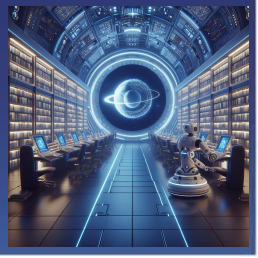
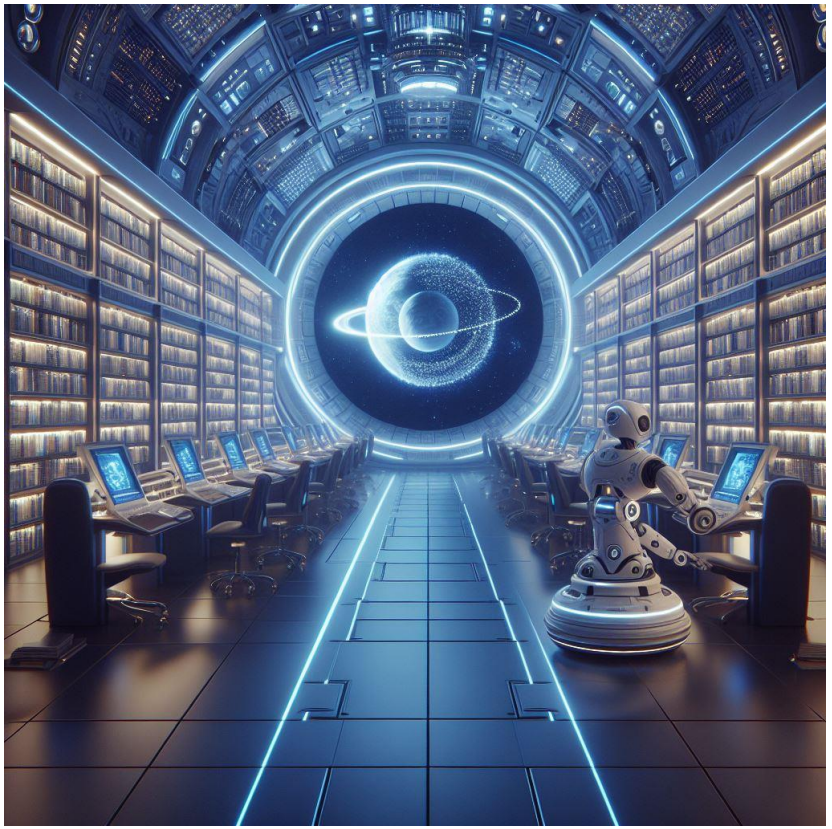


Software Verification and Generative AI

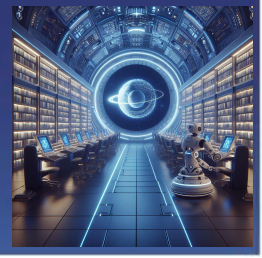


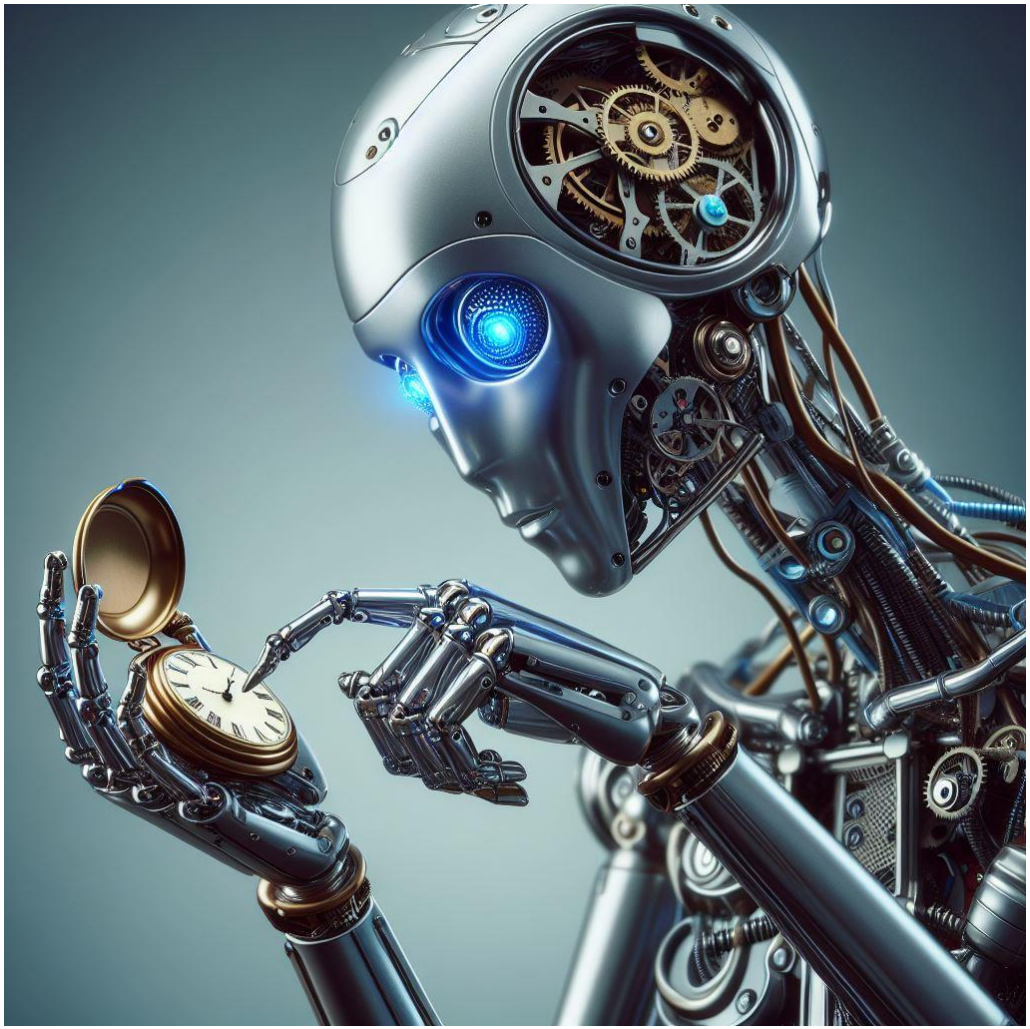
NOTE: most of the images in this presentation have been generated with Microsoft's Image Creator

Maurizio Martignano
Spazio IT – Soluzioni Informatiche s.a.s
Via Manzoni 40
46051 San Giorgio Bigarello, Mantova
https://spazioit.com

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

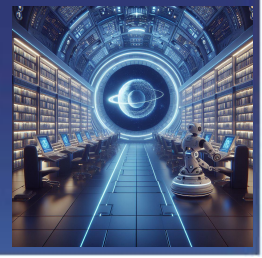
Agenda





© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

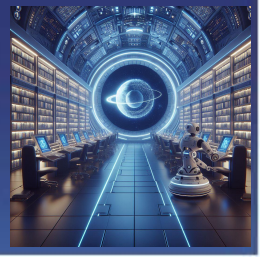
Agenda



- Generative AI
- Large Language Models
- Experiments with Language Models
- Local Language Models
- Software Verification
- Retrieval-Augmented Generation
- Future Activities

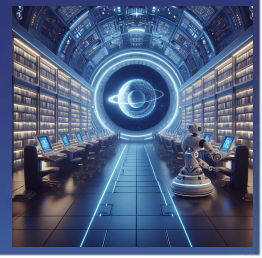
© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

Generative AI



© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

Generative AI



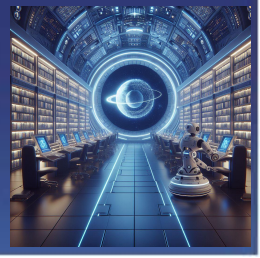
- Generative AI and Large Language Models (LLMs) are related but distinct concepts in the field of artificial intelligence.

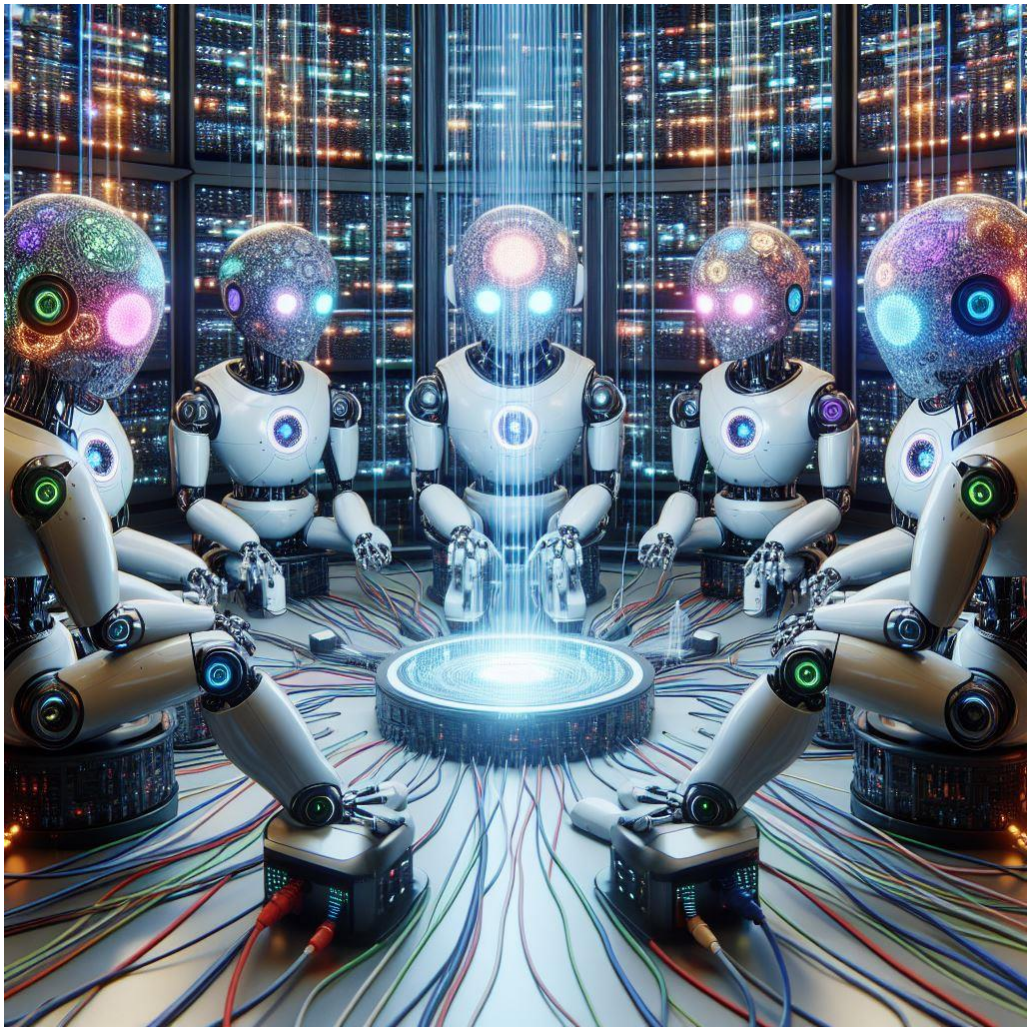
- The term **Generative AI** refers to any AI system whose primary function is to **generate content**. This could include a variety of AI models that generate different types of content, such as images, text, code, audio and video. Generative AI emphasizes the content-creating function of these systems.

- On the other hand, **Large Language Models** (LLMs) are **a specific type of AI system that works with language**. They are designed to analyze and produce text. LLMs are a form of generative AI, but they specifically deal with text-based content.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

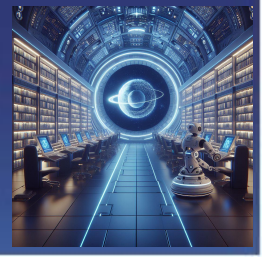
Large Language Models





© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

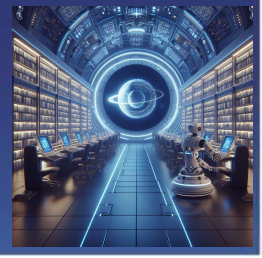
Large Language Models



- Some notable LLMs are:
  - **OpenAI's family of GPT** (Generative Pre-trained Transformer) models – both OpenAI ChatGPT and Microsoft Copilot are based on GPT;
  - **Google's PaLM**(Pathways Language Model) and Gemini;
  - **Meta's LLaMA** (Large Language Model Meta AI) open-source models.
  - The landscape of **Open-Source Language Models** is diverse, ranging from large to small in terms of parameter count, and from generic to domain-specific in their applications. Indeed, Small Language Models (SLMs) refer to versions of language models that have fewer parameters compared to larger models.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

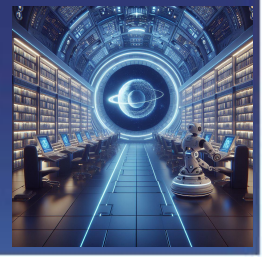
Language Models Key Parameters



- Some key parameters that characterize a language model are:
  - **Model Architecture**: The architecture of an LM determines how it processes input data and generates output. Common architectures include transformer-based models like BERT, GPT, and T5.
  - **Model Size**: The number of parameters in an LM significantly impacts its performance. Larger models tend to perform better but require more computational resources.
  - **Vocabulary Size**: The size of the vocabulary (i.e., the number of unique tokens) used by the LM affects its ability to handle diverse language.
  - **Context Window**: LMs consider a certain number of previous tokens (context) to predict the next token. The context window size influences the model's understanding of long-range dependencies.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

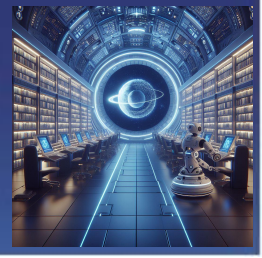
Language Models Key Parameters



- **Embedding Dimension**: The dimension of the embedding space (where the tokens live) affects how well the model captures semantic information.
- **Training Data**: The quality and quantity of training data impact the LM's performance. More diverse and relevant data lead to better generalization.
- **Pre-training Objective**: LMs are pre-trained on large corpora using objectives like masked language modeling (predicting masked tokens) or next sentence prediction.
- **Fine-Tuning**: After pre-training, LMs can be fine-tuned on specific downstream tasks (e.g., sentiment analysis, question answering) using task-specific data.
- **Hyperparameters**: Parameters like learning rate, batch size, and optimizer settings affect the training process.
- **Regularization Techniques**: Techniques like dropout, weight decay, and layer normalization help prevent overfitting

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

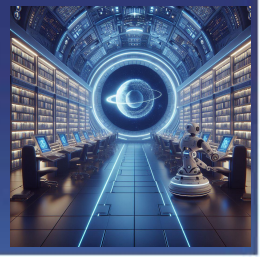
Parameters in Ollama Modelfile

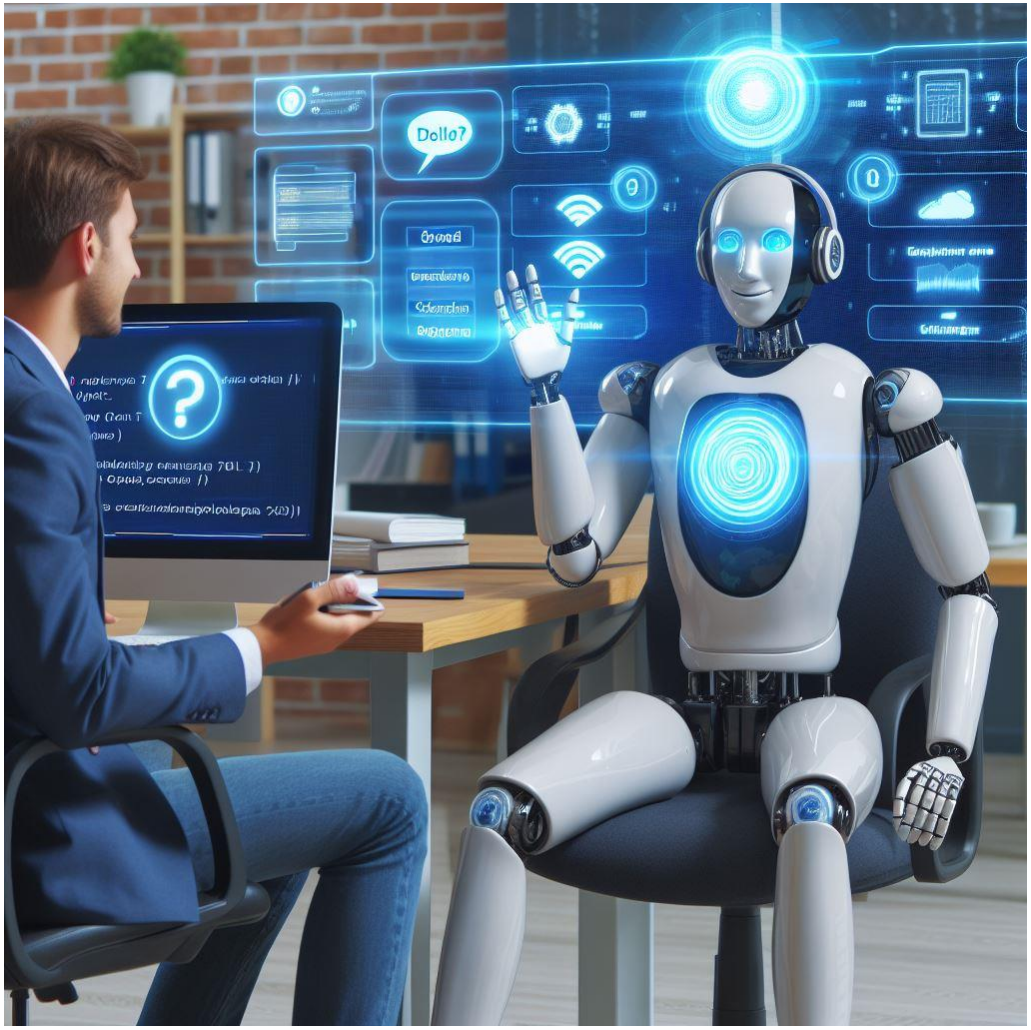


Pameter	Description	Value Type	Example Usage
mirostat	Enable Mirostat sampling for controlling perplexity. (default: 0, 0 = disabled, 1 = Mirostat, 2 = Mirostat 2.0)	int	mirostat 0
mirostat_eta	Influences how quickly the algorithm responds to feedback from the generated text. A lower learning rate will result in slower adjustments, while a higher learning rate will make the algorithm more responsive. (Default: 0.1)	float	mirostat_eta 0.1
mirostat_tau	Controls the balance between coherence and diversity of the output. A lower value will result in more focused and coherent text. (Default: 5.0)	float	mirostat_tau 5.0
num_ctx	Sets the size of the context window used to generate the next token. (Default: 2048)	int	num_ctx 4096
repeat_last_n	Sets how far back for the model to look back to prevent repetition. (Default: 64, 0 = disabled, -1 = num_ctx)	int	repeat_last_n 64
repeat_penalty	Sets how strongly to penalize repetitions. A higher value (e.g., 1.5) will penalize repetitions more strongly, while a lower value (e.g., 0.9) will be more lenient. (Default: 1.1)	float	repeat_penalty 1.1
temperature	The temperature of the model. Increasing the temperature will make the model answer more creatively. (Default: 0.8)	float	temperature 0.7
seed	Sets the random number seed to use for generation. Setting this to a specific number will make the model generate the same text for the same prompt. (Default: 0)	int	seed 42
stop	Sets the stop sequences to use. When this pattern is encountered the LLM will stop generating text and return. Multiple stop patterns may be set by specifying multiple separate stop parameters in a modelfile.	string	stop "AI assistant:"
tfs_z	Tail free sampling is used to reduce the impact of less probable tokens from the output. A higher value (e.g., 2.0) will reduce the impact more, while a value of 1.0 disables this setting. (default: 1)	float	tfs_z 1
num_predict	Maximum number of tokens to predict		

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

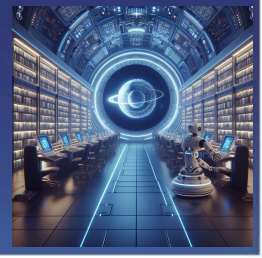
Experiments with Language Models





© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

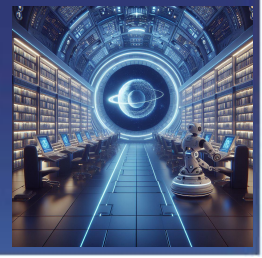
Experiments with Language Models



- The conducted experiments are reported on Spazio IT Website, at https://spazioit.com/pages_en/sol_inf_en/experiments-with-language-models/.
- These experiments put under text different models:
  - In the case of text, they compared ChatGPT with Salesforce's xgen-7b-8k-inst and Mistral AI's Mixtral-8x7b from the ; Salesforce's xgen-7b-8k-inst and Mixtral-8x7b models were executed on a local computing platform – a gaming laptop with an Intel i7-10875H CPU @ 2.30GHz CPU, 32GB of RAM and an NVIDIA GPU RTX 2080 GPU.
  - For source code, ChatGPT was compared against Mistral AI's Mixtral-8x7b-instruct and Meta's Codellama-70b-instr; these last two models were executed via the Perplexity AI GUI presumably on AWS P4d instances, which are powered by NVIDIA A100 Tensor Core GPUs.
  - All local examples of "Artificial Hallucinations" and "Artificial Illiteracy" have been produced using models available in the Ollama Models Library and executed on the same gaming laptop described above.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

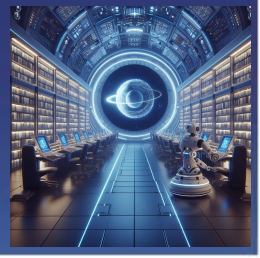
Experiments with Language Models

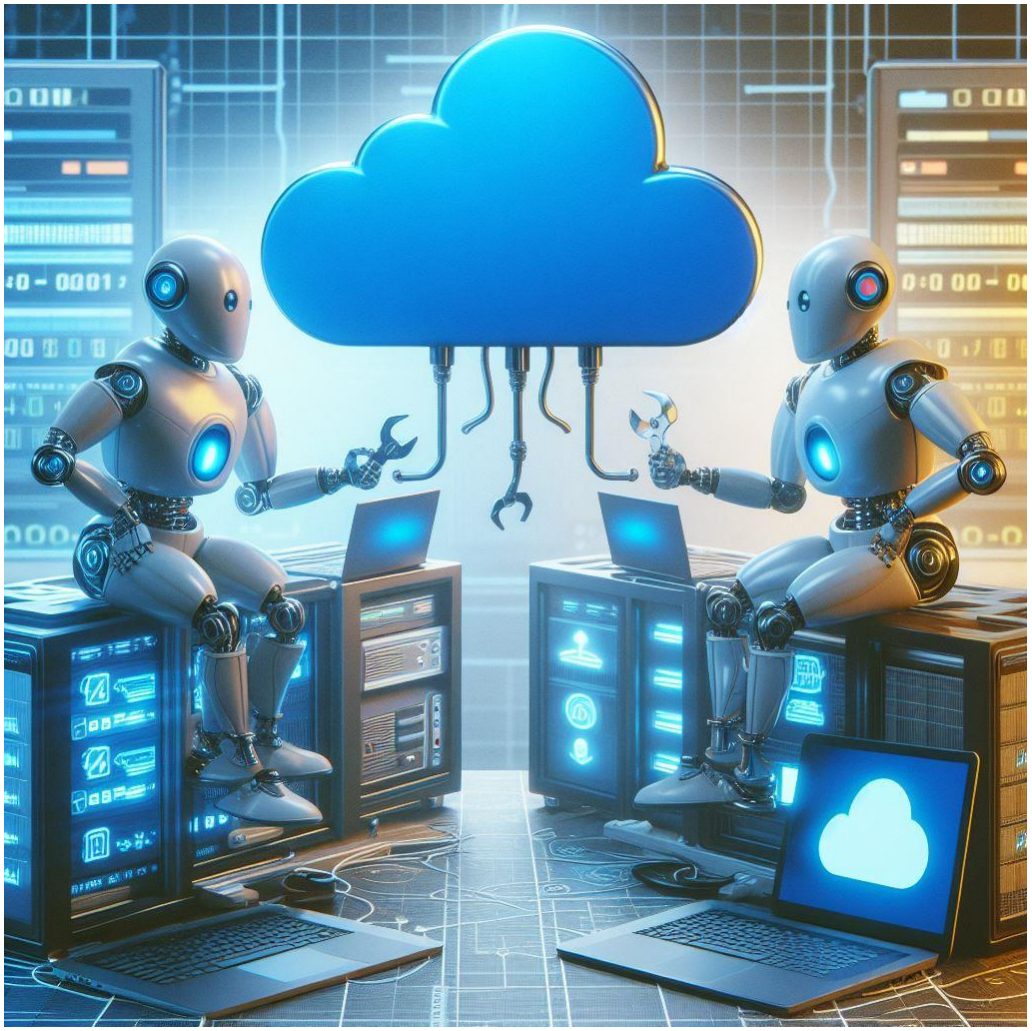


- What is common in most of these experiments is the presence in the prompt (i.e. in the query) of two main elements:
  - A **piece of text, a piece of code** to be analysed
  - **Some questions** about this piece of text, piece of of code
- Creating a **"good" prompt**, a query with a proper association between these two elements is very, **very important**.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

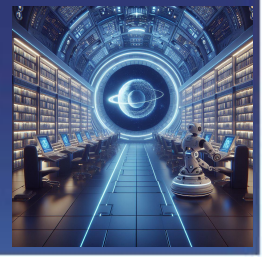
Local Language Models





© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

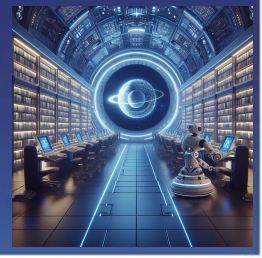
Local Language Models



- When users utilize generative AI systems like OpenAI's ChatGPT, Google's Gemini, or others provided by major tech companies for free, **there's no assurance** that the data they input, i.e. the Content, specifically **the information within the prompts, remains confidential**.
- Certainly, **it is always feasible to set up** commercial registrations or **contractual agreements** that prohibit the service provider from retaining and exploiting user content, **but these are merely business assurances**.
- **The sole method for users to ensure their content isn't misused is by exclusively utilizing open-source models** (with a thorough understanding of their functions) and **operating these models on a private and local computing platform**.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

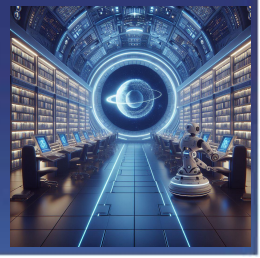
Local Language Models



- Large language models demand huge computational resource.
- To be able to run these models on local and affordable computing platforms a technique, called “**quantization**” is used. Quantization refers to the process of reducing the precision of model parameters, typically by converting floating-point numbers (which have decimal precision) into integers with a smaller bit-width representation.
- While quantization introduces some loss of information, **dithering**, that is adding small amounts of random noise to the data before quantization, can help mitigate this loss by spreading the quantization error.
- In addition to streamlining or "downsizing" the model, another method of conserving resources is to utilize **efficient software platforms or frameworks**, such as "ggml".

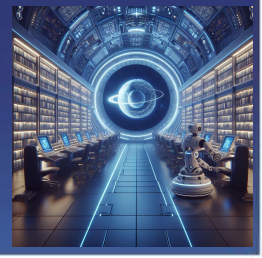
© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

Software Verification



© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

Software Verification

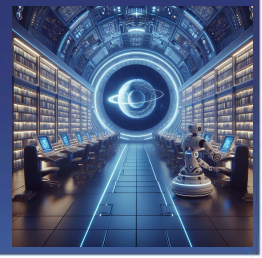


- Software Verification entails reading and analyzing big if not huge documents sets and code bases.

- Both the documents and the sources can be described as list of «things», list of «objects», e.g.:
  - A System Specification (RS) is a list of System Requirements
  - a Software Specification is a list of of Software Requirements
  - a DDF is a list of Architectural Components
  - a code base is a list of compilation units
  - each compilation units is a list of classes, procedures and functions (of course the actual names of these «things» depend on the programming language).

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

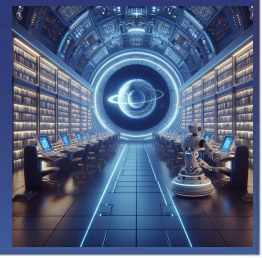
Software Verification



- All «things» in a list are identified by some kind of «ID»
- The lists are usually at different levels of abstraction, e.g. a list of specifications implements a list of requirements.
- Traceability information, binds together, establishes a relationship between «things» usually belonging to lists at different level of abstraction, e.g.:
  requirements → design → code
- Automatic traceability checks have been working (up to now) only at «ID» level.
- Generative AI opens new opportunities for a **semantic match**.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

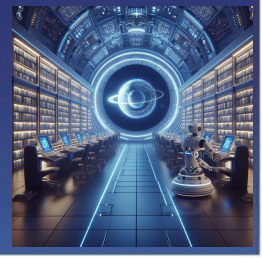
Software Verification – Typical Prompts



- Prompt 1:
  - Piece of text / piece of code
  - Is it correct or does it contain any error or contradiction?
- Prompt 2
  - Piece of text #1
  - Piece of text #2 (following the traceability relationship)
  - Are they consistent? Does piece #2 implement piece #1?
- Prompt 3
  - Piece of text
  - Piece of code (following the traceability relationship)
  - Are they consistent? Does piece of code implement piece of text?

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

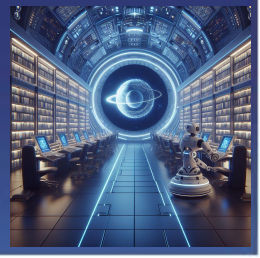
Software Verification – Typical Prompts – But...

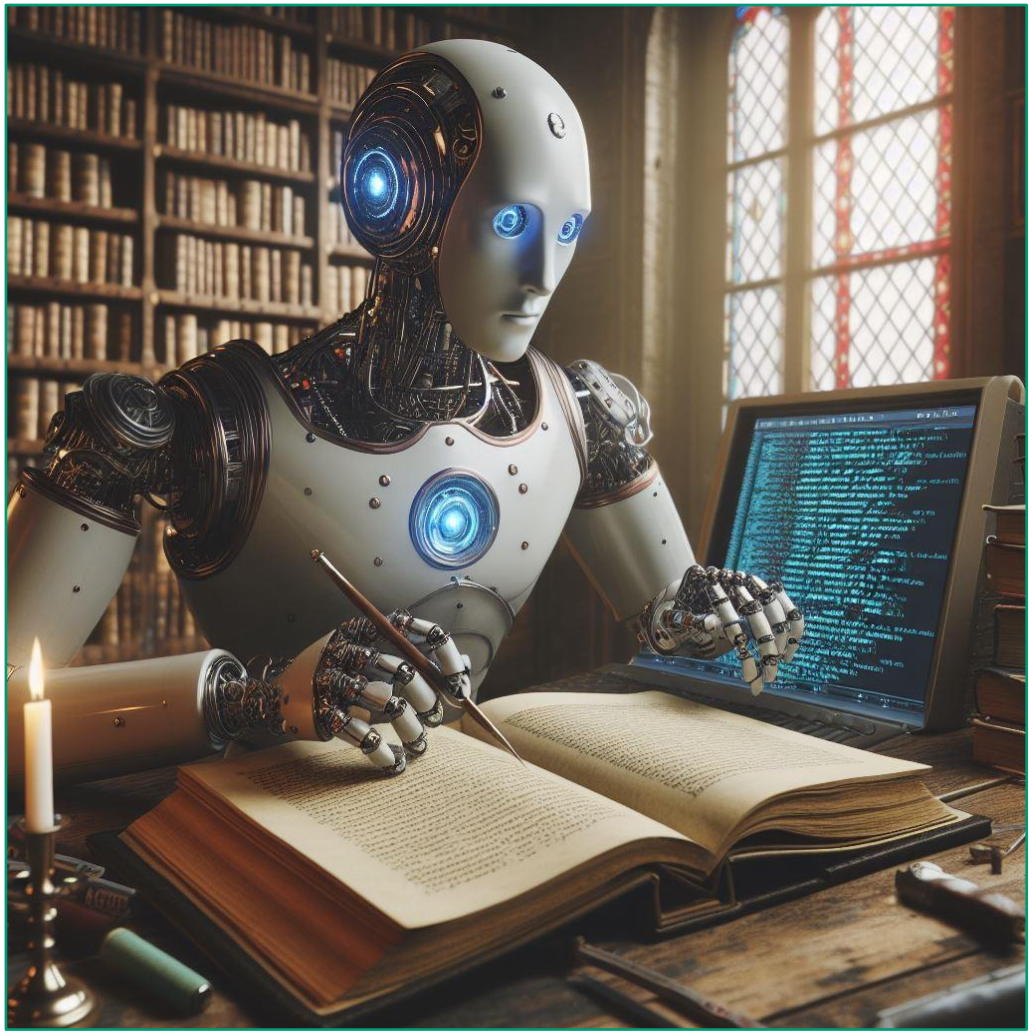


- How to make the model «digest» huge documents sets and code bases?
- How to build the correct prompts?
- Enter Retrieval-Augmented Generation and Spazio IT work on RAG.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

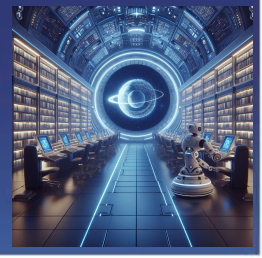
Retrieval-Augmented Generation





© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

What is Retrieval-Augmented Generation?



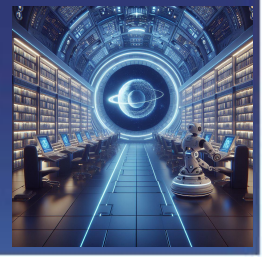
- “Retrieval-Augmented Generation (RAG) is the process of optimizing the output of a large language model, so it references an authoritative knowledge base outside of its training data sources before generating a response. Large Language Models (LLMs) are trained on vast volumes of data and use billions of parameters to generate original output for tasks like answering questions, translating languages, and completing sentences.

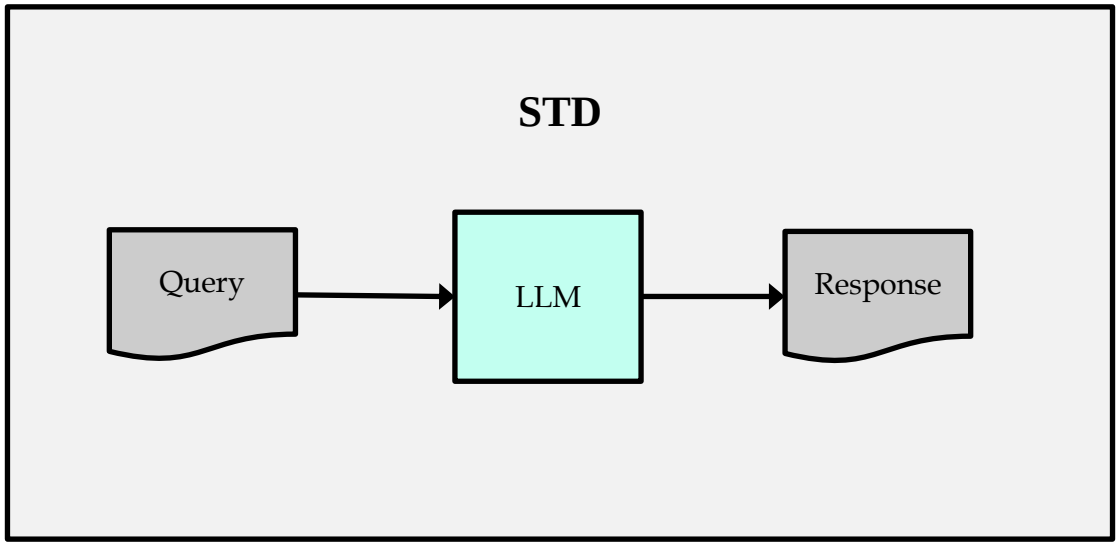
- RAG extends the already powerful capabilities of LLMs to specific domains or an organization's internal knowledge base, all without the need to retrain the model. It is a cost-effective approach to improving LLM output, so it remains relevant, accurate, and useful in various contexts.”

[https://aws.amazon.com/what-is/retrieval-augmented-generation]

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

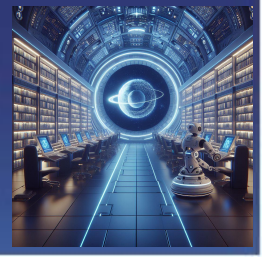
Retrieval-Augmented Generation: Experiments / Directions

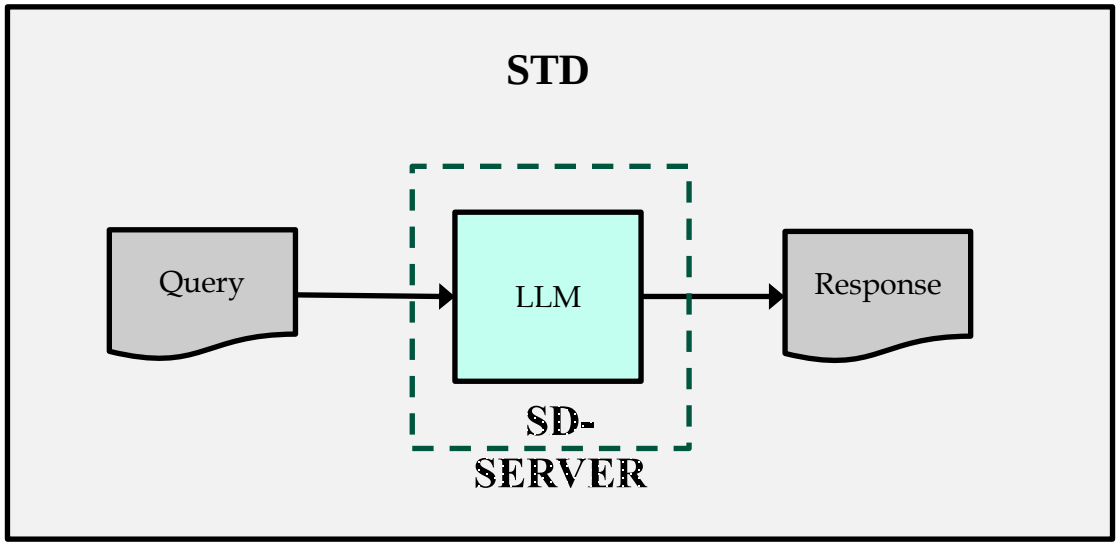




© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

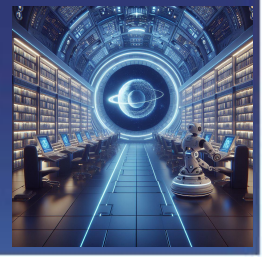
Retrieval-Augmented Generation: Experiments / Directions

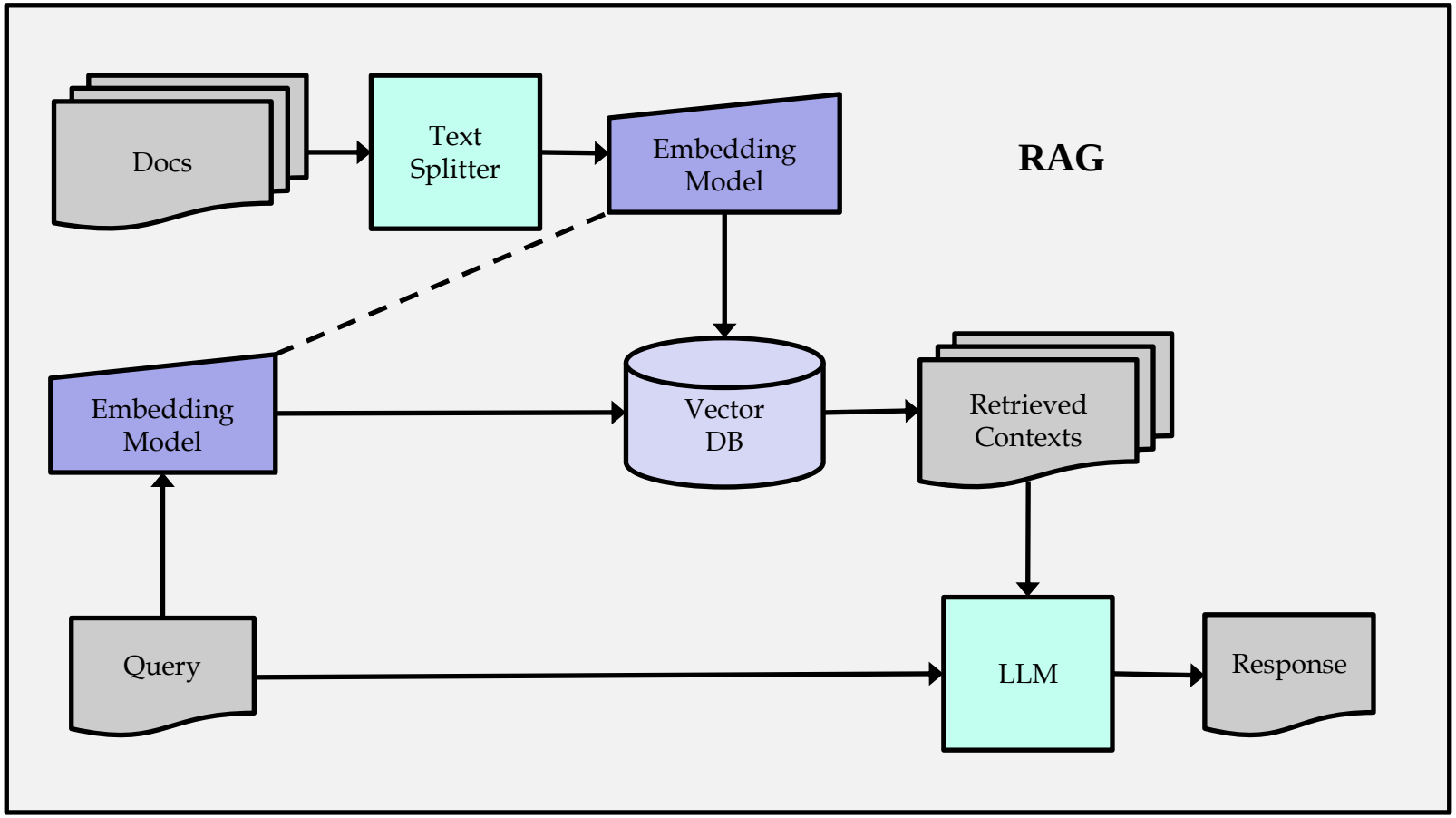




© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

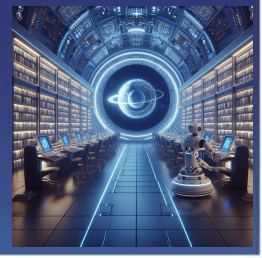
Retrieval-Augmented Generation: Experiments / Directions

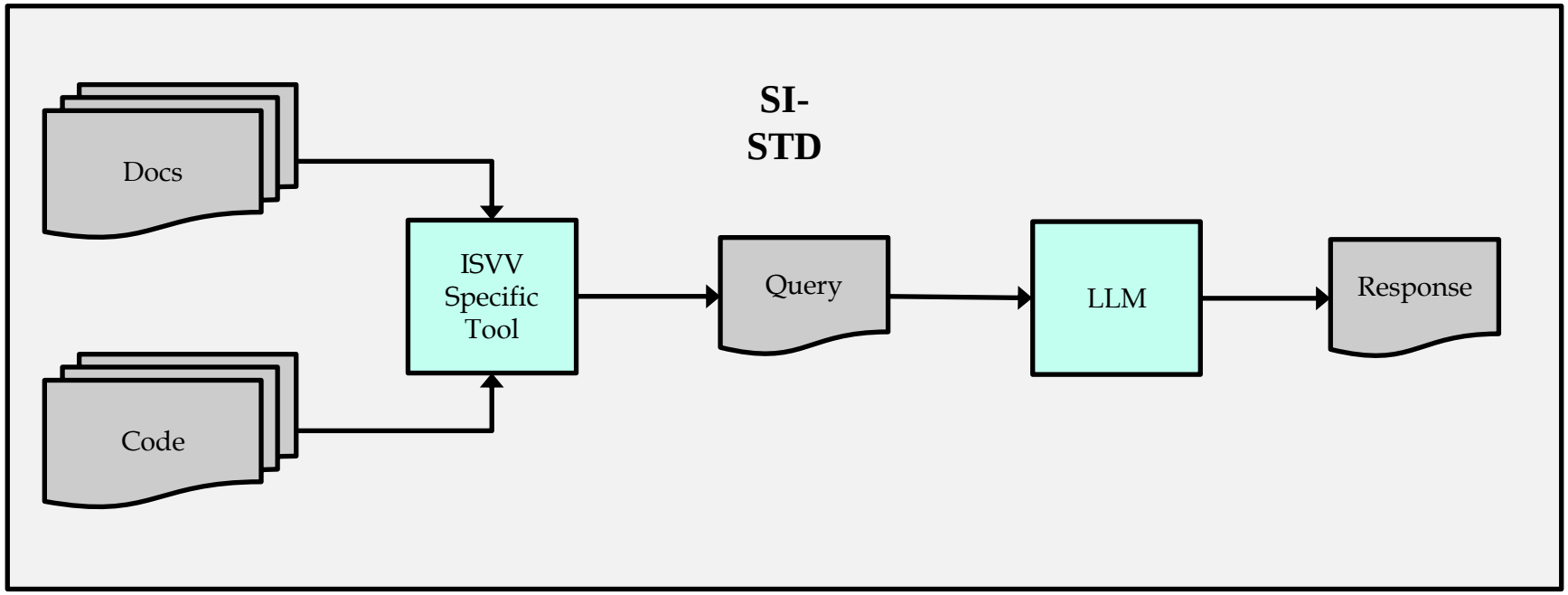




© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

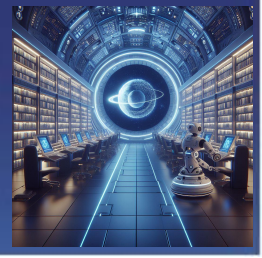
Retrieval-Augmented Generation: Experiments / Directions

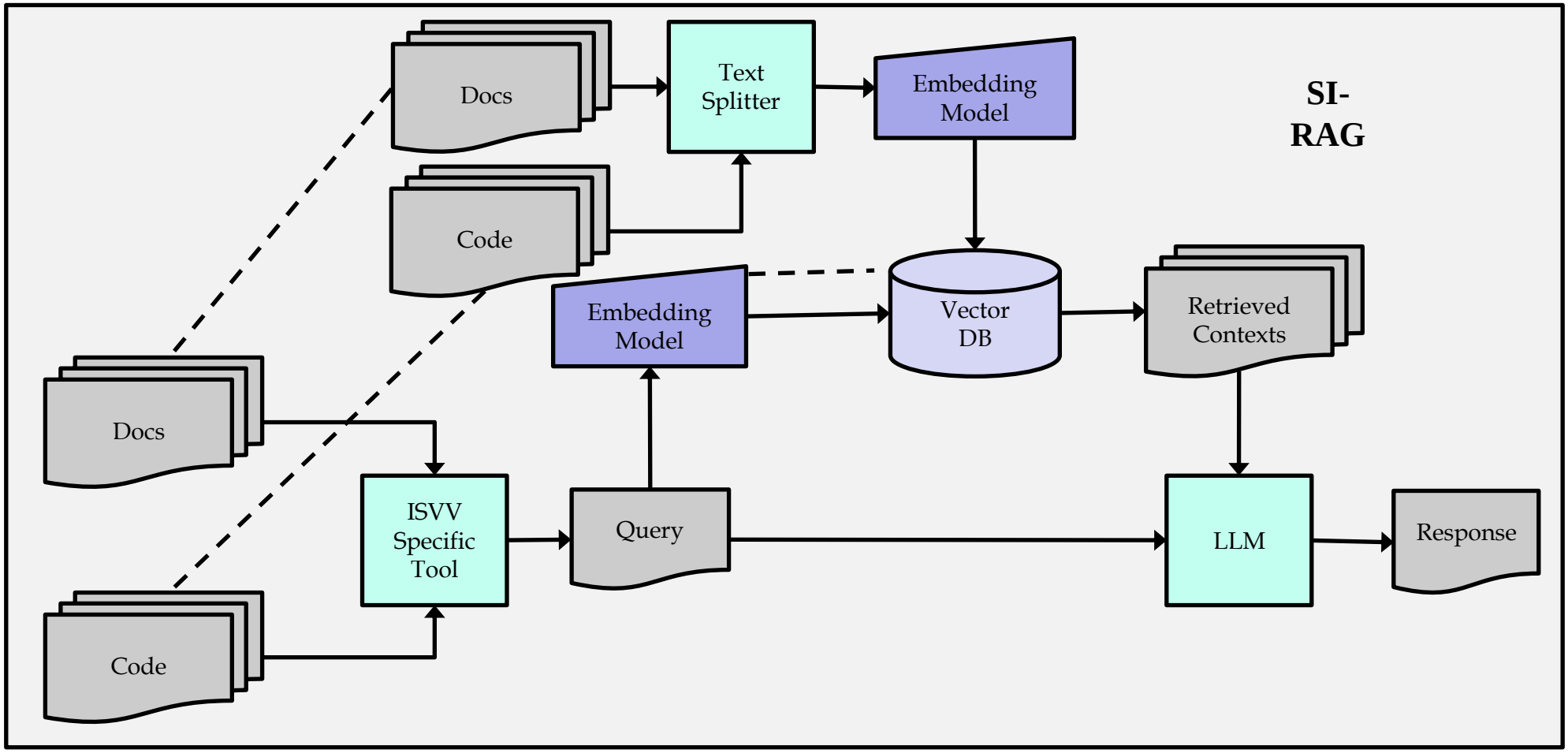




© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

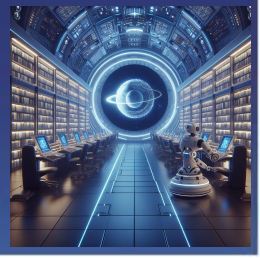
Retrieval-Augmented Generation: Experiments / Directions

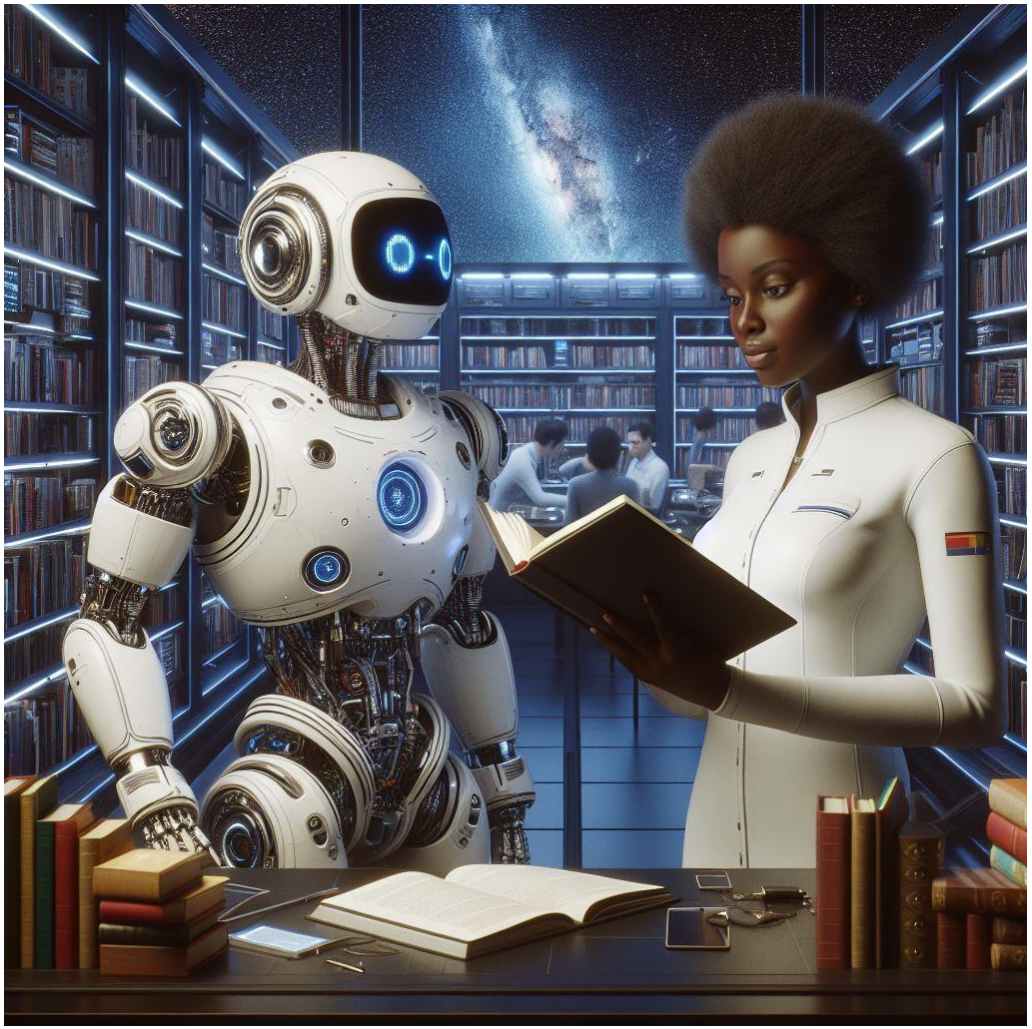




© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

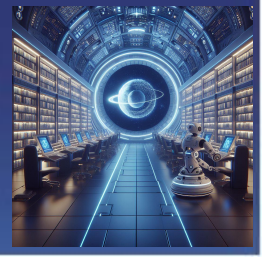
Future/Current Activities





© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

Future/Current Activities

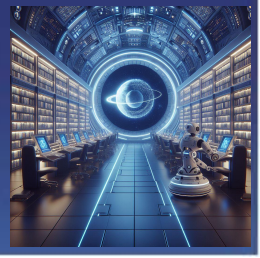


- Work on RAG:
  - Experimenting with models on different actual ISVV projects.
  - Develop a sensible RAG platform (probably a matter of putting together the proper open-source models and software components).

- Integrate this RAG Platform within the SAFe Toolset https://spazioit.com/pages_en/sol_inf_en/code_quality_en/safe-toolset-en/.

© 2024 Spazio IT - Soluzioni Informatiche s.a.s.

Thank you for your time!



© 2024 Spazio IT - Soluzioni Informatiche s.a.s.