# LentEx: Generalizable Latent Entity Extraction via Synthetic Data and Instruction-Tuned LLMs


1st Umesh Bodhwani
*Amazon*
Seattle, WA, USA
bodhwani@amazon.com

2nd Yuan Ling
*Amazon*
Seattle, WA, USA
yualing@amazon.com

3rd Cibi Chakravarthy Senthilkumar
*Amazon*
Seattle, WA, USA
sentocb@amazon.com

4th Shujing Dong
*Amazon*
Seattle, WA, USA
shujdong@amazon.com

5th Yarong Feng
*Amazon*
Seattle, WA, USA
yarongf@amazon.com

6th Hongfei Li
*Amazon*
Seattle, WA, USA
lihongfe@amazon.com

7th Ayush Goyal
*Amazon*
Seattle, WA, USA
ayushg@amazon.com



***Abstract*—Latent entity extraction (LEE) tackles the challenge of identifying implicit, contextually inferred entities within free text—an area where traditional entity extraction methods fall short. In this paper, we introduce LentEx, a novel framework for latent entity extraction that leverages synthetic data generation and instruction fine-tuning to optimize smaller, efficient large language models (LLMs). Latent entities, which are often abstract and thematic, are crucial for applications such as retrieval-augmented generation (RAG), customer persona analysis, and knowledge graph enrichment. LentEx addresses the scarcity of labeled datasets by employing a template-based approach to generate diverse, contextually rich synthetic data, ensuring high variability and alignment with real-world distributions. To our knowledge, LentEx is the first to systematically approach LEE through the lens of LLMs. LentEx demonstrates significant performance improvements across multiple tasks, notably surpassing state-of-the-art models on the MTEB Clustering Benchmark. Furthermore, our methodology enables robust generalization to unseen domains, making LentEx highly applicable in real-world NLP tasks, including RAG and clustering, thereby establishing a new paradigm for latent entity understanding and extraction in natural language processing.**



***Index Terms*—llm, retrieval augmented generation, entity extraction, fine-tuning, information retrieval, clustering, synthetic data generation**


## I. Introduction

Latent Entity Extraction (LEE) refers to identifying entities implicitly present in textual data, inferred from contextual nuances rather than explicitly mentioned. Unlike traditional Named Entity Recognition (NER), which focuses on overt entities such as names of organizations, locations, or persons, LEE aims to uncover abstract or thematic entities—such as the persona behind a conversation or an implied industry sector—deduced from context. For instance, while NER may quickly identify mentions of "Google" or "New York," LEE endeavors to infer broader, latent concepts such as "technology company" or "urban environment" based solely on contextual cues.

Prior research on LEE has predominantly relied on classification or topic modeling. Classification methods are inherently constrained by their reliance on predefined labels, often requiring extensive retraining as new categories emerge [1], [2]. Topic modeling, although adept at identifying general themes, frequently produces clusters of keywords that lack interpretability and fail to provide actionable insights regarding specific latent entities [3]–[6]. These approaches generate high-level thematic summaries but often lack the granularity to infer nuanced, context-specific information. In retrieval-augmented generation (RAG) systems, LEE enhances document filtering by extracting latent entities from queries and documents. For instance, a query like *What are the latest advancements in AI?* can be enriched with latent entities such as *deep learning techniques* or *natural language processing frameworks*, enabling the RAG system to retrieve more relevant documents.

With the advent of large language models (LLMs), natural language processing has seen unprecedented success across a range of tasks, owing to the impressive capability of LLMs to generate human language [7]–[13]. Despite these advancements, the application of LLMs for LEE remains largely unexplored.

This paper introduces LentEx, a novel, supervised learning framework specifically designed to address the challenges of LEE. LentEx leverages synthetic data generation and self-instruction to fine-tune smaller, efficient LLMs, thereby offering a scalable solution to LEE. By generating diverse training datasets that mirror the complexity of real-world text, LentEx establishes a robust foundation for extracting latent entities across multiple domains. These extracted entities serve as structured representations that improve text organization (clustering) and relevance filtering (retrieval-augmented generation). Our approach ensures generalization to unseen domains, making it highly applicable in real-world scenarios such as knowledge discovery, content personalization, and beyond. This work shifts the paradigm toward a deeper understanding of the implicit dimensions of text with the following key contributions:

1) Formalizing domain-agnostic latent entity extraction problem and developing a systematic approach to ad-

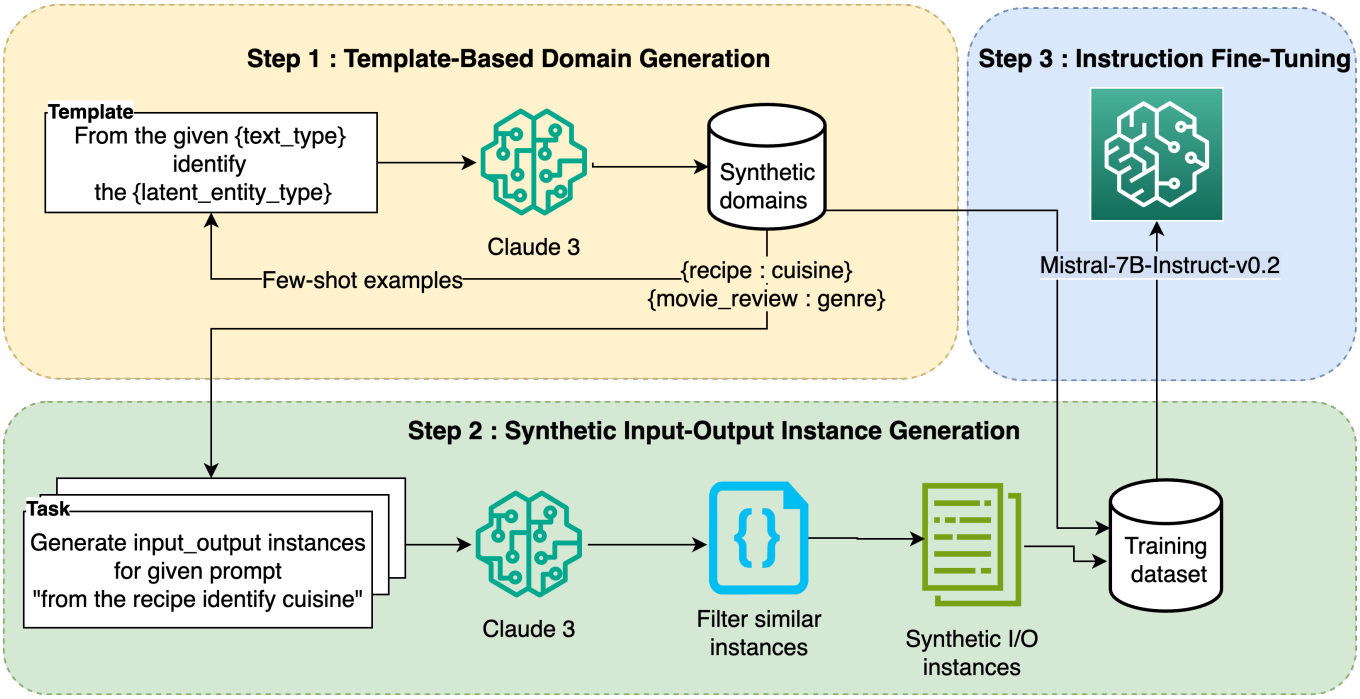


Fig. 1. Overview of the LentEx workflow in three steps: (1) Template-based domain generation using Claude-3 with few-shot examples, (2) Synthetic input-output instance generation with similarity filtering, and (3) Instruction fine-tuning using Mistral-7B-Instruct-v0.2

dress it.

2) Developing a novel, template-based synthetic data generation process, producing diverse datasets without manually curated seed data.
3) Demonstrating the effectiveness of fine-tuning a small LLM using this synthetic dataset, achieving superior performance over state-of-the-art embedding-based and generation-based methods on multiple downstream tasks.

## II. Related Work

### A. Latent Entity Extraction

Previous work in topic modeling has primarily centered around clustering textual data and identifying representative keywords to infer thematic patterns [3]–[6]. These models typically require selecting an embedding space, followed by post-processing steps to extract interpretable topics from keyword clusters. While useful for high-level theme identification, such approaches often produce clusters of keywords that lack the granularity and specificity needed for actionable entity extraction [1], [2]. Traditional entity recognition and other information extraction tasks have relied heavily on supervised learning frameworks that map predefined labels to overtly mentioned entities [14], thus limiting their adaptability to new, unseen categories.

Our approach overcomes these limitations by enabling the open-ended extraction of latent entities without reliance on predefined labels, advancing the field of entity extraction. By leveraging a more flexible framework, our method facilitates dynamic entity prediction even for out-of-distribution data, addressing the evolving nature of real-world datasets. This represents a leap forward in entity extraction, as it allows for adapting to new entity types and shifting data landscapes without retraining.

### B. Synthetic Data Generation

The high cost of curating and labeling data has long posed a challenge for training robust machine learning models [15]. Recent advancements in synthetic data generation offer a promising solution, enabling the creation of large-scale, diverse training datasets at a fraction of the cost of manual annotation. Self-Instruct paradigm [16] demonstrates the power of prompting large language models to self-generate instructions, inputs, and outputs, resulting in labeled data points from limited seed data in few-shot setting. SeqGPT [17] collects data from multiple sources and leverages ChatGPT for data augmentation and label generation, while InstructPTS [18] applies instruction-tuning techniques to generate product titles for retail domains. To this end, our approach introduces a novel, template-based synthetic data generation process that produces diverse, contextually rich datasets without manually curated seed data. This method lowers the barrier to developing high-quality training data for latent entity extraction and contributes to the growing body of work focused on efficient, scalable synthetic data generation for NLP tasks.

## III. LentEx

### A. Problem Definition

Latent Entity Extraction (LEE) involves identifying entities that are implied within the context of the text but are not


Fig. 2. Diversity of synthetic data. Inner circle represents the 100 synthetic domains in synthetic data. Outer circle represents the domains clustered based on their similarity

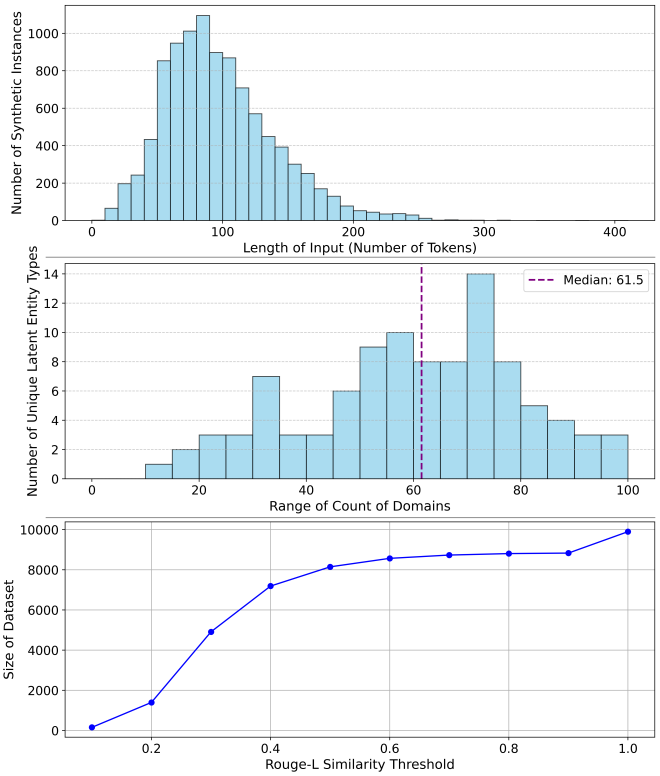


Fig. 3. Synthetic data distributions: Top figure shows the distribution of length of input tokens across instances, middle figure shows the distribution of range of count of domains and the corresponding number of unique entities, while the bottom figure shows the size of dataset at different values of Rouge-L

explicitly mentioned. Defining formally:

$$\mathrm{E} = \mathcal{F}(\mathrm{I}, \mathrm{T}_{\text{entity}}, \mathrm{T}_{\text{text}}, [\mathbf{O}]) \tag{1}$$

where $\mathcal{F}$ represents the function, i.e. the fine-tuned LLM, I is the input text, E denotes the identified latent entity ($\mathrm{E} \notin \mathrm{I}$), $\mathrm{T}_{\text{entity}}$ denotes the latent entity type (e.g., persona, industry), $\mathrm{T}_{\text{text}}$ refers to the text type (e.g., narrative, description), and $[\mathbf{O}]$ is an optional set of predefined labels for supervised learning scenarios. This formulation enables LEE to function flexibly, addressing both unsupervised settings (e.g., topic modeling, clustering) and supervised tasks (e.g., classification).

### B. Method Overview

LentEx addresses the challenge of labeled data scarcity through a novel methodology combining synthetic data generation and instruction fine-tuning, as illustrated in Figure 1. This approach enables effective latent entity extraction across multiple domains while using efficient, smaller LLMs.

### C. Synthetic Data Generation

*1) Template-Based Domain Generation:* Instead of manually curating seed dataset [16], we adopt a template-based approach for synthetic data generation. We start with the prompt - "From the given {text_type}, identify the {latent_entity_type}," as shown in the prompts in Table I Using a few-shot learning setup, we leverage Claude 3 Sonnet [7] to iteratively generate 100 unique domain combinations of text_type, latent_entity_type, such as {*sport_commentary: sport_name*}. This ensures a wide coverage of domain diversity. New combinations are generated at each iteration by selecting a random subset of previously created domains, and duplicates are filtered out. Figure 2 illustrates the distribution of latent entity types clustered into broader parent domains.

*2) Synthetic Input-Output Instance Generation:* For each domain, we create synthetic input-output pairs using Claude 3 Sonnet. Each instance comprises a synthetically generated text and its corresponding latent entity label inferred from the context. This process is repeated over 10 iterations per domain, generating a total of 10000 diverse synthetic samples. The diversity of these samples is validated using Rouge-L similarity scores, with over 80% of instances exhibiting scores below 0.5, underscoring their variability. Figure 3 shows the distribution of input token lengths, domain-entity mappings, and dataset sizes at different Rouge-L thresholds.

### D. Instruction Fine-Tuning

*1) Training Data Construction:* We construct training data by concatenating instruction templates, domain descriptions, latent entity types, and input texts to form coherent model inputs, with the inferred latent entity as the target label. To ensure diversity and avoid explicit mentions of entities in the input text, we filter out instances with Rouge-L similarity scores above 0.5, forcing the model to rely on contextual inference. We mitigate over-fitting by employing paraphrased instruction templates (shown in Table I, introducing variation

TABLE I
PROMPTS FOR MULTIPLE STAGES. FIRST ROW PROMPT IS USED IN STEP 1, SECOND ROW IN STEP 2 OF LENTEX WORKFLOW. THIRD ROW SHOWS THE PARAPHRASED PROMPTS USED IN STEP 3 FOR INSTRUCTION FINE-TUNING THE MODEL

| Task | Prompt |
|---|---|
| Domain Generation | Generate a synthetic data dictionary containing 10 example entries. For each entry, the input should describe a broad category of text, and the output should specify a general category of latent entities. Latent entities are those that can be inferred from a text, even though they are not explicitly mentioned in the text. Ensure that the examples span a diverse range of topics, including but not limited to entertainment, education, technology, and health. The output categories should be broad, general, in 1-3 words, like 'movie genre' or 'product type', indicative of the kind of inference a reader can make from the given text type. Example: seed_task_dict = { "sports commentary": "sport_name", "Google place review": "place category", "book summary": "book genre", "legal case_summary": "legal case type" } |
| Input-output Instance Generation | Generate a list of 10 detailed instances, each with their corresponding type or category. For each instance, provide comprehensive content that is representative of its category, ensuring clarity, relevance, and authenticity. The instances should cover a diverse range of subjects, showcasing unique and specific characteristics of each category. Format the output as follows: Each instance should be encapsulated in a dictionary with two keys: "text_type" and "latent_entity_type". The "text_type" key should include the full content of the instance, detailed and structured appropriately for its type. The "latent_entity_type" key should specify the category or type that the instance belongs to. Structure each dictionary entry as a separate line in a JSONL format, where each line represents a single, complete dictionary corresponding to one instance and its category. Encapsulate the output in tag ¡rows¿. |
| Paraphrased Instructions for Training and Inference | 1) From the given {text_type}, determine the {latent_entity_type}.<br>2) Based on the provided {text_type}, identify the {latent_entity_type}.<br>3) From the specified {text_type}, ascertain the {latent_entity_type}.<br>4) Using the given {text_type}, pinpoint the {latent_entity_type}.<br>5) Examine the {text_type} provided and determine the {latent_entity_type}. |

TABLE II
EXPERIMENT RESULTS FROM AUTOMATED EVALUATION OF MODEL PREDICTIONS ON LATENT ENTITY EXTRACTION TASK

| Model | Mean Semantic Similarity % |
|---|---|
| Mistral-7B-Instruct-v0.2 | 61.17 |
| Claude-3-Haiku | 66.48 |
| LentEx (Ours) | 77.79 |

in sequence structures, and applying subtle modifications in spacing and line breaks [19].

*2) Model Selection and Training:* We fine-tune Mistral-7B-Instruct-v0.2 [20] using Low-Rank Adaptation (LoRA) [21] technique for efficient training. The model is fine-tuned on four NVIDIA A10G GPUs for one epoch, with a batch size of $8$, a warmup ratio of $0.01$, and a learning rate of $0.0001$. LoRA parameters include rank $16$ and lora_alpha $32$. Mistral-7B-Instruct was selected for its superior performance among models of similar size, and LoRA was chosen for its computational efficiency. Our methodology can be extended to other models and training setups.

## IV. EXPERIMENTATION

We perform a series of experiments to evaluate the effectiveness of LentEx on latent entity extraction as well as its downstream applications in Clustering and Retrieval-Augmented Generation (RAG).

### A. *Latent Entity Extraction Performance*

In this experiment, we evaluate LentEx's ability to extract latent entities accurately. We evaluate the LentEx framework by comparing predicted latent entities against reference labels generated from synthetic data. We compute the semantic similarity between model predictions and synthetic labels using the all-mpnet-base-v2 [22] embedding model. Mean semantic similarity is calculated for all instances to quantify the alignment between predicted and reference entities. We compare LentEx against two baselines: Mistral-7B-Instruct-v0.2 [20] and Claude-3-Haiku [7]. We randomly sample $50$ synthetic instances from each domain, totaling $5000$ instances for evaluation.

**Result:** As shown in Table II, LentEx achieves higher semantic similarity with the reference labels, compared to Mistral-7B-Instruct-v0.2 and Claude-3-Haiku, establishing the baseline effectiveness of LentEx as an entity extraction model.

### B. *Clustering with Latent Entity Representations*

To demonstrate the effectiveness of LentEx in real-world scenarios, we apply LentEx to clustering tasks using $11$ P2P (paragraph input) and S2S (sentence input) datasets from the Multilingual Textual Entailment Benchmark (MTEB) [23]. These datasets span diverse domains, including arXiv, bioRxiv, medRxiv, Reddit, StackExchange, and 20-Newsgroups. We benchmark LentEx against state-of-the-art embedding-based models: gte-Qwen2-7B-instruct [24], bge-en-icl [25], NV-Embed-v2 [26] (as of Oct 2024), as well as prompting based methods: Mistral-7B-Instruct-v0.2 [20] and Claude-3-Haiku [7].

For all generation methods including ours, we conduct zero-shot inference to infer the latent entity. We embed the latent entities, followed by a mini-batch k-means model with batch size $500$ and $k$ equal to the number of different labels [27]. We choose the metric in accordance with MTEB, i.e., v-measure [28] for Clustering, which is computed as the harmonic mean

TABLE III
CLUSTERING PERFORMANCE (V-MEASURE SCORES) ACROSS DIFFERENT DOMAINS. P2P AND S2S DENOTE PARAGRAPH-TO-PARAGRAPH AND SENTENCE-TO-SENTENCE TASKS RESPECTIVELY. BEST SCORES ARE IN BOLD, SECOND-BEST ARE UNDERLINED

| **Inference without entity options (Clustering Task)** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **Average** | **arxiv-*** | | **biorxiv-*** | | **medrxiv-*** | | **reddit-*** | | **stackexchange-*** | | **twenty-newsgroups** |
| | | **p2p** | **s2s** | **p2p** | **s2s** | **p2p** | **s2s** | | **p2p** | | **p2p** | |
| gte-Qwen2-7B-instruct | 56.92 | 56.46 | 51.74 | 50.09 | 46.65 | 46.23 | 44.13 | 73.55 | 74.13 | 79.86 | 49.41 | 53.91 |
| bge-en-icl | 57.89 | 54.44 | 49.33 | 53.05 | 48.38 | 45.86 | 44.33 | 72.33 | 72.72 | 81.32 | 46.05 | **68.98** |
| NV-Embed-v2 | 58.46 | 55.80 | 51.26 | **54.09** | **49.60** | 46.09 | 44.86 | 71.10 | 74.94 | **82.10** | 48.36 | 64.82 |
| Mistral-7B-Instruct-v0.2 | 39.74 | 40.18 | 37.30 | 36.94 | 35.16 | 40.74 | 39.53 | 48.10 | 36.58 | 45.81 | 35.41 | 41.44 |
| Claude 3 | 40.72 | 43.76 | 39.17 | 37.20 | 35.88 | 43.82 | 42.21 | 44.80 | 40.45 | 41.59 | 35.23 | 43.78 |
| LentEx (Ours) | **59.54** | **61.42** | **55.71** | 52.58 | 48.96 | **49.15** | **46.85** | **74.63** | **79.59** | 64.90 | **56.63** | 64.56 |

TABLE IV
RETRIEVAL METRICS FOR COLIEE AND BIOASQ DATASETS. CAPTAIN RESULTS ARE ONLY AVAILABLE FOR COLIEE AS IT IS A LEGAL DOMAIN-SPECIFIC SYSTEM

| Model | COLIEE | | | BioASQ | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F-1 | Precision | Recall | F-1 |
| RAG (Baseline) | 0.248 | 0.167 | 0.199 | 0.612 | 0.441 | 0.513 |
| RAG + Mistral | 0.644 | 0.536 | 0.585 | 0.681 | 0.496 | 0.574 |
| RAG + Claude | 0.715 | 0.663 | 0.688 | 0.704 | 0.528 | 0.603 |
| RAG + Pre-trained NER | 0.652 | 0.513 | 0.574 | 0.657 | 0.513 | 0.576 |
| RAG + Topic Modeling | 0.667 | 0.591 | 0.627 | 0.649 | 0.502 | 0.566 |
| CAPTAIN | 0.787 | **0.708** | **0.745** | - | - | - |
| **RAG + LentEx (Ours)** | **0.792** | 0.681 | 0.732 | **0.742** | **0.595** | **0.66** |

of distinct homogeneity and completeness scores.

$$V_\beta = (1+\beta) \cdot \frac{H \cdot C}{(\beta \cdot H) + C} \tag{2}$$

where **H** is the homogeneity score, **C** is the completeness score, $\beta$ is a parameter that balances the trade-off between homogeneity and completeness. If $\beta = 1$, then the V-measure is the harmonic mean of homogeneity and completeness.

**Result:** Experimental results in Table III demonstrate that LentEx consistently outperforms both embedding-based and prompting-based baselines across most datasets. LentEx achieves the highest average v-measure score of $59.54$, outperforming strong embedding-based baselines like NV-Embed-v2 ($58.46$) and bge-en-icl ($57.89$). The performance gains were particularly pronounced in paragraph-level tasks, with notable improvements in arxiv-p2p ($61.42$ vs. $56.46$) and reddit-p2p ($79.59$ vs. $74.94$). These results demonstrate LentEx's robustness and generalizability across different text granularities (p2p and s2s) and domain types, from scientific literature to social media content.

### C. *Improving Retrieval in RAG with Latent Entity Filtering*

While clustering demonstrates LentEx's information structuring capabilities, we further evaluate it in retrieval tasks by integrating it into a RAG pipeline.

*1) **Datasets**:* We conduct our evaluation on two distinct domain-specific datasets: COLIEE [29] for legal domain and BioASQ [30] for biomedical domain. The COLIEE test set comprises $100$ legal queries and $375$ supporting documents. We identify the latent entity types for this dataset including *case_type, parties_involved, jurisdiction, case_outcome*. For the BioASQ-2023 *11b Phase A* dataset [31], we utilize the Summary questions from training set, which consists of $1130$ questions and $4719$ supporting documents. The latent entities extracted for this domain are specifically selected to capture key biomedical concepts: *diseases, drugs, symptoms, treatments, anatomical_terms, and biomarkers*. These latent entities serve as crucial pre-filters for enhancing the relevance of evidence retrieval.

*2) **Baselines**:* We compare our RAG+LentEx system against a suite of baselines to assess the effectiveness of latent entity filtering. The primary baseline implemented a standard RAG architecture [32] performing retrieval without latent entity pre-filtering. We further compare against four RAG variants incorporating named and latent entity extraction methodologies: (i) RAG+Mistral, utilizing Mistral-7B-Instruct-v0.2 [20], (ii) RAG+Claude, utilizing Claude-3-Haiku [7] for latent entity extraction, (iii) RAG+NER: RAG augmented with a pre-trained Named Entity Recognition (NER) model [33], and (iv) RAG+Topic Modeling, which implemented topic-based entity pre-filtering [3]. All systems maintain identical retrieval and embedding configurations, including paragraph-level chunking and dual-encoder architecture with Qwen1.5-7B-instruct [34].

*3) **Experimental Setup**:* We segment documents into paragraph-level chunks (max length: $512$ tokens, overlap: $10$ tokens) for fine-grained retrieval, embedding them using Langchain's OpenSearch with gte-Qwen1.5-7B-instruct [34].

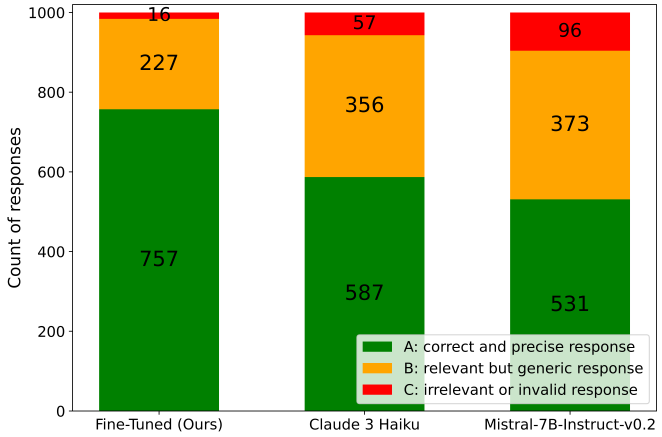


Fig. 4. LLM evaluation of model responses over 1000 data points. Claude-3-Sonnet model was prompted to rate model responses as A, B, or C based on the SOP shown. Evaluations for different models were conducted independently, allowing for identical ratings across models.

LentEx extracts latent entities from paragraphs and stores them as metadata. During inference, LentEx extracts entities from queries in real-time and filters documents based on metadata alignment before retrieval to enhance result relevance. Initial observations show that excessive filtering parameters excludes relevant documents, while insufficient filtering yields non-relevant results. The heterogeneous nature of entity types across diverse questions presented challenges in developing a universally applicable query protocol. To address these limitations, we implement dynamic queries using OpenSearch Query-DSL [35] with varying entity combinations as filters. This approach leverages OpenSearch's *min_should_match* parameter to specify the required quantity of matching filters, iteratively adjusting query restrictiveness. This facilitates query-specific configuration of both the combination and quantity of filters necessary for optimal alignment between question and document metadata. Following retrieval, we apply ms-marco-MiniLM-L-6-v2 [36] for re-ranking, extracting the top 10 results.

**Results** The experimental results in Table IV demonstrate the effectiveness of our RAG+LentEx approach across both legal and biomedical domains. On the COLIEE legal dataset, RAG+LentEX achieved the highest precision (0.792) and competitive recall (0.681), resulting in an F1-score of 0.732. This represents a significant improvement over the baseline RAG system, which achieves only 0.199 F1-score. Notably, our system's performance is comparable to CAPTAIN [37], the current state-of-the-art system, even slightly outperforming it in precision (0.792 vs 0.787). In the biomedical domain (BioASQ), RAG+LentEx achieves the highest scores across all metrics (P: 0.742, R: 0.595, F1: 0.66). The second-best performance is achieved by RAG+Claude (F1: 0.603), suggesting the effectiveness of large language models in entity extraction for specialized domains. The performance of LentEx across both domains demonstrates its robustness in diverse domain-specific retrieval tasks.

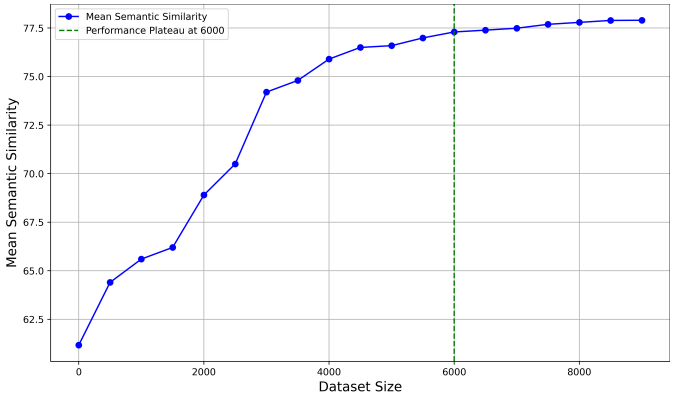


Fig. 5. Effect of dataset size on model performance. Using synthetic outputs as reference labels, semantic similarity is measured between model predictions and labels. The y-axis represents the mean semantic similarity on test dataset.

### D. *LLM as an Evaluator*

Next, we assess the relevance of model outputs by categorizing them into three buckets: A representing correct and precise response, B representing relevant but generic response, and C representing irrelevant or invalid response. We use Claude-3.5-Sonnet [38] as the evaluator model [39], which classifies the output based on these categories. The order of model outputs is shuffled for each evaluation to ensure unbiased assessments.

**Result:** As depicted in Figure 4, LentEx significantly outperforms both baseline models, with 98.4% of instances classified as either A or B, confirming the robustness of its fine-tuning approach.

## V. Ablation Studies

### A. *Impact of Dataset Size on Model Fine-Tuning*

We systematically investigate the effect of varying dataset sizes on model fine-tuning to determine the optimal amount of data required for efficient training. The model is fine-tuned using progressively larger subsets of training data, and its performance is evaluated on a held-out test set. As depicted in Figure 5, the model's performance exhibits consistent improvement up to a dataset size of 6000. Beyond this point, the performance plateaus, suggesting that 6000 samples constitute a sufficient data size for achieving optimal fine-tuning outcomes under the current setup. This result highlights the model's efficiency in learning from a moderate-sized dataset without extensive data collection.

### B. *Effect of Synthetic Data Generation on Data Quality*

To evaluate the effectiveness of the template-based synthetic data generation strategy introduced in Section III-C1, we conduct a manual assessment on 2000 randomly sampled examples, with 20 instances drawn from each domain. The evaluation focuses on two critical aspects: (1) the alignment between the input text and its designated domain and (2) the accuracy of the output latent entity labels. As shown in Table V, 98% of the sampled instances demonstrate alignment with their respective text types, and 91.5% of the output labels

TABLE V
MANUAL DATA QUALITY EVALUATION OF INPUT TEXTS AND LABELS FROM SYNTHETIC DATA, ON A STRATIFIED SAMPLE OF 2000 EXAMPLES

| **Quality Review** | **Yes %** |
|---|---|
| Is synthetic input text aligned with the text type? | 98 |
| Is latent entity label correct? | 91.5 |

are deemed correct. In contrast, when the output labels are generated independently after all input instances are created [16], the correctness rate drops to $91\%$. These results affirm that generating input-output pairs within the same model call leads to higher data quality and reinforces the efficacy of our template-based synthetic data generation pipeline.

## VI. RESULTS AND DISCUSSION

LentEx excels in addressing latent entity extraction (LEE) through a practical and cost-effective approach that leverages synthetic data generation and instruction fine-tuning. This methodology is particularly well-suited for low-resource environments where labeled data is scarce, enabling the model to generalize across a wide range of domains. Importantly, while LentEx is trained exclusively on synthetic data, our comprehensive evaluation on diverse real-world datasets—including 11 MTEB clustering benchmarks spanning scientific literature, social media, and news, along with specialized legal (COLIEE) and biomedical (BioASQ) retrieval datasets—demonstrates robust generalization capabilities. The strong performance across varied datasets validates that our synthetic data generation successfully captures the complexity and distribution of naturally occurring text, enabling effective transfer to practical applications without domain-specific labeling. The model's versatility is further evidenced by its performance in zero-shot settings, while its integration into Retrieval-Augmented Generation (RAG) systems boosts retrieval relevance and precision. LentEx's optimal performance with moderate dataset size underscores its efficiency in balancing data quality and diversity, making it a highly adaptable tool for both academic research and industry applications where latent entity recognition enhances content personalization, knowledge discovery, and information retrieval systems.

## VII. ERROR ANALYSIS

Analysis of LentEx on clustering and RAG experiments revealed key insights into it's performance and limitations. In clustering, errors primarily occurred in ambiguous entity assignments, where multiple latent entities existed but LentEx selects a suboptimal one. This effect is more profound in datasets such as twenty-newsgroups and stackexchange, where the type of latent entity is generic. This leads to mis-classified clusters, leading to inferior performance. In RAG experiment, errors are linked to over-restrictive filtering, where relevant documents are excluded due to incomplete latent entity extraction. This is particularly noticable in COLIEE dataset, where case law often involves intersecting legal categories (e.g., *contract law vs. tort law*), and LentEx may miss documents if it fails to infer all relevant latent entities from the query. Addressing these limitations through domain-specific fine-tuning and adaptive entity selection strategies could further enhance LentEx's performance in real-world applications.

## VIII. CONCLUSION AND FUTURE WORK

We present LentEx, a novel, supervised framework for latent entity extraction, leveraging synthetic data generation and instruction fine-tuning to overcome the challenge of labeled data scarcity. LentEx demonstrates superior generalization across multiple domains and tasks, outperforming state-of-the-art models. Our experiments validate the efficacy of LentEx on various downstream tasks, including clustering and retrieval-augmented generation, which enhances the quality and relevance of retrieved results.

While LentEx sets a strong foundation for latent entity extraction, several avenues remain for future research. First, further refinement of filtering mechanisms in retrieval-augmented systems is needed to address the observed drop in Recall. Incorporating adaptive filtering strategies, possibly informed by context-aware retrieval, could mitigate this limitation. Although our synthetic data generation strategy has proven effective, exploring more sophisticated techniques, such as self-supervised training, contrastive learning or reinforcement learning, could further enhance the model performance. Another critical direction is the application of LentEx to more complex, domain-specific tasks, such as temporal reasoning or multi-hop inference, where latent entities can play a pivotal role in understanding and decision-making. LEE provides richer, structured representations that can be integrated into knowledge graphs, search engines, and content recommendation systems. Lastly, addressing ethical concerns and potential biases inherent in large-scale language models will be crucial. We highly encourage researchers to build on LentEx by exploring training strategies, expanding generalization capabilities, and addressing ethical considerations.

## REFERENCES


[1] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Knight, A. Nenkova, and O. Rambow, Eds. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 260–270. [Online]. Available: https://aclanthology.org/N16-1030

[2] E. Shoshan and K. Radinsky, "Latent entities extraction: How to extract entities that do not appear in the text?" in *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 2018, pp. 200–210.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[4] Y. Miao, E. Grefenstette, and P. Blunsom, "Discovering discrete latent topics with neural variational inference," 2018.

[5] M. Grootendorst, "Bertopic: Neural topic modeling with a class-based tf-idf procedure," *arXiv preprint arXiv:2203.05794*, 2022.

[6] W. Xu, W. Hu, F. Wu, and S. Sengamedu, "Detime: Diffusion-enhanced topic modeling using encoder-decoder based llm," in *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 2023. [Online]. Available: http://dx.doi.org/10.18653/v1/2023.findings-emnlp.606

[7] Anthropic, "Model card and evaluations for claude models," https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf, 2023, accessed: 2023-12-05.

[8] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022. [Online]. Available: https://arxiv.org/abs/2210.11416

[9] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, E. Goffinet, D. Heslow, J. Launay, Q. Malartic, B. Noune, B. Pannier, and G. Penedo, "Falcon-40B: an open large language model with state-of-the-art performance," 2023.

[10] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[13] "Chatgpt," 2023, https://chat.openai.com/, Accessed: 2023-04-20.

[14] K. Lu, X. Pan, K. Song, H. Zhang, D. Yu, and J. Chen, "Pivoine: Instruction tuning for open-world information extraction," *arXiv preprint arXiv:2305.14898*, 2023.

[15] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Alpaca: A strong, replicable instruction-following model," *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, vol. 3, no. 6, p. 7, 2023.

[16] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language models with self-generated instructions," *arXiv preprint arXiv:2212.10560*, 2022.

[17] T. Yu, C. Jiang, C. Lou, S. Huang, X. Wang, W. Liu, J. Cai, Y. Li, Y. Li, K. Tu *et al.*, "Seqgpt: An out-of-the-box large language model for open domain sequence understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 19 458–19 467.

[18] B. Fetahu, Z. Chen, O. Rokhlenko, and S. Malmasi, "Instructpts: Instruction-tuning llms for product title summarization," *arXiv preprint arXiv:2310.16361*, 2023.

[19] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022.

[20] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[22] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," 2020.

[23] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "Mteb: Massive text embedding benchmark," *arXiv preprint arXiv:2210.07316*, 2022.

[24] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, and Z. Fan, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[25] S. Xiao, Z. Liu, P. Zhang, and N. Muennighoff, "C-pack: Packaged resources to advance general chinese embedding," 2023.

[26] C. Lee, R. Roy, M. Xu, J. Raiman, M. Shoeybi, B. Catanzaro, and W. Ping, "Nv-embed: Improved techniques for training llms as generalist embedding models," 2025. [Online]. Available: https://arxiv.org/abs/2405.17428

[27] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: Massive text embedding benchmark," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 2014–2037. [Online]. Available: https://aclanthology.org/2023.eacl-main.148

[28] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, J. Eisner, Ed. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 410–420. [Online]. Available: https://aclanthology.org/D07-1043

[29] Y. Kano, M.-Y. Kim, M. Yoshioka, Y. Lu, J. Rabelo, N. Kiyota, R. Goebel, and K. Satoh, "Coliee-2018: Evaluation of the competition on legal information extraction and entailment," in *New Frontiers in Artificial Intelligence: JSAI-IsAI 2018 Workshops, JURISIN, AI-Biz, SKL, LENLS, IDAA, Yokohama, Japan, November 12–14, 2018, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, 2018, p. 177–192. [Online]. Available: https://doi.org/10.1007/978-3-030-31605-1_14

[30] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. Alvers, D. Weißenborn, A. Krithara, S. Petridis, D. Polychronopoulos, Y. Almirantis, J. Pavlopoulos, N. Baskiotis, P. Gallinari, T. Artieres, A.-C. Ngonga Ngomo, N. Heino, E. Gaussier, L. Barrio-Alvers, and G. Paliouras, "An overview of the bioasq large-scale biomedical semantic indexing and question answering competition," *BMC Bioinformatics*, vol. 16, p. 138, 04 2015.

[31] A. Nentidis, G. Katsimpras, A. Krithara, S. Lima López, E. Farré-Maduell, L. Gasco, M. Krallinger, and G. Paliouras, *Overview of BioASQ 2023: The Eleventh BioASQ Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering*. Springer Nature Switzerland, 2023, p. 227–250. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-42448-9_19

[32] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[33] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Knight, A. Nenkova, and O. Rambow, Eds. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 260–270. [Online]. Available: https://aclanthology.org/N16-1030/

[34] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, "Towards general text embeddings with multi-stage contrastive learning," *arXiv preprint arXiv:2308.03281*, 2023.

[35] OpenSearch Project, "Query dsl documentation," https://opensearch.org/docs/latest/query-dsl/, 2024, accessed: 2024-02-06.

[36] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[37] C. Nguyen, P. Nguyen, T. Tran, D. Nguyen, A. Trieu, T. Pham, A. Dang, and L.-M. Nguyen, "Captain at coliee 2023: Efficient methods for legal information retrieval and entailment tasks," 2024. [Online]. Available: https://arxiv.org/abs/2401.03551

[38] Anthropic, "Claude 3 model card addendum," 2024, accessed: 2024-09-29.

[39] C.-H. Chiang and H.-y. Lee, "Can large language models be an alternative to human evaluations?" in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 15 607–15 631. [Online]. Available: https://aclanthology.org/2023.acl-long.870