme/Regional Activity Centre. Available online: https://paprac.org (accessed on 1 March 2025).
14. UNEP/MAP. UNEP/MAP: All Publications. Available online: https://www.unep.org/unepmap/resources/publications?/resources (accessed on 1 March 2025).
15. United Nations Environment Program. United Nations Environment Program: Documents Repository. Available online: https://wedocs.unep.org/discover?filtertype=author&filter_relational_operator=equals&filter=UNEP%20MAP (accessed on 1 March 2025).
16. Scraping Library. Available online: https://github.com/INFO-RAC/KMP-library-scraping (accessed on 1 March 2025).
17. Inforac Ground Truth Data Used for the User Evaluation Experiment. Available online: https://github.com/IREA-CNR-MI/inforac_ground_truth (accessed on 1 March 2025).
18. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv* **2020**, arXiv:1910.03771. [CrossRef]
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; NIPS'17; pp. 6000–6010.
20. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MI, USA, 6 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; pp. 4171–4186. [CrossRef]
21. Ding, M.; Zhou, C.; Yang, H.; Tang, J. CogLTX: Applying BERT to Long Texts. In *Proceedings of the Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: San Jose, CA, USA, 2020; Volume 33, pp. 12792–12804.
22. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Hong Kong, China, 3–7 November 2019. [CrossRef]
23. Bird, S.; Klein, E.; Loper, E. *Natural Language Processing with Python*, 1st ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2009.
24. Yager, R.R. On the fuzzy cardinality of a fuzzy set. *Int. J. Gen. Syst.* **2006**, *35*, 191–206.
25. Reimers, N.; Espejel, O.; Cuenca, P.; Aarsen, T. Msmarco-Distilbert-Cos-v5 Model Card. Available online: https://huggingface.co/sentence-transformers/msmarco-distilbert-cos-v5 (accessed on 1 March 2025).

26. Reimers, N.; Espejel, O.; Cuenca, P.; Aarsen, T. All-MiniLM-L6-v2 Model Card. Available online: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (accessed on 1 March 2025). [CrossRef]
27. Xiong, L.; Xiong, C.; Li, Y.; Tang, K.F.; Liu, J.; Bennett, P.; Ahmed, J.; Overwijk, A. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *arXiv* **2020**, arXiv:2007.00808.
28. Reimers, N.; Espejel, O.; Cuenca, P.; Aarsen, T. Msmarco-Roberta-Base-Ance-Firstp Model Card. Available online: https://huggingface.co/sentence-transformers/msmarco-roberta-base-ance-firstp (accessed on 1 March 2025).
29. Reimers, N.; Espejel, O.; Cuenca, P.; Aarsen, T. Msmarco-Bert-Base-Dot-v5 Model Card. Available online: https://huggingface.co/sentence-transformers/msmarco-bert-base-dot-v5 (accessed on 1 March 2025).
30. Reimers, N.; Espejel, O.; Cuenca, P.; Aarsen, T. Msmarco-Distilbert-Base-Tas-b Model Card. Available online: https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b (accessed on 1 March 2025).
31. UNESCO Thesaurus. Available online: http://vocabularies.unesco.org/thesaurus (accessed on 1 March 2025).
32. Get-It Inforac Thesaurus—Linked Open Data Access. Available online: http://fuseki1.get-it.it/dataset.html (accessed on 1 March 2025).
33. Get-It Inforac Thesaurus—Sparql Endpoint (Web Service). Available online: http://fuseki1.get-it.it/inforac/sparql (accessed on 1 March 2025). [CrossRef]
34. Beitzel, S.M.; Jensen, E.C.; Frieder, O. MAP. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: Boston, MA, USA, 2009; pp. 1691–1692. [CrossRef]
35. Smucker, M.D.; Allan, J.; Carterette, B. A comparison of statistical significance tests for information retrieval evaluation. In Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, New York, NY, USA, 6–10 November 2007; CIKM'07; pp. 623–632. https://doi.org/10.1145/1321440.1321528.