Producing Structured Data from Historical Sources: A Preliminary Application to French Senate *Tables*

Joël Féral[1], Joseph Chazalon[2], and Marie Puren[2]

[1] École Nationale des Chartes, Paris, France
[2] LRE, EPITA, Le Kremlin-Bicêtre, France

Abstract

We address the extraction of structured data from noisy historical documents (namely, the 1931 *Tables nominatives* of the French Senate) using a LLM guided by lightly constrained generation rather than strict post-hoc validation. Our contribution is threefold: (1) a minimal, application-driven target schema (speaker name + list of page references) expressed so it can be injected into the prompt to steer generation; (2) a hybrid pipeline that decouples OCR from schema-oriented generation, leveraging the LLM's tolerance to OCR noise while limiting hallucinations via an expected JSON format; (3) an evaluation protocol for structured outputs using optimal record matching and a continuous Integrated Matching Quality metric that overcomes precision/recall brittleness. Code and data are publicly available at `https://github.com/EPITAResearchLab/feral.25.chr`.

Keywords: Parliamentary Archives, Structured Data, Generative Models, Evaluation

1 Introduction

The growing use of artificial intelligence by historians [3] is multiplying the possibilities for producing historical datasets. The advent of large language models (LLMs) is further changing the landscape, especially for processing textual data corpora, with a proliferation of uses and experiments in the humanities and social sciences.[1] Zero-shot LLMs are capable of performing a wide range of tasks without the need for task-specific examples or fine-tuning [12; 23; 27] and have demonstrated their ability to carry out many time-consuming tasks in historical research, such as transcription [6], information extraction [9], or annotation [25].

The use of large language models (LLMs) opens new perspectives for extracting structured data [13] from historical documents. In the context of historical data extraction, a central challenge lies in obtaining structured outputs. One approach is structured generation, which constrains an LLM to directly produce information in a predefined format such as JSON. Alternatively, structure can be imposed through post-processing of free-form text outputs. Regardless of the approach, producing structured data enables traceability back to the original document and facilitates source verification. It also supports downstream uses, such as integration into a database or further computational analysis.

Two fundamental issues still remain: (1) how to move from raw text to an exploitable structured representation, such as a table or CSV file; and (2) how to assess the quality and reliability of the extracted data. This article addresses both aspects through a concrete case study: the extraction

Joël Féral, Joseph Chazalon, and Marie Puren. "Producing Structured Data from Historical Sources: A Preliminary Application to French Senate *Tables*." In: *Computational Humanities Research 2025*, ed. by Taylor Arnold, Margherita Fantoli, and Ruben Ros. Vol. 3. Anthology of Computers and the Humanities. 2025, 904–920. https://doi.org/10.63744/oXn2aMxza3iJ.

[1] The "DH@LLM: Grands modèles de langage et humanités numériques" conference program, held in Paris in July 2025, is a good illustration of this: https://www.crihn.org/nouvelles/2025/01/16/colloque-dhllm-grands-modeles-de-langage-et-humanites-numeriques-sorbonne-universite/

of structured information from the 1931 *Tables nominatives* or *Tables des noms* of the French Senate (index of senatorial activity ordered by name). We explore a lightly constrained generation approach using an LLM and propose a method to represent target data, guide the extraction process, and evaluate system performance. Beyond this specific case, the study contributes reflections on the feasibility and limitations of generative models for structuring historical data.

The *Tables des noms* of the French Senate was published during the French Third Republic (1870–1940).[2] Within the broader documentary ecosystem of the *Journal Officiel*—which seeks to reconstruct parliamentary activity and its legal or regulatory outcomes in France—the Senate's *Tables nominatives* offer a concise and systematic record of senators' interventions during public sessions. These indexes were designed to accompany the transcription of debates[3] and to facilitate their consultation. Manually compiled once a year, they recorded each intervention by senators or members of the government who spoke during the sessions, the subject of their speech, and the corresponding page number. The term *"interventions"* is used here to denote any active participation in parliamentary proceedings, including speeches, questions, statements, and legislative initiatives. While these tables were particularly useful at a time when full-text search in digitized parliamentary debates was not possible, they still hold significant value for historians today. Systematically extracting data from them would make it possible to track parliamentary activity over the long term, quantify the interventions of specific senators affiliated with particular political movements, or support the cross-validation of named entities extracted from the debates themselves. Our objective is to extract structured data from these *Tables*; for our initial experiments, we focus on a single *Table nominative*, namely that of 1931. The early 1930s marked the beginning of the decline of French parliamentarism, culminating in the fall of the Third Republic in 1940 [18]. Analyzing the 1931 *Table* allows us to lay the groundwork for a broader study that will extend across the entire decade, with the aim of capturing the parliamentary activity of the Senate and, subsequently, of the Chamber of Deputies.

After reviewing existing approaches to structured data extraction and evaluation (Section 2), we present three main contributions. (1) We design and implement a schema-guided data processing pipeline for extracting structured information from the *Tables nominatives* of the French Senate, combining OCR transcription and prompt-based generation with a large language model (Section 3); (2) We introduce a dedicated evaluation protocol tailored to this task, including a matching-based alignment method and a continuous quality metric that accounts for partial and noisy outputs (Section 4); and (3) We provide an empirical assessment of LLM-based extraction in this historical setting, showing that model performance is significantly shaped by the design of the prompt and data schema—elements we propose to treat as critical parameters of the overall modeling process (Section 5).

2 Related Work

Building upon our initial argument, we structure our review around three key questions: (1) How can structured data be effectively modeled? (2) How can the quality of structured data produced by such approaches be evaluated? (3) How can structured data be generated from text?

2.1 Modeling Structured Data

Structured data can take various forms, including:

[2] These tables are part of the *Tables annuelles* (yearly activity index), which can be consulted on the digital library of the French national library (*BnF*): https://gallica.bnf.fr/ark:/12148/cb371291967/date.item.

[3] The complete transcriptions of Senate debates can be consulted via Gallica: https://gallica.bnf.fr/ark:/12148/cb34363182v/date.

Record Sets: These are unordered sets of tuples, similar to database tables where columns represent attributes of objects. This structure is commonly employed in Information Extraction tasks, such as named entity recognition [19] or relation extraction [16].

Record Sequences: These are ordered versions of record sets, where the sequence of records holds significance. The order may reflect criteria such as time or facilitate tasks like cross-validation, as seen in directories.

Trees: These hierarchical structures are often used to represent nested relationships or dependencies, such as in dependency parsing [15].

Graphs: These are flexible structures used to represent consolidated knowledge, such as ontologies or knowledge graphs. While these are widely studied, their evaluation typically falls outside the scope of Information Extraction and is beyond the focus of this work.

In this paper, we focus on record sets and sequences, as they are the most relevant for our case study involving the extraction of structured data from parliamentary documents. The various models we experimented with are described in Appendix 3.

2.2 Evaluating the Quality of Structured Data

The evaluation of structured data quality can be broadly categorized into two types of metrics: edit distance metrics and matching metrics, as defined in [2].

Edit Distance Metrics: These metrics involve complex optimization processes and are often computationally intensive. Additionally, their interpretability is limited, as they do not provide a direct comparison between the produced and expected data. Examples include the classical Levenshtein distance for character-level comparisons and Tree-Edit Distance [26] for tree structures. While general graph edit distance metrics exist, their complexity often renders them impractical for real-world applications.

Matching Metrics: Matching metrics are more interpretable, as they explicitly identify the elements that match between the produced and expected data. Most approaches rely on bipartite matching between the predicted and reference data sets, computing scores based on the number of matched elements [2]. Common metrics include the F1 score, which combines precision and recall, and the Jaccard index, which measures set similarity. However, fewer studies address structured data or partial matching, where the produced data may not perfectly align with the expected data. It is interesting to note that the computer vision community share the exact same problem, and a similar framework is proposed in the context of the COCO Panoptic Segmentation Challenge [7]. The evaluation protocol relies on an optimal matching between the surfaces of the predicted and ground truth segmentations to jointly evaluate detection, segmentation and classification of regions—which is similar to the approach of Chen et al. [2].

In our work, we adopt a matching metric based on optimal matching between structured data sets, which generalizes the bipartite matching approach while accommodating partial matches. This approach not only provides a quantitative evaluation of data quality but also identifies missed or hallucinated elements, offering actionable insights.

2.3 Producing Structured Data from Text

Approaches for generating structured data from text can be broadly divided into two categories: detection-based (extractive) and generation-based (generative or abstractive).

Detection-Based Approaches: These methods focus on identifying text fragments corresponding to specific fields or elements of the structured data. Traditional approaches relied on rule-based methods, such as regular expressions or heuristics. More recently, machine learning models, such as sequence labeling models (e.g., CRF-based methods [5]) or transformer-based encoder-only models (e.g., BERT [4]), have become prevalent. Transformer-based models are

particularly effective due to their intrinsic capabilities and their ability to be fine-tuned on specific tasks with relatively small datasets. Additionally, their design enforces strict alignment between input text and output labels, reducing the risk of hallucinations (i.e., generating data not present in the input). However, these approaches require task-specific training, which demands data, computational resources, expertise, and time.

Generation-Based Approaches: These methods leverage autoregressive models to generate structured data by "translating" the input text into the desired format. The advent of LLMs has spurred interest in this approach due to their generalization capabilities across diverse tasks without requiring task-specific training [1; 21; 22]. Outputs can be constrained to specific formats, such as JSON, or more complex structures, as long as the set of valid tokens can be dynamically computed [24]. These models can produce complex, nested structures by generating elements sequentially and can infer implicit elements not explicitly present in the input text. However, their main drawback is their susceptibility to hallucinations, which are challenging to detect.

In this paper, we explore the effectiveness of generation-based approaches for handling repetitive structures in parliamentary indexes or *Tables*. We leverage the zero-shot capabilities of LLMs and evaluate the viability of this approach for generating structured data in this specific context.

3 Schema-Guided Extraction of Parliamentary Interventions

3.1 The 1931 *Tables nominatives* of the French Senate

There is an edition of the *Tables du Journal Officiel* for each year, typically comprising around 450 pages in the 1930s. The section entitled *Tables des noms*—which includes both the Senate and the Chamber of Deputies—spans approximately forty pages, with the Senate portion generally covering around fifteen. For the year 1931, the Senate's *Tables des noms* consists of 14 pages and roughly 300 entries, each corresponding to an intervention in the assembly. Each entry is associated with a speaker and details various types of actions (requests for interpellation, bill discussions, reading of committee reports, submission of amendments, etc.), along with a page reference directing the reader to the full transcription of the intervention. These transcriptions are published in the Senate's *Débats parlementaires*. The *Tables* are therefore functionally linked to the transcriptions through page numbers. Moreover, as pagination is continuous throughout the year, each page reference makes it possible to accurately determine the date of the corresponding intervention.

3.2 Pipeline for Schema-Guided Structured Data Extraction

Despite recent advances in Large Vision-Language Models (LVLMs), their end-to-end, zero-shot performance remains insufficient for high-accuracy OCR tasks [14]. To address this, we adopt a straightforward pipeline that leverages the strengths of specialized components to maximize overall extraction accuracy.

First, each page image is processed independently using the PERO OCR engine [8; 10; 11] to detect and transcribe text. Given the persistent challenges in general page layout segmentation, we generate three transcription variants per page to capture the variability inherent in such pipelines. Such variants will be described in more detail in Section 4.1.

Once the text for each page is obtained, we concatenate the transcriptions from all relevant pages to form a single text stream. This aggregated text is then provided as input to a Large Language Model (LLM), which is instructed to produce the target structured data. To simplify the evaluation process, we limit our current study to single-page extraction.

While LLMs can be prompted to generate structured outputs, it is essential to constrain their generations to match the expected format. Several methods exist for enforcing such constraints; the most effective to date involves filtering valid tokens during inference using an external validator, such as a finite state automaton [24], and is commonly available in commercial LLM APIs.

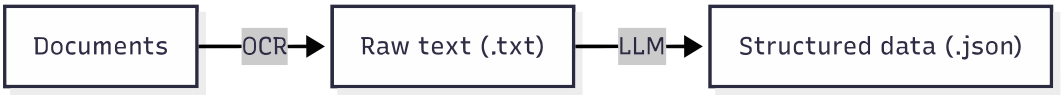

Figure 1: Document processing pipeline. The input to the LLM consists of the concatenated OCR-extracted raw text, a natural language prompt, and a predefined schema describing the target structure. The LLM generates a structured JSON object as output.

Our information extraction process thus relies on submitting OCR-processed text to the LLM and obtaining output that conforms to a predefined JSON schema. This setup requires three key components: **a data schema**, **a prompt**, and **an API supporting constrained output**.

For this study, we selected the Mistral API,[4] using the Ministral 8B Instruct v2410 model [17]. This choice is motivated by the model's strong zero-shot performance, cost-effectiveness, and the public availability of its weights for research purposes.[5]

3.3 Data Modeling and Schema Definition

The primary goal of this study is to extract structured information on parliamentary activity in the French Senate for the year 1931, with the practical objective of building an interactive timeline that visualizes the density of interventions over time. To ensure the reliability and interpretability of this timeline, it is crucial to provide clear indicators of extraction quality.

Directly linking extraction reliability to the confidence in answering historical research questions remains an open challenge. Therefore, we focus our evaluation on well-defined metrics that quantify the similarity between the predicted intermediate data structure and a reference (ground truth) structure. While these metrics do not yet account for the semantic impact of each error, they offer a transparent basis for assessing extraction performance.

The core extraction task centers on identifying individual entries in the *Tables nominatives* —specifically, the names of senators and their associated page references. These page numbers serve as indirect temporal markers, as the source documents use continuous pagination throughout the year. The extracted information is represented in JSON format, which is both structured and interoperable, facilitating downstream processing and conversion to tabular formats (e.g., CSV). Figure 2 illustrates the target output structure.

The data schema guiding LLM inference is defined at the granularity of speaker names and their corresponding page references, which suffices for constructing the intended timeline. We use Pydantic,[6] a Python library for data modeling, to specify the schema in a JSON-compatible format with strict type validation and embedded descriptions. These description fields serve as semantic tags, enhancing both prompt clarity and LLM guidance. Figure 5 (see Appendix C) presents the simplified schema used in this work.

3.4 Prompting the LLM for Structured Data Extraction

The prompt provided to the LLM (see Appendix B) is designed to guide the extraction of political participants—primarily senators and ministers—and their associated page references from the input text. Its structure is as follows:

- **Task Definition:** The prompt clearly states that the input consists of entries, each corresponding to an individual involved in Senate activities (e.g., senators, ministers), and that

[4] The Mistral API Documentation is available at `https://docs.mistral.ai/api/`.
[5] Ministral 8B weights are available at `https://huggingface.co/mistralai/Ministral-8B-Instruct-2410`.
[6] Pydantic documentation is available at `https://docs.pydantic.dev`.

```
{
"list_of_speakers": [
{
"name": "Dentu",
"page_references": [
1024,
1031,
1560,
1563,
1564
]
},
{
"name": "Desjardins (Charles)",
"page_references": [
563
]
}
]
}
```


Figure 2: Example JSON output representing structured data extracted from the *Tables nominatives*. Each entry corresponds to a participant (e.g., a senator) and a list of page numbers where they are mentioned. These references act as temporal markers due to the continuous pagination of the source. For simplicity, intervention categories are omitted in this study.

the goal is to extract their names and the page numbers referencing their interventions.

- **Key Term Clarification:** Definitions are provided for essential terms such as "entries" and "actions" to ensure unambiguous interpretation.
- **Handling Special Cases:** The prompt specifies procedures for cases such as index cross-references (where no page numbers are given, only a reference to another entry) and split entries spanning multiple pages (which are to be ignored in this preliminary study to maintain extraction simplicity).
- **Formatting Instructions:** Explicit guidelines are given for representing names (e.g., first names in brackets after last names) and formatting page references.

Optionally, the prompt could be further improved by incorporating additional historical context or representative examples (few-shot prompting).

3.5 Iterative Refinement of the Data Model and Prompt

Both the data schema and the extraction prompt underwent iterative refinement during this study. This process was motivated by the observation that the LLM occasionally proposed alternative, and sometimes more effective, ways of structuring the extracted information than initially anticipated. For instance, the model would naturally deduplicate repeated page references for a given speaker, streamlining the output beyond the original schema specification. Consequently, the construction of the ground truth required a balance between adhering to formal schema constraints and accommodating the LLM's practical structuring tendencies.

To ensure unbiased evaluation, all schema and prompt adjustments were based exclusively on performance observed on a single development page (referred to as **"page 02"** in experiments),

with the remaining pages reserved for final testing. This approach aligns with standard machine learning practice, maintaining a clear separation between development and evaluation data.

Future work could automate this refinement loop or incorporate more systematic prompt engineering strategies.

4 Experiments

The evaluation setup was established through an initial development phase, during which the data model, prompt instructions, and reference structured data for a selected development page were iteratively refined. After finalizing this phase, we applied the baseline model to a broader set of pages and manually corrected the outputs to construct an unbiased ground truth for evaluation.

This section details the resulting dataset, the variants generated for analysis, and the evaluation protocol employed to rigorously assess prediction quality.

4.1 Dataset

Ground truth data were constructed to rigorously evaluate extraction quality across varying levels of OCR text fidelity. For each selected page, three distinct OCR variants were generated using the PERO OCR engine [8; 10; 11]: (1) a manually corrected version serving as the gold standard, (2) a version with manual layout segmentation, and (3) a raw, uncorrected OCR output. This design enables systematic assessment of the extraction pipeline's robustness to noise and layout artifacts.

A random sample of five sequential pages was selected for manual transcription and annotation. Because pages are not necessarily contiguous, some speaker entries may be incomplete; in such cases, the LLM was instructed to omit these partial elements. This setup reflects realistic extraction challenges, as entries may span multiple pages. (Handling such cases in production would require multipage or streaming input, which is beyond the scope of this study.)

Each page was processed independently to avoid optimistic bias from sequential context. This sometimes resulted in extractions starting mid-entry, providing the LLM with truncated information and highlighting its ability to interpret partial context. Although the source documents are generally well-digitized, occasional distortions (e.g., page folds) and fragmented input introduce realistic difficulties, exposing model limitations in non-ideal conditions.

Structured outputs were generated using the Ministral 8B model, with a fixed prompt and a Pydantic schema to enforce output consistency. A temperature of zero was used to ensure deterministic results. For each page and OCR variant, the model produced structured JSON outputs, which were then compared to manually curated ground truth representations.

The evaluation covers 109 entries across five pages, with each entry assessed for all three OCR conditions.

4.2 Structured Output Evaluation Protocol

The evaluation aims to rigorously assess the structured outputs generated by the LLM (denoted as P) against a manually curated ground truth (G). As described in Section 3.3, each data instance consists of a **list of speaker entries**, where each entry comprises a **speaker name** and a **list of pages** referencing their speeches.

A key challenge arises from the fact that the model may produce the correct set of entries but in a different order, or with minor structural variations. To address this, we adopt a flexible alignment strategy inspired by prior work [2; 7], leveraging optimal transport to establish a one-to-one correspondence between predicted and ground truth entries. This approach accommodates order invariance and tolerates minor discrepancies, enabling a robust evaluation of extraction quality.

4.2.1 *Entry-level Distance and Optimal Assignment*

To rigorously compare predicted and ground truth entries, we define a normalized entry-level distance function that quantifies the similarity between each pair.

For the textual component (senator name), we compute the Ratcliff/Obershelp distance, which measures the similarity based on the longest common subsequence, after lowercasing and trimming whitespace. This yields a normalized distance $d_n(g_i, p_j) \in [0, 1]$, where 0 indicates an exact match and 1 indicates complete dissimilarity.

For the list of referenced pages, we use the Intersection-over-Union (IoU) set distance:

$$d_p(g_i, p_j) = 1 - \frac{|\text{ref\_pages}(g_i) \cap \text{ref\_pages}(p_j)|}{|\text{ref\_pages}(g_i) \cup \text{ref\_pages}(p_j)|}.$$

The overall entry distance is defined as the product of these two components:

$$d_e(g_i, p_j) = d_n(g_i, p_j) \times d_p(g_i, p_j).$$

To establish a one-to-one correspondence between predicted and ground truth entries, we employ optimal transport [20]. This approach finds the assignment that minimizes the total distance across all pairs, accommodating order invariance and structural discrepancies. The resulting alignment provides a principled basis for evaluating extraction quality at the entry level.

4.2.2 *Limitations of Standard Metrics: Precision, Recall, and F1-Score*

Conventional metrics such as **precision**, **recall**, and **F1-score** are widely used to evaluate extraction tasks. Precision quantifies the proportion of correctly generated entries among all model outputs, while recall measures the fraction of ground truth entries successfully retrieved. The F1-score, as their harmonic mean, is intended to provide a balanced summary of performance.

However, in our evaluation protocol—where predicted and ground truth entries are aligned one-to-one using optimal transport—these metrics become unreliable. The injective nature of the alignment ensures that the number of predicted entries always matches the number of ground truth entries. As a result, precision is trivially maximized, regardless of the actual quality of the matches, and recall fails to reflect missing or spurious entries, since all elements are forcibly paired. Consequently, the F1-score inherits these distortions, leading to an inflated and potentially misleading assessment of model performance.

4.2.3 *Integrated Matching Quality (IMQ): A Robust Evaluation Metric*

While standard metrics fail to capture the nuanced quality of structured matches, our protocol leverages the entry-level distance d_e to quantify the fidelity of each aligned pair. For each match, we define a quality score $q_i = 1 - d_e(g_i, p_i)$, where $q_i \in [0, 1]$ reflects the closeness of the predicted entry to its ground truth counterpart.

$$F(t) = \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}_{q_k \geq t}$$

The **Integrated Matching Quality (IMQ)** is then defined as the area under this curve:

$$\text{IMQ} = \int_0^1 F(t)\, dt$$

IMQ provides a comprehensive summary of extraction quality, rewarding both the number and the closeness of matches. A score of 1 indicates perfect alignment, while lower values reflect increasing divergence. This continuous, threshold-free metric is particularly well-suited to evaluating LLM outputs, where minor deviations are common even under strong structural constraints. IMQ thus enables a principled, fine-grained assessment of structured extraction performance.

Source	Precision (Biased)	Recall (Biased)	IMQ	Ground Truth Entries	Predicted Entries	Matches
page 02	1.0000	0.9565	0.9059	23	22	22
page 03	1.0000	1.0000	0.8928	25	25	25
page 04	1.0000	1.0000	0.9591	19	19	19
page 05	1.0000	1.0000	0.8636	19	19	19
page 10	1.0000	1.0000	0.8193	23	23	23

Table 1: Summary of results for each OCR-scanned page (without any correction).

5 Results and Analysis

We applied our matching method to five different pages (109 entries), each processed independently and exhibiting varying OCR qualities. Table 1 presents the results for each OCR-scanned page—processed without manual segmentation or correction—along with the corresponding sizes of the ground truth and predicted sets, and the number of matches ultimately selected by optimal transport.

All pages exhibit perfect biased precision and recall; however, as previously discussed, these metrics are inherently limited in our context. Since they are directly derived from the optimal assignment—enforcing a one-to-one matching between the two sets, at least when they are of equal size—they do not fully reflect alignment quality.

The IMQ, on the other hand, offers a more nuanced assessment by capturing the distribution of match quality across all possible thresholds. For all processed pages, IMQ scores remain consistently high (ranging from 0.8193 to 0.9591), reflecting a strong homogeneity among correspondences. The IMQ also indirectly assesses the quality of the predicted dataset: not only are the matches structurally complete, but they also maintain an overall high level of semantic and syntactic proximity. Thus, the IMQ can be interpreted as a hybrid metric, functioning as a qualitative recall indicator while also integrating a proxy for precision, through penalization of low-quality matches.

Minor variations are observed across pages. Pages 5 and 10 show somewhat lower IMQ scores, likely due to document-specific typographic inconsistencies. On these pages, a significant number of first names are not enclosed in parentheses following the last names, contrary to the formatting assumed in the ground truth. Specifically, 21% of entries on page 5 and 39% on page 10 exhibit this discrepancy, compared to 0% on the other pages. This typographic variation increases the string comparison cost and negatively impacts match quality.

Page 3, although exhibiting perfect biased recall and precision, has a slightly lower IMQ (0.8928). This may be attributed to OCR quality issues, particularly a fold in the gutter that introduces visual noise and degrades recognition performance.

Conversely, page 2 shows a high IMQ (0.9059), despite an imperfect recall. This is likely due to a sampling bias, as the prompting process for the LLM was initially designed with the structure of this specific page in mind. Accordingly, the strong performance on this page should be interpreted cautiously, as it does not necessarily generalize to the others. However, we can see that the device adapts well to page 4, which has the best score.

The 95.65% recall on page 2 stems from a specific edge case in the source document, where an individual is listed twice (once as a senator and once as a minister). This leads to two distinct entries in the ground truth, while the LLM output consolidates them into a single prediction. We chose to preserve this functional distinction in the ground truth, while the model opted to factorize the information. This weakness is therefore linked to the design of the ground truth.

Comparison with the deemed-perfect OCR is presented in Table 2. Performance is better overall, i.e., IMQ metric values are all greater. The potentially superior performance of the noisy OCR might stem from its capture of running headers at the top of pages, providing better context in

Source	Precision (Biased)	Recall (Biased)	IMQ	Ground Truth Entries	Predicted Entries	Matches
page 02	1.0000	1.0000	0.9513	23	23	23
page 03	1.0000	1.0000	0.9430	25	25	25
page 04	1.0000	1.0000	0.9821	19	19	19
page 05	1.0000	1.0000	0.8778	19	19	19
page 10	1.0000	1.0000	0.8966	23	23	23

Table 2: Summary of results for the deemed-perfect OCR.

situations where entries at the beginning of a page are truncated. In the case of page 2, the LLM reproduced the repetition and thus the distinction between a senator holding two functions. This drop in results is less linked to the LLM's superior performance in a noisy context than to a better alignment of its behavior with the ground truth's expectations.

Consequently, there would also be a need to evaluate the prompt subsequently, as it is a crucial parameter for avoiding these errors. It's worth noting, however, that bypassing these exceptions in a massive extraction scenario would imply perfect knowledge of specific cases, which can be typographical or related to highly situational institutional choices. A schema with a reasonable degree of granularity and generic prompting allows for obtaining results that can be reasonably trusted, without requiring atomic knowledge of the documents' form. The calculation of statistics per page also offers bundles of clues about internal exceptions within the document structure, which can be significant.

6 Conclusion

This work explored using large language models for structured data generation from historical sources, focusing on a specific case study, namely the 1931 *Tables nominatives* of the French Senate. The approach—combining OCR, schema-guided structuring, and constrained generation via LLM—produced results evaluated with a more appropriate metric for an optimal alignment protocol, linking reference data with predicted data. The introduction of the IMQ metric was particularly crucial, allowing us to assess structuring quality beyond the traditional precision/recall scores, which are inadequate in this context.

Several avenues emerge for strengthening the approach's robustness and generalization. A central challenge, already noticeable in this work, lies in better connecting response hypotheses to research questions with automatically produced intermediate data. This will allow for evaluating their long-term robustness concerning the production process. Conversely, evaluating the prompt itself remains to be done to achieve a truly complete evaluation protocol. In any case, data structuring cannot be considered a simple, neutral pre-processing step: it dictates the form of possible historical analyses.

From this perspective, the prompt and the data model must be considered as "meta-parameters" of the entire historical data production system. It becomes essential to conceptualize their generation and adjustment as an integral part of the historical data processing chain. A promising path would involve systematizing and automating this meta-optimization process to make these approaches reproducible, transparent, and accessible to non-expert users.

References

[1] Brown, Tom B. et al. “Language models are few-shot learners”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS ’20. Vancouver, BC, Canada: Curran Associates Inc., 2020. DOI: 10.5555/3495724.3495883.

[2] Chen, Yunmo, Gantt, William, Chen, Tongfei, White, Aaron, and Van Durme, Benjamin. “A Unified View of Evaluation Metrics for Structured Prediction”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics, 2023, pp. 12868–12882. DOI: 10.18653/v1/2023.emnlp-main.795.

[3] Clavert, Frédéric and Muller, Caroline. “L’histoire au temps des algorithmes : Une réflexion prospective sur l’introduction de l’intelligence artificielle en histoire au 21e siècle”. In: *20 & 21. Revue d’histoire 162*, no. 2 (2024), pp. 13–26. DOI: 10.3917/vin.162.0013.

[4] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[5] Finkel, Jenny Rose, Grenager, Trond, and Manning, Christopher. “Incorporating non-local information into information extraction systems by Gibbs sampling”. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL ’05*. Ann Arbor, Michigan: Association for Computational Linguistics, 2005, pp. 363–370. DOI: 10.3115/1219840.1219885.

[6] Humphries, Mark, Leddy, Lianne C., Downton, Quinn, Legace, Meredith, McConnell, John, Murray, Isabella, and Spence, Elizabeth. “Unlocking the Archives: Using Large Language Models to Transcribe Handwritten Historical Documents”. In: *Historical Methods: A Journal of Quantitative and Interdisciplinary History 58*, no. 3 (2025), pp. 175–193. DOI: 0.1080/01615440.2025.2500309.

[7] Kirillov, Alexander, He, Kaiming, Girshick, Ross, Rother, Carsten, and Dollár, Piotr. “Panoptic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9404–9413.

[8] Kišš, Martin, Beneš, Karel, and Hradiš, Michal. “AT-ST: Self-training Adaptation Strategy for OCR in Domains with Limited Transcriptions”. In: *Document Analysis and Recognition – ICDAR 2021*. Springer International Publishing, 2021, pp. 463–477. DOI: 10.1007/978-3-030-86337-1_31.

[9] Knutsen, Gunnar W. “Alimenter des bases de données grâce à l’intelligence artificielle”. In: *Histoire & mesure 39*, no. 2 (2024). DOI: 10.4000/140kk.

[10] Kodym, Oldřich and Hradiš, Michal. “Page Layout Analysis System for Unconstrained Historic Documents”. In: *Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II*. 2021, pp. 492–506. DOI: 10.1007/978-3-030-86331-9_32.

[11] Kohút, Jan and Hradiš, Michal. “TS-Net: OCR Trained to Switch Between Text Transcription Styles”. In: *Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV*. 2021, pp. 478–493. DOI: 10.1007/978-3-030-86337-1_32.

[12] Kojima, Takeshi, Gu, Shixiang Shane, Reid, Machel, Matsuo, Yutaka, and Iwasawa, Yusuke. “Large Language Models are Zero-Shot Reasoners”. 2022. DOI: 10.5555/3600270.3601883.

[13] Liu, Yu, Li, Duantengchuan, Wang, Kaili, et al. “Are LLMs good at structured outputs? A benchmark for evaluating structured output capabilities in LLMs”. In: *Information Processing & Management 61*, no. 5 (2024), p. 103809. DOI: 10.1016/j.ipm.2024.103809.

[14] Lv, Tengchao et al. “KOSMOS-2.5: A Multimodal Literate Model”. Aug. 21, 2024. DOI: 10.48550/arXiv.2309.11419.

[15] Marneffe, Marie-Catherine de, Dozat, Timothy, Silveira, Natalia, Haverinen, Katri, Ginter, Filip, Nivre, Joakim, and Manning, Christopher D. “Universal Stanford dependencies: A cross-linguistic typology”. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Reykjavík, Iceland: European Language Resources Association (ELRA), May 2014, pp. 4585–4592.

[16] Mintz, Mike, Bills, Steven, Snow, Rion, and Jurafsky, Daniel. “Distant supervision for relation extraction without labeled data”. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. ACL-IJCNLP 2009, ed. by Keh-Yih Su, Jian Su, Janyce Wiebe, and Haizhou Li. Suntec, Singapore: Association for Computational Linguistics, Aug. 2009, pp. 1003–1011. (Visited on 07/17/2025).

[17] Mistral AI. “Un Ministral, des Ministraux: Introducing the world’s best edge models”. Mistral AI Blog. Oct. 2024.

[18] Morel, Benjamin. *Le Parlement, temple de la République : de 1789 à nos jours*. Paris: Passés composés, 2024.

[19] Nadeau, David and Sekine, Satoshi. “A survey of named entity recognition and classification”. In: *Lingvisticæ Investigationes 30*, no. 1 (2007). Publisher: John Benjamins Type: Journal Article, pp. 3–26. ISSN: 0378-4169. DOI: 10.1075/li.30.1.03nad.

[20] Peyré, Gabriel and Cuturi, Marco. “Computational Optimal Transport”. 2020. DOI: 10.1561/2200000073.

[21] Radford, Alec, Narasimhan, Karthik, Salimans, Tim, and Sutskever, Ilya. “Improving Language Understanding by Generative Pre-Training”. OpenAI, 2018.

[22] Radford, Alec, Wu, Jeffrey, Child, Rewon, Luan, David, Amodei, Dario, and Sutskever, Ilya. “Language Models are Unsupervised Multitask Learners”. OpenAI, 2019.

[23] Wei, Jason et al. “Emergent Abilities of Large Language Models”. 2022. DOI: 10.48550/arXiv.2206.07682.

[24] Willard, Brandon T and Louf, Rémi. “Efficient Guided Generation for Large Language Models”. 2023. DOI: 10.48550/arXiv.2307.09702.

[25] Yuan, Yunshuang and Sester, Monika. “Leveraging LLMs and attention-mechanism for automatic annotation of historical maps”. In: *AGILE: GIScience Series 6* (2025), p. 52. DOI: 10.5194/agile-giss-6-52-2025.

[26] Zhang, Kaizhong and Shasha, Dennis. “Simple Fast Algorithms for the Editing Distance between Trees and Related Problems”. In: *SIAM Journal on Computing 18*, no. 6 (Dec. 1989). Publisher: Society for Industrial and Applied Mathematics, pp. 1245–1262. DOI: 10.1137/0218082. (Visited on 07/17/2025).

[27] Zhao, Wayne Xin et al. “A Survey of Large Language Models”. 2023. DOI: 10.48550/arXiv.2303.18223.

A An example of “Tables du Journal Officiel”

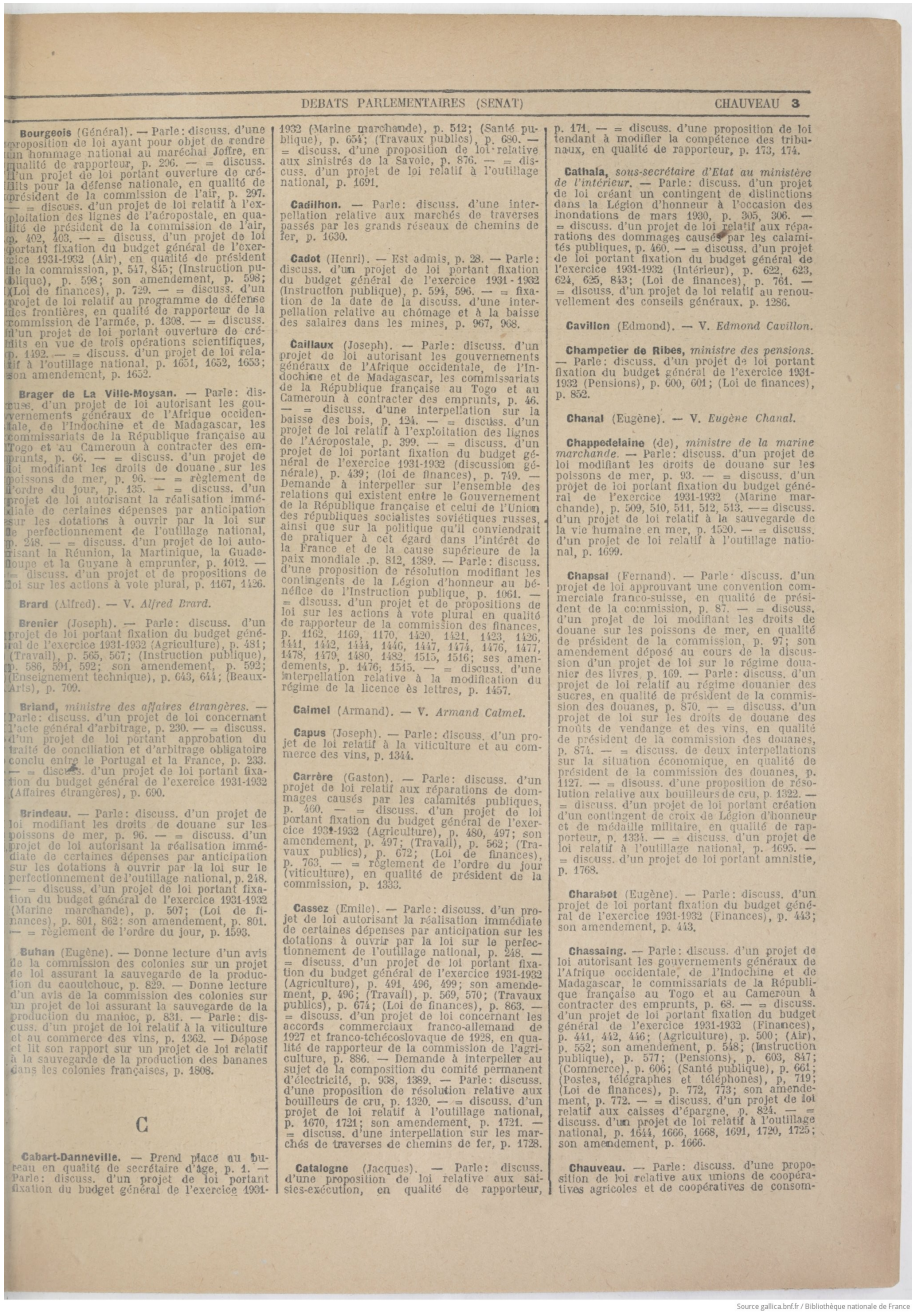

Figure 3: A page from the 1931 Senate *Table des Noms*

B Full prompt

This appendix features an English translation of the complete prompt used in our experiment, visible in Figure 4.

```
<TASK TO DO>: Extract from the text I am about to give you,
information from each entry, each of which relates to one person.
<NEED TO KNOW>: First of all, be aware that there is one entry per
person and that, for context, the people mentioned have participated
in the activity of the Senate. They are generally senators, ministers,
undersecretaries, etc.
<ENTRIES>: Each entry consists of: the NAME and sometimes the FIRST
NAME of a speaker (str); sometimes his role (this is not always
specified); a list of ACTIONS he has performed or which concern him.
<ACTION>: Each action concerning a speaker is generally linked to one
or more page numbers. When there is a page reference, you can be sure
that it is a reference to an action concerning the stakeholder.
<INDEX REFERENCE CASE>: In the case where an entry does NOT set out
actions or facts and/or pages concerning a speaker, but a simple
nominal mention, then it is an index reference. In this case, you
should indicate the reference of the reference (str). These references
are generally the first names and surnames of contributors. These
references therefore do not refer to pages, but to other nominal
entries.
<HERE IS THE INFORMATION TO BE EXTRACTED>: So I want you to give me
the surnames (and first names if there are any); as well as the page
numbers relating to the descriptions of the actions or interventions
of each speaker -- List[int]-- OR, if there is no action, just say
that it is an index reference ("<index reference>") -- (str).
<NOTE>: - When there is no index reference (a str), adopt this syntax
at the appropriate level: "references_pages":"<index_reference>".
- First names must be placed in brackets and after the name.
- If a page reference appears several times in an entry (i.e. for the
same speaker), there is no need to repeat it.
<ATTENTION>: The text submitted to you may be truncated. If this is
the case, ignore the incomplete text and consider only the complete
entries.
SO HERE IS THE TEXT from which you need to extract the information:
```

Figure 4: Submitting a request to an LLM via the Mistral API, including entity extraction instructions, the raw text, and the expected data schema.

C Full data schema

```
class Speaker(BaseModel):
name: str = Field(..., description="Name (and first name if applicable)")
page_references: List[int] = Field(...,
description="List of page numbers where the speaker is referenced,
else <index cross-reference>")

class SenatorsInterventions(BaseModel):
list_of_speakers: List[Speaker] = Field(...,
description="List of all speakers")
```

Figure 5: Simplified data schema defined using Pydantic. Each participant is described by a name and a list of page numbers, which serve as indirect temporal markers. Description fields embedded in the schema act as semantic tags, guiding the language model in entity extraction.

D Full experimental results

The following three tables present the complete experimental results discussed in Appendix 5 of the main paper:

- Table 3 reproduces the results for the deemed-perfect OCR condition (copy of Table 2 in main paper). The text used as input to the LLM has been manually corrected to ensure high fidelity.
- Table 4 presents the results for the OCR condition based on manual segmentation. The text used as input to the LLM has been processed with manual layout segmentation but without further correction.
- Table 5 reproduces the results for the raw OCR condition (copy of Table 1 in main paper). The text used as input to the LLM is the raw output of the OCR engine, without any manual correction or segmentation. It represents the most realistic scenario for large-scale extraction, and can contain errors due to both layout detection and character recognition.

Source	Precision (Biased)	Recall (Biased)	IMQ	Ground Truth Entries	Predicted Entries	Matches
page 02	1.0000	1.0000	0.9513	23	23	23
page 03	1.0000	1.0000	0.9430	25	25	25
page 04	1.0000	1.0000	0.9821	19	19	19
page 05	1.0000	1.0000	0.8778	19	19	19
page 10	1.0000	1.0000	0.8966	23	23	23

Table 3: Results for deemed-perfect OCR (copy of Table 2 in main paper).

Source	Precision (Biased)	Recall (Biased)	IMQ	Ground Truth Entries	Predicted Entries	Matches
page 02	1.0000	0.9130	0.9016	23	21	21
page 03	1.0000	1.0000	0.9395	25	25	25
page 04	1.0000	1.0000	0.9784	19	19	19
page 05	1.0000	1.0000	0.8793	19	19	19
page 10	1.0000	1.0000	0.8873	23	23	23

Table 4: Results for OCR based on manual segmentation

Source	Precision (Biased)	Recall (Biased)	IMQ	Ground Truth Entries	Predicted Entries	Matches
page 02	1.0000	0.9565	0.9059	23	22	22
page 03	1.0000	1.0000	0.8928	25	25	25
page 04	1.0000	1.0000	0.9591	19	19	19
page 05	1.0000	1.0000	0.8636	19	19	19
page 10	1.0000	1.0000	0.8193	23	23	23

Table 5: Results for raw OCR, without segmentation or correction (copy of Table 1 in main paper).