# TYPE-CONSTRAINED TWO-STAGE STRUCTURED DECODING FOR LOW-COST JAPANESE DOCUMENT-LEVEL RELATION EXTRACTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language models (LLMs) are an attractive option for low-cost document-level relation extraction (DocRE) and knowledge graph construction, but one-shot extraction often over-generates plausible yet unsupported triples, leading to low precision. This problem is acute in Japanese DocRE, where relations frequently require cross-sentence inference and LLM in-context learning has been reported to underperform supervised approaches on JacRED **?**. We address this reliability gap with a two-stage pipeline that uses JSON Schema-constrained decoding for syntactically valid structured outputs, separates recall-oriented candidate generation from precision-oriented verification, and applies deterministic relation-specific domain and range filtering using empirically observed (head_type, tail_type) pairs from training data. On 10 JacRED development documents selected by character-length-based stratified sampling, the two-stage method consistently reduces false positives across Gemini Flash family configurations (2.0/2.5/3 preview, with and without thinking), yielding precision gains in every setting and improving micro-F1 up to 0.27 compared to 0.20 for the best baseline on the sample. These results indicate that verification plus lightweight type constraints can make low-cost LLM-based Japanese DocRE substantially more usable without fine-tuning.

## 1 INTRODUCTION

Document-level relation extraction (DocRE) seeks to identify directed semantic relations between entities mentioned anywhere in a document, rather than within a single sentence. This setting is motivated by knowledge graph construction and downstream reasoning, where facts are often expressed across multiple sentences and require integrating dispersed evidence **??**. In DocRED, for example, a substantial fraction of relational facts require multi-sentence evidence, and evaluation protocols emphasize micro-averaged relation-level F1 and evidence annotation to capture document-scale reasoning **?**.

Recent interest in large language models (LLMs) has revived generative approaches to information extraction, where a model is prompted to directly emit structured records such as entities and relation triples **?**. This paradigm is practically appealing: it can avoid task-specific training, unify multiple extraction tasks through prompting, and leverage general-purpose language understanding. However, it also introduces a reliability challenge: generative models can hallucinate, over-generalize, and produce inconsistent structures. In relation extraction, these issues are amplified by the quadratic number of potential entity pairs in a document, which increases the opportunity for spurious relations and makes naive “extract everything” prompting particularly prone to false positives **?**.

The gap between practical convenience and extraction reliability becomes especially salient for Japanese DocRE. JacRED is the first public Japanese Wikipedia-based DocRE benchmark with evidence sentences, containing 2,000 documents and 35 relation types, and it shows that LLM in-context learning performs poorly relative to supervised Japanese DocRE models **?**. For cost-sensitive deployments, users may prefer low-latency, low-cost LLM variants such as “flash” models. Yet these configurations often exhibit a failure mode that is costly in downstream curation: one-shot triple extraction can produce many plausible but unsupported triples, resulting in low precision and noisy knowledge graphs.

This paper targets a concrete practical question left open by prior benchmarking: how can we improve the precision of low-cost, API-based Japanese DocRE without fine-tuning? The challenge is hard for three reasons. First, document-level extraction requires cross-sentence aggregation and robust entity identity handling; both are known to drive errors and ambiguity in DocRED-style datasets **?**. Second, generative extraction must balance completeness with faithfulness, and naive prompting often favors fluent completion rather than strict grounding to textual evidence **?**. Third, even when the underlying relation semantics are correct, unconstrained generation can produce ill-formed outputs (missing fields, inconsistent identifiers), which harms evaluation reliability and makes post-processing brittle.

We propose a two-stage structured extraction pipeline designed around these constraints. Conceptually, we decouple "finding possible relations" from "deciding which relations are supported," and we enforce machine-readable structure throughout. Stage 1 is recall-oriented candidate generation: the model is instructed to output a broad set of candidate triples. Stage 2 is precision-oriented verification: the model re-reads the document and judges candidate triples in batches, rejecting unsupported ones. Both stages use JSON Schema-constrained decoding to ensure syntactic validity of the structured outputs, a technique increasingly studied in structured generation settings **?**. Finally, we apply a deterministic filter based on relation-specific domain and range constraints learned from training data as empirically observed (head_type, tail_type) pairs. This final step injects lightweight schema knowledge without manual ontology engineering.

We evaluate the approach on 10 documents from the JacRED development set selected via character-length-based stratified sampling, enabling a cost-aware yet heterogeneous comparison. We compare a one-shot extraction baseline against the proposed pipeline across Gemini Flash family configurations (Gemini 2.0 Flash, 2.5 Flash, and 3 Flash Preview; with and without thinking). Using micro-averaged precision, recall, and F1 over the sampled documents, we find a consistent reduction of false positives and precision gains across all configurations. Micro-F1 improves up to 0.27 in the best observed settings, while recall remains low overall, suggesting that candidate coverage and entity alignment remain primary bottlenecks.

Contributions: - We demonstrate that decomposing low-cost LLM DocRE into candidate generation and verification under schema-constrained decoding yields consistent false-positive reductions on Japanese JacRED documents. - We show that deterministic, data-driven relation-specific type-pair constraints can further improve precision as a lightweight post-processing layer. - We provide a controlled comparison across multiple Gemini Flash configurations, characterizing how "thinking" interacts with one-shot extraction versus two-stage verification. - We identify recall as the dominant remaining limitation, motivating future work on candidate generation and entity alignment under document-level Japanese discourse phenomena.

Future work may explore improving recall through stronger candidate generation prompts, better handling of relation directionality and entity alignment, and tighter integration of evidence-aware prompting aligned with DocRE evidence annotation practices **??**.

## 2 Related Work

Our work sits at the intersection of (i) document-level relation extraction as a benchmarked NLP task, (ii) LLM-based generative information extraction, and (iii) constrained decoding for structured outputs. We organize related work by these themes and clarify what remains insufficient for the low-cost, Japanese, API-based extraction setting we study.

Document-level relation extraction and its bottlenecks. DocRED established DocRE as a distinct problem from sentence-level relation extraction, emphasizing cross-sentence evidence, multi-hop reasoning requirements, and the importance of evidence-aware evaluation **?**. The broader DocRE literature surveyed by Delaunay et al. highlights three persistent technical bottlenecks that are directly relevant to our setting: the quadratic number of entity-pair candidates per document, sensitivity to coreference and aliasing, and the need for global reasoning mechanisms to connect dispersed evidence **?**. However, most work in this cluster assumes supervised training and focuses on designing neural architectures to score entity pairs. In contrast, our setting is cost-driven and training-free: we treat an LLM API as the extraction engine and ask how to control false positives in a generative pipeline.

Japanese DocRE resources and evaluation context. JacRED addresses a core limitation of prior DocRE research: the scarcity of Japanese document-level corpora with evidence annotations and a well-defined relation schema **?**. The JacRED study shows that LLM in-context learning performs poorly compared to supervised Japanese models, suggesting that naive prompting alone is not competitive. What is missing relative to our question is an analysis of reliability strategies for low-cost LLM extraction when one cannot afford fine-tuning or repeated per-relation querying. Our work complements JacRED by focusing on precision-oriented pipeline design and by quantifying false-positive reduction under multiple low-cost LLM configurations on sampled dev documents.

Generative information extraction with LLMs and reliability strategies. Surveys of LLM-based generative IE identify common failure modes such as hallucination, sensitivity to prompts, and mismatches between natural language generation and strict structured extraction requirements **?**. The survey perspective motivates decomposition and verification patterns, but it does not resolve how such strategies behave in DocRE specifically, where candidate enumeration and relation directionality are central and the long-context, cross-sentence nature of evidence makes verification nontrivial. Our work operationalizes this reliability framing in a DocRE pipeline by explicitly separating candidate generation from verification.

Constrained decoding and structured outputs. Enforcing schema-compliant JSON via constrained decoding has become a practical mechanism for robust downstream parsing. Geng et al. provide a benchmark and study of generating structured outputs, demonstrating that constrained decoding can increase compliance and sometimes improve downstream task accuracy **?**. Yet structured-output studies are typically task-agnostic and focus on coverage and compliance metrics. Our work differs conceptually: we use structured outputs as an enabling interface for a two-stage DocRE procedure (batch verification and deterministic filtering), and we evaluate impact on extraction precision and F1 rather than schema compliance.

Why not supervised joint extraction baselines? End-to-end joint NER+RE models using pretrained encoders can reduce pipeline error propagation in supervised sentence-level settings **?**. These approaches are not directly applicable to our goal because they assume labeled training data and target primarily intra-sentence relations, whereas our focus is document-level extraction using an LLM API without fine-tuning. Moreover, our central concern is controlling false positives under generative extraction; joint neural models typically address this through learned scoring and thresholding rather than explicit verification and type constraints.

In summary, DocRE research provides the task definition and highlights why document-scale relation prediction is hard **??**, Japanese resources such as JacRED establish the evaluation context and motivate the need for practical LLM-based methods **?**, and generative IE plus constrained decoding offer tools for structured generation and reliability **??**. Our contribution is to combine these tools into a low-cost, two-stage Japanese DocRE pipeline and to empirically characterize its effect on false-positive reduction under multiple Gemini Flash configurations.

## 3 Background

Task definition and notation. We consider document-level relation extraction in a triple-extraction formulation. A document d contains entities E = e1, …, eN, where each entity may have multiple mentions. The goal is to predict a set of directed relation triples R$_hat, whereeachtripleis(h, r, t)withheadentityh\in$ E, relation label r ∈ L, and tail entity t ∈ E. This aligns with knowledge graph construction settings in which outputs are consumed as edges between entities.

Evaluation protocol. We evaluate extraction quality by comparing predicted triples against gold triples and aggregating counts across documents. Let TP be the number of predicted triples that match gold triples, FP the number of predicted triples that do not match any gold triple, and FN the number of gold triples not predicted. Micro-averaged precision, recall, and F1 are computed as Precision = TP / (TP + FP), Recall = TP / (TP + FN), and F1 = 2PR / (P + R). This micro-averaged formulation is standard in DocRE evaluation **??**.

Dataset context. JacRED is a Japanese Wikipedia-based DocRE dataset with 2,000 documents, 35 relation types, and sentence-level evidence annotations **?**. It uses an entity type system based on 8 IREX-style types. In this work we use the JacRED development set as the source of evaluation documents, while treating extraction as a generative LLM task rather than training supervised DocRE models.

Generative structured extraction and constrained decoding. In generative IE, an LLM produces an output y conditioned on an input x and a prompt p, which can be summarized as sampling from p(y | x, p) **?**. When y must follow a machine-readable schema such as JSON, constrained decoding restricts token generation so that outputs are syntactically valid under the schema, improving robustness to formatting errors and downstream parsing failures **?**. In this paper, "structured outputs" refers to JSON Schema-constrained decoding as supported in the Gemini API setup used in our implementation.

Assumptions for our setting. We assume that each entity in a document has an associated type label (as in JacRED), enabling type-based constraints during post-processing. We further assume that relation-specific domain and range constraints can be approximated from training data by collecting the set of empirically observed (head$_{t}ype, tail_{t}ype)pairsforeachrelation.Thisassumptiondoesnotrequireahand-builtontology;itonlyrequiresaccesstotrainingannotationstocomputeobservedtypepairs.$

## 4 METHOD

Conceptual overview. Our method is a two-stage LLM pipeline for document-level relation triple extraction designed to address a specific failure mode of one-shot generative DocRE: low precision driven by over-generation of unsupported triples. The pipeline separates the roles of proposing relations and validating them, while enforcing a structured interface to prevent ill-formed outputs. Given a document x, the method produces a candidate set C, then a verified set V, and finally an output set T$_{h}atafterdeterministictypefiltering.$

Stage 1: recall-oriented candidate generation. Stage 1 prompts the LLM to generate a broad set of candidate triples C from document x. The objective in this stage is coverage: the prompt is designed to err on the side of proposing plausible relation instances. To reduce downstream brittleness, we require the output to conform to a predefined JSON schema (EXTRACTION$_{S}CHEMA)viaJSONSchema-constraineddecoding.Thisensuresthateachcandidatetripleisemittedwithrequiredfieldsandconsistentstructure.$

Stage 2: precision-oriented verification with batching. Stage 2 takes as input the document x and the candidate set C and asks the LLM to verify candidates, accepting only those supported by the document. Verification is performed in batches of size 10 candidates to reduce API overhead and cost. Outputs are again constrained to a JSON schema (VERIFICATION$_{S}CHEMA),producingstructuredverificationdecisionsforeachcandidate.Thisstageisexplicitlyint$
$evaluatingcandidatetriplesunderastricteracceptancecriterion.$

Deterministic relation-specific type-pair filtering. To further mitigate residual false positives, we apply a deterministic post-processing filter derived from training data. For each relation label r, we compute a set of allowed type pairs A$_{r}consistingofempiricallyobserved(head_{t}ype,tail_{t}ype)combinationsinthetrainingset.Foranyverifiedtriple(h,r,t)$
A$_{r}.Thisstepencodeslightweightdomainandrangeconstraintsandtargetserrorswherearelationlabelisplausiblelingui$

Implementation choices relevant to interpretation. The pipeline is implemented using the Google GenAI SDK with the Gemini API structured outputs feature. Both stages use schema-constrained decoding, and verification uses fixed batch size 10. We treat the LLM model variant and its "thinking" configuration as experimental factors rather than as algorithmic components of the method.

Design rationale. The two-stage structure follows a reliability pattern emphasized in generative IE discussions: separate generation from validation to control hallucination and over-generation **?**. In DocRE, this addresses both the large candidate space and the tendency of one-shot prompts to output unsupported relations. Schema-constrained decoding provides predictable interfaces between stages, and type-pair constraints provide a low-cost, deterministic safeguard derived from existing annotations.

## 5 EXPERIMENTAL SETUP

Data selection and scope. Experiments are conducted on the JacRED development set **?**. To control API cost while retaining heterogeneity in document length, we select 10 documents using character-length-based stratified sampling. This sampling strategy supports rapid iteration but restricts the statistical generality of the findings to an indicative comparison on a small dev subset.

Task instantiation. For each sampled document, systems output a set of directed relation triples under the JacRED relation schema. Outputs are evaluated against gold triples from the dataset.

Evaluation metrics. We report micro-averaged precision, recall, and F1 aggregated over the 10 documents. These are computed from aggregate TP, FP, and FN counts: Precision = TP / (TP + FP), Recall = TP / (TP + FN), and F1 = 2PR / (P + R). Micro-averaging is standard for DocRE benchmarks **??**.

Compared methods. We compare two extraction conditions: - Baseline (One-shot extraction): one LLM call extracts entities and relations simultaneously, followed by simple filtering. - Proposed (Two-Stage KG Extraction): Stage 1 candidate generation, Stage 2 batch verification, and final deterministic type constraint filtering based on empirically observed relation-specific (head$_{t}ype, tail_{t}ype)pairsderivedfromtrainingdata.$

Models and controlled variations. We evaluate three Gemini Flash family models: Gemini 2.0 Flash, Gemini 2.5 Flash, and Gemini 3 Flash Preview. For Gemini 2.5 Flash and Gemini 3 Flash Preview, we vary the “thinking” configuration between off and a setting labeled 2048; for Gemini 2.0 Flash, we use a configuration labeled none. For each (model, thinking) configuration, we evaluate both Baseline and Proposed conditions, holding the dataset, sampling, and metrics constant.

Structured outputs and implementation details that affect reproducibility. The pipeline uses the Google GenAI SDK with JSON Schema-constrained decoding and two schemas: EXTRACTION$_{S}CHEMAforcandidategenerationandVERIFICATION_{S}CHEMAforverification.Verificationis$
$based, wedonotreporthardwarespecifications; theintendedexecutionenvironmentisalocalorcloudPythonruntimewi$

Experimental design limitations. The small sample size (10 documents) limits the ability to compute stable confidence intervals and makes the study primarily diagnostic. In addition, the final type constraints are derived from training data and thus reflect dataset-specific typing regularities rather than a general ontology.

## 6 Results

We report micro-averaged precision, recall, and F1 on the 10-document JacRED dev sample for each Gemini Flash configuration and for both Baseline (one-shot) and Proposed (two-stage) extraction. Because each configuration is evaluated once on a single sampled set, we report the observed metrics and counts (TP, FP, FN) without confidence intervals.

Pattern 1: False positives consistently decrease under the proposed pipeline. Across all evaluated model configurations, the proposed two-stage pipeline reduces FP relative to the one-shot baseline, and precision increases in every case. This directly supports the hypothesis that verification plus deterministic type constraints can mitigate over-generation.

Pattern 2: Recall remains low and is the dominant bottleneck. Recall varies between 0.12 and 0.22 across all settings and does not show a uniform improvement under the proposed pipeline. This indicates that many gold relations remain missed (high FN), so improvements in precision do not necessarily translate into consistent F1 gains.

Pattern 3: The effect of “thinking” is configuration-dependent. Enabling thinking improves baseline performance in some cases, but in the proposed pipeline it can sometimes coincide with fewer true positives, consistent with the interpretation that verification may become overly conservative under some settings.

Experiment group A: Gemini 2.0 Flash (thinking: none). Baseline achieves Precision 0.20, Recall 0.15, F1 0.17 with TP 22, FP 86, FN 126. Proposed achieves Precision 0.35, Recall 0.19, F1 0.25 with TP 28, FP 51, FN 121. The main change is a large FP reduction (86 to 51), producing a sizeable precision increase and an F1 gain.

Experiment group B: Gemini 2.5 Flash (thinking: off vs 2048). With thinking off, Baseline yields Precision 0.17, Recall 0.16, F1 0.17 (TP 24, FP 115, FN 124). Proposed yields Precision 0.30, Recall 0.12, F1 0.17 (TP 18, FP 42, FN 130). Here precision improves strongly but recall drops, resulting in no F1 improvement. With thinking 2048, Baseline yields Precision 0.18, Recall 0.17, F1 0.17 (TP 25, FP 115, FN 123), while Proposed improves to Precision 0.36, Recall 0.21, F1 0.27 (TP 31, FP 54, FN 117). In this configuration, the proposed pipeline improves both precision and recall relative to baseline, producing the best observed F1 for Gemini 2.5 Flash.

Experiment group C: Gemini 3 Flash Preview (thinking: off vs 2048). With thinking off, Baseline yields Precision 0.26, Recall 0.16, F1 0.20 (TP 24, FP 70, FN 124). Proposed yields Precision 0.36, Recall 0.22, F1 0.27 (TP 32, FP 56, FN 116). This setting achieves high F1 without additional thinking overhead, matching the analysis that it provides a strong cost-performance trade-off. With thinking 2048, Baseline yields Precision 0.31, Recall 0.22, F1 0.26 (TP 33, FP 74, FN 115), and Proposed yields Precision 0.37, Recall 0.20, F1 0.26 (TP 30, FP 52, FN 118). Here precision increases and FP decreases, but recall decreases slightly and F1 remains unchanged.

Synthesis and limitations. Overall, the proposed method is robust in improving precision and reducing false positives across all models and settings, but F1 improvements depend on whether verification and type constraints retain enough true positives. The consistently low recall suggests that improving candidate generation and entity alignment is necessary for further gains. Results are limited by the small evaluation set (10 documents) and should be interpreted as evidence for a practical trend rather than a definitive benchmark.

Figures. No figures were provided in the experimental logs, so none are included.

## 7 Conclusion

Low-cost LLMs make document-level relation extraction attractive for rapid knowledge graph construction, but one-shot extraction can be unreliable due to a high rate of false positives, especially in Japanese DocRE settings such as JacRED **?**. We investigated this practical precision bottleneck and proposed a two-stage pipeline that enforces structured outputs, separates candidate proposal from candidate verification, and applies deterministic relation-specific type-pair constraints derived from training data.

On a cost-aware evaluation over 10 JacRED development documents sampled by character length, the method consistently reduced false positives and improved precision across all tested Gemini Flash family configurations. These precision gains translated into micro-F1 improvements up to 0.27 in the best observed settings, while leaving recall as the central unresolved limitation.

The broader implication is that, even without fine-tuning, reliability-oriented pipeline design can substantially improve the usability of low-cost LLMs for Japanese DocRE by controlling over-generation. Future work should focus on recall: improving candidate generation coverage, strengthening entity alignment and relation directionality handling, and calibrating verification so that it remains strict on unsupported triples without discarding fragile but correct relations.

This work was generated by AIRAS (**?**).