CienaLLM: Generative Climate-Impact Extraction from News Articles with Autoregressive LLMs

Javier Vela-Tambo
Student
Pyrenean Institute of Ecology (IPE-CSIC)
Zaragoza, Spain
jvela@ipe.csic.es

Jorge Gracia
Supervisor
Aragon Institute of Engineering Research, Universidad de Zaragoza
Zaragoza, Spain
jogracia@unizar.es

Fernando Dominguez-Castro
Co-supervisor
Pyrenean Institute of Ecology (IPE-CSIC)
Zaragoza, Spain
fdominguez@ipe.csic.es

***Abstract*—Understanding and monitoring the socio-economic impacts of climate hazards requires extracting structured information from heterogeneous news articles on a large scale. To that end, we have developed CienaLLM, a modular framework based on schema-guided Generative Information Extraction. CienaLLM uses open-weight Large Language Models for zero-shot information extraction from news articles, and supports configurable prompts and output schemas, multi-step pipelines, and cloud or on-premise inference. To systematically assess how the choice of LLM family, size, precision regime, and prompting strategy affect performance, we run a large factorial study in models, precisions, and prompt engineering techniques. An additional response parsing step nearly eliminates format errors while preserving accuracy; larger models deliver the strongest and most stable performance, while quantization offers substantial efficiency gains with modest accuracy trade-offs; and prompt strategies show heterogeneous, model-specific effects. CienaLLM matches or outperforms the supervised baseline in accuracy for extracting drought impacts from Spanish news, although at a higher inference cost. While evaluated in droughts, the schema-driven and model-agnostic design is suitable for adapting to related information extraction tasks (e.g., other hazards, sectors, or languages) by editing prompts and schemas rather than retraining. We release code, configurations, and schemas to support reproducible use.**

***Index Terms*—Information Extraction, Large Language Models, Generative Information Extraction, Prompt Engineering, Climate Impacts, Drought, News Articles**

I. Introduction and Related Work

Extreme climate and weather events, including floods, hailstorms, and heatwaves, are among the most disruptive manifestations of climate change, producing severe socio-economic and environmental consequences. Such events can damage ecosystems, threaten food security, and in the most severe cases endanger lives [1]. Among these hazards, drought is responsible for one of the largest losses, reducing agricultural production, stressing the water and energy systems, and increasing social vulnerability [2]–[5]. With climate change increasing both the frequency and severity of drought events [6], [7], systematic knowledge of their evolution is urgently needed.

Traditional drought indices, such as the Standardized Precipitation Index or the Palmer Drought Severity Index [8]–[10], capture the physical evolution of drought events, but do not reflect their social and ecological impacts. Understanding these dimensions requires information on the impacts that droughts produce across sectors and regions. Drought impacts databases like the European Drought Impact Report Inventory [11] provide valuable evidence but remain uneven in coverage and detail, limiting systematic monitoring and adaptation planning. By contrast, newspapers have documented the consequences of droughts for decades at national or regional scale, making them one of the richest sources of impact information [12]. However, the impact information in newspapers is highly dispersed and heterogeneous. Previous studies have used manual or semi-automatic content analysis of news articles to reconstruct drought events and their impacts, including those carried out in Ireland [13], [14], the United States [15], Australia [16]–[18], the United Kingdom [19], and Spain [20], [21]. These efforts underscore the potential of news as an impact source, but their reliance on human annotation limits both scale and generalizability, restricting analyses to narrow regions or short timeframes.

Automatically extracting structured information from news articles is a challenging task. They are often long and heterogeneous, combining relevant details with unrelated narrative elements. Impacts are described in varied and nuanced ways, and geographic references are often implicit, appearing through mentions of rivers, reservoirs, or towns rather than administrative units, and therefore require contextual inference [22]–[24]. These challenges are not limited to drought: similar difficulties arise when analyzing other events such as floods, hailstorms, or heatwaves, or when seeking related information.

Transformer-based models such as BERT and its derivatives leverage self-attention to capture context, enabling fine-tuned supervised systems that learn rich representations from text and achieve strong task performance [25]–[27]. This has enabled the training of supervised models that build on autoencoding language models, allowing them to learn rich contextual representations and achieve strong task performance. In the climate domain, supervised NLP approaches have been widely applied, covering tasks such as classifying article relevance, assigning impact categories, and recognizing toponyms and sectoral entities. Many of these systems build on machine learning methods, including encoder-based language models with task-specific heads trained on carefully curated datasets [28]–[35].

López-Otal et al. [36]introduce SeqIA, a supervised approach that trains separate classifiers on top of Transformer encoders to detect drought-relevant articles and to classify their impacts in Spanish news articles. SeqIA achieves strong in-domain performance, but, like other supervised systems, it depends on new annotations and retraining when applied to different hazards, languages, regions, or information schemas. Moreover, although the system attempts location extraction, it identifies all locations mentioned rather than isolating only those directly affected by drought, a distinction that is crucial for spatial analysis. Accurately extracting the truly impacted locations is challenging, as it often requires inferring them from indirect references such as rivers, reservoirs, or nearby towns.

Autoregressive generative Large Language Models (LLMs), such as GTP-4 [37], or Llama [38],represent a shift beyond encoder-based approaches. Rather than producing fixed representations for downstream classifiers, these models are trained on massive text corpora to generate text token by token, capturing broad contextual and world knowledge. Unlike supervised encoders, which require task-specific retraining, generative LLMs can adapt to new tasks with little or no additional supervision, making them highly versatile across domains [39], [40].

Schema-guided Generative Information Extraction (GenIE) builds directly on this capability by prompting LLMs to output structured information aligned to a predefined schema (e.g., impact type, location) in JSON format [41]–[49]. This paradigm eliminates the need for task-specific fine-tuning and enables rapid adaptation to evolving schemas and cross-domain applications. GenIE has shown promise in areas as diverse as health, biomedicine, and chemistry [50]–[53]. Li et al. [54] have explored this paradigm for climate impacts, demonstrating feasibility. However, their work uses a fixed taxonomy (EM-DAT [55]), evaluates only a narrow LLM coverage, and relies on Wikipedia, a more structured and unambiguous source of information.

We adopt a schema-guided GenIE approach to process drought-related news from Spain, addressing three tasks: (i) detecting article relevance, (ii) identifying multiple impact types, and (iii) extracting impacted locations. We use open-weight LLMs, whose weights are downloadable for local inference (e.g. Gemma [56], Llama [38], Qwen [57]). This enables on-premise experiments through ecosystems such as Ollama[1], improving privacy, controllability, reproducibility, and often cost and energy efficiency in low-resource settings. These advantages stand in contrast to closed API-only systems (e.g., GPT-4 [37], Gemini [58]). We compare model families and model sizes, using the number of parameters as a proxy for capacity. We also compare full-precision against quantized precision regimes. Quantization reduces latency and memory requirements while introducing only limited accuracy loss [59], [60]. In addition, we analyze the effect of different prompt engineering strategies [61]. Our evaluation focuses on three dimensions: accuracy (F1, precision, recall), efficiency (latency and compute), and reliability. Ensuring response reliability is challenging, as models may produce malformed JSON [62], hallucinate information [63], [64], or show sensitivity to prompt wording [65], [66].

Despite promising demonstrations of GenIE [54], no systematic evaluation has examined the capabilities of open-weight LLMs for climate-impact extraction from news. Key open questions remain: (i) how model family, size, and precision regime influence accuracy, efficiency, and reliability; (ii) how consistently models generate parsable, schema-conformant outputs; (iii) whether prompt engineering strategies can meaningfully improve extraction; and (iv) what performance–efficiency trade-offs arise across different models and configurations.

In this work, we pursue GenIE in a zero-shot setting: rather than training new classifiers, we rely on the general pretrained knowledge of LLMs and guide them solely with natural language prompts to extract structured information. A key advantage of this design is that the system inherits improvements in new LLM releases, including reasoning ability, factual coverage, and structured output reliability. These benefits are obtained without requiring additional annotation or retraining. We evaluate this approach on drought as a representative case while designing methods intended to generalize to other climate hazards and domains.

To address these gaps, we contribute the following.

1) **Framework**. We introduce CienaLLM[2] (Climate Impact Extraction from News Articles using LLMs), a modular, open-source toolkit for schema-guided GenIE.
2) **Evaluation**. We present a large-scale factorial study covering 384 configurations across 12 open-weight models, spanning families, sizes, and quantization regimes, under controlled prompting and parsing strategies.
3) **Comparison**. We benchmark CienaLLM against SeqIA [36] on shared datasets for drought impact extraction from Spanish news articles.
4) **Insights**. We provide a detailed analysis of performance–efficiency trade-offs, parsing reliability, and quantization viability, identifying which prompt strategies are most effective for different model families and sizes.
5) **Release**. We release code, configurations, and schema definitions to support reproducible research.

The remainder of this paper is organized as follows: Section II introduces the datasets, Section III presents the methodology and the CienaLLM framework, Section IV describes the experimental setup, and Section V reports the results. Finally, Section VI discusses the findings and Section VII concludes with directions for future work.

[1] https://ollama.com/

[2] https://github.com/lcsc/ciena_llm

II. Datasets

This study builds on and extends the datasets compiled by López-Otal et al. [36]. Their collection [67] includes two components: entire news articles labeled for drought relevance, and individual sentences from other articles annotated for specific drought impacts. We reuse these datasets but add new annotations that enable article-level impact extraction, spatial localization of drought effects, and mitigation of impacts imbalance.

To complement these datasets, we have downloaded the entire online archives of major Spanish outlets to assemble broader news corpora (see Appendix E). Combining national and regional sources ensures coverage across scales, making the corpora well suited for future large-scale analyses of climate-related impacts in Spain using the extraction tools developed in this study. All articles follow a standardized structure based on the *NewsArticle* schema[3], which provides consistent metadata such as headline, body text, publication date, and URL.

A. Drought Impacts Dataset

To evaluate the performance of our system in extracting drought-related impacts from news articles, we use two dataset collections: the *Drought Impact Identification Training Datasets* and the *End-to-End* (E2E) dataset [67]. Both have been reannotated and adapted in our work to support article-level impact evaluation. The news articles were sourced from *El País*[4] and Grupo Z, now acquired by Prensa Ibérica[5].

For the *Drought Impact Identification Training Datasets*, we reannotated a subset of 244 articles at the article level, assigning one or more impact labels based on the presence of relevant information anywhere in the article, rather than by sentence. Drought relevance annotations and articles from underepresented impact types were added to the original dataset. An important caveat of the original datasets is that the same article could contribute a sentence to the training split for one impact and to the test split for another, which limits direct comparability with SeqIA. The E2E dataset was also reannotated at the article level following the same criteria. The substantial inter-annotator agreement (Cohen's kappa = 0.695 ± 0.140) with a second expert who reannotated the E2E dataset at the article level highlights the inherent difficulty of annotating drought impacts in news articles [68].

Finally, we merged both reannotated datasets into a single evaluation resource, the *Drought Impacts Dataset* (DID). Only articles labeled as drought-related were retained, as the objective is to assess impact extraction after relevance filtering. The final dataset comprises 386 annotated articles with multi-label assignments for the four impact types: agriculture, livestock, hydrological resources, and energy. We applied a stratified 70/30 split into validation and test subsets, excluding combinations of labels with fewer than two samples to ensure meaningful stratification. Since this dataset is only used for evaluation, no training split was defined. Table I summarizes the composition of the merged dataset and its split.

TABLE I
Distribution of class labels across splits in the Drought Impacts Dataset.

Label	Validation	Test	Total
Agriculture	126	55	181
Livestock	67	30	97
Hydrological Resources	128	56	184
Energy	42	17	59
Total Articles	269	117	386

B. Drought Relevance Dataset

The *Drought Relevance Dataset* (DRD) [36] consists of 2,240 news articles from El País, published between 1976 and 2023, and labeled as either drought-related (1,270) or not (970). The dataset was originally split into training and test subsets for supervised classification, as shown in Table II. We reuse this dataset to compare the performance of our LLM-based approach with that of a traditional supervised method. Both approaches aim to identify drought-relevant news articles, but differ in how relevance is determined: via a trained classifier or a prompted language model.

TABLE II
Distribution of positive and negative labels across splits in the Drought Relevance Dataset.

Label	Train	Test	Total
Positive	903	367	1270
Negative	665	305	970
Total Articles	1568	672	2240

C. Drought Impact Locations Dataset

To evaluate the ability of our system to extract the geographical scope of drought-related impacts, we created the *Drought Impact Locations Dataset* (DILD). The DILD dataset consists of 100 drought-related articles from El País, annotated with the Spanish provinces affected by droughts as reported in news articles. We chose Spanish provinces as the target unit of analysis because drought impacts typically affect large regions rather than isolated municipalities, making provinces a suitable level of granularity for spatial annotation and evaluation, and is often used in drought research [2], [69]. Annotation was performed by a domain expert who assigned to each article the list of provinces where drought-related impacts were reported. In total, the dataset contains 835 annotated province-level entries across 100 articles, with each article linked to an average of 8.35 provinces. The number of provinces per article varies significantly, with a median of 1, but a maximum of 50 in some broad, national-level reports. All 50 Spanish provinces are represented in the dataset, ensuring full geographical coverage.

[3] https://schema.org/NewsArticle

[4] https://elpais.com

[5] https://www.prensaiberica.es

III. Methodology and Framework

A. Large Language Models

We selected twelve open-weight LLMs covering three major model families: Gemma by Google[6], Llama by Meta[7], and Qwen by Alibaba[8]. The selection spans three size tiers: small ($<$ 7B parameters), medium (7 $-$ 25B), and large ($>$ 25B). It also includes two precision regimes: full-precision (unquantized; `fp16`) and 4-bit mixed-precision quantization (`q4_K_M`). To directly assess the effect of quantization, we include for each family an unquantized medium-sized model with the same configuration otherwise. All models were obtained from the Ollama model library[9] at the time of study. Table III lists the main features of the selected models. In the table, and throughout the rest of this paper, each model configuration is given a unique identifier (`LLM`) that combines family, size, and quantization.

B. Prompt Design

The concrete prompt templates used in our experiments are detailed later in Section IV (Experimental Setup) and in Appendices A, B, and C. In this section we describe the overall design principles and prompt engineering techniques.

To construct the base prompt template, we analyzed the drought-impact extraction task and identified the specific elements to be extracted. These elements were then expressed as explicit natural language instructions to reduce ambiguity and improve reproducibility. The resulting template includes:

- **Task variables**, including the name of the climate event and its associated impact categories.
- **Input placeholders** for the news article's content, such as the headline, body, and publication date, which may provide contextual information for extraction.
- **Role prompting strategy**, where the LLM is instructed to act as "an expert in environmental analysis." This technique has been shown to improve performance by aligning model behavior with the target domain [70], [71].

Although the input articles are written in Spanish, we formulate all prompts in English. This design choice is motivated by evidence that LLMs trained primarily on English corpora perform more reliably when prompted in English [72]. The final template was refined iteratively through empirical testing and meta prompting [73], [74] on a small independent validation set to ensure consistency and robustness across different LLM architectures and sizes.

To improve performance beyond the base prompt, we evaluate four prompt engineering techniques commonly used in LLM-based task design:

- **Summarization** (`SUM`): We first perform a separate LLM call to generate a summary of the news article. The resulting summary, rather than the full article, is then inserted into the extraction prompt. This strategy aims to reduce prompt length and eliminate irrelevant or superfluous content [75], [76].
- **Chain of Thought** (`CoT`): We append the instruction "Reason step by step and explain your reasoning before giving the final answer" to the base prompt. Zero-shot CoT techniques encourage intermediate reasoning steps, which have been shown to improve factual accuracy in complex tasks [77].
- **Self-Criticism** (`SC`): Inspired by Self-Refine [78], we introduce a second LLM call that prompts to review its initial response. This self-critique may lead to more robust extractions by allowing the model to correct or refine prior outputs [79].
- **Impact Descriptions** (`DESC`): Instead of relying on short label names for impact categories, we incorporate detailed natural language descriptions directly into the base prompt template. This approach aims to enhance the model's ability to disambiguate overlapping or nuanced categories, at the cost of additional effort required from the researcher or system user to define these descriptions and an increase of the prompt's lenght.

Summarization and self-criticism require sequential LLM calls. They are implemented through dynamic prompt chaining, where the output of one step is parsed and embedded into the prompt for the next [80]. Although this approach increases execution time, it enables the model to focus more effectively on intermediate subtasks, potentially leading to improved overall extraction performance.

C. Structured Output Parsing

The LLM output must be converted to a structured format to enable downstream processing and evaluation. Our approach uses a schema that specifies the fields to be extracted (e.g., `event_type`, `impacts`, `locations`) and serves as the blueprint for both guiding the LLM output and parsing it after generation.

We explore two distinct methods for instructing the model to generate its output in a structured JSON format, and denote this configuration variable as `JGEN`.

- **Format-enforcing prompts** (`JGEN = prompt`): In this approach, the prompt explicitly provides JSON format instructions for the LLM to follow. These instructions are derived from the extraction schema and specify the expected structure, required fields, and valid data formats [42], [62].
- **Tool Calling Method** (`JGEN = tool`): For LLM APIs that support tool calling, we bind the extraction schema as a tool that the model can invoke [81], [82]. The LLM is instructed, via system message or API configuration, to return a tool call using the schema as its signature. This enables the model to produce natively structured output that can be parsed reliably without relying on prompt adherence to syntax.

In addition to the method used to generate the structured output, we also vary the response parsing strategy. This

[6] https://deepmind.google/models/gemma/

[7] https://www.llama.com/

[8] https://chat.qwen.ai/

[9] https://ollama.com/library

TABLE III
OVERVIEW OF THE OPEN-WEIGHT LLMS EVALUATED IN THIS STUDY.

LLM	Model Name	Family	Parameters (B) (Size)	Quantization	Release Date
gemma_4b	gemma3:4b-it-q4_K_M	Gemma	4 (S)	q4_K_M	March 10, 2025
gemma_12b	gemma3:12b-it-q4_K_M	Gemma	12 (M)	q4_K_M	March 10, 2025
gemma_12b_f	gemma3:12b-it-fp16	Gemma	12 (M)	fp16	March 10, 2025
gemma_27b	gemma3:27b-it-q4_K_M	Gemma	27 (L)	q4_K_M	March 10, 2025
llama_3b	llama3.2:3b-instruct-q4_K_M	Llama	3 (S)	q4_K_M	September 25, 2024
llama_8b	llama3.1:8b-instruct-fp16	Llama	8 (M)	fp16	July 23, 2024
llama_8b_f	llama3.1:8b-instruct-q4_K_M	Llama	8 (M)	q4_K_M	July 23, 2024
llama_70b	llama3.3:70b-instruct-q4_K_M	Llama	70 (L)	q4_K_M	December 7, 2024
qwen_3b	qwen2.5:3b-instruct-q4_K_M	Qwen	3 (S)	q4_K_M	September 19, 2024
qwen_7b	qwen2.5:7b-instruct-q4_K_M	Qwen	7 (M)	q4_K_M	September 19, 2024
qwen_7b_f	qwen2.5:7b-instruct-fp16	Qwen	7 (M)	fp16	September 19, 2024
qwen_72b	qwen2.5:72b-instruct-q4_K_M	Qwen	72 (L)	q4_K_M	September 19, 2024

determines when structuring takes place in the pipeline, as controlled by the configuration variable `RPARSE`:

- **Single-Step Parsing** (`RPARSE = False`): The model is instructed to extract and structure the information in a single LLM call. This is the most efficient method but also the most error-prone, as it involves the extraction and the response formatting task.
- **Two-Step Parsing** (`RPARSE = True`): The extraction and formatting steps are decoupled. The model first performs information extraction in free-form text. A second LLM call is then used to transform this unstructured output into the desired JSON format. This method increases execution time but often improves formatting reliability, especially when models struggle to produce well-formed JSON in one step.

D. CienaLLM Framework

We have developed CienaLLM, an open-source Python framework for experimentation with LLM-based information extraction and is available at https://github.com/lcsc/ciena_llm. CienaLLM implements all the prompting strategies and response parsing methods discussed in Sections III-A– III-C. The framework leverages LangChain[10] to orchestrate prompt construction, LLM calls, output parsing, and schema enforcement. Parsing is implemented using Pydantic[11], which validates whether the generated output conforms to this structure. It integrates with Ollama to support local inference of open-weight models via llama.cpp[12].

CienaLLM is programmatically configurable in Python, allowing users to define (i) an extraction schema, (ii) a base prompt template dynamically populated with task information and additional prompt techniques, (iii) the LLM backend, and (iv) optional multi-step inference through additional LLM calls such as summarization or response parsing. All components are implemented as extensible modules, and configurations can be specified in a YAML file to enable reproducible and parameterized experiments. This modular design makes CienaLLM well suited for rapid experimentation in low-resource settings and across domains.

Figure 1 illustrates the main steps of the CienaLLM pipeline. A news article is ingested and parsed for metadata such as headline, body, and publication date. If enabled, a summarization step performs an initial LLM call. The resulting prompt is sent to the LLM to extract information, which may be returned in plain text or structured JSON. Optionally, a second LLM call reformats the response into structured output. A self-criticism step can further refine the answer. The final result is saved as a structured JSON object and exported in CSV format for evaluation. It includes robust logging and error handling mechanisms to ensure traceability and reliability: malformed outputs and JSON parsing failures are automatically detected and logged, and execution times for each processing step are recorded to facilitate performance diagnostics.

IV. EXPERIMENTAL SETUP

We design a series of experiments to evaluate the ability of open-weight LLMs to extract structured information of drought events from news articles. The main experiment explores how model family, size, quantization, and prompt engineering techniques affect drought impact extraction performance. In addition to this core task, we conduct two additional experiments: one to detect whether a news article mentions a drought event, and another to identify the geographic locations affected by drought events. Based on the validation results of the primary drought impact extraction task, we select three representative configurations capturing different points of the performance–efficiency trade-off. These profiles are then used for the final test evaluation of the core task and reused in the two secondary tasks.

A. Drought Impact Extraction

The primary goal of this evaluation is to assess the performance of large language models in extracting drought impacts from Spanish news articles. We use CienaLLM to run and evaluate the different model configurations, prompting strategies, and parsing techniques introduced in Section III. The task is formulated as a multi-label classification problem, where the system must determine which of the following impact categories are present in each article: agriculture, livestock, hydrological resources, and energy. We evaluate

[10] https://www.langchain.com/
[11] https://docs.pydantic.dev/latest/
[12] https://github.com/ggml-org/llama.cpp

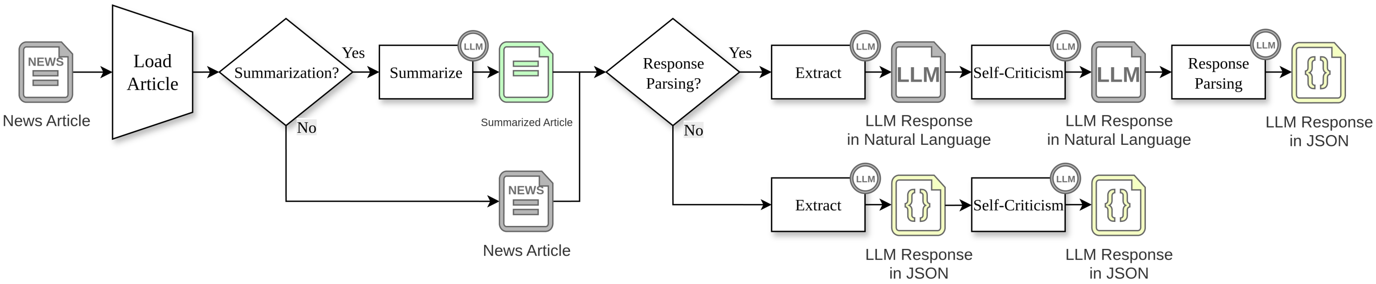

Fig. 1. Main components of the CienaLLM pipeline.

model performance on the DID introduced in Section II-A using both the validation and test splits.

To evaluate the impact of model choice and prompting strategies on extraction performance, we conduct an exhaustive exploration of all possible combinations of the 12 LLMs analyzed (Section III-A), the 4 prompt design options (Section III-B), and whether parsing is performed (Section III-C), making a total of 384 configurations. This exhaustive experimental design allows us to systematically analyze the individual and combined effects of each factor on extraction performance. The `JGEN` variable is fixed to `prompt` in all experiments to ensure consistency, as the `gemma` models do not support tool-calling in Ollama.

We configure the CienaLLM framework with a task-specific prompt template and a schema that defines the structured output. The prompt is dynamically assembled from a base prompt template containing the core task instructions and placeholders for the article's headline and body. The activated experimental flags determine its final form: `SUM` replaces the full article with a summary generated in an additional LLM call, `DESC` inserts natural language definitions of each impact category in the base prompt, `CoT` appends an instruction to the base prompt to reason step by step before answering, and `SC` triggers a second LLM call to review and correct the initial output. Finally, When `RPARSE = false`, the model is instructed to output JSON directly within the main extraction prompt; and when `RPARSE = true`, a second LLM call reformats the initial free-text extraction into JSON using a dedicated response parsing prompt. LLM responses are expected to follow a structured JSON format consisting of four binary fields as defined in the extraction schema (`agriculture`, `livestock`, `hydrological_resources`, and `energy`). A complete set of prompt templates, including examples for each component, as well as the complete schema definition, are provided in Appendix A.

To ensure reproducibility and consistency across runs, all LLM generations in the CienaLLM pipeline are performed with temperature set to 0, enforcing a greedy decoding strategy in which the model deterministically selects the most probable next token at each step. In addition, a fixed seed is set in the Ollama backend to ensure the same output is obtained across different machines and environments. To prevent excessively long generations or infinite loops, a maximum output length of 2,048 tokens is enforced for each generation call. The context window of the models is set at a limit of 32,768 tokens, which accommodates all prompt variants and news articles in the datasets.

We evaluate model performance on the drought impact extraction task using standard multi-label classification metrics (accuracy, precision, recall, and F1 score), computed as micro-averages across the four impact categories [83]–[86]. These metrics are calculated only on successfully parsed outputs, ensuring that extraction accuracy is assessed independently of formatting issues. To quantify reliability, we report the parsing error rate, defined as the percentage of outputs that could not be converted into the expected JSON format according to the extraction schema. Efficiency is measured as execution time per article, reflecting the wall-clock time to process one article through the full extraction pipeline, excluding model loading and warm-up.

For most validation-set results in the impact extraction task, we report three core metrics: F1 score, parsing error rate, and execution time per article. Together, these capture accuracy, reliability, and efficiency for comparing configurations during exploratory analysis. For the final test-set evaluation, we additionally report accuracy, precision, and recall alongside these core metrics.

To determine whether specific design choices significantly affect performance and efficiency, we conduct a series of non-parametric statistical tests across the full factorial space of experiments. We apply the Wilcoxon Signed-Rank test [87], pairing configurations that differ only in the factor under analysis while keeping all other variables constant. Tests are run on every evaluation metric, using only complete pairs. Statistical significance is assessed at $\alpha = 0.05$ with Bonferroni correction [88], [89] for multiple comparisons.

In addition to pairwise significance testing, we conduct a multi-objective Pareto-front analysis [90], [91] to characterize trade-offs between extraction performance and computational cost, treating each configuration as a point in the space defined by F1 score and execution time per article. From the results of the validation set, we identify three representative configurations from the Pareto front: *Best-F1*, *Efficient*, and *Fastest* (see Section V-D). These capture different points along the trade-off curve and are reused for the evaluation of the supplementary tasks.

B. Drought Relevance Classification

In addition, we asses the ability of large LLMs to detect whether a news article provides drought-relevant information, formulated as a binary classification. The evaluation is performed on the DRD test split, using the three representative configurations from the primary drought impact extraction task. A simplified prompt, derived from the base extraction template, asks the model to determine whether the article reports on a drought event, returning a JSON object with a single boolean field (`"drought"`). See the Appendix B for further information on the prompts and schema.

C. Drought Impact Location Extraction

Lastly, we evaluate the ability of our methodology to extract the geographic scope of drought impacts, specifically identifying the Spanish provinces mentioned or implied in news articles. The evaluation uses the DILD dataset and the three selected representative configurations.

The prompt asks the model to identify all Spanish provinces affected by drought according to the article. The expected output is a JSON object with the field `"provinces"` containing a list of province names. The prompt and corresponding schema are included in the Appendix C. After model inference, predicted province names are post-processed and normalized to match canonical Spanish province names. This step accounts for possible variations introduced by the model, including alternative spellings, punctuation differences, and the use of regional languages.

D. Infrastructure and Execution Environment

Although, initially tested on a consumer-grade NVIDIA RTX A4000 GPU (16 GB), the extensive factorial experiment with 384 configurations required a scalable infrastructure. We utilized the Galicia Supercomputing Center (CESGA)[13], specifically their GPU cluster with 64 nodes, each featuring two NVIDIA A100 GPUs (40 GB each). The CienaLLM framework (version v0.3.0[14]) was used to execute the extraction pipeline on an Ollama server on-premises. All parameters and outputs were systematically logged for traceability and reproducibility, enabling efficient, controlled exploration of model and prompt combinations.

V. Results

Additional results tables are provided in Appendix D.

A. Response Parsing and Reliability

Experiments without response parsing exhibited failures across models (e.g., trailing commas, missing required fields), causing loss of valid outputs. Averaged across all configurations and models, the parsing error rate without response parsing (`RPARSE = False`) reached 4.0%, compared to only 0.7% with response parsing (`RPARSE = True`) (see Table D.1).

At the model level (see Table D.2), the impact of `RPARSE` was uneven. Some models, such as `llama_8b`, failed to produce valid JSON for nearly 19% in some configurations, whereas `qwen` and `gemma` models already showed near-perfect formatting ($< 1\%$ errors). `RPARSE` reduced average parsing error rates for the most problematic ones i.e. `llama_8b` and `llama_3b`, from 13% and 19% respectively to below 4.5% in the least reliable configurations (see Figure 2).

However, parsing did not affect extraction accuracy. The F1 scores, computed only on successfully parsed outputs, remained almost identical for both settings, averaging 0.803 ± 0.061 without response parsing and 0.808 ± 0.048 with it (adjusted $p = 1.0$). The main drawback of enabling RPARSE is a moderate increase in execution time, from $10.2s \pm 10.6$ to $16.1s \pm 15.0$ per article, but this cost is outweighed by its reliability gains. The substantial loss of information caused by parsing errors in some configurations makes the results not directly comparable to those of more reliable models. Consequently, to ensure that all models are evaluated under equally reliable conditions, we exclude from the analysis configurations that lack response parsing.

B. Model-Level Performance Analysis

Figure 3 shows performance metrics for each model across configurations. Larger models exhibit both high average F1 and low variance, reflecting strong and stable performance across prompt configurations. In contrast, smaller models show lower scores and greater variance, with performance ranging widely depending on the configuration. This variability reflects each model's prompt sensitivity, that is, its susceptibility to changes in prompt formulation.

On average, `llama_70b` achieves the highest F1 score (0.858), followed by `qwen_72b` (0.852) and `gemma_27b` (0.833), all with perfect parsing reliability (see Table D.3). These results confirm the advantage of scale in achieving both accuracy and consistency. Smaller models like `gemma_4b` and `qwen_3b` trail behind in average performance and are more affected by prompt changes.

The best configuration for each model (see Table D.4) shows that most top-performing setups include `DESC`, often combined with `CoT`. For instance, `qwen_72b` reaches the highest F1 overall (0.878) with `CoT + DESC`, and `gemma_27b` reaches 0.865 using `SUM + CoT + DESC`. For `gemma`, `SUM` appears particularly effective and are featured in the best configuration of every model of the family. However, smaller models like `llama_3b` or `qwen_3b` achieve their best scores without prompt enhancements.

Full-precision (`fp16`) models consistently outperform their quantized (`q4_K_M`) counterparts by small but statistically significant margins: +0.008 F1 for `gemma`, +0.017 for `llama`, and +0.008 for `qwen` (see Table D.5). However, execution times are reduced by a 30%–40% while parsing reliability remains unaffected.

[13] https://www.cesga.es

[14] https://github.com/lcsc/ciena_llm/releases/tag/v0.3.0

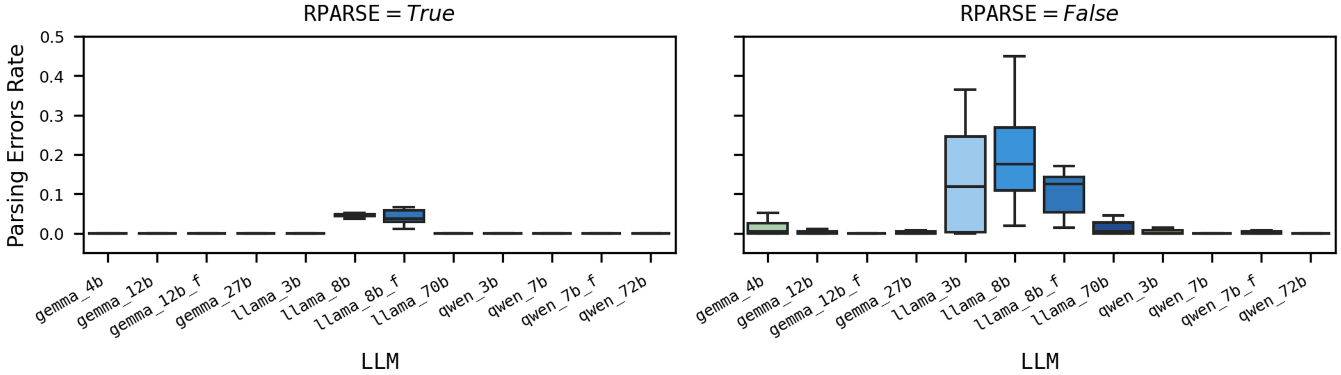

Fig. 2. Parsing error rates by model (LLM) with response parsing (RPARSE) enabled (left) and disabled (right).Without response parsing, several models, particularly the smaller Llama variants, exhibit high and variable error rates. In contrast, response parsing eliminates such failures across nearly all models.

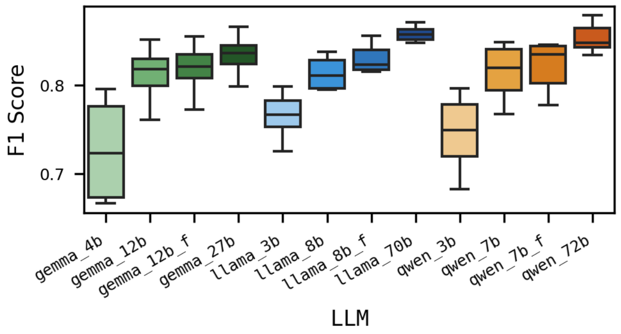

Fig. 3. Distribution of F1 scores across prompt configurations for each mode (LLM). Larger models generally achieve higher and more stable performance.

C. Effect of Prompt Strategies Across Models

We evaluate the impact of each prompt strategy on model performance and show the results in Table D.6. On average across all models, performance differences are small, indicating no universally dominant strategy. Execution time increases notably for strategies requiring extra LLM calls: SUM (+4.1 s/article), SC (+7.8 s/article), and CoT (+2.5 s/article), while parsing error rates remain largely unaffected. Clearer patterns emerge when analyzing at the model (see Figure 4 and Table D.7) and family level (see Table D.8):

- **Summarization** (SUM): Consistently improves performance in all gemma models, with statistically significant gains averaging +0.054, but has a negative effect on llama and qwen models, particularly in the medium-size range. This suggests gemma benefits from shorter, focused inputs, whereas the others rely more on full-article context.
- **Self-criticism** (SC): Provides no measurable gains. For many llama and qwen models, responses were identical with and without SC. Logs confirm that the instruction was ignored, suggesting the models tended to rely on their initial outputs, adding latency without benefit.
- **Chain of Thought** (CoT): Yields only modest improvements (±0.01), significant in a few medium/large models such as gemma_27b and qwen_7b_f. Larger checkpoints appear able to exploit explicit reasoning, whereas smaller ones lack sufficient capacity.
- **Impact Descriptions** (DESC): Slightly improves results for larger models (e.g., +0.009 for llama_70b and +0.021 for qwen_72b), while smaller models like qwen_3b and llama_3b achieve better results without the extra descriptive text. This indicates that richer label guidance pays off only when model capacity is sufficient.

These findings are consistent with previous section results (see Table D.4): the best-performing configuration for most models include DESC, SUM appears only for gemma, CoT helps about half the time, and small models often use no prompt enhancements at all.

D. Efficiency vs Performance Trade-Offs

To better understand the trade-off between extraction performance and computational cost, we conducted a multi-objective Pareto-front analysis using the F1 score and the execution time per article. Figure 5 visualizes all 384 evaluated configurations, highlighting those on the Pareto front, that is, configurations that cannot be outperformed simultaneously in both accuracy and efficiency. The results shows a clear performance–efficiency gradient: large models dominate the high-F1 region but incur high inference costs, while small and medium models achieve competitive efficiency, with some approaching the accuracy of much larger models.

From the Pareto front, we highlight three representative configurations that capture the main trade-offs: *Best-F1*, *Efficient*, and *Fastest* (see Table D.9). The *Best-F1* configuration (qwen_72b + CoT + DESC + PARSE) corresponds to the setting with the highest overall F1 score (0.878), though at a cost of 35.8 s/article. The *Fastest* setup (qwen_3b + PARSE) represents the minimum execution time, processing an article in only 2.6 s with moderate accuracy (F1 = 0.726). The *Efficient* profile (qwen_7b + DESC + PARSE) was explicitly chosen to represent a balanced compromise, achieving F1 = 0.844 at just 3.2 s/article.

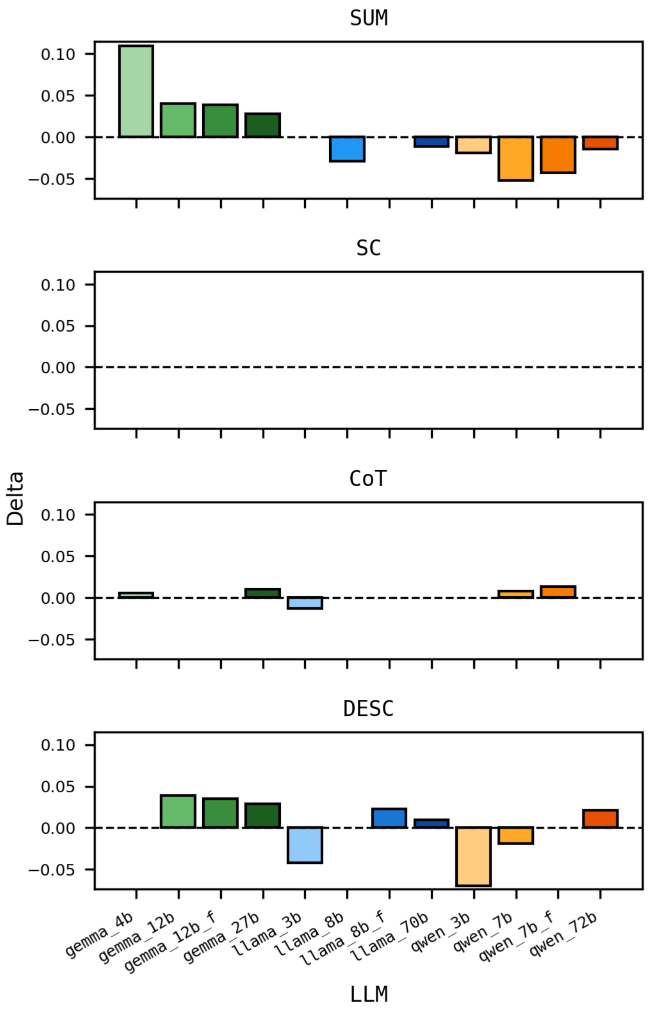

Fig. 4. Effect of prompt strategies on F1 scores by model (`LLM`). Bars show the change in mean F1 (Δ) relative to the base prompt for Summarization (`SUM`), Self-Criticism (`SC`), Chain of Thought (`CoT`), and Impact Descriptions (`DESC`), showing only statistically significant changes.

These configurations align with our earlier findings on prompt strategies. The *Best-F1* model benefited from richer guidance through `CoT` and `DESC`, while omitting summarization, which was ineffective outside `gemma`. The *Fastest* configuration used no enhancements, consistent with the tendency of small models to ignore or degrade under additional prompting. The *Efficient* setup leveraged `DESC` effectively, confirming its value as the most cost-efficient strategy when model capacity is sufficient. In terms of resources, the computational requirements scale sharply with model size: the `qwen_72b` model demands high GPU resources, the `qwen_3b` model could run on the CPU. The `qwen_7b` setup offers a practical compromise, delivering accuracy near larger LLMs on consumer-grade GPUs.

Altogether, the three selected configurations capture the performance–efficiency spectrum and will be used as reference points in the following sections to evaluate high-accuracy, cost-effective, and resource-constrained setups.

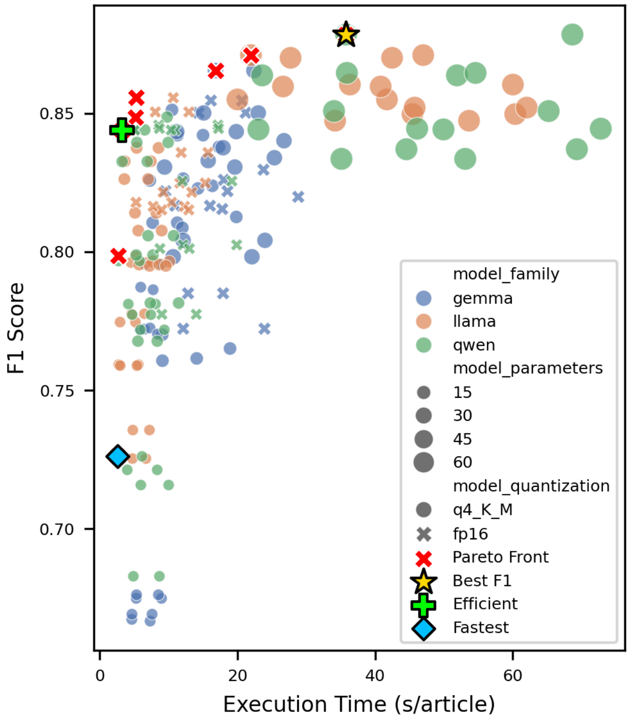

Fig. 5. Multi-objective Pareto-front analysis of all evaluated configurations, showing the trade-off between F1 score and execution time per article. Pareto-optimal configurations are marked with red crosses, and three representative setups, *Best-F1*, *Efficient*, and *Fastest*, are highlighted.

E. Drought Impact Extraction Performance

We evaluate the three selected configurations on drought impact extraction using the DID test split. Overall, *Best-F1* achieves the strongest results (F1 = 0.873, recall = 0.911, and precision = 0.837), but at the cost of long runtimes (over 36 s/article). *Fastest* processes articles in just 2.47 s, but its performance drops (F1 = 0.646), particularly due to poor recall (0.538). *Efficient* reaches a competitive performance (F1 = 0.808) with much faster inference (3.47 s/article), offering a solid balance between accuracy and efficiency. None of the three configurations exhibited parsing errors. These results are summarized in Table IV.

Breaking down performance by impact category (see Table V) shows that *Best-F1* leads on Agriculture (0.868), Livestock (0.966), and Hydrological Resources (0.824), while *Efficient* achieves the best score for Energy (0.970). *Fastest* underperforms across categories confirming the limitations of smaller models in capturing complex or implicit impacts.

To ensure a fair comparison with SeqIA [36], we use the E2E dataset for evaluation, which was specifically created for this purpose [67]. DID is excluded from this comparison to avoid training overlap, as specified on Section II-A. CienaLLM's *Best-F1* configuration surpasses SeqIA (F1 = 0.782 vs. 0.769) but at $\sim$ 20×higher per-article latency (30.07 s vs. 1.54 s/article). The *Efficient* and *Fastest* profiles approach

SeqIA's runtime (2.54 s and 1.81 s/article) but with lower performance (F1 = 0.645 and 0.452, respectively). These results are presented in Table VI.

F. Drought Relevance Classification Performance

Relevance classification results of CienaLLM's configuration and SeqIA on the DRD test split are summarized in Table VII. The *Best-F1* configuration achieved the strongest overall performance (F1 = 0.968), while maintaining near-perfect parsing reliability (0.6%). However, it was also the slowest, requiring 11.28 s per article. The *Efficient* setup offered a strong trade-off, with F1 = 0.929, balanced precision (0.968) and recall (0.894), and perfect parsing reliability, at a much faster 1.74 s per article. The *Fastest* profile processed articles in 1.36 s, but its accuracy degraded markedly (F1 = 0.779), especially due to lower recall (0.698), and its parsing error rate rose to 25%. Compared to SeqIA (F1 = 0.961, execution time = 0.18 s/article), *Best-F1* achieved nearly identical performance, but with higher latency ($\sim 60\times$ faster). *Efficient* remained competitive at $\sim 10\times$ SeqIA's latency, while *Fastest* underperformed both in speed ($\sim 8\times$ slower) and accuracy. These results show that while CienaLLM can match or slightly surpass SeqIA in accuracy, efficiency remains a critical limitation.

G. Drought Impact Location Extraction Performance

Table VIII summarizes the performance of the three selected configurations on drought impact location extraction. SeqIA results are not included here, as the system extracts all mentioned toponyms, whereas CienaLLM was prompted to extract only the provinces affected by drought, making the tasks not directly comparable.

Best-F1 achieved the strongest results, though performance remained modest (F1 = 0.465), a notable parsing error rate (2%), and high inference cost (22.39 s/article). The *Efficient* setup reached lower F1 (0.233) and suffered from unreliable parsing error rate (50%). The *Fastest* profile, while processing articles in just 1.37 s/article, produced an almost negligible F1 of 0.007. Across all configurations, precision remained moderate (~ 0.75), but recall was consistently poor (0.34–0.00), along with parsing issues. Overall, these results underscore both the difficulty of location extraction and the substantial performance gap across model scales.

VI. Discussion

Structured extraction of climate impacts from news articles supports effective monitoring and understanding of the socioeconomic consequences of extreme events. While supervised systems can achieve strong in-domain accuracy [28], [36], they require costly annotation and retraining whenever new tasks, hazards, or languages are introduced. Previous explorations of generative approaches have remained narrow, relying on sources like Wikipedia that are far less ambiguous and heterogeneous than news articles [54]. To our knowledge, no prior work has systematically evaluated open-weight LLMs for climate-impact extraction from news, nor compared across model families, sizes, precision regimes, and prompting strategies. Our study addresses this gap through the development of CienaLLM, a schema-guided GenIE framework that uses zero-shot prompting to deliver competitive accuracy without retraining, and flexibly adapts to evolving schemas. These features position CienaLLM as a practical and adaptable alternative for transforming news into structured datasets that enable systematic climate-risk monitoring and adaptation planning.

Building on this framework, our experiments highlight several factors that critically affect extraction performance and reliability. Response parsing proved essential, since without it some models often produced trivial JSON defects that compromised cross-model comparability. These effects are consistent with findings [62] that show that even recent LLMs struggle to generate valid JSON outputs across diverse schemas. An additional parsing step almost entirely eliminated such errors without reducing accuracy, so findings reported in the parsing-enabled setting remain valid in a no-parsing regime. Although this added latency, ensuring near-perfect parsability was essential; otherwise, valuable information from news articles would be lost.

We find a clear temporal trend in model reliability, as newer LLM releases generate valid JSON far more consistently. This improvement appears linked to recent training practices that emphasize structured output for tool use [43], [92] together with the growing number of benchmarks that encourage schema compliance [49], [93]. In contrast, older models remain far less reliable (see release dates in Table III). We found no evidence that parsing success varied by article characteristics; instead, it depends primarily on each model's ability to follow output instructions. As structured response methods continue to advance, the reliance on explicit response parsing may diminish, but remains indispensable for fair and reliable evaluation in our experiments.

As expected, one of the strongest patterns in our results is that larger models consistently outperform smaller ones, not only in peak accuracy but also in stability and robustness. Larger LLMs show much lower variance across prompt configurations, handling implicit or subtle mentions of impact more reliably. Smaller models, by contrast, are more sensitive to prompt design and show higher variance, having some prompt configurations approach the performance of their larger counterparts. Parameter count closely tracks F1 score and per-article runtime, while model family plays a comparatively minor role.

Precision strongly shapes efficiency, since in our experiments quantization reduced latency and memory demands substantially while incurring only modest accuracy penalties. Similar findings have been reported in broader evaluations of quantization techniques [94]. This balance makes quantized models especially attractive for deployment in settings with limited hardware resources. Additionally, studies on quantization [95] show that larger models are more resilient to reductions in precision: even when quantized, many high-parameter models show only modest decreases in accuracy relative to full-precision versions. While our setup used one

TABLE IV
DROUGHT IMPACT EXTRACTION PERFORMANCE ON THE TEST SPLIT OF THE DID DATASET FOR THE THREE REPRESENTATIVE CONFIGURATIONS.

Configuration	Accuracy	Precision	Recall	F1 Score	Parsing Error Rate	Exec. Time (s/article)
Best-F1	0.684	0.837	0.911	0.873	0.000	36.432
Fastest	0.436	0.810	0.538	0.646	0.000	2.468
Efficient	0.598	0.832	0.785	0.808	0.000	3.320

TABLE V
PER-IMPACT DROUGHT IMPACT EXTRACTION PERFORMANCE ON THE TEST SPLIT OF THE DID DATASET FOR THE THREE REPRESENTATIVE CONFIGURATIONS.

Configuration	Agriculture	Livestock	Hydrological Resources	Energy
Best-F1	0.868	0.966	0.824	0.914
Fastest	0.634	0.542	0.674	0.737
Efficient	0.780	0.893	0.746	0.970

TABLE VI
DROUGHT IMPACT EXTRACTION PERFORMANCE ON THE E2E DATASET FOR CIENALLM'S THREE REPRESENTATIVE CONFIGURATION AND SEQIA.

Configuration	Accuracy	Precision	Recall	F1 Score	Parsing Error Rate	Exec. Time (s/article)
Best-F1	0.742	0.737	0.832	0.782	0.000	30.070
Fastest	0.634	0.769	0.320	0.452	0.000	1.811
Efficient	0.686	0.791	0.544	0.645	0.000	2.540
SeqIA	0.788	0.735	0.806	0.769	0.000	1.540

TABLE VII
DROUGHT RELEVANCE CLASSIFICATION PERFORMANCE ON THE TEST SPLIT OF THE DRD DATASET FOR CIENALLM'S THREE REPRESENTATIVE CONFIGURATIONS AND SEQIA.

Configuration	Accuracy	Precision	Recall	F1 Score	Parsing Error Rate	Exec. Time (s/article)
Best-F1	0.965	0.967	0.970	0.968	0.006	11.282
Efficient	0.925	0.968	0.894	0.929	0.000	1.738
Fastest	0.792	0.880	0.698	0.779	0.249	1.355
SeqIA	0.957	0.957	0.965	0.961	0.000	0.180

TABLE VIII
DROUGHT IMPACT LOCATION EXTRACTION PERFORMANCE ON THE DIDL DATASET FOR CIENALLM'S THREE REPRESENTATIVE CONFIGURATIONS.

Configuration	Accuracy	Precision	Recall	F1 Score	Parsing Error Rate	Exec. Time (s/article)
Best-F1	0.540	0.734	0.340	0.465	0.020	22.389
Efficient	0.400	0.762	0.138	0.233	0.500	1.845
Fastest	0.330	0.750	0.004	0.007	0.620	1.375

specific quantization regime, studies [96], [97] present alternative schemes that offer different trade-offs between speed, memory usage, and precision.

Since this study's experiments, improved model releases have become available. For example, Qwen 3[15] and Llama 4[16], released in April 2025. CienaLLM's modular design allows updated models and quantization techniques to be integrated via simple configuration changes. Following empirical scaling laws [98], which show performance scaling with model size, dataset size, and compute, we expect that overall performance of our methodology, in terms of accuracy, robustness, and efficiency, should continue to improve as newer models evolve.

Prompt interventions showed no universal recipe for performance gains, but they remain a valuable lever for adaptation. On average, their effects were modest yet highly dependent on model family and scale. Some models benefited from condensed inputs or richer label descriptions, consistent with previous findings [99], whereas others relied on full context or ignored additional guidance. These results suggest that prompt design is best seen as a model- and task-specific hyperparameter. It is not a guaranteed accuracy booster, but rather a practical way to extract more reliability from limited models [100]. Importantly, this means that when resources are constrained, smaller models can approximate the performance of larger ones through carefully tuned prompt engineering, offering a flexible path to balance cost and accuracy.

Across impact extraction, relevance classification, and location identification, our results demonstrate that schema-guided GenIE with open-weight LLMs is a viable alternative to supervised baselines like SeqIA [36]. While SeqIA continues to dominate in throughput and efficiency, CienaLLM achieves competitive or superior accuracy for impact extraction and

[15] https://qwenlm.github.io/blog/qwen3/

[16] https://www.llama.com/models/llama-4/

relevance classification, while uniquely supporting schema-flexible location extraction. Additionally, performance scales predictably with LLM generation, meaning that CienaLLM can seamlessly integrate future model releases to improve accuracy, stability, and efficiency without retraining. This adaptability positions CienaLLM and the GenIE approach as a practical alternative for climate-impact extraction.

On the drought impact extraction task, CienaLLM's selected configurations achieve strong results on the DID dataset, but performance drops by 10–20 points on the E2E dataset [67], even though the performance for *Best-F1* and SeqIA is still matched. This gap is explained by differences between the datasets: DID contains only drought-related news and was designed specifically for impact extraction, whereas E2E evaluates the full pipeline, including drought relevance detection and therefore includes drought-unrelated articles. Internal validations restricted to drought-related E2E articles confirm that CienaLLM performs better in that setting, although still below DID levels. Beyond this, differences in dataset composition might contribute further to the gap: E2E exhibits broader thematic coverage, fewer impacts per article, and more imbalanced impact labels. Overall, DID is the more suitable benchmark for assessing impact extraction, while E2E remains the only option for a fair comparison with SeqIA.

For detecting drought-relevant articles, CienaLLM reaches accuracy comparable to the supervised SeqIA classifier, but at a much higher inference cost, making it unsustainable for large-scale preprocessing. In such scenarios, faster approaches are preferable. Supervised classifiers, when annotated data is available, provide reliable filtering at low latency, while keyword matching offers a simple and inexpensive alternative when labeled datasets are lacking, albeit at lower accuracy [36]. For new events without an existing classifier, keyword filtering may be the most practical option. A hybrid strategy, which uses lightweight supervised or keyword-based filtering to pre-select relevant articles and then applies CienaLLM for detailed impact extraction, offers a balanced trade-off between efficiency, flexibility, and accuracy.

The difficulty of impacted location extraction reflects a broader challenge in climate information extraction, namely the need to move from unstructured mentions in news articles to precise geographic entities. Previous work [36] has often relied on named entity recognition (NER) to identify place names, followed by post-processing or gazetteer matching to disambiguate and geolocate them. Yet, ambiguity persists: When an article mentions a river, does the impact extend to all provinces it crosses? When it refers to "the northeast of the peninsula," how should that region be delineated? More fundamentally, translating text into mappable units requires choosing an appropriate spatial scale. Drought typically affects broad areas, which we operationalize at the provincial level, but other hazards may demand finer (e.g., municipal) resolutions. Defining these boundaries is inherently complex, as climate impacts rarely align with administrative divisions.

Our results highlight this challenge: high precision but low recall indicates that models are cautious about overgeneralization, yet often fail to capture implicitly referenced locations. By extracting impacted rather than merely mentioned locations, CienaLLM produces spatially specific and decision-relevant signals that SeqIA does not target, while its schema-guided design remains adaptable to different spatial granularity required for other hazards.

These insights point toward several practical recommendations for deploying such systems in real-world pipelines:

- **Response parsing**: Enable only if models produce frequent formatting errors, and skip in newer LLM generations that emit valid JSON consistently. The role of response parsing will decline as model reliability improves.
- **Model size and precision**: Choose accordingly to latency and hardware budgets. Use quantized small/mid-size models for routine large-scale deployments, full precision only for critical use.
- **Model recency**: Favor latest LLM checkpoints, which consistently improve stability and accuracy.
- **Prompt design**: Adapt to capacity using shorter prompts for small models, and rich descriptions and additional guidance for larger ones.
- **Operational evaluation**: Where feasible, run a small-scale test with labeled data across candidate models to estimate the runtime–accuracy balance before scaling up.
- **Pipeline integration**: Combine supervised classifiers or alternative methods for fast relevance filtering with GenIE approaches for schema-flexible information extraction tasks like impacts and locations.
- **Geo-resolution**: Complement LLM predictions with deterministic geo-resolution methods (gazetteers, basin mapping, or province normalization) to mitigate recall issues and enforce spatial consistency.
- **Sustainability**: Prefer open-weight, small and quantized models whenever possible to minimize compute, memory, and energy footprints in line with sustainability principles [101], [102].

Because CienaLLM is zero-shot rather than label-trained, its generalization prospects are strong. Extending from drought to floods, hail, or heatwaves requires only defining category descriptions, not relabeling corpora and retraining. The same flexibility applies across languages: prompts and schemas can be localized without retraining, enabling rapid adaptation to multilingual or regional outlets. Moderate domain shifts, such as evolving reporting styles or new terminology, can be addressed with lightweight monitoring and occasional audits, rather than expensive re-annotation campaigns. Our methodology's design decouples extraction logic from task-specific definition, making the framework portable across hazards, geographies, and languages.

To support reproducibility, we provide the full codebase, including the framework source code, configurations, and automation scripts. Although the datasets used in this study are not yet public, they will be released with a future peer-reviewed paper. Regarding validity, several steps were taken to ensure consistency: temperatures were fixed to zero and seeds set in the LLM backend to minimize stochasticity, and outputs that failed to parse were excluded to avoid formatting issues with extraction quality. Our evaluation is also centered on Spanish media, and transferability to other languages, outlets, or time periods may shift model priors and should therefore be assessed.

The study has some limitations. The experiments focused exclusively on droughts, leaving other hazards for future evaluation. Comparisons with SeqIA were constrained to the E2E dataset, given training overlaps. Statistical testing across configurations was limited, and more rigorous analyses could strengthen interpretation of performance differences. Finally, infrastructure benchmarks covered only a subset of consumer and HPC environments; a broader hardware study would provide a more complete picture of deployment costs.

VII. Conclusions

Extreme weather events, and droughts in particular, pose complex challenges that cannot be fully captured by physical indices alone. Understanding and monitoring their socioeconomic consequences requires systematic approaches that integrate heterogeneous reportage into structured, decision-relevant signals.

In this work we introduced CienaLLM, a modular framework for schema-guided Generative Information Extraction (GenIE) using open-weight LLMs. Through a large-scale factorial evaluation spanning 384 configurations of model families, sizes, quantization regimes, and prompting strategies, we showed that CienaLLM can reliably extract drought impacts, classify relevance, and identify affected locations from Spanish news. Compared to the supervised SeqIA baseline [36], CienaLLM delivers competitive or superior accuracy in several tasks, while offering unique advantages in schema-flexible extraction. Our analysis highlights key determinants of performance: larger models provide the most accurate and stable results, quantization delivers large efficiency gains with modest trade-offs, and prompt strategies act as tunable hyperparameters whose benefits depend on model family and scale. Location extraction remains the most difficult task, underscoring the need for further advances in handling implicit geographic references.

Beyond task-level findings, the broader significance of CienaLLM lies in its generalizability. Because the system is schema-guided rather than label-trained, extending the framework to new hazards (e.g., floods, hail, heatwaves), new impacts, or new geographic domains requires only adapting prompts and schema definitions, not relabeling corpora or retraining models. More generally, the same method can be applied outside the climate domain to extract other forms of structured information from heterogeneous text, underscoring the portability of the approach. Our design enables near–real-time generation of indicators from national-scale news. In doing so, CienaLLM bridges physical drought indices with the socio-economic consequences documented in media and provides decision-makers with timely evidence for adaptation planning.

Looking ahead, several directions remain open: extending evaluation to additional hazards and languages, developing ensemble strategies that combine complementary model families and sizes, and experimenting with alternative prompting paradigms such as few-shot or retrieval-augmented generation that might improve performance. Future work should also emphasize turning extracted outputs into actionable insights, tracing the evolution of drought impacts across sectors and regions.

Taken together, our results show that schema-guided GenIE with open-weight LLMs is a practical way to turn unstructured news into structured, decision-ready evidence of climate impacts. CienaLLM complements supervised pipelines rather than replacing them, offering schema flexibility, transparent configurations, and reproducible evaluation. Because the framework is portable across models, hazards, and languages, improvements in LLMs translate directly into better extraction without retraining. By releasing code and documenting the full experimental setup, we aim to make these gains cumulative and comparable. In short, CienaLLM helps move from scattered reportage to consistent indicators, a necessary step toward adaptive, data-driven climate-risk monitoring.

VIII. Acknowledgments

This work has been supported by the research projects TED2021-129152B-C42 and PID2022-137244OB-I00, financed by the Spanish Ministry of Science and FEDER, and CSIC's Interdisciplinary Thematic Platform Clima (PTI-Clima).

This research project was made possible through the access granted by the Galician Supercomputing Center (CESGA) to its supercomputing infrastructure. The supercomputer FinisTerrae III and its permanent data storage system have been funded by the NextGeneration EU 2021 Recovery, Transformation and Resilience Plan, ICT2021-006904, and also from the Pluriregional Operational Programme of Spain 2014-2020 of the European Regional Development Fund (ERDF), ICTS-2019-02-CESGA-3, and from the State Programme for the Promotion of Scientific and Technical Research of Excellence of the State Plan for Scientific and Technical Research and Innovation 2013-2016 State subprogramme for scientific and technical infrastructures and equipment of ERDF, CESG15-DE-3114.

References

[1] W. M. O. (WMO), "WMO Atlas of Mortality and Economic Losses from Weather, Climate and Water Extremes (1970–2019)," WMO, Geneva, Tech. Rep., 2021.

[2] M. Peña-Gallardo, S. M. Vicente-Serrano *et al.*, "The impact of drought on the productivity of two rainfed crops in Spain," *Natural Hazards and Earth System Sciences*, vol. 19, no. 6, pp. 1215–1234, Jun. 2019.

[3] M. T. H. Van Vliet, J. Sheffield *et al.*, "Impacts of recent drought and warm years on water resources and electricity supply worldwide," *Environmental Research Letters*, vol. 11, no. 12, p. 124021, Dec. 2016.

[4] X. Zhao, G. Huang *et al.*, "Responses of hydroelectricity generation to streamflow drought under climate change," *Renewable and Sustainable Energy Reviews*, vol. 174, p. 113141, Mar. 2023.

[5] Intergovernmental Panel On Climate Change (Ipcc), *Climate Change 2022 – Impacts, Adaptation and Vulnerability: Working Group II Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, 1st ed. Cambridge University Press, Jun. 2023.

[6] S. M. Vicente-Serrano, D. Peña-Angulo *et al.*, "Global drought trends and future projections," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 380, no. 2238, p. 20210285, Oct. 2022.

[7] S. M. Vicente-Serrano, T. R. McVicar *et al.*, "Unraveling the influence of atmospheric evaporative demand on drought and its response to climate change," *WIREs Climate Change*, vol. 11, no. 2, p. e632, 2020.

[8] M. D. Svoboda, B. A. Fuchs *et al.*, *Handbook of Drought Indicators and Indices*. World Meteorological Organization Geneva, Switzerland, 2016, vol. 2.

[9] S. Bachmair, C. Svensson *et al.*, "A quantitative analysis to objectively appraise drought indicators and model drought impacts," *Hydrology and Earth System Sciences*, vol. 20, no. 7, pp. 2589–2609, 2016.

[10] D. A. Wilhite, "Chapter 1 Drought as a Natural Hazard: Concepts and Definitions," in *Drought: A Global Assessment*, 2000.

[11] K. Stahl, I. Kohn *et al.*, "Impacts of European drought events: Insights from an international database of text-based reports," *Natural Hazards and Earth System Sciences*, vol. 16, no. 3, pp. 801–819, Mar. 2016.

[12] M. T. Boykoff and J. M. Boykoff, "Climate change and journalistic norms: A case-study of US mass-media coverage," *Geoforum*, vol. 38, no. 6, pp. 1190–1204, Nov. 2007.

[13] P. O'Connor, C. Murphy *et al.*, "Relating drought indices to impacts reported in newspaper articles," *International Journal of Climatology*, vol. 43, no. 4, pp. 1796–1816, Mar. 2023.

[14] J. Eva, C. Arlene *et al.*, "Irish Drought Impacts Database v.1.0 (IDID)," Oct. 2022.

[15] K. Dow, "News coverage of drought impacts and vulnerability in the US Carolinas, 1998–2007," *Natural Hazards*, vol. 54, no. 2, pp. 497–518, Aug. 2010.

[16] S. Bell, "The driest continent and the greediest water company: Newspaper reporting of drought in Sydney and London," *International Journal of Environmental Studies*, vol. 66, no. 5, pp. 581–589, Oct. 2009.

[17] A. Hurlimann and S. Dolnicar, "Newspaper coverage of water issues in Australia," *Water Research*, vol. 46, no. 19, pp. 6497–6507, Dec. 2012.

[18] S. Rutledge-Prior and R. Beggs, "Of droughts and fleeting rains: Drought, agriculture and media discourse in Australia†," *Australian Journal of Politics & History*, vol. 67, no. 1, pp. 106–129, Mar. 2021.

[19] C. Dayrell, C. Svensson *et al.*, "Representation of Drought Events in the United Kingdom: Contrasting 200 years of News Texts and Rainfall Records," *Frontiers in Environmental Science*, vol. 10, p. 760147, Mar. 2022.

[20] M. C. Llasat, M. Llasat-Botija *et al.*, "An analysis of the evolution of hydrometeorological extremes in newspapers: The case of Catalonia, 1982–2006," *Natural Hazards and Earth System Sciences*, vol. 9, no. 4, pp. 1201–1212, Jul. 2009.

[21] J. D. Ruiz Sinoga and T. León Gross, "Droughts and their social perception in the mass media (southern Spain)," *International Journal of Climatology*, vol. 33, no. 3, pp. 709–724, Mar. 2013.

[22] J. L. Leidner, "Toponym resolution in text: Annotation, evaluation and applications of spatial grounding," *ACM SIGIR Forum*, vol. 41, no. 2, pp. 124–126, Dec. 2007.

[23] G. DeLozier, B. Wing *et al.*, "Creating a Novel Geolocation Corpus from Historical Texts," in *Proceedings of the 10th Linguistic Annotation Workshop Held in Conjunction with ACL 2016 (LAW-X 2016)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 188–198.

[24] S. Grunewald and R. Mostern, "Working with Named Places: How and Why to Build a Gazetteer," *Programming Historian*, no. 13, Mar. 2024.

[25] A. Vaswani, N. Shazeer *et al.*, "Attention Is All You Need," Aug. 2023.

[26] J. Devlin, M.-W. Chang *et al.*, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 2019.

[27] Y. Liu, M. Ott *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," Jul. 2019.

[28] J. Sodoge, C. Kuhlicke, and M. M. de Brito, "Automatized spatio-temporal detection of drought impacts from newspaper articles using natural language processing and machine learning," *Weather and Climate Extremes*, vol. 41, p. 100574, Sep. 2023.

[29] J. Pita Costa, L. Rei *et al.*, "Towards improved knowledge about water-related extremes based on news media information captured using artificial intelligence," *International Journal of Disaster Risk Reduction*, vol. 100, p. 104172, Jan. 2024.

[30] S. Duarte, G. A. Corzo Perez *et al.*, "Application of Natural Language Processing to Identify Extreme Hydrometeorological Events in Digital News Media: Case of the Magdalena River Basin, Colombia," in *Special Publications*, 1st ed., G. A. Corzo Perez and D. P. Solomatine, Eds. Wiley, Feb. 2024, pp. 283–318.

[31] B. Zhang, F. Schilder *et al.*, "TweetDrought: A Deep-Learning Drought Impacts Recognizer based on Twitter Data," Dec. 2022.

[32] J. Domala, M. Dogra *et al.*, "Automated Identification of Disaster News for Crisis Management using Machine Learning and Natural Language Processing," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. Coimbatore, India: IEEE, Jul. 2020, pp. 503–508.

[33] L. Zou, Z. He *et al.*, "Multi-class multi-label classification of social media texts for typhoon damage assessment: A two-stage model fully integrating the outputs of the hidden layers of BERT," *International Journal of Digital Earth*, vol. 17, no. 1, p. 2348668, Dec. 2024.

[34] K. Lai, J. R. Porter *et al.*, "A Natural Language Processing Approach to Understanding Context in the Extraction and GeoCoding of Historical Floods, Storms, and Adaptation Measures," *Information Processing & Management*, vol. 59, no. 1, p. 102735, Jan. 2022.

[35] H. Otudi, S. Gupta *et al.*, "Classifying Severe Weather Events by Utilizing Social Sensor Data and Social Network Analysis," in *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*. Kusadasi Turkiye: ACM, Nov. 2023, pp. 64–71.

[36] M. López-Otal, F. Domínguez-Castro *et al.*, "SeqIA: A Python framework for extracting drought impacts from news archives," *Environmental Modelling & Software*, vol. 187, p. 106382, Apr. 2025.

[37] OpenAI, J. Achiam *et al.*, "GPT-4 Technical Report," Mar. 2024.

[38] H. Touvron, T. Lavril *et al.*, "LLaMA: Open and Efficient Foundation Language Models," Feb. 2023.

[39] T. B. Brown, B. Mann *et al.*, "Language Models are Few-Shot Learners," Jul. 2020.

[40] S. Bubeck, V. Chandrasekaran *et al.*, "Sparks of Artificial General Intelligence: Early experiments with GPT-4," Apr. 2023.

[41] D. Xu, W. Chen *et al.*, "Large language models for generative information extraction: A survey," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186357, Dec. 2024.

[42] F. Bai, J. Kang *et al.*, "Schema-Driven Information Extraction from Heterogeneous Tables," 2023.

[43] D. Y.-B. Wang, Z. Shen *et al.*, "SLOT: Structuring the Output of Large Language Models," May 2025.

[44] M. Josifoski, N. De Cao *et al.*, "GenIE: Generative Information Extraction," in *NAACL 2022: THE 2022 CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES*. Assoc Computat Linguist, N Amer Chapter; Amazon Sci; Bloomberg Engn; Google Res; LivepersoMetan; ByteDance; KENSH; Grammarly; Megagon Labs; Microsoft; Reveal Brainspace; Cohere; GResearch; Relativity; Servicenow; ASAPP; Duolingo; Adobe; Linkedin; Babelscape; Rakuten Inst Technol; UC Santa Cruz, Baskin Engn; Nat Language Proc; NSF; ETS; OpenAI; TIAA; Two Sigma; Mag Data, 2022, pp. 4626–4643.

[45] L. Rettenberger, M. F. Münker *et al.*, "Using Large Language Models for Extracting Structured Information From Scientific Texts," *Current Directions in Biomedical Engineering*, vol. 10, no. 4, pp. 526–529, Dec. 2024.

[46] X. Wei, X. Cui *et al.*, "ChatIE: Zero-Shot Information Extraction via Chatting with ChatGPT," May 2024.

[47] F. Shiri, V. Nguyen *et al.*, "Decompose, Enrich, and Extract! Schema-aware Event Extraction using LLMs," Jun. 2024.

[48] N. Popovič, A. Kangen *et al.*, "DocIE@XLLM25: In-Context Learning for Information Extraction using Fully Synthetic Demonstrations," Jul. 2025.

[49] S. Geng, H. Cooper *et al.*, "JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models," Feb. 2025.

[50] J. Dagdelen, A. Dunn *et al.*, "Structured information extraction from scientific text with large language models," *Nature Communications*, vol. 15, no. 1, p. 1418, Feb. 2024.

[51] J. Liu, J. Wang *et al.*, "Improving LLM-Based Health Information Extraction with In-Context Learning," in *Health Information Processing.*

Evaluation Track Papers, H. Xu, Q. Chen *et al.*, Eds. Singapore: Springer Nature Singapore, 2024, vol. 2080, pp. 49–59.

[52] I. C. Wiest, F. Wolf *et al.*, "LLM-AIx: An open source pipeline for Information Extraction from unstructured medical text based on privacy preserving Large Language Models," Sep. 2024.

[53] E. Hsu and K. Roberts, "LLM-IE: A python package for biomedical generative information extraction with large language models," *JAMIA Open*, vol. 8, no. 2, p. ooaf012, Mar. 2025.

[54] N. Li, S. Zahra *et al.*, "Using LLMs to Build a Database of Climate Extreme Impacts," in *Proceedings of the 1st Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2024)*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 93–110.

[55] D. Delforge, V. Wathelet *et al.*, "EM-DAT: The Emergency Events Database," Dec. 2023.

[56] G. Team, T. Mesnard *et al.*, "Gemma: Open Models Based on Gemini Research and Technology," Apr. 2024.

[57] Qwen, A. Yang *et al.*, "Qwen2.5 Technical Report," Jan. 2025.

[58] G. Team, R. Anil *et al.*, "Gemini: A Family of Highly Capable Multimodal Models," May 2025.

[59] X. Zhu, J. Li *et al.*, "A Survey on Model Compression for Large Language Models," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 1556–1577, Nov. 2024.

[60] T. Dettmers, A. Pagnoni *et al.*, "QLoRA: Efficient Finetuning of Quantized LLMs," May 2023.

[61] S. Schulhoff, M. Ilie *et al.*, "The Prompt Report: A Systematic Survey of Prompting Techniques," Jul. 2024.

[62] Y. Lu, H. Li *et al.*, "Learning to Generate Structured Output with Schema Reinforcement Learning," Mar. 2025.

[63] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring How Models Mimic Human Falsehoods," May 2022.

[64] Z. Ji, N. Lee *et al.*, "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, Dec. 2023.

[65] M. T. Ribeiro, T. Wu *et al.*, "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai *et al.*, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 4902–4912.

[66] K. Zhu, Q. Zhao *et al.*, "PromptBench: A Unified Library for Evaluation of Large Language Models," Aug. 2024.

[67] M. Lopez Otal, F. Domínguez-Castro *et al.*, "SeqIA - Annotated drought-related news articles," 2025.

[68] J. R. Landis and G. G. Koch, "The Measurement of Observer Agreement for Categorical Data," *Biometrics*, vol. 33, no. 1, p. 159, Mar. 1977.

[69] C. Salvador, R. Nieto *et al.*, "Quantification of the Effects of Droughts on Daily Mortality in Spain at Different Timescales at Regional and National Levels: A Meta-Analysis," *International Journal of Environmental Research and Public Health*, vol. 17, no. 17, p. 6114, Jan. 2020.

[70] Z. Chu, J. Chen *et al.*, "Navigate through Enigmatic Labyrinth A Survey of Chain of Thought Reasoning: Advances, Frontiers and Future," Jun. 2024.

[71] J. White, Q. Fu *et al.*, "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT," Feb. 2023.

[72] J. Fu, S.-K. Ng, and P. Liu, "Polyglot Prompt: Multilingual Multitask Prompt Training," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 9919–9935.

[73] L. Reynolds and K. McDonell, "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. Yokohama Japan: ACM, May 2021, pp. 1–7.

[74] Y. Zhou, A. I. Muresanu *et al.*, "Large Language Models Are Human-Level Prompt Engineers," Mar. 2023.

[75] Y. Li, B. Dong *et al.*, "Compressing Context to Enhance Inference Efficiency of Large Language Models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, 2023, pp. 6342–6353.

[76] D. S. Muthukumar, B. Karthik *et al.*, "A Framework for Analyzing and Summarizing News and Articles using Large Language Model," in *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*. Gobichettipalayam, India: IEEE, Dec. 2024, pp. 1252–1256.

[77] T. Kojima, S. S. Gu *et al.*, "Large Language Models are Zero-Shot Reasoners," Jan. 2023.

[78] A. Madaan, N. Tandon *et al.*, "SELF-REFINE: Iterative Refinement with Self-Feedback," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[79] J. Huang, S. Gu *et al.*, "Large Language Models Can Self-Improve," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, 2023, pp. 1051–1068.

[80] Y. Yang, B. Chen *et al.*, "Multi-step Iterative Automated Domain Modeling with Large Language Models," in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*. Linz Austria: ACM, Sep. 2024, pp. 587–595.

[81] T. Schick, J. Dwivedi-Yu *et al.*, "Toolformer: Language Models Can Teach Themselves to Use Tools," Feb. 2023.

[82] S. He, "Achieving Tool Calling Functionality in LLMs Using Only Prompt Engineering Without Fine-Tuning," Jul. 2024.

[83] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," Oct. 2020.

[84] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, Jul. 2009.

[85] M.-L. Zhang and Z.-H. Zhou, "A Review on Multi-Label Learning Algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[86] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2010, pp. 667–685.

[87] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, p. 80, Dec. 1945.

[88] J. Neyman and E. S. Pearson, "On the Use and Interpretation of Certain Test Criteria for Purposes of Statistical Inference: Part I," *Biometrika*, vol. 20A, no. 1/2, p. 175, Jul. 1928.

[89] R. A. Armstrong, "When to use the B onferroni correction," *Ophthalmic and Physiological Optics*, vol. 34, no. 5, pp. 502–508, Sep. 2014.

[90] T. Fukazawa and T. Miyake, "Pareto front analysis and multi-objective Bayesian optimization for (R, Z)(Fe,Co,Ti)12 (R = Y, Nd, Sm; Z = Zr, Dy)," *Journal of the Physical Society of Japan*, vol. 92, no. 1, p. 014801, Jan. 2023.

[91] Y. Liu and S. V. Kalinin, "The Power of the Pareto Front: Balancing Uncertain Rewards for Adaptive Experimentation in scanning probe microscopy," Apr. 2025.

[92] C. Qu, S. Dai *et al.*, "Tool Learning with Large Language Models: A Survey," *Frontiers of Computer Science*, vol. 19, no. 8, p. 198343, Aug. 2025.

[93] B. Cao, M. Ren *et al.*, "StructEval: Deepen and Broaden Large Language Model Assessment via Structured Evaluation," Aug. 2024.

[94] R. Jin, J. Du *et al.*, "A Comprehensive Evaluation of Quantization Strategies for Large Language Models," Jun. 2024.

[95] S. Badshah and H. Sajjad, "Quantifying the Capabilities of LLMs across Scale and Precision," May 2024.

[96] E. Kurtic, A. Marques *et al.*, ""Give Me BF16 or Give Me Death"? Accuracy-Performance Trade-Offs in LLM Quantization," May 2025.

[97] D. Lee, S. Choi, and I. J. Chang, "Qrazor: Reliable and Effortless 4-bit LLM Quantization by Significant Data Razoring," Feb. 2025.

[98] J. Kaplan, S. McCandlish *et al.*, "Scaling Laws for Neural Language Models," Jan. 2020.

[99] Q. Liu, W. Wang, and J. Willard, "Effects of Prompt Length on Domain-specific Tasks for Large Language Models," Feb. 2025.

[100] P. Sahoo, A. K. Singh *et al.*, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications," Mar. 2025.

[101] R. Schwartz, J. Dodge *et al.*, "Green AI," Aug. 2019.

[102] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," Jun. 2019.

Appendix A
Prompt Templates and Output Schema for Drought Impact Extraction

This appendix contains the complete descriptions and exact templates used in the drought impact extraction task (see Section IV-A). It documents the base prompt, all flag-specific modifications as well as the format instructions and the output schema. All prompts and schemas are assembled dynamically by the CienaLLM framework according to the selected configuration.

- **Base Prompt**: Always included. Introduces the extraction task and inserts the article headline and body text (see Figure A.1)
- **Impact Descriptions**: When enabling `DESC`, natural language definitions of each impact category are inserted into the base prompt to guide interpretation (see Figures A.2, A.3, A.4, A.5, and A.6).
- **Zero-shot Chain-of-Thought**: When enabling `CoT`, appends an instruction to reason step-by-step before answering (see Figure A.7).
- **Article Summarization**: When enabling `SUM`, performs an initial LLM call to summarize the article. The summary replaces the original article text in the main extraction prompt (see Figure A.8).
- **Self-Criticism**: When enabling `SC`, the model's first output is passed to a second LLM call that reviews and corrects it if necessary (see Figure A.9).
- **Response Parsing Strategy**: When enabling `RPARSE`, the initial model output is unstructured text. A second LLM call reformats it into structured JSON using the same impact descriptions and format instructions (see Figure A.10).
- **Format Instructions**: The JSON format instructions (see Figure A.11), derived from the output schema (see Figure A.12), are inserted either directly in the main extraction prompt (`RPARSE = false`) or in the second parsing prompt (`RPARSE = true`).

Appendix B
Prompt Templates and Output Schema for Drought Relevance Classification

This appendix describes the prompt template and output schema used for the **drought relevance classification** task (see Section IV-B). The setup follows the same modular structure, format instructions, and optional response parsing mechanisms described in Appendix A. The base prompt asks whether the article mentions a drought-related event, as shown in Figure B.1. When `RPARSE = true`, a second prompt is used to reformat the initial output into structured JSON, illustrated in Figure B.2. The final output is expected to follow a simplified schema consisting of a single boolean field `"drought"`, shown in Figure B.3.

```
You are an expert in environmental
analysis.  Your task is to analyze the
following news article and determine
whether it reports or  mentions any
impact caused by {event} on specific
aspects.

The aspects to consider are: {impacts}

Please carefully read the article and
determine for each aspect whether there
is a reported impact caused specifically
by {event}. Do not infer impacts unless
they are clearly stated or strongly
implied in the text.

Article to analyze:
{text}
```

Fig. A.1. Base Prompt for Drought Impact Extraction.

```
You are an expert in environmental
analysis. Your task is to analyze the
following news article and determine
whether it reports or mentions any
impact caused by {event} on specific
aspects.

The aspects to consider are: {impacts}

Each aspect is briefly described below to
guide interpretation, but these
definitions are not exhaustive:

{impact_descriptions}

Please carefully read the article and
determine for each aspect whether there
is a reported impact caused specifically
by {event}. Do not infer impacts unless
they are clearly stated or strongly
implied in the text.

Article to analyze:
{text}
```

Fig. A.2. Base Prompt with Descriptions for Drought Impact Extraction.

```
News about the impacts of drought on
agriculture usually refer to losses in
both rainfed and irrigated crops. It is
often mentioned that part of the harvest
has been lost or will be lost.
```

Fig. A.3. Description of Drought Impact "Agriculture".

```
News about the impacts of drought on
livestock usually refer to the loss of
pastures that feed the livestock. In more
extreme droughts, it may be mentioned that
there is no water available to give the
livestock to drink.
```

Fig. A.4. Description of Drought Impact “Livestock”.

```
News about the impacts of drought on
hydrological resources usually mention
the decrease in river flows, reservoir
levels, or groundwater. It is also
mentioned the lack of water that this
causes in the populations, with water
cuts being decreed or water not being
used for certain uses, or the need to
bring water from other locations.
```

Fig. A.5. Description of Drought Impact “Hydrological Resources”.

```
News about the impacts of drought on
energy usually mention that due to the
low flow of rivers or reservoirs, it is
not possible to turbine and therefore
the generation of hydroelectric energy
decreases.
```

Fig. A.6. Description of Drought Impact “Energy”.

```
Reason step by step and explain your
reasoning before giving the final answer.
```

Fig. A.7. Chain-of-Thought Instruction.

```
Summarize the following article.

Text:
{text}
```

Fig. A.8. Summarization Prompt.

```
Given the following prompt:
{prompt}

And the following response:
{response}

Analyze the response and determine
whether it is correct or incorrect. If
it is incorrect, provide a brief
explanation of why it is incorrect and
the correct response.
If it is correct, provide the same correct
response.
```

Fig. A.9. Self-Criticism Prompt.

```
Extract whether the following LLM
response says the article mentions an
impact of {event} on {impacts}.

The impacts are defined as follows:
{impact_descriptions}

Text:
{text}
```

Fig. A.10. Response Parsing Prompt for Drought Impact Extraction.

```
Format instructions:
{format_instructions}
Make sure to include a single JSON in your
response instead of multiple JSONs.
```

Fig. A.11. Response Parsing Format Instruction.

```
{
    "agriculture": <true or false>,
    "livestock": <true or false>,
    "hydrological_resources":
        <true or false>,
    "energy": <true or false>
}
```

Fig. A.12. Output Schema for Drought Impact Extraction.

```
Analyze the following article and
determine if the news article mentions
an event related to {event}.

Text:
{text}
```

Fig. B.1. Base Prompt for Drought Relevance Classification.

```
Extract whether the following LLM
response says the article mentions an
event related to {event}.

Text:
{text}
```

Fig. B.2. Response Parsing Prompt for Drought Relevance Classification.

```
{
    "drought": <true or false>
}
```

Fig. B.3. Output Schema for Drought Relevance Classification.

Appendix C
Prompt Templates and Output Schema for Drought Impact Location Extraction

This appendix describes the prompt template and output schema used for the drought impact location extraction task (see Section IV-C). It builds on the same modular construction and response parsing logic described in Appendix A, with task-specific modifications. The base prompt instructs the model to identify Spanish provinces affected by drought, as shown in Figure C.1. If response parsing is enabled, a second prompt is used to convert the output into a valid JSON structure, as seen in Figure C.2. The expected schema consists of a field "provinces" containing a list of province names, illustrated in Figure C.3.

```
Given a news article describing {event}
impacts in Spanish regions, return the
list of affected provinces.
Identify provinces explicitly mentioned
or infer them from the described
locations.
If you cannot identify specific
provinces, return all provinces in the
regions mentioned.
Note that the text may refer to
autonomous communities or specific
cities, towns, and municipalities,
which are not the provinces being
requested. Do not include these
autonomous communities or
municipalities in the output.
Return the province names in Spanish.

Text:
{text}
```

Fig. C.1. Base Prompt for Drought Impact Location Extraction

```
Extract from the following LLM response
the provinces.
If other locations are mentioned, infer
the provinces.

Text:
{text}
```

Fig. C.2. Response Parsing Prompt for Drought Impact Location Extraction

```
{
    "response": [
        <province>,
        ...
    ]
}
```

Fig. C.3. Output Schema for Drought Impact Location Extraction

Appendix D
Results Tables

This appendix compiles the tables referenced in Section V. Table D.1 reports the overall comparison of average performance with and without response parsing (`RPARSE`), while Table D.2 presents the corresponding model-level metrics. Table D.3 summarizes average performance and variability per LLM across all prompt configurations, and Table D.4 lists the best-performing configuration for each model. Table D.5 compares full-precision and quantized variants across model families. Table D.6 shows average performance by configuration factor, with Tables D.7 and D.8 detailing factor effects at the model and family levels, respectively. Finally, Table D.9 presents the three Pareto-optimal configurations used throughout the efficiency–performance analysis.

For discussion and interpretation of these results, see Section V.

Appendix E
News Corpora

We collected broad news corpora by crawling the full online archives of three major Spanish outlets: El País, ABC, and 20 Minutos. We summarize their coverage in Table E.1. In addition, we incorporate a previously compiled collection of drought-related articles from Grupo Zeta (a media conglomerate now integrated into Prensa Ibérica), originally assembled by López-Otal et al. [36] using a keyword-based search focused on drought terms [36]. Unlike our outlet-wide crawls (which include *all* articles in the specified ranges), the Grupo Zeta collection is *task-focused* (drought-only) and not exhaustive over the full archive period reported in Table E.1.

To reflect distinct access modalities at ABC, we report two entries: (i) the HTML archive crawl and (ii) the digitized historical *hemeroteca* (PDF) crawl. Together with El País and 20 Minutos, these national outlets provide long temporal coverage suitable for retrospective analyses. Crucially, the scale and breadth of these corpora make them well suited for large-scale, longitudinal assessments of climate-related impacts in Spain using our CienaLLM extraction framework.

TABLE D.1

AVERAGE DROUGHT IMPACT EXTRACTION PERFORMANCE WITH AND WITHOUT RPARSE.

RPARSE	F1 Score	Parsing Error Rate	Exec. Time (s/article)
False	0.803 ± 0.061	0.040 ± 0.083	10.175±10.575
True	0.808 ± 0.048	0.007 ± 0.017	16.096±15.044
Δ (p-value)	+0.005 ($p = 1.0$)	–0.033*** ($p = 5.5 \times 10^{-13}$)	+5.921*** ($p = 9.7 \times 10^{-32}$)

Note. Significance stars are used consistently across all tables: * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$.

TABLE D.2

AVERAGE DROUGHT IMPACT EXTRACTION PERFORMANCE PER MODEL WITH AND WITHOUT RPARSE.

LLM	RPARSE	F1 Score	Parsing Error Rate	Exec. Time (s/article)
gemma_4b	False	0.750 ± 0.056	0.014 ± 0.017	4.461 ± 0.200
	True	0.727 ± 0.057	0.000	6.875 ± 0.146
gemma_12b	False	0.796 ± 0.032	0.003 ± 0.004	6.908 ± 0.329
	True	0.811 ± 0.030	0.000	12.902 ± 0.386
gemma_12b_f	False	0.805 ± 0.028	0.000	10.181 ± 0.516
	True	0.818 ± 0.028	0.000	17.688 ± 0.519
gemma_27b	False	0.838 ± 0.023	0.010 ± 0.024	11.411 ± 0.552
	True	0.833 ± 0.022	0.000	18.272 ± 0.552
llama_3b	False	0.776 ± 0.051	0.135 ± 0.139	3.553 ± 0.132
	True	0.766 ± 0.025	0.000	4.939 ± 0.148
llama_8b	False	0.793 ± 0.053	0.186 ± 0.123	5.312 ± 0.202
	True	0.813 ± 0.017	0.044 ± 0.008	7.054 ± 0.201
llama_8b_f	False	0.758 ± 0.146	0.104 ± 0.055	8.415 ± 0.335
	True	0.830 ± 0.015	0.041 ± 0.019	10.762 ± 0.314
llama_70b	False	0.855 ± 0.014	0.021 ± 0.033	27.754±16.993
	True	0.858 ± 0.009	0.000	41.631±13.327
qwen_3b	False	0.773 ± 0.027	0.007 ± 0.012	3.731 ± 0.208
	True	0.747 ± 0.039	0.000	6.202 ± 0.229
qwen_7b	False	0.819 ± 0.020	0.000 ± 0.001	5.165 ± 0.199
	True	0.815 ± 0.030	0.000	7.134 ± 0.247
qwen_7b_f	False	0.826 ± 0.009	0.002 ± 0.003	8.280 ± 0.342
	True	0.823 ± 0.025	0.000	11.971 ± 0.458
qwen_72b	False	0.846 ± 0.015	0.000	26.928±14.112
	True	0.852 ± 0.015	0.000	47.725±15.842

TABLE D.3

AVERAGE DROUGHT IMPACT EXTRACTION PERFORMANCE PER MODEL.

LLM	F1 Score	Parsing Error Rate	Exec. Time (s/article)
gemma_4b	0.727 ± 0.057	0.000	6.875 ± 0.146
gemma_12b	0.811 ± 0.030	0.000	12.902 ± 0.386
gemma_12b_f	0.818 ± 0.028	0.000	17.688 ± 0.519
gemma_27b	0.833 ± 0.022	0.000	18.272 ± 0.552
llama_3b	0.766 ± 0.025	0.000	4.939 ± 0.148
llama_8b	0.813 ± 0.017	0.044 ± 0.008	7.054 ± 0.201
llama_8b_f	0.830 ± 0.015	0.041 ± 0.019	10.762 ± 0.314
llama_70b	0.858 ± 0.009	0.000	41.631 ± 1.333
qwen_3b	0.747 ± 0.039	0.000	6.202 ± 0.229
qwen_7b	0.815 ± 0.030	0.000	7.134 ± 0.247
qwen_7b_f	0.823 ± 0.025	0.000	11.971 ± 0.458
qwen_72b	0.852 ± 0.015	0.000	47.725 ± 1.584

TABLE D.4
DROUGHT IMPACT EXTRACTION PERFORMANCE FOR BEST-PERFORMING CONFIGURATION PER MODEL.

LLM	Prompt Techniques	F1 Score	Parsing Error Rate	Exec. Time (s/article)
gemma_4b	SUM + CoT + DESC	0.796	0.000	6.591
gemma_12b	SUM + DESC	0.851	0.000	10.514
gemma_12b_f	SUM + CoT + DESC	0.855	0.000	16.169
gemma_27b	SUM + CoT + DESC	0.865	0.000	16.816
llama_3b	-	0.798	0.000	2.724
llama_8b	CoT + DESC	0.838	0.048	5.388
llama_8b_f	DESC	0.856	0.026	5.321
llama_70b	DESC	0.871	0.000	21.974
qwen_3b	-	0.797	0.000	2.741
qwen_7b	CoT	0.849	0.000	5.257
qwen_7b_f	CoT + DESC	0.846	0.000	8.589
qwen_72b	CoT + DESC	0.878	0.000	35.793

TABLE D.5
AVERAGE DROUGHT IMPACT EXTRACTION PERFORMANCE PER MODEL FAMILY WITH QUANTIZED AND FULL-PRECISION.

LLM Family	LLM Quantization	F1 Score	Parsing Error Rate	Exec. Time (s/article)
gemma	fp16	0.818 ± 0.027	0.000	17.688 ± 5.028
	q4_K_M	0.811 ± 0.029	0.000	12.902 ± 3.741
	Δ (p-value)	–0.008** $(p = 0.005)$	0.000 $(p = \text{—})$	–4.786*** $(p < 10^{-4})$
llama	fp16	0.830 ± 0.015	0.041 ± 0.019	10.762 ± 3.040
	q4_K_M	0.813 ± 0.016	0.044 ± 0.008	7.054 ± 1.945
	Δ (p-value)	–0.017** $(p = 0.005)$	0.003 $(p = 0.469)$	–3.708*** $(p < 10^{-4})$
qwen	fp16	0.823 ± 0.025	0.000	11.971 ± 4.434
	q4_K_M	0.815 ± 0.029	0.000	7.134 ± 2.393
	Δ (p-value)	–0.008* $(p = 0.010)$	0.000 $(p = \text{—})$	–4.837*** $(p < 10^{-4})$

TABLE D.6
AVERAGE DROUGHT IMPACT EXTRACTION PERFORMANCE PER PROMPT STRATEGY.

Configuration	Value	F1 Score	Parsing Error Rate	Exec. Time (s/article)
SUM	False	0.807 ± 0.055	0.007 ± 0.017	14.032 ± 13.418
	True	0.809 ± 0.040	0.007 ± 0.017	18.161 ± 16.178
	Δ (p-value)	0.002 $(p = 1.0)$	0.000 $(p = 1.0)$	4.129*** $(p < 10^{-16})$
SC	False	0.808 ± 0.048	0.007 ± 0.017	12.202 ± 10.803
	True	0.808 ± 0.048	0.007 ± 0.017	19.990 ± 17.415
	Δ (p-value)	–0.000 $(p = 1.0)$	0.000 $(p = \text{—})$	7.788*** $(p < 10^{-16})$
CoT	False	0.806 ± 0.048	0.006 ± 0.013	14.835 ± 13.575
	True	0.809 ± 0.047	0.009 ± 0.020	17.358 ± 16.212
	Δ (p-value)	0.002 $(p = 0.186)$	0.003* $(p = 0.014)$	2.523*** $(p < 10^{-8})$
DESC	False	0.807 ± 0.039	0.006 ± 0.015	16.178 ± 14.732
	True	0.809 ± 0.055	0.008 ± 0.019	16.015 ± 15.272
	Δ (p-value)	0.002 $(p = 0.510)$	0.002* $(p = 0.013)$	–0.163 $(p = 1.0)$

TABLE D.7
AVERAGE EFFECT SIZE (Δ) ON DROUGHT IMPACT EXTRACTION PERFORMANCE PER MODEL PER PROMPT STRATEGY.

LLM	SUM	SC	CoT	DESC
gemma_4b	0.110** $(p = 0.008)$	0.000 $(p = 0.144)$	0.006* $(p = 0.023)$	0.009 $(p = 0.312)$
gemma_12b	0.040** $(p = 0.008)$	-0.002 $(p = 0.249)$	-0.002 $(p = 0.742)$	0.039** $(p = 0.008)$
gemma_12b_f	0.039** $(p = 0.008)$	0.001 $(p = 0.180)$	0.006 $(p = 0.148)$	0.035** $(p = 0.008)$
gemma_27b	0.028** $(p = 0.008)$	0.000 $(p = 0.180)$	0.010** $(p = 0.008)$	0.029** $(p = 0.008)$
llama_3b	-0.014 $(p = 0.148)$	0.000 $(p = \text{—})$	-0.013^{**} $(p = 0.008)$	-0.042^{**} $(p = 0.008)$
llama_8b	-0.029^{**} $(p = 0.008)$	0.000 $(p = \text{—})$	-0.005 $(p = 0.312)$	0.005 $(p = 0.383)$
llama_8b_f	-0.011 $(p = 0.148)$	0.000 $(p = \text{—})$	0.004 $(p = 0.312)$	0.022** $(p = 0.008)$
llama_70b	-0.012^{**} $(p = 0.008)$	0.000 $(p = \text{—})$	-0.001 (p = 1.000)	0.009* $(p = 0.039)$
qwen_3b	-0.019^{**} $(p = 0.008)$	0.000 $(p = \text{—})$	0.002 $(p = 0.742)$	-0.070^{**} $(p = 0.008)$
qwen_7b	-0.053^{**} $(p = 0.008)$	0.000 $(p = \text{—})$	0.008** $(p = 0.008)$	-0.019^{**} $(p = 0.008)$
qwen_7b_f	-0.043^{**} $(p = 0.008)$	0.000 $(p = \text{—})$	0.013** $(p = 0.008)$	-0.011 $(p = 0.148)$
qwen_72b	-0.014^{*} $(p = 0.039)$	0.000 $(p = \text{—})$	0.001 $(p = 0.742)$	0.021** $(p = 0.008)$

TABLE D.8
Average effect size (Δ) on drought impact extraction performance per model family per prompt strategy.

LLM Family	SUM	SC	CoT	DESC
gemma	0.054^{***} $(p = 0.000)$	0.000 $(p = 0.826)$	0.005^{**} $(p = 0.004)$	0.028^{***} $(p = 0.000)$
llama	-0.016^{***} $(p = 0.000)$	0.000 $(p =$—$)$	-0.004 $(p = 0.080)$	-0.001 $(p = 0.747)$
qwen	-0.032^{***} $(p = 0.000)$	0.000 $(p =$—$)$	0.006^{*} $(p = 0.019)$	-0.020^{**} (p = 0.007)

TABLE D.9
Drought impact extraction performance for top configurations on the Pareto front.

Configuration	LLM	Prompt Techniques	F1 Score	Parse Error Rate	Exec. Time (s/article)
Best-F1	qwen_72b	CoT + DESC	0.878	0.000	35.792
FASTEST	qwen_3b	DESC	0.726	0.000	2.633
Efficient	qwen_7b	—	0.844	0.000	3.243

TABLE E.1
Overview of news articles corpora.

News Outlet	Extraction Method	Date Range	URL	# Articles
El País	All news articles (HTML archive)	1976-05-03 – 2024-08-19	https://elpais.com/archivo/	3,733,758
ABC	All news articles (HTML archive)	2001-01-10 – 2024-11-01	https://www.abc.es/archivo/	3,826,855
ABC (PDF)	All news articles (PDF)	1891-05-10 – 2024-10-18	https://www.abc.es/hemeroteca/	6,435,816
20 Minutos	All news articles (HTML archive)	2005-01-16 – 2024-12-31	https://www.20minutos.es/archivo	2,631,446
Grupo Zeta	Drought keyword search (task-focused)	2004-02-06 – 2022-11-28	*N/A*	52,495