OmniQUM: Multilingual Query-to-Location Understanding Model for Local Search

Avijit Shah **Srini Dwarakanathan** **Yash Kamdar**
Danielle Class **Alessandro Moschitti**
Amazon
{shahavij, dwarakas, ykamdar, dcclss, amosch}@amazon.com

Abstract

Local search systems must determine WHERE to search before retrieving results. We present OmniQUM, a fine-tuned 7B model for generative location disambiguation: given a query and geo-coordinates, it produces a disambiguated location target and bounding box decision in a single pass. Unlike NER + gazetteer pipelines, it jointly reasons over query intent and geographic context. We train across 14 locales and 7 languages with a failure-driven agentic pipeline generating targeted samples iteratively. We introduce expanding donut retrieval, sampling candidates from concentric, slightly overlapping rings to improve recall over naive bounding box retrieval. On 52K queries reflecting real traffic distribution, we achieve F1 0.94 on location disambiguation and 32.7% retrieval recall improvement, surfacing 86K additional relevant POIs. Deployed with vLLM and bf16 on dual GPUs, we achieve sub-150ms TM95 latency at 0.12× baseline cost, with 7–38% cache hit rates.

1 Introduction

Local search is a foundational capability for voice assistants and mobile applications. When a user asks "restaurants in Portland" or "Starbucks on High Street," the system must first determine *which* Portland or *which* High Street the user means before it can retrieve relevant results. This location disambiguation step is critical for returning relevant Point of Interests (POIs) in a multilingual, multilocale setting: getting it wrong means searching the wrong geographic area entirely, returning irrelevant results regardless of how good the downstream ranking is.

Location resolution in search systems has been addressed through rule-based geocoding, NER-based extraction, entity linking, and more recently LLM-based approaches (Zhang and Bethard, 2025; Zhang et al., 2020; GDELT Project, 2024; Hu et al., 2024). Traditional approaches extract location spans via NER, then link them to a gazetteer such as GeoNames (Zhang and Bethard, 2023; Halterman, 2023). This pipeline is fundamentally limited: disambiguation operates independently of extraction, and neither step has access to the user's real-time geo-coordinates. When a user in Leeds searches "restaurants on High Street," the system returns hundreds of candidates worldwide with no principled selection mechanism.

The challenge is acute: search queries are short (2–5 words) with minimal context for disambiguation, and multilingual markets amplify this with cross-country ambiguity (e.g., "Córdoba" in Spain, Argentina, Mexico, and Colombia), code-switching, brand names containing place names, and compound multi-reference queries. Table 1 illustrates these challenges. These demonstrate why traditional extract-then-link pipelines fail: they require joint reasoning over query text, geographic context, and world knowledge. For instance, "Manhattan Bagel near me" requires knowing it is a chain, not a location, which NER pipelines cannot reliably determine.

We propose a generative alternative. OmniQUM is a fine-tuned 7B model that takes the user's query and geo-coordinates as joint input and produces a disambiguated location target and a bounded/unbounded retrieval decision in a single pass. By conditioning on geo-coordinates, the model resolves ambiguity even when query text alone is insufficient, and by generating full disambiguation chains rather than linking to a gazetteer, outputs are directly geocodable (e.g., "Portland, OR, USA#Location" → precise coordinates). We train via a failure-driven agentic pipeline that iteratively targets disambiguation gaps, validated by scaling from en_US to 14 locales across four iterations. We further introduce an expanding donut retrieval strategy that samples from successive concentric and slightly overlapping rings rather than

Challenge	Query	User Geo	Predicted Target
Geo disambiguation	find coffee shops in Portland	45.52, -122.68 (Portland, OR)	Portland, OR, USA#Location
	Cervecerías en Portland	43.66, -70.26 (Portland, ME)	Portland, Maine, EE.UU.#Location
Complete extraction	ibis Budget Sheffield Centre St Marys Gate	53.38, -1.46 (Sheffield, UK)	St Marys Gate, Sheffield, England, UK#Location
Neighborhood	cafes in Greenwich Village	40.71, -74.01 (New York, NY)	Greenwich Village, NYC, NY, USA#Location
Brand-in-entity	Manhattan Bagel near me	40.06, -74.16 (Lakewood, NJ)	*null*
Compound query	Cafes on Queen St W near Trinity Bellwoods	43.65, -79.42 (Toronto, ON)	Queen St West, Trinity Bellwoods, Toronto, ON, Canada#Location
Street disambiguation	is Costa on Market Street open today	53.48, -2.23 (Manchester, UK)	Market St, Manchester, England, UK#Location
Code-switching	find me Apotheke near Hauptbahnhof	48.14, 11.56 (München, DE)	Hauptbahnhof, München, Bayern, Deutschland#POI
Informational	when does Eiffel Tower open	48.86, 2.29 (Paris, FR)	*null*

Table 1: Disambiguation challenges requiring joint reasoning over query and geo-coordinates.

a flat bounding box (Figure 1) to improve downstream retrieval recall. We report deployment with systematic inference optimization across GPU architectures, achieving TM95 sub-150ms latency at 50 TPS (transactions per second) serving 14 locales. Our contributions are:

Geo-conditioned generative disambiguation: a generative model that conditions on real-time user coordinates for location disambiguation in local search queries, jointly predicting the search target (disambiguated location where to search) and the retrieval strategy (whether to apply a bounding box around the target) in a single model call. To our knowledge, this is a novel approach for deployed local search.

Failure-driven agentic data pipeline: a systematic agentic methodology that identifies disambiguation failure patterns, categorizes them into 15 scenario types, and generates targeted training samples per locale iteratively. The pipeline is schema-driven, enabling expansion to new locales or extraction fields without code changes.

Expanding donut retrieval: given a geocoded target center c, we query a fixed schedule of m concentric, slightly overlapping rings whose radii grow with distance, keep the top-n candidates per ring, and take the union. This samples densely near the target while retaining distant coverage, improving recall over selecting the same number of candidates from a single flat bounding box.

Deployment-scale inference: latency and cost benchmarks from a system serving 14 locales across 7 languages, optimized across GPU architectures, serving frameworks, and numerical precision, with gains from query caching.

2 Related Work

2.1 Toponym Resolution and Geoparsing

Toponym resolution disambiguates place names to geographic coordinates or database entries, traditionally via generate-and-rank: candidates are retrieved from a geospatial ontology, then reranked using heuristics or learned models.

GeoNorm (Zhang and Bethard, 2023) generates candidates from GeoNames and reranks them with a transformer incorporating ontology features, resolving entities iteratively from less to more ambiguous. **Mordecai 3** (Halterman, 2023) performs end-to-end geoparsing using neural ranking over GeoNames entries for news and event reports. Both operate on long-form documents with rich context and do not incorporate user location signals. A recent survey by Zhang and Bethard (2025) notes that most systems target document-level corpora and that short-text geocoding remains an open challenge.

2.2 LLM-Based Location Resolution

Recent work has explored LLMs for toponym resolution. **Hu et al.** (Hu et al., 2024) fine-tune Llama2-13B with LoRA, producing structured addresses from place names in context. This is the closest prior work in output format, but operates on English-only document text, does not condition on geo-coordinates, and is not deployed.

The **GDELT Project** (GDELT Project, 2024) found that zero-shot GPT-3.5/4.0 are unreliable geocoders, hallucinating coordinates and performing worse in underrepresented geographies, motivating our fine-tuning approach. **ReaGeo** (Cui et al., 2026) generates geohash coordinates end-to-end from address text using chain-of-thought reasoning, but does not handle search queries or condition on user location. **Masis and O'Connor** (Masis and O'Connor, 2026) address compositional location references using LLMs, a complementary problem handling spatial relations rather than entity disambiguation. For humanitarian applications, Cafferata et al. (2026) demonstrate LLM-based geolocation from crisis texts with im-

System	Approach	Input	Multi-lingual	Depl.	Geo-cond.	Joint target+bbox
GeoNorm	Candidate gen + rerank	Document text	✗	✗	✗	✗
Mordecai 3	Neural ranking	News text	Partial	✗	✗	✗
Hu et al. (2024)	Fine-tuned Llama2-13B	Document text	✗	✗	✗	✗
GDELT (2024)	Zero-shot GPT-3.5/4.0	Place name	Partial	✗	✗	✗
ReaGeo (2026)	Generative (LLM)	Address text	✗	✗	✗	✗
Zhang et al. (2020)	Intent + entity linking	Search query	✗	✗	✗	✗
Ours	**Generative (7B)**	**Query + geo**	✓	✓	✓	✓

Table 2: Comparison with prior work on location disambiguation.

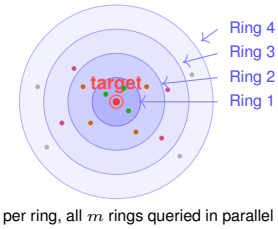


Figure 1: Expanding donut strategy.

proved fairness for underrepresented regions.

2.3 Spatial Query Understanding

Zhang et al. (Zhang et al., 2020) propose joint intent detection and entity linking on spatial domain queries, the closest prior work in the search domain. However, it uses entity *linking* to a fixed KB rather than generative disambiguation, and does not condition on user geo-coordinates. **GeoLM** (Li et al., 2023) incorporates geographic knowledge into pre-training for geospatially grounded understanding but targets general grounding rather than search disambiguation. **GER-LLM** (Zhu et al., 2025) uses LLMs for geospatial entity resolution (matching duplicate records), a different task from disambiguating user intent.

Table 2 summarizes the key differences with prior work.

3 System Architecture

3.1 Overview

OmniQUM operates as the first stage in the local search pipeline, determining WHERE to search before downstream retrieval. End-to-end flow is:

1. User issues a search query with device geo-coordinates
2. OmniQUM generates a disambiguated target and a bounding box decision (bounded vs. unbounded retrieval)
3. The target is geocoded to obtain precise coordinates
4. POI retrieval executes using the expanding donut strategy around the geocoded target (or user coordinates when target is null)

3.2 Model Input and Output

OmniQUM takes a single string combining the user query and geo-coordinates (rounded to 2 decimal places to balance precision with generalization within privacy constraints). Geo-coordinates come from standard device location services, which users consent to when enabling local search. Since disambiguation operates at city/neighborhood granularity, positioning errors of several hundred meters have negligible impact, and last-known coordinates suffice for poor-signal scenarios. When geo-coordinates are uninformative, the model relies on semantic context (e.g., "ski rental in Aspen" from New York resolves to Aspen, CO). The output is a JSON object with target and bounding box fields.

The **target** field is a fully disambiguated location (name, city, region, country), making the output directly geocodable. Each target is tagged `#POI` (e.g., landmarks, parks) or `#Location` (e.g., cities, streets); the tag serves as a fallback when geocoding fails, directing the system to search the POI index. The target is null (user's geo) when no location reference is present (e.g., "pharmacy near me").

The **bounding box** field determines retrieval scope. Yes indicates a category search (restaurants, hotels), retrieving within a bounded area. No indicates a specific entity (Starbucks, Hilton), requiring full index search since it may not exist near the target (e.g., Lidl in California).

3.3 Geocoding and Retrieval

We use a commercial geocoding service to map each predicted target to a coordinate c. Retrieval then queries a fixed schedule of concentric annuli around c. Let $0 = \rho_0 < \rho_1 < \cdots < \rho_m$ be the ring boundaries and let $0 < \delta < \rho_1$ be a small overlap. Writing $\ell(v)$ for the geocoded location of an indexed candidate v and d for geodesic distance, ring i is the candidate set

$$R_i = \{\, v \;:\; a_i \le d\big(\ell(v),\, c\big) \le \rho_i \,\},$$

for $i = 1, \ldots, m$, where $a_1 = 0$ and $a_i = \rho_{i-1} - \delta$ for $i > 1$. The intervals $[a_i, \rho_i]$ cover $[0, \rho_m]$; the overlap δ prevents candidates near a boundary from being missed due to geocoding jitter. Within each ring we rank candidates by lexical and semantic similarity and keep the top n, giving

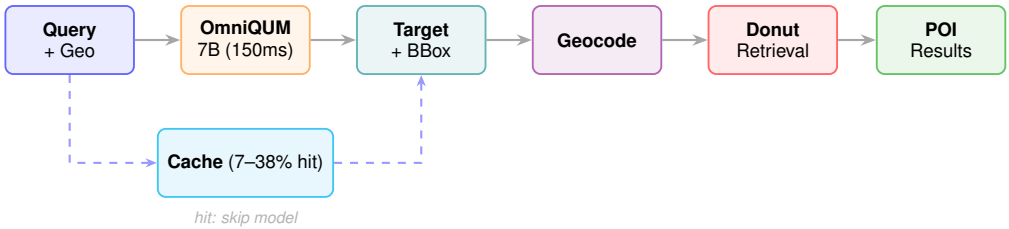


Figure 2: End-to-end architecture. On cache miss, OmniQUM produces a disambiguated target and bounding box; the target is geocoded, then donut retrieval finds POIs from successive distance rings.

$C_i \subseteq R_i$ with $|C_i| \leq n$. All m rings are queried in parallel, and the retrieved set is $C = \bigcup_{i=1}^{m} C_i$, hence $|C| \leq m\,n$ (Figure 1).

Concretely, we use m=8 rings with δ=0.1mi and n=25, for at most $m\,n$=200 candidates per query; the outer radii ρ_i are 1, 3, 6, 10, 20, 50, 100, and 200 miles. Because ring width grows with distance, candidates near the target are sampled densely while distant ones remain reachable; this is what improves recall over drawing the same number of candidates from a single flat bounding box. Recall is primarily sensitive to n, and in preliminary tuning we observed diminishing returns beyond n=25. We select this configuration empirically to maximize recall within our latency budget.

4 Agentic Data Pipeline

We develop a multi-agent pipeline that iterates from data generation through model evaluation. Five agents coordinate the pipeline: a Planner that identifies coverage gaps based on a coverage design (Section 4.2), a Generator that produces natural queries, a Critic that validates quality, an FSA (Failure Space Analysis) agent that analyzes prediction failures and identifies weak scenario categories, and an independent bigger LLM-based Annotator for evaluation labeling. The pipeline is LLM-agnostic and schema-driven, with agents reading YAML configuration, enabling expansion to new locales or output fields without code changes.

4.1 Pipeline Flow

We bootstrapped agentic flow with approximately 4,500 queries for en_US, annotated by human annotator. FSA on the first model trained on a subset of this dataset revealed 15 recurring failure patterns (Table 3). From this point, the pipeline iterates (Figure 3):

1. **Plan**: The Planner identifies underrepresented cells in the coverage matrix (categories $\times$ locales $\times$ target types) and produces a generation plan (see Section 4.2).
2. **Generate**: The Generator produces natural queries grounded in a per-locale geographic reference, maintaining persistent context across batches to avoid repetition.
3. **Validate**: The Critic enforces quality gates: zero overlap with existing data, format compliance, and bounding box balance (55–65% Yes).
4. **Train & Evaluate**: Deterministic stages handle transformation, training, and inference. Evaluation uses two-stage matching: exact string match, then LLM-judged semantic equivalence for non-matching pairs (e.g., "Manhattan, New York, NY, USA#Location" vs. "Manhattan, NYC, NY, USA#Location").
5. **Analyze**: The FSA agent categorizes each prediction error by query structure and error type (e.g., brand-as-location, missing city reference, geo-disambiguation failure), quantifies underrepresented categories, and produces a prioritized gap report feeding back to the Planner.

Human involvement is limited to approving generation plans and validating the correctness of generated samples. The deployment evaluation set (52K queries, 14 locales) is independent, annotated separately by the bigger LLM-based annotator and never seen during development.

4.2 Coverage Design

The planner generates training data structures that cover:

Scenario categories. Table 3 shows 15 query types from FSA analysis, ranging from simple category searches (D) to complex multi-reference queries (L) and code-switching (I), covering the full spectrum of disambiguation challenges.

Target types. Each locale's plan enforces proportional representation across target types: cities, neighborhoods, streets, POIs, addresses, zip codes, states/regions, countries, and null (no target). This prevents the model from biasing toward any single granularity.

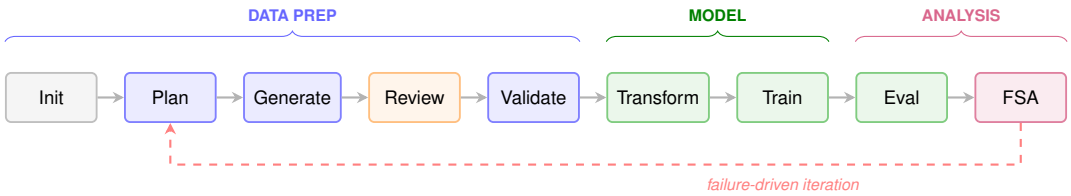


Figure 3: Data pipeline across three phases. Blue = LLM agents. Green = deterministic code. Orange = human approval. Dashed loop: FSA drives next iteration's planning.

Cat	Scenario	Example
A	Direct Subject vs Reference	"restaurants near Eiffel Tower" vs "Eiffel Tower hours"
B	Chain + City Disambiguation	"Hilton in der Innenstadt München"
C	Cross-Country Ambiguous Names	"hoteles en Córdoba" (ES vs AR)
D	Generic Category + City	"pharmacies in Brooklyn"
E	BBox Edge Cases	"food near Walmart" (yes) vs "Walmart hours" (no)
F	Event Queries	"Carnaval Salvador"
G	No Location / Informational	"does Costco accept EBT?"
H	Directions / Navigation	"comment aller à Montmartre"
I	Code-Switching / Colloquial	"find me Apotheke near Hauptbahnhof"
J	Generic Category, No City	"padarias perto de mim"
K	POI Geo-Disambiguation	"Universal Studios" (Orlando vs LA)
L	Complex Multi-Reference	"coffee on Bath Road near Heathrow"
M	Common-Name Geo-Association	"ristoranti in Via Roma"
N	Place-Name-in-Entity	"Boston Market near me"
O	Natural Variations	"resturants times sq"

Table 3: 15 scenarios for targeted data generation.

Hard negatives and ambiguous references. The Planner builds a geographic reference per locale that includes:

- Brands containing place names (Boston Market, California Pizza Kitchen, Paris Baguette) to teach null-target recognition
- Cross-country homonyms (Portland in OR/ME, Córdoba in Spain/Argentina, London in UK/Ontario) to teach geo-conditioned disambiguation
- Common street names present in hundreds of cities (High Street, Via Roma, Hauptstraße) to teach geo-association

Disambiguation samples are generated contrastively: the same query with different geo-coordinates must resolve to different targets (e.g., "restaurants on High Street" resolves to Leeds or Oxford depending on user geo). This explicitly teaches the model to attend to coordinates rather than defaulting to the most popular location.

4.3 Dataset Construction

The training dataset evolved through four versions (Table 4), each driven by gaps identified in the previous iteration by the FSA agent. The final dataset (V4) comprises 21,674 samples across 14 locales with balanced representation across target types (full distribution in Appendix A).

Ver	Samples	Change
V1	4,500	Bootstrap (en_US)
V2	7,741	+10 locales
V3	14,674	+4 locales
V4	21,674	+7K (9 patterns)

Table 4: Dataset evolution via FSA.

Metric	Mistral	Qwen3
Precision	**.968**	.952
Recall	.994	**.996**
F1	**.981**	.975
TM95 (ms)	**285**	363

Table 5: Model selection.

The schema defines extraction fields, target types, scenario categories, domain distribution, and per-locale geographic references. We validated extensibility by scaling from en_US to 14 locales and adding temporal features for weather queries, both without code changes.

5 Experiments

We present our experimental setup, followed by disambiguation results, retrieval impact, and deployment metrics.

5.1 Setup

Model Selection To select the base model, we trained Mistral 7B and Qwen 3 4B on our V2 dataset (7.7K samples) and compared disambiguation performance on 3K queries per locale (Table 5). We selected these candidates for their multilingual support, permissive licensing, and latency budget. Mistral achieves higher precision and F1 while delivering 22% lower latency (TGI, 4×A10G, TPS=1), which we attribute in part to faster decoding from its smaller vocabulary (32K tokens vs. Qwen's 150K+). This aligns with Hu et al. (2024), who find that fine-tuned Mistral 7B outperforms all other lightweight models (7B–13B) on toponym resolution. We select Mistral 7B v0.3 as the base model for all subsequent iterations.

Fine-Tuning We apply Low-Rank Adaptation (LoRA) (Hu et al., 2022) with rank r=8 and α=16 to all attention projections (`q/k/v/o_proj`), reducing trainable parameters by over 99%, en-

Locale	N	NER+Gazetteer				Zero-shot Mistral 7B				Zero-shot Llama 70B				Few-shot Llama 70B				OmniQUM 7B (Ours)			
		P	R	F1	BB	P	R	F1	BB	P	R	F1	BB	P	R	F1	BB	P	R	F1	BB
en_US	9385	.940	.805	.867	–	.498	1.0	.664	.475	.771	.985	.865	.793	.928	.993	.959	.910	.957	.984	.970	.950
en_GB	7958	.931	.696	.797	–	.454	1.0	.624	.514	.763	.974	.856	.798	.902	.986	.942	.913	.981	.967	.974	.934
en_CA	7283	.938	.778	.851	–	.419	1.0	.590	.471	.772	.987	.866	.743	.909	.992	.949	.876	.982	.979	.980	.955
en_IE	948	.922	.719	.808	–	.376	.994	.545	.442	.697	.990	.818	.759	.887	.989	.935	.921	.956	.973	.964	.959
de_DE	3581	.810	.708	.756	–	.337	.999	.504	.504	.719	.972	.827	.779	.846	.980	.908	.896	.938	.963	.950	.890
de_AT	773	.769	.727	.747	–	.317	1.0	.482	.433	.709	.987	.825	.715	.845	.992	.913	.907	.915	.976	.944	.930
fr_FR	4545	.781	.649	.709	–	.377	.999	.547	.438	.582	.963	.725	.775	.829	.971	.895	.878	.877	.953	.913	.873
fr_CA	883	.770	.672	.718	–	.315	1.0	.479	.447	.653	.972	.781	.747	.833	.983	.901	.852	.902	.956	.928	.856
es_ES	4516	.760	.695	.726	–	.356	1.0	.525	.381	.659	.946	.777	.727	.835	.963	.894	.886	.863	.971	.914	.888
es_MX	3805	.747	.654	.697	–	.327	1.0	.493	.360	.651	.955	.774	.692	.822	.972	.891	.874	.813	.981	.889	.901
es_US	393	.816	.870	.842	–	.218	1.0	.358	.418	.868	.958	.911	.527	.982	.982	.982	.927	.982	1.0	.991	.927
it_IT	3833	.702	.648	.674	–	.394	1.0	.565	.398	.653	.946	.773	.734	.833	.961	.892	.876	.833	.961	.893	.872
pt_BR	4459	.756	.630	.688	–	.408	.999	.580	.457	.707	.953	.812	.704	.844	.958	.897	.848	.827	.963	.890	.858
nl_NL	74	.800	.800	.800	–	.167	1.0	.286	.833	.833	1.0	.909	.833	.833	1.0	.909	.833	1.0	1.0	1.0	.500
Avg	**52436**	.817	.718	.763	–	.354	.999	.517	.469	.717	.970	.823	.738	.866	.980	.919	.885	**.916**	**.973**	**.943**	**.878**

Table 6: Per-locale disambiguation results. N = queries per locale. All LLMs include geo-coordinates in prompt. BB = bounding box accuracy (NER+Gazetteer does not predict BB). nl_NL (N=74) reflects limited traffic; post-hoc evaluation (N=1,151) yields F1=0.920, BB=0.837.

Locale	Flat	+Donut	+Tgt	+Tgt+BB	Incremental Gain (%)			
					Δ Donut	Δ Tgt	Δ BB	Δ Total
en_US	.724	.771	.907	.962	+6.5	+17.6	+6.1	+32.9
en_GB	.730	.789	.870	.926	+8.1	+10.3	+6.4	+26.8
en_CA	.671	.781	.814	.903	+16.4	+4.2	+10.9	+34.6
en_IE	.748	.769	.789	.856	+2.8	+2.6	+8.5	+14.4
de_DE	.690	.724	.883	.909	+4.9	+22.0	+2.9	+31.7
de_AT	.790	.812	.919	.934	+2.8	+13.2	+1.6	+18.2
fr_FR	.567	.658	.817	.861	+16.0	+24.2	+5.4	+51.9
fr_CA	.713	.764	.884	.917	+7.2	+15.7	+3.7	+28.6
es_ES	.560	.652	.825	.894	+16.4	+26.5	+8.4	+59.6
es_MX	.655	.663	.793	.827	+1.2	+19.6	+4.3	+26.3
es_US	.667	.676	.904	.916	+1.3	+33.7	+1.3	+37.3
it_IT	.614	.656	.910	.943	+6.8	+38.7	+3.6	+53.6
pt_BR	.626	.659	.860	.899	+5.3	+30.5	+4.5	+43.6
nl_NL	.724	.762	.819	.836	+5.2	+7.5	+2.1	+15.5
Avg	.677	.724	.857	**.899**	**+6.9**	**+18.4**	**+4.9**	**+32.7**

Table 7: Macro recall and incremental gain per component. Δ Donut = gain over Flat; Δ Tgt = gain over +Donut; Δ BB = gain over +Tgt; Δ Total = gain over Flat. Gains in the **Avg** row are computed from the averaged recalls, not by averaging the per-locale gains (the latter yields $+33.9\%$ total).

abling single-GPU training via gradient checkpointing and bfloat16. The model is trained on 16,969 training samples with early stopping on 4,705 validation samples (patience of 2). Each sample follows the Mistral instruction template with a ∼130-token system prompt (selected from a 35–200 token range as the best quality-latency tradeoff); output is a JSON object of 21–28 tokens (Appendix B), where JSON adds only ∼4–5 structural tokens over a minimal delimited format, chosen for parseability and robustness. Full hyperparameters are reported in Appendix C.

Post-training, we merge the learned adapter weights into the base model to produce a single dense artifact, enabling FP8 quantization and tensor parallelism at inference.

5.2 Location Disambiguation Results

We evaluate on 52,436 deployment-representative queries across 14 locales (per-locale counts in Table 6), covering all 15 scenario categories from Table 3. Ground truth is generated by a bigger LLM Annotator independently from training. Predictions are compared via two-stage matching: exact string match, then LLM-judged semantic equivalence for non-matching pairs.

Instance	Config	TM95	TPS/inst	50 TPS cost	FP8 match
4×A10G (96GB)	TGI bf16	321ms	4	1x	–
4×A10G (96GB)	vLLM FP8	239ms	5	0.77x	98%
1×L40S (48GB)	vLLM FP8	320ms	8	0.22x	98%
2×RTX PRO 6000 (192GB)	**vLLM bf16**	**135ms**	**120**	**0.12x**	–
2×RTX PRO 6000 (192GB)	vLLM FP8	90ms	120	0.12x	90%
4×RTX PRO 6000 (384GB)	vLLM FP8	82ms	120	0.23x	90%

Table 8: Serving configurations: TM95 at TPS=2. TPS/inst = max throughput tested per instance <400ms. FP8 match = exact match vs. bf16.

Table 6 compares OmniQUM against four baselines: (1) NER+Gazetteer using SpaCy NER (Honnibal et al., 2020) and GeoNames (GeoNames, 2024) with geo-proximity ranking, analogous to Mordecai 3 (Halterman, 2023), F1=0.763; (2) zero-shot Mistral 7B, the same base model without fine-tuning, F1=0.517; (3) zero-shot Llama 3.1 70B, F1=0.823; and (4) few-shot Llama 3.1 70B, F1=0.919. OmniQUM (F1=0.943) outperforms all baselines, including the 10× larger few-shot Llama, using a ∼130-token prompt (∼20× shorter) at sub-150ms latency. We find that NER+Gazetteer (F1=0.763) suffers from low recall on short, lowercase queries (e.g., "restaurants fuenlabrada" missed by NER) and produces false positive extractions ("Manhattan Bagel" → Manhattan). Zero-shot Llama 70B (F1=0.823) over-predicts targets when no location exists and often fails geo-disambiguation ("East Village bars" in London predicts NYC). Few-shot Llama 70B (F1=0.919) improves but still over-extracts, consistent with prior findings that prompted LLMs are unreliable geocoders (GDELT Project, 2024). These findings suggest that task-specific fine-tuning is more effective than model size alone, enabling deployment at a fraction of the inference cost.

While reported F1 is 0.94, error analysis shows only 3.6% of predictions are genuinely harmful to downstream retrieval. Most false positives are retrieval-benign, producing valid targets that aid retrieval despite differing from ground truth (e.g., "Santa Clara Library" as a library in Santa Clara in prediction vs. a named POI in ground truth). The dominant harmful pattern is wrong geographic expansion (2% of queries), concentrated in Romance locales with shared place names.

Human Validation. On 4,051 human-reviewed queries, the LLM annotator agrees with human judgment in 93.6% of cases, and OmniQUM achieves F1=0.925 vs. human labels (0.948 vs. LLM labels), indicating minimal circularity impact. The semantic-equivalence judge is needed for only 31.6% of predictions (68.4% resolve by exact string match), with 95.6% human-judge agreement confirmed on 2,000 sampled cases.

5.3 Retrieval Impact

Between model prediction and retrieval, targets are geocoded via a geocoding service: across 10,938 unique targets in 14 locales, 74.2% resolve within 5km (fine-grained: addresses, POIs, streets) and 93.7% within 100km (cities, states, regions). The retrieval gold set is built from web search API results, filtered by proximity and semantic relevance, then matched to POIs in our index (Maritan et al., 2026), following automated ground truth construction methodology (Kambhatla et al., 2026). Table 7 shows the cumulative impact on retrieval recall as each component is added. Each gain is relative to the previous stage, compounding to +32.7% total over the flat baseline. This translates to an estimated 86K additional relevant POIs surfaced across the evaluation set.

5.4 Deployment

OmniQUM is deployed serving 14 locales across 7 languages. To identify the optimal serving configuration, we conducted a systematic ablation over GPU instances, inference frameworks, and numerical precision (Table 8).

Based on this ablation, we deploy with vLLM + bf16 + TP (tensor parallelism) =2 on 2×RTX PRO 6000, achieving 150ms TM95 at 50 TPS (135ms at TPS=2, Table 8) at 0.12× baseline cost. This configuration provides a balanced optimization across accuracy, latency, and cost. We select bf16 over FP8 because quantization fidelity varies by GPU architecture (98% exact match on A10G/L40S vs. 90% on RTX PRO 6000); quantization-aware training to close this gap is ongoing work.

Cost varies due to latency scaling: e.g., FP8 A10G instances cliff sharply (239→409ms, TPS 2→8) from KV cache exhaustion, requiring many instances, while RTX PRO 6000 maintains flat curves (90→105ms FP8, TPS 2→120; 135→150ms bf16, TPS 2→50) via higher memory bandwidth, enabling fewer instances. vLLM's prefix caching further reduces per-query cost by reusing KV computations for the shared system prompt.

Caching. We implement a distributed in-memory cache keyed on stop-word-filtered query text and geo-coordinates. Since model outputs are independent of index updates, invalidation is needed only on model schema changes. We observe 7–38% hit rates, yielding 60–100+ms latency savings with <5ms overhead.

6 Conclusion

We presented OmniQUM, a generative approach to location disambiguation that conditions on geo-coordinates to jointly predict search targets and retrieval strategy across 14 locales. Combined with expanding donut retrieval, the system achieves F1 of 0.94 and 32.7% recall improvement over baseline, deployed at sub-150ms latency. The failure-driven agentic pipeline enables iterative expansion without code changes. The "Omni" in OmniQUM reflects our vision of universal locale coverage; since submission, the system has expanded to 21 locales across 10 languages, including non-Latin scripts (Arabic, Hindi, Japanese), using the same pipeline without architectural changes. Future work includes expanding to more locales and domain-adaptive pre-training on geographic corpora to strengthen coordinate-to-region mappings, particularly for long-tail geographic entities. Complementary directions include distance-aware ranking metrics for multi-answer retrieval (Zhou et al., 2026) and geo-semantic cross-encoder reranking of retrieved POIs (Bai et al., 2026).

7 Limitations

We identify the following limitations of our current approach. First, geo-coordinates are treated as text tokens with no explicit spatial encoding.

The model cannot compute distances or reason geometrically; it relies on patterns learned from training data to associate coordinate ranges with regions. This is consistent with findings from Truong et al. (2026), who show that LLMs struggle with geometric GPS computations while performing better on real-world geographic reasoning. This causes occasional errors near city borders where a small coordinate difference maps to a different city. Second, most of the evaluation ground truth is LLM-generated, introducing potential circularity. While our error analysis shows most mismatches are retrieval-benign, a fully human-annotated evaluation set would strengthen confidence. Third, the model's geographic knowledge is bounded by its training data. Rare cities, newly named neighborhoods, or places not represented in training may be incorrectly disambiguated. The failure-driven pipeline mitigates this through iterative gap-filling, but coverage is never complete.

References

Zhengwei Bai, Moreno D'Incà, Alessandro Moschitti, and Danielle Class. 2026. H2ce: Modeling geo-semantic interactions for POI reranking with heterogeneous two-stage cross-encoders. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management*, Rome, Italy.

G. Cafferata, T. Demarco, K. Kalimeri, Y. Mejova, and M. G. Beiró. 2026. Large language models for geolocation extraction in humanitarian crisis response. *arXiv preprint arXiv:2602.08872*.

Jian Cui, Zhiyuan Ren, Desheng Weng, Yongqi Zhao, Wenbin Gong, Yu Lei, and Zhenning Dong. 2026. Reageo: Reasoning-enhanced end-to-end geocoding with llms. *arXiv preprint arXiv:2604.21357*.

GDELT Project. 2024. The surprisingly poor performance of llm-based geocoders, geographic bias & why gpt 3.5 & gemini pro outperform gpt 4.0 in underrepresented geographies.

GeoNames. 2024. Geonames geographical database.

Andrew Halterman. 2023. Mordecai 3: A neural geoparser and event geocoder. *arXiv preprint arXiv:2303.13675*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Xuke Hu, Jens Kersten, Friederike Klan, and Sheikh Mastura Farzana. 2024. Toponym resolution leveraging lightweight and open-source large language models and geo-knowledge. *International Journal of Geographical Information Science*, pages 1–28.

Gauri Kambhatla, Xiaokun Zhou, and Alessandro Moschitti. 2026. Automatic construction of ranked ground truth for POI retrieval evaluation using large language models. *arXiv preprint*.

Zekun Li, Wenxuan Zhou, Yao-Yi Chiang, and Muhao Chen. 2023. Geolm: Empowering language models for geospatially grounded language understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5227–5240, Singapore. Association for Computational Linguistics.

Nicola Maritan, Alessandro Moschitti, Federico Borazio, Xiaokun Zhou, and Zhengwei Bai. 2026. POISS: A large-scale multilingual dataset for point-of-interest search. In *Proceedings of EMNLP*.

Tessa Masis and Brendan O'Connor. 2026. Coordinates from context: Using llms to ground complex location references. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026)*, pages 1591–1606.

Thinh Hung Truong, Jey Han Lau, and Jianzhong Qi. 2026. Gpsbench: Do large language models understand gps coordinates? *arXiv preprint arXiv:2602.16105*.

Lei Zhang, Runze Wang, Jingbo Zhou, Jingsong Yu, Zhenhua Ling, and Hui Xiong. 2020. Joint intent detection and entity linking on spatial domain queries. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4937–4947.

Zeyu Zhang and Steven Bethard. 2023. Improving toponym resolution with better candidate generation, transformer-based reranking, and two-stage resolution. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 48–60.

Zeyu Zhang and Steven Bethard. 2025. A survey on geocoding: algorithms and datasets for toponym resolution. *Language Resources and Evaluation*, 59:1775–1796.

Xiaokun Zhou, Alessandro Moschitti, and Danielle Class. 2026. Rank-deviation quality: A distance-aware metric for multi-answer ranking. *arXiv preprint*.

Haojia Zhu, Zhicheng Li, and Jiahui Jin. 2025. Gerllm: Efficient and effective geospatial entity resolution with large language model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23261–23277.

A Dataset Distribution

Locale	City	Neigh.	POI	Street	Null	Other	Total
en_US	940	291	392	141	2,782	202	4,748
en_GB	172	321	156	117	568	23	1,357
en_CA	307	149	123	126	538	34	1,277
en_IE	352	1	125	72	484	166	1,200
de_DE	293	167	153	22	570	55	1,260
de_AT	204	81	162	35	503	215	1,200
fr_FR	332	103	147	115	541	84	1,322
fr_CA	267	111	109	113	439	8	1,047
es_ES	370	129	132	104	516	57	1,308
es_MX	380	117	142	114	519	37	1,309
it_IT	306	114	122	133	513	53	1,241
pt_BR	309	168	128	68	493	34	1,200
nl_NL	441	158	147	0	545	47	1,338
es_US	508	99	136	79	566	46	1,434
Total	**5,181**	**2,009**	**2,174**	**1,239**	**9,577**	**1,061**	**21,241**

Table 9: Dataset distribution by locale and target type (V4). Null = no location target. Other includes address, zip code, state/region, and country targets. Remaining 433 (21,674 − 21,241) samples are spread across locales from iterative gap-filling.

B Training Template

Format:

```
<s>[INST] {system_prompt}
{model_input} [/INST]
{model_output}</s>
```

C Training Hyperparameters

Parameter	Value
Base model	Mistral 7B v0.3
LoRA rank (r)	8
LoRA alpha (α)	16
LoRA dropout	0.05
Target modules	q/k/v/o_proj
Learning rate	1×10^{-4}
Batch size (per device)	1
Gradient accumulation steps	8
Effective batch size	8
Max epochs	6
Precision	bfloat16
Eval frequency	every 1,000 steps
Early stopping patience	2 evaluations
Metric for best model	validation loss
Max sequence length	4,096 tokens
Gradient checkpointing	enabled
Instance	1×A10G (24GB)

Table 10: Full training hyperparameters for LoRA fine-tuning.