Budgeted Cascaded Routing for LLM-Based Structured Extraction from Noisy Conversations

Tianzhe Ning
Donghua University
Shanghai, China
2222751@mail.dhu.edu.cn

Guohua Liu*
Donghua University
Shanghai, China
ghliu@dhu.edu.cn

***Abstract*—Structured extraction from noisy conversational transcripts is challenging due to fragmented cross-turn evidence and evolving taxonomies. While LLMs are accurate, exhaustive inference is too costly; lightweight rules or retrieval lack robustness. We propose a *budget-constrained cascaded inference* framework that constructs context blocks, screens *(block, slot)* candidates via dense semantic matching and lexical gating, and queries an LLM only for routed candidates, followed by normalization and order-level aggregation. We also provide an optional uncertainty-based deferral mechanism. We release a de-identified benchmark with standardized splits, dual-granularity evaluation, and strong baselines. Under matched `calls/order` budgets, our cascade outperforms strong baselines while substantially reducing LLM usage.**

I. Introduction

Conversational transcripts from customer-facing scenarios (e.g., test-drive consultations) contain rich signals about user intent, preferences, and constraints. We treat transcripts as fixed noisy inputs from upstream transcription systems. Automatically converting such free-form conversations into structured attributes (slots) enables scalable downstream applications including intent analytics, personalized recommendation, and operational decision support. However, structured extraction from real-world conversations remains challenging: evidence for a slot is often fragmented across turns and speakers, transcript noise (e.g., disfluencies and informal expressions) breaks brittle lexical rules, and business-driven taxonomies evolve continuously, making supervised systems costly to maintain [1], [2].

Large language models (LLMs) offer a promising direction for conversational information extraction [3], [4]. Yet, naively applying an LLM to all utterances or all dialogue windows is prohibitively expensive and latency-limited at scale [5], [6]. Training a dedicated classifier or fine-tuning an LLM can reduce inference cost, but typically requires substantial supervision and is sensitive to taxonomy changes [7]. Continued pretraining on in-domain text is another adaptation strategy [8]; however, our empirical findings suggest it does not reliably improve structured extraction under noisy, fragmented evidence. Conversely, retrieval- or rule-based pipelines are efficient but often sacrifice robustness and struggle with cross-turn dependencies [9], [10]. These trade-offs motivate a controllable inference-time solution that explicitly optimizes accuracy under a strict budget.

This paper asks: *How can we maximize structured extraction quality from noisy conversational transcripts under a strict inference budget?* We propose a *budget-constrained cascaded inference* framework that front-loads cheap, high-recall screening and reserves LLM calls for a small set of informative *(block, slot)* candidates [11]. The framework constructs context blocks to capture cross-turn evidence, performs multi-stage candidate screening that combines dense semantic matching with hard/soft lexical gating, invokes a prompted LLM for closed-set labeling on routed candidates, and normalizes and aggregates predictions at the session/order level to match business evaluation. We further support optional uncertainty-driven human adjudication to selectively review ambiguous cases with minimal manual effort [12].

To facilitate reproducible evaluation of budget-constrained structured extraction, we establish a de-identified benchmark of noisy conversational transcripts with a standardized split, dual-granularity metrics (sentence- and order-level), and strong baselines. The benchmark supports systematic comparisons under matched `calls/order` budgets and enables analysis of accuracy–cost trade-offs. We will release the benchmark together with evaluation scripts and baseline configurations.

Our contributions are threefold:

(1) We formulate budget-constrained structured extraction from noisy conversational transcripts and introduce a unified cascaded inference framework integrating context construction, multi-stage screening, LLM labeling, and order-level aggregation.

(2) We develop an explicit cost model and practical control knobs (e.g., top-K, similarity thresholds, and uncertainty thresholds) that enable systematic accuracy–cost trade-offs under matched `calls/order` budgets.

(3) On the proposed benchmark, extensive experiments and ablations show that our cascade consistently outperforms strong baselines—including keyword rules, single-stage dense screening, direct LLM labeling, parameter-efficient fine-tuning, and continued pretraining—under the same call budgets while substantially reducing LLM usage.

II. Related Work

We briefly review related work in four areas below.

Conversational slot extraction from transcripts. Transcript slot filling is challenging due to dispersed cross-turn

evidence and ASR noise. Recent work studies LLM prompting with constrained outputs for transcript-based extraction, but remains sensitive to noise and long-context effects [3]. We further emphasize *within-transcript* evidence selection under a strict `calls/order` budget.

Instruction-based LLM information extraction. LLMs have been used as universal extractors via instruction following and schema grounding, often with constrained decoding and normalization. Benchmarks such as IEPile and InstructIE highlight schema-conditioned (and bilingual) extraction settings [13], [14]. Different from full-context IE, we route a small set of high-value *(block, slot)* pairs under a strict budget and aggregate predictions at the order/session level.

Budget-aware routing and cascaded inference. Budget-aware methods reduce LLM cost via query routing across model pools or test-time compute allocation, including cascades and learned routers [15], [16], [17]. Our approach is complementary: instead of routing *queries*, we route *(block, slot)* candidates within each transcript via cheap multi-stage screening, enabling explicit accuracy–cost control in `calls/order`.

Weak supervision and human-in-the-loop annotation. Hybrid pipelines combining weak supervision and selective verification are widely used to scale structured prediction. Snorkel formalizes data programming with multiple noisy labeling sources [18], and recent surveys summarize LLM-assisted annotation and verification protocols [19]. We adopt a pragmatic protocol with expert seeds, LLM-assisted prefill, and human review for disputed cases.

III. Problem Formulation and Budget Model

We study structured extraction from noisy conversational transcripts under a strict inference budget. Transcripts are treated as fixed noisy inputs from upstream transcription systems.

A. Task Definition

A transcript is a time-ordered sequence of utterances $\mathcal{U} = \{u_1, \ldots, u_T\}$, and extraction targets are defined by a slot set $\mathcal{L} = \{\ell_1, \ldots, \ell_M\}$ with value spaces $\mathcal{V}_\ell$ (categorical or free-form). Given a session/order identifier o, the goal is to predict

$$\hat{\mathbf{y}}^{(o)} = \{\hat{y}_\ell^{(o)}\}_{\ell \in \mathcal{L}}, \quad \hat{y}_\ell^{(o)} \in \mathcal{V}_\ell \cup \{\varnothing\}, \tag{1}$$

where $\varnothing$ denotes "not mentioned." We evaluate both block/sentence-level labeling and order-level outputs.

B. Context Blocks

To capture cross-turn evidence, we form context blocks with window radius w:

$$b_t = [u_{t-w}, \ldots, u_t, \ldots, u_{t+w}], \tag{2}$$

ignoring out-of-range indices. Let $\mathcal{B} = \{b_1, \ldots, b_T\}$ denote all blocks.

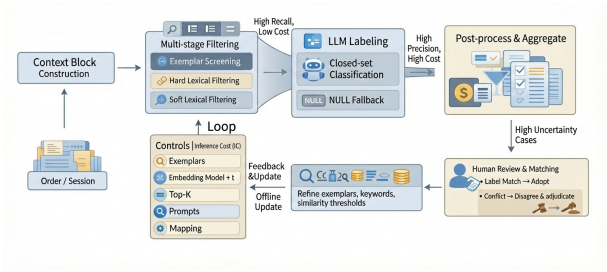

Fig. 1: Overall framework of budget-constrained cascaded inference for conversational structured extraction.

C. Budget-Constrained Inference

A single LLM inference evaluates a *(block, slot)* pair. For order o, let $\mathcal{C}^{(o)} \subseteq \mathcal{B}^{(o)} \times \mathcal{L}$ be the routed set, where $(b, \ell) \in \mathcal{C}^{(o)}$ triggers one LLM call with a slot-specific prompt P_ℓ. In large-scale deployment, the primary bottleneck is the number of LLM calls [15], [16], [17], so we define

$$C_{\text{call}}^{(o)} = \left|\mathcal{C}^{(o)}\right|. \tag{3}$$

Over an evaluation split O, we report average `calls/order` as $\frac{1}{|O|}\sum_{o \in O} |\mathcal{C}^{(o)}|$. Token usage is treated as a secondary factor controlled by prompt templates and block length.

For ambiguous cases, the system may optionally defer a subset to human adjudication. Let $\mathcal{H}$ be the deferred set and

$$C_{\text{hum}} = |\mathcal{H}|. \tag{4}$$

We define the overall cost

$$C(\theta) = C_{\text{call}}(\theta) + \lambda\, C_{\text{hum}}(\theta), \tag{5}$$

where θ controls routing/deferral (e.g., top-K, similarity and uncertainty thresholds), and $C_{\text{call}}(\theta) = \frac{1}{|O|}\sum_{o \in O} C_{\text{call}}^{(o)}(\theta)$ is the average call budget over O. By default we disable deferral ($\mathcal{H} = \emptyset$), so $C(\theta) = C_{\text{call}}(\theta)$ for all main results; when enabled (Table VI) we fix the deferral rate to 2% and report C_{hum} separately without instantiating λ.

D. Optimization Objective

$$\max_{\theta} \ \text{Perf}(\theta) \quad \text{s.t.} \quad C(\theta) \le B, \tag{6}$$

where B is the total budget and $\text{Perf}(\theta)$ is instantiated by sentence/block-level and/or order-level metrics.

IV. Budget-Constrained Cascaded Inference

A. Overview

We propose a budget-constrained cascaded inference framework for structured extraction from noisy conversational transcripts (Fig. 1). The core idea is to front-load cheap, high-recall screening and invoke an LLM only on a small routed set of *(block, slot)* pairs, enabling explicit accuracy–cost control under `calls/order`.

Given a transcript $\mathcal{U}$ for an order/session o, we construct context blocks $\mathcal{B}^{(o)}$ to capture cross-turn evidence. For each slot $\ell \in \mathcal{L}$, we build the routed set $\mathcal{C}^{(o)}$ via a three-stage

screening cascade: (i) exemplar-based dense matching, (ii) hard lexical gating, and (iii) soft lexical gating. We then query an instruction-following LLM with a slot-specific prompt P_ℓ on each routed pair $(b, \ell) \in \mathcal{C}^{(o)}$ under a closed output set, normalize the raw outputs into canonical values, and aggregate block-level evidence to produce the final structured prediction $\hat{\mathbf{y}}^{(o)}$. Optionally, we defer a small subset of ambiguous *(order, slot)* cases to human adjudication.

B. Context Block Construction

Evidence for a slot is often scattered across multiple turns and speakers, making single-utterance labeling unreliable [3]. We therefore represent local conversational context by constructing blocks. For each utterance u_t, we form a block b_t by concatenating neighboring utterances within a window radius w as defined in Section III. We treat w as a tunable hyperparameter.

We apply lightweight formatting (speaker identifiers and delimiters) and cap each block to a maximum length for stable latency and context-window compliance. If the concatenated text exceeds the length limit, we truncate peripheral utterances while preserving the center utterance and its nearest neighbors. All subsequent screening and labeling operate on blocks.

C. Candidate Screening Cascade

Screening determines which block–slot pairs are routed to the LLM and therefore directly controls the call budget. For each slot $\ell \in \mathcal{L}$, screening is applied independently over blocks in $\mathcal{B}^{(o)}$ and is designed to be progressively more selective while remaining much cheaper than LLM inference.

Exemplar matching. We start with recall-oriented dense screening using a small set of slot-specific positive exemplars curated offline. Let $\mathcal{E}_\ell = \{e_\ell^{(i)}\}_{i=1}^{|\mathcal{E}_\ell|}$ denote the exemplar set and $f(\cdot)$ be a sentence embedding model [20], [21], [22]. For a block b, we compute

$$s_{\text{ex}}(b, \ell) = \max_{e \in \mathcal{E}_\ell} \cos\big(f(b), f(e)\big). \tag{7}$$

We retain candidates using a loose threshold or a top-K rule to favor recall at this stage.

Hard lexical filter. To remove obvious false positives introduced by dense matching under noise, a block b is retained if it contains at least one curated hard keyword via substring match. This deterministic gate is inexpensive and improves precision for well-defined expressions.

Soft lexical filter. To recover paraphrases and handle transcription noise, we maintain a set of soft keywords $\mathcal{K}_\ell$ and use a lightweight embedding function $g(\cdot)$ to compute

$$s_{\text{soft}}(b, \ell) = \max_{k \in \mathcal{K}_\ell} \cos\big(g(b), g(k)\big). \tag{8}$$

A candidate is retained if $s_{\text{soft}}(b, \ell)$ exceeds a threshold. This gate complements hard matching by capturing semantically related expressions that do not share exact surface forms.

Only surviving *(block, slot)* pairs form the routed set $\mathcal{C}^{(o)}$. Hyperparameters (e.g., top-K retention and similarity thresholds) control $|\mathcal{C}^{(o)}|$ and thus the accuracy–cost trade-off under the call budget.

D. LLM Labeling Module

For each routed pair $(b, \ell) \in \mathcal{C}^{(o)}$, we query an instruction-following LLM with a slot-specific prompt template P_ℓ to obtain a block-level prediction. The prompt specifies slot semantics, enumerates allowed outputs, and enforces a strict response format to enable deterministic post-processing.

We formulate categorical slots as closed-set prediction: the model must output either a discrete label ID (e.g., $\{0,1,2,3\}$) or a canonical option string chosen from a predefined list, without additional text. Given a block b, we concatenate P_ℓ and b with lightweight formatting and obtain a raw output $\tilde{y}_{b,\ell} \in \mathcal{V}_\ell \cup \{\varnothing\}$. Any invalid or out-of-set response is deterministically mapped to $\varnothing$. We use deterministic decoding (e.g., temperature $= 0$) to stabilize outputs across runs.

The labeling module is intentionally pluggable: different instruction-following LLMs can be substituted without modifying routing, prompts, or aggregation. We evaluate zero-shot, continued pretraining, and supervised fine-tuning variants experimentally while keeping the inference pipeline fixed.

E. Normalization and Order-Level Aggregation

The labeling module produces block-level outputs for routed pairs $(b, \ell) \in \mathcal{C}^{(o)}$. We normalize raw predictions into canonical slot values and aggregate them across blocks to obtain order/session-level outputs. This stage is deterministic, interpretable, and introduces no additional LLM calls.

For each routed pair (b, ℓ), the LLM returns $\tilde{y}_{b,\ell}$. We apply a slot-specific normalization function

$$y_{b,\ell} = \text{Norm}_\ell(\tilde{y}_{b,\ell}) \in \mathcal{V}_\ell \cup \{\varnothing\}, \tag{9}$$

where $\varnothing$ denotes "not mentioned." Invalid/out-of-set responses are mapped to $\varnothing$, ensuring a uniform representation across prompts and model backbones. We treat $y_{b,\ell} = \varnothing$ as insufficient evidence and exclude it from positive aggregation.

Let $\mathcal{B}^{(o)}$ be the set of blocks for order/session o. For each slot ℓ, we collect non-empty block-level predictions

$$\mathcal{Y}_\ell^{(o)} = \{y_{b,\ell} \mid b \in \mathcal{B}^{(o)},\, y_{b,\ell} \neq \varnothing\}. \tag{10}$$

If $\mathcal{Y}_\ell^{(o)}$ is empty, we set $\hat{y}_\ell^{(o)} = \varnothing$. Otherwise, we determine $\hat{y}_\ell^{(o)}$ via majority voting over $\mathcal{Y}_\ell^{(o)}$ and break rare ties by selecting the value that appears earliest in the session, reflecting temporal precedence. For free-form strings after normalization, we first canonicalize equivalent variants and then apply the same frequency-based selection.

F. Uncertainty-Driven Deferral

We optionally defer a small subset of ambiguous *(order, slot)* cases to human adjudication, consistent with the deferred set $\mathcal{H}$ in the budget model. Importantly, deferral does not increase LLM calls; it only adds manual effort that is accounted for separately.

We quantify uncertainty using deterministic disagreement signals from block-level predictions. Let $\mathcal{Y}_\ell^{(o)}$ be the set of non-empty predictions for slot ℓ in order/session o. We defer when $\mathcal{Y}_\ell^{(o)}$ contains more than one distinct value, or when

the vote margin between the most frequent and runner-up values falls below a threshold. Deferred cases are collected into $\mathcal{H}$ and presented to human annotators with supporting blocks for efficient review. The adjudicated value replaces the automatically aggregated decision for the corresponding slot, while non-deferred slots remain unchanged.

G. Model-Agnostic Design

The proposed pipeline is model-agnostic: routing, prompts, normalization, and aggregation are fixed, while we only swap embedding backbones (for screening) and LLM labelers in experiments. This enables controlled comparisons of backbone choice and training paradigms under identical `calls/order` budgets.

V. Data Construction and Training Paradigms

A. Conversational Transcript Dataset

We study real-world conversational transcripts from customer-facing automotive test-drive consultations. Each transcript corresponds to a single session/order o and contains a time-ordered sequence of utterances between a customer and an agent. Transcripts are treated as fixed noisy inputs from upstream transcription systems [3]. We will release de-identified transcripts, labels, standardized splits, and evaluation scripts; no raw audio or personally identifiable information is included.

The corpus is used at multiple granularities: order/session is the unit for final structured outputs, while utterances are grouped into context blocks (Section IV) for screening and labeling. We maintain two data pools: (i) a small expert-labeled seed set used to define slot guidelines and support the labeled development/test splits, and (ii) a larger unlabeled pool used for assisted annotation and large-scale inference.

B. Annotation Protocol

Fully manual labeling of long, noisy transcripts is expensive, so we adopt a hybrid protocol combining expert labeling, LLM-assisted weak supervision, and selective human adjudication [18], [19]. Experts first annotate a seed set at the order/session level to establish slot definitions and provide reference examples.

We then apply the proposed cascaded inference pipeline to unlabeled sessions to generate provisional order-level labels under closed-set constraints. Human annotators review each predicted *(order, slot)* value together with the retrieved supporting blocks: correct predictions are accepted, while disputed cases are escalated to a second adjudication round by senior annotators. The resulting corpus provides complete order-level slot labels (or $\varnothing$ when not mentioned) and a record of disputed cases for later analyses.

C. Training Paradigms

Our inference pipeline is independent of model training; we evaluate three backbone instantiations under identical routing and aggregation. **Zero-shot** directly applies an instruction-following LLM with slot-specific prompts to routed *(block, slot)* pairs. **Continued pretraining (CPT)** further pretrains the backbone with a language-modeling objective on in-domain conversational text (excluding the evaluation split) mixed with general text [8]. **Supervised fine-tuning (SFT)** fine-tunes on labeled sessions produced by the annotation protocol to better align outputs with the slot schema and closed-set format. We compare these variants under matched `calls/order` budgets in experiments.

VI. Experiments

All experiments use real-world conversational transcripts from automotive consultations in the customer's face. Each transcript is a single session/order with time-ordered utterances. Only a small subset is manually annotated to form dev/test splits; a larger unlabeled pool supports assisted labeling and large-scale inference (and provides CPT text when applicable) [18], [19].

Our cascaded inference framework is model-agnostic. We instantiate the labeling module with instruction-following LLMs from the Qwen, Llama, and DeepSeek families [23], [24], [25], [26], and use BGE-M3 for semantic screening and soft lexical matching [22]. All comparisons share the same pipeline and are evaluated under matched call budgets.

We consider three backbone paradigms: zero-shot prompting, continued pretraining (CPT), and supervised fine-tuning (SFT). Context blocks use window radius w (tunable). Routing hyperparameters (thresholds/top-K, exemplar sizes) are tuned on the dev split and fixed unless varied in ablations. Decoding is deterministic and efficiency is measured by `calls/order`. Deferral is disabled by default ($\mathcal{H} = \emptyset$); when enabled, we defer the top 2% most uncertain predictions and report the deferral rate.

A. Evaluation Metrics

We evaluate structured extraction at the session/order level. Let $\mathcal{O}$ denote the set of evaluated sessions/orders and $\mathcal{L}$ the slot set. For each order $o \in \mathcal{O}$ and slot $\ell \in \mathcal{L}$, the ground-truth value is $y_\ell^{(o)} \in \mathcal{V}_\ell \cup \{\varnothing\}$ and the prediction is $\hat{y}_\ell^{(o)} \in \mathcal{V}_\ell \cup \{\varnothing\}$, where $\varnothing$ denotes "not mentioned."

We report overall order-level accuracy by averaging exact matches across all (o, ℓ) pairs:

$$\mathrm{Acc}_{\mathrm{all}} = \frac{1}{|\mathcal{O}||\mathcal{L}|} \sum_{o \in \mathcal{O}} \sum_{\ell \in \mathcal{L}} \mathbb{I}\Big[\hat{y}_\ell^{(o)} = y_\ell^{(o)}\Big]. \tag{11}$$

Because transcript datasets are typically sparse, we additionally report accuracy on empty and non-empty regimes. Define $\mathcal{S}_{\mathrm{emp}} = \{(o, \ell) \mid y_\ell^{(o)} = \varnothing\}$ and $\mathcal{S}_{\mathrm{non}} = \{(o, \ell) \mid y_\ell^{(o)} \neq \varnothing\}$. We compute

$$\begin{aligned} \mathrm{Acc}_{\mathrm{emp}} &= \frac{1}{|\mathcal{S}_{\mathrm{emp}}|} \sum_{(o,\ell) \in \mathcal{S}_{\mathrm{emp}}} \mathbb{I}\Big[\hat{y}_\ell^{(o)} = \varnothing\Big], \\ \mathrm{Acc}_{\mathrm{non}} &= \frac{1}{|\mathcal{S}_{\mathrm{non}}|} \sum_{(o,\ell) \in \mathcal{S}_{\mathrm{non}}} \mathbb{I}\Big[\hat{y}_\ell^{(o)} = y_\ell^{(o)}\Big]. \end{aligned} \tag{12}$$

Here $\mathrm{Acc}_{\mathrm{non}}$ is the "non-empty accuracy" used in practice and directly reflects correctness on mentioned slots.

TABLE I: Order/session-level metrics and cost signals.

Metric	Definition
$\mathrm{Acc}_{\mathrm{all}}$	Exact-match accuracy over all (o, ℓ)
$\mathrm{Acc}_{\mathrm{emp}}$	Accuracy on empty cases ($y_\ell^{(o)} = \varnothing$)
$\mathrm{Acc}_{\mathrm{non}}$	Accuracy on non-empty cases ($y_\ell^{(o)} \neq \varnothing$)
$\mathrm{Prec}_{\mathrm{men}}$	Mention precision using $\mathbb{I}[x \neq \varnothing]$
$\mathrm{Rec}_{\mathrm{men}}$	Mention recall using $\mathbb{I}[x \neq \varnothing]$
C_{call}	Routed calls, $\lvert\mathcal{C}\rvert$

To characterize mention detection independently of value correctness, we also report mention-level precision and recall by treating $\mathrm{Mention}(x) = \mathbb{I}[x \neq \varnothing]$ as a binary decision. Let

$$\mathrm{TP}_{\mathrm{men}} = \sum_{o \in \mathcal{O}} \sum_{\ell \in \mathcal{L}} \mathbb{I}\Big[\hat{y}_\ell^{(o)} \neq \varnothing\Big]\, \mathbb{I}\Big[y_\ell^{(o)} \neq \varnothing\Big] \tag{13}$$

denote the number of correctly predicted mentions, and define

$$\mathrm{P}_{\mathrm{men}} = \sum_{o \in \mathcal{O}} \sum_{\ell \in \mathcal{L}} \mathbb{I}\Big[\hat{y}_\ell^{(o)} \neq \varnothing\Big], \quad \mathrm{G}_{\mathrm{men}} = \sum_{o \in \mathcal{O}} \sum_{\ell \in \mathcal{L}} \mathbb{I}\Big[y_\ell^{(o)} \neq \varnothing\Big]. \tag{14}$$

We then compute

$$\mathrm{Prec}_{\mathrm{men}} = \frac{\mathrm{TP}_{\mathrm{men}}}{\mathrm{P}_{\mathrm{men}}}, \qquad \mathrm{Rec}_{\mathrm{men}} = \frac{\mathrm{TP}_{\mathrm{men}}}{\mathrm{G}_{\mathrm{men}}}. \tag{15}$$

We define $\mathrm{Prec}_{\mathrm{men}} = 0$ when $\mathrm{P}_{\mathrm{men}} = 0$ and $\mathrm{Rec}_{\mathrm{men}} = 0$ when $\mathrm{G}_{\mathrm{men}} = 0$. These metrics are useful when exact value matching is difficult (e.g., taxonomy drift or normalization ambiguity) while the mention decision remains stable. Table I summarizes all metrics.

Since our method explicitly optimizes cost under a call budget, we report efficiency primarily in terms of LLM calls. Let $\mathcal{C} = \bigcup_{o \in \mathcal{O}} \mathcal{C}^{(o)}$ be the routed set over the split; we report total calls $|\mathcal{C}|$ and average `calls/order` $|\mathcal{C}|/|\mathcal{O}|$. When uncertainty-driven deferral is enabled, we also report the deferral rate $|\mathcal{H}|/(|\mathcal{O}||\mathcal{L}|)$. In all accuracy–cost comparisons, we match methods under comparable call budgets by tuning routing hyperparameters (thresholds/top-K) on the development split and then compare $\mathrm{Acc}_{\mathrm{non}}$ and $\mathrm{Acc}_{\mathrm{all}}$ under the matched budgets.

B. Baselines

We compare the proposed cascade with baselines that isolate screening and labeling choices (Table II). Unless otherwise stated, all methods share the same context-block construction, normalization $\mathrm{Norm}_\ell(\cdot)$, and order-level aggregation. For budget-fair comparison, any LLM-based method is evaluated under matched call budgets by tuning routing hyperparameters (e.g., top-K or similarity thresholds) on the development split [15], [16], [17].

Keyword-only rules. Deterministic hard-keyword rules map matched patterns to slot options; otherwise output $\varnothing$. This is a zero-call efficiency baseline.

Embedding-only exemplar matching. Nearest-neighbor exemplar matching in embedding space outputs the exemplar's canonical value if similarity exceeds a threshold; otherwise output $\varnothing$. This tests whether semantic similarity alone can replace LLM labeling.

LLM-full (no screening). The LLM is invoked on all (block, slot) pairs ($\mathcal{C} = \mathcal{B} \times \mathcal{L}$, or the maximum feasible subset). This provides a reference upper envelope but is not deployment-feasible.

LLM-top-K (light screening). Dense exemplar similarity routes top-K blocks per slot to the LLM without lexical gating; K is tuned to match the call budget. This isolates the benefit of lexical gates beyond dense screening.

Single-stage retrieval+LLM. A single dense screening stage (threshold or top-K) selects candidates for LLM labeling. This is a common cost-saving baseline without multi-stage lexical gating.

Cascade ablations and backbone variants. We also evaluate component-removal ablations of our cascade (e.g., removing exemplar/hard/soft gates) and labeling-backbone variants under the same budgeted routing and evaluation protocol; details are reported in the corresponding sections.

TABLE II: Baseline families. "budgeted" means routing is tuned to match a target `calls/order`.

Method	Screening	LLM Calls	Order Aggreg.
Keyword-only rules	lexical	0	yes
Embedding-only matching	semantic	0	yes
LLM-full	none	high	yes
LLM-top-K	semantic	medium	yes
Single-stage retrieval+LLM	semantic	budgeted	yes
Ours (full cascade)	semantic+lexical	budgeted	yes

TABLE III: Main results under matched call budgets (`calls/order`). LLM-based rows use the same labeling backbone (Qwen3-8B by default) and match budgets by tuning routing on the development split.

Method	Target budget	$\mathrm{Acc}_{\mathrm{all}}$	$\mathrm{Acc}_{\mathrm{non}}$	$\mathrm{Prec}_{\mathrm{men}}$	$\mathrm{Rec}_{\mathrm{men}}$
Keyword-only rules	0.0	0.845	0.602	0.66	0.51
Embedding-only exemplar matching	0.0	0.861	0.641	0.69	0.58
Single-stage retrieval+LLM	1.0	0.889	0.712	0.74	0.69
LLM-top-K (no lexical)	1.0	0.893	0.721	0.75	0.71
Ours (full cascade)	1.0	**0.904**	**0.756**	**0.78**	**0.73**
Single-stage retrieval+LLM	2.0	0.903	0.748	0.79	0.74
Ours (full cascade)	2.0	**0.913**	**0.781**	**0.81**	**0.76**
LLM-full (reference)	–	0.918	0.798	0.82	0.78

C. Main Results

Table III summarizes the main comparisons under *matched* LLM call budgets. Unless otherwise stated, all methods share the same context-block construction, slot-specific normalization, and order-level aggregation; they differ only in routing/screening strategies and (when applicable) the LLM used for labeling. We report $\mathrm{Acc}_{\mathrm{all}}$, $\mathrm{Acc}_{\mathrm{non}}$, and mention precision/recall, and express budgets in `calls/order`. Routing hyperparameters are tuned on the labeled development split to meet each target budget, and results are reported on the held-out test split.

TABLE IV: Labeling backbone comparison for **Ours** under matched budgets. "Thinking" is controlled by inference-time parameters when supported (e.g., Qwen3 `enable_thinking`).

Backbone	Thinking	Budget	$\mathrm{Acc}_{\mathrm{all}}$	$\mathrm{Acc}_{\mathrm{non}}$	$\mathrm{Prec}_{\mathrm{men}}/\mathrm{Rec}_{\mathrm{men}}$
Qwen3-8B	off	1.0	0.904	0.756	0.78 / 0.73
Qwen3-8B	on	1.0	0.907	0.764	0.79 / 0.74
Llama-3.1-8B	–	1.0	0.900	0.748	0.77 / 0.72
DeepSeek-V3	–	1.0	0.910	0.770	0.80 / 0.74
DeepSeek-R1	on	1.0	0.912	0.776	0.80 / 0.75
Qwen3-8B	off	2.0	0.913	0.781	0.81 / 0.76
Qwen3-8B	on	2.0	0.916	0.788	0.82 / 0.77
Llama-3.1-8B	–	2.0	0.910	0.774	0.80 / 0.75
DeepSeek-V3	–	2.0	0.918	0.792	0.82 / 0.77
DeepSeek-R1	on	2.0	0.920	0.797	0.82 / 0.78

Unless otherwise stated, the default labeling backbone is **Qwen3-8B** with deterministic decoding (temperature $= 0$) and *thinking disabled* (e.g., `enable_thinking=false` when supported by the runtime).

Across budgets, the proposed cascade consistently achieves the best accuracy among budget-feasible methods. Under a strict budget (1.0 call/order), the cascade improves $\mathrm{Acc}_{\mathrm{non}}$ over single-stage routing, indicating that staged screening retains high-value evidence while suppressing noisy false positives. As the budget increases, all LLM-based methods improve, but the cascade remains superior at comparable budgets, suggesting that lexical gates provide robustness complementary to dense screening.

D. Backbone Comparison

We compare labeling backbones under an *identical* inference pipeline (Table IV). We keep context blocks, the screening cascade, prompts, normalization, and aggregation fixed; only the LLM (and, when supported, *thinking*) changes. Budgets (1.0 and 2.0 calls/order) are matched by tuning routing on the development split, with deterministic decoding and the same closed-set constraints.

Overall, stronger backbones and/or enabling thinking improve accuracy under the same call budget, with larger gains under tighter budgets where each call must extract maximal value. Importantly, the qualitative conclusion is stable across backbones: the cascade provides a robust inference scaffold, and better labelers translate into predictable gains without changing routing or aggregation.

E. Efficiency–Accuracy Trade-off

We characterize the efficiency–accuracy trade-off by sweeping routing hyperparameters that control the routed set size, including exemplar similarity thresholds, soft-lexical thresholds, and per-slot/per-order top-K retention [15], [16], [17]. For each configuration, we record realized `calls/order` and compute accuracy on the test split. Fig. 2 shows the resulting accuracy–cost envelopes for $\mathrm{Acc}_{\mathrm{non}}$ and $\mathrm{Acc}_{\mathrm{all}}$.

We compute an empirical upper envelope by sorting configurations by realized `calls/order` and taking the running maximum accuracy under cost $\leq x$. Across budgets, the proposed cascade forms a stronger accuracy–cost envelope than

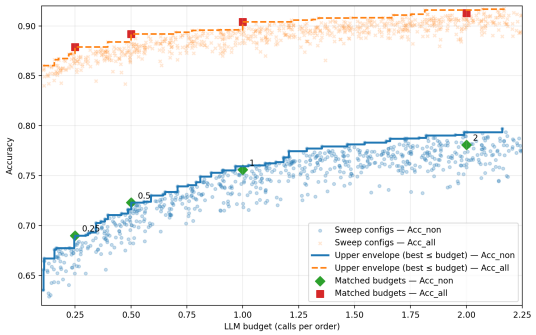

Fig. 2: Accuracy–cost trade-off of our cascaded routing (sweeping routing hyperparameters).

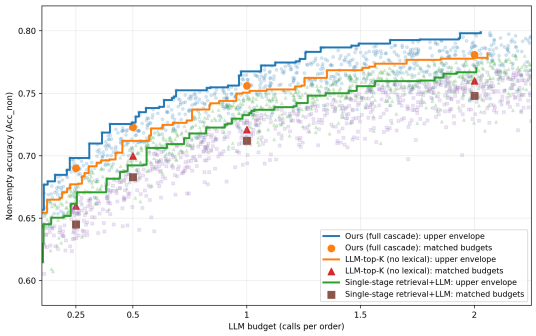

Fig. 3: Cross-method accuracy–cost envelopes under matched call budgets.

single-stage screening baselines (Fig. 3). Under tight budgets, dense screening alone may route weakly related candidates (hurting precision) or miss rare but decisive evidence (hurting recall); lexical gates improve selectivity while preserving high-value candidates, increasing $\mathrm{Acc}_{\mathrm{non}}$ at the same cost. As budgets increase, all LLM-based methods improve, but the gap remains consistent, indicating that staged screening provides robustness complementary to the labeler. We also report representative operating points in Table V.

F. Ablation Studies

We conduct ablations to isolate contributions of cascade components (Table VI). Unless otherwise stated, we fix context blocks, prompts, normalization, and aggregation, and use Qwen3-8B with thinking disabled. All variants are budget-matched at 1.0 call/order by re-tuning remaining routing thresholds on the development split.

Removing exemplar matching causes the largest drop in $\mathrm{Acc}_{\mathrm{non}}$, indicating that semantic screening is essential for paraphrased or noisy mentions. Removing the hard lexical gate mainly reduces precision, while removing the soft lexical gate reduces robustness to surface variations and transcription noise. Disabling aggregation or normalization degrades order-level accuracy, confirming that cross-block evidence fusion and slot-specific normalization are necessary. Finally, uncertainty-driven deferral improves accuracy without increas-

TABLE V: Representative operating points on the trade-off curves (best $\mathrm{Acc}_{\mathrm{non}}$ per budget).

Calls/order	Single-stage retrieval+LLM		Ours (full cascade)	
	$\mathrm{Acc}_{\mathrm{non}}$	$\mathrm{Acc}_{\mathrm{all}}$	$\mathrm{Acc}_{\mathrm{non}}$	$\mathrm{Acc}_{\mathrm{all}}$
0.25	0.645	0.870	**0.690**	**0.879**
0.50	0.683	0.881	**0.723**	**0.892**
1.00	0.712	0.889	**0.756**	**0.904**
2.00	0.748	0.903	**0.781**	**0.913**

TABLE VI: Ablations under a matched budget of 1.0 call/order.

Variant	$\mathrm{Acc}_{\mathrm{all}}$	$\mathrm{Acc}_{\mathrm{non}}$	$\mathrm{Prec}_{\mathrm{men}}$	$\mathrm{Rec}_{\mathrm{men}}$
Full cascade (Exemplar + Hard + Soft)	**0.904**	**0.756**	**0.78**	**0.73**
w/o exemplar matching (Hard + Soft only)	0.892	0.721	0.80	0.66
w/o hard lexical filter (Exemplar + Soft)	0.897	0.734	0.74	0.74
w/o soft lexical filter (Exemplar + Hard)	0.899	0.741	0.77	0.70
Exemplar only (no lexical)	0.893	0.721	0.75	0.71
w/o order aggregation (block-level only)	0.887	0.709	0.76	0.69
w/o normalization (raw string voting)	0.890	0.715	0.75	0.70
w/o deferral (no human review)	0.904	0.756	0.78	0.73
+ deferral (top uncertain 2%)	0.912	0.771	0.79	0.74

ing LLM calls, providing a complementary reliability knob under the same call budget.

VII. Conclusion

We studied budget-constrained structured extraction from long, noisy conversational transcripts, where the dominant deployment cost is the number of LLM calls. We introduced a cascaded screening-and-routing framework that selects a small set of high-value (block, slot) candidates, and performs closed-set LLM labeling with deterministic normalization and order-level aggregation under matched call budgets. Experiments on real-world automotive consultation transcripts demonstrate that the proposed cascade consistently improves extraction quality over strong single-stage routing baselines at the same call budget, and achieves a better efficiency–accuracy Pareto frontier. In future work, we will explore stronger uncertainty modeling for deferral, tighter latency/token-aware budgets, and broader evaluations across domains and slot schemas.

References

[1] M. Rana, K. Hacioglu, S. Gopalan, and M. Boothalingam, "Zero-shot slot filling in the age of llms for dialogue systems," in *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, 2025, pp. 697–706.

[2] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, and E. Chen, "Large language models for generative information extraction: A survey," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186357, 2024.

[3] G. Sun, S. Feng, D. Jiang, C. Zhang, M. Gasic, and P. Woodland, "Speech-based slot filling using large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 6351–6362.

[4] J. Dagdelen, A. Dunn, S. Lee, N. Walker, A. S. Rosen, G. Ceder, K. A. Persson, and A. Jain, "Structured information extraction from scientific text with large language models," *Nature communications*, vol. 15, no. 1, p. 1418, 2024.

[5] X. Zhang, Z. Huang, E. O. Taga, C. Joe-Wong, S. Oymak, and J. Chen, "Efficient contextual llm cascades through budget-constrained policy learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 91 691–91 722, 2024.

[6] M. Yue, J. Zhao, M. Zhang, L. Du, and Z. Yao, "Large language model cascades with mixture of thought representations for cost-efficient reasoning," in *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[8] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8342–8360.

[9] V. Karpukhin, B. Oguz, S. Min, P. S. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering." in *EMNLP (1)*, 2020, pp. 6769–6781.

[10] S. Robertson, H. Zaragoza *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and trends® in information retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[11] M. Rashid, J. Meem, Y. Dong, and V. Hristidis, "Ecorank: Budget-constrained text re-ranking using large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 13 049–13 063.

[12] C. Fanconi and M. van der Schaar, "Cascaded language models for cost-effective human–ai decision-making," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[13] H. Gui, L. Yuan, H. Ye, N. Zhang, M. Sun, L. Liang, and H. Chen, "Iepile: Unearthing large scale schema-conditioned information extraction corpus," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2024, pp. 127–146.

[14] H. Gui, S. Qiao, J. Zhang, H. Ye, M. Sun, L. Liang, J. Z. Pan, H. Chen, and N. Zhang, "Instructie: A bilingual instruction-based information extraction dataset," in *International Semantic Web Conference*. Springer, 2024, pp. 59–79.

[15] L. Chen, M. Zaharia, and J. Zou, "Frugalgpt: How to use large language models while reducing cost and improving performance," *Transactions on Machine Learning Research*, 2023.

[16] D. Ding, A. Mallick, C. Wang, R. Sim, S. Mukherjee, V. Rühle, L. V. Lakshmanan, and A. H. Awadallah, "Hybrid llm: Cost-efficient and quality-aware query routing," in *The Twelfth International Conference on Learning Representations*, 2024.

[17] D. Ding, A. Mallick, S. Zhang, C. Wang, D. Madrigal, M. D. C. H. Garcia, M. Xia, L. V. Lakshmanan, Q. Wu, and V. Rühle, "Best-route: Adaptive llm routing with test-time optimal compute," in *Forty-second International Conference on Machine Learning*, 2025.

[18] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: rapid training data creation with weak supervision," *The VLDB Journal*, vol. 29, no. 2, pp. 709–730, 2020.

[19] Z. Tan, D. Li, S. Wang, A. Beigi, B. Jiang, A. Bhattacharjee, M. Karami, J. Li, L. Cheng, and H. Liu, "Large language models for data annotation and synthesis: A survey," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 930–957.

[20] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781.

[21] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[22] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," *arXiv preprint arXiv:2402.03216*, 2024.

[23] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[24] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv e-prints*, pp. arXiv–2407, 2024.

[25] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[26] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi *et al.*, "Deepseek-r1 incentivizes reasoning in llms through reinforcement learning," *Nature*, vol. 645, no. 8081, pp. 633–638, 2025.