applied sciences

MDPI

Article

Leveraging Large Language Models for Sensor Data Retrieval

Alberto Berenguer, Adriana Morejón, David Tomás * and Jose-Norberto Mazón

Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain; aberenguer@dlsi.ua.es (A.B.); adriana.morejon@ua.es (A.M.); jnmazon@ua.es (J.-N.M.)
* Correspondence: dtomas@dlsi.ua.es

Abstract: The growing significance of sensor data in the development of information technology services finds obstacles due to disparate data presentations and non-adherence to FAIR principles. This paper introduces a novel approach for sensor data gathering and retrieval. The proposal leverages large language models to convert sensor data into FAIR-compliant formats and to provide word embedding representations of tabular data for subsequent exploration, enabling semantic comparison. The proposed system comprises two primary components. The first focuses on gathering data from sensors and converting it into a reusable structured format, while the second component aims to identify the most relevant sensor data to augment a given user-provided dataset. The evaluation of the proposed approach involved comparing the performance of various large language models in generating representative word embeddings for each table to retrieve related sensor data. The results show promising performance in terms of precision and MRR (0.90 and 0.94 for the best-performing model, respectively), indicating the system's ability to retrieve pertinent sensor data that fulfil user requirements.

Keywords: sensor data; large language models; word embeddings; data retrieval; FAIR principles

Citation: Berenguer, A.; Morejón, A.; Tomás, D.; Mazón, J.-N. Leveraging Large Language Models for Sensor Data Retrieval. *Appl. Sci.* **2024**, *14*, 2506. https://doi.org/10.3390/app14062506

Academic Editors: Katia Lida Kermanidis, Phivos Mylonas and Manolis Maragoudakis

Received: 16 February 2024
Revised: 9 March 2024
Accepted: 12 March 2024
Published: 15 March 2024

1. Introduction

Sensor data have become increasingly important for developing value-added information technology (IT) services and products [1]. As highlighted in Ref. [2], most sensors present data through web pages with diverse modes of presentation, layouts, and structures. Additionally, sensor data often fail to adhere to FAIR (Findable, Accessible, Interoperable, and Reusable) principles [3], suggesting that while sensor data may exist, accessing or processing them proves challenging [4]. Consequently, automatically identifying sensors that precisely cater to specific data needs remains a challenging task [5].

Existing methods for sensor data retrieval typically rely on inputting keywords or spatial locations, leading to the identification of relevant sensors. However, the matched sensor data should then be manually filtered and selected by the users. This selection is demanding as there are usually massive connected sensors [6].

To tackle this challenge, this paper introduces an innovative approach that utilises large language models (LLMs) [7] to convert raw sensor-generated data into a tabular format compliant with FAIR principles. These data are then processed by LLMs to derive word embedding representations, facilitating the indexing of the data for subsequent retrieval through semantic comparison. Given an input tabular dataset provided by the user, the proposed data retrieval system enables the identification of the most suitable sensor data to augment the provided dataset by incorporating new instances into it. As far as the authors are aware, there have been no prior approaches that utilise LLMs for sensor data retrieval with the intention of augmenting existing datasets by adding new relevant instances.

The performance of the retrieval system was assessed using a sensor dataset specifically collected for this purpose. The evaluation involved comparing five different state-of-the-art LLMs in the domain of word embedding representation. The findings indicate

that the system can achieve high precision and MRR (0.90 and 0.94 for the best model, respectively) in the task of row extension over sensor data.

The main objective of this paper is to leverage and evaluate LLMs for sensor data retrieval. The contributions of this paper towards achieving this objective are as follows:

- An architecture for automatically ensuring compliance of sensor data with FAIR principles. This architecture leverages LLMs for data transformation during the data gathering process.
- An approach for indexing FAIR-compliant sensor data to facilitate their retrieval. This approach relies on word embeddings to compute vectors that semantically represent sensor data.
- A framework for integrating both data gathering and retrieval approaches to enable efficient searches of sensor data.
- A case study in the domain of smart farming is presented to demonstrate the transformation of relevant sensor data into a FAIR-compliant format, thereby enhancing their searchability.
- A quantitative evaluation of different state-of-the-art LLMs in the domain of sensor data retrieval, accompanied by the development of a dataset specifically tailored for this evaluation.

The structure of the remaining sections of this document is as follows: Section 2 provides a review of existing literature related to LLMs, word embeddings for tabular data retrieval, and sensor data retrieval; Section 3 elaborates on the design and structure of our proposed system; Section 4 offers an illustrative case study, showcasing the system's application in a specific example; Section 5 explores the system's evaluation, focusing on the efficacy of various LLMs in creating word embeddings for tabular data representation and retrieval; and Section 6 concludes the paper, summarising key points and suggesting directions for future research.

2. Related Work

The system presented in this study harnesses cutting-edge technologies within the realm of natural language processing (NLP). Specifically, LLMs serve a dual purpose within the system. Firstly, they are utilised to extract structured data (e.g., in CSV format) from unstructured sensor data. Secondly, LLMs are employed to generate word embeddings, enabling the representation of tabular data and facilitating semantic comparison during the data retrieval process. This section provides an overview of pertinent literature concerning these technologies.

2.1. Large Language Models

LLMs have rapidly emerged as a cornerstone in the field of NLP, fundamentally transforming the landscape with their unprecedented capabilities. A goal of a language model predicts the order of words in a sentence. These models are "large" not only in their physical size, often comprising billions of parameters, but also in the vast amounts of text data that they are trained on. LLMs leverage deep learning techniques, particularly neural networks like Transformers [8], to analyse and produce human-like text.

These models have demonstrated remarkable proficiency across a wide spectrum of NLP tasks, including but not limited to text generation [9], machine translation [10], and answering questions [11], among others.

LLMs are pre-trained on diverse datasets from the internet, including books, articles, and websites, allowing them to learn a wide range of language patterns, contexts, and nuances. After the pre-training phase, these models can be fine-tuned for specific NLP tasks, rendering them versatile and adaptable tools for a wide range of applications.

BERT (Bidirectional Encoder Representations from Transformers) [12] represented a breakthrough in the field of language models. BERT introduced the concept of bidirectional learning, whereas its predecessors processed text in a unidirectional manner. This model

simultaneously considered both left and right contexts when training on a word, facilitating a comprehensive understanding of its entire context.

Inspired by the BERT architecture, numerous other models have emerged. For example, GPT-3 [13] (175B parameters) demonstrated its potential to generate text in a highly human-like style. Another notable model is Megatron-Turing [14] (53B parameters), which exhibited capabilities in reading comprehension and generation, thereby facilitating tasks such as summarisation and text generation. Equally noteworthy is BLOOM [15], an open-source LLM capable of handling 46 natural languages and 13 programming languages.

ChatGPT (https://chat.openai.com/, accessed on 15 January 2024) is a model built upon the GPT-3 architecture, which has exhibited outstanding performance in replicating the text generation capabilities of humans, expanding its capabilities to encompass multilingual conversations. This model has been fine-tuned for conversations, offering responses to a diverse array of questions and prompts. Similarly, Llama 2 [16] is another general-purpose open-source LLM developed by Meta AI. Like ChatGPT, it provides functionalities for text generation in various languages.

The capabilities of these models extend beyond human language, demonstrating proficiency in understanding HTML as well. These abilities encompass parsing the HTML of a webpage, semantic classification of HTML elements, generating descriptions of HTML inputs, and navigating web pages [17].

In the present work, LLMs are employed for two distinct purposes. The first is to transform raw HTML sensor information into structured data suitable for reuse. The second is to utilise LLMs for creating word embedding representations of this tabular data obtained from sensors, enhancing semantic search capabilities within tabular datasets, and improving the overall performance of sensor data retrieval systems.

2.2. Word Embeddings for Table Retrieval

Word embeddings are dense vectors that represent a word's meaning by positioning it as a point within a semantic space. They capture the distributional meaning of words, wherein similar representations are acquired for words found in analogous contexts [18]. Notable examples of such word representations include Word2vec [19], fastText [20], and Glove [21].

In recent years, word embeddings have gained extensive popularity for various semantic tasks within the field of NLP, including text classification [22], sentiment analysis [23], machine translation [24], and dialogue systems [25].

Typically, word embedding vectors have between 50 and 1000 dimensions, which is significantly less than standard sparse semantic vectors. The reduction in dimensions relies on generalisations that encapsulate the semantic relationships of words according to their context of occurrence.

Word embedding methodologies create a comprehensive lexicon from the unique terms within texts, giving every word a singular representation while overlooking their potential to convey various meanings in different scenarios. These are tagged as *static* models, which fail to encapsulate a word's multiple interpretations. Conversely, contemporary contextual word embeddings have the capacity to capture the multiple meanings of polysemous words, as each vector signifies not just a word but a specific sense. Consequently, each word can be depicted using distinct word embeddings, each tailored to a particular context in which the word may be encountered. Notable examples include ELMo [26], ULMFit [27], and BERT [12], among others [28].

Recent methodologies for table retrieval utilise word embeddings to represent tabular data, employing vector similarity metrics to assess the relevance among tables. Depending on the nature of the search query, table retrieval can be classified into keyword-based or table-based searches [29]. In keyword-based search, the query consists of a set of keywords, resembling the conventional approach used by search engines like Google. In contrast, in table-based searches, the query itself takes the form of a table, and the goal is to determine a similarity score between the input table and potential candidate tables. In the experiments

described in Section 5, tables function as queries to retrieve relevant tables, conducting a table-based search within the dataset.

The study conducted by Chen et al. [30] employed contextual word embeddings to enable table-based searches. They utilised a pretrained version of BERT, incorporating various types of data present in the table, including textual and numerical information, to provide context to BERT. This context was derived from the table's title, caption, column headings, and cell values.

In Ref. [31], the authors propose a method to preserve table structure and context in semantic representations. The proposed ConvTab method, utilising Convolutional Neural Networks, enhances table classification and semantic feature generation for query-table similarity, demonstrating significant improvements in standard table retrieval metrics over existing methods. In this proposal, tables are treated as 2D images. In the approach proposed in Ref. [32], image and graphs were combined. The authors used WordNet to represent cell content, constructing a graph to capture the similarities of the table content.

In the study by Shraga et al. [33], Word2vec was utilised by the authors to generate semantic vectors. Table information was segmented into four semantic spaces: description, schema, records, and facets. The proposal included using multilayer perceptrons (MLP) for schema, recurrent convolutional neural networks for description, and 3D convolutional neural networks to represent records and facets.

Nargesian et al. [34] proposed a method to retrieve tables compatible with an input table. To this end, the approach tries to assess whether the content of the table belongs to the same domain. Three statistical models were proposed to achieve this goal: intersecting content between two columns, semantic similarity between content by mapping columns to classes in an ontology, and using word embeddings to measure the similarity of textual content.

The work in Ref. [35] proposed a novel technique for pretraining embeddings on a large web table corpus. The authors applied their models to various data analysis tasks, including table retrieval, across different data sources. Evaluation demonstrated that models using pretrained Web table embeddings outperformed those pretrained on text.

The approach in Ref. [36] uses graph-based methods to represent tabular data. More specifically, data from relational databases was represented as a graph with three types of nodes: content, header, and file name.

In the domain of question answering from tables, notable models include TAPAS [37], which leverages BERT's encoder. This model flattens the table into smaller tokens and concatenates them with the query answer. Additionally, token embeddings are integrated with table-aware positional embeddings before being input into the model.

Lastly, also in the field of question-answering models, TaBERT [38] aims to acquire contextual representations for both utterances and the structured schema of tables. This is achieved by generating a content snapshot comprising relevant information related to the utterance, encoding both the utterance and content snapshot using BERT, and applying a vertical self-attention layer to aggregate information from various rows within the content snapshot, effectively capturing cross-row dependencies among cell values.

To the best of the authors' knowledge, there are no previous approaches that leverage LLMs for sensor data retrieval with the aim of augmenting existing datasets with new relevant instances.

2.3. Sensor Data Retrieval

Information retrieval systems for sensor data encompasses advancements in techniques and technologies aimed at enhancing the efficiency, accuracy, and scalability of data collection, processing, and analysis from diverse sensor networks. These systems are increasingly critical in various domains, including environmental monitoring, healthcare, smart cities, and industrial automation. With the exponential growth of Internet of Things (IoT) devices, efficient information retrieval from large-scale sensor networks has become paramount.

In the context of sensor data storage in information retrieval systems, various schemas and data models are utilised to efficiently manage and retrieve sensor data. Location-Free schemas do not rely on geographic information for data storage and retrieval. Instead, they may use distributed hash tables on combinatorial tile adjacency graphs and double-ruling schemes within each tile for efficient routing and load balancing [39].

A uniform representation model for sensor information recordings was proposed in Ref. [40] to handle heterogeneous, high-dimensional, and voluminous data in IoT-based sensor networks. This includes designing query models that represent different information query styles and developing indexing mechanisms to support historical and real-time search for spatiotemporal observation data.

The use of NoSQL databases like Hypertable (https://hypertable.org/, accessed on 15 January 2024) is considered suitable for storing massive amounts of sensor data due to their ability to handle heterogeneous, redundant, real-time streaming data [41]. These databases require manual query programming and can be optimised for sensor data storage and retrieval tasks.

Data-centric storage schemes, such as Geographic Hash Tables (GHT), distribute sensor data across the network based on the data content itself, allowing queries to be directed towards the location where relevant data are stored, thus reducing overall network traffic and improving efficiency [42].

Another way of storing sensor data is by applying data warehousing techniques, developing multidimensional schemata to capture contextual metadata alongside sensor datasets, supporting the analysis of datasets long after collection and providing a structured way to manage and query large volumes of sensor data [43].

The application of word embeddings and LLMs for sensor data retrieval is a largely unexplored field. One relevant work in this area is presented in Ref. [44]. This paper proposes a method to generate word embeddings for IoT devices and sensors in a smart home based on their activity, extending the theory of word embeddings to the area of IoT devices. The model, akin to Word2Vec and App2Vec [45], considers the time between device firings in its calculations. These embeddings can be used to find similar devices in an IoT device store or as a signature for each type of IoT device, among other use cases. The feasibility study on the CASAS dataset and a private real-world dataset of IoT device activity logs demonstrated the potential of this approach for identifying patterns in embeddings of various types of IoT devices in a household.

Although the method of representing sensor data information relies on word embeddings, the purpose of the previous article differs from the one proposed here in the task objective, as the present work aims at sensor data retrieval with the intention of augmenting datasets with similar data. Furthermore, the embedding models used in Ref. [44] are static, with the limitations mentioned earlier, whereas in the present work contextual embeddings are employed.

3. Architecture of the System

This section outlines the architecture of the proposed system, which consists of two primary components. The first one focuses on efficiently gathering structured data from sensor web pages. This phase leverages LLMs to convert raw sensor-generated data into a more reusable format, aligning it with FAIR principles.

Upon transforming the data into a suitable format (such as CSV, as shown in the experiments conducted in Section 5), the second component of the architecture enables indexing these data for subsequent exploration. This component furnishes functionalities to retrieve and integrate relevant sensor data from the previously collected datasets. To enhance retrieval performance, this component uses LLMS to create word embedding representations of tabular data for semantic comparisons between sensor datasets. The following subsections describe these two components in more detail.

3.1. Data Gathering

The aim of the first component is to gather sensor data and ensure that they are in a structured reusable format that is suitable for further exploitation by the second component. As mentioned previously, sensor data frequently originate from web pages [2], presenting challenges in their handling due to system and data heterogeneity. Consequently, effective management requires significant effort. The primary objective of this first component is to improve sensor data accessibility by harnessing the contextual comprehension and text generation capabilities inherent in modern LLMs to convert raw sensor data into a reusable format.

This procedure comprises two steps: data identification and, if needed, employing a LLM to parse raw sensor data and convert them into a structured format suitable for sharing and reuse. Figure 1 presents a visual summary of this process. The following lines elaborate on each of these steps.

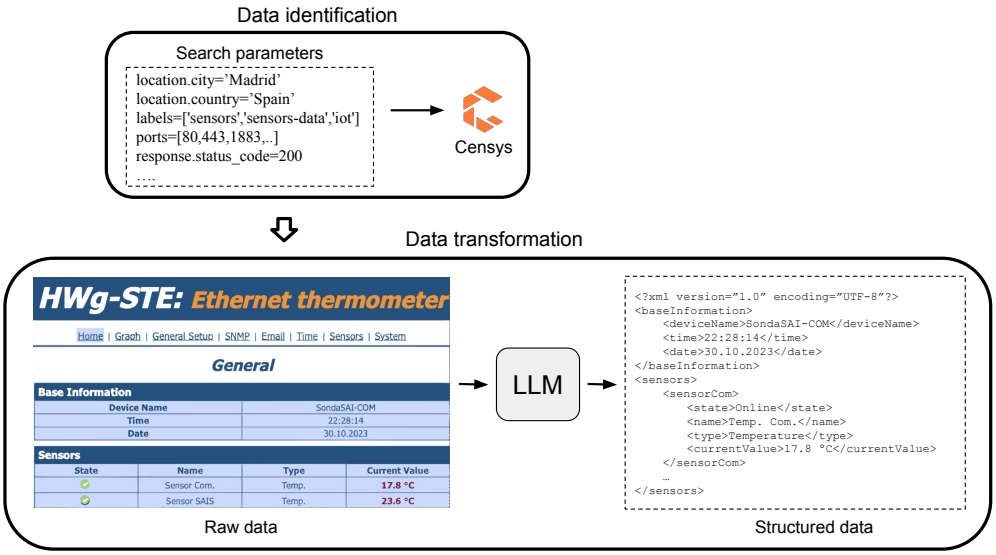

Figure 1. Two steps in the data gathering process: data identification and data transformation.

3.1.1. Data Identification

Before diving into the data identification process, it is crucial to ascertain the data prerequisites necessary to accomplish the desired objectives and select a suitable output format for subsequent utilisation. For example, the objective of the data gathering process could be to acquire weather sensor data from a particular region in CSV format.

After defining the objective, it is necessary to pinpoint web portals that facilitate the attainment of the goal established in the previous step. A range of options exist for retrieving data from sensors, including Shodan (https://www.shodan.io/, accessed on 15 January 2024), Censys (https://search.censys.io/, accessed on 15 January 2024), and ZoomEye (https://www.zoomeye.org/, accessed on 15 January 2024). These portals provide different tools, including APIs and filtering capabilities. It should be noted that a large portion of these data are in formats that make their reuse difficult, such as HTML.

One of the problems with these portals is the access limitations. APIs usually have access limits, and there are restrictions on the number of search filters that can be applied. Moreover, not all the geographical regions are covered equally. For insntace, ZoomEye is specially focused in China. For these reasons, it is necessary to properly verify that the chosen portal is suitable for the objectives established.

The process of data identification requires querying sensors in the selected portals. To this end, specific search parameters are required to extract pertinent information. Different

filters can be applied in the search for sensors with specific features. In the case of Censys, geographic area can be narrowed down by using filters like `location.city: Madrid, Spain`, and sensors can be identified by using `tag:IoT` or `tag:Sensor`. It should be noted that the definition of filters changes from one portal to another. Thus, it is required to know their specific syntax; although, all portals share a common syntax basis.

In the case of Censys, applying the previous filters reveal that most sensors have HTTP or MQTT (https://mqtt.org/, accessed on 15 January 2024) services enabled. Usually the open ports are 80 and 1883 (used by MQTT). Moreover, filtering for a response status code of 200 (successful response) using `response.status` is necessary to avoid defunct or malfunctioning sensors. These filters allow retrieving sensor data that are mainly shared in HTML format, combined with XML or JSON responses.

3.1.2. Data Transformation

When the sensor data returned in the previous step are unstructured, this second step allows these data to be transformed into reusable structured format. For this purpose, LLM is utilised to process raw sensor data, transforming them into structured information. This procedure involves creating a comprehensive prompt that outlines the specific data needed (e.g., "meteorological information") and specifying the desired output format. In this study, to seamlessly handle the subsequent retrieval process (see Section 3.2), data are transformed into CSV format.

Figure 2 shows an example of a prompt defined for an LLM to transform meteorological raw sensor data into CSV format. In this example, `{input}` represents the raw data from the sensor that will be transformed into CSV. This prompt facilitates the parsing of sensor data provided in formats such as HTML. The generated output produces data in a more reusable format, in accordance with the FAIR principles.

```
Instruction: Extract all meteorological information from the input and
return it in CSV format.
Exclude any additional content. Provide only the CSV output.
Input: {input}
Response:
```

Figure 2. Prompt to transform raw data into CSV format.

3.2. Data Retrieval

Once sensor data have been collected and properly formatted, the second component of the system manages user requests for that data. This component consists of a table-based search system that accepts user requests in the form of an initial dataset in tabular format. It then retrieves the most relevant indexed sensor datasets related to the user-provided dataset. This fulfils the requirement to retrieve sensor information that can complement the user's initial data.

At the core of this component lies the utilisation of word embeddings to conduct semantic comparisons between different sensor datasets. This component involves a three-step process: data preprocessing, word embeddings indexing, and the final stage of search and retrieval. The following lines describe each of them in more detail.

3.2.1. Data Preprocessing

Before indexing the data, a preprocessing and cleaning phase is often necessary. This includes tasks like removing punctuation from text, splitting CamelCase words, and converting all text to lowercase. These cleaning procedures ensure that the data are in a suitable format for subsequent processing.

Furthermore, in this step, the extraction of all columns from each table is carried out. LLMs typically have a token limit for processing, often around 512 tokens. Going beyond this limit risks losing essential information. To avoid such losses, in this step, the content of each column is divided into appropriately sized chunks of tokens. This not only prevents

the loss of information but also enables parallel processing, expediting the computation of word embeddings.

3.2.2. Word Embeddings Indexing

Once the tables have undergone cleaning and formatting, the following phase involves computing the word embeddings of each column and indexing them for further searches. The top part of Figure 3 shows the word embedding indexing step.

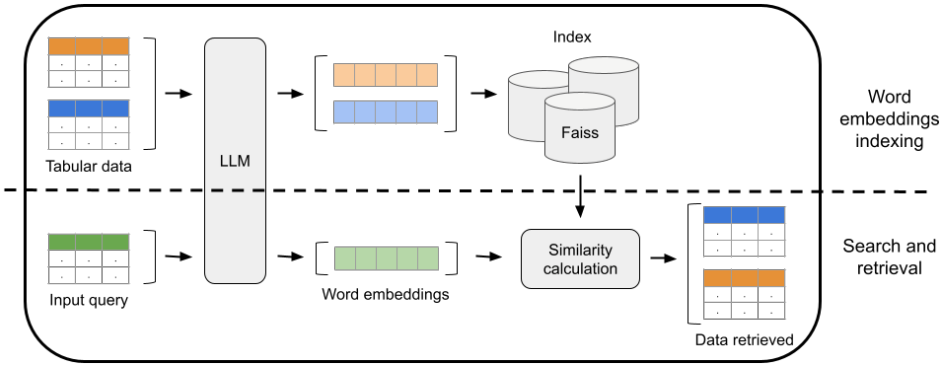

Figure 3. Word embeddings indexing (top part) and subsequent search and retrieval (bottom part).

During the indexing step, every preprocessed column passes through an LLM, generating a dense vector for each one. The vector's dimensionality varies, typically between 384 and 1024 dimensions depending on the model employed. In cases where the column's content exceeds the token limit of the model, as described in the previous step, the content is split, and a different word embedding is created for each chunk, averaging these vectors to create a single representative embedding for the column.

Following word embedding generation, the vectors are stored in an index using the Faiss library (https://faiss.ai/, accessed on 15 January 2024). Faiss provides an efficient means of storing and executing similarity searches based on dense vectors.

3.2.3. Search and Retrieval

The search and retrieval phase aims to identify sensor data that can be related with the user's input dataset. Specifically, this process entails seeking out other tables with highly relevant content that can be combined with the initial table to augment its content. The bottom part of Figure 3 shows the search and retrieval process.

In the same way as in the previous step for indexed tables, the input table is split into columns, and word embeddings are generated for each column. Following this, the system uses each column's embedding to query the index, retrieving the top-k similar columns that have been indexed. The similarity between columns is computed using cosine similarity, ranging between -1 and 1, where a value closer to 1 indicates greater similarity. The k-value, representing the total number of documents retrieved by a query, was set to 1000 in the experiments carried out in Section 5.

Next, to calculate the overall similarity for a table, the retrieved columns that belong to the same table are grouped, and their similarity scores are aggregated. The resulting sum is divided by the total number of columns in the input table, yielding a similarity score for each resultant table. Finally, the tables are ranked based on this score, providing an ordered list according to their relevance.

4. Case Study

This section presents a case study to better illustrate the process of data gathering and retrieval. This case study focuses on smart farming, which aims to utilise IoT technology along with machine learning algorithms to address current challenges in agricultural sustainability [46], such as predicting soil parameters or crop yields. To be conveniently

applied, machine learning algorithms require data obtained from sensors, climatic records, and satellite images [46]. However, the selection of the right data to be used in prediction systems remains a challenge in current smart farming approaches [46].

This case study focuses on demonstrating the usefulness and feasibility of the proposed retrieval approach for predicting evapotranspiration. Evapotranspiration is the loss of moisture from a surface by direct evaporation together with the loss of water by transpiration of vegetation, i.e., combined processes that move water from the Earth's surface into the atmosphere [47]. The prediction of evapotranspiration is crucial for estimating the amount of water required by agriculture projects, thus necessitating a variety of data sources, including data collected from weather stations [47].

The objective set in this use case is to develop a machine learning approach for predicting evapotranspiration in a specific region. Atmospheric variables are often available through public institutions via open data portals, but these institutions do not cover all territories, and there are places where weather stations are far from the required location [48]. Fortunately, many citizens, non-governmental organisations, and academic institutions have their own weather stations and publish data on the web in portals such as the aforementioned ZoomEye or Censys. Following the first step of the data gathering process, the required data are retrieved from these portals regardless of the format. In this case study, data from the website shown in Figure 4 are retrieved (https://www.mesonet.org/, accessed on 15 January 2024).

USA Mesonet Realtime Display

02/03 14:28:36 pm CST

Non-mobile device users please hover the mouse over the maximum or minimum value to obtain a date/time stamp of when the value occurred

Display Windchill	@ 6.56 ft				@ 32.81 ft		SeaLv	Today's Extremes (Hi/Lo)					Yesterday's Extremes (Hi/Lo)				
Location/Obs time	Temp	HtIdx	DewPt	Hum	WdDir	WdSpd	Press	Rain	WdSpd	Temp	DewPt	Hum	Rain	WdSpd	Temp	DewPt	Hum
Switch to Metric units	°F	°F	°F	%	°	mph	inHg	in	mph	°F	°F	%	in	mph	°F	°F	%
Agricola *2024-02-03 14:26:00*	**61.74**	*N/A*	47.88	60.34	90.19	1.97	29.73	0.00	7.09	58.59 39.31	42.32 36.30	93.90 50.81	0.00	7.09	58.59 39.31	42.32 36.30	93.90 50.81
Andalusia *2024-02-03 12:10:10*	**67.89**	*N/A*	41.53	38.25	140.36	13.08	29.87	0.00	6.80	55.74 35.18	40.54 32.58	92.90 49.09	0.00	6.80	55.74 35.18	40.54 32.58	92.90 49.09
Ashford North *2024-02-03 14:25:18*	**69.58**	*N/A*	45.07	41.34	85.14	4.68	15.10	0.00	15.64	70.16 41.11	47.50 38.88	93.60 37.86	0.00	11.25	68.74 45.54	43.90 35.58	83.80 31.96
Atmore *2024-02-03 14:25:08*	**64.79**	*N/A*	42.13	43.57	146.47	8.48	29.79	0.00	7.89	57.16 35.56	41.87 33.17	95.50 44.86	0.00	7.89	57.16 35.56	41.87 33.17	95.50 44.86
Bay Minette FS *2024-02-03 14:24:56*	**64.03**	*N/A*	43.72	47.56	187.72	1.39	29.78	0.00	15.35	65.44 41.76	48.82 40.38	96.10 44.87	0.00	9.28	69.58 40.97	50.42 39.71	96.80 35.25
Bayou La Batre *2024-02-03 14:25:12*	**61.49**	*N/A*	56.02	82.23	109.32	8.40	29.77	0.00	18.63	62.60 40.20	64.45 40.78	100.00 69.04	0.00	11.11	64.45 40.78	53.37 40.78	100.00 52.04

Figure 4. Mesonet portal containing data from weather stations.

Since these data are in HTML format and thus not properly structured for further analysis, the second step of the approach involves leveraging LLMs to transform the raw weather data shown in Figure 4 into the CSV format depicted in Figure 5.

Location/Obs time	Temp(@ 6.56 ft)	HtIdx(@ 6.56 ft)	DewPt(@ 6.56 ft)	Hum(@ 6.56 ft)	WdDir(@ 32.81 ft)	WdSpd(@ 32.81 ft)	SeaLv(Press)	Rain(Today's Extremes (Hi/Lo))	WdSpd(Today's Extremes (Hi/Lo))	Temp(Today's Extremes (Hi/Lo))	DewPt(Today's Extremes (Hi/Lo))	Hum(Today's Extremes (Hi/Lo))
Agricola/2024-02-03 14:26:00	61.74	N/A	47.88	60.34	90.19	1.97	29.73	0.00	7.09	58.59/39.31	42.32/36.30	93.90/50.81
Andalusia/2024-02-03 12:10:10	67.89	N/A	41.53	38.25	140.36	13.08	29.87	0.00	6.80	55.74/35.18	40.54/32.58	92.90/49.09
Ashford North/2024-02-03 14:25:18	69.58	N/A	45.07	41.34	85.14	4.68	15.10	0.00	15.64	70.16/41.11	47.50/38.88	93.60/37.86
Atmore/2024-02-03 14:25:08	64.79	N/A	42.13	43.57	146.47	8.48	29.79	0.00	7.89	57.16/35.56	41.87/33.17	95.50/44.86
Bay Minette FS/2024-02-03 14:24:56	64.03	N/A	43.72	47.56	187.72	1.39	29.78	0.00	15.35	65.44/41.76	48.82/40.38	96.10/44.87
Bayou La Batre/2024-02-03 14:25:12	61.49	N/A	56.02	82.23	109.32	8.40	29.77	0.00	18.63	62.60/40.20	56.48/40.20	100.00/69.04
Castleberry/2024-02-03 14:25:16	68.21	N/A	41.46	37.72	132.27	3.58	29.81	0.34	7.23	58.06/30.89	41.36/30.89	100.00/47.90
Elberta/2024-02-03 14:26:18	63.02	N/A	52.60	68.75	123.96	4.68	29.77	0.00	10.08	59.65/36.17	42.92/35.86	100.00/46.13
Fairhope/2024-02-03 14:25:14	62.30	N/A	48.81	61.29	91.81	18.93	29.75	0.00	8.40	58.80/37.59	42.04/33.57	87.90/47.14
Florala/2024-02-03 14:24:58	69.15	N/A	45.06	41.92	133.32	2.41	29.85	0.00	9.28	56.01/33.00	41.74/31.85	97.50/52.44
Foley/2024-02-03 14:25:36	62.75	N/A	54.23	73.68	96.90	9.72	29.76	0.00	8.91	60.10/36.37	43.02/34.95	97.00/44.81
Gasque/2024-02-03 14:25:08	61.67	N/A	56.06	81.81	66.59	10.89	29.75	0.00	10.67	60.40/43.40	44.82/37.97	88.20/47.60
Geneva/2024-02-03 14:25:10	69.58	N/A	42.84	37.69	128.86	6.65	29.82	0.00	8.04	56.21/32.96	41.43/30.89	95.10/44.87
Grand Bay/2024-02-03 14:25:10	61.53	N/A	54.44	77.53	89.31	10.52	29.75	0.00	8.62	59.50/35.89	45.05/34.26	94.90/49.17
Jay/2024-02-03 14:25:09	66.24	N/A	41.10	39.81	142.33	9.94	29.80	0.00	7.97	56.75/37.12	41.46/34.05	90.70/51.18
Kinston/2024-02-03 14:25:10	69.12	N/A	42.27	37.73	134.04	2.41	29.81	0.00	14.98	69.57/37.49	46.50/36.27	97.50/33.99
Leakesville/2024-02-03 14:27:13	62.21	N/A	45.98	55.24	141.15	7.23	29.75	0.00	7.67	56.68/35.46	42.33/34.91	99.00/54.38
Loxley/2024-02-03 14:27:07	62.51	N/A	47.22	57.31	133.98	11.18	29.78	0.00	9.72	57.97/34.07	42.44/33.31	98.50/50.25
Mobile (Dog River)/2024-02-03 14:25:26	61.47	N/A	51.34	69.33	122.49	4.60	29.78	0.00	6.36	57.67/38.74	46.10/36.92	94.60/49.72
Mobile (USA Campus West)/2024-02-03 14:25:10	62.67	N/A	44.98	52.36	126.24	7.09	29.77	0.00	7.67	57.49/37.54	40.56/35.30	94.40/44.35
Mount Vernon/2024-02-03 14:25:24	64.79	N/A	43.88	46.75	49.20	2.41	29.79	0.00	8.43	59.39/32.14	47.03/32.14	100.00/46.24
Pascagoula/2024-02-03 14:26:12	60.67	N/A	55.29	82.43	132.93	1.32	29.72	0.00	8.62	57.67/39.76	43.81/36.91	91.00/56.74
Poarch Creek/2024-02-03 14:25:08	64.76	N/A	45.75	50.09	108.25	9.06	29.77	0.00	19.73	65.89/39.19	47.41/37.62	95.40/42.24
Robertsdale/2024-02-03 14:25:05	63.11	N/A	46.56	54.73	121.29	11.11	29.76	0.00	9.43	58.84/37.56	42.00/35.62	95.80/50.25
Saraland/2024-02-03 14:25:38	64.27	N/A	41.10	42.65	98.21	5.26	29.76	0.00	7.67	57.97/34.72	41.87/33.01	94.80/46.46

Figure 5. CSV output after data transformation.

After the data gathering process ends, the data retrieval starts. The following step in this use case involves the machine learning expert querying the retrieval system with a table containing sensor data that the expert wants to expand with additional relevant data. This input dataset could include different instances of sensor data including attributes such as location, temperature, dew point, and humidity. Figure 6 illustrates an example of how such an input table may be used to retrieve relevant data.

Location	Temperature	Dew Point	Humidity
Dog River	61.74	47.88	60.34

Figure 6. Example input tabular data for retrieving related sensor content.

In the final step of the data retrieval process, the tabular search engine would retrieve a ranking of tabular datasets that the system considered relevant according to the input data. These datasets can then be used to further train machine learning algorithms in the task of evapotranspiration prediction.

5. Evaluation

As described in Section 3, the system comprises two primary components: data gathering and data retrieval. Concerning the data gathering component, prior work by the authors [49] evaluated various LLMs, including ChatGPT and Llama 2, to assess their performance in transforming raw sensor data into a structured format. The highest precision and recall achieved were 93.51% and 85.33%, respectively, demonstrating their suitability for the task.

Therefore, this study focuses solely on evaluating the second component, data retrieval, utilising different LLMs to generate representative word embeddings for each table. Specifically, given an input table containing sensor data, this evaluation measures the precision and Mean Reciprocal Rank (MRR) [50] of the retrieval system in delivering relevant indexed sensor datasets that could be combined with the query data content.

The primary objective of this experimentation is to assess the effectiveness and efficiency of the proposed LLMs in a specific, well-defined context rather than to exhaustively explore all potential variables and scenarios. The experimental setup has been designed to provide a clear and focused investigation into the models' capabilities, particularly in relation to execution time and retrieval accuracy. These metrics were chosen because they are critical to the practical application of these models in real-world settings, where performance and speed are often key considerations for deployment.

Applying this approach to a specific use case will allow assessing the feasibility of applying it more broadly to other contexts. Future work may extend this foundation, exploring additional dimensions and scenarios to build upon the insights presented herein.

The remainder of this section presents the models utilised, the datasets collected, and the outcomes of each proposed experiment. Furthermore, a final section is included, discussing the limitations of this approach and the privacy and security issues that may arise from the use of LLMs in the context of sensor data and IoT.

5.1. Large Language Models

The architecture proposed in this work implies using an LLM to transform tabular data into word embedding vectors. The evaluation aims to assess the models' effectiveness in generating precise word embeddings to represent sensor data content.

Previous work [51] has shown the superiority of contextual word embeddings, such as BERT and RoBERTa [52], over static word embeddings like Word2vec and fastText, as well as traditional information retrieval techniques such as BM25 [53]. For this reason, this evaluation focuses on five different language models featuring diverse architectures that produce contextual word embeddings:

1. Universal AnglE Embedding [54]: This model is based on BERT and introduces angle optimisation in a complex space. This approach effectively overcomes the limitations of

the cosine function's saturation zone, which can interfere with gradient and optimisation processes. The model tested, *UAE-Large-V1* (https://huggingface.co/WhereIsAI/UAE-Large-V1, accessed on 10 December 2023), produces 1024-dimensional vectors and is currently the state of the art for Semantic Textual Similarity, as per the Massive Text Embedding Benchmark (MTEB) Leaderboard from Huggingface (https://huggingface.co/spaces/mteb/leaderboard, accessed on 15 January 2024.)

2. BGE Embedding [55]: It is an open-source general embedding model that was trained on large-scale pair data using contrastive learning. The model provides embeddings ranging from 384 to 768 dimensions. Three different versions of this model that vary in the number of parameters were tested: *bge-large-en-v1.5* (https://huggingface.co/BAAI/bge-large-en-v1.5, accessed on 10 December 2023), *bge-base-en-v1.5* (https://huggingface.co/BAAI/bge-base-en-v1.5, accessed on 10 December 2023) *and bge-small-en-v1.5* (https://huggingface.co/BAAI/bge-small-en-v1.5, accessed on 10 December 2023).
3. Ember: The model has undergone training on a vast collection of text pairs covering diverse domains, such as finance, science, medicine, law, and various others. The resulting word embedding vectors consist of 1024 dimensions. The specific model employed was *ember-v1* (https://huggingface.co/llmrails/ember-v1, accessed on 10 December 2023).
4. General Text Embeddings [56]: GTE models primarily rely on the BERT framework and currently come in three sizes: large, base, and small. They are trained on a comprehensive corpus of relevant text pairs, spanning various domains and scenarios. The model undergoes training using multi-stage contrastive learning techniques. The models tested were *gte-large* (https://huggingface.co/thenlper/gte-large, accessed on 10 December 2023) and *gte-base* (https://huggingface.co/thenlper/gte-base, accessed on 10 December 2023).
5. SentenceBERT [57]: This model generates contextual embedding vectors by employing siamese and triplet network architectures to derive semantically meaningful sentence embeddings. Specifically, the model *all-MiniLM-L6-v2* (https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, accessed on 10 December 2023), trained on one billion sentence pairs, was used in this experiments. This model produces word embedding vectors of 768 dimensions.

The selection of these models was based on their high rankings in the MTEB Leaderboard. Notably, a conventional model like BERT currently holds the 100th position on this list.

In the current body of literature, there is not a formal agreement regarding the minimum parameter scale required for LLMs, as the model's capacity is intertwined with both data volume and total computational resources [58]. In this work, the proposed models are considered LLMs as they are at least one order of magnitude larger, regarding the number of parameters, than small-scale language models like BERT.

5.2. Dataset

The evaluation proposed aims to measure the precision and MRR of the data retrieval component. However, no existing dataset was available for testing retrieval systems involving sensor data. Hence, it was necessary to curate a dataset from scratch.

Initially, raw sensor data were retrieved by following the architecture outlined in Section 3.1 focusing on meteorological data. The selected portal was the aforementioned Censys, which is a scanning service specifically engineered to aggregate information about Internet devices and resources. It is open source and freely available for academic purposes. The search carried out on meteorological data provided 35 datasets in HTML format.

Subsequently, an LLM (GPT-3.5 Turbo from OpenAI) was employed to transform these datasets into CSV format, as described in Ref. [49]. The average number of tokens in each input file was 98,885.87, so most of the files to be transformed by the LLM had to be split into several files that met the maximum input tokens of the model (16,385 tokens in

the case of GPT-3.5 Turbo). Once transformed, each individual file was merged back into a single file resulting in the complete transformation of the original file to CSV. The average transformation time per complete file was 2308.89 s. Note that this time depends on the infrastructure provided by OpenAI and may be constrained by API demand and file size.

To complement this collection with additional CSV files for the evaluation, various open data portals were scanned for sensor data: Government of Spain (https://datos.gob.es/, accessed on 10 December 2023) (1194 tables), Government of the United States (https://data.gov/, accessed on 10 December 2023) (54 tables), and Valencia City Hall (https://valencia.opendatasoft.com/, accessed on 10 December 2023) (16 tables). The final dataset comprises 1299 tables in CSV format encompassing data from sensors across different domains, including environment, transportation, energy, science, and technology. Table 1 provides an overview of the primary characteristics of the compiled dataset.

Table 1. Characteristics of the sensors dataset gathered.

Characteristic	Value
Number of tables	1299
Total number of rows	163,981,050
Total number of columns	10,487
Total number of numerical columns	4746
Avg. number of rows	126,236.37
Avg. number of columns	8.07
Avg. number of numerical columns	3.65
Max. number of rows	324,000
Max. number of columns	26
Max. number of numerical columns	20

5.3. Results and Discussion

This section presents the evaluation conducted to measure the performance of the previously mentioned LLMs in the data retrieval task. The first part of this section delves into the accuracy assessment of these models in retrieving pertinent sensor data to augment an initial dataset. The second part provides details regarding the execution time of the proposed models during the indexing and retrieval processes.

5.3.1. Retrieval Performance

This section presents the evaluation of the performance of the different LLMs. In order to automatically evaluate the retrieval performance of the models, for each CSV file in the dataset gathered, 10% of the rows were randomly extracted and used to configure a new table to be used as a query in the evaluation. The remaining 90% of the content of each table was indexed and used as the target of the data retrieval system.

Thus, when a query table with 10% of the information was issued to the system, the goal was to retrieve the table with the 90% of information that complements it. If this is the case, there is a match, and it is considered that the system satisfactorily retrieved a dataset that can augment the query data.

The system was evaluated in terms of precision at one (P@1) and MMR, in the task of retrieving relevant sensor data to augment a query table. The P@1 measure indicates the model's accuracy in retrieving the most relevant dataset as the top result, while MRR provides an average of the reciprocal ranks of the models' results, giving a sense of the overall quality of the retrieved datasets across the top-ranked positions. Other popular metrics in information retrieval, such as nDCG or recall, cannot be applied in this context since they require more than one relevant document to be implemented.

Table 2 provides the performance achieved by each model. The results show that the current cutting-edge model *UAE-Large-v1* is not yielding the expected outcomes, showing the poorest scores in P@1 and MMR. This behaviour might be attributed to its backbone model, the pre-trained uncased BERT base model (with 110M parameters), fine-tuned on a

wide range of data primarily focusing on other specific topics. This indicates that the model may struggle to provide representative embeddings for the experimental data without prior fine-tuning.

Table 2. P@1 and MMR for each model in the data retrieval task. Best results are boldfaced.

Model	P@1	MMR
UAE-Large-V1	0.33	0.51
bge-large-en-v1.5	0.37	0.51
bge-base-en-v1.5	**0.90**	**0.94**
bge-small-en-v1.5	0.75	0.85
ember-v1	0.48	0.62
gte-large	0.73	0.83
gte-base	0.67	0.77
all-MiniLM-L6-v2	0.52	0.69

The *bge-base-en-v1.5* model exhibited exceptional accuracy, achieving 0.90 P@1 and 0.94 MRR, indicating that it is highly accurate at not only placing the most relevant dataset at the top of its retrieval list but also maintaining high relevance across its top-ranked results. This suggests that *bge-base-en-v1.5* has a robust understanding of tabular data representation. Interestingly, the results of its large version did not surpass its smaller model. The *bge-small-en-v1.5* model, utilising only 384 dimensions, exhibited remarkable performance, achieving the second-best results and emerging as a promising option for real-time systems due to its speed and lightweight nature.

Ember's results, despite holding a top position on the MTEB leaderboard, did not reflect similarly in these experiments. This discrepancy might be related to their training data not aligning with the data used in this experiment, resulting in non-representative word embeddings.

On the other hand, the GTE models achieved moderately good results, with the large model showing slightly higher precision compared to the base models. Conversely, the *all-MiniLM-L6-v2*, one of the smallest and oldest models, delivered scores slightly higher than *ember-v1*, maintaining competitiveness but being outperformed by more recent models like the BGE variants.

The results suggest that base versions of models (*bge-base-en-v1.5* and *gte-base*) provide a good balance between size and performance. This could imply that for the task of representing tabular data and retrieving similar datasets, having an excessively large model does not necessarily translate to better performance.

5.3.2. Execution Time

These experiments were performed using an A100-SXM4 GPU with 40 GB of RAM. Table 3 presents the execution times for each model in the indexing and retrieval phases. Notably, the processing times for indexing a table was consistently low, typically ranging between 60 and 160 milliseconds. This efficiency enabled the indexing of the entire dataset comprising 1299 files using all proposed models in less than 20 min.

Table 3. Indexing (seconds) and retrieval (hours and minutes) execution time by model.

Model	Indexing (s)	Retrieval (hh:mm)
UAE-Large-V1	184	01:27
bge-large-en-v1.5	217	01:28
bge-base-en-v1.5	128	01:27
bge-small-en-v1.5	127	01:28
ember-v1	216	01:28
gte-large	218	01:28
gte-base	127	01:26
all-MiniLM-L6-v2	86	01:26

Variations in the time taken for word embedding calculations were observed among different models, influencing their overall speed. Notably, larger models generating embeddings of size 1024 and smaller models of size 768 exhibited similar duration for the indexing process. Conversely, the smallest models, specifically *all-MiniLM-L6-v2* with a size of 384, were notably faster, indexing all datasets in just 86 s.

In contrast, the retrieval operation demanded substantially more time than the indexing phase. On average, the entire retrieval process across all models required approximately 12 h, averaging around 1 h and 27 min per model. Unlike the indexing phase primarily relying on embedding calculations, the retrieval process involves a series of steps outlined in Section 3. This includes retrieving the top-k comparable columns for each column in the query table, computing similarity scores, and re-ranking the results. The duration increases with the number of indexed datasets. It is notable that the choice of model does not significantly impact the average time spent on the retrieval process, ranging from 3.2 to 5.9 s per query. Nevertheless, this time range can be considered reasonable for a production environment.

5.4. Limitations

The approach proposed in this work, while promising, presents several challenges and limitations that require consideration. The data identification process requires expert knowledge within the specific field of study to comprehensively identify and evaluate relevant data portals and sources. Additionally, access to these data sources may be restricted due to a variety of factors, including limited query allowances, proprietary data restrictions, licensing requirements, lack of API access, and permission constraints. Such limitations necessitate extensive familiarisation with API documentation and the customisation of data collection scripts to ensure compatibility with selected data sources.

Furthermore, the application of LLMs introduces its own set of challenges. Despite their advanced comprehension abilities, issues such as slow response generation, handling data in varied formats, and mitigating the occurrence of hallucinations (the generation of inaccurate or nonsensical responses) are areas that require further research and development to fully leverage the capabilities of LLMs in enhancing data portal usability and efficiency.

Furthermore, data security and privacy issues must be taken into account when using LLMs in the context of sensor data, particularly within IoT ecosystems. As these models process vast amounts of data, including potentially sensitive information, they can inadvertently introduce risks and limitations that need to be addressed. Sensor data can include personal and sensitive information, especially in domains like healthcare and smart homes. LLMs trained on such data might learn to replicate or infer personal information, leading to privacy breaches.

Ensuring compliance with data protection regulations (such as GDPR (http://data.europa.eu/eli/reg/2016/679/oj, accessed 15 January 2024) and HIPAA (https://www.hhs.gov/hipaa/index.html, accessed 15 January 2024) can be challenging when using LLMs. The models must be designed and operated in ways that respect data privacy, consent, and the right to be forgotten, which can be difficult when models integrate and learn from vast datasets.

Large language models can inadvertently encode and reveal sensitive information from their training data, raising significant privacy concerns. For example, models like GPT-2 have been shown to leak personally identifiable information through membership inference attacks [59].

The leakage of training data, intentional or accidental, can pose a significant risk, particularly when models are trained on sensitive or proprietary datasets. This includes the potential for models to memorise and replicate exact snippets of sensitive training data [60]. Furthermore, inference attacks, such as model inversion or membership inference attacks, can be used to extract detailed information about the training data or about individuals' data used in the training set, leading to privacy violations [61].

Malicious actors could potentially insert backdoors or other vulnerabilities into LLMs, which could be exploited later to compromise data privacy or security. These backdoors could be activated by specific inputs to cause the model to output predetermined or harmful responses [62].

Moreover LLMs are often seen as "black boxes" due to their complex and opaque nature, making it difficult to understand how they process data and arrive at specific outputs [63]. This lack of transparency can hinder the identification and rectification of privacy and security issues within the model.

6. Conclusions and Future Work

This paper proposed an innovative method for searching relevant sensor data, encompassing two key components: data gathering and data retrieval. The former employs LLMs to convert sensor data into a reusable format, when necessary, showcasing the potential of language models in promoting FAIR compliance for sensor data. The latter utilises LLMs to create word embedding representations of tabular data, providing semantic search capabilities to extract the most relevant indexed sensor datasets for augmenting user-provided data.

Several contributions were made in this paper to achieve these goals. First, an architecture was proposed to automate the alignment of sensor data with the FAIR principles. This architecture leverages LLMs to facilitate the transformation of data during the data gathering phase. Additionally, an approach is developed for indexing sensor data that comply with FAIR principles, thereby facilitating their retrieval. This approach utilises word embeddings to generate vectors that semantically represent sensor data, aiding in efficient retrieval. Furthermore, a framework is introduced to integrate both data gathering and retrieval approaches, enhancing the searchability of sensor data. In order to demonstrate the practical application of these contributions, a case study was presented within the domain of smart farming. This case study illustrates the transformation of relevant sensor data into a FAIR-compliant format, consequently improving their accessibility and searchability.

The evaluation of the proposed approach involved comparing the performance of different LLMs in generating representative word embeddings for each table in order to retrieve relevant sensor data for a given tabular query. An evaluation dataset containing 1299 CSV files with sensor data was specifically gathered for this task.

The evaluation measured the P@1 and MRR of the retrieval system. The performance of the different language models varied, with the *bge-base-en-v1.5* model exhibiting exceptional accuracy with a precision of 0.90 and MRR of 0.94. In contrast, the *UAE-Large-V1* model showed the poorest scores in precision and MRR, despite being the top-performing model in NLP tasks according to the MTEB leaderboard. The fact that this model was fine-tuned on a wide range of data suggests that the model may struggle to provide representative embeddings for the experimental data without prior fine-tuning.

The study also evaluated the indexing and retrieval execution times for each model. The time taken for indexing a table was low for all the models, ranging between 60 and 160 milliseconds. On the contrary, the retrieval operation required notably more time, taking into account that this process implied retrieving the top-k comparable columns for each column in the query table, computing similarity scores, and re-ranking the results.

Future work involves enhancing the retrieval component by fine-tuning language models specifically on sensor data, which could improve their ability to generate representative embeddings for these datasets. Moreover, further exploring the effectiveness of smaller models, such as *bge-small-en-v1.5*, might offer a lightweight and efficient solution for real-time systems.

Furthermore, the application to other scenarios is a future line that must be explored to demonstrate the effectiveness and flexibility of the proposed system across a wide range of domains.

Author Contributions: Conceptualisation, J.-N.M. and D.T.; methodology, J.-N.M. and D.T.; software, A.B. and A.M.; validation, A.B., A.M. and D.T.; formal analysis, A.B. and D.T.; investigation, A.B. and D.T.; resources, A.M.; data curation, A.B. and A.M.; writing—original draft preparation, A.B., A.M., D.T. and J.-N.M.; writing—review & editing, J.-N.M. and D.T.; visualisation, A.B. and D.T.; supervision, J.-N.M. and D.T.; project administration, J.-N.M.; funding acquisition, J.-N.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by MCIN/AEI/10.13039/501100011033 and by the European Union Next Generation EU/PRTR as part of the projects TED2021130890B-C21 and PID2021-122263OB-C22, as well a by REMARKABLE project (HORIZON-MSCA-2021-SE-0 action number: 101086387). The APC was funded by CIAICO/2022/019 project from Generalitat Valenciana. Alberto Berenguer has a contract for predoctoral training with "Generalitat Valenciana" and the European Social Fund, funded by grant number ACIF/2021/507.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to copyright issues.

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

References

1. McCreadie, R.; Albakour, D.; Manotumruksa, J.; Macdonald, C.; Ounis, I. Searching the Internet of Things. In *Building Blocks for IoT Analytics Internet-of-Things Analytics*; River Publishers: Nordjylland, Denmark, 2022; pp. 39–80.
2. Liu, M.; Li, D.; Xu, C.; Zhou, J.; Huang, W. Discovery of multimodal sensor data through webpage exploration. *IEEE Internet Things J.* **2019**, *6*, 5232–5245. [CrossRef]
3. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.W.; da Silva Santos, L.B.; Bourne, P.E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3*, 160018. [CrossRef] [PubMed]
4. Bodenbenner, M.; Montavon, B.; Schmitt, R.H. FAIR sensor services-Towards sustainable sensor data management. *Meas. Sens.* **2021**, *18*, 100206. [CrossRef]
5. Pattar, S.; Buyya, R.; Venugopal, K.R.; Iyengar, S.; Patnaik, L. Searching for the IoT resources: Fundamentals, requirements, comprehensive review, and future directions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2101–2132. [CrossRef]
6. Liu, M.; Li, D.; Zeng, Y.; Huang, W.; Meng, K.; Chen, H. Combinatorial-oriented feedback for sensor data search in Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 284–297. [CrossRef]
7. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.* **2020**, *63*, 1872–1897. [CrossRef]
8. 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 Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 5998–6008.
9. Li, J.; Tang, T.; Zhao, W.X.; Wen, J.R. Pretrained Language Model for Text Generation: A Survey. In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI-21) International Joint Conferences on Artificial Intelligence Organization, Montreal, QC, Canada, 19–27 August 2021; pp. 4492–4499. [CrossRef]
10. Rivera-Trigueros, I. Machine translation systems and quality assessment: a systematic review. *Lang. Resour. Eval.* **2022**, *56*, 593–619. [CrossRef]
11. Calijorne Soares, M.A.; Parreiras, F.S. A literature review on question answering techniques, paradigms and systems. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *32*, 635–646. [CrossRef]
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
13. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020; Curran Associates, Inc.: New York, NY, USA, 2020; Volume 33, pp. 1877–1901.
14. Smith, S.; Patwary, M.; Norick, B.; LeGresley, P.; Rajbhandari, S.; Casper, J.; Liu, Z.; Prabhumoye, S.; Zerveas, G.; Korthikanti, V.; et al. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. *arXiv* **2022**, arXiv:2201.11990.
15. BigScience Workshop; Scao, T.L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A.S.; Yvon, F.; et al. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv* **2023**, arXiv:2211.05100.

16. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
17. Gur, I.; Nachum, O.; Miao, Y.; Safdari, M.; Huang, A.; Chowdhery, A.; Narang, S.; Fiedel, N.; Faust, A. Understanding HTML with Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics EMNLP 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 2803–2821. [CrossRef]
18. Li, Y.; Yang, T. Word Embedding for Understanding Natural Language: A Survey. In *Guide to Big Data Applications*; Springer: Cham, Switzerland, 2018; pp. 83–104. [CrossRef]
19. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems—Volume 2, Lake Tahoe, NV, USA, 3–6 December 2013; pp. 3111–3119.
20. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
21. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [CrossRef]
22. Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P.S.; He, L. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 1–41. [CrossRef]
23. Schouten, K.; Frasincar, F. Survey on Aspect-Level Sentiment Analysis. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 813–830. [CrossRef]
24. Stahlberg, F. Neural Machine Translation: A Review. *J. Artif. Intell. Res.* **2020**, *69*, 343–418. [CrossRef]
25. Forgues, G.; Pinea, J.; Larchevêque, J.M.; Tremblay, R. Bootstrapping dialog systems with word embeddings. In Proceedings of the NIPS–The Modern Machine Learning & Natural Language Processing Workshop, Montreal, QC, Canada, 12–14 December 2014; Volume 2.
26. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237. [CrossRef]
27. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 328–339.
28. Liu, Q.; Kusner, M.J.; Blunsom, P. A Survey on Contextual Embeddings. *arXiv* **2020**, arXiv:2003.07278.
29. Zhang, S.; Balog, K. Ad Hoc Table Retrieval using Semantic Similarity. In Proceedings of the World Wide Web Conference on World Wide Web—WWW '18, Lyon, France, 23–27 April 2018; ACM Press: New York, NY, USA, 2018; pp. 1553–1562. [CrossRef]
30. Chen, Z.; Trabelsi, M.; Heflin, J.; Xu, Y.; Davison, B.D. Table Search Using a Deep Contextualized Language Model. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 589–598. [CrossRef]
31. Agarwal, V.; Bhardwaj, A.; Rosso, P.; Cudré-Mauroux, P. ConvTab: A Context-Preserving, Convolutional Model for Ad-Hoc Table Retrieval. In Proceedings of the IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–28 December 2021; pp. 5043–5052. [CrossRef]
32. Du, L.; Gao, F.; Chen, X.; Jia, R.; Wang, J.; Zhang, J.; Han, S.; Zhang, D. TabularNet: A Neural Network Architecture for Understanding Semantic Structures of Tabular Data. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 14–18 August 2021; pp. 322–331. [CrossRef]
33. Shraga, R.; Roitman, H.; Feigenblat, G.; Cannim, M. Web Table Retrieval using Multimodal Deep Learning. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 1399–1408. [CrossRef]
34. Nargesian, F.; Zhu, E.; Pu, K.Q.; Miller, R.J. Table Union Search on Open Data. *Proc. Very Large Data Base Endow.* **2018**, *11*, 813–825. [CrossRef]
35. Günther, M.; Thiele, M.; Gonsior, J.; Lehner, W. Pre-Trained Web Table Embeddings for Table Discovery. In Proceedings of the 4th Workshop in Exploiting AI Techniques for Data Management, New York, NY, USA, 20–25 June 2021; pp. 24–31. [CrossRef]
36. Cappuzzo, R.; Papotti, P.; Thirumuruganathan, S. EmbDI: Generating Embeddings for Relational Data Integration. In Proceedings of the 29th Italian Symposium on Advanced Database Systems (SEBD 2021), Pizzo Calabro, Italy, 5–9 September 2021; Volume 2994, pp. 331–338.
37. Herzig, J.; Nowak, P.K.; Müller, T.; Piccinno, F.; Eisenschlos, J. TaPas: Weakly Supervised Table Parsing via Pre-training. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Virtual, 5–10 July 2020; pp. 4320–4333. [CrossRef]
38. Yin, P.; Neubig, G.; tau Yih, W.; Riedel, S. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Virtual, 5–10 July 2020; pp. 8413–8426. [CrossRef]
39. Fang, Q.; Gao, J.; Guibas, L.J. Landmark-Based Information Storage and Retrieval in Sensor Networks. In Proceedings of the IEEE INFOCOM 25th IEEE International Conference on Computer Communications, Barcelona, Spain, 23–29 April 2006; pp. 1–12. [CrossRef]
40. Liu, M.; Li, D.; Chen, Q.; Zhou, J.; Meng, K.; Zhang, S. Sensor Information Retrieval From Internet of Things: Representation and Indexing. *IEEE Access* **2018**, *6*, 36509–36521. [CrossRef]

41. Marinov, M.; Georgiev, G.; Popova, E. NoSQL approach for sensor data storage and retrieval. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 1427–1432. [CrossRef]
42. Albano, M.; Chessa, S.; Nidito, F.; Pelagatti, S. Dealing with Nonuniformity in Data Centric Storage for Wireless Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.* **2011**, *22*, 1398–1406. [CrossRef]
43. Dobson, S.; Golfarelli, M.; Graziani, S.; Rizzi, S. A Reference Architecture and Model for Sensor Data Warehousing. *IEEE Sens. J.* **2018**, *18*, 7659–7670. [CrossRef]
44. Singla, K.; Bose, J.; Varshney, N. Word Embeddings for IoT Based on Device Activity Footprints. *Comput. Sist.* **2019**, *23*, 1043–1053. [CrossRef]
45. Ma, Q.; Muthukrishnan, S.; Simpson, W. App2Vec: Vector modeling of mobile apps and applications. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Davis, CA, USA, 18–21 August 2016; pp. 599–606. [CrossRef]
46. Sharma, A.; Jain, A.; Gupta, P.; Chowdary, V. Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access* **2020**, *9*, 4843–4873. [CrossRef]
47. Vaz, P.J.; Schütz, G.; Guerrero, C.; Cardoso, P.J. Hybrid neural network based models for evapotranspiration prediction over limited weather parameters. *IEEE Access* **2022**, *11*, 963–976. [CrossRef]
48. Zubcoff, J.J.; Olcina, J.; Morales, J.; Mazón, J.N.; Mayoral, A.M. Usefulness of open data to determine the incidence of COVID-19 and its relationship with atmospheric variables in Spain during the 2020 lockdown. *Technol. Forecast. Soc. Change* **2023**, *186*, 122108. [CrossRef] [PubMed]
49. Berenguer, A.; Morejón, A.; Tomás, D.; Mazón, J.N. Using large language models to enhance the reusability of sensor data. *Sensors* **2023**, *24*, 347. [CrossRef]
50. Voorhees, E.M. The TREC-8 Question Answering Track Report. In Proceedings of the 8th Text REtrieval Conference (TREC), Gaithersburg, MD, USA, 17–19 November 1999; Volume 99, pp. 77–82.
51. Pilaluisa, J.; Tomás, D.; Navarro-Colorado, B.; Mazón, J.N. Contextual word embeddings for tabular data search and integration. *Neural Comput. Appl.* **2023**, *35*, 9319–9333. [CrossRef]
52. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
53. Robertson, S.E.; Walker, S.; Jones, S.; Hancock-Beaulieu, M.M.; Gatford, M. Okapi at TREC-3. In Proceedings of the 3rd TREC Conference, Gaithersburg, MD, USA, 2–4 November 1994; Volume 500–225, pp. 109–126.
54. Li, X.; Li, J. AnglE-optimized Text Embeddings. *arXiv* **2023**, arXiv:2309.12871.
55. Xiao, S.; Liu, Z.; Zhang, P.; Muennighof, N. C-Pack: Packaged Resources To Advance General Chinese Embedding. *arXiv* **2023**, arXiv:2309.07597.
56. Li, Z.; Zhang, X.; Zhang, Y.; Long, D.; Xie, P.; Zhang, M. Towards General Text Embeddings with Multi-stage Contrastive Learning. *arXiv* **2019**, arXiv:1908.10084.
57. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv* **2019**, arXiv:1908.10084.
58. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models. *arXiv* **2023**, arXiv:2303.18223.
59. Lukas, N.; Salem, A.; Sim, R.; Tople, S.; Wutschitz, L.; Zanella-Béguelin, S. Analyzing Leakage of Personally Identifiable Information in Language Models. In Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–25 May 2023; pp. 346–363. [CrossRef]
60. Carlini, N.; Tramèr, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, Ú.; et al. Extracting Training Data from Large Language Models. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21) USENIX Association, Online, 11–13 August 2021; pp. 2633–2650.
61. Mattern, J.; Mireshghallah, F.; Jin, Z.; Schoelkopf, B.; Sachan, M.; Berg-Kirkpatrick, T. Membership Inference Attacks against Language Models via Neighbourhood Comparison. In Proceedings of the Findings of the Association for Computational Linguistics (ACL 2023), Toronto, ON, Canada, 2–6 May 2023; pp. 11330–11343. [CrossRef]
62. Huang, H.; Zhao, Z.; Backes, M.; Shen, Y.; Zhang, Y. Composite Backdoor Attacks against Large Language Models. *arXiv* **2023**, arXiv:2310.07676.
63. Zhao, H.; Chen, H.; Yang, F.; Liu, N.; Deng, H.; Cai, H.; Wang, S.; Yin, D.; Du, M. Explainability for Large Language Models: A Survey. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 20. [CrossRef]