EUROCC ITALY
HPC Competence Center

VECTORIA:

a private and sustainable enterprise AI solution

Authors: Leonardo Baroncelli, Carolina Berucci, Eric Pascolo, Andrea Proia

Summary

AI at Work: Should Your Model Sign a Non-Compete Agreement?

In the contemporary business environment, artificial intelligence has become central, not only as a technological tool but as an authentic collaborator. An AI algorithm can work alongside human colleagues, acquiring skills through interaction and their experience. Integrated into various business functions, it supports both daily activities and complex roles, analyzing data and learning autonomously to help manage information and optimize processes.

Working side-by-side with human staff enables the AI algorithm to understand business dynamics and to acquire sector-specific knowledge, constantly improving its effectiveness. Continuous interaction with the human team enables AI to better understand work contexts and to rapidly adapt to volatile market needs. AI not only performs operative tasks, but also supports strategic decisions, providing detailed analysis and forecasts based on hard data. Its presence ensures a continuous flow of up-to-date relevant information critical for maintaining the company's competitiveness.

Losing information gained by AI is equivalent to losing valuable human resources. Every piece of data collected, and every experience assimilated represents added value that must be preserved to avoid favoring competitors. Like a human colleague, AI possesses expertise that needs to be protected and treasured.

Moreover, the AI model does not merely replicate the expertise of a single individual but collects and integrates knowledge transmitted by several people through daily interactions. This makes it a container of diverse and interconnected skills, spanning multiple sectors and business units. The loss of information accrued by AI is therefore a considerable challenge, as it involves the waste of a collective wealth of knowledge, result of the combined effort and experience of a whole team. Protecting this information is not only a matter of security, but of preserving a competitive advantage gained from the integration of multiple perspectives and expertise within a single digital entity.

Leaking information is equivalent to yielding a competitive advantage to other market players. It is therefore crucial to implement data protection systems that can ensure the confidentiality and security of AI-managed information, thereby safeguarding business capital.

When using AI tools based on online APIs, data privacy can be compromised, as processed information could leak to external servers, putting enterprise confidentiality

at risk. In contrast, solutions such as Vectoria, which implement AI systems completely on premises, offer a higher degree of security. This approach ensures that all operations of information retrieval and response generation take place within the company's private infrastructure, eliminating the risk of sensitive data being exposed to third parties. Information protection then becomes a competitive advantage, ensuring that the wealth of knowledge accumulated by AI remains confined and protected, preserving the integrity and security of company data.

LLM and RAG: two different ways of learning

Currently, complex tasks are deferred to models such as Large Language Models (LLMs). In this section, we will compare the basic usage of these models with a technique that enhances their potential, called Retrieval-Augmented Generation (RAG). LLMs are AI models that process information and generate content. Pre-trained on large amounts of text data, they are able to understand and produce natural language with high accuracy. RAG systems overcome the limitations of LLMs by retrieving specific information, offering precise and contextualized answers. This makes them ideal for business applications that require high accuracy in data management.

Large Language Models (LLMs): traditional generative intelligence

LLMs are artificial intelligence systems pre-trained on vast amounts of heterogeneous data. This training phase allows them to acquire an incredible ability to model natural language and generate responses that are coherent and linguistically sophisticated. However, their proficiency is based on statistical correlations among the data seen during training, which leads to some significant limitations:

- **Dependence on trained data**: LLMs cannot access information that is new or not present in the original dataset. This implies that a pre-trained model has a temporally limited knowledge and cannot know confidential information of companies or individuals.
- **Tendency for "hallucinations"**: as LLMs are probabilistic models, they can generate answers that are inaccurate or not based on real facts, generating so-called "hallucinatory" phenomena.
- **Knowledge scalability**: to increase precision and development of more sophisticated expressive capabilities, increasingly large and articulated training datasets are required, resulting in high computational costs.

Retrieval-Augmented Generation (RAG): the hybrid between generation and search

RAG pipelines mitigate the limitations of LLMs by adopting a modular approach, in which response generation is coupled with information retrieval. This methodology features two main steps:

1. **Retrieval**: relevant content is searched in a structured repository, usually represented by a vector database. It stores documents in the form of multidimensional vectors, following a subdivision into semantic units called “chunks”.
2. **Generation**: retrieved data is integrated in the context of the generative model, which uses it to produce accurate and up-to-date answers.

Thanks to this architecture, RAG pipelines offer several advantages:

- **Access to up-to-date information**: they allow for the integration of new or domain-specific data without having to retrain the generative model, thus enabling large reductions in training-related costs.
- **Customization**: they can be tailored to vertical domains and business applications. Ideally, the entire company documentation can be provided to the RAG pipeline for reference, like an encyclopedia, in order to obtain specific, well-contextualized answers.
- **Data privacy**: information management can be confined within secure infrastructures, ensuring the protection of sensitive data and information.

In summary, while LLMs represent a monolithic and generalist approach, RAG pipelines enable a flexible system that can combine the generative power of LLMs with the accuracy of targeted search, better adapting to dynamic and specific scenarios.

AI solutions: do we really need massive computing power to build them?

The amount of computing power needed, as mentioned in the title, can differ widely: it might be as little as what a standard laptop provides or as much as what's found in a supercomputer.

The heart of the matter lies in the **components** that make up the RAG system, in particular the **AI models** it uses for its two main stages: information **retrieval** and answer **generation**.

1. **The "Researcher" (Retrieval Model)**: the first stage involves finding relevant information within a large knowledge base (documents, databases, websites...). This is where a specialized model comes in to "understand" the user's query and search for the most relevant pieces of text.
    - **"Light" scenario:** if a relatively simple search model is used, for example based on small keywords or "embeddings" (numerical representations of the meaning), the required resources are modest. A good CPU and a reasonable amount of RAM may be enough.
    - **"Heavy" scenario:** if, on the other hand, a very sophisticated retrieval model is employed, capable of understanding complex nuances of language and of semantically parsing huge amounts of data with high-dimensionality embeddings, the computational demand increases. More powerful processors (often GPUs) and a lot more memory will be needed to handle the necessary indices and computations.
2. **The "Writer" (Generative model - LLM):** Once the relevant information is found, it is passed to an LLM, which is tasked with using it to formulate a coherent and complete answer. This is where the differences can become enormous.
    - **"Light" scenario:** smaller LLMs can be used, with relatively few "parameters" (the internal variables that the model learns during training, in the order of a few billions). These models, while still capable, require less memory (RAM and GPU VRAM) and computational power. A RAG system based on a compact LLM could efficiently run on a high-end laptop or a decent workstation.
    - **"Heavy" scenario:** If the application requires the highest quality, creativity, and complex reasoning capabilities, one will opt for state-of-the-art LLMs, colossal models with hundreds of billions or even trillions of parameters. These giants require huge amounts of memory (tens or hundreds of gigabytes of VRAM) and massive parallel computing power, provided by advanced GPU clusters. Loading and running inference on these models is a task that often requires dedicated infrastructure, powerful servers, or in the most extreme cases, access to supercomputers.

By combining retrieval and generation models of different complexities and sizes it is possible to build RAG systems to suit specific needs and budgets; there are also intermediate cases to those listed above. It is possible to start with a "light" configuration on affordable hardware for specific tasks, then scale up to much more powerful – and expensive – solutions when high-level performance and capabilities are

required. The choice depends on the desired balance between cost, speed, and quality of the final answer.

Vectoria: a RAG workflow for a privacy-friendly LLM

Vectoria is an innovative solution developed as part of the European EuroCC2 project, designed to address the growing need for increased awareness, expertise and technological development through advanced artificial intelligence systems. Designed specifically for enterprise use, Vectoria enables fast and secure access to internal documentation through chatbot-like conversational interfaces. The strategic objectives of Vectoria are:

- **Centralization of corporate knowledge**: the dispersion of documentation across multiple repositories is solved by consolidating data into a single structure that is easily accessible by the generative model. This is done automatically by the software library which implements the RAG pipeline and is easily configurable to meet different demands.
- **Search accuracy**: retrieval techniques on vector databases using similarity metrics allow the LLM to identify the most relevant information with respect to user queries. This information allows the LLM to respond in a precise manner, avoiding "hallucinations" or irrelevant answers.
- **Respect for privacy**: all operations take place within the company's private infrastructure, avoiding private data leaks to external generative AI services. There is a total guarantee of privacy as well as avoiding the need to subscribe and manage subscriptions to the aforementioned external services.

Vectoria's RAG pipeline workflow

Vectoria uses a modular and flexible architecture designed to handle all the necessary steps to generate contextually relevant responses. Here is a detailed description of the key components:

1. **Initial configuration**: the system is highly customizable through configuration files that allow Vectoria to be tailored to specific needs. Flexibility of use then becomes a cornerstone, allowing the system to be specialized as needed: different types of documents can be processed, and the underlying

encoding/generation models, retrieval parameters, prompts, etc. can all be modified.

2. **Document preprocessing**: the system allows for the loading and cleaning of documents, which are processed to remove unnecessary elements that could "spoil" the information content of the text. Subsequently, division into "chunks" takes place: each document is fragmented into manageable portions of text, in order to obtain well-divided and self-contained information, optimizing the granularity of data for the search phase.
3. **Embedding generation**: using pre-trained open-source encoding models, chunks are converted into numerical vectors that capture their semantic meaning. This operation is necessary to be able to exploit similarity metrics between vectors, allowing the questions asked by users to be compared to the information contained in the available documentation. In doing so, only the most relevant pieces of information are retrieved.
4. **Vector database storage**: embeddings are stored using a vector database, alongside metadata that enhances filtering and search capabilities. A vector database stores, manages, and indexes high-dimensional vector data. Dat ais stored as numerical arrays that are grouped based on similarity, allowing low-latency queries. Unlike traditional relational databases (with rows and columns), in a vector database data is represented by vectors with a fixed number of dimensions. Each dimension of the dense vector corresponds to a latent feature or aspect of the data, *i.e.*, an underlying attribute that is not observed directly but is inferred from the data through mathematical models or algorithms.
5. **Inference and response**: the user query is converted into an embedding and compared with embeddings in the database, to identify the most similar chunks. The selected chunks are used to construct a prompt that, combined with the generative model, generates an accurate and contextualized response.

Integration with supercomputers: Vectoria is "HPC-Ready"

A distinctive feature of Vectoria is the ability to use High-Performance Computing (HPC) infrastructure to power the entire pipeline. This choice allows for the increased scalability of the system, enabling it to handle large volumes of data and use generative models that are prohibitively large for a normal workstation. No modifications to the underlying code are required: since the library is "HPC-ready", it is simply possible to use a specific configuration filed for use on HPC that interfaces with the appropriate job scheduler. Furthermore, using Vectoria in a high-capacity computing infrastructure increases the speed of the system, resulting in reduced processing times both during the retrieval phase and the response generation.

Leonardo Spa – use case of Vectoria in corporate activities

Leonardo S.p.A. is a multinational company operating globally in the Defense, Aerospace, and Security sectors. The company has always been committed to technological innovation, aiming to optimize its business processes and improve operational efficiency in all areas in which it is involved.

The Business Management System (BMS) is a strategic platform that serves as a central point for storing and managing company documentation. Leonardo's BMS also includes a body of documentation describing company procedures for various operational aspects, from internal processes to safety and quality management. These procedures are critical for ensuring consistency and efficiency in day-to-day activities, but consulting and using them often requires significant effort, especially for new hires, who may not yet be fully familiar with company practices. In the context of digitalization and technological transformation, Leonardo has sought to make accessing and understanding this corporate information easier and more immediate. This goal has become even more relevant with the entry of new talent, who may face a huge amount of documentation, sometimes complex and difficult to navigate, but also for experienced users who need faster and more efficient consultation.

To address this challenge, the company decided to experiment with an innovative solution based on the use of an RAG-based chatbot. The RAG system is designed to optimize accesso to information stored in the BMS, using advanced AI techniques to retrieve and generate highly relevant and personalized responses based on users' specific questions.

To prototype this solution, the Vectoria open-source library, offering powerful information retrieval and generation capabilities based on advanced LLMs, was used. The entire system was hosted on the company's DaVinci-1 supercomputer, a key resource for ensuring processing speed and system reliability. The solution was tested with a subset of the corporate documentation in order to verify its effectiveness and ability to meet user needs. The initial results of the experiment were very promising. The RAG chatbot was able to respond quickly and accurately to user queries, making it easier to access information and significantly reducing search time.

In the next section we will go more into detail on the operational architecture adopted for the implementation of the RAG solution, with a focus on the use of the Vectoria open-source library, which is the technological heart of the experiment.

All the main components of the process, from the initial parsing of the documentation to the creation of chunks, to the retrieval and reranking phase are natively managed by Vectoria, greatly simplifying the construction of the pipelines and reducing the engineering effort required to integrate external modules. The company documentation, on which the experiment was conducted, was provided in .docx format, something Vectoria is able to process natively. This enabled robust and structured parsing, thanks to the library's ability to interpret the internal hierarchy of the document (headings, paragraphs, sections), thus keeping the logical structure of the content intact. Next, Vectoria automatically segmented the text into 512-character chunks, with 256-character overlaps between segments. This approach ensures semantic continuity and improves accuracy in subsequent steps. Each chunk was also enriched with meaningful metadata, also automatically generated by the library, such as document name and source paragraph numbers, improving traceability and interaction with the results returned to the user. For the embedding phase, the BAAI/bge-m3 model, which can also be natively integrated with Vectoria, was employed. This model was chosen because of its effectiveness in semantic text representation and its excellent performance in retrieval tasks. The RAG pipeline constructed in this way involves an initial retrieval phase based on the cosine similarity between the embeddings of the query and those of the indexed chunks, followed by a reranking phase that allows the selection of the most relevant content to be refined. Reranking is also a feature directly supported in Vectoria, which in this case used the BAAI/bge-reranker-v2-m3 model to improve the sorting of returned documents according to the user query. The LLaMA-3.2 70B model, a state-of-the-art Large Language Model capable of producing consistent, accurate, and contextually relevant outputs by exploiting information retrieved from documents, was adopted for the generation of the final response. Given the size of the models involved, particularly for the generation phase, the entire pipeline was run on the company DaVinci-1 supercomputer, leveraging high-performance multi-GPU computing capabilities. This ensured competitive response times and high reliability even in the experimental phase.

This experience demonstrated how the adoption of RAG-based technologies can be a strategic resource for Leonardo S.p.A., not only to simplify the use of corporate documentation, but also to support the process of digitalization and continuous innovation within the company. The application of RAG in the context of Leonardo is a clear example of how artificial intelligence solutions can be leveraged to improve operational efficiency and optimize corporate knowledge management.

Vectoria: can I have a private chatbot in my SME?

Thanks to its flexible architecture, Vectoria can also be deployed on private company infrastructure such as workstations or local servers. This makes it an ideal solution for small and medium-sized enterprises looking for a secure, fully in-house chatbot, eliminating the need to depend on external cloud/HPC services. This configuration ensures that all operations, from document processing to response generation, remain within the company environment, preserving data confidentiality and information. In addition, the system is designed to leverage standard hardware resources and open-source software, enabling businesses to implement an advanced artificial intelligence solution without needing to invest in specialized infrastructure or software licenses.

Local installation is straightforward, and to perform it, the only requirement is a Python virtual environment in which all the necessary packages are installed. Next, the system is launched via an Ansible playbook, an open-source tool that automates system configuration, application deployment, and IT infrastructure management. Ansible simplifies the installation process by managing the technical details, performing downloads of the AI models that Vectoria needs to run, and allowing the system to start without having to manually configure each component. For remote installation, SSH access to the host server is required. Also in this situation, the only requirement is a Python environment and the use of Ansible to install and configure the system. This configuration ensures a secure and reliable installation while maintaining confidentiality of corporate data and taking full advantage of Vectoria's potential, especially when integrated with high-capacity computing infrastructure.