Domain-Aware Error Correction for Citation NER in Medieval Hebrew Responsa

Shmuel Liebeskind[1], **Maayan Zhitomirsky-Geffet**[1], **Binyamin Katzoff**[1], **Nati Ben-Gigi**[1], **Jonathan Schler**[2]
[1]Bar-Ilan University, Ramat-Gan, Israel
[2]Holon Institute of Technology (HIT), Holon, Israel
israellieb@gmail.com, {maayan.zhitomirsky-geffet, Binyamin.Katzoff }@biu.ac.il,
{nati.bengigi, schler }@gmail.com

Abstract

Citation identification in historical and ancient texts poses challenges that extend beyond surface-level pattern recognition, including implicit references, morphological fusion, and discourse-driven ambiguity. In this work, we address citation Named Entity Recognition (NER) in medieval Hebrew Responsa literature using a modular, LLM-based correction pipeline. Rather than treating large language models as end-to-end predictors, we leverage them as structured components: an initial prompt-based expert tagger, complementary LLM judges for systematic error detection, and domain-aware correction grounded in philological regularities. Our approach requires no end-to-end fine-tuning and only minimal labeled supervision (a small validation set for training a lightweight error-detection classifier), narrowing the performance gap to strong supervised models trained on domain-specific data. The results suggest that explicit error handling and interpretability-driven design offer a promising direction for historical NLP in low-resource settings.

Keywords: Historical NLP, Citation Extraction, Medieval Hebrew, Large Language Models, Low-Resource Languages

1. Introduction

Historical and ancient language corpora present enduring challenges for natural language processing, especially when the task necessitates detailed structural interpretation rather than superficial pattern recognition. The Medieval Hebrew Responsa literature, a collection encompassing over 1,300 years of rabbinic legal discourse, illustrates this challenge. Citations in these texts are pivotal to academic reasoning; however, they exhibit significant heterogeneity: references are implicit, abbreviated, morphologically fused with prefixes, and frequently integrated into intricate discourse structures that lack standardized formatting.

Accurate citation Named Entity Recognition (NER) in this domain is therefore not merely a technical preprocessing step, but a prerequisite for large-scale digital humanities research, including citation network construction, intellectual lineage tracing, and diachronic analysis of legal discourse. Prior work has demonstrated that strong supervised models, such as BERT-CRF architectures trained on domain-specific corpora, can achieve high performance on this task. However, these approaches require substantial task-specific annotation and retraining, which is costly and difficult to replicate across historical domains and languages.

In this work, we explore an alternative paradigm: leveraging large language models (LLMs) not as end-to-end taggers, but as structured, rule-aware components within a correction pipeline tailored to historical texts. Our approach builds on a carefully engineered initial prompting stage that functions as a hybrid between a rule-based expert system and a lexicon-guided annotator. This system prompt encodes domain expertise about medieval Hebrew Responsa literature, explicitly defines a hierarchical NER taxonomy, injects lexical anchors for common rabbinic abbreviations, and enforces deterministic tagging constraints that reflect Hebrew morphology and citation practice. The system prompt was generated and iteratively refined by the LLM using solely the validation set, implementing domain knowledge through a validation-driven adaptation process without any access to the test set.

While this prompt-driven baseline already yields competitive performance, its remaining errors are systematic rather than random. We address these errors using an LLM-as-Judge correction pipeline, in which complementary judge personas evaluate token-level predictions and identify likely mistakes. Correction decisions are then guided by domain-aware heuristics that encode philological regularities, such as the distinction between quoted lemma text and actual citations, or between discourse-level mentions of sages and authoritative references. Notably, our approach does not rely on explicit modeling of citation structure; instead, improvements arise from targeted error detection and correction grounded in domain knowledge.

Our contributions are fourfold: (1) we intro-

Proceedings of LT4HALA 2026 @ LREC 2026, pages 96–105.

duce a prompt-as-expert baseline for historical citation NER, developed and refined through a validation-driven iterative process; (2) we propose a lightweight LLM-as-Judge correction framework that identifies systematic token-level errors without directly generating replacement labels; (3) we present a domain-aware correction strategy that achieves substantial performance gains without relying on structural heuristics or task-specific fine-tuning; and (4) we demonstrate an annotation-efficient pipeline that achieves competitive performance without large-scale supervised training, reducing reliance on costly expert annotation and enabling rapid adaptation to new historical corpora.

Together, these findings show that in historical and ancient language settings, LLM-based systems are most effective when embedded within interpretable, error-aware pipelines rather than used as monolithic predictors.

2. Related Work

Our work builds on three lines of research: citation and reference extraction, NLP for Hebrew and historical texts, and the use of LLMs as evaluators or corrective components.

Citation and Reference Extraction. Early approaches to citation parsing and reference extraction relied on rule-based systems and Conditional Random Fields (CRFs), typically targeting modern scholarly literature with standardized formats (Councill et al., 2008). Neural sequence models such as BiLSTM-CRF and later BERT-CRF architectures significantly improved performance in well-resourced settings (Lample et al., 2016; Devlin et al., 2019). However, these approaches generally assume stable orthography and explicit citation markers, assumptions that break down in historical corpora.

Within the domain of Hebrew and Jewish texts, prior work has addressed reference detection under more constrained conditions. HaCohen-Kerner et al. (2010) focused on modern Responsa literature, where citation conventions are relatively standardized. Zhitomirsky-Geffet and Prebor (2019) applied pattern-based extraction methods to identify rabbinic sages in the Mishnah. More recently, Ben-Gigi et al. (2025) demonstrated that domain-specific pretraining is crucial for medieval Responsa citation NER, achieving strong results with a BEREL-CRF model trained on Rabbinic Hebrew.

Beyond modern bibliometric parsing, a parallel line of research has addressed citation extraction in historical and classical corpora. Romanello (2013) developed methods for canonical citation extraction from classical texts, later extended to large-scale citation network construction (Kokash et al., 2024). Berti (2019) introduced a framework for extracting fragmentary citations of classical authors, demonstrating that domain-specific structural knowledge is indispensable for texts with non-standard reference conventions. Colavizza et al. (2023) proposed the Humanities Citation Index (HuCI), offering a systematic methodology for citation indexing across historical disciplines. Our work extends this tradition to the medieval Hebrew Responsa domain, where morphological fusion, abbreviation density, and implicit references pose additional challenges beyond those encountered in Greek and Latin corpora.

NLP for Historical and Ancient Languages. A growing body of work highlights the challenges of applying NLP techniques to historical and ancient texts, including diachronic variation, non-standard spelling, and limited annotated data (Piotrowski, 2012; Ehrmann et al., 2016). In response, researchers have emphasized interpretability, robustness, and reuse of linguistic knowledge over purely end-to-end optimization (Bollmann, 2019). Our approach aligns with this paradigm by explicitly encoding philological constraints and systematic error patterns within a modular pipeline.

LLMs as Judges and Corrective Modules. Recent work has explored the use of LLMs as evaluators, critics, or judges of model outputs, particularly in generation and dialogue tasks (Liu et al., 2023; Zheng et al., 2023). These approaches typically frame LLMs as meta-evaluators rather than primary predictors. We extend this paradigm to token-level structured prediction in a historical setting, using LLMs to localize systematic errors in NER outputs. Unlike prior LLM-as-Judge work focused on preference ranking or benchmarking, our framework integrates judging with lightweight supervision and domain-aware correction.

In contrast to existing approaches, we do not aim to replace supervised models or to infer citation structure explicitly. Instead, we demonstrate how LLMs can function as modular, interpretable components that amplify domain knowledge and improve robustness in low-resource historical NLP scenarios.

3. Task and Data

3.1. Task Definition

We address the *internal component identification* task for citation Named Entity Recognition (NER) in medieval Hebrew Responsa literature, following the formulation introduced by Ben-Gigi et al. (2025). Rather than treating each citation as a single span, the task requires identifying and labeling

the atomic components that jointly constitute a reference. This fine-grained decomposition is a prerequisite for downstream tasks such as citation normalization and citation network construction. Unlike flat or nested-span representations, component-level decomposition is essential because downstream tasks such as citation normalization require separate access to each element (e.g., distinguishing the book title from the chapter identifier within the same span) to resolve a reference to a canonical entry. Nested spans would preserve surface boundaries but obscure the functional role of each token.

Each token is assigned one of seven entity types: BN (Book Name, e.g., tractates); AN (Author Name); AA (Author Adjective derived from names); BA (Book Adjective, rare); R (Reference identifiers like chapter/folio); RW (Reference Words like "section" or "page"); and O (Outside, for non-citation tokens).

To illustrate the annotation schema, consider the following sentence fragment from a typical responsum: u-khtav [O] ha-Rambam [AN] be-P"A [R] me-Hilkhot [RW] Isurei [BN] Bi'ah [BN] "And the Rambam wrote, in Chapter 1 of the Laws of Forbidden Relations". Here, ha-Rambam is tagged AN (Author Name), be-P"A (an abbreviation for "in Chapter 1") is tagged R (Reference), me-Hilkhot is tagged RW (Reference Word meaning "laws of"), and Isurei Bi'ah is tagged BN (Book Name). The conjunction u-khtav ("and he wrote") is tagged O (Outside) as a discourse verb, not part of the citation itself.

This schema captures both lexical and structural aspects of rabbinic citations, which are often implicit, abbreviated, and embedded within running discourse.

3.2. Corpus and Experimental Design

All experiments are based on the citation-annotated Responsa corpus released by Ben-Gigi et al. (2025). The corpus consists of medieval Hebrew Responsa texts spanning approximately 1000–1500 CE and covering diverse geographic and cultural traditions, including Ashkenaz, France, Spain, and North Africa. The texts represent a wide range of authors, stylistic conventions, and orthographic practices.

Domain experts in Rabbinic Hebrew and rabbinic literature followed a detailed annotation guideline to produce annotations. The corpus is small compared to modern NER benchmarks due to high annotation costs and expertise, representing low-resource historical NLP settings.

In the original work, the corpus is divided into training and test sets, and results are reported using cross-validation over the training data, reflecting a supervised fine-tuning paradigm. Our setting differs fundamentally. Since our approach does not involve end-to-end model training or fine-tuning, there is no requirement for a large labeled training corpus. Instead, we retain the original test set unchanged to ensure direct comparability with prior results, while constructing a smaller development set sampled from the original training data.

Specifically, we sample a subset of sentences from the original training split to serve as a combined validation and development set. This subset is randomly sampled while preserving the original tag distribution, and is matched to the test set in size and entity composition. This validation set is used exclusively for prompt refinement, iterative error analysis, and training the lightweight error detection classifier. No information from the test set is used during these stages, ensuring strict separation between development and evaluation.

We note that the corpus exhibits considerable internal variability across geographic traditions and time periods. While random sampling preserves global tag distributions, it may under- or over-represent certain stylistic or orthographic conventions. Robustness checks using multiple independent samples, or stratified sampling by tradition and period, would further strengthen reproducibility and are planned for future work.

This experimental design reflects the core methodological goal of our work: enabling effective citation extraction without requiring large-scale supervised training, while maintaining full comparability with prior supervised baselines.

3.3. Data Splits and Statistics

Table 1 summarizes the resulting data splits used in our experiments. We report statistics only for the validation/development set and the test set, as no full training split is required for our method.

Split	Sentences	Tokens	Entity Ratio
Validation	294	3,144	55.2%
Test	263	2,733	52.9%

Table 1: Dataset statistics. Entity ratio denotes the proportion of tokens annotated as citation components (non-*O*); for example, an entity ratio of 55.2% indicates that 55.2% of tokens in the split belong to citation entities, while the remaining 44.8% correspond to regular text (*O*).

Table 2 shows the token-level distribution of entity types in both the validation and test sets. As anticipated, O tokens make up about half of the data, while R and BN are the largest entity classes. Adjective categories (AA, BA) are uncommon, but play a significant linguistic role.

In addition to the primary citation entity categories, the corpus includes auxiliary structural labels grouped under the *N/A* category. These tokens correspond to citation-adjacent or structural

elements such as reference indicators, boundary markers, editorial additions, and discourse signals. While these elements are excluded from named entity evaluation as non-target categories, they provide important contextual cues that assist in identifying citation boundaries and improving overall reference resolution.

Tag	Val (N)	Val (%)	Test (N)	Test (%)
O	1,407	44.8	1,287	47.1
R	609	19.4	481	17.6
BN	346	11.0	277	10.1
RW	260	8.3	186	6.8
AN	135	4.3	176	6.4
AA	85	2.7	101	3.7
BA	30	1.0	6	0.2
N/A	272	8.7	219	8.0

Table 2: Token-level entity distribution in the validation and test sets.

The skewed distribution of entity types in the corpus directly informs several design choices in our method. In particular, the extreme sparsity of adjectival categories—most notably *BA*, which accounts for less than 1% of annotated tokens—makes them especially vulnerable to over-prediction in prompt-based settings. This observation motivates our decision to explicitly mark such categories as rare in the system prompt and to prioritize reference tokens (*R*) in ambiguous contexts.

Similarly, the dominance of *R* and *BN* tokens, combined with the high proportion of non-entity (*O*) tokens, explains both the conservative fallback rules at the prompt stage and the need for a downstream correction mechanism to suppress systematic false positives. The presence of structurally informative but non-evaluated tokens (*N/A*) further reinforces the importance of structural reasoning when identifying citation spans.

These data characteristics highlight why domain-aware constraints and post-hoc error correction are essential for robust citation NER in this corpus.

These distributional properties also shape the error patterns observed at evaluation time, a point we return to in the analysis section, where we examine how sparsity, ambiguity, and boundary effects interact with the correction pipeline.

It is important to note that the relatively high entity ratios reflect the corpus construction methodology, which focuses on citation-rich passages rather than uniformly sampled running text. As a result, citation components are substantially more frequent than in general-domain NER datasets.

The test set is strictly held out and used only for final evaluation. At no stage does the correction pipeline access gold labels from the test data, ensuring a clean separation between development and evaluation and preventing information leakage.

4. Method

Figure 1 provides an overview of the full correction pipeline, illustrating the separation between prompt-based initial tagging, systematic error detection using LLM judges, and domain-aware correction. The following sections describe each component in the order shown in the figure. Our approach consists of a four-stage pipeline designed to separate initial tagging, error detection, and error correction, while explicitly encoding domain knowledge relevant to medieval Hebrew Responsa literature. Crucially, all components are developed under strict train/validation/test separation.

Task Setting and Baseline Assumptions We address the task of token-level Named Entity Recognition (NER) for internal citation components in medieval Hebrew Responsa texts. Given the limited availability of annotated historical data and the high cost of task-specific fine-tuning, we adopt a few-shot prompting setup as a realistic baseline. Rather than treating the LLM as an end-to-end solution, we position it as a modular component within a structured pipeline.

4.1. System Prompt as a Structured Expert Module

The core of our baseline is a carefully engineered system prompt that functions as a hybrid between a rule-based expert system and a lexicon-guided annotator. The prompt was refined over 2–3 validation-only iterations, with the goal of operationalizing philological knowledge rather than optimizing surface-level performance.

Prompt Construction and Refinement. The initial system prompt was not created spontaneously, but generated by the LLM based on specific task definitions and annotation guidelines derived from previous research on medieval Hebrew citation NER, particularly the schema established by Ben-Gigi et al. (2023). This document encompasses the citation taxonomy, boundary conventions, and instances of prevalent rabbinic abbreviations and reference patterns.

Using these specifications as input, the LLM generated an initial system prompt that operationalizes the task as a rule-aware expert annotator. This prompt was then refined through an iterative process on the validation set only, during which the model evaluated its own predictions, identified systematic errors, and adjusted the prompt accordingly. No information from the test set was used at any stage of prompt construction or refinement.

This process characterizes prompt engineering as a lightweight, validation-driven adaptation, as

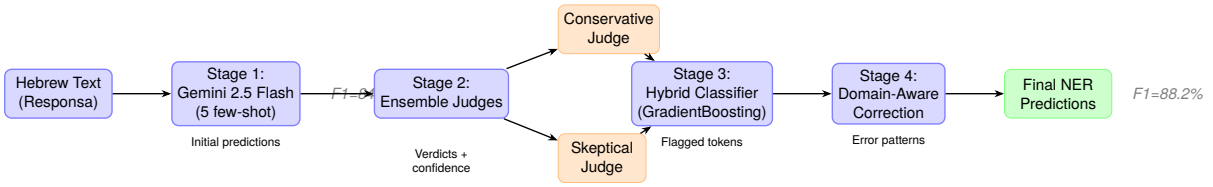

Figure 1: LLM-as-Judge correction pipeline. Stage 1 generates initial predictions; Stage 2 uses complementary judge personas to evaluate each token; Stage 3 employs a GradientBoosting classifier (trained on validation set only) to flag likely errors; Stage 4 applies domain-specific corrections.

opposed to manual rule creation or task-specific fine-tuning.

The final prompt consists of the following components:

Persona and Domain Narrowing. The model is instructed to assume the role of an expert in medieval Hebrew Responsa literature, explicitly prioritizing Rabbinic Hebrew and Aramaic conventions over Modern Hebrew usage. This narrows the model's attention to the appropriate linguistic register and discourse norms.

Hierarchical NER Taxonomy with Prioritization. We define a closed set of entity types (BN, AN, AA, BA, R, RW, O) with explicit definitions. The prompt prioritizes reference tokens (R) as the most important class and marks adjectival modifiers (BA) as rare. This serves as an implicit prioritization heuristic in ambiguous situations.

Lexical Anchors and Abbreviation Awareness. The prompt includes canonical examples of common rabbinic abbreviations and reference markers (e.g., *P"Q* for *Perek Kama* [first chapter], *A"A* for *Amud Aleph* [page side A], and *HaRambam* for Maimonides). These anchors function as a lightweight lookup table, reducing false negatives for highly frequent but opaque forms.

Some reference markers originate from the original authors, while others, such as page indicators and structural divisions, were introduced by later editors. From the perspective of citation NER, both types function as valid anchors for locating references and are therefore treated uniformly within our tagging and correction framework.

Anticipatory Disambiguation Constraints. To address known confusion patterns, the prompt explicitly contrasts similar-looking forms that belong to different categories (e.g., *D'AZ* [of Tractate Avodah Zarah] as a book name versus *P"Q* [First Chapter] as a reference marker). These contrastive explanations guide the model to attend to morphemic roots rather than surface prefixes.

Deterministic Tagging Rules. Finally, the prompt enforces strict structural constraints: all tokens must be tagged in order; Hebrew prefixes remain attached to their base tokens; and When uncertainty persists, the model defaults to tagging the token as *R*. This fallback strategy promotes tagging consistency and reduces fragmentation errors at the prompt stage. While this bias intentionally favors recall, potential over-generation is subsequently addressed by the downstream error detection and correction pipeline

All prompt-based annotation and judge evaluations were performed using the Gemini-2.5 Flash LLM, without any task-specific fine-tuning. This model was selected for practical considerations: it was the largest freely accessible LLM at the time of experimentation, provided reliable structured output generation, and supported the long-context windows required for our prompting setup. We do not claim that Gemini-2.5 Flash is necessarily the best-performing model for this task; no systematic cross-model comparison was conducted, and we expect that other large LLMs would yield comparable results given appropriate prompt re-optimization.

4.2. Initial LLM Tagging

Using this system prompt, we apply a few-shot prompting setup with 5 curated examples that emphasize rare entity types and complex citation patterns. This stage produces the initial token-level NER predictions that serve as input to subsequent correction stages.

4.3. LLM-as-Judge for Error Detection

To identify systematic errors in the initial predictions, we employ an LLM-as-Judge framework with two complementary judge personas:

- **Conservative Judge:** biased toward high precision and skeptical of entity assignments.
- **Skeptical Judge:** biased toward high recall and attentive to missed entities in entity-dense contexts.

The 0.8 confidence threshold for the Skeptical Judge was selected based on validation set

tuning. Thresholds of 0.7, 0.8, and 0.9 were compared; 0.8 provided the best balance between flagging genuine errors and avoiding excessive false alarms, yielding the highest downstream correction F1 on the validation set.

Each judge independently evaluates the predicted tag for each token and outputs a binary verdict (correct / incorrect) with a confidence score. Judges are not allowed to propose alternative tags; their role is limited to error detection.

4.4. Hybrid Error Detection Classifier

Judge outputs alone are insufficient due to frequent disagreement in ambiguous cases. We therefore train a lightweight Gradient Boosting classifier to predict the likelihood that a token is incorrectly tagged, using scikit-learn default hyperparameters. The classifier was trained using 25 features across six categories: judge confidence scores and agreement levels (4), aggregated verdict indicators and vote counts (4), agreement pattern types (4), one-hot encoding of proposed tag types (7), token-level linguistic features including word length and Hebrew prefix detection (3), and positional features capturing token location within the sentence and citation span (3).

This feature design enables the classifier to combine structural, semantic, and consensus-based signals to resolve ambiguous tagging decisions

The classifier is trained only on the validation set using gold labels for error detection. Once trained, it is frozen and applied to the test set without access to ground truth labels, ensuring no leakage.

Among the six feature categories, judge agreement and confidence scores were the strongest predictors of classification errors, followed by tag-type indicators such as predicted entity label, and positional features including token location and entity density.

We emphasize that the Gradient Boosting classifier constitutes a lightweight supervised component trained on gold error-detection labels from the validation set. The pipeline therefore involves minimal labeled supervision at the error-detection stage, while the core NER predictions at Stages 1, 2, and 4 remain unsupervised. We adopt the term "minimal supervision" rather than "no fine-tuning" to accurately characterize this design.

4.5. Domain-Aware Error Correction

Tokens flagged as likely errors are passed to a correction stage that applies domain-aware prompting. Correction prompts encode philological regularities observed during validation analysis, including:

- **Lemma formulas:** Recognition of markers such as *D"H* (abbreviation for *Dibbur Ha-Matkhil*) or *Be-Dibbur Ha-Matkhil* ("In the comment beginning with"), which signal that subsequent tokens are quoted text rather than location references (R).
- **Discourse Context:** Distinguishing discourse mentions of sages from authoritative citations. For instance, the system corrects instances where names are over-tagged in narrative contexts, such as *Ben Shamoa Omer* ("Ben Shamoa says"), which should be Outside (O), versus actual citations like *Ke-de-pirash Ben Shamoa* ("As Ben Shamoa explained"), where the name is an Author Name (AN).
- **Entity Ambiguity:** Resolving common confusions between authors and their works, specifically identifying whether a marker like *Ha-Rosh* refers to the person (AN) in the context of *Katav Ha-Rosh* ("The Rosh wrote") or the book (BN) in the context of *Be-Sefer Ha-Rosh* ("In the book of the Rosh").

Corrections are applied conservatively and only to flagged tokens, preserving the original predictions elsewhere.

Table 3 summarizes the distinct prompt configurations used across the pipeline.

5. Results

Table 4 shows the main test set results. The prompt-based baseline achieves a high F1 score, indicating that a well-designed system prompt with domain knowledge and constraints can capture citation behavior in medieval Hebrew texts.

We also evaluated both one-shot and few-shot prompting configurations. The performance differences between them were marginal and not statistically significant, indicating that explicit task instructions and domain-aware constraints contributed more to performance than the inclusion of demonstration examples.

Applying the LLM-as-Judge correction pipeline yields a substantial improvement of approximately four F_1 points over the prompt-only baseline. This gain is driven primarily by a marked increase in precision, alongside a smaller but consistent improvement in recall. These results indicate that the correction pipeline is particularly effective at suppressing systematic false positives while still recovering a meaningful number of missed citations.

Notably, the corrected prompt-based system narrows the performance gap to a strong supervised BEREL-CRF model trained on domain-specific data, despite requiring no task-specific fine-tuning or additional annotated data. This highlights the

Table 3: Overview of the distinct prompt configurations used across the pipeline. Each component employs a role-specific prompt with different objectives and inductive biases, enabling modular tagging, error detection, and targeted correction.

Aspect	Tagger	Conservative Judge	Skeptical Judge	Correction Module
Primary role	Assign citation-related tags to all tokens	Validate predicted tags and protect correct assignments	Aggressively identify potential tagging errors	Apply targeted fixes to flagged tokens only
Input	Raw text	Text + predicted tags	Text + predicted tags	Text + predicted tags + flagged token positions
Output	Token-level tags	Correct/Incorrect verdict per token	Correct/Incorrect verdict per token	Corrected tags for flagged tokens
Inductive bias	Favor recall; when uncertain, default to *R*	Favor precision; default to CORRECT when evidence is ambiguous	Favor recall; default to INCORRECT under uncertainty	Preserve non-flagged predictions exactly
Typical error patterns addressed	Boundary ambiguity; abbreviation confusion	Over-tagging of non-citation material	Missed or weakly marked citation boundaries	Systematic philological error patterns

Method	P	R	F1
Prior Work			
BEREL-CRF	90.9	88.4	89.6
Prompting Methods			
Zero-shot	81.4	87.4	84.3
Few-shot	78.7	90.9	84.4
Our Approach			
+ LLM-as-Judge	**84.5**	**92.3**	**88.2**

Table 4: Test set results. Our pipeline improves few-shot by +3.8 F1 points, approaching BEREL-CRF performance without task-specific fine-tuning.

effectiveness of explicit error detection and correction as an alternative to retraining in low-resource historical settings.

We started with citation subtype analyses as a diagnostic tool. We found that explicit citation structure modeling did not affect performance gains. To maintain a structure-agnostic and portable pipeline, we exclude such analyses from the final system and report only token-level error detection and correction results.

5.1. Ablation Study

To assess the contribution of individual components within the correction pipeline, we conduct an ablation study summarized in Table 5. Starting from the prompt-based baseline, we incrementally add (i) LLM judges with majority voting, (ii) a hybrid error detection classifier, and (iii) the full domain-aware correction stage.

Note: The ablation results in Table 5 are evaluated on the validation set, whereas Table 4 reports test set performance. The difference in few-shot F1 (85.3 on the validation set versus 84.4 on the test set) reflects natural variation between the two splits and does not indicate an inconsistency.

Configuration	F1	Δ
Few-shot only	85.3	—
+ Ensemble Judges + Classifier	87.5	+2.2
+ Correction	**89.8**	+2.3
Error Detection (Flagging)		
Ensemble Voting only	55.9	—
+ Hybrid Classifier	**66.4**	+10.5

Table 5: Ablation study (validation set). The hybrid classifier improves error detection F1 by +10.5% over voting alone by using token-level features. Note: The few-shot baseline F1 differs from Table 4 because the two tables report on different evaluation splits (validation vs. test).

Introducing LLM judges alone yields a moderate improvement, indicating that judge agreement provides a useful but incomplete signal for identifying erroneous predictions. Adding the hybrid classifier further improves performance, demonstrating that combining judge signals with token-level features substantially enhances error detection. The full correction pipeline achieves the highest F_1 score, confirming that performance gains arise from the interaction between systematic error detection and targeted, philologically grounded correction.

Notably, we exclude ablations based on explicit

citation subtype modeling, as preliminary experiments showed that correction gains did not depend on such structural assumptions. This design choice reflects our goal of maintaining a structure-agnostic and portable pipeline.

6. Analysis

We analyze the behavior of the correction pipeline to better understand the sources of improvement and remaining limitations.

6.1. Judge Agreement Analysis

We first analyze agreement patterns between the two LLM judges to assess the reliability of error detection signals. When both judges agree that a prediction is correct, accuracy reaches 94.2%, indicating a high-confidence region where correction is unnecessary. Conversely, when both judges flag a prediction as erroneous, 71% of such cases correspond to true errors, demonstrating that joint disagreement provides a strong signal for corrective intervention. Across the test set, the LLM judges flag approximately 14% of tokens as potential errors. The majority of these cases correspond to genuine misclassifications. Disagreements between judges occur primarily in boundary cases involving short functional tokens, abbreviations, or morphologically fused forms, which are known sources of ambiguity in Rabbinic Hebrew.

Complementary Judge Biases. The two LLM judges are deliberately designed to embody complementary error-detection biases. The *Conservative Judge* is provided with extensive guidance on valid citation patterns and is instructed to avoid false positives, defaulting to CORRECT when evidence is ambiguous. In contrast, the *Skeptical Judge* operates as an aggressive auditor, explicitly instructed to assume that tags may be incorrect and to flag any token whose correctness confidence falls below 0.8, thereby prioritizing recall over precision.

This asymmetric design—one judge protecting correct predictions and the other actively hunting for errors—explains the agreement patterns observed in practice. Joint CORRECT decisions correspond to a high-precision regime, while joint INCORRECT decisions yield a strong true positive signal for error detection. Disagreement cases naturally occupy an intermediate confidence region, motivating the use of a hybrid classifier to resolve ambiguity.

Why a Classifier? Simple voting rules are a natural baseline for aggregating judge decisions, but ineffective. Using majority voting alone for error detection yields only 55.9% F_1. In contrast, the hybrid error detection classifier enhances performance by +10.5 percentage points to 66.4% F_1.

This improvement is primarily driven by the classifier's ability to resolve "split-decision" cases, where the two judges disagree. Across the test set, such disagreements occur for 191 tokens. By incorporating token-level features alongside judge signals, the classifier successfully disambiguates many of these cases, leading to more accurate error localization than voting alone.

These results justify the inclusion of a lightweight supervised classifier and demonstrate that effective error detection requires integrating complementary judgment signals with local contextual information.

Importantly, the two judges exhibit complementary biases: one judge adopts a more conservative stance, favoring high precision in error detection, while the other is more skeptical, prioritizing recall. This asymmetry proves beneficial, as disagreements between judges frequently arise in such boundary cases. The hybrid error detection classifier effectively integrates these complementary signals, improving robustness beyond simple majority voting.

These results validate that structured error localization, not redundant or correlated judgments, is the source of performance gains and support the design decision to use multiple judge personas.

6.2. Error Reduction Analysis

To quantify the practical impact of the correction stage, we analyze the distribution and effect of corrected tokens. Across the test set, the pipeline flags 377 tokens (13.8%) across 189 sentences as candidates for correction. This selective intervention indicates that the system operates conservatively, modifying only a small subset of predictions rather than reprocessing all tokens.

Precision increases by 5.8% and recall by 1.4 points during the correction phase. This pattern indicates that the pipeline effectively reduces false positives and recovers missed citation entities. The increase in precision and recall indicates that gains are not solely due to conservative pruning, but also to meaningful error correction.

These findings are consistent with the ablation results, which show that performance improvements emerge only when error detection is followed by domain-aware correction, rather than from judge signals alone.

Dominant Correction Patterns. Qualitative inspection reveals that most successful corrections follow a small number of recurring philological patterns. A dominant category involves lemma formulas, where quoted text following formulaic markers is frequently misclassified as a citation by the

baseline model. The correction stage consistently reassigns such tokens to the non-citation class. Another common pattern concerns discourse-level mentions of sages or authorities that are not intended as authoritative references, which the correction heuristics successfully suppress.

Reference Type as a Diagnostic Signal. Although reference type information is not used by the correction pipeline, we analyze model behavior across reference categories as a diagnostic tool. Reference types were automatically inferred by the LLM in a post-hoc analysis, without access to gold labels, based solely on the predicted citation spans and local context.

Table 6 shows that error rates vary substantially across reference types. Explicit structural references (e.g., chapter or folio identifiers) exhibit the lowest error rates, while implicit or discourse-level references are considerably more error-prone. Notably, lemma-based references—where quoted text follows formulaic markers—account for a disproportionate share of false positives, consistent with the qualitative correction patterns discussed above.

Ref. Type	N	Init	Final	△F1	Net
Explicit	67	76.5	87.0	+10.4	+48
Consecutive	50	86.2	90.6	+4.4	+25
Recursive	17	80.4	83.2	+2.8	+5
Hierarchical	75	88.7	89.7	+1.0	+7
Partial	38	90.7	88.2	-2.5	-4
Unknown	10	90.0	81.5	-8.5	-9

Table 6: Error analysis by reference type (test set). Net = errors fixed − errors introduced. Clear structures (explicit, consecutive) benefit most; ambiguous types show regressions.

These findings explain why explicit modeling of reference type did not improve performance: reference categories correlate with error likelihood but cannot reliably determine token-level prediction accuracy. However, the analysis highlights systematic challenges that may inform future work on citation modeling and normalization.

Stability and Practical Considerations. While LLM-based predictions exhibit some stochasticity across runs, the correction pipeline acts conservatively, intervening only on tokens flagged as likely errors. In practice, this stabilizes predictions rather than amplifying variance. Computationally, the pipeline incurs additional latency due to multiple LLM calls, making it more suitable for offline or batch processing scenarios typical of digital humanities research.

In terms of computational cost, the full pipeline processes approximately 3–5 sentences per minute. when run sequentially against the LLM‘s API, with most of the latency attributable to the three LLM calls per sentence (initial tagger and two judges) and optional rule-based correction. For the 263-sentence test set, end-to-end processing takes approximately 40 minutes. This latency is acceptable for offline digital humanities workflows but would require batching or parallelization for large-scale corpora comprising thousands of Responsa.

Overall,The analysis supports our central claim that performance gains result from systematic error detection and domain-aware correction based on philological knowledge, rather than implicit citation structure modeling or increased model complexity.

7. Conclusion

We developed a modular LLM-based correction pipeline for medieval Hebrew, addressing linguistic variation and limited data. To avoid end-to-end prediction, we use LLMs as structured, rule-aware components in an error-driven pipeline.

We discovered that systematic error detection and targeted correction can improve performance without task-specific fine-tuning. Validated system prompts provide a lightweight expert base, while complementary LLM judges identify errors for conservative, domain-aware correction based on philological regularities.

Beyond performance improvements, the approach substantially reduces reliance on large annotated corpora. While prior work depends on fully annotated training sets, our pipeline operates effectively using only a small validation set and iterative LLM-guided refinement, enabling efficient adaptation and scalable annotation workflows in low-resource historical domains.

Although the overall pipeline architecture (prompt-based expert tagging, LLM judges, and a hybrid classifier, domain-aware correction) is language- by design, practical portability requires re-engineering two corpus-specific components: the lexical anchors and abbreviation tables embedded in the system prompt, and the philological correction heuristics of Stage 4. While applying this pipeline to other historical traditions (e.g., Talmudic commentary, medieval Latin legal texts, or classical Greek citations) requires domain-expert input to replace these specific modules components, the underlying structural framework and judge-classifier architecture remain directly reusable. This highlights the potential of interpretable, error-aware LLM pipelines as a practical, scalable alternative to traditional supervised approaches for ancient and specialized texts.

8. Reproducibility

To support reproducibility and further research, all system components are available at https://github.com/semlie/Correction-Pipeline-for-Citation. This release includes: (1) the complete system and judge prompts used across all pipeline stages; (2) the Gradient Boosting classifier implementation and trained model weights; (3) the feature extraction code and domain-aware correction rule set; and (4) the validation/test split definitions. The annotated corpus is available through the original release by Ben-Gigi et al. (2025).

9. Bibliographical References

Nati Ben-Gigi, Maayan Zhitomirsky-Geffet, Jonathan Schler, and Binyamin Katzoff. 2025. Automatic construction of the citation network from the medieval jewish responsa literature. *ACM Journal on Computing and Cultural Heritage*, 18(2):1–18.

Monica Berti. 2019. *Digital classical philology: Ancient Greek and Latin in the digital revolution*, volume 10. Walter de Gruyter GmbH & Co KG.

Marcel Bollmann. 2019. A large-scale comparison of historical text normalization systems. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 3885–3898.

Giovanni Colavizza, Silvio Peroni, and Matteo Romanello. 2023. The case for the humanities citation index (huci): a citation index by the humanities, for the humanities. *International Journal on Digital Libraries*, 24(4):191–204.

Isaac Councill, C Lee Giles, and Min-Yen Kan. 2008. Parscit: an open-source crf reference string parsing package. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Maud Ehrmann, Chiara Palladino, and Giuseppe GA Celano. 2016. Diachronic named entity recognition and linking. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 1–9. The COLING 2016 Organizing Committee.

Yaakov HaCohen-Kerner, Nadav Schweitzer, and Yaakov Shoham. 2010. Automatic identification of biblical quotations in hebrew-aramaic documents. In *International Conference on Knowledge Discovery and Information Retrieval*, volume 2, pages 320–325. SCITEPRESS.

Natallia Kokash, Matteo Romanello, Ernest Suyver, and Giovanni Colavizza. 2024. The brill knowledge graph: A database of bibliographic references and index terms extracted from books in humanities and social sciences. *Research Data Journal for the Humanities and Social Sciences*, 9(1):1–21.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 260–270.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 2511–2522.

Michael Piotrowski. 2012. *Natural language processing for historical texts*. Morgan & Claypool Publishers.

Matteo Romanello. 2013. Creating an annotated corpus for extracting canonical citations from classics-related texts by using active annotation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 60–76. Springer.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Maayan Zhitomirsky-Geffet and Gila Prebor. 2019. Sagebook: Toward a cross-generational social network for the jewish sages' prosopography. *Digital Scholarship in the Humanities*, 34(3):676–695.