**Extraction Approach Description**

# The description provided below will be used to evaluate the approach developed by your team to automatically extract financial data of MNE Groups. This description will be evaluated by the Evaluation panel based on the criteria described in the Evaluation tab of the Extraction Challenge and used for the ranking of your team for the Reusability and Innovativeness Awards.

# **Methodology *(Algorithm reusability and scalability, Data-driven approaches, Availability, quality and maintainability of documentation)***

Please provide a detailed description of the methodology used to develop algorithm-based approaches which automatically extract financial data of MNE Groups. The description should contain (1) the data processing steps, (2) the methods and models used, (3) references to the scientific papers/sources that present the methods and models used, and (4) the time it took to process the data set and extract the relevant financial data.

Bear in mind that the workflow will be also evaluated based on the criteria for the Reusability and Innovativeness Awards.

*This section will be evaluated for:*

*(1) the Algorithm reusability and scalability criterion: likeliness that the described approach can successfully reproduce the solution submitted by the team for the Accuracy award.*

*(2) the Data-driven approaches criterion: evaluated based on whether it is data-driven rather than heuristic. More data-driven approaches will receive higher scores.*

*(3) the Availability, quality and maintainability of documentation criterion: assessed based on the maintainability of the code.*

|  |
| --- |
| **Methodology for Automatic Identification of Public Web Sources of Annual Financial Data of MNE Groups**  Our solution employs a data-driven methodology that combines traditional web scraping with the capabilities of a large language models to smartly search for and refine results of financial data.  The core of the system is an iterative workflow that aims to identify the most recent and specific public web sources of annual financial data for a predefined list of MNE Groups.   1. **Data Processing Steps**   For each MNE Group in the input listand for each variable to extrapolate, the process follows a structured workflow to ensure comprehensive and accurate data collection:  A tailored **prompt** is generated using the **PromptGenerator**, incorporating any known details about the company to guide subsequent searches.  The **WebScraperModule** conducts an initial sweep across the web to gather foundational data. Key tasks include:   * + Identifying the company’s **official website**.   + Extracting relevant links to financial disclosures and other pertinent documents.   + Extract the value for the target variable of reference   The **core AI module** processes the collected data, leveraging advanced LLMs to refine searches and extract insights.   * A **Multi-Agent System** optimizes performance by:   + Distributing tasks for efficiency.   + Ensuring **quota compliance** (e.g., API rate limits, data retrieval constraints). * Supported LLMs:   + gemini-1.0-flash   + gemini-2.0-flash   + gemini-1.5-flash-8b   Error handling and user-agent rotation are built-in to increase reliability and mitigate blocking.  Once the initial data is collected, the system enters an iterative loop capped at a maximum number of iterations for search and validation. During each iteration, the FinancialSourceFinder queries the Google Gemini model with the prompt specialized for the company. The model returns a JSON response containing a predicted URL and a corresponding fiscal year.  The system performs an **API request** to verify the URL’s accessibility:   * + **If successful**: The result is **validated**, and the loop exits   + **If failed**: The system **retries** with adjusted.   If the system exhausts its maximum iterations without validating a URL, it triggers an **alternative AI-driven approach**:  The **AI module** (Gemini) is prompted to generate a **Python script** tailored to:   * Scrape financial data **directly from the company’s website** (e.g., Investor Relations pages). * Handle dynamic content (e.g., JavaScript-rendered tables) via libraries like BeautifulSoup * Get into the target variable of the company   The script includes:   * Target URL(s) or navigation logic. * Data extraction rules (e.g., XPaths/CSS selectors for tables). * Error handling (e.g., timeouts, missing element   The Output is validated via an API request to ensure correctness:   * Success: Data is stored; the process concludes. * Failure: The system retries.   The fallback process exits after Successful validation or reaching a retry limit.  Should multiple attempts fail to produce a validated result, the system is designed to switch strategies. One future enhancement includes leveraging Gemini to generate a custom web scraping script tailored for the company’s website. Another fallback option could involve employing a simpler, heuristic-based search.  In cases where more than one valid URL is discovered, the system is intended to generalize accordingly by capturing and storing all relevant links.  Once a satisfactory result is obtained or the iteration limit is reached the final outcome is stored. For each company, this includes the selected URL, the associated fiscal year, the validation status (where applicable), any feedback from the refinement process, and contextual data from the initial scraping phase. These outputs are consolidated into a CSV file.  **(2) Methods and Models Used**  The foundation of our solution lies in a blend of web scraping and large language model technologies. For scraping, we rely on standard Python libraries like requests and BeautifulSoup, which allow us to navigate web structures and extract relevant content. These tools are used to identify key areas of a website that may contain financial information, such as specific links or downloadable reports.  The intelligence layer of our system is powered by the Google Gemini models. This LLM is responsible for retrieving URLs and fiscal years based on prompt input, and for suggesting prompt improvements in response to validation feedback. It may also be used in the future to generate scraping scripts for particularly difficult websites.  Prompt engineering plays a central role in this methodology. The PromptGenerator builds initial prompts and refines them as needed. Prompts are crafted to be clear, informative, and structured, ensuring that the LLM understands both the task and the format of the desired output.  To enhance efficiency, our system includes parallel processing capabilities via the ThreadPoolExecutor. This enables the concurrent processing of multiple MNE Groups, dramatically reducing the overall runtime compared to sequential execution.  **(3) References to Scientific Papers/Sources**  Our methodology is grounded in established best practices from web scraping and current research in large language models. For web scraping, we reference the official documentation of the libraries used—such as [requests](https://requests.readthedocs.io/en/latest/) and [BeautifulSoup](https://www.crummy.com/software/BeautifulSoup/bs4/doc/)—which provide comprehensive guides and examples.  Regarding the use of LLMs, the capabilities of Google Gemini are documented in technical publications and research announcements found at [Google AI Research](https://ai.google.com/research/). The specific use of LLMs for tasks like prompt optimization and document retrieval is informed by research in the areas of question answering and information extraction.  Prompt engineering, a key aspect of our approach, draws inspiration from evolving resources like OpenAI’s [prompt engineering guide](https://platform.openai.com/docs/guides/prompt-engineering), which outlines strategies for eliciting accurate and relevant responses from LLMs.  **(4) Time to Process the Data Set**  The total time required to process a given dataset of MNE Groups depends on multiple factors. These include the number of companies in the input list, the speed and structure of the target websites, the number of iterations needed per company, the latency of Gemini API responses, and the level of parallelism achieved during execution. An estimation of Company-wise processing time is about 200s to generate 5 ULRs. However, these results depend on quota and AI responses.  **Evaluation Criteria Alignment**  **Data-driven approaches:** Our methodology is fundamentally data-driven. It uses real-time web scraping and an LLM trained on massive datasets to perform intelligent searches and adapt to feedback. The iterative refinement of prompts based on validation insights embodies a learning process rooted in data, enhancing both precision and recall. The code’s modular structure, with clear separation between scraping, prompting, validation, and orchestration, ensures transparency and traceability in the data pipeline.  **Availability and quality of documentation:**  The code is well-documented and easy to reproduce. The Contributing.md file explain how to set-up the repo while the README\_SUBMISSION explain the core functions. |

## **Architecture *(Architecture)***

Please provide a description of the architecture of your approach. A diagram of the architecture is considered of additional value. Indicate what modifications would be required to apply the approach to similar datasets on a larger scale.

*This section will be evaluated for:*

1. *the Architecture criterion: evaluated based on its modules, their cohesion and their configurability; an architecture which is modular and includes clear connections between modules or components receives a higher score.*

|  |
| --- |
| **Description of the Architecture:**  The architecture revolves around the FinancialSourceFinder class, which acts as the orchestrator, managing the interaction between the other key modules:   * **WebScraperModule:** This module is responsible for all web interaction, including finding official company websites targeting the variable of interest, navigating to investor relations pages, extracting links to financial report. It encapsulates the logic for handling HTTP requests, parsing HTML content, and managing potential issues like connection errors. * **PromptGenerator:** This module focuses on creating and refining the search queries (prompts) sent to the AI model. This separation allows for independent modification of the prompting strategies without affecting the scraping or validation logic. * **FinancialSourceFinder:** This central module manages the overall workflow for each company. It loads the list of companies, initializes the other modules, coordinates the iterative search-validation-optimization loop, and saves the final results. Its design promotes a clear separation of concerns and facilitates the management of the entire process.   **Robustness at Larger Scale:** The modular design enhances the robustness of the approach. If a specific web scraping function or AI interaction encounters an issue for a particular company, it is less likely to impact the processing of other companies due to the encapsulated nature of the modules. The retry mechanisms within the WebScraperModule also contribute to robustness by handling transient network issues. However, at a very large scale, we might need to implement more sophisticated error handling and potentially queueing mechanisms to manage API rate limits and ensure that the process can gracefully recover from failures.  **Performance Stability with Increased Scale:** The performance of the approach is expected to remain relatively stable as the scale increases, primarily due to the parallel processing capabilities. The processing time for each individual company should remain consistent, assuming the complexity of their websites and the availability of financial data do not drastically change. However, the overall throughput (number of companies processed per unit time) will scale with the number of parallel workers. Potential bottlenecks at a very large scale could arise from API rate limits of the AI models or the target websites. Implementing strategies like intelligent rate limiting and distributed processing would be crucial to maintain stable performance. |

# **Hardware Specifications *(Algorithm reusability and scalability)***

Please describe the hardware specifications of the machines that were used to run the methodology.

*This section will be evaluated for:*

*(1) the Algorithm reusability and scalability criterion*

**Machine 1**

|  |  |
| --- | --- |
| CPUs | i386 (8 physical, 8 logical) |
| GPUs | Not used |
| TPUs | Not used |
| Disk space | 228 GiB |

**Machine 2**

|  |  |
| --- | --- |
| CPUs | CPU name and capacity |
| GPUs | GPU name and capacity |
| TPUs | TPU name and capacity |
| Disk space | The space required to calculate and store the data |

# **Libraries *(Availability, quality and maintainability of documentation)***

Please provide the libraries used for approach, if any, as well as the links to these libraries, if available.

*This section will be evaluated for:*

*(1) the Availability, quality and maintainability of documentation criterion: The use of libraries which are regularly maintained will yield higher scores. (Examples include pytorch, tensorflow, scikit-learn, pandas, numpy, tidyverse, etc.).*

|  |
| --- |
| **"google-generativeai",**  **"pandas",**  **"python-dotenv",**  **"futures",**  **"pyyaml",**  **"beautifulsoup4",**  **"retry"** |

## **Similarities/differences to State-of-the-Art techniques *(Originality of the approach)***

Please provide a list of similarities and differences between the used methodology and to the state-of-the-art techniques.

*This section will be evaluated for:*

*(1) the Originality of the approach criterion: compare the approach used to the state-of-the-art, i.e. currently published approaches that are closest to the approach applied for the submission, and the extent to which the submission represents an improvement over these approaches.*

|  |
| --- |
| Our approach to automatically finding public annual financial data for multinational companies blends the familiar techniques of web scraping and natural language processing with a novel, intelligent twist. Like many current systems, we use web scraping to navigate the internet and NLP, powered by Google Gemini, to understand and extract information. However, where we diverge is in how we orchestrate these tools.  Imagine our system as an intelligent agent that doesn't just passively search but actively learns and refines its strategy. We don't just scrape and then analyze; we use Gemini's reasoning to guide the very search process. It's an iterative loop: Gemini suggests a potential source based on our prompt.  This "AI talking to AI" to refine the search is a key element of our originality.  Furthermore, we're not just looking for any financial data; we're specifically aiming for the direct URL to the most authoritative source, like the annual report PDF itself. And when direct links are elusive, we're exploring the innovative idea of having Gemini actually write small programs – web scraping scripts – on the fly to extract the data from more complex websites. This dynamic, AI-driven script generation could be a significant leap beyond relying on pre-set rules or manual coding for tricky sites.  Essentially, we're putting the intelligence of a powerful language model at the heart of the entire process, from the initial search to the potential data extraction. This tight integration and the AI-driven refinement loop, along with the concept of AI-generated scrapers, are what set our methodology apart and offer the potential for a more adaptable, accurate, and automated way to find crucial financial information online. We believe this represents a step forward in how AI can be used for targeted and efficient web data extraction generalizing the search for a variable of interest |

## **Contribution to scientific field *(Future orientation)***

Please describe how your submission contributed to the scientific field, what impact it could have and what could potentially be future work to improve the solution.

*This section will be evaluated for:*

*(1) the Future orientation and impact criterion: the potential effect of the approach used will be evaluated; this includes the scale of impact it has on the problem of extracting financial information from the Internet; the impact will be evaluated based on potential efficiency improvements and cost reductions.*

|  |
| --- |
| Our submission contributes to the scientific field at the intersection of **information retrieval**, **natural language processing (NLP)**, and **financial data extraction**. Specifically, it explores a novel approach to automating the identification of authoritative financial data sources on the web by deeply integrating the reasoning and generative capabilities of Large Language Models (LLMs) into the search and refinement process.  **Contribution and Potential Impact:**   1. **Advancing AI-Driven Web Information Extraction:** Our work demonstrates a paradigm shift in how LLMs can be used for targeted information retrieval. Instead of solely relying on keyword-based searches or post-processing of scraped data, we showcase the potential of LLMs to actively guide the search strategy through dynamic prompt generation and optimization. This could inspire further research into leveraging the semantic understanding of LLMs for more intelligent and efficient web data extraction across various domains. 2. **Pioneering AI-Generated Scraping Strategies:** The idea of using an LLM to dynamically generate web scraping scripts based on the challenges encountered during direct URL identification opens a new avenue for automated data extraction from complex websites. This could significantly reduce the manual effort involved in developing and maintaining website-specific scrapers, a common bottleneck in web data acquisition. 3. **Potential for Efficiency Improvements and Cost Reductions:** If successfully implemented and scaled, our approach has the potential to significantly improve the efficiency of financial data extraction. Automating the identification of direct, authoritative sources reduces the time and resources spent on manual searching, data aggregation from potentially less reliable sources, and the development of extensive manual scraping rules. This could lead to substantial cost reductions for financial analysts, researchers, and regulatory bodies that rely on timely and accurate financial data. 4. **Scalability and Generalizability:** The core principles of our AI-driven search and refinement process could potentially be generalized to discover other types of specific information on the web beyond financial data. The modular design of our system also promotes scalability by allowing for the integration of more sophisticated LLMs and web scraping techniques as they evolve. |

## **Lessons Learned *(Future orientation)***

Please state any lessons learned during the competition.

*This section will be evaluated for:*

*(1) the Future orientation and impact criterion: what were the lessons learnt during the competition, and what could potentially be future work to improve the solution.*

|  |
| --- |
| The competition has illuminated several key insights that will significantly shape our future development efforts. Firstly, we've experienced firsthand the remarkable power of Large Language Models in intelligent information retrieval. However, this power comes with complexity; effectively guiding these models to achieve precise tasks with high accuracy demands iterative experimentation and meticulous prompt engineering. Secondly, the critical need for robust and automated validation has become strikingly apparent. Without a reliable way to verify the results, even the most advanced search algorithms can falter. Therefore, developing an effective AI-driven validation mechanism is a top priority for our next steps.  Furthermore, we've recognized the close relationship between source extraction and the potential need for data extraction. Anticipating the challenges of extracting data during the initial search phase, such as identifying promising pages requiring complex scraping techniques, is crucial for creating a comprehensive solution. Our modular design has proven to be a significant advantage, enabling focused development and experimentation on individual components like the prompt generator or the web scraper without disrupting the entire system.  The development of an effective AI-driven solution is inherently iterative. Our initial concepts will undoubtedly require refinement through ongoing experimentation and rigorous evaluation. Embracing this adaptive approach and being prepared to pivot based on our findings is essential for progress. Finally, we've underscored the significance of anticipating and planning for edge cases and potential failures in the data extraction process, such as encountering inaccessible websites or ambiguous information. Implementing robust error handling and fallback mechanisms will be crucial for building a truly practical and reliable system. These invaluable lessons learned during the competition will serve as a compass, guiding our efforts as we continue to refine our approach and strive to create a more robust, efficient, and accurate system for the automated extraction of financial data. This challenge has provided an invaluable platform for understanding the intricacies and opportunities within this exciting field. |

# **Short description of the Team – area of expertise**

Please provide a description of the team, your area of expertise and contact information.

|  |
| --- |
| **Paolo Barba: Data Scientist,** [**paolobarba50@gmail.com**](mailto:paolobarba50@gmail.com)**., cell. +39 3451600069**  **Raffaele Liguori: Data Engineer,** [**rafflig8@gmail.com**](mailto:rafflig8@gmail.com) **,cell. +39 366 878 9163**  **Francesco Cellitti: Data Analyst,** [**francellitti@gmail.com**](mailto:francellitti@gmail.com) **, cell. +39 338 107 3582**  **GianmarioSangiovanni:phdCandidateDataAnalysis,** [**gianmario.sangiovanni@uniroma1.it**](mailto:gianmario.sangiovanni@uniroma1.it)**, cell.** **+39 377 548 2859** |