Toward a national repository of administrative procedures: web crawling and process modeling with large language models*

Francesca De Luzi[1,†], Mattia Macrì[1] and Massimo Mecella[1]

[1]Department of Computer, Control and Management Engineering, Sapienza Università di Roma, Via Ariosto 25, Rome, Italy

Abstract

This paper presents a scalable approach to the identification, extraction, and modeling of administrative procedures within the Italian public administrative sector. Despite ongoing reform efforts, Italy lacks an official, comprehensive catalogue of such procedures—an absence that hinders administrative simplification and digital transformation. To address this gap, we propose *(i)* CRAWL4PA, a nationwide web crawling targeting the public administration websites, where procedural documentation is legally mandated to be published, and *(ii)* the development of ADMPMODELER, a pipeline that uses Large Language Models and prompt engineering techniques to convert legal texts into structured, human-readable process models. This work lays the foundation for a national repository of administrative procedures—that supports transparency and accountability, but also enable advanced policy experimentation, simplification initiatives, and automated compliance verification.

Keywords

E-Government, Administrative Procedures, Large Language Models, Prompt Engineering

1. Introduction

Public administrations operate through well-defined, regulated procedures. However, due to the wide variety of administrative bodies present in a highly bureaucratized country like Italy, the exact number of such procedures is unknown. Moreover, most administrative procedures concerning individuals or private companies are managed at the regional, provincial, metropolitan, or local level, rather than by national offices. The decentralization of procedure management implies that procedures can be handled differently, for example, from one region to another. Consequently, for each procedure, some administrative bodies may manage it more efficiently than others, leading to inefficiencies and loss of know-how. Therefore, in addition to not knowing the exact number of Italian administrative procedures, it is also unknown, for each of them, what variations are applied by each administrative entity.

In this context, while current reform efforts under the PNRR (Italy's Recovery and Resilience Plan) claim a target of simplifying at least 600 procedures, interviews with domain experts reveal a surprising truth: there is no official or comprehensive catalogue of these procedures and their implementation variants. This ambiguity poses a fundamental challenge that is a necessary prerequisite for achieving the simplification goal of at least 600 procedures, namely, an exhaustive mapping of the procedures themselves and of how they are implemented in practice. Without a concrete baseline, it is impossible to measure progress or even define the scope of reform. It is essential to develop a method capable of automatically identinfy, extracting, and structuring administrative procedures from heterogeneous sources. To address this challenge, we are developing a two-step AI-based system that combines large-scale web crawling with automated process modeling, laying the foundation for building an exhaustive, structured catalogue of Italian administrative processes. The authors believe that, to properly design and develop AI-based systems, it is essential to integrate both the well-established principles of design

Artificial Intelligence for Digital Public Services, June 11, 2025, University of Koblenz

*Corresponding author.

†These authors contributed equally.

deluzi@diag.uniroma1.it (F. D. Luzi); macri@diag.uniroma1.it (M. Macrì); mecella@diag.uniroma1.it (M. Mecella)
http://conceptbase.sourceforge.net/mjf/ (M. Mecella)
0000-0002-9896-2528 (F. D. Luzi); 0009-0004-7679-6153 (M. Macrì); 0000-0002-9730-8882 (M. Mecella)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-12-10

science research [1], and ad-hoc design approaches that reflect the rapid advancements in generative AI [2]. These more recent perspectives highlight the importance of modular, evolvable system architectures [3], and emphasize the need for extensive human evaluation to effectively inject domain expertise into the assessment of AI-generated outputs [4]. In the continuation of this ongoing research, we are committed to embracing these principles as guiding criteria for the future development of our systems. This paper is organized as follows. In Section 2, we present our approach and tools, developed within a national-level initiative aimed at simplifying the execution of administrative procedures. Section 3 discusses the broader applicability of the tools, and outlines challenges and future work.

2. A scalable approach: AdmPModeler & Crawl4PA

Administrative procedures are the ordered set of actions normally carried out, according to the channel legally established, to issue an administrative act. In the PNRR context, the "1000 Esperti" project[1] aims to simplify so-called "complex (administrative) procedures"[2], namely procedures that require the involvement of multiple actors and concern the following areas of intervention: environmental assessments and authorisations, land reclamation, renewable energy, waste management, construction and urban planning, public procurement, and digital infrastructure.

In our work, we collaborated with domain experts involved in the "1000 Esperti" project and proposed a two-step pipeline to address the challenges of automatically identifying public administrative procedures and assisting experts in finding them. The input of the pipeline consists of a list of seed URLs corresponding to the official websites of Italian public administrations. First, Crawl4PA (see Subsec. 2.1), a dedicated crawler, systematically collects procedural documents from these websites, applying targeted crawling and content extraction techniques. The output of Crawl4PA — a structured collection of procedural documents and metadata — serves as the input for the second component, AdmPModeler, an LLM-based system that extracts a model of the the underlying administrative procedures (see Subsec. 2.2). The pipeline outputs a visual process diagram and a structured CSV describing the procedure's metadata, both easily convertible into standard process modelling formats such as BPMN.

Figure 1 provides a graphical visualization of the overall pipeline, while in the following, we provide a detailed description of the two tools.

2.1. Crawl4PA

The extraction of structured information from public websites has been widely studied, using techniques ranging from traditional crawling and scraping to deep learning and language models. Web crawling enables large-scale content acquisition by systematically navigating websites [5], while scraping transforms HTML content into structured data, especially when no machine-readable formats are available [6]. These techniques have also been applied in the public sector, where structured data is essential for transparency and efficiency. Examples include the Italian initiatives OpenCivitas[3] and IndicePA[4], which aim to centralize public administration data. However, these efforts largely depend on manual input and rigid taxonomies, resulting in costs that are not always sustainable. In this context, LLMs offer new opportunities: recent work explores combining large-scale web crawling with LLM-based modeling [7, 8]. However, to our knowledge, none of the existing approaches pursue objectives similar to ours.

This work addresses that gap by integrating two key components into a unified system. The first component is Crawl4PA, a custom web crawling and scraping system designed to automatically identify and extract web pages containing administrative procedures from Italian public administration

[1] This Italian project, cf. https://www.espertipnrr.it/

[2] Cf. https://www.gazzettaufficiale.it/eli/id/2021/11/29/21A07051/sg

[3] https://www.opencivitas.it/

[4] https://www.indicepa.gov.it/ipa-portale/

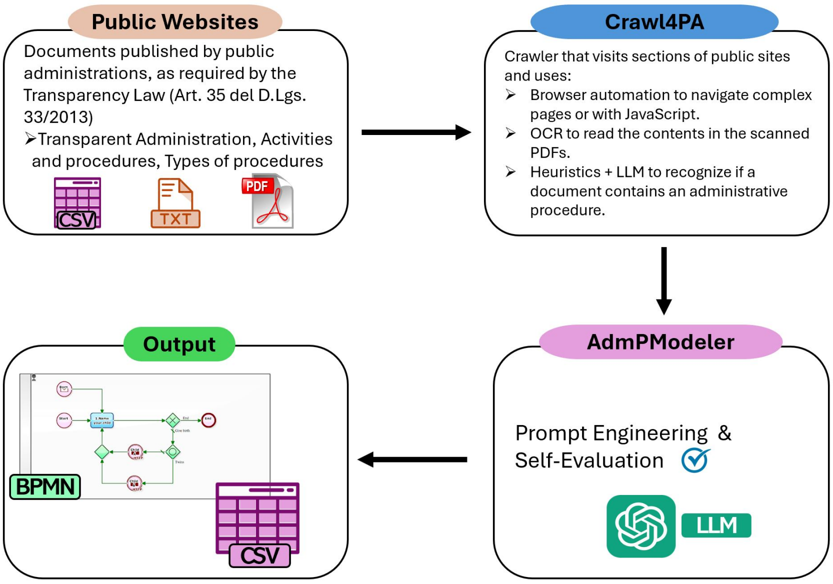

Public Websites
Documents published by public administrations, as required by the Transparency Law (Art. 35 del D.Lgs. 33/2013)
➢Transparent Administration, Activities and procedures, Types of procedures
CSV
TXT
PDF
Crawl4PA
Crawler that visits sections of public sites and uses:
➢ Browser automation to navigate complex pages or with JavaScript.
➢ OCR to read the contents in the scanned PDFs.
➢ Heuristics + LLM to recognize if a document contains an administrative procedure.
AdmPModeler
Prompt Engineering &
Self-Evaluation
LLM
Output
BPMN
CSV

Figure 1: Overview of the integrated pipeline.

websites. Crawl4PA was developed to address the challenges posed by the high variability in website structures, formats, and content quality.

The architecture of Crawl4PA includes several modules: a scheduler to manage the list of target URLs, a browser automation tool to navigate dynamic content, and a parser to extract text from web pages. In contrast to generic crawlers, Crawl4PA is optimized for public sector websites, which often rely on outdated platforms, inconsistent HTML structures, and documents published as scanned PDFs. In particular, the crawler begins its operation from known entry points—typically the “Amministrazione Trasparente” sections mandated by Italian law. Then it explores relevant subpages, identifying those likely to contain procedural information. For each page, it uses heuristic filters and language models to assess whether the content matches the structure and vocabulary of an administrative procedure. When such pages are found, Crawl4PA extracts the text, applies preprocessing (including Optical Character Recognition for scanned documents), and stores the content in a structured format. These outputs are then passed to AdmPModeler for semantic interpretation and modeling.

This component of our approach emerged from expert interviews conducted during the system validation phase. These discussions highlighted the need for a national-scale data collection effort specifically targeting the ”Amministrazione Trasparente” sections of Italian government websites. According to Law 33/2013[5], all public entities are required to publish their active procedures online. These pages are often similarly structured, providing a consistent target for large-scale crawling. Although the quality and consistency of the publications vary (PDFs, HTML tables, etc.), they represent the most concrete and widespread manifestation of procedural knowledge across thousands of Italian municipalities.

Crawl4PA is still in an early stage of development, hence it has not yet undergone rigorous validation. However, preliminary results on a diverse sample of municipal websites showed that it was able to detect and extract procedural content in the majority of cases, even on websites with poor accessibility or non-standard layouts. In addition, Crawl4PA adheres to ethical and legal requirements: it respects robots.txt directives, limits request rates, and focuses exclusively on publicly available and legally mandated content. Together with AdmPModeler, this system enables the large-scale, automated discovery and semantic interpretation of administrative procedures across Italy’s public administration ecosystem.

[5] https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:decreto.legislativo:2013-03-14;33!vig

2.2. AdmPModeler

The first version of AdmPModelertool has already been presented in a previous publication. For a detailed discussion of the tool, we refer the reader to [9] This tool moves beyond traditional event log-based approaches [10] to modeling structured processes since, in this work, the processes of interest are described in natural language rather then in datalog. Previous studies have proposed the use of intermediate representations (e.g., mermaid.js, Petri nets or directed graphs) to reduce structural complexity and focus on activity sequencing [11, 12], or phase-based and conversational strategies to extract process metadata [13, 14]. AdmPModeler leverages advanced prompt engineering techniques to tackle the complex task of extracting process models from unstructured legal texts. It uses prompt chaining, which breaks down tasks into sequential subtasks to improve output quality [15], and chain-of-thought reasoning to guide the model through intermediate reasoning steps [16, 17]. The tool also applies role prompting, instructing the model to act as a domain expert to enhance relevance and precision [18], and the LLM-as-a-judge approach [19, 20], enabling self-evaluation and correction of the generated results [21, 22].

AdmPModeler is a pipeline developed to convert legal texts that describe administrative procedures into formal, human-readable process models using LLMs. Firstly, we created a technique to extract generic process models. Secondly, we adapted this technique to the public administration domain. The technique leverages various prompt engineering methods, including prompt chaining, chain of thought, role prompting, and LLMs-as-a-judge. AdmPModeler takes as input one or more documents describing the administrative procedure to be modeled and returns a representation of the procedure in both diagram format and a tabular CSV description. The diagram provides a clear and immediate visualization of the activity flow within the procedure, while the CSV offers a detailed metadata representation for each activity. Moreover, the tool's output can be easily converted into standard process modeling formats, such as BPMN. To achieve this, AdmPModeler follows a four-step process that transforms legal text into a formal process model. First, it generates a semi-structured description of the procedure in a simple, human-readable format. Second, the self-evaluation phase allows the LLM to review and correct its output. Third, the process is converted into a structured representation, typically a table, ready for computational use. Finally, additional metadata is extracted for each activity, providing essential context derived directly from the legal text. Fig. 2 shows graphically the sequence of the four prompts.

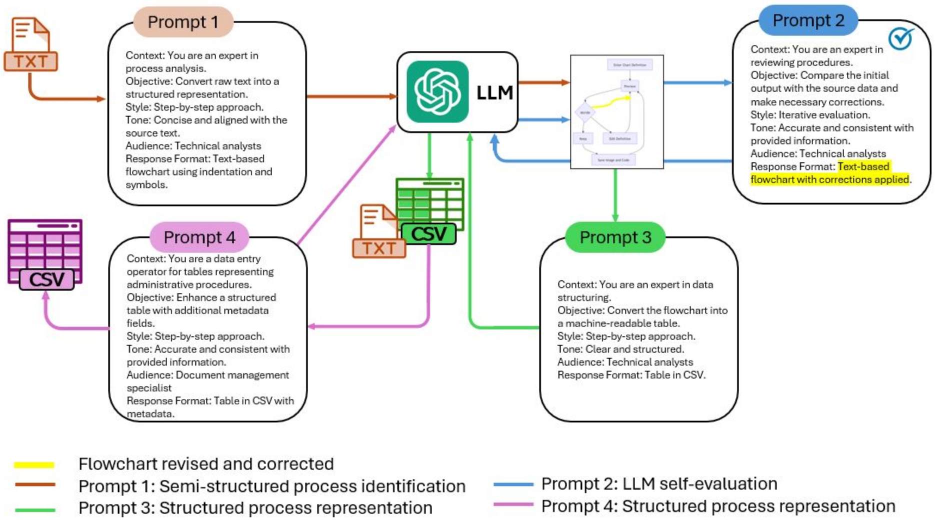
The evaluation of AdmPModeler was conducted on three administrative procedures. For each, AdmPModeler extracted structured data from legal texts using the OpenAI GPT-4o model[6] (version `gpt-4o-2024-08-06`). A domain expert then manually created reference tables and compared them with the tool's output, classifying each metadata field as correct or incorrect. The evaluation showed that AdmPModeler achieved an average accuracy of 78.2% in extracting metadata from administrative procedures. Results vary by metadata type, with best performance in summarization tasks and lower accuracy for logic- or fact-based fields, such as document requirements. Despite some need for manual correction, experts confirmed the tool significantly reduces workload and provides a reliable starting point.

3. Challenges and discussion points

Our two-step pipeline brings to light several key challenges that deserve attention from the research community:

- **Scalability with respect for local specificity:** Italy counts 7,986 municipalities, many of which publish administrative procedures that partially overlap yet retain local differences. Is it possible to cluster these procedures to measure their true diversity while preserving meaningful distinctions?

[6] GPT-4o model's web page: https://platform.openai.com/docs/models/gpt-4o


TXT
Prompt 1
LLM
Prompt 2
CSV
TXT
CSV
Prompt 4
Prompt 3
Flowchart revised and corrected
Prompt 1: Semi-structured process identification
Prompt 3: Structured process representation
Prompt 2: LLM self-evaluation
Prompt 4: Structured process representation

Prompt 1

Context: You are an expert in process analysis.
Objective: Convert raw text into a structured representation.
Style: Step-by-step approach.
Tone: Concise and aligned with the source text.
Audience: Technical analysts
Response Format: Text-based flowchart using indentation and symbols.

Prompt 2

Context: You are an expert in reviewing procedures.
Objective: Compare the initial output with the source data and make necessary corrections.
Style: Iterative evaluation.
Tone: Accurate and consistent with provided information.
Audience: Technical analysts
Response Format: Text-based flowchart with corrections applied.

Prompt 3

Context: You are an expert in data structuring.
Objective: Convert the flowchart into a machine-readable table.
Style: Step-by-step approach.
Tone: Clear and structured.
Audience: Technical analysts
Response Format: Table in CSV.

Prompt 4

Context: You are a data entry operator for tables representing administrative procedures.
Objective: Enhance a structured table with additional metadata fields.
Style: Step-by-step approach.
Tone: Accurate and consistent with provided information.
Audience: Document management specialist
Response Format: Table in CSV with metadata.

Figure 2: AdmPModeler detailed pipeline.

- **Heterogeneous and inconsistent data sources:** The information retrieved from the Transparent Administration sections of institutional websites includes both structured formats (e.g., tables) and unstructured documents (e.g., PDFs of varying quality). How can we design robust extraction and analysis techniques that accommodate this heterogeneity? Can Large Language Models (LLMs) reliably process and extract valuable information across such diverse formats?
- **Human-in-the-loop validation:** Despite the promising results of LLM-based extraction, expert review remains essential to ensure accuracy and legal soundness. How can we optimize this interaction to achieve an effective balance between scalability and reliability?

The ultimate goal of this work is to build a national repository of administrative procedures — semantically tagged, searchable, and continuously updatable. Such a repository would not only support transparency and accountability but also facilitate advanced activities in administrative simplification, policy experimentation, and automated compliance verification.

Preliminary validation of AdmPModeler has already demonstrated that, when properly guided, LLMs can generate accurate structured representations from complex legal texts, achieving an average accuracy of over 78%. However, this evaluation has been conducted in isolation, focusing exclusively on the second stage of the pipeline.

The next phase of this research will involve a comprehensive, end-to-end validation of the entire two-step pipeline, covering all stages from web crawling to process modeling. This broader evaluation will allow us to assess the overall effectiveness, robustness, and scalability of the proposed approach under real-world, large-scale conditions.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4.5 for grammatical correction only.

References

[1] P. Johannesson, E. Perjons, Design Science, Springer, 2014.

[2] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258 (2021).

[3] F. Bianchini, M. Calamo, F. De Luzi, M. Macrì, M. Mecella, Enhancing complex linguistic tasks resolution through fine-tuning llms, rag and knowledge graphs (short paper), in: International Conference on Advanced Information Systems Engineering, Springer, 2024, pp. 147–155.

[4] F. De Luzi, M. Macrì, M. Mecella, T. Mencattini, Cicero: An ai-based writing assistant for legal users, in: International Conference on Advanced Information Systems Engineering, Springer, 2023, pp. 103–111.

[5] M. A. Khder, Web scraping or web crawling: State of art, techniques, approaches and application., International Journal of Advances in Soft Computing & Its Applications 13 (2021).

[6] R. R. NR, M. Vijayalakshmi, et al., Web scrapping tools and techniques: A brief survey, in: 2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT), IEEE, 2023, pp. 1–4.

[7] UncleCode, Crawl4AI: Open-source LLM friendly web crawler & scraper, https://github.com/unclecode/crawl4ai, 2024. Accessed: 2025-06-23.

[8] H. Lai, X. Liu, I. L. Iong, S. Yao, Y. Chen, P. Shen, H. Yu, H. Zhang, X. Zhang, Y. Dong, et al., Autowebglm: A large language model-based web navigating agent, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 5295–5306.

[9] M. Macrì, F. De Luzi, M. Mecella, Admpmodeler: Modeling administrative processes using large language models. a case study, in: Electronic Government, 2025.

[10] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, Essential Process Modeling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2018, pp. 75–115. URL: https://doi.org/10.1007/978-3-662-56509-4_3. doi:10.1007/978-3-662-56509-4_3.

[11] A. Beiser, D. Penz, Making llms reason? the intermediate language problem in neurosymbolic approaches, arXiv preprint arXiv:2502.17216 (2025).

[12] M. Forell, S. Schüler, Modeling meets large language models, in: Modellierung 2024 Satellite Events, Gesellschaft für Informatik eV, 2024, pp. 10–18420.

[13] H. Kourani, A. Berti, D. Schuster, W. M. van der Aalst, Process modeling with large language models, in: International Conference on Business Process Modeling, Development and Support, Springer, 2024, pp. 229–244.

[14] A. Nour Eldin, N. Assy, O. Anesini, B. Dalmas, W. Gaaloul, Nala2bpmn: Automating bpmn model generation with large language models, in: International Conference on Cooperative Information Systems, Springer, 2024, pp. 398–404.

[15] S. Sun, R. Yuan, Z. Cao, W. Li, P. Liu, Prompt chaining or stepwise prompt? refinement in text summarization, in: Findings of the Association for Computational Linguistics ACL 2024, 2024, pp. 7551–7558.

[16] M. A. Sultan, J. Ganhotra, R. F. Astudillo, Structured chain-of-thought prompting for few-shot generation of content-grounded qa conversations, arXiv preprint arXiv:2402.11770 (2024).

[17] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.

[18] N. Wu, M. Gong, L. Shou, S. Liang, D. Jiang, Large language models are diverse role-players for summarization evaluation, in: CCF International Conference on Natural Language Processing and Chinese Computing, Springer, 2023, pp. 695–707.

[19] C. Ryu, S. Lee, S. Pang, C. Choi, H. Choi, M. Min, J.-Y. Sohn, Retrieval-based evaluation for llms: a case study in korean legal qa, in: Proceedings of the Natural Legal Language Processing Workshop 2023, 2023, pp. 132–137.

[20] J. Saad-Falcon, O. Khattab, C. Potts, M. Zaharia, Ares: An automated evaluation framework for retrieval-augmented generation systems, arXiv preprint arXiv:2311.09476 (2023).

[21] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, et al., A survey on llm-as-a-judge, arXiv preprint arXiv:2411.15594 (2024).

[22] D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, et al., From generation to judgment: Opportunities and challenges of llm-as-a-judge, arXiv preprint arXiv:2411.16594 (2024).