PROCEEDINGS OF SCIENCE

An Artificial Intelligence-based service to automatize the INFN CNAF User Support

Elisabetta Ronchieri,[a,b,*] Matteo Barbetti,[a] Alberto Trashaj,[b] Carmelo Pellegrino,[a] Daniele Cesini,[a] Carmen Giugliano,[a] Daniele Lattanzio,[a] Lucia Morganti,[a] Alessandro Pascolini,[a] Andrea Rendina[a] and Aksieniia Shtimmerman[a]

[a] *INFN CNAF*
Viale Berti Pichat 6/2, Bologna, Italy

[b] *Department of Statistical Sciences, University of Bologna,*
Via Belle Arti 41, Bologna, Italy

E-mail: elisabetta.ronchieri@cnaf.infn.it, matteo.barbetti@cnaf.infn.it, albertotrashaj31@gmail.com, carmelo.pellegrino@cnaf.infn.it

[*] Speaker

The INFN CNAF *User Support* unit acts as the first-contact interface between the users and the CNAF data center that provides computing resources to over 60 scientific communities in the fields of Particle, Nuclear and Astro-particle Physics, Cosmology and Medicine. Since its duties span from repetitive tasks to supporting complex scientific-computing workflows, there is room for enabling automation mechanism by relying on modern Artificial Intelligence (AI) techniques that have recently shown to successfully cope Natural Language Processing (NLP) problems. Indeed, part of the users' requests cannot be directly addressed without the intervention of one of the other specialized INFN CNAF units that act as a second level of support. In these cases, disposing of an automatic AI-based labeling can be exploited to promptly notify the relevant units with the pending requests. Over the many years of activity of the User Support group, several thousands of users' bilingual e-mail messages, both in Italian and English, have been received. Such collection of e-mails provides the ideal sample for training Machine Learning (ML) models, and validating them with new coming users' requests. These messages can be organized in threads including user requests together with the corresponding solutions, as well as the messages of the involved second-level support unit, which are implicitly labelled by the recipients list of the e-mail. In this study, we have applied a set of Machine Learning classification models, such as k-Nearest Neighbors, Random Forest, Extreme Gradient Boosting, and Feed-forward Neural Network, to the features extracted through NLP solutions aiming to automatize the e-mail labeling. The performance of the defined models has been compared by considering various feature extraction techniques, such as Bag of Words, Term Frequency - Inverse Document Frequency, Bag of *n*-Grams, and WordEmbedding. Ongoing developments aim to involve the best performing model in combination to Large Language Models (e.g., GPT-3.5, Llama 2) to build an AI-powered Digital User Support Assistant. It will be designed to receive text via e-mail and provides a reply based on the acquired knowledge base. A first prototype has been implemented in Python through the usage of several ML/AI libraries, among them nltk, scikit-learn, and LangChain. A set of User Supporters have been involved for test and validation. In conclusion, our study not only showcases the technical prowess of AI in enhancing the INFN CNAF User Support activities, but also emphasizes the broader considerations of user satisfaction, scalability, and future readiness.

International Symposium on Grids and Clouds (ISGC2024)
24-29 March, 2024
Academia Sinica Computing Centre (ASGC), Institute of Physics, Academia Sinica Taipei, Taiwan

1. Introduction

INFN CNAF [1] focuses on technological development and knowledge transfer to heterogeneous experiments, national and European projects, and industry (whenever possible). It hosts over 1500 active users, over 60 experiments (not only LHC and not only from the Physics field), to which provides over 65,000 CPU cores, 70 PB of disk, and 130 PB of tapes for long-term storage.

The *User Support* (US) unit represents the INFN CNAF entry point [2], whose main mission is to help the users with any problem encountered accessing the computing resources. The US unit is also responsable for writing/updating the documentation, whose aim is to improve the usage of solutions and standard tools the Data Center provides: as an example, HTCondor, a batch system for High Throughput Computing (HTC) centers, SLURM, a batch system for High Performance Computing (HPC) centers, and `gfal2-util`, which is the tool for data transfer/management via Grid. Among the US activities, one can find the onboarding of new scientific communities (e.g., projects and experiments), supporting the use of specific software (sometimes through user scripts and customized environment), managing the user registration procedure concerning recognition, authorization, account creation issues, writing documentation for users (e.g. INFN CNAF Tier-1 User Guide [3], communicating with users through personal emails, chat, mailing list, special events, and dedicated meetings with experiments' people.

The data produced by the regular US activities [4] provides the ideal samples to design, train, and validate *Artificial Intelligence* (AI) solutions. Advanced AI techniques and Generative AI (GenAI) models have been investigated to assist the INFN CNAF unit in supporting Tier-1 users and adopting the latest software technologies. In this study we have considered the following research questions:

RQ1 Can AI-based technologies successfully be integrated into the daily US activity?

RQ2 Can AI-based technologies efficiently support INFN Tier-1 users?

2. Methodology

During this study, we have aimed to deal with the following two tasks according to the workflow described in Figure 1:

1) automatize the e-mail classification, guaranteeing the inclusion of specific INFN CNAF units to deal with specific problems;

2) manage a high-latency reply to users' e-mails based on the collected e-mails and the knowledge base through the definition of an e-mail-based framework.

The first task relies deeply on Natural Language Processing (NLP) techniques. In particular, once got raw text data, the following steps are performed: text data-cleaning, dara pre-processing, feature engineering, models building, training, and performance evaluation. To perform the multi-label classification, one needs to prepare the set of e-mails to be used during the training phase, transforms text features into numerical features, trains a set of models, and compares the obtained results. One can also use embedding models to preprocess texts and select features.

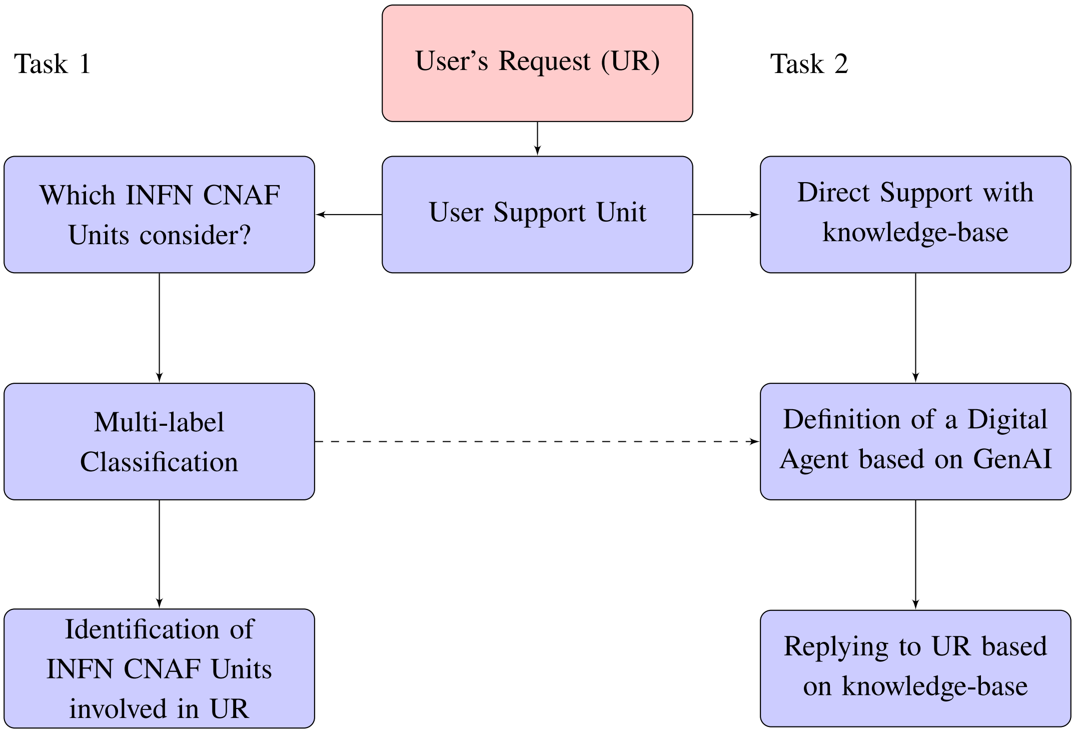
Task 1
User's Request (UR)
Task 2
Which INFN CNAF Units consider?
User Support Unit
Direct Support with knowledge-base
Multi-label Classification
Definition of a Digital Agent based on GenAI
Identification of INFN CNAF Units involved in UR
Replying to UR based on knowledge-base

Figure 1: Methodology workflow.

The second task exploits Retrieval-Augmented Generation (RAG) model [5] that combines advanced text searching capabilities with natural language generation: the retrieval component is responsible for recovering relevant information from a predefined knowledge base, while the generation component is responsible for producing new text based on the retrieved information and the input query.

3. Task 1: automatize the e-mail classification

For task 1, we have first extracted a set of e-mails based on issues addressed by the US team. The considered e-mails cover the period 06/2017 - 05/2023. The original e-mails have been examined and converted into JSON files, stripped of most of the HTML content and attachments, and base64 decoded. Data is thus represented by a set of 28,500 JSON files for a total size of 260 MB. An e-mail example is as follows:

```
{
  "from": "7c4351dc-63a9-4d7a-bde5-c94b3183e36d",
  "to": "c77ab8f7-ab8d-4a1e-8d98-206957010e49",
  "subject": "Fwd: [exp-sup-cnaf-gr2] [user-support] INFN-T1
     Down on ...",
  "content": "Just a reminder for the intervention of today
     ....",
  "type": "text/plain",
  "date": "Thu, 1 Oct 2020 08:03:41 +0000",
  "cc": [""],
  "id": "<E3F638AC-FC4A-4473-922F-93D5C29BB87D@cern.ch>",
  "parent": null
}
```

where the `from`, `to` and `parent` keys contain e-mail information as value. If the `parent` is not *null*, the e-mail message is a reply to another JSON file.

All the considered e-mails have been opportunely anonymized with a unique identifier whenever possible. All references to people in the e-mail body have been anonymized with a placeholder coded with `[NAME]`. The anonymization has been performed by using ad-hoc BASH and Python scripts and manual changes to metadata.

Furthermore, each e-mail has been categorized (i.e. properly labelled) based on the text in the e-mail body. The categories concerning the INFN-CNAF Unit and fundamental activities have been defined, such as *user-support*, *computing*, *storage*, *networking*, *research and development* (R&D), *account request*, *high-performance computing* (HPC), and *others*. All labels belonging to e-mails of the same threads have been propagated up to the thread's first message.

Figure 2 shows the number of first e-mails per category: there have been observed about 30k e-mails. Figure 3 shows the total number of e-mails per category: it can be observed that the e-mail category can change in the same e-mail thread over time.

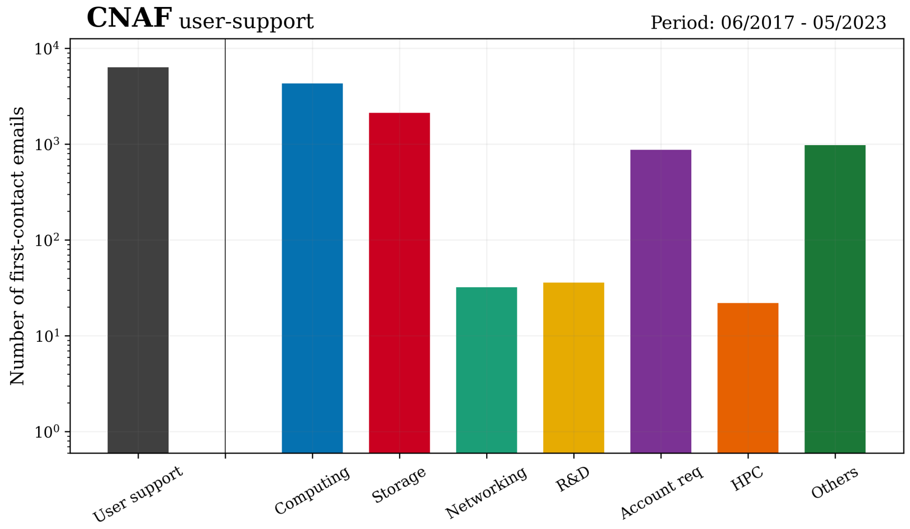

Figure 2: Number of first e-mail per category.

The text cleaning step involves restructuring the e-mail body, and considers handling the e-mail text with both a sequence of sentences and pieces of code. The former one is needed because the e-mail body can be written in plain text or contain HTML tags. The latter is for the following reasons: text language can be both in English and Italian or just in one of the two languages; code is usually written in English or uses English terms. At the end of this step, texts in the e-mails contain just words, placeholder and punctuation.

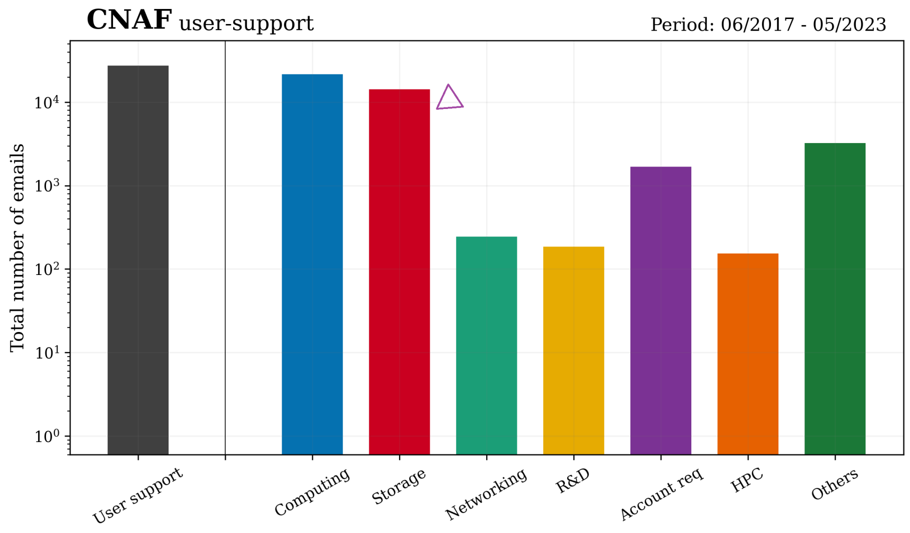

Figure 3: Total number of e-mails per category.

Figure 4 shows the distribution of e-mail threads (96 is the maximum length) over the period considered. There is an intense interaction among the users, the US unit, and the other INFN-CNAF units.

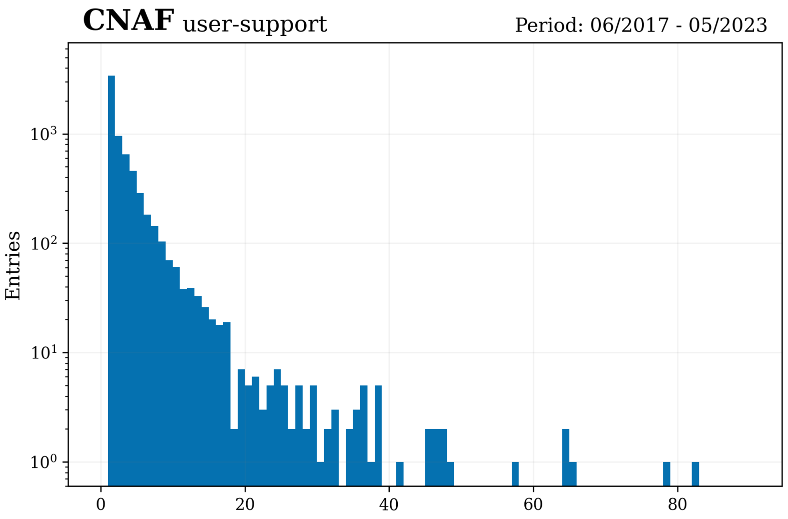

Figure 4: Length of e-mail threads.

During the pre-processing step, unstructured text is transformed into a structured one to prepare it for the analysis via an NLP standard procedure. We have removed the noise and other inconsistencies in the text, made the text easier to understand, and performed tokenization, normalization, and lemmatization operations to reach the base form of each word and consequently reduce the number of the overall features. Since any Machine Learning (ML) algorithm requires fixed-length numeric vectors, the pre-processed text is translated into numeric vectors through vectorization. A set of NLP techniques, summarized in Table 1, have been used to identify features relevant to our study.

Bag of Words (BoW)	***n*-Grams**	**TF-IDF**	**Conti. BoW**	**Skip-GRAM**
– describes e-mails by word occurrences – throws out word order	– describes e-mails by *n*-contiguous words occurrences – keeps word order and some context info	– highlights uncommon words in e-mail, and more common words across e-mails	– predicts a word, leveraging all words in its neighborhood context	– predicts the context words based on a word

Table 1: Feature selection techniques used in this study.

The pre-processing and feature selection steps can be performed by using alternative approaches. They consider sentence-transformer models available on HuggingFace [6]. They derive from BERT-based *fine-tuned* models. The models have been designed for semantic textual similarity, semantic search, or paraphrase mining. The multilingual models support over 50 languages, including Italian. The pre-processing and vectorization steps are performed by the sentence-transformers models summarized in Table 2.

Model Name	all-mpnet-base-v2	all-MiniLM-L6-v2	paraphrase-multilingual-mpnet-base-v2	paraphrase-multilingual-MiniLM-L12-v2
Model Base	MPNet	MiniLM	XLM-RoBERTa	MiniLM
Quality	high	high	good	good
Speed	1	×5	∼ 1	×2.5
Dimension	768	384	768	384
Max num tokens	384	256	128	128
Language	English	English	Multiple Lang.	Multiple Lang.

Table 2: BERT-based models.

The embeddings, resulting from the previous step, are employed as *input features* for the following multi-label classifiers: k-Nearest Neighbors (kNN) [7], Random Forest (RF) [8], Extreme Gradient Boosting (XGBoost) [9], and Feed-forward Neural Network (FNN) [10]. The performances achieved by the previous models are further improved by relying on an ensemble (EN) model [11]. The latter results are obtained by combining the output of all the trained models as follows: the balanced approach is a plain average of the output probabilities; the randomized approach is a random combination of the output probabilities; the weighted approach where the FNN and XGB models dominate the classification decision except for the low-represented classes (i.e., HPC, R&D and networking) for which RF is preferred.

For the performance evaluation of the various models we have relied on Area Under the Curve (AUC) and Receiver Operating Characteristics (ROC) as standard metrics. Table 3 shows AUC results obtained for the various models fed with various feature selection techniques.

Figure 5 shows ROC curves for the test set organized according to each category. The results are provided for the FNN model fed with EnLang-MiniLLM embedding model.

Table 4 shows AUC results obtained for the Ensemble (EN) model. Properly combining the output of the most promising models in the EN one, we have obtained AUC equal to 94% for the

	kNN$_{AUC}$	**RF**$_{AUC}$	**XGB**$_{AUC}$	**FNN**$_{AUC}$
BoW	0.843	0.916	**0.938**	0.909
n-Grams	0.860	0.887	0.886	0.880
TF-IDF	0.892	0.897	0.919	0.904
CBOW	0.807	0.837	0.856	0.761
SGRAM	0.816	0.843	0.855	0.787
MPNet (en-only)	0.914	0.895	0.924	0.914
MiniLM (en-only)	0.917	0.895	0.926	**0.927**
XLM-RoBERTa (multi)	0.908	0.890	0.922	0.916
MiniLM (multi)	0.902	0.888	0.915	0.915

Table 3: AUC results.

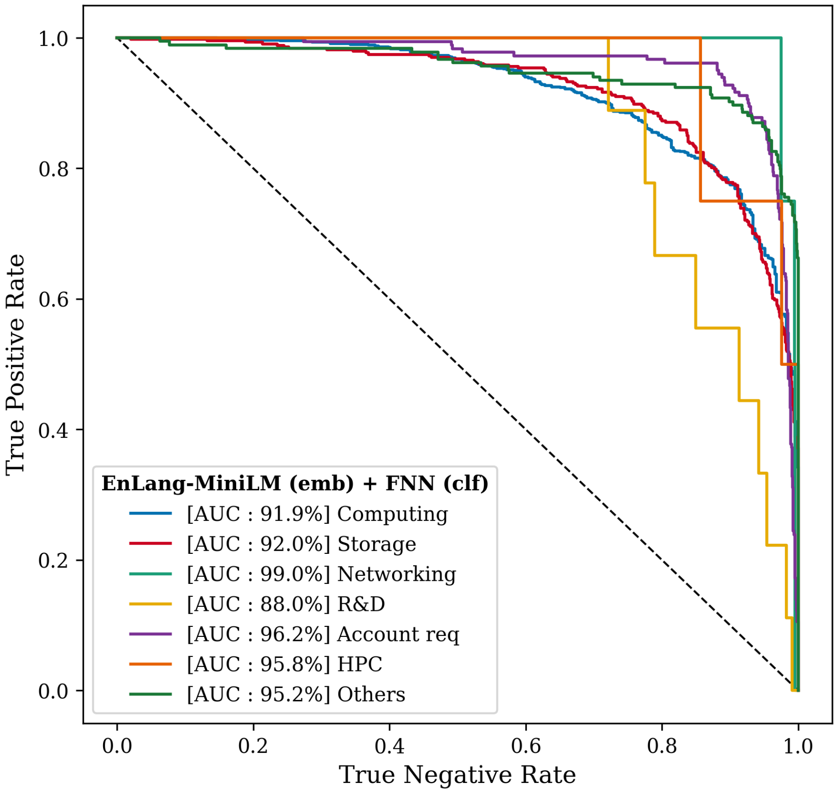

Figure 5: ROC curves for test set with EnLang-MiniLM and FNN models.

EN$_{weighted}$ model. Figure 6 shows ROC curve by considering the EN model.

For this task, we have measured the computational costs requested to train the classifier models. The operation has been performed on the computation nodes of the INFN-CNAF computing farm without using of GPU resources. Figure 7 shows HEPscore [12] per hour for labelling, text cleaning and embeddings. Figure 8 shows HEPscore per hour for training 64 models given by 4 classifiers × 4 embeddings × 4 pre-processing strategies.

	$\mathrm{EN}_{balanced}$	EN_{random}	$\mathrm{EN}_{weighted}$
AUC	0.939	0.939	**0.940**

Table 4: AUC for the EN model.

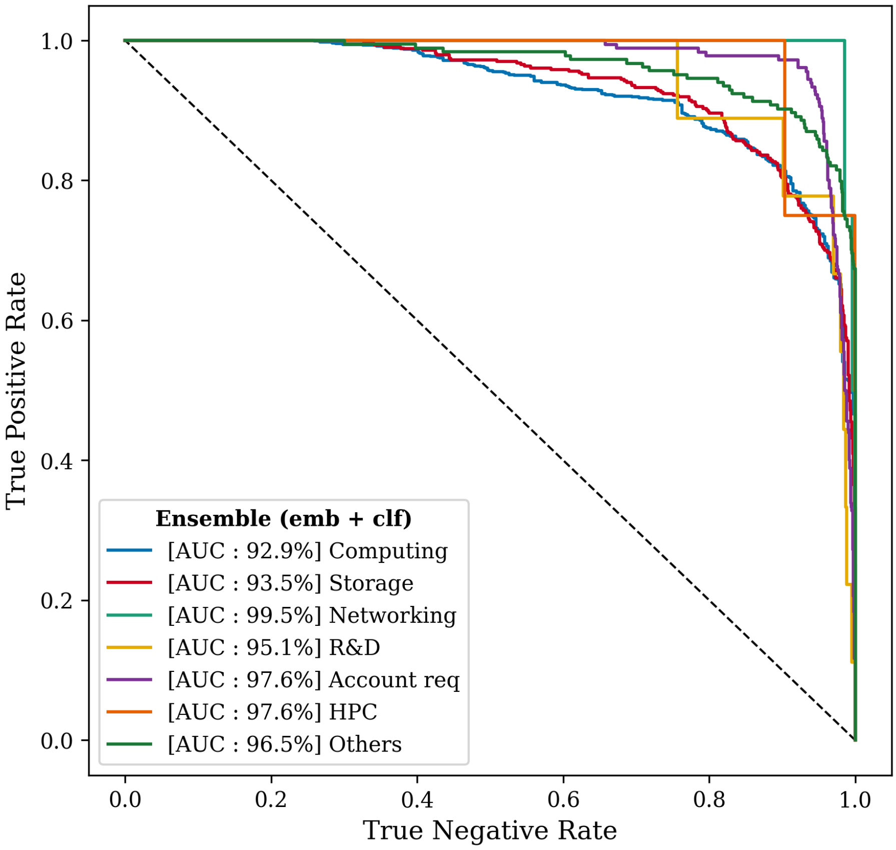

Figure 6: ROC curves for test set with EN model.

4. Task 2: manage a high-latency reply to users' e-mails

As second task, we have considered a RAG model. It is composed by a parametric-memory generation model with a non-parametric memory retriever. The retrieval component employs techniques such as *semantic search* to find the most relevant pieces of information related to the input query. The generation component relies on *Generative Models* powered by the attention mechanism and implementing a transformer architecture [13]. A schematic representation of the RAG model is depicted in Figure 9.

The User Guide [3], written and maintained by the US team, represents the knowledge base to feed our model. Once properly scraped, the information contained in the documentation has been transformed into vectors through the use of the `all-mpnet-base-v2` model from HuggingFace [14]. The embedding result has been stored in a vector database implemented by using Chroma. Chromais able to keep vectors of large size with the corresponding metadata, such as the

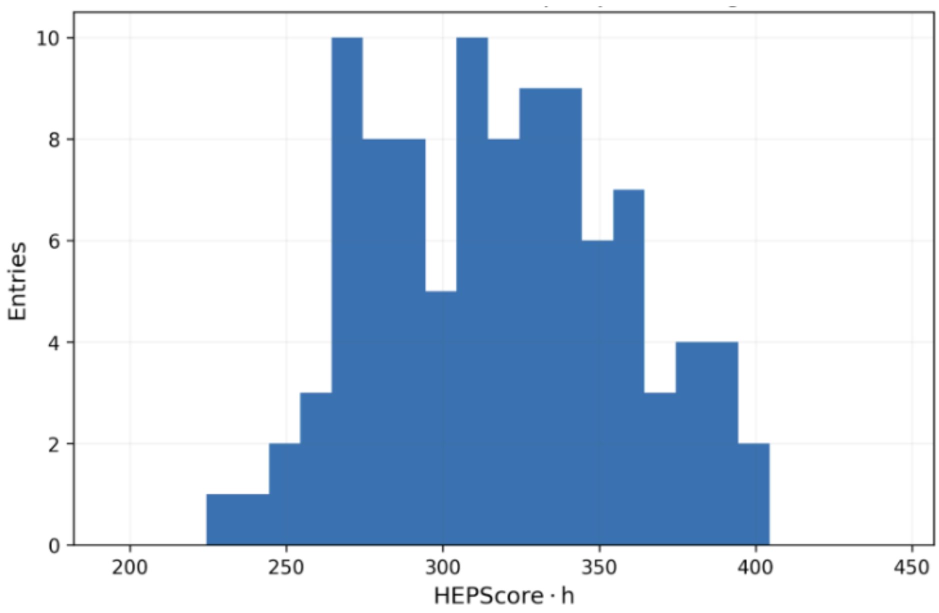

Figure 7: Benchmark for data pre-processing.

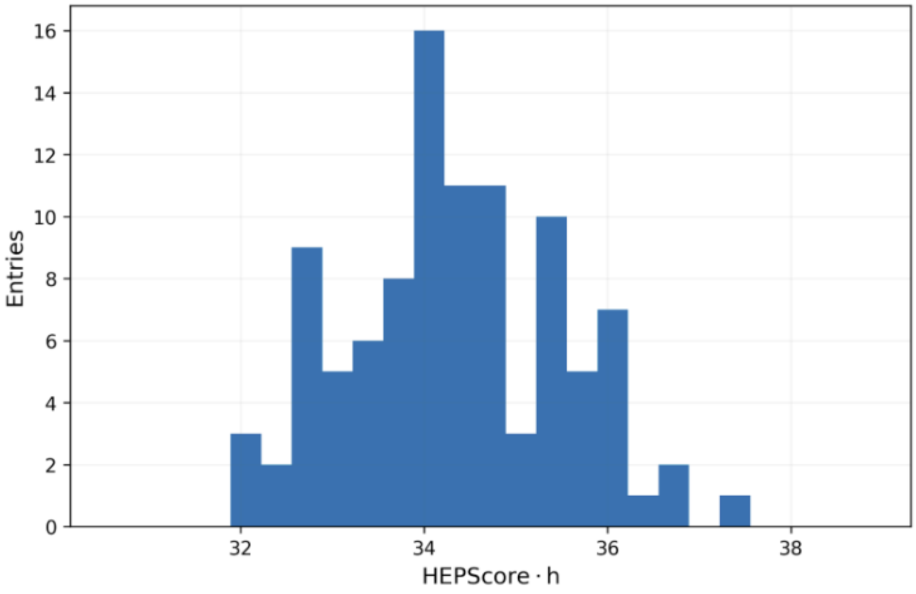

Figure 8: Benchmark for model training.

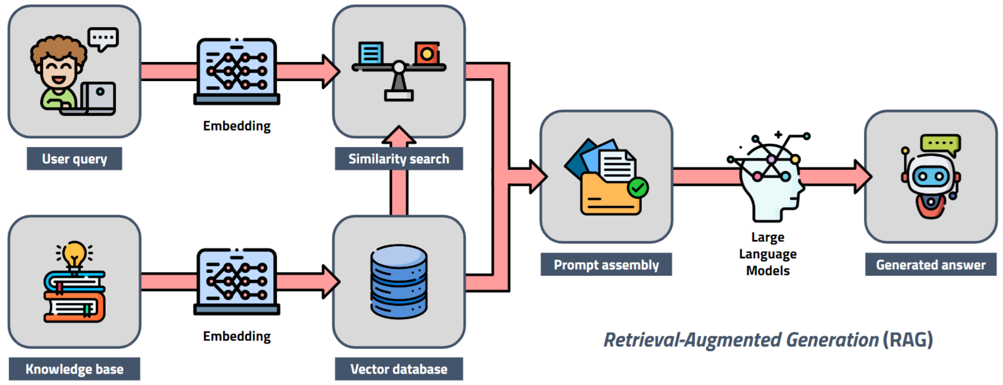

Figure 9: RAG-based Digital Agent.

source text, and to allow us to perform efficient queries to the vector space.

Given the user question *"Dear User Support, is it possible to use Jupyter Notebook?"*, the following activities take place:

- Embedding the user question by the MPNet-based model, similarly to what has been done to the knowledge base (i.e., the User Guide).
- The embedded query can now be used to perform a similarity search and retrieve from the knowledge base stored in the vector database the *most similar vectors* (MSVs) to the user question.
- The MSVs can then be processed by an advanced language model capable to use the information contained in the User Guide to generate a reply for the user question.

A prototype for this question-reply framework has been developed by relying on a *open-source* Large Language Model (LLM). In particular, it is based on Llama 2 [15, 16] with 7B parameters. To cope the query mentioned above, the RAG model powered by Llama 2 takes in input both the query and the related information retrieved from the User Guide, and generate a proper reply by following the template prompt reported below:

```
Dear User,

Thank you for reaching out to us regarding your query about using
Jupyter notebooks. Based on the context provided, it appears that
at Tier-1, Jupyter notebooks are now available through JupyterHub.
To access these notebooks, you will need to log in with your account
bastion credentials and select the experiment that has pledged CPU
resources on the batch system. Once logged in, you can customize
the Jupyter environment as per the instructions provided.

To answer your question directly, yes, you can use Jupyter notebooks
```

```
through the JupyterHub service at Tier-1. The service is reachable via
browser at the following page: <https://jupyterhub-t1.cr.cnaf.infn.it>.
Please note that you will need to have an account and be logged in to
access the notebooks.

If you have any further questions or concerns, please feel free to ask.

Best regards,

[Your Name]
```

Even if the LLM is not expert that a specific topic, it succeeds in extracting the correct context by relying on the RAG architecture, and in formulating a valid solution to the user problem according to the defined template form, namely structuring an e-mail that reports what described in the User Guide.

5. Conclusions

In this work, we have presented the preliminary work done to introduce automatic procedures within the INFN-CNAF US activities by relying on AI-based technologies. Our results show that the knowledge base contained within the CNAF experts e-mails and the User Guide can be retrieved by means of AI techniques, and employed to support users and help with the daily activities of the Data Center. Modern NLP techniques can be successfully employed to face the classification task, reaching a score of 0.94 AUC by using the Ensemble$_{weighted}$ model. The prototype of the question-reply framework has shown promising results, demonstrating its capabilities to assist the US activities. Validation campaigns are currently ongoing to test the limits and reliability of the developed framework.

As future developments we plan to test the performance achieved by using the latest open-source Large Language Models, like Llama 3 [17] and Gemma 2 [18] for both the classification and generation tasks. Furthermore we aim to extend the knowledge base available to the RAG model with additional documentations coming from the Internet and potentially useful for the Tier-1 users, enhancing and simplifying the daily activities of the INFN-CNAF US unit.

Acknowledgments

The work presented in this paper is partially performed in the framework of the ICSC project – *Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing*, funded by the NextGenerationEU European initiative through the Italian Ministry of University and Research, PNRR Mission 4, Component 2: Investment 1.4, Project code CN00000013 - CUP I53C21000340006.

References

[1] INFN CNAF, "INFN CNAF Web Page." https://www.cnaf.infn.it.

[2] C. Pellegrino, D. Cesini, F. Fornari, D. Lattanzio, L. Morganti, A. Pascolini et al., *Support for experiments at INFN-T1*, *EPJ Web of Conferences* **295** (2024) 08019.

[3] INFN CNAF, "INFN CNAF Tier-1 User Guide (July 2024 - v19)." `https://l.infn.it/t1guide`.

[4] D. Lattanzio, A. Rendina, A. Pascolini, L. Morganti, F. Fornari, C. Pellegrino et al., *Experiments support at INFN-T1*, in *Proceedings of International Symposium on Grids & Clouds (ISGC) 2023 in conjunction with HEPiX Spring 2023 Workshop – PoS(ISGC&HEPiX2023)*, 2023, DOI.

[5] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi et al., *Retrieval-Augmented Generation for Large Language Models: A Survey*, `2312.10997`.

[6] T. Wolf, L. Debut, J. Sanh, Victor et alland Chaumond, C. Delangue, A. Moi, P. Cistac et al., *HuggingFace's Transformers: State-of-the-art Natural Language Processing*, `1910.03771`.

[7] A. Mucherino, P.J. Papajorgji and P.M. Pardalos, *k-Nearest Neighbor Classification*, in *Data Mining in Agriculture*, pp. 83–106, Springer New York (2009), DOI.

[8] L. Breiman, *Random Forests*, *Machine Learning* **45** (2001) 5.

[9] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, `1603.02754`.

[10] N. Ketkar and J. Moolayil, *Feed-Forward Neural Networks*, in *Deep Learning with Python*, pp. 93–131, Apress (2021), DOI.

[11] F. Acito, *Ensemble Models*, in *Predictive Analytics with KNIME*, pp. 255–265, Springer Nature Switzerland (2023), DOI.

[12] D. Giordano, J.-M. Barbet, T. Boccali, G.M. Borge, C. Hollowell, V. Innocente et al., *HEPScore: A new CPU benchmark for the WLCG*, *EPJ Web of Conferences* **295** (2024) 07024.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez et al., *Attention Is All You Need*, in *31st International Conference on Neural Information Processing Systems (NeurIPS)*, 6, 2017 [`1706.03762`].

[14] Hugging Face, "Hugging Face Web page." `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`.

[15] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix et al., *LLaMA: Open and Efficient Foundation Language Models*, `2302.13971`.

[16] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei et al., *Llama 2: Open Foundation and Fine-Tuned Chat Models*, `2307.09288`.

[17] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman et al., *The Llama 3 Herd of Models*, `2407.21783`.

[18] Gemma Team collaboration, *Gemma 2: Improving Open Language Models at a Practical Size*, 2408.00118.