Critical Comparative Evaluation of Five CV Information-Extraction Pipelines

Critical Comparative Evaluation of Five CV Information-Extraction Pipelines

ChatGPT, Gemma 4 MoE, Qwen3.6 MoE, Hybrid NLP, and BERT MULTI across 2,484 CVs and 24 occupational domains

A mathematically explicit, implementation-audited, PDF-grounded, and critically reflexive study for retrieval-augmented CV search

Empirical corpus: 2,484 source PDFs

Evaluation strata: 24 occupational categories

Compared extraction systems: ChatGPT, Gemma, Qwen, Hybrid NLP (Fast, Local), and NLP BERT MULTI (DJL)

Report revised: 2026-07-25

Submission note: This report separates direct lexical evidence, schema validation, and machine-assisted atomic adjudication. It does not treat ChatGPT agreement as ground truth.

Abstract

This study critically compares five curriculum-vitae information-extraction pipelines—ChatGPT, locally served quantized Gemma and Qwen mixture-of-experts (MoE) models, Hybrid NLP (Fast, Local), and NLP BERT MULTI (DJL)—across 2,484 PDF CVs in 24 occupational categories. The report does not collapse heterogeneous measurements into a universal rank. Instead, it defines three evidence tiers: a common direct lexical/structural audit for ChatGPT, Gemma and Qwen; canonical atomic workbooks for Gemma and Qwen; and a shared later PDF-grounded atomic protocol for Hybrid NLP and BERT MULTI. The original MoE mathematics, reasoning-mode analysis, causal critique and model-selection framework are retained and extended with formal models of schema-aware deterministic extraction and GLiNER span inference reconstructed from the supplied Java implementation. Under the Gemma/Qwen canonical layer, Qwen has higher weighted recall (73.65%), weighted F1 (82.94%) and completeness (75.52%), whereas Gemma has a lower unsupported-content rate (1.24% versus 2.69%). Under the directly comparable HNLP/BERT protocol, Hybrid NLP reaches weighted F1 85.94% versus 71.51% for BERT MULTI and leads in all 24 categories; the mean paired difference is 14.10 percentage points (50,000-resample category-bootstrap 95% interval 12.63–15.82; Wilcoxon $p=1.19\times10^{-7}$). Segment reversals are theoretically important: BERT leads on certifications, languages and seniority, while Hybrid NLP leads on experience, skills, education and profile. The result is therefore interpreted as task–architecture interaction, not proof that one family is universally superior. The analysis identifies source conversion, prompt asymmetry, sparse routing, quantization, reasoning mode, chunking, thresholding, dictionary coverage, schema mapping and evaluator design as competing causes. A field-routed, source-aligned ensemble is recommended, with human-adjudicated gold data and controlled ablations required before causal or employment-screening claims.

Keywords: CV extraction; information retrieval; retrieval-augmented generation; mixture of experts; BERT; GLiNER; Hybrid NLP; named-entity recognition; precision; recall; causal inference; JSON Schema; auditability.

Academic integrity and reproducibility statement

All numerical results are computed from the supplied benchmark artifacts. External literature defines methods and theory; the construction of Hybrid NLP and BERT MULTI is derived from the supplied edoc-260 Java source. The statistical appendix contains the five-system evidence tiers, category and segment values, paired tests, architecture matrix, source manifest and figures. Original three-system report content is preserved through the REPORT-008 scholarly base, while the replaced REPORT-005 document is retained in a pre-rebuild backup. Automated workbooks are not treated as a human gold standard, and cross-tier values are never promoted to a causal five-system ranking.

Table of contents

List of figures

List of tables

1. Introduction

Retrieval-augmented generation combines parametric language generation with non-parametric external evidence [1]. In CV search, the evidence is candidate-specific and consequential: employment history, skills, education, certifications, languages, dates, and source passages. The extraction layer is therefore not a convenience-only preprocessing step. It controls what information can be retrieved, which facts can be cited, and which candidates may become false negatives.

Precision, recall, and F1 are established information-retrieval measures [2]. Their interpretation is particularly important here. High precision supports using extracted facts as positive evidence; lower recall means that absent structured values cannot safely be treated as negative evidence. The same asymmetry appears in the present results: the local pipelines usually emit PDF-supported content, yet omit a material portion of the facts represented in the machine-assisted atomic evidence.

Transparent evaluation also requires documentation of the data, model context, intended use, and limitations. Datasheets [4] and model cards [5] provide established frameworks for this purpose, while NIST AI RMF 1.0 emphasizes context-sensitive measurement, governance, mapping, and management of risk [6]. This report adopts those principles by preserving provenance, disaggregating results by category and field, and avoiding a single opaque quality score.

1.1 Problem statement

The source materials were created through multiple extraction, evaluation, consolidation, and reporting processes. This creates two risks: numerical drift between category and aggregate reports, and apparent model differences caused by evaluator implementation rather than extraction quality. The study therefore recomputes the principal measures from canonical inputs and places direct, uniformly implemented metrics beside—not inside—the heterogeneous atomic-evaluation layer.

1.2 Research questions

Table 1. Research questions and analytical evidence

ID	Research question	Evidence
RQ1	Do the three pipelines produce parseable, schema-valid outputs for the full PDF corpus?	Archive matching and Draft 2020-12 validation.
RQ2	How faithfully does each output preserve extractive text found in the PDFs?	Exact-object support and four-token PDF-support precision.
RQ3	How much broad source text is captured by each system?	PDF lexical-coverage proxy; interpreted descriptively.
RQ4	How do Gemma and Qwen compare on PDF-grounded atomic precision, recall, F1, completeness, and unsupported content?	Canonical TP/Partial/FP/FN workbooks.
RQ5	Are differences stable across occupational categories and semantic segments?	Paired category analysis, segment matrices, and document sensitivity analysis.
RQ6	What deployment controls follow for RAG-based CV search?	Evidence asymmetry, field-aware retrieval, fallback, and human review.
RQ7	Which MoE mechanisms could plausibly affect extraction, and which are observable here?	Formal router, load, active-parameter, memory, and quantization model.
RQ8	How should reasoning-disabled results be interpreted relative to reasoning-enabled operation?	Competing hypotheses and a controlled on/off/budgeted design.
RQ9	Which alternative models would provide informative controls?	Within-family dense siblings, independent dense baselines, and factorial testing.
RQ10	How do Hybrid NLP and BERT MULTI compare when evaluated under their shared PDF-grounded protocol?	Paired 24-category weighted-F1 differences, percentile bootstrap interval, sign pattern, and Wilcoxon test.
RQ11	How are the two non-generative extraction methods constructed in the supplied Java implementation?	Code-level reconstruction of preprocessing, scoring, inference, schema mapping, validation, and fallback stages.
RQ12	Which five-system comparisons are valid, and which would conflate incompatible evidence layers?	Explicit evidence-tier matrix and a comparability indicator requiring common corpus, target, evaluator, and aggregation.
RQ13	Which error mechanisms can explain the observed deviations without assuming intrinsic model superiority?	Stage-wise failure model, competing hypotheses, counterfactual tests, and controlled ablation design.
RQ14	How do Gemma, Qwen, Hybrid NLP and BERT MULTI compare when every raw JSON archive is re-evaluated under one frozen PDF-grounded protocol?	Canonical ID alignment, common fact splitter and matcher, common thresholds and missing-output policy, document bootstrap, and all six paired category contrasts.

1.3 Contributions

- A corrected, all-category comparison based on 2,484 PDFs and the user-designated canonical workbooks.
- A common direct-evidence layer that quantifies ChatGPT without treating it as ground truth.
- Full validation against the supplied Draft 2020-12 aiextract.json schema.
- Paired category-level uncertainty, effect-size, rank-correlation, and sensitivity analyses.
- Field-level evidence for a risk-aware RAG architecture.
- A reproducible reconciliation of prior aggregate claims with canonical recalculation.

2. Materials and methods

2.1 Corpus and unit of analysis

The corpus comprises 2,484 source CV PDFs distributed across 24 predefined occupational categories. The categories are sampling strata supplied by the dataset rather than labels inferred by the models. Document IDs are normalized from filenames and matched within category. Candidate names are not used for matching or reporting. The primary unit for pipeline integrity is the document; the unit for atomic metrics is the evaluated fact; and the primary unit for cross-domain comparison is the occupational category.

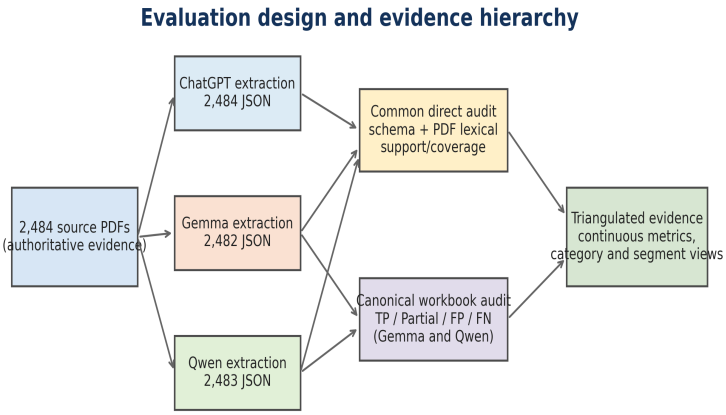

Figure 1. Evaluation design and evidence hierarchy.

> *Note. PDFs are authoritative. ChatGPT agreement is diagnostic; canonical workbook judgments form a separate evidence layer. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

2.2 Compared systems

Table 2. Systems and supplied inference configurations

System	Model identifier	Run context	Provenance limitation
ChatGPT	User-supplied provenance: "model 5.6 on high"	High-recall section-aware JSON extraction using prompt.2	Provider build, sampling details and run IDs were not retained.
Gemma	gemma-4-26B-A4B-it-qat-UD-Q4_K_XL.gguf	llama-server; Vulkan1; 65,536 context; parallel slots 4; reasoning off	Quantized local inference; exact llama.cpp build and driver manifest absent.
Qwen	Qwen3.6-35B-A3B-UD-Q4_K_XL.gguf	llama-server; Vulkan1; 65,536 context; parallel slots 4; reasoning off	Quantized local inference; exact llama.cpp build and driver manifest absent.
Hybrid NLP	Hybrid NLP (Fast, Local)	Local Java; schema-aware rules, dictionaries, fuzzy matching, regex and optional OpenNLP	No retained runtime log proving whether optional OpenNLP model files were loaded.
BERT MULTI	NLP BERT MULTI (DJL); local modelId models/djl/gliner_multi-v2.1	Local Java; DJL tokenizer; ONNX Runtime; GLiNER span extraction; deterministic schema mapping	The subclass changes local modelId but inherits the parent remote fallback; archive an artifact hash before claiming exact weights.

The filenames indicate locally packaged model variants, but this study does not infer architectural causality from filenames alone. The observed comparison is between complete pipelines: model weights, quantization, prompt handling, server configuration, XML-to-JSON mapping, and evaluation implementation.

2.2.1 Generative extraction pipelines and the MoE distinction

ChatGPT, Gemma and Qwen are generative document-to-structure pipelines: a prompt conditions an autoregressive decoder, the decoder emits a textual representation, and a parser maps that representation to the controlled JSON contract. Gemma and Qwen are additionally sparse mixture-of-experts (MoE) systems. Their feed-forward sub-layers route each token to a small subset of experts, as formalized in Section 3.8. ChatGPT is treated as a hosted generative system because the supplied provenance does not expose an architecture manifest. The MoE label therefore applies to Gemma and Qwen only; Hybrid NLP and BERT MULTI are not MoE systems.

The evaluated Gemma and Qwen runs used the same llama-server command structure, 65,536-token context, four parallel slots, Vulkan execution, quantized GGUF checkpoints, and reasoning disabled. This reduces—but does not remove—implementation variation. Model weights, tokenizers, chat templates, active-parameter topology, quantization sensitivity and prompt-following behavior remain different. ChatGPT used the user-recorded configuration “model 5.6 on high”; provider build, token budget and sampling trace were not retained.

2.2.2 Hybrid NLP (Fast, Local): code-derived construction

The HNLP UI strategy resolves to HNlpExtractionStrategy with strategy identifier nlp. It requires a Markdown source, reduces the active JSON schema to NLP-facing top-level fields, applies the configured schema to the extractor, and merges NLP summary/raw fields into existing JSON. The extraction is not one statistical model. It is a composed

deterministic pipeline whose decisions arise from section aliases, heading penalties, language-aware dictionaries, synonym expansion, regular-expression patterns, fuzzy similarities, evidence scores, local normalization and final contract validation.

HNlpExtractor first cleans Markdown and suppresses boilerplate. It identifies candidate headings, assigns schema-aware section interpretations, and accumulates evidence from literal aliases, fuzzy similarity, dictionary membership, entity-pattern matches and positional context. It then extracts profile, experience, skills, education, certification and language evidence; normalizes seniority and category values; resolves ambiguous spans; deduplicates repeated evidence; and validates the canonical segment contract. OpenNLP tokenizer and sentence models are optional and lazily loaded. The implementation explicitly continues in regex/token-heuristic mode when those artifacts are absent. Because the benchmark archive contains no corresponding runtime log, OpenNLP activation is an unresolved treatment variable rather than an established cause of the measured result.

2.2.2.1 HNLP initialization, schema reduction, and resource binding

The proprietary HNLP method is reconstructed from HNlpExtractionStrategy and HNlpExtractor rather than inferred from its product label. At initialization, the strategy obtains the shared DictionaryService, initializes the local category predictor, loads configuration, reads the active JSON schema, loads skill resources, and attempts to initialize optional OpenNLP components. The active schema is not merely an output validator. Its property names, titles, descriptions, heading_aliases, nested item descriptions, x_segment_guidance and x_non_comparison_headings become executable extraction configuration. Consequently, changing descriptive schema metadata can change HNLP recognition even when the Java binary is unchanged.

For each schema field, keyword candidates are normalized to lower case, underscores and hyphens are converted to spaces, punctuation is removed, and both the complete phrase and informative component tokens are retained. Tokens shorter than three characters are excluded; longer plural tokens can contribute a singular form; and adjacent bigrams are added when sufficiently informative. The resulting schemaSectionKeywords, output-to-section map, section-specific penalties, segment guidance, and ignored-heading set define the vocabulary and permissible boundaries used by the parser. This makes HNLP configurable, but it also creates a versioning obligation: the schema, configuration, and dictionary files are part of the implemented method and must be archived with the binary for a reproducible experiment.

2.2.2.2 Dictionaries, multilingual aliases, and synonym expansion

HNLP uses several distinct lexical resources. RuntimeTagCatalog supplies canonical tag names, while hnlp_skills_en.txt and djl_skills_en.txt extend the skill vocabulary. Configured section, language, education, certification, seniority, category, and negative-evidence dictionaries are loaded from classpath resources; blank and comment lines are ignored. Localized dictionary resolution substitutes the detected language suffix where a localized file exists. These resources are not interchangeable: section aliases determine where evidence is admitted, skill dictionaries identify candidate concepts, normalization dictionaries map surface forms to canonical values, and negative resources suppress known ambiguities.

Multilingual support is implemented as a controlled expansion, not as unrestricted semantic translation. The extractor detects a document language and activates language-specific section keywords. Very short or structurally sparse text defaults conservatively to English because statistical language detection would be unstable. For non-default languages, and only when synonym expansion is enabled, DictionaryService.expandQueryTerms is invoked for each section term; nonblank novel alternatives are added to that language's section vocabulary. Available target languages are obtained from the dictionary service and derived section vocabularies are prepared lazily. Thus, a multilingual synonym can improve heading recall only if language detection, dictionary availability, and the expansion resource all agree. The mechanism cannot recover a language or domain term absent from those resources merely because a human would regard it as synonymous.

Skill-list recognition combines lexical membership with list structure. After bullet removal, text following a colon is considered separately and comma, semicolon, and pipe delimiters are used to identify candidate items. A list signal normally requires at least three items and at least three short items; it is strengthened by two or more dictionary hits, while a longer list of at least five short items can itself provide structural evidence. Exact phrase lookup is attempted first. Compact alphanumeric variants are indexed into

prefix buckets and compared with bounded edit distance only under length and prefix constraints. This limited fuzzy path tolerates selected compound-form variation without turning every near string into a skill.

Table 21B. HNLP dictionary, language, and synonym resources

Resource or mechanism	Construction in the proprietary implementation	Decision role	Failure implication
Schema-derived vocabulary	Field keys, titles, descriptions, heading_aliases, nested descriptions, singular tokens, and adjacent bigrams	Defines section candidates, output mapping, guidance, and penalties	A vague or incomplete schema changes recognition as well as serialization
Canonical skill catalog	RuntimeTagCatalog plus hnlp_skills_en.txt and djl_skills_en.txt	Phrase and compact-variant evidence for skills and skill-list headings	Out-of-vocabulary, misspelled, or domain-specific skills can be omitted
Localized dictionaries	Configured resource name resolved to a detected-language suffix when available	Language-, education-, certification-, seniority-, and category-specific recognition	Missing localized resources silently reduce coverage to available rules
Multilingual synonyms	DictionaryService expansion of section terms for non-default languages when enabled	Recovers known translated or synonymous headings	Incorrect language detection or absent synonym entries prevents recovery
Negative evidence	Ignored schema headings and configured suppressors	Prevents boilerplate or non-comparison content from entering a target section	Overbroad suppression can remove valid content
Compound fuzzy variants	Prefix-bucketed compact forms with bounded Levenshtein distance and length constraints	Recovers limited spelling/format variants of known skills	Severe OCR errors and unconstrained abbreviations remain unmatched

2.2.2.3 Heading search, fuzzy matching, and section state transitions

Section parsing is a stateful boundary-detection algorithm. Each cleaned line is evaluated for Markdown, layout, capitalization, punctuation, length, schema alias, dictionary, and content-evidence signals. Candidate heading text is normally constrained to 3-50 characters. Exact alias matches receive the strongest direct score (1.00), followed by prefix (0.95), containment (0.90), and restricted fuzzy (0.82) matches. A base acceptance threshold of 0.75 is adapted by structural penalty and heading confidence and bounded to the interval [0.45, 0.95]. Schema-specific penalties are then applied. A canonical transition requires the winning score to exceed the adaptive threshold and retain direct heading support; evidence alone cannot freely relabel every prose line as a heading.

The fuzzy heading test is deliberately narrow. The candidate and alias must have equal token counts and no more than three phrase tokens; each compared token must normally contain at least six characters; token-length difference is at most one; and bounded Levenshtein distance is at most one per token and one over the phrase. An exact match is handled by the exact branch rather than reported as fuzzy. This design protects precision, but it explains why an inventive heading, translated heading without an alias, abbreviation, or heavily corrupted OCR string can fail even when its intended meaning is obvious.

When a canonical heading is accepted, the parser flushes the preceding section, initializes the new section, may carry a compact introductory fragment, and protects the first following body line from immediate reclassification. A recognized non-comparison heading moves the parser into an ignored state until another valid boundary appears. Local Markdown subheadings that resemble role, employer, or education labels can remain inside the current section rather than terminate it. Ambiguous headings are scored with look-ahead content windows and competing section evidence; if dominance is insufficient, the line remains body text. Headerless content can still be assigned by content inference, but this recovery is weaker than an explicit boundary and is intentionally constrained to avoid turning every experience-like sentence into a complete experience section.

Table 21C. HNLP heading recognition and ambiguity controls

HNLP control	Implemented value or rule	Purpose	Sensitivity to test
Heading length	Normally 3-50 cleaned characters	Reject prose and visually implausible labels	Short acronyms and long narrative headings
Exact / prefix / contains	1.00 / 0.95 / 0.90	Prioritize direct schema and dictionary evidence	Alias ablation and collision tests
Restricted fuzzy score	0.82 after token and edit-distance gates	Recover minor heading spelling variation	Synthetic one- and two-edit perturbations
Base threshold	0.75	Minimum canonical-heading confidence before adaptation	Threshold sweep with boundary gold labels
Adaptive range	Bounded to [0.45, 0.95]	Combine layout confidence and structural penalty	Layout-stratified calibration
Fuzzy distance	At most one edit per token and one over the phrase	Limit false semantic equivalence	OCR and misspelling challenge set
Ignored heading state	Discard content until a later valid heading	Exclude references, declarations, or non-comparison blocks	False-suppression audit
Ambiguity resolution	Look-ahead evidence plus dominance margin	Avoid arbitrary section assignment	Persist runner-up scores and confusion labels

2.2.2.4 Content evidence, field extraction, normalization, and output contract

After segmentation, HNLP applies field-specific evidence models rather than a single universal classifier. Experience evidence includes date ranges, year-present patterns, and role/company structure; skills combine section context, list shape, and dictionary phrases; education uses degree, institution, and year cues; certification uses certification terms and temporal evidence; and spoken-language extraction combines a language lexicon with proficiency terms while distinguishing programming-language taxonomy. Schema-derived evidence phrases are weighted by informativeness, and a non-overlapping match selection prevents the same characters from being counted repeatedly. Competing section scores and a dominance margin provide an explicit abstention mechanism when the evidence does not uniquely support one field.

Field values are normalized and deduplicated only after candidate generation. Seniority and category mappings are therefore outputs of proprietary normalization dictionaries and rules, not direct observations from the CV. The strategy then merges extracted summary and raw fields into the existing JSON object and applies canonical contract validation. Structural validity at this final stage proves that a value has an allowed shape; it does not prove that the value was present in the PDF or assigned to the correct field. Academic error analysis must retain separate labels for boundary error, candidate miss, ambiguity rejection, normalization error, merge loss, and final validation loss.

Table 21A. Reconstructed HNLP execution algorithm

Phase	Reconstructed HNLP operation	Primary input	Required audit artifact
1. Initialize	Load configuration, schema, DictionaryService, skill catalogs, and optional OpenNLP	Code and versioned resources	Artifact manifest with hashes and load log
2. Normalize	Repair/clean Markdown and suppress boilerplate while preserving line structure	Converted CV text	PDF-to-Markdown lineage and normalized-text snapshot
3. Detect language	Select language-aware resources; default sparse/short evidence conservatively	Document text and heading evidence	Detected language and confidence/reason
4. Build vocabulary	Derive schema keywords, aliases, synonyms, penalties, and ignored headings	Schema plus dictionaries	Expanded vocabulary and schema hash
5. Parse sections	Score explicit, fuzzy, ambiguous, ignored, and headerless boundaries	Normalized lines	Per-line candidate scores and state transitions
6. Extract evidence	Apply field-specific regex, dictionary, list, date, and contextual rules	Sectioned text	All candidates including rejected competitors
7. Resolve	Use dominance, normalization, deduplication, and ambiguity controls	Candidate evidence	Winner/runner-up scores and rejection reason
8. Assemble	Merge summary/raw fields and validate the canonical JSON contract	Resolved field values	Field-to-source lineage and validation report

2.2.3 NLP BERT MULTI (DJL): code-derived construction

The selected BERT strategy resolves to DjlTokenClassificationMultilingualExtractionStrategy with identifier nlp_bert_multi. The preset points to models/djl/gliner_multi-v2.1 and describes a multilingual GLiNER model executed through a Java DJL tokenizer and ONNX Runtime. Unlike HNLP, it retains the full schema. The schema's x_gliner metadata is transformed into dynamic entity labels, section restrictions, grouping rules and label-to-field mappings. The method is therefore better described as schema-conditioned span extraction than as ordinary fixed-label BERT token classification.

The runtime constructs broad section chunks and focused line-context chunks, limits a chunk to 220 words with 30-word overlap, queries at most four labels per inference prompt, and accepts spans at the configured default confidence threshold of 10%. DJL tokenization and ONNX inference produce candidate spans; sigmoid scores are decoded, sanitized, occasionally relabelled, deduplicated across overlap, constrained by schema rules, grouped into field objects, and assembled with Markdown summary/backfill before canonical validation. Every stage can increase precision while reducing recall. Consequently, the reported BERT result is a property of the encoder, chunker, label prompt, threshold, overlap resolver, mapper and backfill policy jointly.

A provenance inconsistency must remain visible: the multilingual subclass overrides defaultModelId but does not override the inherited remoteModelId. The supplied record therefore links the local multi path with an onnx-community/gliner_medium-v2.1 fallback. Without an archived model manifest and cryptographic hash, it is not academically defensible to assert which weights executed. The report treats 'BERT MULTI' as the selected application strategy and records the unresolved artifact identity as a limitation.

2.2.3.1 Schema-to-query planning and multilingual model binding

The proprietary BERT path begins before neural inference. SchemaDrivenEntityPlanBuilder recursively traverses the active schema and interprets field-level x_gliner metadata as an executable query plan. Each rule can define one or more textual labels, allowed sections, destination path, assignment mode, split expression, maximum item count, grouping window, group key, deduplication scope, line mode, span-expansion policy, section-chunk switch, score floor, and textual acceptance constraints. Child fields inherit their parent's section context unless they override it. When an explicit label is absent, a label can be derived from the schema property name. The schema is therefore part of the learned-system interface: it changes what the model is asked to find and how accepted spans become JSON.

The user-facing name 'NLP BERT MULTI (DJL)' identifies an application preset, not a sufficient scientific model identity. The preset's local path is models/djl/gliner_multi-v2.1, inference is executed through DJL and ONNX Runtime, and the query mechanism is GLiNER-style textual label conditioning. Because the inherited remote fallback can refer to a different medium-v2.1 identifier, a reproducible study must record the resolved ONNX graph, tokenizer files, model configuration, native runtime provider, numerical precision, and hashes before processing the first document. Without that manifest, the Java control flow is reproducible but the realized neural function is not.

2.2.3.2 Section chunks, line-context chunks, and label batching

The runtime generates two complementary views of the same Markdown. Broad section chunks favor recall by exposing larger contextual regions. Focused line-context chunks favor precision by centering inference on lines selected by schema and section evidence; they retain focus offsets and can request line-scoped decoding or controlled span expansion. Query rules can disable section chunks when broad context would add noise. The default word window is 220 words with 30-word overlap, and overlap is constrained relative to the window so that progress remains well-defined. This dual-view construction explains how a candidate can be produced more than once and why later deduplication is an essential component of measured performance.

Schema-derived labels are partitioned into batches of at most four labels per prompt. This cap reduces tensor and prompt complexity, but label-conditioned scores need not be invariant to batch composition. Semantically adjacent labels can compete differently depending on which alternatives appear together. A rigorous ablation should therefore hold the model and text fixed while varying label order and batch grouping. The reported benchmark did not archive such an ablation, so apparent BERT errors cannot be assigned exclusively to representation quality; they may also arise from query planning or batching.

2.2.3.3 GLiNER span scoring, fallback acceptance, and source offsets

For each chunk-label batch, the runtime concatenates prompt words and document words, provides text_lengths, builds candidate span indices and masks up to the model's supported width, and executes the ONNX graph. Both three- and four-dimensional logit layouts are handled. A sigmoid converts each span-label logit to a score, and the default global acceptance threshold is 0.10. Accepted token spans are projected back through chunk offsets to source character positions. Correct offset reconstruction is methodologically important: an apparently correct entity string is not auditable unless it can be located in the converted source and, ultimately, on the PDF.

For supported line-scoped modes, a limited fallback may retain one dominant span when no candidate crosses the global threshold. Its minimum score is max(0.05, 0.6 times the configured threshold), after which optional span expansion can replace a compact model span with a line fragment prescribed by the query rule. This fallback is proprietary decision logic, not a property of BERT or GLiNER. It may improve recall for weak but localized evidence, while also changing the meaning of the neural score because the emitted text can exceed the tokens originally scored by the encoder. Evaluation should distinguish the raw predicted span from any expanded output span.

2.2.3.4 Sanitization, relabelling, overlap resolution, and deduplication

DjlEntitySanitizer applies plausibility checks to decoded entities and can occasionally relabel a candidate before mapping. The runtime then merges duplicates created by chunk overlap and the section/line dual view. Offset tolerance is max(60, min(260, 8 times chunkOverlapWords plus 40)); at the default overlap it is 260 characters. For strongly overlapping candidates, a new candidate normally replaces a retained candidate only when its score is more than 0.08 higher. If labels and sections agree and one span nests the other, a longer candidate can replace the shorter when its score plus 0.06 is at least the retained score. These constants encode a preference for stability and longer contextual spans, and they can alter precision and recall independently of the neural model.

Raw, sanitized, and merged candidate counts are recorded by the runtime, but the benchmark workbooks do not expose a complete candidate-level lineage. Consequently, a missing final value has at least four distinct explanations: the encoder produced no span, the score did not cross an acceptance rule, sanitization rejected or relabelled it, or deduplication replaced it. Collapsing these mechanisms into 'BERT failed' would be academically incorrect and would prevent targeted improvement.

Table 22D. BERT MULTI decoding and duplicate-resolution controls

Control	Implemented default or rule	Function in final output	Required sensitivity analysis
Chunk window	220 words	Bounds contextual evidence available to one inference call	Grid over short, default, and longer windows
Chunk overlap	30 words; overlap bounded relative to window	Recovers boundary entities but creates duplicates	Boundary-recall versus duplicate-rate curve
Labels per prompt	At most 4	Controls label competition and runtime cost	Repeated label-order and batch-composition trials
Global score threshold	0.10 after sigmoid	Defines broad neural candidate acceptance	Per-label precision-recall and calibration curves
Line fallback floor	max(0.05, 0.6 x threshold)	Retains one dominant localized candidate below global threshold	Report fallback-only precision and recall
Duplicate offset tolerance	max(60, min(260, 8 x overlapWords + 40))	Associates candidates from overlapping views	Tolerance ablation with source-span gold data
Score replacement margin	Candidate score greater by 0.08	Allows a stronger duplicate to replace retained evidence	Error analysis of replaced true positives
Nested-span margin	Longer score + 0.06 at least retained score	Favors a longer same-label/section span when close in score	Short-versus-long span exactness analysis

2.2.3.5 Schema mapping, grouping, splitting, and canonical assembly

DjlSchemaMapper does not copy every decoded span directly to JSON. It first resolves accepted labels and enforces the rule-specific minimum score and allowed-section set. Text constraints can require minimum or maximum character and word counts, require a regular-expression match, or reject a regular-expression match. Near duplicates with the same normalized text and overlapping or nearby positions are collapsed in favor of the stronger entity. These filters can improve precision, but a correct neural candidate can disappear because its section, surface form, or length violates a handcrafted mapping rule.

Three principal assignment modes are implemented. set_first writes the first acceptable scalar; set_joined_text joins evidence; and append_object creates array objects. A split_pattern can turn one entity into multiple items, bounded by split_max_items (with a default maximum of 50). For skills[].skill_name_v, the compatibility rule splits comma-

separated text and performs document-level deduplication. Object fields can be grouped by proximity around a group_key using a default 220-character window, after which missing subfields are merged into the same object. Template variables such as text, label, confidence, start, and end can render values; integer extraction can take the first one- to three-digit sequence. Each of these operations is a proprietary semantic transformation that must be evaluated separately from entity detection.

The mapper is followed by summary/backfill and canonical validation. Backfill can improve apparent completeness without demonstrating that GLiNER found the field, whereas strict constraints can reduce apparent completeness despite a correct candidate. A scientifically adequate benchmark should therefore report at least four nested outcomes: encoder candidate recall, accepted-candidate recall, mapped-field recall, and final-JSON recall, with a source span and transformation history for every final value.

Table 22E. Schema-driven BERT mapping and aggregation controls

Schema control	Implemented behavior	Benefit	Potential proprietary error
x_gliner labels	Explicit textual labels or property-name-derived fallback labels	Adapts one encoder to the active ontology	Vague or overlapping labels alter candidate competition
Allowed sections	Inherited or field-specific section restrictions	Suppresses implausible field assignments	Heading/section error rejects a correct span
Text constraints	Character/word bounds and require/reject regex rules	Removes malformed candidates	Rule excludes valid linguistic variation
set_first / set_joined_text	Scalar-first or joined-evidence assignment	Controls cardinality	Ordering can hide a later, stronger value
append_object	Create array items and merge nearby subfields	Builds structured experience/education objects	Proximity can join unrelated facts or split one fact
split_pattern	Regex split bounded by split_max_items	Expands list-like spans such as comma-separated skills	Delimiter ambiguity changes item count
group_window_chars	Proximity grouping; default 220 characters	Associates fields from one local record	Layout flattening can place unrelated records nearby
dedupe_scope	Field or document-level signature deduplication	Controls repeated items	Legitimate repeated roles or credentials can collapse
Templates	Render text, label, confidence, and source offsets	Supports schema-specific value construction	Rendered value may differ from the scored span
Backfill and validation	Complete selected summaries and enforce canonical shape	Improves usability and structural validity	Conceals whether evidence originated from the encoder

2.2.3.6 End-to-end reconstructed BERT MULTI algorithm

Table 22C. Reconstructed BERT MULTI execution algorithm

Phase	Reconstructed BERT MULTI operation	Primary input	Required audit artifact
1. Bind runtime	Resolve ONNX model, tokenizer, DJL/ONNX provider, and multilingual preset	Model directory and runtime	Immutable model/runtime manifest
2. Build plan	Compile recursive x_gliner metadata into labels, sections, constraints, and destination rules	Full JSON schema	Serialized query plan and schema hash
3. Construct views	Create broad section chunks and focused line-context chunks	Markdown and query plan	Chunk text, offsets, mode, and source lineage
4. Batch labels	Partition query labels into prompts of at most four	Chunk and label set	Label order and batch membership
5. Infer spans	Tokenize, enumerate masked spans, run ONNX, sigmoid-score label/span pairs	Prompt plus chunk	Raw logits/scores and token-to-character offsets
6. Accept/sanitize	Apply threshold or limited line fallback, span expansion, plausibility checks, and relabelling	Raw entities	Reason code for every accepted/rejected entity
7. Merge candidates	Resolve overlap and nested duplicates using offset and score margins	Section and line candidates	Replacement graph and candidate counts
8. Map schema	Apply sections, constraints, split, cardinality, grouping, templates, and dedupe scope	Merged candidates and query rules	Candidate-to-field transformation lineage
9. Assemble	Apply summary/backfill and canonical validation	Mapped fields and Markdown	Backfill provenance and final source-span audit

2.2.4 Conditions for reliable Hybrid NLP operation

HNLP works optimally when the document makes its semantic structure visible through stable lexical and layout cues. The ideal CV contains explicit section headings—such as “Skills”, “Technical Skills”, “Professional Experience”, “Education”, “Certifications” and “Languages”—followed by the content belonging to that section. Conventional headings allow the alias and fuzzy-heading mechanisms to establish reliable section boundaries. Equivalent headings such as “Core Competencies” may work when represented in the configured aliases or synonym resources. A novel heading such as “My Toolbox”, an omitted heading, or a heading detached from its content weakens this signal.

The expected consequence of a missing or unrecognized heading is not limited to one metric. Recall may fall because the following lines are never admitted to the intended section. Precision may also fall if the lines are attached to an incorrect neighboring section. If the extractor retains the text but assigns it to the wrong field, overall textual coverage can remain high while field-level precision and recall both deteriorate. The effect should therefore be measured as section-detection recall, boundary accuracy and field-assignment accuracy rather than described only as a generic precision loss.

Lexical quality is the second major condition. Dictionary and regular-expression methods depend on recognizable surface forms. Correctly spelled terms that match the skill, language, education, certification and synonym resources are the most reliable inputs. Minor heading variation may be recovered by fuzzy matching, and known synonyms may recover semantic variants, but this does not make the pipeline semantically equivalent to an LLM. OCR corruption, split words, unusual hyphenation, abbreviations absent from the dictionary, or misspellings such as “Pyhton” for “Python” can remove the exact evidence used for subject recognition. A misspelling will not necessarily destroy the complete extraction—other section and context signals may survive—but it can completely suppress that particular dictionary-based recognition when no fuzzy, synonym or pattern rule covers it.

HNLP further assumes that PDF-to-Markdown conversion preserves line order, heading boundaries, bullet lists and separation between columns. Flattened two-column CVs can interleave unrelated lines; decorative headings may disappear; and tables may lose row associations. Since HNLP uses local structural context rather than a global semantic reconstruction, these conversion errors can propagate directly into section and entity decisions. The pipeline is also strongest when facts are explicit. It can recognize “Java” in a skills list, but it should not infer Java expertise solely from a project description unless a configured rule treats that evidence as a skill. This conservative behavior protects factual precision while limiting semantic recall.

Language and domain coverage must match the deployed dictionaries. A multilingual CV may use translated headings, local degree names, sector-specific abbreviations or inflected skill terms not represented in the resources. The implementation contains language detection and multilingual dictionaries, but optimal performance still requires versioned, domain-tested lexicons. Optional OpenNLP tokenizer and sentence models must also correspond to the document language and be demonstrably loaded. Otherwise, the effective treatment is the regex/token-heuristic fallback.

$$A_{HNLP,d} = \omega_h H_{structure} + \omega_x X_{text} + \omega_d D_{coverage} + \omega_l L_{layout} + \omega_e E_{explicit} \quad \text{(C1)}$$

> $A_{(HNLP,d)}$ is a conceptual applicability score for document d: recognizable heading structure H_structure, clean text X_text, dictionary coverage D_coverage, preserved layout L_layout and explicit evidence E_explicit increase suitability. The ω coefficients are not fitted by the benchmark and must not be interpreted as measured production weights.

Table 22A. Conditions for reliable HNLP extraction

HNLP condition	Optimal document state	Likely degradation when violated	Observable diagnostic	Mitigation
Section headings	Conventional or configured aliases followed by their content	Lower section recall; boundary errors; possible wrong-field precision loss	Heading-detection recall and field-confusion matrix	Expand aliases; fuzzy heading calibration; fallback section classifier
Spelling and OCR	Correct terms, intact words and punctuation	Dictionary/regex misses; fragmented entities; lower recall	Unknown-token, OCR-error and unmatched-dictionary rates	Spell/OCR normalization with source preservation; typo variants; character-level matching
Dictionary coverage	Domain, language and abbreviation vocabulary represented	Unseen concepts omitted or misclassified	Held-out-vocabulary recall by domain and language	Versioned lexicons; controlled synonym expansion; human review queue
Layout preservation	Reading order, bullets, headings and columns survive Markdown conversion	Interleaved sections and incorrect boundaries	Page-image versus Markdown alignment audit	Layout-aware conversion; retain page/line coordinates
Explicit wording	Facts are directly stated near compatible headings	Implicit skills and cross-sentence relations are omitted	Explicit-versus-implicit gold annotation	Contextual fallback model; do not silently infer unsupported facts
Ambiguity	Terms have clear CV-specific meaning	False positives for ambiguous tokens such as short technologies or acronyms	Context-conditioned precision by term	Negative dictionaries, context windows and calibrated abstention
OpenNLP resources	Correct language models loaded and logged	Fallback tokenization/sentence segmentation may differ	Runtime model-load manifest	Pin and hash model artifacts; report regex-only and OpenNLP modes separately

2.2.5 Conditions for reliable BERT MULTI operation

BERT MULTI is less dependent than HNLP on exact dictionary membership and conventional headings because contextual token representations can recognize a span from surrounding words. It is not independent of document structure. The supplied runtime derives labels and section restrictions from the schema and constructs both broad section chunks and focused line-context chunks. Recognizable headings therefore still improve context selection and reduce label ambiguity, even though their absence need not eliminate recognition in the way it can for a section-led heuristic pipeline.

Its first condition is faithful text and tokenizer alignment. The PDF/Markdown text must retain the characters, word order and offsets expected by the tokenizer paired with the executed ONNX model. OCR substitutions, broken Unicode, merged columns and inserted line breaks can split an entity into unfamiliar subwords or invalidate character offsets. BERT-style encoders are generally more tolerant of lexical variation than exact dictionaries because subword representations share information across forms. They are not immune to severe misspellings, OCR corruption or a tokenizer/model mismatch; these can lower span scores or make the reconstructed span incorrect.

The second condition is alignment between the target and span extraction. GLiNER is well suited to explicit, contiguous, entity-like evidence such as a certificate name, language, employer or technology. It is less naturally matched to a target that requires copying an entire long section, linking evidence across distant lines, inferring seniority from multiple jobs, or constructing a normalized category from the complete career history. Those tasks depend on chunk aggregation, mapping and backfill rather than the encoder alone. BERT can therefore detect correct local entities while the final JSON remains incomplete.

The label descriptions generated from x_gliner metadata must be distinct, semantically informative and consistent with the training distribution of the actual model artifact. Vague or overlapping labels—such as “experience”, “role” and “profile” without operational definitions—can produce competing spans. Querying no more than four labels per prompt reduces prompt size but can make results sensitive to how labels are batched. The local model path, tokenizer and ONNX graph must refer to the same model revision; the unresolved multilingual-versus-medium fallback provenance is consequently a substantive condition, not a documentation detail.

Chunking and calibration determine the precision–recall operating point. A 220-word chunk with 30-word overlap works best when the evidence needed for one decision fits inside a chunk and entities do not straddle damaged boundaries. Long sections, cross-section references and multi-line certificates may lose context. The 10% confidence threshold is permissive at the encoder stage, but later sanitizer and schema rules can restore precision by discarding candidates; they can also remove true positives. Optimal operation therefore requires separate measurement of pre-threshold candidate recall, post-threshold recall, post-deduplication recall and final mapped recall.

Finally, the output schema must be compatible with the label-to-field mapper. Cardinality, grouping, allowed sections and normalization rules must not reject valid spans or attach them to the wrong object. The application needs sufficient memory and a deterministic ONNX execution environment, but computational success alone is not semantic success. Model confidence must be calibrated on representative CV languages, layouts and categories, and uncertain or derived fields should permit abstention or human review.

$$A_{BERT,d} = \nu_t T_{alignment} + \nu_s S_{span} + \nu_q Q_{label} + \nu_c C_{context} + \nu_m M_{schema} \quad \text{(C2)}$$

> $A_{(}$BERT,d$_{)}$ summarizes applicability through tokenizer/model alignment T_alignment, span-like target structure S_span, label quality Q_label, sufficient chunk context C_context and correct schema mapping M_schema. The ν coefficients are conceptual and must be estimated through controlled ablations.

Table 22B. Conditions for reliable BERT MULTI extraction

BERT condition	Optimal document/system state	Likely degradation when violated	Observable diagnostic	Mitigation
Text and tokenizer integrity	Clean Unicode text, correct order/offsets, tokenizer matched to ONNX model	Fragmented tokens, low span scores or wrong reconstructed offsets	Tokenizer round-trip and page-span alignment tests	Pin tokenizer/moc hashes; normalize carefully while retaining source o
Language/model alignment	Executed weights cover the document language and terminology	Reduced recall and unstable confidence on unsupported languages	Language-stratified calibration and artifact manifest	Verify multilingua artifact; route unsupported langu to another model
Target is span-like	Explicit contiguous entities or short relations	Long sections, dispersed evidence and derived judgments remain incomplete	Score encoder candidate recall separately from final JSON recall	Section reconstruc relation layer or generative/determ fallback
Label quality	Distinct, descriptive schema-derived GLiNER labels	Competing labels, wrong field assignment or missed entities	Per-label confusion and prompt-batch sensitivity	Rewrite labels; ontology review; label-batch experi
Chunk context	Relevant evidence fits within 220 words and survives 30-word overlap	Boundary omissions and loss of distant context	Error rate by distance to chunk boundary	Chunk/overlap gri section-adaptive windows; long-co alternative
Threshold calibration	Confidence threshold calibrated on representative CVs	High threshold lowers recall; low threshold raises candidate noise	Precision–recall and reliability curves before and after mapping	Per-label threshol calibrated abstenti
Deduplication and mapping	Schema constraints preserve true candidates and group fields correctly	Correct encoder spans disappear or are attached to wrong objects	Stage-level lineage from candidate to final JSON	Persist candidates mapper unit tests; reversible provena
Runtime provenance	Exact ONNX, tokenizer, DJL and runtime artifacts are hashed	Results cannot be reproduced or attributed to the claimed model	Immutable execution manifest	Disable ambiguou fallback or record resolved artifact b inference

Table 20. Five-system architecture, inference, and provenance matrix

System	Computational family	Primary extraction mechanism	Post-processing	Reasoning mode
ChatGPT	Hosted generative transformer	Prompt-conditioned autoregressive schema generation	JSON parsing and schema validation	Provider-specific; recorded as high
Gemma	Sparse generative MoE	Autoregressive verbatim-block generation	XML/JSON mapping and evaluation	Explicitly off
Qwen	Sparse generative MoE	Autoregressive verbatim-block generation	XML/JSON mapping and evaluation	Explicitly off
Hybrid NLP	Symbolic/statistical local pipeline	Section, regex, dictionary and fuzzy evidence	Normalization, merge and validation	Not applicable
BERT MULTI	Bidirectional span encoder (GLiNER)	Schema-conditioned entity-span scoring	Thresholding, deduplication, mapping, backfill	Not applicable

Table 21. Java-derived HNLP configuration and epistemic status

HNLP component	Established from Java	Unresolved benchmark condition
Input	Markdown source is required.	PDF-to-Markdown engine/version is not embedded in the combined report.
Schema	Reduced top-level NLP schema and schema-derived section aliases.	Exact deployed schema snapshot should be hash-locked per run.
Evidence	Regex, dictionaries, fuzzy headings, synonyms and contextual scoring.	Runtime dictionary/synonym versions require an immutable manifest.
OpenNLP	Tokenizer/sentence models are optional and lazily loaded.	No run log proves whether optional models were active.
Output	Raw/summary merge, normalization and canonical validation.	Intermediate decisions are not preserved in the benchmark workbook.

Table 22. Java-derived BERT MULTI configuration and epistemic status

BERT component	Configured value or behavior	Validity concern
Model path	models/djl/gliner_multi-v2.1	Executed artifact hash is absent.
Remote fallback	Inherited parent remoteModelId	May not identify the same multilingual artifact.
Runtime	DJL 0.33.0; ONNX Runtime 1.22.0 defaults	Resolved jar hashes and native provider are not archived.
Chunking	220 words; 30-word overlap	Boundary sensitivity was not ablated.
Label batches	Maximum four labels per inference prompt	Batch composition may alter scores.
Threshold	10% default	No precision–recall calibration curve is available.
Mapping	Schema-driven constraints, grouping and backfill	Mapper errors are confounded with encoder errors.

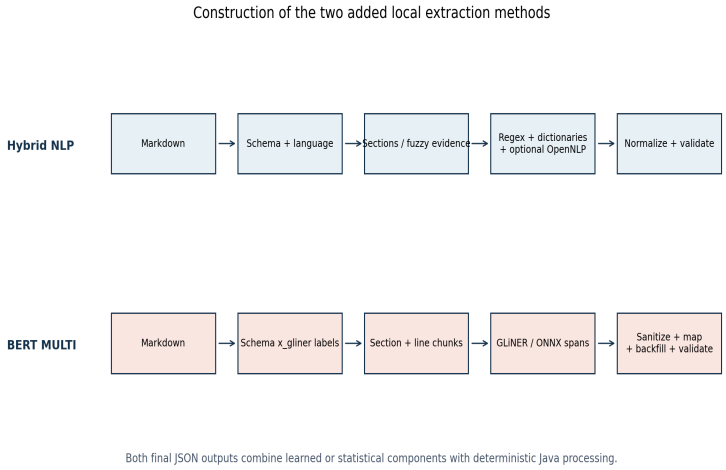

Figure 16. Code-derived architecture of Hybrid NLP and BERT MULTI.

Source: reconstruction from the supplied edoc-260 Java classes. Dashed conceptual boundaries denote stages, not measured causal effects.

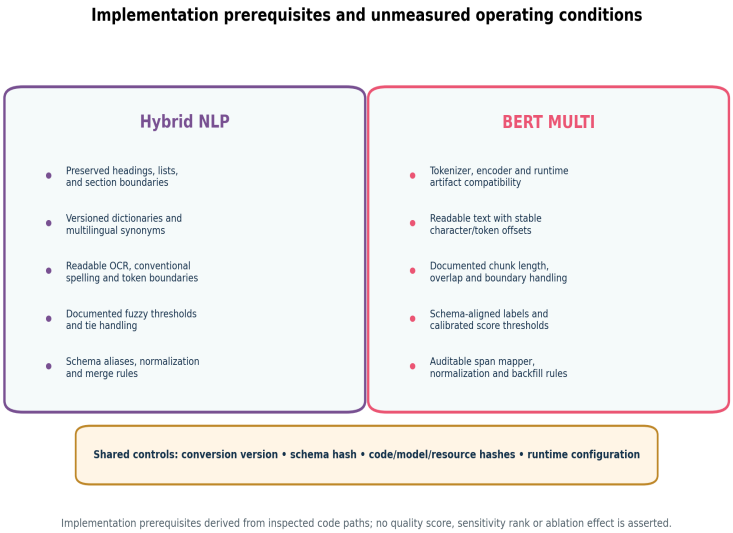

Figure 17. Implementation prerequisites and unmeasured operating conditions.

The diagram records code-derived prerequisites only. It contains no quality score, sensitivity ranking or estimated ablation effect.

2.3 Extraction contract and schema

The local-model system and user prompts require exactly one XML block per requested headline, copied from the source without rewriting, invention, normalization, translation, or correction. ChatGPT prompt.2 requires section-aware, high-recall extraction into eight controlled segments. All unwrapped outputs are validated against the supplied aiextract.json schema using JSON Schema Draft 2020-12 [7]. Schema validity is treated as structural integrity and is not conflated with factual correctness.

2.4 Evaluation layers

Layer 1 evaluates archive presence, JSON parsing, wrapper accessibility, schema validity, deterministic metadata population, and eight-segment population. Layer 2 applies one lexical implementation to all three systems: exact normalized object containment, four-token PDF-support precision, broad PDF lexical coverage, and agreement with ChatGPT. Layer 3 recomputes local-model atomic metrics from TP, Partial, FP and FN counts in the 48 canonical category workbooks. The layers are reported separately because they answer different questions and have different validity constraints.

2.5 Metric definitions

Partial facts receive a fixed weight of 0.5, as specified by the evaluation prompt. The principal formulas are:

- Weighted precision = (TP + 0.5 × Partial) / (TP + Partial + FP)
- Weighted recall = (TP + 0.5 × Partial) / (TP + Partial + FN)
- Weighted F1 = 2 × precision × recall / (precision + recall)
- Completeness = (TP + Partial) / (TP + Partial + FN)
- Unsupported rate = FP / (TP + Partial + FP)

2.6 Statistical analysis

The primary paired comparison uses the 24 occupational categories, consistent with recommendations for comparing learning systems over multiple datasets [3]. The analysis reports the mean and median paired weighted-F1 difference, a 50,000-resample percentile

bootstrap interval over categories [8], a paired t-test, Wilcoxon signed-rank test, exact sign test, and paired-standardized effect size (Cohen's dz). Pearson and Spearman coefficients quantify whether the two evaluators produce similar category ordering. Document-level tests are explicitly exploratory because category workbooks use heterogeneous atomization and matching implementations. Pipeline-success intervals use Wilson score intervals.

2.7 Reproducibility controls

The audit preserves source paths and SHA-256 hashes for prompts, schema, and canonical workbooks; reports document-ID overlap; records unmatched files; avoids silent JSON repair; and uses a fixed random seed (20260723) for bootstrap analysis. The statistical appendix contains all values used in the figures.

3. Formal Mathematical Framework and MoE Theory

This section makes the evaluation model explicit. Symbols are defined before use, all ratios identify their denominator, and partial credit is separated from strict correctness. The purpose is not cosmetic formalism: an unexamined denominator can change the substantive conclusion, and a sparse architecture cannot be interpreted from its parameter count alone.

3.1 Units, sets, and evidence relation

Table 12. Mathematical symbols and units of analysis

Symbol	Definition
N	Number of source documents (2,484).
M	Compared extraction systems: ChatGPT, Gemma, and Qwen.
C	Set of 24 occupational category strata.
S	Set of semantic segments or field families.
$G_{i,s}$	PDF-supported atomic facts for document i and segment s.
$\hat{G}_{i,s}^{(m)}$	Facts predicted by model m for document i and segment s.
TP, Q, FP, FN	True, partial, unsupported, and omitted atomic facts.
α	Partial-credit weight; α = 0.5 in the canonical weighted metrics.

$$D = \{d_i | i = 1, \ldots, N\}, N = 2484 \quad (1)$$

$$G_{i,s} \subset T, \widehat{G}_{i,s}^{(m)} \subset T \quad (2)$$

> **Interpretation.** The PDF-supported set and the predicted set are conceptually distinct. ChatGPT agreement can diagnose disagreement, but only the PDF-supported evidence relation can establish factual support.

3.2 Strict precision, recall, and harmonic mean

For model m and segment s, strict precision asks what fraction of emitted atomic claims is fully supported; strict recall asks what fraction of PDF-supported facts is fully recovered. The harmonic mean penalizes systems that optimize only one side.

$$P_{m,s} = \frac{TP_{m,s}}{TP_{m,s} + FP_{m,s}} \quad (3)$$

$$R_{m,s} = \frac{TP_{m,s}}{TP_{m,s} + FN_{m,s}} \quad (4)$$

$$F_{1,m,s} = \frac{2P_{m,s}R_{m,s}}{P_{m,s} + R_{m,s}} \quad (5)$$

Strict precision is undefined when a model emits no claims and strict recall is undefined when a segment contains no applicable PDF-supported facts. Such cases must not be silently converted to perfect or zero performance. The canonical workbooks provide pooled counts; the report therefore distinguishes count-weighted micro aggregation from unweighted category averages.

3.3 Partial-credit model

A partially correct fact can preserve useful evidence while omitting a qualifier, date, entity, or relation. Let α be the declared credit allocated to a partial item. The canonical analysis uses one-half credit. This is a normative scoring choice and must be subjected to sensitivity analysis rather than treated as a physical constant.

$$TP_{m,s}^{(w)} = TP_{m,s} + \alpha Q_{m,s}, \alpha = \frac{1}{2} \quad (6)$$

$$P_{m,s}^{(w)} = \frac{TP_{m,s} + \alpha Q_{m,s}}{TP_{m,s} + Q_{m,s} + FP_{m,s}} \quad (7)$$

$$R_{m,s}^{(w)} = \frac{TP_{m,s} + \alpha Q_{m,s}}{TP_{m,s} + Q_{m,s} + FN_{m,s}} \quad (8)$$

$$F_{1,m,s}^{(w)} = \frac{2P_{m,s}^{(w)} R_{m,s}^{(w)}}{P_{m,s}^{(w)} + R_{m,s}^{(w)}} \quad (9)$$

> **Interpretation.** Weighted F1 is not the arithmetic mean of weighted precision and weighted recall. It is their harmonic mean and therefore remains sensitive to the weaker component.

3.4 Completeness, unsupported content, and schema quality

$$C_{m,s} = \frac{TP_{m,s} + Q_{m,s}}{TP_{m,s} + Q_{m,s} + FN_{m,s}} \quad (10)$$

$$U_{m,s} = \frac{FP_{m,s}}{TP_{m,s} + Q_{m,s} + FP_{m,s}} \quad (11)$$

$$S_{m,i} = \frac{\sum_{k=1}^{K} I\left(z_{m,i,k} \neq null\right)}{K} \quad (12)$$

Completeness gives full document-coverage credit to partial items, whereas weighted recall gives only α credit. Unsupported rate measures emitted unsupported claims. Schema completeness counts required populated fields, not factual correctness. A syntactically complete object may still be factually incomplete or wrong.

$$V_m = \frac{\sum_{i=1}^{N} I\left(\text{parsed} \wedge \text{schema-valid}\right)}{N} \quad (13)$$

3.5 Direct lexical support proxies

The common three-model layer applies a single normalization and matching implementation. It is intentionally lexical. Let ν map a string to normalized tokens and let Ω_4 denote its four-token windows. Exact-object support and lexical coverage are defined below.

$$E_{m,i} = \frac{\sum_{x \in \widehat{G}_{m,i}} I\left(\nu(x) \subseteq \nu\left(d_i\right)\right)}{|\widehat{G}_{m,i}|} \quad (14)$$

$$L_{m,i} = \frac{|\Omega_{4,i} \cap \Omega_{4,m,i}|}{|\Omega_{4,i}|} \quad (15)$$

> **Interpretation.** Lexical support is a reproducible proxy for extractiveness. It cannot detect a semantically wrong relation assembled from words that all occur somewhere in the PDF, and full-PDF lexical coverage is not semantic recall.

3.6 Paired comparison and uncertainty

$$\Delta_c = F_{1,Qwen,c}^{(w)} - F_{1,Gemma,c}^{(w)} \quad (16)$$

$$\bar{\Delta} = \frac{\sum_{c=1}^{C} \Delta_c}{C}, C = 24 \quad (17)$$

$$t = \frac{\bar{\Delta}}{\frac{s_\Delta}{\sqrt{C}}}, d_z = \frac{\bar{\Delta}}{s_\Delta} \quad (18)$$

The bootstrap interval resamples the 24 observed category differences. It describes sensitivity to those strata, not uncertainty over all possible occupations. The paired t-test assumes approximately normal differences; the signed-rank and sign tests relax parts of that assumption but still do not remove evaluator confounding.

3.7 RAG scoring and decision utility

A retrieval score should distinguish relevance, field compatibility, and evidentiary support. Let q_j be criterion j, e a source-aligned evidence span, A_{js} a compatibility matrix, and g(e) an evidence-quality function. The following form makes that policy explicit.

$$\text{Score}\left(d_i, q\right) = \sum_j w_j \max\left(A_{j,s}\text{sim}\left(q_j, e\right) g(e)\right) - \lambda r_i \quad (19)$$

$$U = b_{TP}n_{TP} - c_{FN}n_{FN} - c_{FP}n_{FP} - c_{lat}T \quad (20)$$

> **Interpretation.** A model ranking changes when the cost of omitting a qualified candidate differs from the cost of presenting an unsupported claim. No single F1 value specifies that policy.

3.8 Sparse Mixture-of-Experts architecture: mathematical model

Both local systems are sparse MoE models. A router maps each token state to expert scores; only the highest-ranked experts are activated, usually alongside a shared expert. Sparse activation increases total representational capacity without applying all expert parameters to every token. The mathematical form follows the foundational sparsely gated layer and later Transformer-scale implementations [9–10,19].

Table 13. Published architecture characteristics of the evaluated model families

Model	Total	Active	Experts	Layers	Native context	Sequence architecture
Gemma 4 26B A4B	25.2B	3.8B	128 routed; 8 active + 1 shared	30	256K	Hybrid local/global attention
Qwen3.6 35B A3B	35B	3B	256; 8 routed + 1 shared	40	262,144	Gated DeltaNet/full-attention hybrid

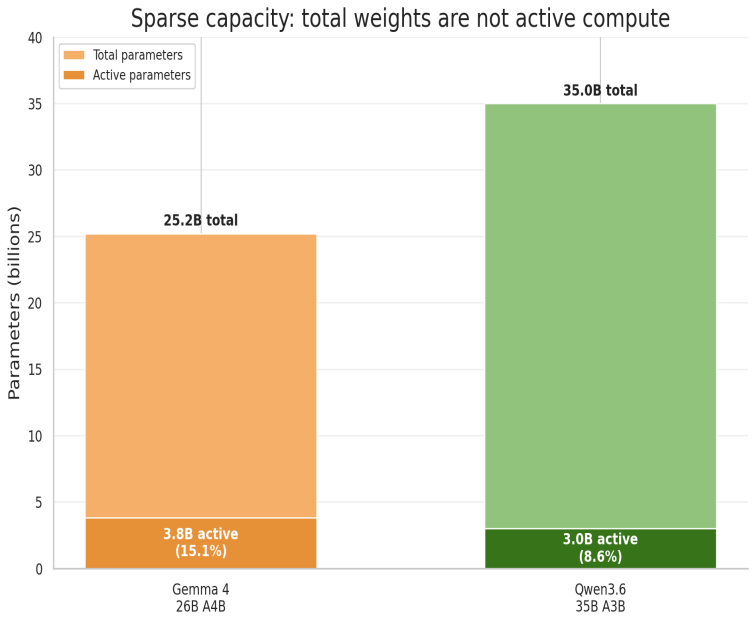

Figure 11. Total and active parameter counts for the two evaluated MoE families.

> *Note. Published architecture counts. Active-parameter ratios are descriptive and do not equal end-to-end FLOP or latency ratios. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

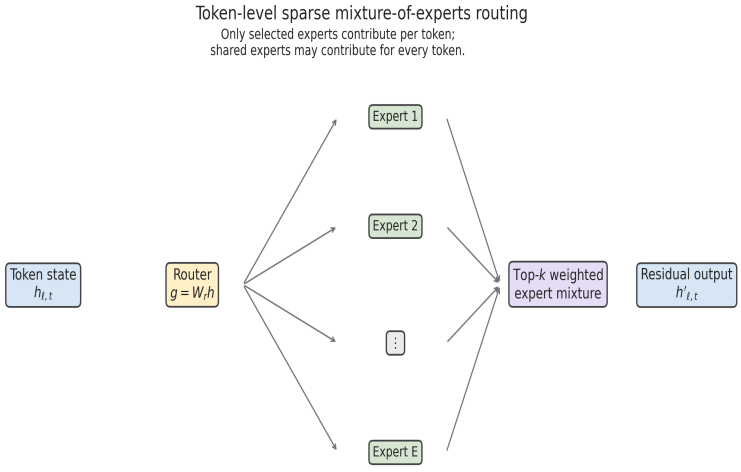

Figure 12. Generic token-level sparse MoE routing mechanism.

> Note. Conceptual architecture diagram only. No router traces, expert-specialization measurements, or routing-stability values were observed.

$$z_{\ell,t} = W_{r,\ell} h_{\ell,t} + b_{r,\ell} \quad (21)$$

$$p_{e,\ell,t} = \frac{e^{z_{e,\ell,t}}}{\sum_{j=1}^{E} e^{z_{j,\ell,t}}} \quad (22)$$

$$K_{\ell,t} = Top_k\left(p_{1,\ell,t}, \ldots, p_{E,\ell,t}\right) \quad (23)$$

$$h_{\ell,t}{}' = h_{\ell,t} + \sum_{e \in K_{\ell,t}} p_{e,\ell,t} E_{e,\ell}\left(h_{\ell,t}\right) + E_{shared,\ell}\left(h_{\ell,t}\right) \quad (24)$$

The router operates per token and layer. A CV heading, employer name, date, skill token, or multilingual degree term can therefore traverse different expert paths. The current benchmark stores only final text, so it cannot determine whether a category reversal arose from expert specialization, router instability, or a downstream prompt/evaluator mechanism.

3.9 Expert load, entropy, and specialization diagnostics

$$f_e = \frac{\sum_t I(e \in K_{\ell,t})}{kT} \quad (25)$$

$$L_{bal} = E \sum_{e=1}^{E} f_e \tilde{p}_e \quad (26)$$

$$H_{\ell,t} = -\sum_{e=1}^{E} p_{e,\ell,t} \log p_{e,\ell,t} \quad (27)$$

Routing frequency, load imbalance, and entropy are required to test a genuine MoE explanation. Low entropy may indicate decisive specialization or router collapse; high entropy may indicate uncertain routing. Neither interpretation is valid without task-conditioned evidence. A defensible follow-up should report these quantities by segment, category, language, and layout.

3.10 Active compute, memory, and quantization

$$\rho_m = \frac{P_{active,m}}{P_{total,m}} \Rightarrow \rho_{Gemma} \approx 0.151, \rho_{Qwen} \approx 0.086 \quad (28)$$

$$M_{weights} \approx \frac{b_w}{8} P_{total} + M_{scales} + M_{metadata} \quad (29)$$

$$\widehat{w} = s\text{clip}\left(\text{round}\left(\frac{w}{s}\right), q_{\min}, q_{\max}\right), \varepsilon_q = \widehat{w} - w \quad (30)$$

$$\gamma_{\ell,t} = z_{(k),\ell,t} - z_{(k+1),\ell,t} \quad (31)$$

The active-parameter ratio is not a latency ratio. Attention, embeddings, shared experts, routing, memory transfer, KV cache, Vulkan kernels, and parallel scheduling also contribute. Four-bit quantization reduces weight bandwidth, but quantization error can affect expert outputs and, where router computations are quantized, can alter top-k decisions when the routing margin is small. The supplied outputs contain no full-precision control, so quantization effects are inseparable from model effects.

3.11 Formal model of Hybrid NLP extraction

Let d denote a CV, M_d its Markdown representation, S the active extraction schema, k a candidate section and f a target field. HNLP is represented as a cascade of interpretable scores rather than a single learned conditional probability. The following equations are analytical reconstructions of the Java decision process; they make the components explicit but do not claim undocumented production coefficient values.

$$H_{d,k} = \alpha_1 J_{alias}(h_{d,k}) + \alpha_2 J_{fuzzy}(h_{d,k}) + \alpha_3 J_{layout}(h_{d,k}) - \alpha_4 B_{penalty}(h_{d,k}) \quad (36)$$

> H(d,k) is the heading/section score. J_alias represents schema-alias evidence, J_fuzzy approximate lexical evidence, J_layout structural context, and B_penalty boilerplate or heading-conflict penalties.

$$E_{d,f}(\ell) = w_r R_f(\ell) + w_d D_f(\ell) + w_s S_f(\ell) + w_c C_f(\ell) \quad (37)$$

> For candidate line ℓ, R_f is regex evidence, D_f dictionary/synonym evidence, S_f section compatibility and C_f local context. The weights summarize code-level contributions and should be estimated or logged in a calibrated implementation.

$$A_{d,f} = Dedup\{N(\ell) : E_{d,f}(\ell) \geq \tau_f\} \quad (38)$$

> A(d,f) is the accepted normalized and deduplicated evidence set for field f. The threshold τ_f represents the field-specific acceptance boundary implicit in heading, evidence and ambiguity rules.

$$\hat{y}_d = V_S(G_{merge}(G_{norm}(G_{extract}(M_d, S, z_{NLP})))) \quad (39)$$

> The complete HNLP output ŷ_d is validation V_S applied after extraction, normalization and merge. z_NLP is an indicator for optional OpenNLP resources; the benchmark does not identify its realized value.

3.12 Formal model of GLiNER/BERT span extraction

BERT MULTI implements label-conditioned span detection. For a tokenized chunk $x_{1:n}$ and schema-derived label set L, a bidirectional encoder contextualizes every token using both left and right context. GLiNER then scores candidate spans against textual label representations rather than restricting inference to a fixed classifier head.

$$h_{1:n} = Enc_\theta(x_{1:n}) \quad (40)$$

> $h_{1:n}$ are contextual token states produced by the bidirectional encoder with parameters θ.

$$z_{i,j,\ell} = g_\varphi(h_i, h_j, q_\ell, u_{j-i}) \quad (41)$$

> $z_{(i,j,\ell)}$ is the compatibility logit between token span i…j and schema-derived label ℓ; q_ℓ is the label representation and $u_{(j-i)}$ represents span width.

$$p_{i,j,\ell} = \sigma(z_{i,j,\ell}) = \frac{1}{1+e^{-z_{i,j,\ell}}} \quad (42)$$

> The runtime applies a sigmoid transformation. At the configured default, a candidate is retained when $p_{(i,j,\ell)} \geq 0.10$ before later sanitization and schema mapping.

$$C_{d,c} = \{(i,j,\ell) : p_{i,j,\ell} \geq \tau_\ell \wedge valid(i,j,c)\} \quad (43)$$

> $C_{(d,c)}$ is the candidate set for chunk c. Validity includes chunk boundaries, tokenizer offsets, span-width rules, text sanitization and label admissibility.

$$C_d = Dedup(\bigcup C_{d,c}) \quad (44)$$

> Candidates from broad sections and focused line-context chunks are unioned and deduplicated. Overlap increases boundary recall but also creates duplicate or conflicting candidates that must be resolved.

$$\hat{y}_d = V_S(B_{md}(M_S(C_d))) \quad (45)$$

> M_S maps retained spans to schema fields, B_md performs Markdown assembly/backfill, and V_S validates the final JSON contract. A valid output can therefore still omit source facts if any earlier stage rejects them.

3.13 Stage-wise recall and error propagation

For either non-generative pipeline, observed recall is constrained by every serial stage. If R_r denotes the conditional retention rate at stage r, an approximate pipeline decomposition is:

$$R_{pipe} \approx \prod R_r \quad (46)$$

> The product is diagnostic rather than an independence claim. A modest loss at several stages can produce a substantial final recall deficit even when each component appears individually acceptable.

$$\pi_{omit} = 1 - \prod (1 - \pi_r) \quad (47)$$

> $\pi_{(omit)}$ expresses cumulative omission probability under a simplifying independent-loss approximation. The terms may represent Markdown loss, section detection, chunk boundary, label scoring, thresholding, deduplication, schema mapping and backfill.

Table 23. Stage-wise error propagation and observable signatures

Stage	Hybrid NLP failure signature	BERT MULTI failure signature	Required diagnostic
Source conversion	Missing headings or broken lists	Missing/reordered tokens and offsets	Compare two PDF/Markdown engines with page-image gold.
Segmentation	Heading aliases fail or boilerplate is retained	Entity crosses chunk boundary	Log section/chunk assignment and boundary overlaps.
Candidate generation	Dictionary/regex gap	Label prompt or encoder misses span	Candidate-recall audit before thresholding.
Acceptance	Fuzzy/evidence threshold rejects line	Confidence threshold rejects span	Precision–recall curves and threshold sweeps.
Resolution	Ambiguous line mapped to one section	NMS/dedup removes competing entity	Persist all pre-resolution candidates.
Schema mapping	Normalization or merge loses content	Label-to-field constraints reject content	Unit tests against a span-to-schema gold set.
Validation	Schema-valid but semantically incomplete output	Schema-valid but semantically incomplete output	Separate structural validity from factual recall.

Figure 18. Stage-wise failure mechanisms in Hybrid NLP and BERT MULTI.

The diagram distinguishes model or heuristic candidate generation from losses introduced by conversion, chunking, acceptance, mapping and validation.

3.14 Evidence-tier comparability model

A numerical comparison is confirmatory only when its systems share the same target construct and measurement process. Let a and b denote two pipelines. The comparability indicator κ(a,b) requires equality of corpus, source representation, target ontology, matching/adjudication, aggregation, and missing-output policy.

$$\kappa_{a,b} = \prod \mathbb{1}[Z_{a,r} = Z_{b,r}] \quad (48)$$

> κ=1 licenses direct paired inference for the specified metric. κ=0 does not make descriptive values useless, but it prohibits a single universal rank. In this benchmark κ=1 for HNLP versus BERT under their shared evaluator and for

> Gemma versus Qwen within their canonical layer; it is not established across those two atomic layers.

3.15 Decision-theoretic model for extraction in RAG

$$U_m = \lambda_R R_{w,m} + \lambda_P P_{w,m} - \lambda_U U_{rate,m} - \lambda_C C_{compute,m} - \lambda_L L_{latency,m} \quad (49)$$

> Model or pipeline choice should maximize predeclared utility U_m under hard safety constraints. The λ coefficients express the application's relative cost of omission, unsupported content, compute and latency; employment screening requires explicit governance of these weights rather than post-hoc preference for the largest F1.

$$\min U_{rate,m} subject\ to R_{w,m} \geq R_{\min} \wedge Q_{critical,m} \leq Q_{\max} \quad (50)$$

> A high-precision system is not deployable as a negative filter unless recall exceeds a predeclared minimum and critical unsupported claims remain below a predeclared maximum. Empty fields otherwise represent missing measurement, not evidence of absence.

3.16 Reproducible retrieval and blinded evaluation of extraction results

The numerical results in the primary common-evaluator layer were retrieved from the frozen source artifacts rather than generated by asking another language model which extraction 'looks better'. The unit of evidence is an atomic extracted fact, the source PDF is the evidentiary anchor, and the scoring kernel is functionally blind to system identity. Archive identity is required to locate files and is restored only after scoring for aggregation and plotting; the fact-level evaluator receives the candidate fact, its schema segment and the normalized PDF evidence, but no model name. Consequently, the same textual candidate receives the same classification regardless of whether it originated from Gemma, Qwen, Hybrid NLP or BERT MULTI.

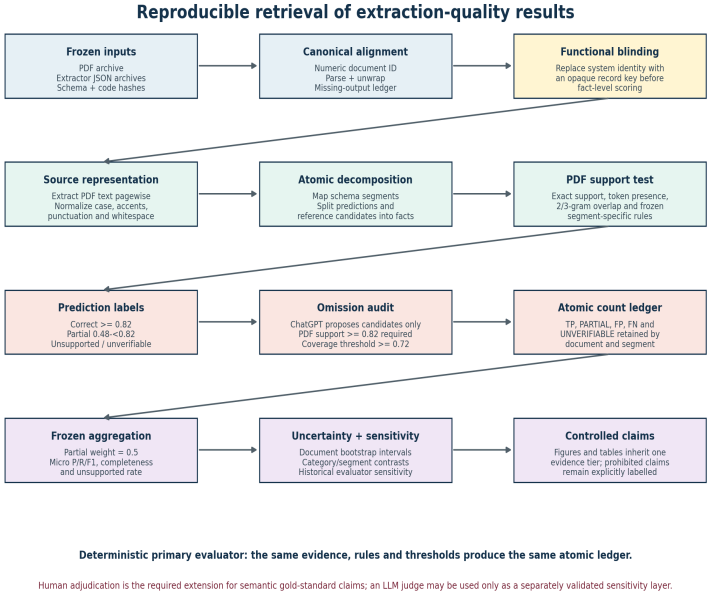

Figure 30. Reproducible evidence-to-result workflow and functional blinding.

The workflow separates source acquisition, identity-invariant atomic scoring, aggregation and interpretation. ChatGPT is used only to propose omission candidates, each of which must independently pass the PDF-support rule.

3.16.1 Frozen inputs, alignment and provenance

For every category, the original PDF archive, each extractor's JSON archive, the extraction schema, prompt files and evaluator code constitute the analysis inputs. Records are aligned by the canonical numeric document identifier obtained after removing the .pdf, .json and .aiextract suffixes. Candidate names, archive ordering and extracted job titles are deliberately excluded from record linkage because they can be missing, altered by extraction or personally identifying. Missing, malformed and duplicate records are retained in an explicit inventory instead of silently disappearing from denominators.

This design implements a reproducible data lineage: raw archive -> canonical record -> parsed extraction object -> segment text -> atomic facts -> PDF-supported classification -> document/segment/category ledger -> aggregate statistic -> figure or claim. The result can therefore be traced backwards from a plotted value to counts and ultimately to the underlying PDF and JSON. File hashes and frozen thresholds protect against unnoticed replacement of inputs or post-hoc tuning.

Table 23A. Frozen stages in the common PDF-grounded evaluation protocol

Stage	Frozen operation	Output retained	Bias controlled
1. Inventory	Enumerate PDFs and JSON files by canonical numeric ID	Presence, duplicate and parse-status ledger	Order and candidate-name matching bias
2. Unwrapping	Apply one wrapper-resolution policy to every extractor	Accessible extraction object or explicit exclusion	System-specific silent repair
3. Source text	Extract PDF text pagewise and normalize identically	Normalized PDF index and readability flag	Reference-extraction circularity
4. Atomicization	Use one segment mapper and fact splitter	Prediction and candidate fact sets	Whole-document length and list-size confounding
5. Support	Apply frozen exact, token and n-gram rules	Correct, partial, unsupported or unverifiable labels	Evaluator preference for a named system
6. Omissions	Admit a ChatGPT candidate only after PDF support	False-negative candidates and coverage decisions	Treating ChatGPT as ground truth
7. Aggregation	Pool the same count definitions and partial weight	P, R, F1, completeness and unsupported rate	Changing denominators after seeing results
8. Uncertainty	Bootstrap documents and retain category/segment strata	Intervals, paired contrasts and sensitivity results	Reliance on one pooled point estimate
9. Reporting	Bind every figure to an explicit evidence tier	Permitted and prohibited claims	Mixing incompatible evaluator workflows

3.16.2 Atomic support score and classification rule

After Unicode normalization, case folding, accent removal and replacement of non-alphanumeric separators, a predicted fact f is compared with the PDF text d. Exact normalized containment receives full support. One- or two-token facts receive full support only when every token occurs in the PDF index. Longer facts are evaluated by the proportion of their bigrams or trigrams found in the PDF. This is deliberately a transparent extractive-fidelity measure rather than an opaque semantic judgment.

$$s_{PDF}(f,d) = 1 \; if \; exact \; support; \; otherwise \; \frac{|G(f) \; intersection \; G(d)|}{|G(f)|} \quad (51)$$

> G denotes the frozen fact n-gram set: bigrams for shorter multi-token facts and trigrams for longer facts. The implemented short-fact rule additionally requires complete token presence.

$c_{pred}(f) = Correct\ if\ s_{PDF} >= 0.82;\ Partial\ if\ 0.48 <= s_{PDF} < 0.82;\ otherwise\ Unsupported\ or$ (52)

> The thresholds are operational definitions fixed before aggregation. They are reproducible measurement rules, not universal boundaries of semantic truth.

Omission discovery is separated from prediction validation. The ChatGPT extraction can propose a potentially missing fact, but the candidate enters the false-negative set only when it is itself supported by the PDF at sPDF >= 0.82. A candidate is counted as covered when exact containment, local support >= 0.78, seniority compatibility or atomic similarity >= 0.72 succeeds. This prevents agreement with ChatGPT from being mistaken for correctness, while acknowledging that facts omitted by both ChatGPT and the tested system remain a recall blind spot.

$$TP_w = TP + 0.5\,PARTIAL;\ \ P_w = \frac{TP_w}{TP + PARTIAL + FP};\ \ R_w = \frac{TP_w}{TP + PARTIAL + FN} \quad (53)$$

> Weighted precision and recall preserve the full count ledger and use the predeclared partial-credit weight 0.5. Strict metrics remain available separately.

3.16.3 Functional blinding and evaluator invariance

Blinding in a software benchmark means that the decision rule cannot condition its judgment on the identity, reputation or architecture of the system being scored. The present evaluator satisfies functional blinding at the atomic decision point: the scoring function has no model-identity argument and applies the same segment-specific rules to every archive. This is stronger than merely asking an analyst to ignore labels, because the code path itself prevents identity from changing the score.

$$E_{theta}(f, d, z, m) = E_{theta}(f, d, z)\ for\ every\ system\ identity\ m \quad (54)$$

> Etheta is the frozen evaluator, f the candidate fact, d the PDF evidence and z the schema segment. Model identity m is absent from the implemented scoring kernel.

Functional blinding does not imply that the entire research team was unaware of archive identities, nor does it remove every design choice. The evaluator author selected normalization, fact splitting and thresholds, and those choices affect construct validity. The defensible claim is therefore identity-invariant reproducibility under a declared operationalization, not an impossible claim of evaluator neutrality without assumptions.

3.16.4 Why an LLM is not the sole primary evaluator

A language model can provide useful semantic review, propose error hypotheses and identify paraphrases missed by a lexical rule. It should not, however, be the sole source of the primary numerical benchmark in this study. An LLM judgment depends on model snapshot, provider-side changes, decoding parameters, hidden system instructions, prompt order, context truncation and stochastic sampling. Re-running the same nominal prompt can therefore change atomic labels and aggregate ranks. If ChatGPT also generated the reference extraction, using a related LLM as judge creates additional risks of self-preference, shared error and circularity.

3.16.4.1 Judge-conditioned scores and the fairness problem

Using a different LLM as judge does not remove this problem; it introduces the judge model as an additional, insufficiently controlled experimental factor. The observed score is then a property of an extractor-judge pair rather than a property of the extractor alone. Different judges may apply different implicit definitions of semantic equivalence, completeness, acceptable normalization, inference and hallucination. They may also favor outputs that resemble their own preferred wording, ontology, reasoning style, tokenizer conventions or model-family priors. These influences are difficult to observe because the relevant training data, routing state, internal representations and decision boundary are not available for inspection.

$$S_{i|j} = Q_i + b_j + a_{i,j} + epsilon_{i,j} \quad (54A)$$

> S_i|j is the observed score for extractor i when judged by LLM j. Q_i is the latent extraction quality of interest, b_j is judge severity or leniency, a_i,j is judge-extractor affinity, and epsilon_i,j is run-specific variability. With one opaque judge these components are confounded.

The interaction term is especially important for fairness. A judge can systematically prefer one extractor because of shared model lineage, similar instruction tuning, familiar formatting, verbosity, or matching assumptions about what may be inferred from a CV. Another judge may prefer a different extractor. A score difference can therefore reflect judge affinity rather than a difference in source-supported extraction quality. Merely choosing a prestigious or more capable judge does not identify or eliminate this interaction.

$$sign(S_{i|j} - S_{k|j}) = sign(S_{i|j'} - S_{k|j'}) \; for \; all \; admissible \; judges \; j \; and \; j' \quad (54B)$$

> A judge-independent ranking would require pairwise ordering to remain invariant across admissible judges. That invariance was not tested and cannot be assumed; a single LLM judge therefore cannot establish a fair ranking of LLM extractors.

This concern is not hypothetical measurement trivia. If replacing judge j with judge j' changes which extractor wins, then the study has measured judge-conditioned preference. Even when the ranking does not reverse, the size of the difference, the error taxonomy and the apparent precision-recall profile may change. Fair evaluation therefore requires either an explicit identity-invariant rule or a multi-rater design in which judge effects, interaction effects and agreement are estimated and reported. One LLM silently deciding all labels provides neither condition.

An unvalidated LLM judge also makes disagreement difficult to audit: a fluent explanation is not a measurement rule, and a reader cannot mechanically reproduce the latent reasoning that converted a PDF passage into Correct, Partial or Unsupported. Academic reproducibility requires that another researcher with the same archived inputs can reconstruct the same ledger without access to an undocumented model state. The deterministic evaluator satisfies that requirement because its normalization, fact splitter, thresholds and formulas are inspectable and executable.

Table 23B. Why an LLM is not the sole primary evaluator

Threat from sole LLM judging	Consequence	Control used in the primary layer	Permitted supplementary use
Stochastic output	Atomic labels and ranks can change between runs	Deterministic code and frozen thresholds	Repeated LLM judging with agreement statistics
Model/provider drift	A nominal model name may not reproduce an earlier snapshot	Versioned local evaluator and artifact hashes	Archive model version, prompt and response for every fact
Self-preference and shared priors	Evaluator may favor style or facts resembling its own outputs	Identity-invariant PDF support rules	Use an unrelated judge and randomized anonymous system labels
Judge-model dependence	Scores become properties of extractor-judge pairs and rankings may reverse	One identity-invariant scoring kernel for every extractor	Use multiple judges, estimate judge-by-extractor interaction and test rank stability
Prompt and position sensitivity	Order and wording can alter judgments	No natural-language judgment prompt in primary scoring	Counterbalanced prompt/order sensitivity analysis
Opaque semantic boundary	Partial versus correct cannot be mechanically reconstructed	Published score thresholds and formulas	Human-authored coding manual and adjudication log
Circular reference use	Agreement can be misreported as factual accuracy	Every reference candidate must pass PDF support	Reference agreement reported separately
Cost and access dependence	Replication requires paid or unavailable infrastructure	Local executable evaluator	Publish cached judge outputs and complete provenance

3.16.5 Why the rule-based protocol is correct for the primary claim

The primary claim concerns source-supported extractive fidelity: whether output facts are visibly supported by the CV text and whether PDF-supported candidate facts are represented. For that construct, a conservative lexical and n-gram rule has three decisive advantages. First, it is falsifiable: every decision follows a declared threshold. Second, it is repeatable: identical inputs yield an identical atomic ledger. Third, it is comparable: all four tested extractors are measured with the same source, fact splitter, thresholds, missing-output policy and aggregation. These properties make differences attributable to the observed outputs under the protocol rather than to changing evaluator behavior.

$$R = F_{theta}(PDF\ archives,\ JSON\ archives,\ schema,\ code) \quad (55)$$

> With theta, inputs and software environment fixed, Ftheta is deterministic. Reproducibility therefore means that an independent rerun reconstructs R, the same count ledger and derived statistics.

This is the correct primary method only within its declared scope. It does not prove deep semantic equivalence, resolve every coreference, or understand that two differently worded descriptions entail the same proposition. The protocol may under-credit paraphrase and over-credit lexical coincidence. Those are acknowledged construct-validity limitations, not reasons to replace a reproducible primary measure with an opaque one. Instead, the academically appropriate design is layered: deterministic measurement for reproducibility, blinded human annotation for semantic validity, and optional LLM judging as a documented sensitivity analysis.

Table 23C. Validity claims, limitations and required extensions

Claim	Status under present protocol	Reason	Required extension
Literal/source-supported extractive fidelity	Supported	PDF-grounded atomic rules are explicit and reproducible	Retain hashes and executable environment
Identity-invariant four-system comparison	Supported for Gemma, Qwen, HNLP and BERT	Same scoring kernel and missing-output policy	Keep archive identity outside the scoring function
Perfect semantic correctness	Not established	Lexical support is an imperfect semantic proxy	Blinded multi-rater human gold standard
Complete recall of all CV facts	Not established	Omission candidates originate from ChatGPT and can be jointly missed	Independent exhaustive human annotation
Equivalent ChatGPT atomic score	Not established	ChatGPT acts as diagnostic omission source in this layer	Score ChatGPT against an independently created gold set
Causal model superiority	Not established	One run per system and pipeline components are confounded	Repeated factorial ablations with controlled preprocessing
Generalization to new CV populations	Not established	Results are corpus- and schema-specific	Held-out multilingual and layout-stratified datasets

3.16.6 Human gold standard and the role of LLM sensitivity analysis

The strongest future evaluation would begin with independently annotated PDF facts rather than facts proposed by any extractor. At least two trained annotators should work with anonymous system labels and a fixed coding manual, record evidence spans and page numbers, and resolve disagreements through a third adjudicator. Inter-rater agreement should be reported before adjudication, and a stratified sample should cover categories, languages, layouts, OCR quality and document length. This would directly test the lexical evaluator's false-positive and false-negative classifications.

An LLM judge can then be evaluated rather than assumed: run a frozen judge snapshot on the same anonymous fact/evidence packets, repeat with counterbalanced order, retain every response, and compare its labels with the human gold standard using agreement, calibration and class-specific error rates. Only after such validation may LLM adjudication be used to expand semantic coverage. Even then, deterministic and human-grounded results should remain separately reported so that model-assisted interpretation cannot silently redefine the benchmark.

$$Validity_{overall} = Reproducible\ rule\ layer\ +\ blinded\ human\ semantic\ layer\ +\ documented\ sensitivit$$
(56)

> The layers answer different questions: rules establish repeatable extractive support, humans establish semantic construct validity, and LLM judges test scalable approximation under measured error.

3.16.7 Result-provenance audit and removal of unsupported claims

A separate provenance audit was applied after report construction. The common four-system evaluator was executed again from the 2,484 PDF files and the four extractor archive sets, without writing intermediate benchmark values, and its complete in-memory result object was compared with the stored audit JSON. The workbook's overall, category, segment, pair, document and alignment sheets were then compared cell-by-cell with that JSON. Historical workbook sheets were also compared with their preserved pre-

merge versions or their named BERT/HNLP source workbooks. Traceability establishes that a value was calculated from supplied artifacts; it does not convert automated lexical judgments into human semantic ground truth.

The audit also searched for numerical-looking material that lacked an empirical source. A manually assigned ordinal heatmap of HNLP/BERT operating-condition sensitivity was removed and replaced by a non-scored prerequisite diagram. Unmeasured ordinal claims about extraction speed and stability across new datasets were removed from the conclusion. They are retained only as mechanism-based hypotheses requiring controlled latency, repeat-run and external-validation experiments. Conceptual MoE routing, causal-path, reasoning-mode and factorial-design figures are explicitly labelled non-empirical.

Table 23D. Result-provenance audit and removal of unsupported claims

Material	Audit outcome	Permitted use	Prohibition
Common four-system results, Figures 20–29 and Tables 24C–24D	Reproduced from raw archives; workbook values match stored audit JSON	Primary comparison under the frozen rule-based evaluator	Do not describe as human-adjudicated semantic truth
Historical Gemma/Qwen and HNLP/BERT workflow-specific results	Traced to preserved or named source workbooks	Sensitivity analysis within the documented evidence tier	Do not pool silently into a universal five-system rank
MoE, causal, reasoning and factorial diagrams	Explicitly labelled conceptual or proposed	Theory, hypotheses and study design	Do not cite as measured effects
HNLP/BERT implementation diagrams	Traced to supplied Java source	Construction, prerequisites and failure-mechanism analysis	Do not infer an unmeasured numerical sensitivity coefficient
Deployment speed and cross-dataset stability	No controlled measurement supplied	Testable architecture-derived hypotheses only	No empirical ordering or superiority claim

4. Results

4.1 Corpus integrity and pipeline success

Table 3. Corpus and pipeline inventory

Artifact	Count	Interpretation
Source PDFs	2,484	Authoritative evidentiary documents
ChatGPT JSON	2,484	All PDF IDs represented
Gemma JSON	2,482	Missing Banking 21796843 and Business Development 12632728
Qwen JSON	2,483	Missing Business Development 12632728
Fully matched four-way records	2,482	PDF plus all three outputs
Non-document archive artifacts	1	Advocate ChatGPT aiextract.json; excluded from CV counts

Pipeline success was 100.00% for ChatGPT, 99.92% for Gemma (95% Wilson interval 99.71%–99.98%), and 99.96% for Qwen (95% Wilson interval 99.77%–99.99%). All available extraction objects passed the authoritative schema. Missing outputs, rather than malformed available objects, account for the local-model schema-validity shortfall.

4.2 Common direct comparison

Table 4. Common direct-archive metrics

Model	Parsed	Schema valid	Metadata populated	Mean nonempty segments	Exact object support	PDF-support precision	co
ChatGPT	2,484/2,484	100.0%	94.7%	6.50	76.6%	94.0%	91
Gemma	2,482/2,484	99.9%	16.7%	4.79	60.3%	94.7%	85
Qwen	2,483/2,484	100.0%	16.7%	6.42	62.2%	93.6%	87

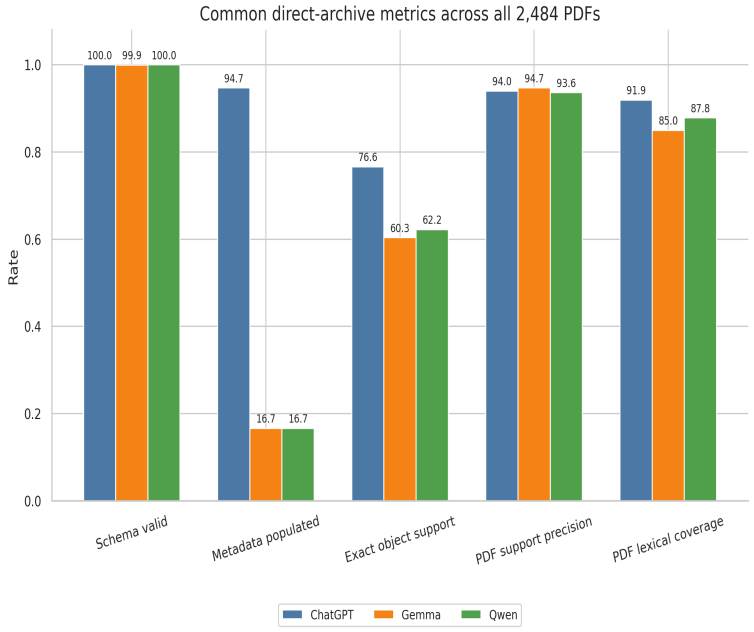

Figure 2. Common direct-archive model profile.

> *Note. Lexical coverage is a broad extractive proxy, not semantic recall. Missing local JSON files count against full-schema validity. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

ChatGPT captured the broadest share of PDF lexical features and populated almost all deterministic metadata. Gemma had the highest PDF-support precision but populated fewer segments and captured less broad source text. Qwen occupied an intermediate position, with substantially more populated segments and broader coverage than Gemma, while retaining a high support rate. Because the prompts allow duplication across semantically compatible segments and PDFs contain non-target text, these measures should not be read as semantic precision or recall.

4.3 Recomputed atomic performance

Table 5. Recomputed local-model atomic metrics

Model	Documents	Weighted precision	Weighted recall	Weighted F1	Completeness	Unsuppo rate
Gemma	2,484	94.6%	64.4%	76.6%	67.3%	1.2%
Qwen	2,484	94.9%	73.7%	82.9%	75.5%	2.7%

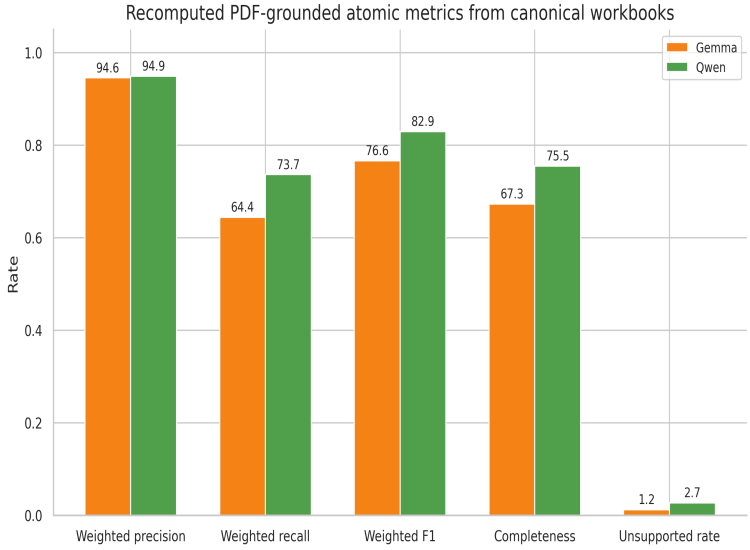

Figure 3. Local-model atomic performance and risk profile.

> *Note. Qwen has higher recall, F1 and completeness; Gemma has the lower unsupported-content rate. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

Qwen's principal advantage is coverage: weighted recall is 9.2 percentage points higher, weighted F1 is 6.3 points higher, and completeness is 8.2 points higher. Gemma's advantage is risk containment: the unsupported-content rate is 1.45 points lower, and the critical-claim document count is approximately one third of Qwen's. Weighted precision differs by only 0.35 points. The operational choice is therefore not a simple accuracy ranking; it is a coverage–risk trade-off.

4.4 Category-level paired analysis

Table 6. Paired category-level statistical analysis

Statistic	Estimate	Interpretation
Qwen category wins	15/24	Gemma wins 9; no ties
Mean Qwen–Gemma F1 difference	3.4%	95% bootstrap interval -2.2% to 9.4%
Median difference	3.4%	Paired categories
Paired t-test	t(23)=1.14	p = 0.265
Wilcoxon signed-rank	W=111.0	p = 0.277
Exact sign test	15 positive signs	p = 0.307
Paired effect size	Cohen's dz=0.23	Standardized mean paired difference
Pearson association	r=0.10	p = 0.638
Spearman rank association	ρ=-0.01	p = 0.977

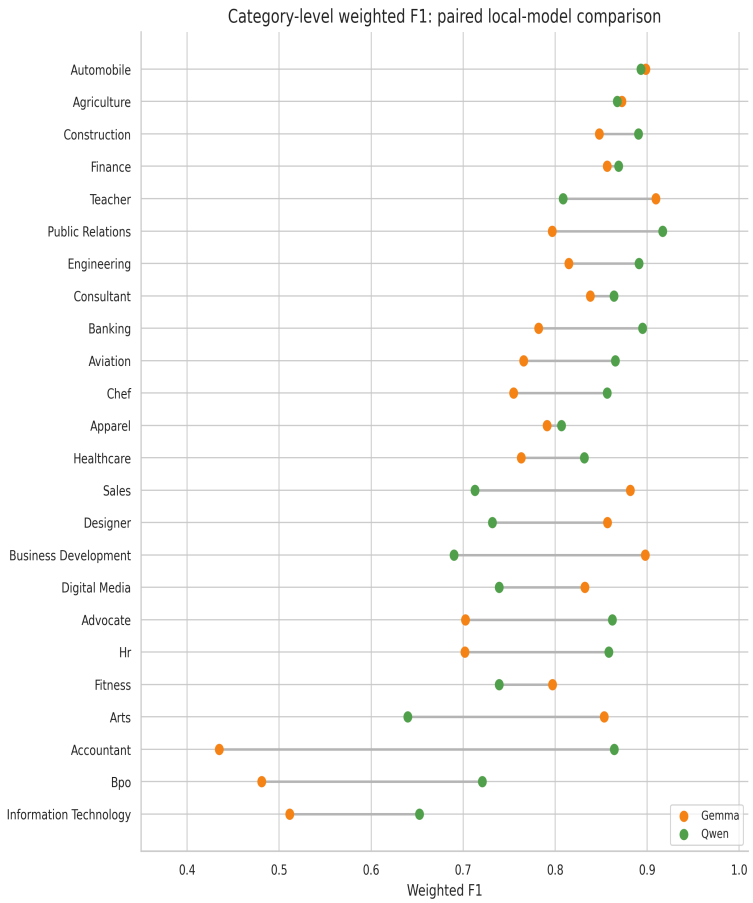

Figure 4. Category-level weighted F1 for Gemma and Qwen.

> *Note. Large category reversals caution against a universal winner and may also reflect evaluator heterogeneity. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

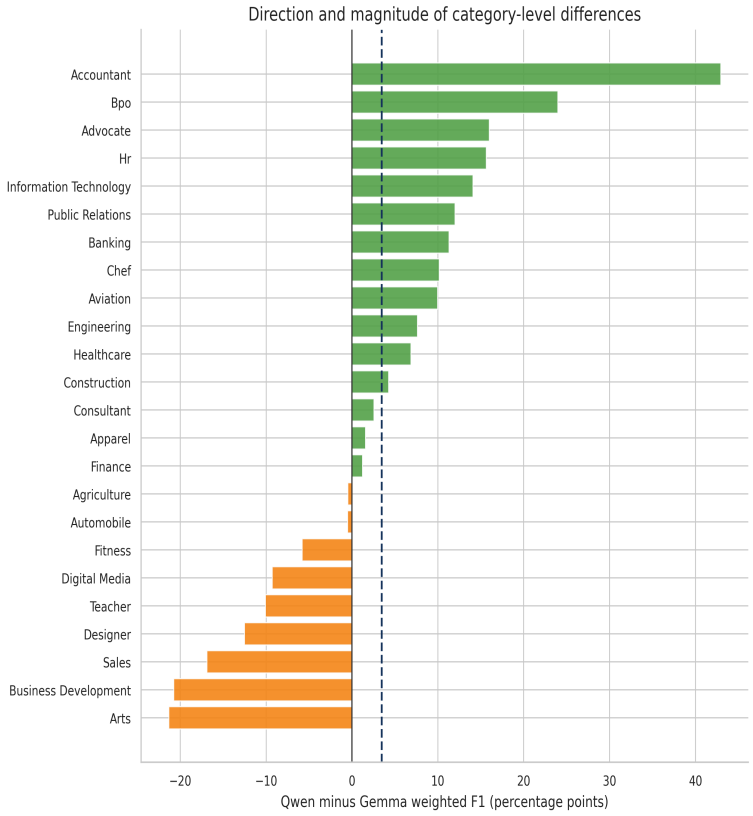

Figure 5. Category-level Qwen–Gemma weighted-F1 differences.

> *Note. Positive values favour Qwen; negative values favour Gemma. The dashed line is the mean paired difference. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

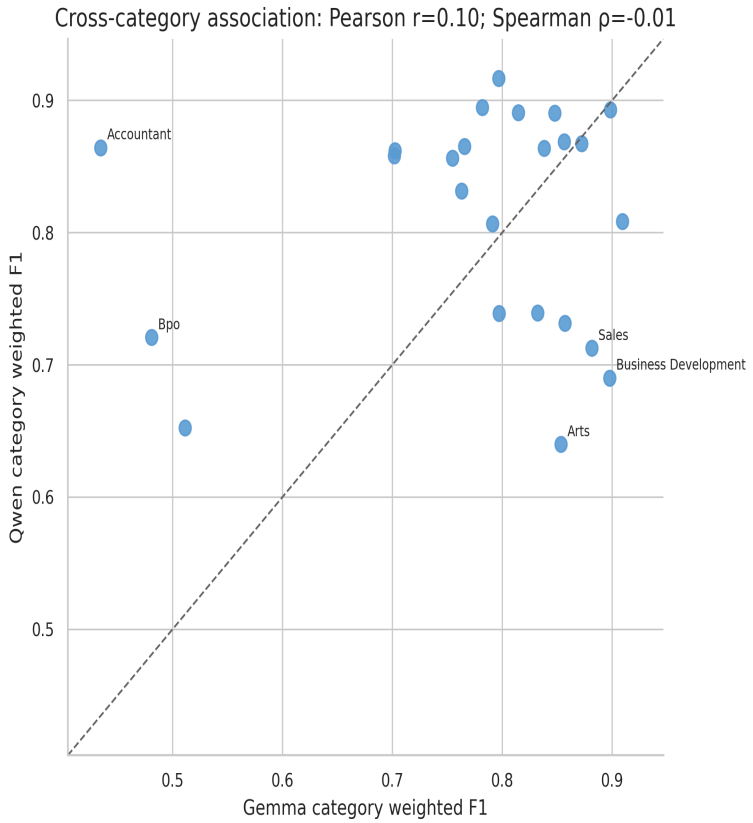

Figure 6. Cross-category association of local-model weighted F1.

> *Note. The identity line marks equal performance. Correlation describes shared category ordering, not agreement in absolute score. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

The mean paired difference favours Qwen, but category-level variability is large relative to the mean. The bootstrap interval and non-parametric tests should be interpreted descriptively because the 24 categories are fixed corpus strata and because evaluator implementations differ. Low cross-category rank association would imply that apparent domain strengths are not stable between report pipelines; high association would indicate shared category difficulty. Neither interpretation isolates model architecture.

4.5 Highest and lowest category results

Table 7. Highest and lowest category weighted F1

Model	Rank group	Category	W. precision	W. recall	W. F1	Completeness	Unsu
Gemma	Lowest	Accountant	89.5%	28.7%	43.5%	32.1%	0.0%
Gemma	Lowest	Bpo	97.6%	31.9%	48.1%	32.2%	1.6%
Gemma	Lowest	Information Technology	75.9%	38.6%	51.1%	50.8%	0.1%
Gemma	Highest	Teacher	97.8%	84.9%	90.9%	85.6%	1.5%
Gemma	Highest	Automobile	100.0%	81.6%	89.8%	81.6%	0.0%
Gemma	Highest	Business Development	98.5%	82.5%	89.8%	83.7%	0.1%
Qwen	Lowest	Arts	71.3%	58.1%	64.0%	59.3%	27.2%
Qwen	Lowest	Information Technology	93.4%	50.1%	65.3%	50.8%	5.3%
Qwen	Lowest	Business Development	91.7%	55.3%	69.0%	56.7%	6.1%
Qwen	Highest	Public Relations	98.3%	85.9%	91.7%	86.3%	1.2%
Qwen	Highest	Banking	98.2%	82.2%	89.5%	82.4%	1.5%
Qwen	Highest	Automobile	97.6%	82.3%	89.3%	83.8%	0.7%

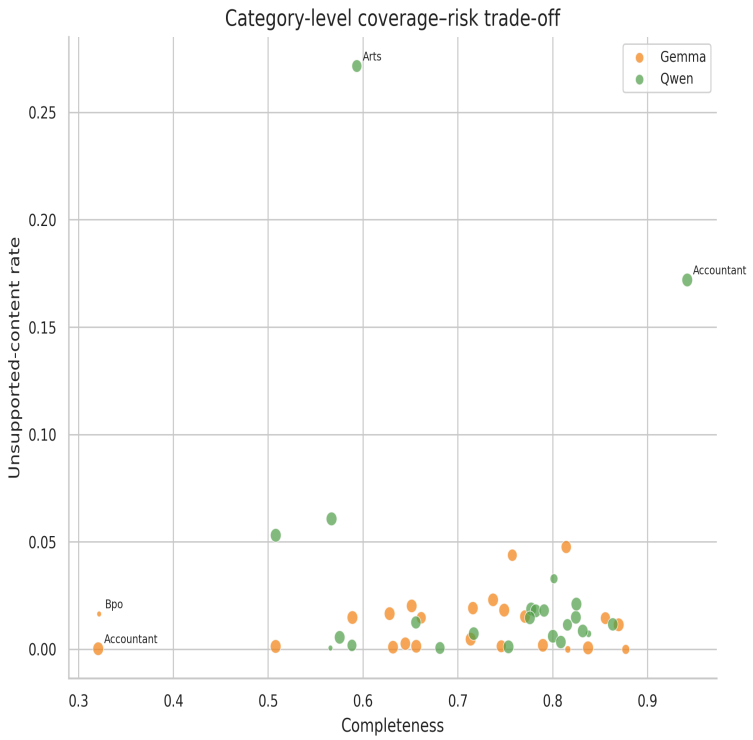

Figure 9. Category-level completeness and unsupported-content trade-off.

> *Note. Point area is proportional to category document count. The preferred region is lower right. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

4.6 Segment-level performance

Table 8. Atomic segment-level weighted F1 and completeness

Field family	Gemma F1	Gemma complete	Gemma unsupported	Qwen F1	Qwen complete	Qwen unsupporte
Profile	83.0%	78.8%	0.4%	88.2%	89.0%	3.8%
Experience	87.7%	84.6%	0.3%	90.9%	85.6%	0.7%
Skills	76.2%	64.8%	0.2%	80.7%	69.7%	1.3%
Education	69.0%	58.8%	0.4%	71.2%	57.8%	1.3%
Certifications	64.6%	56.5%	0.5%	71.3%	66.5%	4.8%
Language	59.4%	51.0%	0.2%	72.1%	61.7%	5.6%
Seniority	4.6%	3.1%	29.9%	64.4%	57.9%	10.4%
Category Classification	13.1%	8.9%	0.2%	63.4%	53.0%	3.8%
Document Metadata	26.9%	19.4%	47.6%	24.2%	18.6%	45.0%
Normalized Fields	5.5%	3.0%	27.5%	34.7%	29.9%	46.5%

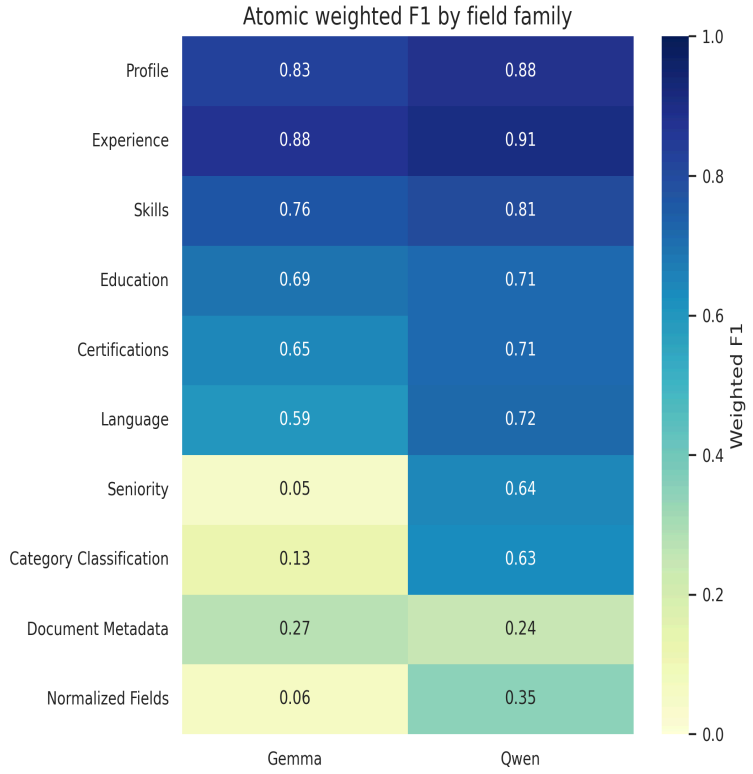

Figure 7. Atomic weighted F1 by segment or field family.

> *Note. Low metadata and normalized-field values support deterministic downstream population rather than model inference. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

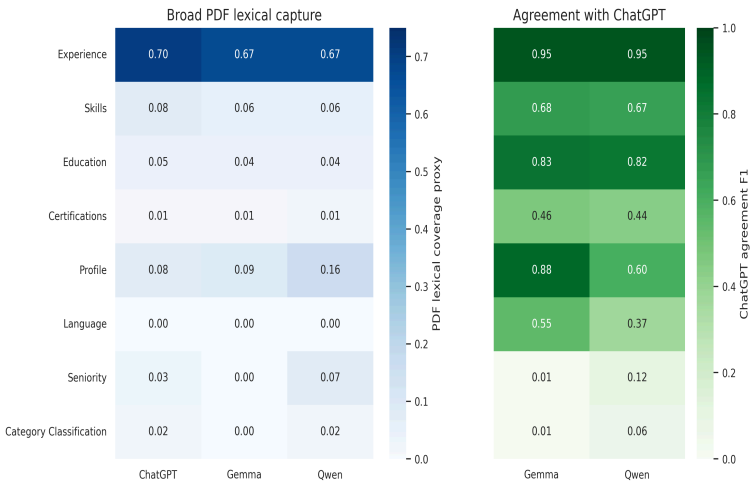

Figure 8. Direct segment-level lexical capture and ChatGPT agreement.

> *Note. Agreement is diagnostic only. Coverage denominators include the entire PDF and therefore are not semantic recall. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

Experience and profile are generally more reliable than metadata, normalized fields, seniority, and category classification. Weak derived fields should not serve as authoritative exclusion filters. Document ID, source filename, corpus category, and schema version can be populated deterministically; seniority and category classification should remain evidence-linked, confidence-scored interpretations.

4.7 Exploratory document-level sensitivity

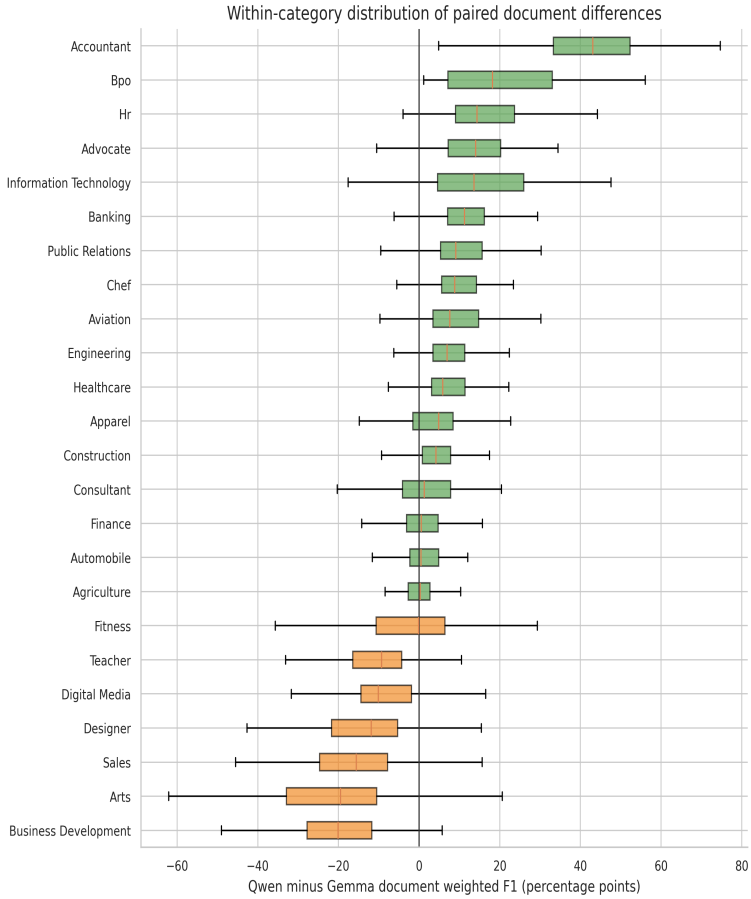

Figure 10. Distribution of paired document-level weighted-F1 differences by category.

> *Note. Exploratory only: document scores originate from model-specific machine-assisted adjudication workflows. Source: author calculations from the supplied benchmark archives and canonical workbooks.*

Across 2,484 paired document rows, the mean Qwen–Gemma weighted-F1 difference is 3.4% and the median is 3.9%. Qwen has the higher document score in 1,531 cases, Gemma in 952, with 1 ties. The Wilcoxon result is $p < .001$. These values are sensitivity evidence only: the two score sets were not produced through a single blinded adjudication implementation.

4.8 Five-system evidence tiers and descriptive operating profiles

Two result layers are retained. Table 24 preserves the historical workflow-specific evidence so that earlier results remain auditable. The new common audit then returns to every raw JSON archive and every source PDF, aligns records by canonical numeric document ID, and applies one frozen fact splitter, matcher, threshold set, missing-output policy and aggregation rule to Gemma, Qwen, Hybrid NLP and BERT MULTI. The common layer is the primary basis for Figures 20–29.

The empirical visual policy is now consistent: Figures 19–29 display Gemma, Qwen, Hybrid NLP and BERT MULTI together. Figures 20–29 use the new common evaluator, so no evaluator separator or two-by-two panel is required. Figures 1–18 retain their original scope only where the construct is genuinely system-specific—for example MoE routing applies to Gemma and Qwen, while Java pipeline diagrams apply to the proprietary HNLP and BERT implementations.

Table 24A. Figure inclusion policy and justified exceptions

Figure range	Systems shown	Inclusion rule	Academic justification
Figures 1–10	ChatGPT, Gemma and Qwen as applicable	Original direct-archive and canonical generative audit	HNLP/BERT were not scored in that original direct evaluator; retrospective insertion would fabricate measurements.
Figures 11–15	Gemma/Qwen or conceptual alternatives	MoE routing, reasoning and causal-design theory	HNLP and BERT are not sparse generative MoE systems, so inclusion would be conceptually false.
Figures 16–18	HNLP and BERT	Code-derived proprietary construction and failure mechanisms	The diagrams concern Java-specific dictionaries, chunks, spans and mapping stages that do not exist in Gemma/Qwen.
Figures 19–29	Gemma, Qwen, HNLP and BERT	Unified empirical layer; Figures 20–29 use one frozen PDF-grounded evaluator	All four perform the same JSON-extraction task and are therefore shown together under controlled measurement.

4.8.1 Common four-system PDF-grounded audit

For document i, model m and segment s, let $L_{i,m,s}$ be the atomic facts emitted by the tested extractor. Let $G_{i,s}$ be candidate facts generated from the ChatGPT extraction, but retained only when independently supported by normalized source-PDF text. Thus ChatGPT increases omission-search sensitivity; it is not declared correct by construction. An emitted fact receives full credit when normalized exact or n-gram support is at least 0.82, partial credit when support is at least 0.48, and is otherwise unsupported. A PDF-supported candidate is an omission when it is not covered by containment, support ≥0.78, or atomic similarity ≥0.72. These thresholds are frozen for all four systems.

Table 24B. Controls in the common four-system audit

Control	Common setting	Why it matters
Corpus	2,484 source PDFs in 24 categories	Eliminates corpus-selection differences.
Alignment	Canonical numeric ID; .pdf and .aiextract suffixes removed	Prevents false non-matches caused by archive naming.
Target	Eight segment-extraction-v2 families	Fixes the extraction ontology.
Evidence	Normalized PDF text; ChatGPT only proposes omission candidates	Avoids treating reference agreement as truth.
Fact unit	Identical sentence/line/semicolon splitter for every system	Controls atomic granularity.
Partial credit	$\omega = 0.50$	Makes partial evidence explicit rather than silently correct.
Missing output	Retained in denominator; supported candidates become omissions	Prevents favorable complete-case bias.
Aggregation	Micro pooled counts; 4,000 document-bootstrap replicates	Reports volume-weighted performance and sampling uncertainty.

The protocol still has asymmetric ascertainment. A fact omitted by both ChatGPT and the tested extractor cannot become an FN, so recall may be optimistic where the reference itself is incomplete. Conversely, paraphrases can be under-credited by lexical matching, and PDF conversion errors can make a true fact appear unsupported. Inferred seniority and category labels without explicit PDF wording are classified as unverifiable and excluded from TP/FP/FN denominators. These design choices are reported because changing them can change the ranking.

$$P_{w,m} = \frac{TP_m + \omega Q_m}{TP_m + Q_m + FP_m} \quad (51)$$

> Weighted precision discounts partially supported facts Q by the predeclared weight ω=0.50.

$$R_{w,m} = \frac{TP_m + \omega Q_m}{TP_m + Q_m + FN_m} \quad (52)$$

> Weighted recall penalizes PDF-supported candidate facts that the tested extraction does not cover.

$$F_{1,w,m} = \frac{2P_{w,m}R_{w,m}}{P_{w,m} + R_{w,m}} \quad (53)$$

> The weighted F_1 harmonic mean prevents high precision alone from masking systematic omission.

$$C_m = \frac{TP_m + Q_m}{TP_m + Q_m + FN_m} \quad (54)$$

> Completeness measures whether a supported fact is represented at all, irrespective of partial-credit weight.

$$U_m = \frac{FP_m}{TP_m + Q_m + FP_m} \quad (55)$$

> The unsupported-content rate estimates the share of emitted scored facts without adequate PDF support.

Table 24C. Common-evaluator four-system results

System	Outputs	W. precision	W. recall	W. F1	95% bootstrap CI	Completeness	U
Gemma	2,482/2,484	99.40%	81.16%	89.36%	88.93%–89.78%	81.57%	0
Qwen	2,483/2,484	99.04%	83.02%	90.32%	89.93%–90.73%	83.72%	0
Hybrid NLP	2,483/2,484	99.25%	75.78%	85.94%	85.27%–86.56%	76.25%	0
BERT MULTI	2,483/2,484	99.44%	55.83%	71.51%	70.63%–72.38%	56.09%	0

Table 24D. All six paired category contrasts under the common evaluator

Contrast	Mean category ΔF1	Median category ΔF1	Category wins	Interpretation
Gemma − Qwen	-0.90%	-0.94%	4–20; ties 0	Paired descriptive contrast under one evaluator; not causal architecture attribution.
Gemma − Hybrid NLP	3.83%	4.23%	21–3; ties 0	Paired descriptive contrast under one evaluator; not causal architecture attribution.
Gemma − BERT MULTI	17.93%	17.54%	24–0; ties 0	Paired descriptive contrast under one evaluator; not causal architecture attribution.
Qwen − Hybrid NLP	4.73%	5.33%	22–2; ties 0	Paired descriptive contrast under one evaluator; not causal architecture attribution.
Qwen − BERT MULTI	18.83%	18.84%	24–0; ties 0	Paired descriptive contrast under one evaluator; not causal architecture attribution.
Hybrid NLP − BERT MULTI	14.10%	13.37%	24–0; ties 0	Paired descriptive contrast under one evaluator; not causal architecture attribution.

The common layer yields the conditional ordering Qwen (90.32% weighted F_1), Gemma (89.36%), Hybrid NLP (85.94%) and BERT MULTI (71.51%). Precision is compressed above 99% for all four, while recall ranges from 83.02% to 55.83%; the principal separation is therefore omission rather than unsupported generation. Qwen exceeds Gemma in 20 of 24 categories, Gemma exceeds HNLP in 21, Qwen exceeds HNLP in 22, and every non-BERT system exceeds BERT in all 24 categories. These paired signs describe the frozen evaluator, not causal architecture effects.

The results are mechanistically plausible but not self-validating. Gemma and Qwen often retain long source-like section text, which improves lexical coverage of PDF-supported candidates. HNLP is highly precise but can omit content when headings, aliases, dictionaries or fuzzy gates do not activate. BERT's span-centric candidate generation and subsequent thresholding, deduplication and schema mapping can preserve precision while losing section-length evidence. The lower common-evaluator HNLP score relative to its historical workflow is itself an important sensitivity result: evaluator construction materially affects the apparent operating point.

Archive integrity was almost complete. Qwen, HNLP and BERT each supplied 2,483/2,484 outputs; Gemma supplied 2,482. BUSINESS-DEVELOPMENT document 12632728 was unreadable through the common PDF parser and lacked all four tested outputs; Gemma additionally lacked BANKING document 21796843. Missing readable outputs remain in the omission denominator. The shared unreadable PDF is retained as a failed operational outcome but cannot receive a factual score without recoverable source text.

Table 24. Five-system empirical profile with evidence-tier rest

System	Direct PDF support	Direct lexical coverage	Weighted precision	Weighted recall	Weighted F1	Completeness
ChatGPT	94.0%	91.9%	N/A	N/A	N/A	N/A
Gemma	94.7%	85.0%	94.6%	64.4%	76.6%	67.3%
Qwen	93.6%	87.8%	94.9%	73.7%	82.9%	75.5%
Hybrid NLP	N/A	N/A	99.2%	75.8%	85.9%	76.3%
BERT MULTI	N/A	N/A	99.4%	55.8%	71.5%	56.1%

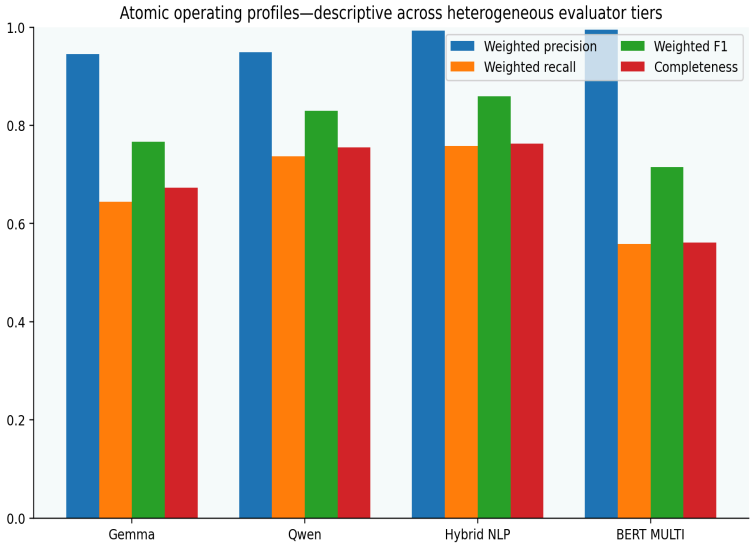

Figure 19. Atomic operating profiles across available evidence tiers.

Values from different atomic evaluator tiers are descriptive and must not be interpreted as a five-system league table.

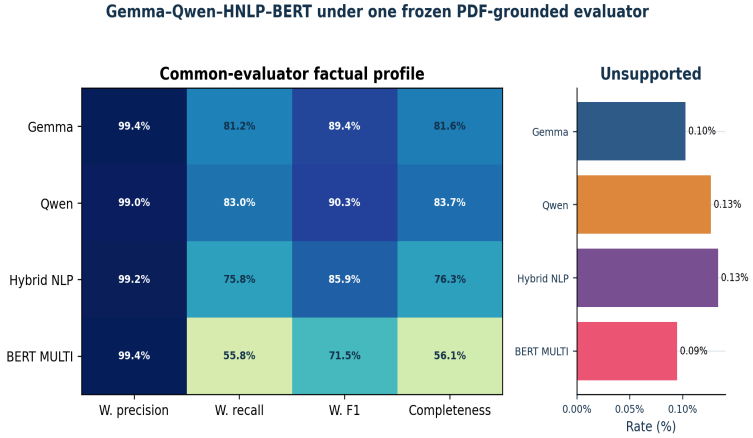

Figure 20. Common-evaluator four-system global factual profiles.

All four systems use the same PDFs, schema, fact splitter, thresholds, omission search and missing-output policy. Unsupported-content rates are isolated because their scale is smaller than the quality metrics.

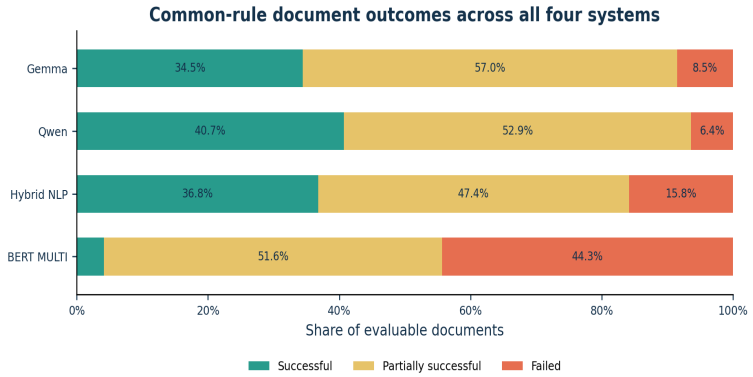

Figure 21. Common-rule document outcomes for all four systems.

Successful, partial and failed classes are recomputed for every system from the same weighted-F1, completeness and unsupported-content thresholds.

The common audit separates measurement differences from extraction differences more effectively than the historical workbook comparison. Remaining differences can arise from architecture, prompt and decoding behavior, source-boundary preservation, dictionary coverage, span acceptance, schema mapping, or differential sensitivity of the lexical evaluator to paraphrase. Because the adjudicator remains automated, the resulting rank is conditional on this operational definition and must be confirmed against a blinded human-annotated subset before it is presented as intrinsic model superiority.

4.9 Historical-workflow sensitivity analysis: Hybrid NLP versus BERT MULTI

Table 25. Paired HNLP–BERT statistical results

Statistic	Estimate	Interpretation
HNLP weighted F1	85.94%	Shared HNLP/BERT evaluator.
BERT weighted F1	71.51%	Shared HNLP/BERT evaluator.
Mean paired difference	14.10%	HNLP minus BERT across 24 categories.
Median paired difference	13.37%	Robust central category difference.
Category bootstrap 95% interval	12.63% to 15.82%	50,000 resamples of the 24 observed strata.
Category wins	HNLP 24; BERT 0; ties 0	Sign consistency.
Wilcoxon signed-rank	W=0; p=1.19e-07	Paired non-parametric sensitivity test.

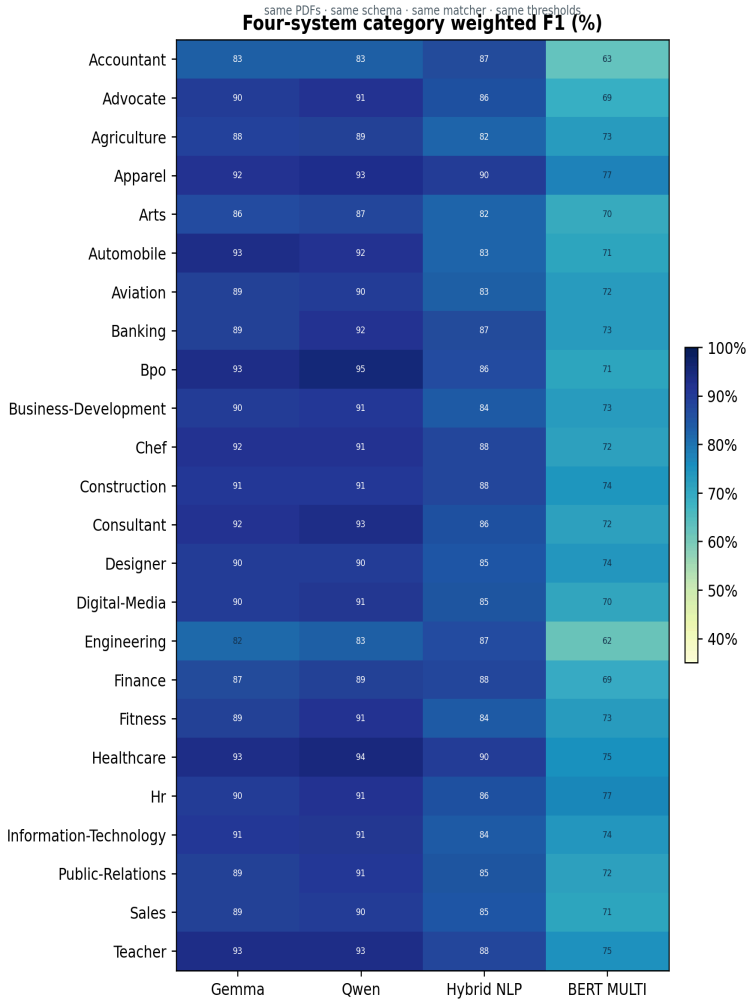

Figure 22. Four-system category weighted-F1 heatmap.

All 96 system–category cells are recomputed by the same PDF-grounded evaluator; values are percentages.

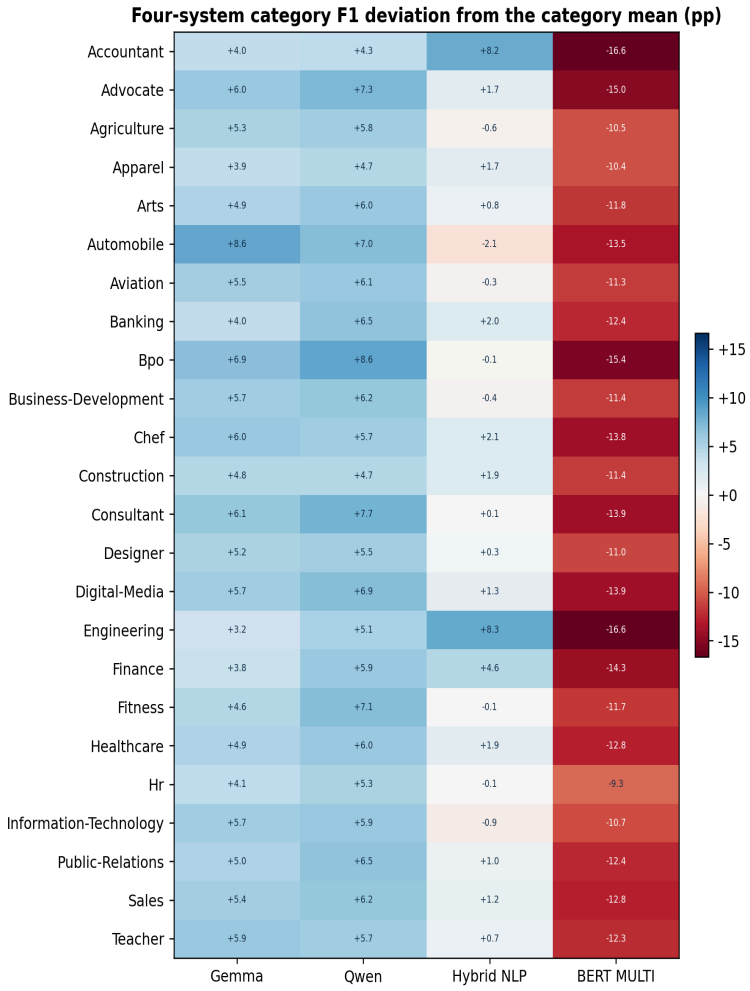

Figure 23. Four-system category deviations from each category mean.

Each cell is the system's weighted F1 minus the four-system mean for that category, in percentage points. This joint display avoids privileged two-by-two contrasts.

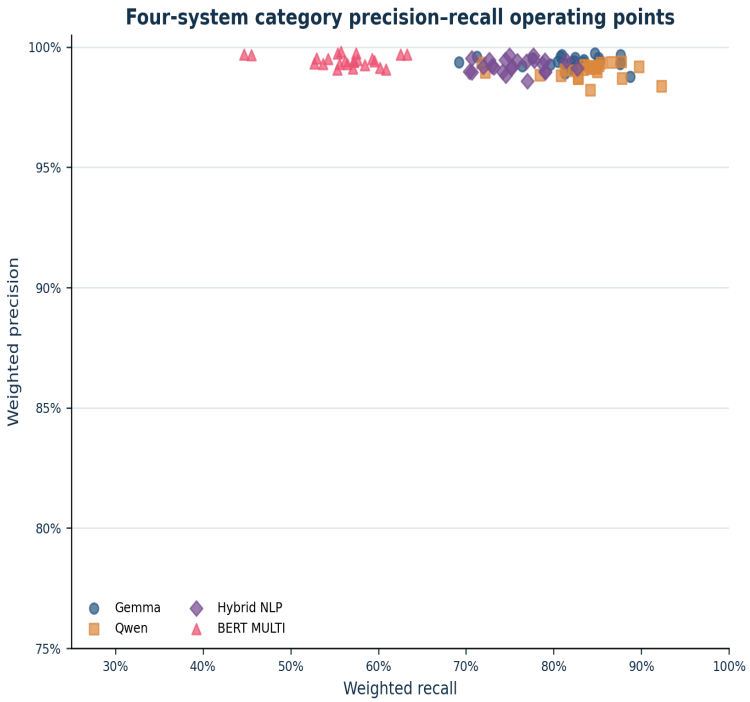

Figure 24. Four-system category precision–recall operating points.

All four category profiles are measured by the common evaluator. Point positions remain conditional on its lexical support assumptions.

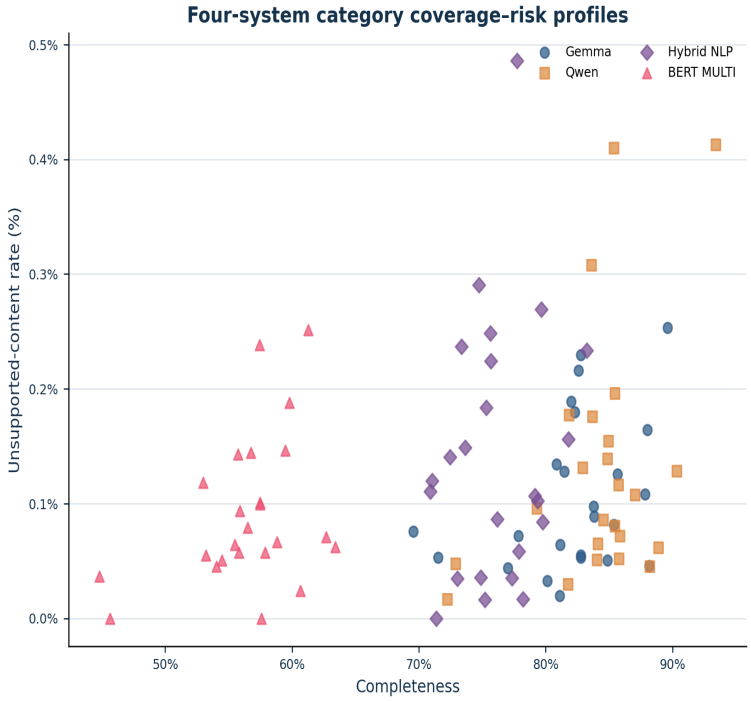

Figure 25. Four-system category completeness–unsupported-content profiles.

The preferred region is lower right. Cross-system distances are evaluator-compatible but not equivalent to independently adjudicated accuracy.

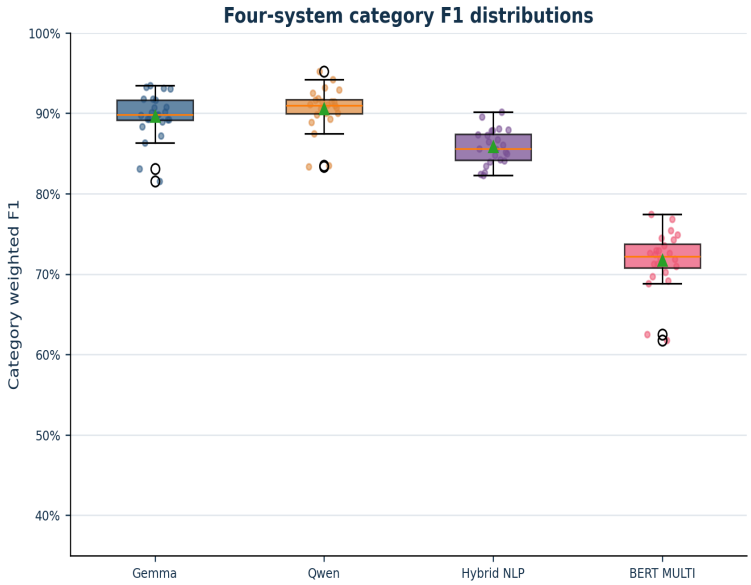

Figure 26. Four-system category weighted-F1 distributions.

Boxes and points show the same 24 categories for every system under the common evaluator.

HNLP exceeds BERT MULTI in every observed category. The mean difference is 14.10 percentage points and the bootstrap interval does not include zero. This is strong evidence that the two concrete implementations occupy different operating points under the shared protocol. It is not evidence that deterministic NLP is universally superior to contextual encoders: HNLP may be better aligned with section-level verbatim extraction, while the BERT pipeline may be optimized for shorter entity spans and then lose recall during mapping and reconstruction.

4.10 Segment-level common results and historical sensitivity

Table 26 preserves the earlier HNLP–BERT workflow as a sensitivity analysis. Figures 27–29 supersede its two-system visual scope by recomputing the same segment families for Gemma, Qwen, HNLP and BERT together. Agreement between layers strengthens a field-specialization hypothesis; disagreement is evidence of evaluator sensitivity and must not be silently averaged.

Table 26. Segment-level HNLP–BERT results

Segment	HNLP F1	BERT F1	Difference	HNLP complete	BERT complete	Interp
experience	86.1%	75.9%	10.2%	76.0%	61.5%	HNLP advanta
skills	90.4%	50.4%	40.0%	84.0%	34.1%	HNLP advanta
education	92.0%	69.2%	22.8%	86.1%	53.5%	HNLP advanta
certifications	53.6%	83.6%	-30.1%	36.9%	73.2%	BERT advanta
profile	85.1%	77.5%	7.6%	75.3%	63.4%	HNLP advanta
language	74.7%	78.9%	-4.2%	60.3%	65.9%	BERT advanta
seniority	71.0%	83.0%	-12.0%	66.2%	76.5%	BERT advanta
category_classification	N/A	N/A	N/A	0.0%	0.0%	Not po

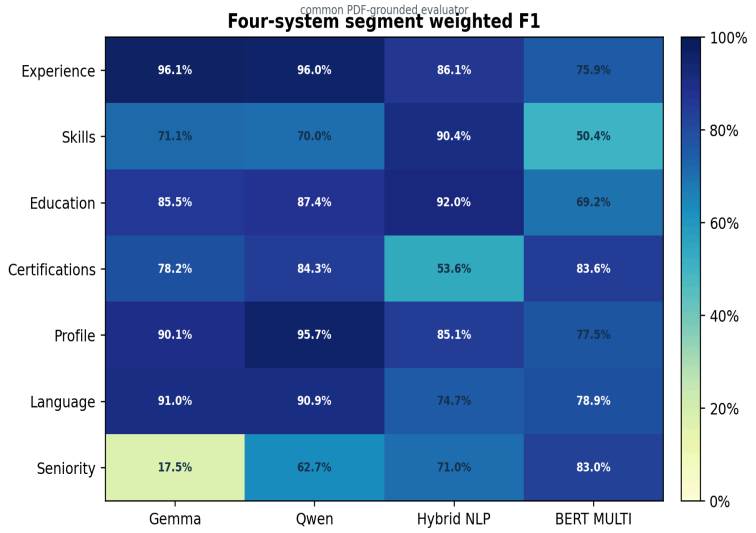

Figure 27. Four-system segment-level weighted F1.

Seven shared segment labels are recomputed for all four systems by the same PDF-grounded evaluator.

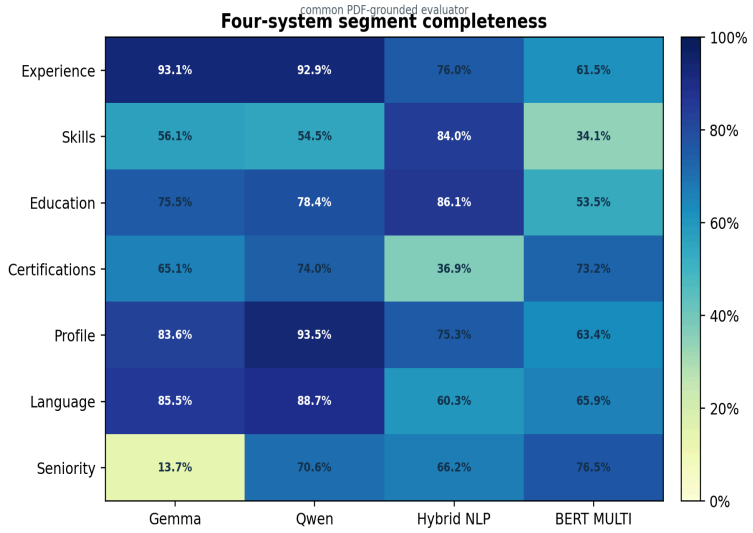

Figure 28. Four-system segment-level completeness.

Completeness complements F1 by showing whether PDF-supported candidate evidence is represented under one common omission rule.

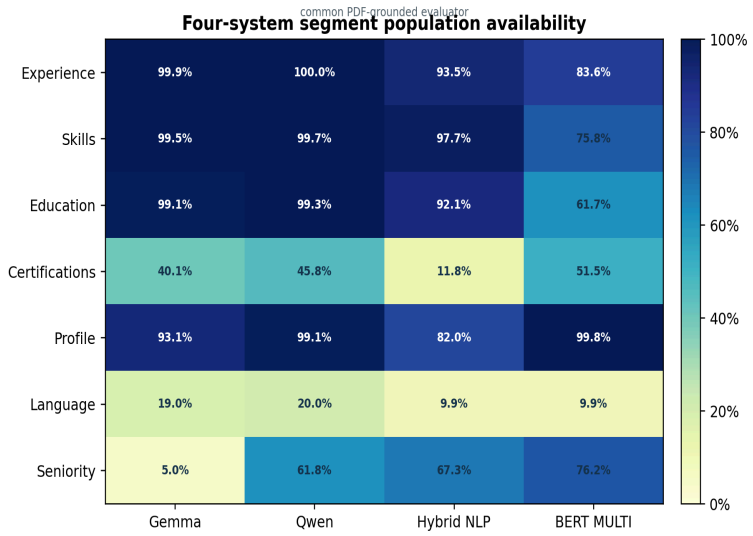

Figure 29. Four-system segment-population availability.

Population is recomputed directly from every raw JSON archive, so all four systems are represented without imputation.

The common segment audit reveals strong task–architecture interaction. HNLP is best for skills (90.4%) and education (92.0%), where dictionaries, synonyms, section boundaries and long-block retention match the target. Qwen is strongest for profile (95.7%) and narrowly for certifications (84.3%); Gemma and Qwen both exceed 90% for experience and language. BERT is strongest for seniority (83.0%) and nearly matches Qwen for certifications (83.6%), consistent with compact entity-like targets, but it falls to 50.4% for skills. These reversals support a field-routed mixture of methods rather than a single universal winner.

4.11 Outcome distributions and why F1 alone is insufficient

Under the common outcome rules, Gemma produced 856 successful, 1,417 partial and 211 failed outcomes; Qwen produced 1,012 successful, 1,314 partial and 158 failed outcomes; Hybrid NLP produced 914 successful, 1,177 partial and 393 failed outcomes; BERT MULTI produced 102 successful, 1,281 partial and 1,101 failed outcomes. BERT's weighted precision of 99.44% coexists with weighted recall of 55.83%. The difference

illustrates why precision cannot be read as end-to-end adequacy: a system may be almost always correct when it emits a fact yet omit enough facts to make complete candidate retrieval unreliable.

Conversely, HNLP's higher recall does not validate every normalized field. Category classification is unpopulated in both combined segment reports, and seniority remains an inferred construct rather than a directly extractive fact. Structural completion and semantic correctness must remain separately measured.

4.12 Claims permitted and prohibited by the merged evidence

Table 27. Five-system claims permitted and prohibited by the evidence

Claim	Status	Reason
ChatGPT has the broadest common-method lexical coverage proxy.	Permitted	All three generative systems share the direct lexical implementation.
Qwen has higher atomic coverage than Gemma in the canonical workbooks.	Permitted with caveat	Same nominal layer, but evaluator heterogeneity limits causal attribution.
HNLP outperforms BERT under the shared HNLP/BERT evaluator.	Permitted with caveat	All 24 categories favour HNLP; automated adjudication remains conditional.
Gemma, Qwen, HNLP and BERT can be compared under the new common automated evaluator.	Permitted with caveat	Corpus, schema, matcher, thresholds, missing-output policy and aggregation are held constant; human gold is still absent.
The highest common-evaluator F1 identifies the intrinsically best architecture.	Prohibited	The result remains conditional on PDF conversion, lexical support, fact splitting and ChatGPT-assisted omission candidates.
ChatGPT is ranked against the other four on atomic factual quality.	Prohibited	ChatGPT is the omission-candidate reference in the common audit and cannot be scored fairly against itself.
MoE routing caused Qwen–Gemma differences.	Prohibited	No dense control, route logs, or factorial isolation.
Reasoning off caused local-model omissions.	Prohibited	No reasoning-on counterfactual run.
BERT multilingual weights definitely executed.	Prohibited	Local/remote model provenance mismatch and no artifact hash.

5. Critical Analysis and Discussion

5.1 Main interpretation

The strongest defensible conclusion is a trade-off, not a universal rank. Qwen provides more complete structured coverage, while Gemma is more conservative. ChatGPT provides the broadest direct lexical capture and the most complete deterministic metadata, but cannot be assigned a directly comparable atomic-fact rank because the available PDF-grounded workbooks evaluate the local model as the prediction and use ChatGPT as a diagnostic reference.

5.2 Why high precision does not justify negative filtering

A supported extracted fact can be used as positive evidence. A missing extracted fact is ambiguous: it may be absent from the source, omitted by the extractor, lost through PDF reading order, or assigned to another segment. Since recall and completeness remain below precision, a RAG system that excludes candidates solely because a structured field is empty would transform extraction omissions into ranking or screening errors.

5.3 Domain variation and evaluator confounding

The category figures show pronounced reversals. Some may reflect genuine model–domain interaction, such as differences in CV length, formatting, terminology, or section structure. However, the source workbooks also vary in atomic-fact granularity, pairing logic, similarity method, and outcome overrides. Consequently, category differences identify where investigation is required; they do not by themselves identify why the difference occurred.

5.4 Why MoE architecture is not yet a causal explanation

Both local filenames describe sparse or mixture-of-experts variants, so this benchmark does not compare a single architectural factor against a controlled dense baseline. Quantization, active parameter routing, prompt interpretation, server implementation, and evaluation workflow are entangled. The phrase 'MoE effect' would therefore overstate the design. The appropriate unit of inference is the end-to-end extraction pipeline.

5.5 Implications for RAG architecture

The original RAG formulation emphasizes external evidence [1]. For CV search, that principle implies that retrieved structured facts must retain links to source passages. A defensible architecture uses field-aware structured retrieval for high-precision positive evidence, raw-section fallback for recall, and criterion-level citations in every generated explanation. Agreement between models may be used as a confidence signal, but shared agreement cannot substitute for source support.

5.6 Analytical attribution: why the results must be questioned

The reported numbers are observations from end-to-end pipelines, not direct measures of intrinsic model intelligence. The dependent variables are produced after PDF decoding, prompt interpretation, generation, parsing, normalization, atomization, matching, and outcome aggregation. A valid analysis therefore asks which mechanisms could generate each observed pattern and what evidence would falsify those mechanisms.

$$Y_{m,i,s} = f\left(X_{PDF}, X_{prompt}, X_{model}, X_{quant}, X_{runtime}, X_{parser}, X_{eval}\right) + \varepsilon_{i,s} \quad (32)$$

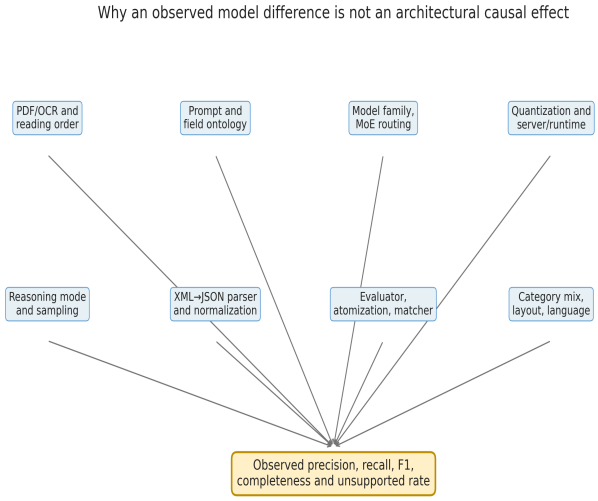

Figure 13. Layered causal decomposition of apparent model deviations.

> Note. Causal-hypothesis diagram only. The paths are plausible mechanisms; no path coefficient or causal effect was estimated.

Table 14. Plausible error sources, observable signatures, and falsification tests

Source	Mechanism	Expected signature	Required test
PDF extraction / reading order	Missing or reordered lines, columns, headers, tables	Low lexical support concentrated in complex layouts	Re-extract with two PDF engines; page-image adjudication
Prompt asymmetry	ChatGPT prompt explicitly permits multi-segment duplication; local prompt requests isolated verbatim blocks	Higher ChatGPT coverage and different segment boundaries	Run all models under one common semantic contract
Field ontology	Experience, skills, seniority, and classification overlap conceptually	Segmentation errors and partial matches	Multi-label source-span annotation with primary/secondary roles
XML-to-JSON mapping	Correct text may be dropped, merged, or assigned to the wrong object	Valid XML but missing JSON facts	Persist raw XML; unit-test every transformation
Metadata population	LLMs asked to infer deterministic values	16.7% local metadata versus 94.7% ChatGPT	Populate IDs, filename and schema version outside the model
Evaluator heterogeneity	Different atomizers, matchers, overrides, and critical-error proxies	Very low cross-category rank association	One frozen evaluator and blinded human gold standard
Quantization	Expert or router perturbation from Q4 representation	Potential model/category-specific instability	Same checkpoint at Q4, Q8 and FP16 on a fixed sample
Context and truncation	Long/irregular documents or output limits omit late evidence	Recall decreases with document/output length	Length-stratified analysis and finish-reason audit
Parallel serving	Four slots alter memory pressure, batching, and tail behavior	Failures correlate with concurrency or request order	Repeat at one and four slots with identical seeds
Reasoning disabled	No explicit deliberative coverage/self-check pass	Omissions in cross-section or derived fields	Controlled on/off/budgeted experiment
Model routing/domain interaction	Tokens activate different expert paths	Stable category- or language-specific reversals	Router traces, entropy, expert overlap and dense controls
Stochastic decoding	Sampling settings and seeds not retained	Within-model rerun variance	Deterministic decoding or repeated runs with recorded seeds

5.7 What the deviations actually show

Qwen's pooled weighted recall exceeds Gemma's by approximately 9.2 percentage points and weighted F1 by 6.3 points, while Gemma's unsupported-content rate is approximately 1.45 points lower. This is consistent with a broader Qwen emission boundary and a more

conservative Gemma boundary. It is not proof that Qwen reasons better or that Gemma's experts are less capable. Prompt sensitivity, routing, quantization, and evaluator differences can produce the same observable trade-off.

The category-level evidence weakens any universal ranking. Qwen wins 15 categories and Gemma nine, but the mean paired advantage is only 3.4 points, its bootstrap interval crosses zero, and the paired tests are not conventionally significant. Pearson association is approximately 0.10 and Spearman association approximately zero. The two evaluation pipelines therefore do not even produce a stable common ordering of category difficulty. That instability is itself a result requiring explanation.

Table 15. Extreme category deviations requiring forensic review

Category	Qwen − Gemma F1	Direction	Observed clue	Academic response
Accountant	+43.0 points	Qwen advantage	Gemma recall 28.7%; Qwen unsupported 17.2%	Audit atomization, prompt output, parser and critical proxy before substantive interpretation
BPO	+24.0 points	Qwen advantage	Gemma recall 31.9%	Check truncation, empty blocks and category-specific matching
Arts	−21.3 points	Gemma advantage	Qwen precision 71.3%; unsupported 27.2%	Inspect unsupported classifications and source-layout effects
Business Development	−20.8 points	Gemma advantage	Qwen recall 55.3%; unsupported 6.1%	Check missing source ID, parser mapping and evaluator rules
Sales	−16.9 points	Gemma advantage	Qwen recall 56.3%	Inspect field-boundary and omission concentration
Advocate	+16.0 points	Qwen advantage	Qwen recall 77.8% versus Gemma 56.1%	Compare evidence spans and partial-credit decisions

Differences of twenty to forty points are too large to absorb into an average and too heterogeneous to describe as a modest generic capability gap. The correct academic response is anomaly investigation: trace source PDF, raw output, mapped JSON, atomic evidence, matcher decision, and aggregate calculation for a stratified sample of the largest reversals.

5.8 Why the two MoE pipelines may differ

The model families differ in total and active capacity, number of experts, layer count, sequence architecture, tokenizer, post-training, multilingual distribution, and chat template [11–12]. Qwen activates a smaller fraction of a larger total parameter pool, while Gemma activates a larger fraction of fewer experts. Qwen's hybrid Gated DeltaNet/attention design and Gemma's local/global attention design may process long or dispersed evidence differently. These differences make domain interaction plausible, but the benchmark contains no internal activations, expert routes, matched dense controls, or full-precision runs. Architecture is therefore a hypothesis generator, not an identified cause.

Prompt behavior is at least as plausible. The ChatGPT contract explicitly asks for high recall, permits duplication across segments, requires source-section context, and contains a silent verification checklist. The local contract is much shorter, requests one block per headline, forbids unrelated text, and contains no explicit final coverage pass. A model that follows each prompt perfectly could therefore display different coverage without any underlying capability difference. Qwen and Gemma also may differ in how they resolve the tension between 'copy only matching text' and ambiguous multi-role spans.

5.9 Competing explanations and discriminating evidence

Table 16. Competing explanations and evidence required to distinguish them

Explanation	Falsifiable prediction	Discriminating experiment
Qwen has intrinsically better extraction recall	Advantage persists under one prompt, parser, gold standard and precision	Paired blinded common-pipeline benchmark
Qwen is simply more aggressive	Recall rises together with unsupported claims and lower abstention	Coverage–risk curves under threshold/prompt variants
Gemma is harmed more by Q4 quantization	Gemma improves disproportionately at Q8/FP16	Within-checkpoint precision ablation
Reasoning off suppresses coverage	Both models improve recall on-mode without unacceptable precision loss	Randomized on/off/budgeted A/B test
Evaluator differences create category reversals	Reversals shrink under one human gold standard	Re-adjudicate extreme categories blind to model
MoE routing specializes by domain	Stable expert-route distributions predict category/segment outcomes	Router logging and dense sibling controls
PDF layout drives omissions	Errors cluster by columns, scans, tables or OCR quality	Layout-stratified page-level audit

5.10 Process critique and evidentiary limits

- The three systems were not prompted identically; the direct comparison measures real pipelines but cannot isolate model capability.
- The local atomic workbooks were produced through model-specific evaluation processes with non-identical headers, matching and overrides.
- The same machine-generated ChatGPT extraction participates as a reference signal, creating a risk of circular preference if agreement is confused with correctness.
- Partial-credit decisions and critical-claim definitions are normative; their sensitivity was not evaluated across alternative weights or adjudicators.
- Only one run per model is represented. Without reruns, generation variance and server-state effects are unmeasured.
- Reasoning mode, quantization level, prompt, parser and architecture all vary or remain uncontrolled; causal attribution is therefore invalid.
- No language, layout, OCR-quality, document-length, or demographic subgroup labels were included in the aggregate analysis.
- Runtime commands were retained, but exact llama.cpp commit, model hash, tokenizer revision, sampling configuration, GPU driver and request log were not.

- The absence of a human-created model-independent gold standard is the central validity limitation.

Accordingly, the corrected report should be read as a rigorous audit of the supplied evidence and as a design for the next experiment. It is not a final causal verdict on the underlying foundation models.

5.11 Five-system causal decomposition

The measured difference between any two systems is not a pure model effect. It is the sum of source-conversion, prompt, architecture, inference, post-processing, evaluator and stochastic components. The decomposition below is conceptual: the components interact and are not separately identified by the current observational archive.

$$\Delta_{observed} = \Delta_{source} + \Delta_{prompt} + \Delta_{architecture} + \Delta_{inference} + \Delta_{mapping} + \Delta_{evaluator} + \varepsilon \quad (56)$$

> A causal statement about architecture requires the other components to be fixed or randomized. None of the cross-family comparisons in the present archive satisfies that requirement.

5.12 Why HNLP and BERT differ

HNLP's inductive bias is section preservation. A heading classified as skills or experience can carry a long block into the output, and dictionary/synonym expansion supplies recall even when local phrasing varies. This favors CVs with conventional headings and densely listed competencies. The same bias can fail on novel headings, tables, creative layouts, code-switching, or skills expressed only through contextual responsibilities.

BERT MULTI's inductive bias is contextual span discrimination. It can identify bounded entities without exact dictionary membership, but the benchmark target often requires complete section reconstruction rather than isolated entities. Chunk boundaries, label batching, a low but non-zero threshold, sanitization, overlap deduplication, field constraints and backfill can each remove a true span. Its high precision and lower recall therefore plausibly reflect both conservative mapping and task mismatch. This explanation is falsifiable through candidate-recall logging before each post-processing stage.

Segment reversals prevent a simplistic conclusion. In the common audit BERT leads seniority and is competitive on certifications, while HNLP leads skills and education; the LLMs lead experience, profile and language. The encoder is therefore not uniformly weak, and the heuristic method is not uniformly more complete. The academically stronger conclusion is conditional field specialization under a stated evaluator.

5.13 Error-source analysis and discriminating tests

Table 28. Architecture-specific errors and discriminating experiments

Hypothesis	Predicted observation	Test that separates it from alternatives
PDF/Markdown conversion loss	All systems miss the same page regions.	Page-image gold and two independent conversion engines.
Prompt-induced conservatism	Local LLM omissions fall after a coverage checklist without model change.	Common prompt factorial with fixed parser/evaluator.
MoE routing instability	Run-to-run or domain-dependent expert routes correlate with errors.	Repeated full-precision runs with route logging and dense siblings.
Reasoning-off effect	Coverage improves under a bounded reasoning budget.	Randomized on/off/budgeted runs with constrained final decoding.
HNLP dictionary dependence	Errors concentrate in unseen terminology and languages.	Dictionary ablation and held-out-vocabulary evaluation.
HNLP heading dependence	Errors concentrate in unconventional layouts/headings.	Heading-alias ablation and layout-stratified annotation.
BERT threshold loss	True candidates exist below 0.10 or are removed later.	Stage-level candidate recall and threshold sweep.
BERT chunk loss	Errors cluster near 220-word boundaries.	Chunk-size/overlap grid with boundary-distance analysis.
BERT mapping loss	Correct spans are present before mapper but absent in JSON.	Persist candidates and score encoder versus mapper separately.
Evaluator preference	Rankings change under blinded human adjudication.	Model-independent dual-rater PDF gold standard.

5.14 Why the evaluation process itself must be questioned

- The same source PDF can be transformed differently before each extractor sees it; source conversion is part of the treatment.
- Prompts and output contracts differ. A high-recall ChatGPT prompt and a conservative verbatim-block prompt do not measure only model capability.
- Machine-assisted atomic adjudication can inherit the linguistic preferences of its reference extraction and matching rules.
- Partial credit fixed at 0.5 is a normative utility assumption. Sensitivity to alternative weights should be reported.
- Category aggregation gives equal inferential weight to heterogeneous occupational strata while document counts and fact density vary.
- Only one observed run per generative configuration prevents estimation of stochastic variance and server-state effects.
- Schema validity proves structural compliance, not evidence support, semantic completeness, fairness or suitability for employment decisions.
- The BERT/HNLP combined reports establish a common evaluator for that pair, but they do not retroactively create a common evaluator for Gemma, Qwen and ChatGPT.

These limitations do not invalidate the benchmark. They change its scientific role: it is a high-value observational audit and hypothesis generator, not a randomized causal comparison. The report therefore separates claims by evidence tier and states the experiment needed to upgrade each descriptive finding into a causal claim.

5.15 A field-routed extraction ensemble

The segment results motivate a field-routed ensemble rather than one global winner. HNLP can propose section-complete skills, experience and education evidence; BERT can propose compact certification, language and seniority spans; generative models can resolve cross-sentence structure and schema completion; and the source PDF remains the final evidentiary authority. Candidate facts should be accepted only when aligned to source spans and should carry extractor, confidence, page/offset and transformation provenance.

$$\hat{y}_{d,f} = g_f(\hat{y}_{d,f,HNLP}, \hat{y}_{d,f,BERT}, \hat{y}_{d,f,LLM}, e_d) \quad (57)$$

> g_f is a field-specific resolver and e_d is PDF-grounded evidence. The resolver may select, union, abstain or request review; it must not create a fact unsupported by the source.

6. Reasoning-Disabled versus Reasoning-Enabled Extraction

6.1 What the runtime flag changes

Both local runs used the llama.cpp option --reasoning off. Current llama.cpp documentation describes reasoning controls at the chat-template/generation layer [13]. The flag suppresses the model's explicit thinking phase; it does not disable the Transformer forward pass, attention, MoE routing, or all internal computation. Equating 'reasoning off' with 'the model did not reason' would therefore be technically incorrect.

Gemma 4 and Qwen3.6 both document configurable thinking behavior [11–12]. Their templates and training procedures differ, so the same server flag need not create identical token-level behavior. The exact llama.cpp revision and rendered prompt should be archived in a future run because reasoning controls evolved across versions.

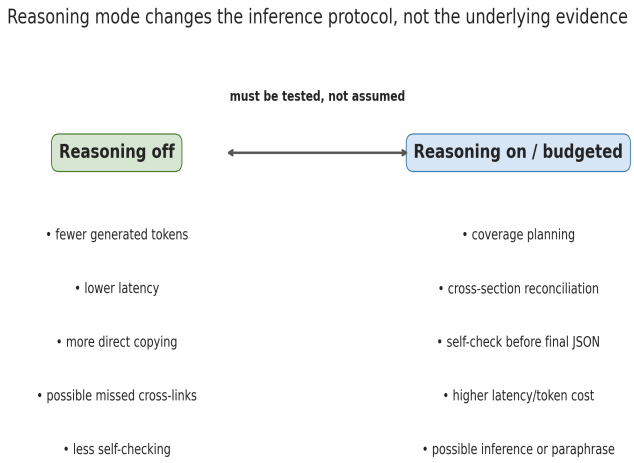

Figure 14. Competing hypotheses for reasoning-disabled and reasoning-enabled extraction.

> Note. Hypothesis diagram only. Both evaluated local runs used reasoning off; no reasoning-enabled extraction run or mode-effect estimate is contained in the benchmark.

6.2 Why reasoning off may help extraction

- The output is shorter and more direct, reducing latency and the opportunity to emit explanatory prose outside the contract.
- A verbatim extraction task often benefits from copying rather than inferential completion.
- Suppressing deliberative text reduces the chance that an inferred normalization or category is accidentally stored as source evidence.
- Fewer generated tokens reduce privacy exposure if server logs are retained.
- Structured-output parsers face fewer special reasoning tags and mode-dependent response formats.

6.3 Why reasoning off may hurt extraction

- The model may not perform an explicit coverage plan across all requested fields before answering.
- Evidence distributed across headings, dates, employers and responsibilities may not be reconciled in a single direct pass.
- Ambiguous spans that support both experience and skills may be assigned narrowly or omitted.
- Derived fields such as seniority and category classification require multi-step evidence aggregation.
- The model may not revisit an unexpectedly empty field or check the final extraction against the source.

These mechanisms are consistent with the omission-heavy profile, but consistency is not causation. The local prompt's conservative design, PDF parsing, quantization and evaluation process can produce the same profile. Chain-of-thought research shows that additional generated computation can improve some multi-step tasks [18], but it does not establish a general benefit for verbatim structured extraction, where inference and paraphrase can themselves become errors.

6.4 Formal quality–cost model for reasoning mode

$$J_r = \lambda_{FN} FN_r + \lambda_{FP} FP_r + \lambda_{fmt} E_{format,r} + \lambda_{lat} T_r + \lambda_{priv} K_{reason,r} \quad (33)$$

$$\Delta_{reason} = Y_{on} - Y_{off}, Y \in \{P, R, F_1, C, U, T\} \quad (34)$$

> **Interpretation.** Reasoning should be selected by a predeclared loss function, not by the assumption that more thinking is always better. A recall gain is unacceptable if unsupported or critical claims exceed a safety margin.

6.5 Required on/off/budgeted experiment

Table 17. Controlled reasoning-mode experiment

Factor	Control or measurement
Weights and quantization	Identical checkpoint for all reasoning conditions
Source and prompt	Byte-identical PDF text, definitions, schema and final-output instructions
Modes	Off, on, and bounded/budgeted thinking where supported
Final output	Same constrained JSON/XML grammar; reasoning content excluded from stored evidence
Randomness	Deterministic decoding or repeated recorded seeds
Primary outcomes	Weighted recall, F1, completeness, unsupported rate, critical claims
Operational outcomes	Time to first token, decode time, total tokens, peak memory, failures
Analysis	Paired document effects with category/layout/language strata and non-inferiority margins

A useful production compromise is hidden or budgeted reasoning followed by constrained final decoding and source-span validation. The reasoning trace should remain ephemeral and access-controlled; storing free-form internal traces containing personal CV data would create unnecessary privacy and governance risk.

6.6 Reasoning modes versus non-generative configuration modes

Reasoning-enabled versus reasoning-disabled is an autoregressive generation intervention. It has no direct analogue in HNLP or BERT MULTI because neither produces a chain of deliberative tokens. Saying that these pipelines run in “non-reasoning mode” would confuse absence of generation with a disabled capability. Their counterfactual controls are algorithmic: dictionary and synonym ablations, OpenNLP model presence, fuzzy thresholds, chunk size, overlap, label batch size, confidence threshold, deduplication policy and schema-mapping constraints.

Table 29. Controlled ablation matrix for generative and non-generative systems

Family	Treatment levels	Held constant	Primary outcomes
Gemma/Qwen	Reasoning off; bounded; on	Checkpoint, quantization, prompt, parser, evaluator	Recall, unsupported rate, tokens, latency
HNLP	Regex only; +dictionary; +fuzzy; +OpenNLP	Markdown, schema, normalization, evaluator	Candidate recall, field F1, latency
BERT MULTI	Threshold × chunk × overlap × mapper	Artifact hash, tokenizer, labels, evaluator	Pre/post-map recall, precision, boundary loss
Five-system ensemble	Field router and abstention policy	Human gold and source-span validator	Utility, coverage, critical errors, cost

A valid comparison should estimate quality–cost frontiers, not compare one reasoning setting with one arbitrary non-generative configuration. Each family should receive a predeclared tuning budget on a development set; the selected configuration should then

be locked before held-out testing.

7. Alternative Models and a Controlled Model-Selection Programme

7.1 Why alternatives must include dense controls

Because both evaluated local models are MoE systems, the present study cannot estimate the effect of sparse routing. The first alternatives should therefore be dense siblings, not unrelated leaderboard winners. Gemma 4 31B dense and Qwen3.6 27B dense provide the most informative within-family controls [14–15]. They reduce, but do not eliminate, confounding from tokenizer, training data, post-training and sequence architecture.

Independent dense models remain useful as external controls. Mistral Small 3.1 24B offers a multilingual, long-context, structured-output-oriented baseline [16]. Phi-4 provides a smaller 14B dense control but has a shorter context and a more English-centric scope [17]. These candidates are proposed for testing; no claim of superiority is made.

Table 18. Alternative local model candidates and their experimental role

Candidate	Architecture	Purpose	Key limitation
Gemma 4 31B-it	Dense, 30.7B	Best within-family control for Gemma MoE	Higher active compute and memory; not architecture-only matched
Qwen3.6-27B	Dense, 27B	Best same-generation family control for Qwen MoE	Higher active compute; compare same prompt and precision
Gemma 4 12B	Dense, ~12B	Lower-resource multilingual family baseline	Lower capacity may trade recall for speed
Mistral Small 3.1 24B	Dense, 24B	Independent multilingual/long-context control with JSON capability	Different tokenizer, training and serving stack
Phi-4 14B	Dense, 14B	Compact lower-bound and efficiency control	16K context and English emphasis require corpus screening
Current Gemma/Qwen ensemble	Two MoE extractors	Exploit complementary category/field behavior	Agreement is not truth; requires PDF support and deduplication

7.2 Model-selection criteria

- PDF-grounded weighted recall, F1, completeness, unsupported rate and critical-claim rate.
- Schema validity, parser recovery, source-span retention and deterministic metadata handling.
- Language-, layout-, length- and category-stratified performance rather than one pooled score.
- Calibration and abstention: whether low-confidence cases can be routed to fallback or review.

- Prefill/decode throughput, tail latency, peak memory, energy per CV and four-slot concurrency.
- Reproducible weights, tokenizer, prompt template, quantization provenance and serving support.
- Privacy, licensing and organizational ability to operate the model locally.

7.3 Factorial benchmark design

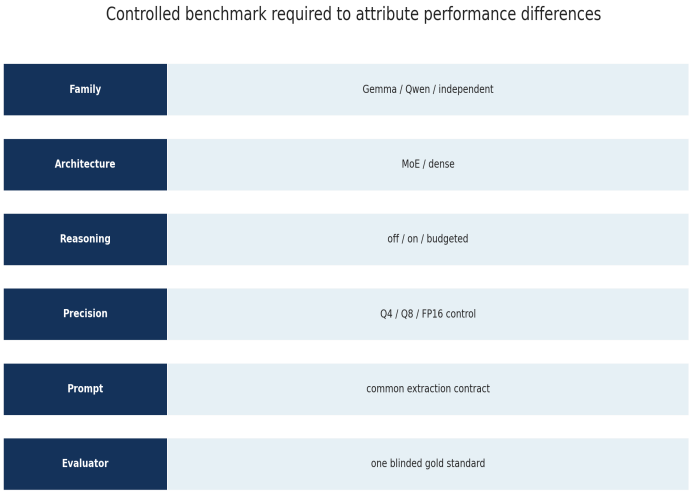

Figure 15. Minimum controlled benchmark for separating model, MoE, reasoning, and quantization effects.

> Note. Proposed factorial design only. It contains no measured benchmark outcome; prompt, source text, parser, and evaluator must be fixed before architecture or reasoning effects can be estimated.

Table 19. Minimum factorial benchmark for model selection

Factor	Levels	Purpose
Model family	Gemma, Qwen, independent dense control	Separates family effects
Architecture	MoE versus dense sibling	Estimates sparse-routing contribution within family
Reasoning mode	Off, on, bounded	Estimates inference-protocol effect
Numerical precision	Q4, Q8, FP16 reference sample	Estimates quantization sensitivity
Prompt	One common contract plus model-native sensitivity arm	Separates fairness from production optimization
Evaluator	One blinded human gold standard and frozen scorer	Removes model-specific evaluation confounding

7.4 Replacement decision rule

$$m = \arg\max_{m} U(m) \text{subject to} P_m \geq P_{\min}, U_m \leq U_{\max}, T_m \leq T_{\max} \quad (35)$$

A replacement should occur only under predeclared non-inferiority constraints for precision and critical errors, plus hardware and latency limits. Selecting the largest pooled F1 after observing the test set would create model-selection bias. The benchmark should therefore define a development set, lock the decision rule, and report one final held-out evaluation.

7.5 Alternative extraction architectures beyond generative LLMs

Alternative models should be selected to test mechanisms, not merely to add fashionable names. Dense transformer controls remain necessary for isolating MoE effects. For extraction itself, the benchmark should also include span encoders, layout-aware document models, constrained seq2seq systems, classical conditional-random-field baselines and deterministic schema parsers. Each alternative should be evaluated under the same source representation and human gold standard.

Table 30. Alternative extraction architectures and experimental roles

Alternative	Experimental role	Expected strength	Critical limitation to test
Dense Gemma/Qwen siblings	Control for sparse routing	Closest within-family causal contrast	Still confounded by size/training differences
Full-precision or higher-bit checkpoints	Quantization ablation	Tests expert/router sensitivity	Higher memory and latency
GLiNER variants with calibrated labels	Span-extraction control	Flexible entity labels and local inference	Long-section reconstruction
LayoutLM-family model	Layout-aware control	Tables, columns and spatial cues	OCR/layout pipeline and domain adaptation
BiLSTM/CRF or transformer-CRF	Fixed-label sequence baseline	Transparent sequence constraints	Ontology rigidity and annotation cost
Constrained seq2seq extractor	Schema-generative dense control	Structured output and cross-span synthesis	Hallucination and decoding cost
Rule/ontology engine	Deterministic baseline	Auditability, speed and abstention	Coverage of novel language
Field-routed ensemble	Operational candidate	Uses complementary specialization	Calibration, complexity and governance

The replacement criterion should be multi-objective and preregistered. A candidate must satisfy non-inferiority bounds for critical precision and unsupported claims, improve recall or utility on held-out data, remain within latency/memory constraints, and preserve source-span provenance. A pooled F1 improvement alone is insufficient for employment-related deployment.

8. Threats to Validity, Ethics, and Responsible Use

Table 9. Threats to validity and mitigations

Validity domain	Threat	Mitigation
Construct validity	Four-token overlap is lexical, not semantic; full-PDF coverage includes non-target text.	Label as proxy; report separately from atomic metrics.
Internal validity	Model-specific evaluator implementations differ.	Canonical recalculation; paired results treated as sensitivity evidence.
External validity	One public-style CV corpus and 24 supplied categories may not represent current applicants or locales.	Avoid population-wide generalization; replicate on in-domain data.
Statistical conclusion	Only 24 category strata; categories are fixed and not independent random samples of all domains.	Bootstrap and tests are descriptive; report effect size and uncertainty.
PDF measurement	Reading order and OCR can distort source text.	Retain PDFs and page-linked evidence; review low-support outliers.
Provenance	Exact ChatGPT build, llama.cpp commit, drivers, seed and sampling are incomplete.	Create immutable run manifests and model/prompt hashes.
Fairness	CV structure and language may correlate with demographic or regional factors not measured here.	Conduct subgroup and language audits before consequential use.
Privacy	CVs contain personal data.	Minimize displayed PII; enforce retention, access control, and purpose limitation.
Automation risk	False negatives can affect employment opportunities.	Human review and PDF fallback for any adverse or exclusionary action.

Employment-related search is a consequential context. NIST AI RMF's emphasis on context, measurement, documentation, monitoring, and human oversight is therefore directly relevant [6]. This benchmark measures technical extraction behavior; it does not establish legal compliance, fairness across protected groups, or suitability for autonomous candidate selection.

9. Recommendations for Deployment and Further Research

Table 10. Recommended deployment controls

Control	Recommendation	Rationale
Evidence retrieval	Index structured segments and source-aligned raw passages together.	Prevents structured omission from becoming a false negative.
Negative decisions	Never infer absence from an empty JSON field without searching the source.	Recall and completeness are materially below precision.
Field routing	Match job criteria only to compatible evidence fields and source contexts.	Reduces semantic leakage from hobbies, metadata, or unrelated sections.
Deterministic metadata	Populate document ID, filename, schema version, and supplied category outside the LLM.	Improves consistency and avoids unnecessary model inference.
Risk scoring	Use source support, model agreement, field quality, and extraction completeness as separate features.	Avoids one opaque confidence score.
Human review	Require review for critical unsupported claims, low-support facts, and adverse decisions.	Controls high-consequence error.
Monitoring	Track field/category drift, missing-file rates, and evidence-support distributions.	Supports lifecycle risk management.
Benchmark redesign	Freeze one atomizer, matcher, thresholds, and blinded adjudication protocol.	Enables a defensible causal model comparison.

9.1 Priority validation study

1. Draw a stratified sample across all 24 categories, document lengths, layouts, and languages.
2. Create one model-independent atomic gold standard from the PDFs.
3. Blind model identity and use at least two trained human raters.
4. Report inter-rater agreement and resolve disagreements through adjudication.
5. Score all five systems with one matcher, one source representation, and one critical-claim definition.
6. Estimate confidence intervals clustered by document and category.
7. Add fairness and language subgroup analyses before employment-related deployment.

10. Reconciliation with Previous Reports

Table 11. Reconciliation with prior aggregate reports

Scope	Metric	Prior report	Canonical recomputation	Difference
Gemma	Parsed/evaluable local JSON	2,482	2,482	+0
Gemma	Weighted precision	94.8%	94.6%	-0.2%
Gemma	Weighted recall	65.5%	64.4%	-1.1%
Gemma	Weighted F1	77.4%	76.6%	-0.8%
Gemma	Completeness	68.2%	67.3%	-0.9%
Gemma	Unsupported rate	1.2%	1.2%	0.0%
Gemma	Mean schema completeness	72.4%	72.5%	0.1%
Gemma	Critical-claim document rate	2.8%	2.8%	-0.0%
Qwen	Parsed/evaluable local JSON	2,483	2,483	+0
Qwen	Weighted precision	95.0%	94.9%	-0.1%
Qwen	Weighted recall	73.7%	73.7%	-0.0%
Qwen	Weighted F1	83.0%	82.9%	-0.1%
Qwen	Completeness	75.5%	75.5%	0.0%
Qwen	Unsupported rate	2.6%	2.7%	0.0%
Qwen	Mean schema completeness	84.1%	80.3%	-3.8%
Qwen	Critical-claim document rate	8.4%	8.3%	-0.1%
Pairwise	Qwen category wins	14	15	+1
Pairwise	Gemma category wins	10	9	-1
Pairwise	Mean Qwen minus Gemma category weighted F1	2.8%	3.4%	0.6%

The earlier Expanded Report workbook is internally labelled as a Banking-only evaluation and cannot serve as the 24-category comparative workbook. The revised comparison uses the user-designated root-level files, parses nonstandard Qwen headers, excludes the Advocate schema artifact from document counts, and reconciles all 2,484 rows for both local-model workbook sets.

11. Conclusion

The five-system evidence supports conditional conclusions rather than a single winner. ChatGPT provides the broadest common direct lexical-coverage proxy. Qwen provides greater Gemma/Qwen canonical atomic coverage, while Gemma is more conservative on unsupported content. Under their distinct shared protocol, Hybrid NLP materially exceeds BERT MULTI in pooled and category weighted F1, but BERT remains stronger for several bounded entity-like fields. The latter reversal demonstrates that pipeline quality depends on the target ontology: section-preserving rules and dictionaries are well aligned with long CV blocks, whereas contextual span models can be stronger for certifications, languages and compact derived labels. Neither result establishes intrinsic architectural superiority. Gemma and Qwen are both quantized MoE systems run with

reasoning disabled, so sparse routing and reasoning effects are not identified. HNLP and BERT have no autoregressive reasoning mode; their relevant controls are dictionaries, optional OpenNLP resources, thresholds, chunks, label batches and mapping rules. The central engineering conclusion is therefore a governed field-routed ensemble with source-span validation, calibrated abstention, raw-document fallback and human review. The central scientific conclusion is equally important: one prompt, one source representation, one blinded human gold standard, repeated runs, artifact hashes and a preregistered factorial design are required before the observed deviations can be attributed to MoE routing, reasoning, BERT encoding or deterministic NLP.

11.1 Deployment trade-off: heuristics, contextual encoders, and LLMs

The practical choice is not between a universally superior heuristic method and a universally superior LLM. It is a multi-objective trade-off among throughput, supported precision, coverage, implementation effort, operational repeatability, maintenance burden and transfer to new CV distributions. The present artifacts contain extraction-quality measurements but do not contain a controlled latency, memory, energy or engineering-effort benchmark. Speed and implementation conclusions must therefore be stated as architecture-based expectations to be verified under identical hardware, concurrency and source-conversion conditions.

Under the newly controlled PDF-grounded audit, the four tested extractors are compared with identical measurement rules: Gemma F_1=89.36%, precision=99.40%, recall=81.16%; Qwen F_1=90.32%, precision=99.04%, recall=83.02%; Hybrid NLP F_1=85.94%, precision=99.25%, recall=75.78%; BERT MULTI F_1=71.51%, precision=99.44%, recall=55.83%. These values replace evaluator mismatch with a common operational definition, but they do not replace human adjudication. Accordingly, measured extraction quality is reported separately from architecture-derived hypotheses about latency, implementation effort and repeatability. The supplied artifacts do not establish an empirical ordering for those deployment properties.

Hybrid NLP avoids neural inference and autoregressive decoding; this mechanism motivates a testable low-latency hypothesis, but latency was not measured and no speed rank is claimed. It produced a high-precision, low-unsupported-content profile under the shared HNLP/BERT evaluator. Its apparent simplicity can be misleading: initial deployment is straightforward when CVs use familiar headings, layouts, languages and terminology, but long-term implementation effort moves into dictionary curation, synonym governance, heading aliases, regular expressions and exception handling. It is stable across repeated runs on the same input and configuration by construction; repeated-run stability was not benchmarked. Transfer to a different CV dataset remains an untested hypothesis because headings, spelling quality, vocabulary, language mix or layout conventions may depart from those rules.

BERT MULTI performs chunked encoder inference rather than autoregressive generation. This architectural difference motivates a latency and repeatability hypothesis, not an observed rank. Contextual subword representations make it less dependent than HNLP on exact dictionary matches. Its implementation is more demanding than a pure heuristic pipeline because tokenizer, ONNX model, DJL runtime, label prompts, chunking, thresholds and schema mapping must remain aligned. It may transfer better to lexical variation and bounded entities, but transfer is not guaranteed across languages, layouts or domains outside the actual model's training and calibration distribution. The benchmark's low BERT recall shows that model-level contextual flexibility does not eliminate pipeline losses in chunking, thresholding and mapping.

Generative LLM extraction can provide semantic flexibility. Prompt changes can add fields and instructions without rewriting a large rule base, and an LLM can relate evidence across distant lines, tolerate paraphrase and spelling variation, and interpret unconventional headings. These properties make it a plausible choice for heterogeneous CV datasets. They are associated with autoregressive serving, more complex runtime requirements, possible provider dependence, stochastic output, prompt and model-version sensitivity, and a non-zero risk of unsupported normalization or completion. Initial implementation may appear easiest because it begins with a prompt, but production implementation remains difficult: constrained output, retries, schema validation, source-span verification, privacy controls, monitoring and regression testing are mandatory.

Table 31. Deployment mechanisms and unmeasured hypotheses for heuristic, encoder and generative extraction

Criterion	Hybrid NLP heuristics	BERT/GLiNER encoder	Generative LLM	Academic interpretation
Latency mechanism (not measured)	Rules and dictionaries; no neural inference or autoregressive decoding	Batched encoder inference over chunks	Autoregressive token generation and model serving	No speed ordering is established; benchmark on identical hardware before ranking
Supported precision	High when rules and section boundaries match the dataset	High at calibrated thresholds; mapping can preserve or remove candidates	Often high for extractive prompts but unsupported synthesis remains possible	Precision must be measured separately from completeness
Recall and semantic coverage	Strong for known headings and vocabulary; weak for implicit or novel expressions	Stronger lexical generalization; weaker for dispersed or section-length targets	Can use paraphrase, distant context and unusual organization	The benchmark does not establish superior coverage outside the measured corpus
Initial implementation	Simple when requirements and vocabulary are narrow	Moderate: model, tokenizer, runtime, labels and mapper	Rapid prototype through prompting; substantial production controls still required	Prototype ease must not be confused with production assurance
Long-term maintenance	Rule, alias, dictionary and exception accumulation	Model/version calibration plus mapping and threshold maintenance	Prompt, model, API/server and regression-management burden	Maintenance shifts location rather than disappearing
Same-input repeatability	Rule execution is deterministic under fixed code and resources	Depends on deterministic runtime behavior and fixed artifacts	Depends on decoding, server and model-version controls	No repeat-run ordering was measured; operational repeatability is not cross-domain validity
Stability across different CV datasets	High only when headings, layouts, language and vocabulary remain in-domain	Moderate when model language/domain and span ontology transfer	Unknown; semantic adaptability and prompt/model drift act in opposing directions	No method is ranked; each materially different dataset requires external validation
Explainability and audit	Strong rule-level traceability	Moderate: spans and scores are traceable; representations are not directly interpretable	Lower internal transparency; source-span evidence can restore output-level auditability	All methods require lineage from final field to source PDF
Best-fit use	High-volume, stable, well-structured and	Explicit entity-like fields across	Heterogeneous, weakly structured and	Field-routed combination is preferable

Criterion	Hybrid NLP heuristics	BERT/GLiNER encoder	Generative LLM	Academic interpretation
	controlled CV formats	moderate lexical variation	semantically implicit CVs	when no single operating condition dominates

Two meanings of stability must therefore remain separate. Repeatability stability asks whether the same pipeline returns the same result for the same CV; the architectures expose different determinism controls, but no repeated-run ranking was measured. Distributional stability asks whether performance remains acceptable when headings, layout, language, profession, vocabulary and document quality change; the present benchmark does not rank the methods on this dimension, which requires external validation and repeated runs. A system can be perfectly repeatable and repeatedly wrong on a shifted dataset.

The recommended production design is consequently conditional and hybrid. Use HNLP for high-throughput fields with stable section and vocabulary conventions; use BERT/GLiNER for compact contextual entities where calibrated span detection is effective; use an LLM for unusual structure, implicit relations and controlled fallback; and validate every accepted fact against the PDF. Route low-confidence, conflicting or high-impact fields to human review. This design converts the speed–precision–coverage trade-off into an explicit routing policy instead of concealing it behind a single model score.

References

[1] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems, 33. https://papers.neurips.cc/paper_files/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[2] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press. https://www-nlp.stanford.edu/IR-book/

[3] Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research, 7, 1–30. https://www.jmlr.org/papers/v7/demsar06a.html

[4] Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for Datasets. Communications of the ACM, 64(12), 86–92. https://doi.org/10.1145/3458723

[5] Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model Cards for Model Reporting. Proceedings of FAT* 2019, 220–229. https://doi.org/10.1145/3287560.3287596

[6] National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

[7] JSON Schema Project. (2020). JSON Schema Core: A Media Type for Describing JSON Documents, Draft 2020-12. https://json-schema.org/draft/2020-12/json-schema-core

[8] Efron, B., & Tibshirani, R. J. (1993). An Introduction to the Bootstrap. Chapman & Hall/CRC. https://doi.org/10.1201/9780429246593

[9] Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. Journal of Machine Learning Research, 23(120), 1–39. https://www.jmlr.org/papers/v23/21-0998.html

[10] Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., & Chen, Z. (2020). GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. https://arxiv.org/abs/2006.16668

[11] Google DeepMind. (2026). Gemma 4 26B A4B model card and architecture record. https://huggingface.co/google/gemma-4-26B-A4B

[12] Qwen Team. (2026). Qwen3.6-35B-A3B model card. https://huggingface.co/Qwen/Qwen3.6-35B-A3B

[13] ggml-org. (2026). llama.cpp command-line documentation: reasoning and reasoning-budget controls. https://github.com/ggml-org/llama.cpp/blob/master/tools/cli/README.md

[14] Qwen Team. (2026). Qwen3.6-27B dense model card. https://huggingface.co/Qwen/Qwen3.6-27B

[15] Google DeepMind. (2026). Gemma 4 31B dense instruction-tuned model card. https://huggingface.co/google/gemma-4-31B-it

[16] Mistral AI. (2025). Mistral Small 3.1 24B Instruct model card. https://huggingface.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503

[17] Microsoft Research. (2024). Phi-4 model card. https://huggingface.co/microsoft/phi-4

[18] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Advances in Neural Information Processing Systems, 35. https://arxiv.org/abs/2201.11903

[19] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. https://arxiv.org/abs/1701.06538

[20] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT, 4171–4186. https://aclanthology.org/N19-1423/

[21] Zaratiana, U., Tomeh, N., Holat, P., & Charnois, T. (2024). GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer. NAACL 2024, 5364–5376. https://doi.org/10.18653/v1/2024.naacl-long.300

[22] Apache Software Foundation. Apache OpenNLP documentation. https://opennlp.apache.org/

[23] Microsoft. ONNX Runtime documentation. https://onnxruntime.ai/docs/

[24] Deep Java Library. DJL documentation. https://djl.ai/

Appendix A. Supplied runtime commands

Gemma

```
llama-server.exe -m ..\models\gemma-4-26B-A4B-it-qat-UD-Q4_K_XL.gguf --device Vulkan1 --split-mode layer -ngl auto -c 65536 -ub 64 -fa auto --reasoning off --host 127.0.0.1 --port 8080 -np 4
```

Qwen

```
llama-server.exe -m ..\models\Qwen3.6-35B-A3B-UD-Q4_K_XL.gguf --device Vulkan1 --split-mode layer -ngl auto -c 65536 -ub 64 -fa auto --reasoning off --host 127.0.0.1 --port 8080 -np 4
```

Appendix A.1. Non-generative implementation record

- Hybrid NLP UI label: Hybrid NLP (Fast, Local); strategy identifier: nlp.
- Hybrid NLP implementation: HNlpExtractionStrategy.java and HNlpExtractor.java.

- BERT UI label: NLP BERT MULTI (DJL); strategy identifier: nlp_bert_multi.
- BERT implementation: DjlTokenClassificationExtractionStrategy.java, DjlTokenClassificationMultilingualExtractionStrategy.java, DjlTokenClassificationRuntime.java, DjlSchemaMapper.java and SchemaDrivenEntityPlanBuilder.java.
- BERT defaults: 220 words per chunk, 30-word overlap, four labels per prompt, 10% confidence threshold, DJL 0.33.0 and ONNX Runtime 1.22.0.
- The source hashes below bind this reconstruction to the inspected Java files. Complete run reproducibility additionally requires the application commit, resolved model and tokenizer hashes, runtime jar/native-provider hashes, schema hash, dictionary/synonym hashes and per-document configuration log.

Table A1. Proprietary HNLP

Method	Java source file	Role in reconstructed method
HNLP	HNlpExtractionStrategy.java	Strategy binding, schema reduction, merge, and validation
HNLP	HNlpExtractor.java	Resource loading, section parsing, dictionaries, fuzzy evidence, normalization
BERT	DjlTokenClassificationExtractionStrategy.java	Base strategy configuration and pipeline assembly
BERT	DjlTokenClassificationMultilingualExtractionStrategy.java	Multilingual preset and local model binding
BERT	DjlTokenClassificationRuntime.java	Chunking, GLiNER inference, span decoding, fallback, and deduplication
BERT	DjlSchemaMapper.java	Constraint filtering, splitting, grouping, templating, and field assignment
BERT	SchemaDrivenEntityPlanBuilder.java	Compilation of x_gliner schema metadata into executable query rules

Appendix B. Statistical interpretation safeguards

- A statistically non-zero paired difference does not establish that the model architecture caused the difference.
- P-values are not measures of practical importance; effect sizes and category dispersion are reported alongside them.
- Category-level bootstrap intervals describe sensitivity to the observed strata, not random sampling from all occupations.
- Document-level tests have large nominal sample size but inherit evaluator heterogeneity and within-category dependence.
- ChatGPT agreement is never used as proof of PDF-grounded correctness.

Appendix C. Reproducibility package

- ChatGPT_Gemma_Qwen_24_Category_Comparative_Audit.xlsx — corrected empirical workbook.
- ChatGPT_Gemma_Qwen_Academic_Statistical_Appendix.xlsx — statistics and figure-source data.
- expand_academic_report.py — figure and report generator.
- figures/ — publication-resolution PNG figures embedded in this report.
- aiextract.json — authoritative extraction schema in the ChatGPT benchmark root.