UltraWiki: A New Benchmark for Ultra-fine-grained Entity Set Expansion with Negative Seed Entities

Yangning Li[1,2], Qingsong Lv[1], Tianyu Yu[1], Yinghui Li[1], Xuming Hu[3], Wenhao Jiang[4]
Hai-Tao Zheng[1,2], Hui Wang[2], and Philip S. Yu[5] *Life Fellow, IEEE*

[1] *Shenzhen International Graduate School, Tsinghua University*
[2] *Pengcheng Laboratory*
[3] *The Hong Kong University of Science and Technology (Guangzhou)*
[4] *Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)*
[5] *University of Illinois Chicago*

***Abstract*—Entity Set Expansion (ESE) aims to identify new entities belonging to the same semantic class as the given set of seed entities. Traditional methods solely relied on positive seed entities to represent the target fine-grained semantic class, rendering them tough to represent ultra-fine-grained semantic classes. Specifically, merely relying on positive seed entities leads to two inherent shortcomings: (i) Ambiguity among ultra-fine-grained semantic classes. (ii) Inability to define "unwanted" semantics. Hence, previous ESE methods struggle to address the ultra-fine-grained ESE (Ultra-ESE) task. To solve this issue, we first introduce negative seed entities in the inputs, which jointly describe the ultra-fine-grained semantic class with positive seed entities. Negative seed entities eliminate the semantic ambiguity by providing a contrast between positive and negative attributes. Meanwhile, it provides a straightforward way to express "unwanted". To assess model performance in Ultra-ESE and facilitate further research, we also constructed UltraWiki, the first large-scale dataset tailored for Ultra-ESE. UltraWiki encompasses 50,973 entities and 394,097 sentences, alongside 236 ultra-fine-grained semantic classes, where each class is represented with 3-5 positive and negative seed entities. Moreover, a retrieval-based framework RetExpan and a generation-based framework GenExpan are proposed to provide powerful baselines for Ultra-ESE. Additionally, we devised two strategies to enhance models' comprehension of ultra-fine-grained entities' semantics: contrastive learning and chain-of-thought reasoning. Extensive experiments confirm the effectiveness of our proposed strategies and also reveal that there remains a large space for improvement in Ultra-ESE. All the codes, dataset, and supplementary notes are available at https://github.com/THUKElab/UltraWiki.**

***Index Terms*—Entity Set Expansion, Knowledge Mining, Language Model, Ultra-fine-grained Semantic Understanding**

1. Introduction

Entity Set Expansion (ESE) is a critical task aiming to expand new entities belonging to the same semantic class as the given set of seed entities [1–4]. For example, given the input seed entity set {*Motorola*, *Xiaomi*, *Nokia*}, an ESE model will output more new entities (e.g., "*Huawei*", "*Samsung*", ...) that all belong to the same `Mobile Phone brands` class as the seed entities. ESE plays an important role for a wide range of user-tailored applications [5–9], thus demanding a high degree of semantic class granularity. For instance, in the context of recommendation systems [10, 11], a more nuanced ESE method can benefit more precise product recommendations.

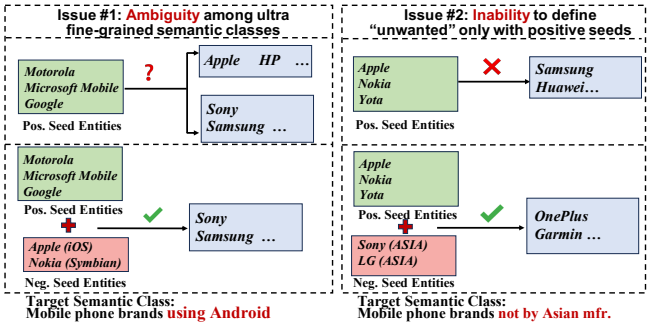

Figure 1. The figure illustrates how relying only on positive seeds can lead to ambiguity between similar classes (e.g., `Mobile phone brands using Android` vs. `American Mobile phone brands`) and the inability to effectively define negative constraints (e.g., `Mobile phone brands not made by Asian manufacturers`). The introduction of negative seed entities in the input solves these issues.

Conventionally, existing ESE methods focus on the expansion of "wanted" fine-grained semantic classes merely based on positive seed entities, such as `Car Brand` and the above mentioned `Phone Brand`. However, current ESE methods struggle to expand ***ultra**-fine-grained semantic classes* [12–14], which involve more specific attribute constraints (e.g., mobile phone brands using Android, Asian mobile phone brands). Since ultra-fine-grained semantic classes from the same concept often exhibit substantial overlap in target entities, rendering them tough to be represented

only with a handful of positive seed entities.

Specifically, as shown in Figure 1, merely relying on positive seed entities to represent ultra-fine-grained semantic classes causes two issues: (i) **Ambiguity among ultra-fine-grained semantic classes.** For instance, with the positive seed entities {*Motorola*, *Microsoft Mobile*, *Google*}, it is confusing whether the user intends to extend `Mobile phone brands using Android` or `American Mobile phone brands`. Since the positive seed entities alone are insufficient to delineate the distinct class features. (ii) **Inability to define "unwanted".** Positive seed entities alone fail to represent the "unwanted" semantic. Considering when users wish to find more `Mobile phone brands not made by Asian manufacturers` entities, it is impractical to define this semantic class by enumerating brands from other continents in positive seed entities. As the size of the positive seed entities is limited, it's also unreasonable to expect users to specify all potentially desired attribute values.

Due to the inherent shortcomings of the task inputs, prior ESE methods failed to conduct ultra-fine-grained expansion. To address this issue, inspired by exploiting negative feedback to model user intent in recommendation system [15–17], we propose enhancing ESE by incorporating negative seed entities, aiming at ultra-fine-grained ESE (Ultra-ESE). The input negative seed entities belong to the same fine-grained semantic class as the positive seed entities but differs in certain attributes. They jointly represent target semantic class, addressing the semantic granularity issue head-on: First, the contrast between positive and negative seed entities in terms of attributes highlights the user's specific interests and eliminates the ambiguity arising from positive seed entities alone. For the aforementioned example, if negative seed entities are about `Mobile phone brands not using Android`, it indicates that the user is focused on operating system rather than manufacturer. Second, negative seed entities constrain the expansion space and can readily express "unwanted" semantics.

Regrettably, previous ESE dataset [18, 19] lacked the concept of negative seed entities, let alone ultra-fine-grained semantic classes. To bridge this gap, we constructed UltraWiki, the first large-scale dataset tailored for Ultra-ESE. Derived from Wikipedia, UltraWiki encompasses 10 fine-grained semantic classes, 50,973 entities, and 394,097 sentences. During the curation, we annotated 2-3 attributes for each fine-grained semantic class to further construct ultra-fine-grained semantic classes. Based on the permutations of attributes, 235 ultra-fine-grained semantic classes are constructed, with each semantic class represented with 3-5 positive and negative seed entities. On average, each ultra-fine-grained semantic class contains 23 (236/10-1) hard negative semantic classes, which belong to the same fine-grained semantic class and may exhibit substantial overlap in target entities. Experiments proved that even advancing GPT-4 can not handle it well.

In experiments, we evaluated existing ESE methods on UltraWiki. Furthermore, to comprehensively assess the efficacy of large language models (LLMs) with two different paradigms on the Ultra-ESE task, we designed a retrieval-based framework, RetExpan, with encoder-only LLM BERT [20] and a generation-based framework, GenExpan, with decoder-only LLM LLaMA [21]. Meanwhile, we proposed three strategies to enhance models' ability to comprehend ultra-fine-grained semantics of entities: contrastive learning, retrieval augmentation, and chain-of-thought reasoning.

In summary, the main contributions are listed as follows:

- We proposed the more challenging ultra-fine-grained ESE task, and incorporated negative seed entities to represent ultra-fine-grained semantic classes more precisely for the first time.
- We constructed the first large-scale dataset UltraWiki, tailored for ultra-fine-grained ESE tasks. It encompasses 10 fine-grained semantic classes and 261 ultra-fine-grained semantic classes.
- We designed both retrieval-based and generation-based frameworks to assess the efficacy of BERT-like and GPT-like LLMs on the UltraWiki dataset. Furthermore, two strategies were proposed for refining the semantic comprehension of ultra-fine-grained entities.
- Extensive experiments confirmed the effectiveness of our proposed strategies and also reveal significant potential for enhancing ultra-fine-grained semantic comprehension of entities by LLMs.

2. Related Work

2.1. Methods of ESE

Recently, increasing emphasis has been placed on the expansion of fine-grained (e.g., US city) rather than coarse-grained (e.g., Location) semantic classes. ProbExpan [3] employs heuristic methods to mine hard negative entities of target semantic classes, thereby refining the semantic boundaries of fine-grained classes through contrastive learning [22, 23]. FGExpan [24] leverages the existing taxonomy to direct BERT in reasoning about more fine-grained names of semantic classes. Besides, MultiExpan [4] integrates multimodal pre-trained models to encode visual information (images) associated with entities, which benefits the differentiation of various fine-grained semantic classes. Nevertheless, these methods do not essentially tackle the issue that solely using positive seed entities fails to sufficiently represent ultra-fine-grained semantic classes. In contrast, our research introduces the novel concept of incorporating negative seed entities alongside positive ones to describe an ultra-fine-grained semantic class.

It is worth mentioning that there has been some work to incorporate negative entities [3, 25–27], though the usage of negative entities in these methods is relatively naive. We point out that the role of the negative entities used in previous work is fundamentally different from ours. Negative entities in previous work are purely used to help determine the boundary of the target set described by positive seed entities. In contrast, negative entities in our model are used

to describe the target ultra-fine-grained classes that cannot be characterized by positive seed entities alone.

2.2. Data Resources of ESE

Queries in the existing ESE dataset merely consist of positive seed entities. The approach of representing ultra-fine-grained semantic classes solely via positive seed entities inherently faces two major issues: (i) Ambiguity among ultra-fine-grained semantic classes. (ii) Inability to define "unwanted" semantics. Consequently, existing datasets lack ultra-fine-grained semantic classes. For instance, Wiki [18] and APR [18], derived from the Wikipedia and Reuters corpora respectively, encompass merely 3 and 8 fine-grained semantic classes. Some named entity recognition datasets are directly used as ESE evaluation benchmarks, they contain semantic classes with coarser granularity. For example, the OntoNotes [19] and CoNLL [28] datasets include semantic classes like organization and location. In contrast, UltraWiki encompasses 261 ultra-fine-grained semantic classes, which surpasses existing ESE datasets in terms of both granularity and quantity of semantic classes.

3. Task Formulation

We focus on the **ultra-fine-grained ESE** task, whose input consists of three components: query S, candidate entities V, and corpus D. The query $S = \{S^{pos} \cup S^{neg}\}$ is a composite set comprising positive seed entities $S^{pos} = \{e_1^{pos}, \ldots, e_k^{pos}\}$ and negative seed entities $S^{neg} = \{e_1^{neg}, \ldots, e_k^{neg}\}$, which belong to the same fine-grained semantic class c. The entities in S^{pos} share the same values for positive attribute set $\mathcal{A}^{pos}$:

$$e.\,\text{get}(a_i) \equiv k_i, \forall e \in S^{pos}, \forall a_i \in \mathcal{A}^{pos} = \{(a_i, k_i)\}_{i=1}^{|\mathcal{A}^{pos}|} \tag{1}$$

where a_i and k_i are attribute and value, respectively. We define positive target entities $\mathcal{P}$ as entities that share the same attribute values with positive seed entities in $\mathcal{A}^{pos}$. Likewise, negative target entities in $\mathcal{N}$ share the same attribute values with negative seed entities in $\mathcal{A}^{neg}$.

Overall, the target ultra-fine-grained semantic class means class c that holds the same values with positive seed entities in $\mathcal{A}^{pos}$, but different values with negative seed entities in $\mathcal{A}^{neg}$. Examples can be found in Figure 2.

Hence, the ultimate objective of ultra-fine-grained ESE is to identify entities from candidate entities V that belong to the $\mathcal{P}$, while being distinct from $\mathcal{N}$ (e.g., $\mathcal{P} - \mathcal{N}$). In an ideal feature space, entities that share more of the same attribute values should be positioned closer.

4. UltraWiki Dataset

Considering the lack of ultra-fine-grained semantic classes and negative seed entities in existing ESE datasets, we constructed UltraWiki, the first large-scale dataset tailored for Ultra-ESE. This section outlines the construction process of UltraWiki. Semantic granularity, annotation consistency, and candidate entities' difficulty are fully considered to guarantee the quality of UltraWiki.

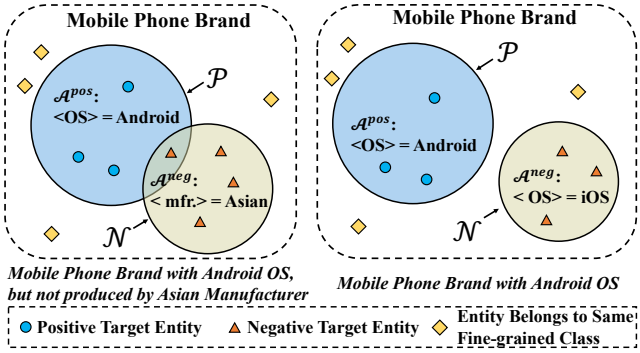

Figure 2. Left target semantic class: mobile phone brand with Android OS, not produced by Asian manufacturer. Right target semantic class: mobile phone brand with Android OS. The target entity set is $\mathcal{P} - \mathcal{N}$.

4.1. Dataset Construction

There are two strategies for constructing UltraWiki. A more direct strategy is to first collect a large number of entities and corresponding context sentences. Then, for each fine-grained semantic class, the human annotator traverses the entire entity vocabulary to identify the corresponding entity and annotates the attribute values, thereby constructing the ultra-fine-grained semantic class. However, despite the availability of numerous entity-centric datasets, this method is labor-intensive and prone to omit entities. Consequently, we chose a more practical strategy. We begin by identifying fine-grained semantic classes along with their corresponding entities, then collecting corresponding sentences and attribute information for each entity in turn. Figure 3 illustrates the entire process.

Step 1. Semantic Classes and Entities Collection. Wikipedia maintains an extensive catalog of entities associated with fine-grained semantic classes. We identified 10 fine-grained semantic classes and then crawled the corresponding entities. Additionally, a substantial number of entities sampled from Wikipedia pages were incorporated into the candidate entity vocabulary as negative entities.

Step 2. Entity-Labeled Sentences Collection. A significant volume of text is also crawled from Wikipedia pages, where entities are uniquely identified by hyperlinks. Since the entities obtained in Step 1 also contain hyperlinks, we can readily align entities with sentences containing them, thereby providing abundant contextual information for entities.

Step 3. Entity Attribute Annotation. For each fine-grained semantic class C, we manually select k independent attributes $\mathcal{A} = \{a_1, a_2, \cdots, a_k\}$, which will be used to generate ultra-fine-grained semantic class in next step. Attributes of each class are shown in supplementary notes. Subjective attributes are avoided like good or bad. We first query Wikidata API for the corresponding attribute value with

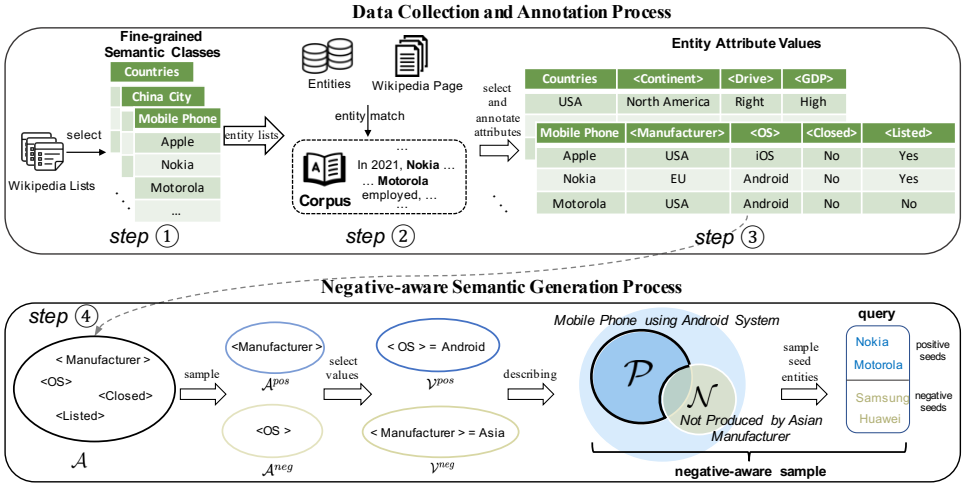

Figure 3. Illustration of the UltraWiki dataset construction process. The intermediate data obtained at each step is bolded in the figure.

Python script. For the remaining attributes that could not be automatically annotated, we employed human annotators to manually label them, ensuring each entity was reviewed three times for accuracy and consistency.

Step 4. Negative-aware Semantic Classes Generation. We devised an algorithm for automated generation of ultra-fine-grained semantic classes based on annotated attributes. Specifically, for attributes $\mathcal{A}$ of each fine-grained semantic class C, we sample m and n attributes as positive and negative attribute set (i.e., $\mathcal{A}^{pos}$ and $\mathcal{A}^{neg}$). Subsequently, we pick a value for all the attributes in $\mathcal{A}^{pos}$ and $\mathcal{A}^{neg}$. This process yields a tuple of positive values $\mathcal{V}^{pos}$ and a tuple of negative values $\mathcal{V}^{neg}$, which jointly constrain the attributes of the target semantic class to achieve ultra-fine-grained control of the semantic class.

When $\mathcal{A}^{pos}$ and $\mathcal{A}^{neg}$ are the same, the role of $\mathcal{A}^{neg}$ is to emphasize attributes concerned by user and mitigate ambiguity. Conversely, when they differ, $\mathcal{A}^{neg}$ serves to convey unwanted semantics. As described in the Section 3, positive and negative target entity sets are denoted as $\mathcal{P}$ and $\mathcal{N}$. Both $|\mathcal{P}|$ and $|\mathcal{N}|$ are ensured to exceed the minimum entity requirement $n_{thred} = 6$.

4.2. Dataset Analysis

Statistics of UltraWiki. UltraWiki is the first large-scale dataset featuring ultra-fine-grained semantic classes. It comprises 50,973 entities and 394,097 sentences sourced from Wikipedia. 10 fine-grained semantic classes are determined, that comprehensively cover five major entity types, including Organization, Location, Product, Person, and Miscellaneous. Leveraging the combination of positive and negative attributes, we further derived 261 ultra-fine-grained semantic classes. On average, each ultra-fine-grained semantic class comprises 63 positive target entities ($\mathcal{P}$) and 60 negative target entities ($\mathcal{N}$) that are not expected to be expanded. Each ultra-fine-grained semantic class encompasses 3 queries, with 3 to 5 positive and negative seed entities, respectively.

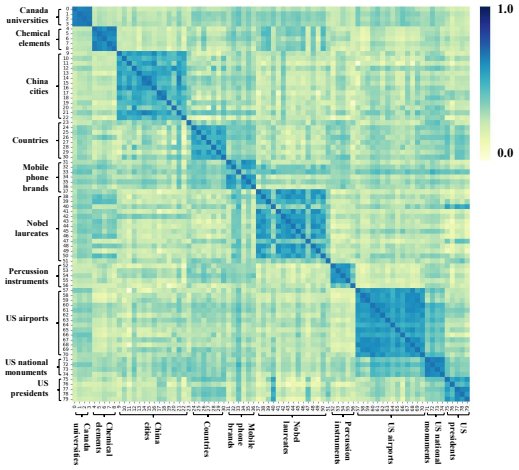

Figure 4. Semantic similarity heatmap of ultra-fine-grained semantic classes. Rows/columns: averaged embeddings of ground-truth positive entities per class; cell colors: pairwise cosine similarities. Classes were proportionally sampled to 80 classes based on total size for clarity.

The majority of the semantic classes are constrained by one positive and one negative attribute.

TABLE 1. COMPARISON OF ESE DATASETS.

	Wiki	APR	CoNLL	ONs	UltraWiki
# Semantic Classes	8	3	4	8	261
Semantic granularity	Fine	Fine	Coarse	Coarse	Ultra-Fine
# Queries per Class	5	5	1	1	3
# Pos Seeds per Query	3	3	10	10	3-5
# Neg Seeds per Query	N/A	N/A	N/A	N/A	3-5
# Candidate Entities	33K	76K	6K	20K	51K
# Sentences of Corpus	973K	1043K	21K	144K	394K
Entity Attribution	✗	✗	✗	✗	✓

Quality of UltraWiki. The fine-grained semantic classes, entity corpus, and partial attributes of UltraWiki are automatically crawled from Wikipedia and Wikidata. Both

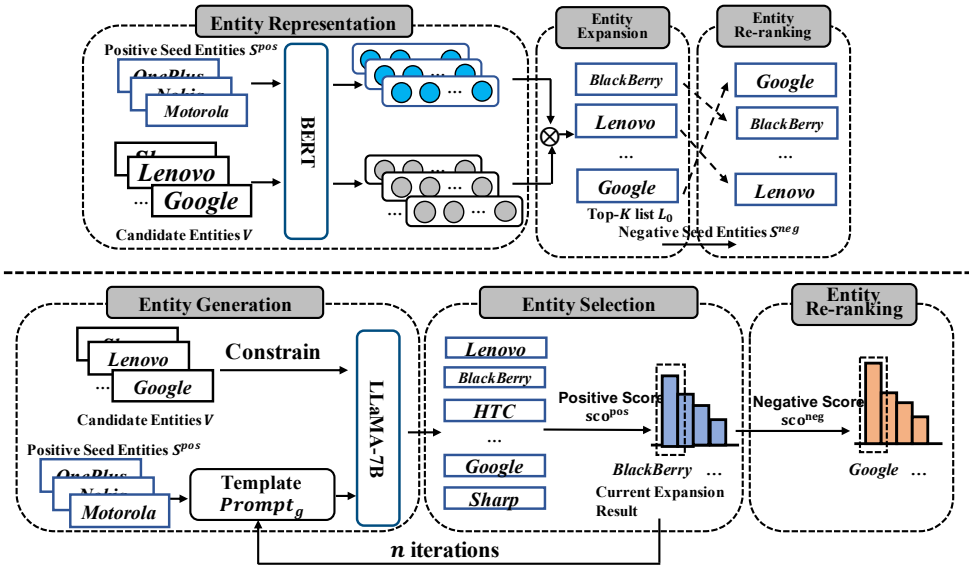

Figure 5. Upper: overall frameworks of our RetExpan. Below: overall frameworks of our GenExpan.

sources are high-quality knowledge bases, curated by numerous domain experts, thereby firmly guaranteeing the quality of UltraWiki. For attributes that require manual annotation, we ensure that each value is annotated by three annotators. Eventually, the inter-annotator agreement measured by Fleiss's Kappa [29] reaches 0.90, indicating satisfying consistency and accuracy of the annotation.

Difficulty of UltraWiki. Ultra-ESE is an inherently challenging task. The challenge arises from the substantial entity overlap among ultra-fine-grained semantic classes (approximately 99% of ultra-fine-grained semantic classes intersect), which requires the model to comprehend the ultra-fine-grained distinctions of entities across varied contexts. Even advanced models like GPT-4 struggle to address this challenge (refer to the results in the section of experiments).

Additionally, our selected semantic classes include a subset of long-tail entities, such as lesser-known Chinese cities, which typically lack extensive contextual knowledge. Simultaneously, employing the BM25-based search, we incorporated entities highly similar to the target entities as hard negative entities in the candidate entity vocabulary. The inclusion of these entity types contributes to the difficulty of UltraWiki. The visualization of semantic similarity in Figure 4 demonstrates that each semantic class in UltraWiki exhibits remarkably high intra-class similarity, where each row/column represents the average embedding of all ground-truth positive entities within a specific ultra-fine-grained semantic class.

5. Methods

This section presents our proposed retrieval-based and generation-based frameworks to leverage both positive and negative seed entities, namely RetExpan and GenExpan. Here, we re-emphasize that the inputs of the two frameworks comprise both positive and negative seed entities, which belong to the same fine-grained semantic class and solely differ in their ultra-fine-grained attributes.

Most prior ESE models are built upon retrieval-based frameworks, e.g., CGExpan and ProbExpan. The retrieval-based framework first encodes entities as features in low-dimensional semantic space, and then measures the probability that a candidate entity belongs to the target semantic class via entity feature similarity, which is naturally more suitable for retrieval tasks like ESE. With the advent of generative LLMs, we note their significant semantic comprehension and reasoning skills, which are developed through pre-training on large-scale corpora. These models can eliminate the intermediate step of entity embedding, allowing for a more streamlined, end-to-end Entity Set Expansion ESE process. Therefore, we also propose a generation-based framework to explore the potential of generative LLMs. Moreover, for both frameworks, enhancement strategies are crafted to enhance the models' capacity for ultra-fine-grained semantic comprehension with negative seed entities. The proposed RetExpan and GenExpan provide comprehensive and strong baselines for Ultra-ESE.

5.1. Retrieval-based Framework: RetExpan

5.1.1. Overall Framework. RetExpan is structured into three steps: entity representation, entity expansion, and entity re-ranking. In the first stage, we design the entity encoder that extracts contextual features of entities in sentences. An entity is represented as the mean of the feature vectors derived from all sentences containing it. The entity prediction task is introduced to refine the entity representation. During the second stage, a preliminary list of target entities is acquired based on their similarity to the positive seed entities. Notably, negative seed entities are excluded in this process to ensure the recall of all entities satisfying fine-grained semantic classes. In the third stage, negative

seed entities are also employed to re-rank the entity list obtained in the previous phase, reducing the ranking of entities aligning with negative attribute values.

Entity Representation. The goal of entity encoder is to extract contextual features of entities from textual data. Initially, we replace entity mentions within a sentence with [MASK] tokens to construct the input for the encoder. Given a contextual sentence T with masked entity mentions, we utilize the $\text{BERT}_{\text{BASE}}$, which comprises a 12-layer Transformer [30], to extract contextual embeddings:

$$H = \{h_1, h_2, ..., h_L\} = \text{BERT}_{\text{BASE}}(T) \tag{2}$$

where L is the length of the tokenized sentences. Ultimately, the contextual feature $\mathbf{h}$ of an entity is computed as the average of the contextual embedding mask $h_{[\text{MASK}]}$ at the mask position across all sentences containing it.

Inspired by ProbExpan, the entity prediction task is introduced to refine the entity representation. Concretely, a classification header $\mathbf{f}$ is appended to the entity encoder. After obtaining the hidden state $h_{[\text{MASK}]}$ at [MASK] position, it is transformed into the probability distribution of the masked entity among the candidate entities via MLP and SoftMax:

$$\hat{y} = \mathbf{f}\left(h_{[\text{MASK}]}\right) = \text{Softmax}\left(\text{MLP}\left(h_{[\text{MASK}]}\right)\right), \hat{y} \in \mathbb{R}^{V_e} \tag{3}$$

where V_e is the size of candidate entities. Then, cross-entropy loss with label smoothing [31] is applied to learn the latent semantics of entities:

$$\begin{aligned}\mathcal{L}_{mask} = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{V_e} & y_i[j]\cdot(1-\eta)\cdot log(\hat{y}_i[j]) \\ & +(1-y_i[j])\cdot\eta\cdot log(1-\hat{y}_i[j])\end{aligned} \tag{4}$$

in which ground-truth y is a one-hot vector and N is the batch size. Smoothing factor η mitigates over-penalization for entities that exhibit similar semantics to the ground-truth entity.

Entity Expansion. We acquire the preliminary entity list comprising entities that belong to the same fine-grained semantic class as the positive seed entities S^{pos}. For each candidate entity e, we define its positive similarity score sco^{pos} as follows:

$$sco^{pos}(e) = \frac{1}{|S^{pos}|}\sum_{e'\in S^{pos}} \cos\_\text{sim}\left(h(e), h(e')\right) \tag{5}$$

We keep top-K entities with the highest sco^{pos} as the initial expansion list L_0.

Entity Re-ranking. This step further considers negative seed entities, excluding entities that share semantics with them (i.e., possess the same value on negative attribute). Analogous to Equation 5, we define the negative similarity score sco^{neg} between the candidate entity and negative seed entities. However, directly re-ranking L_0 in ascending order utilizing sco^{neg} introduces a significant number of noisy entities that do not belong to the same fine-grained semantic class as the seed entities. These irrelevant entities, characterized by low similarity scores with the negative seed entities, may be erroneously assigned over-high ranking.

To address this issue, we propose a simple yet effective strategy: segmented re-ranking. This approach divides L_0 equally into $\left\lceil \frac{|L_0|}{l} \right\rceil$ segments and then conducts a descending re-ranking using sco^{neg} for each segment individually. Hyper-parameter l represents the length of each segment. This strategy facilitates a local fine ranking of L_0 and prevents the assignment of over-high rankings to noisy entities with quite low sco^{neg} values.

5.1.2. Enhancement Strategy 1: Ultra-fine-grained Contrastive Learning. To further enhance the ultra-fine-grained semantic comprehension capability of RetExpan, we devised attribute-aware contrastive learning. Conventional contrastive learning [32] facilitates clearer semantic boundaries for fine-grained semantic classes by drawing the representation of the same semantic class entities closer and the representation of different semantic class entities further apart. Although proven to be effective, basic contrastive learning is not precise enough for Ultra-ESE as it neglects entity differences inside the same semantic class. For instance, "Samsung" and "Motorola" may serve as positive pairs in terms of the attribute "Operating System" but as negative pairs for the attribute "Manufacturer". Ideally, entities with more attributes in common should exhibit closer distance in feature space. To this end, we construct ultra-fine-grained training data based on positive and negative seed entities to implicitly incorporate attribute-based differences into entity representations.

Ultra-fine-grained Training Data. We prompt GPT-4[1] to deduce potential positive (negative) attributes and return T entities from the initial list L_0 that are most similar to the given positive (negative) seed entities to form the entity list L_{pos} (L_{neg}). Entities in L_{pos} and L_{neg} are roughly considered to belong to the positive (target) and negative ultra-fine-grained semantic class $\mathcal{P}$ and $\mathcal{N}$, respectively. Hence, to distinguish entities with different attribute values under the same fine-grained semantic class, entities in L_{pos} and L_{neg} should be pulled away from each other. Specifically, we construct the contrastive training pairs as follows:

$$\begin{aligned}\mathbb{P}_{pos} = \{(x,x')|e(x)\in L_{pos}, e(x')\in L_{pos}\}\cup \\ \{(x,x')|e(x)\in L_{neg}, e(x')\in L_{neg}\}\cup\{(x,x')|e(x)=e(x')\},\end{aligned} \tag{6}$$

$$\begin{aligned}\mathbb{P}_{neg} = \{(x,y)|e(x)\in L_{pos}, e(y)\in L_{neg}\}\cup \\ \{(x,z)|e(x)\in L_{pos}\cup L_{neg}, e(z)\in \overline{L_0}\},\end{aligned} \tag{7}$$

where $e(x)$ indicates the entity of the training sample x. $\overline{L_0}$ denotes entities from other fine-grained semantic classes, which form normal negative pairs with entities from L_{pos} and L_{neg}. $\overline{L_0}$ is necessary to ensure that from the fine-grained semantic level, entities in L_{pos} and L_{neg} remain similar in the semantic space. Moreover, during the training process, we append the corresponding positive and negative

1. All specific prompts used for querying GPT-4 can be found in supplementary notes. We use GPT-4-1106.

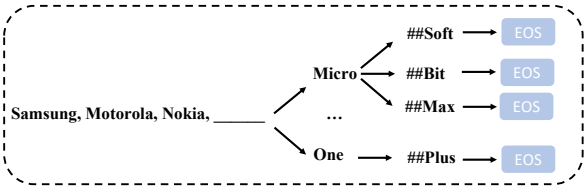

Figure 6. Prefix tree-based constrained decoding.

seed entities after each training sample (i.e., sentence) to implicitly specify the corresponding ultra-fine-grained semantics, avoiding positive-negative conflicts for the same entity pair.

Contrastive Loss. We choose the classical InfoNCE loss [33] as the contrastive loss. Contrastive learning is conducted in a new hypersphere space to prevent semantic collapse, which is transformed by another MLP-based mapping head $\mathbf{f_{cl}}$ and l-2 normalization.

5.2. Generation-based Framework: GenExpan

5.2.1. Overall Framework. To explore the potential of emerging generative LLMs, we devised the generative framework GenExpan. Unlike RetExpan which relies on intermediate semantic features of entities, GenExpan directly infers based on the given preceding text and knowledge stored in LLMs themselves, thus is more efficient. The proposed GenExpan consists of three phases: entity generation, entity selection, and entity re-ranking. In the first phase, GenExpan generates a new set of entities based on positive seed entities and current expanded entities. A constrained decoding strategy is designed to ensure that the generated entities are included in the candidate entities. Subsequently, the entity selection phase filters K entities to be added to the current expansion, based on the similarity scores between the generated entities and positive seed entities. The first two steps are executed iteratively for several rounds. Finally, akin to RetExpan, GenExpan re-ranks the previous expansion results to lower the ranking of entities identical to the negative seed entities in terms of negative attributes. GenExpan utilizes LLaMA-7B [21] as the backbone. Additionally, we continually pre-trained LLaMA-7B with the given corpus D to fully leverage entity semantic information.

Entity Generation. We craft a simple prompt Prompt_g with the given entity set to guide the model toward generating entities semantically similar to the target semantic class. The full prompt can be found in supplementary notes. In the first round of generation, we randomly select 3 entities from positive seed entities S^{pos} as the input entity set. In subsequent rounds, to maintain diversity while ensuring the semantic does not deviate from original positive seed entities, 2 entities and 1 entity are randomly sampled from S^{pos} and current expansion results, respectively.

Vanilla decoding strategies like beam search [34] might uncontrollably generate entities that are not included in the candidate entities. To address this issue, a prefix tree structure is constructed to constrain the decoding process. In this structure, the root node represents the beginning token of an entity, and each path from the root to a leaf node represents a complete candidate entity. During decoding, the process must follow a specific path from root to leaf at one time. In other words, the child nodes of each node represent the tokens allowed for subsequent generation. By employing this prefix-constrained beam search, we ensure the generated entities remain valid.

Entity Selection. For each generated entity e, we calculate the positive similarity score sco^{pos} between it and positive seed entities S^{pos}, which is formulated as follows:

$$sco^{pos}(e) = \frac{1}{|S^{pos}|} \sum_{e' \in S^{pos}} \sqrt[|e'|]{P(e'|f(e))}, \tag{8}$$

in which $f(e)$ is the template "{e} is similar to". $P(e'|f(e))$ represents the conditional probability of LLM generating e' when given $f(e)$. We use geometric mean to balance the various token length $|e'|$ of entities. Top-p entities with the highest positive similarity scores are incorporated into the current expansion result.

Entity Re-ranking. The process of reranking in GenExpan is identical to that in RetExpan except for the calculation of sco^{neg}.

5.2.2. Enhancement Strategy 2: Chain-of-thought Reasoning. Inspired by [35], we enhance the reasoning capability of LLMs for Ultra-ESE through a series of intermediate reasoning steps. Basically, the positive seed entities S^{pos} in GenExpan are directly utilized to construct Prompt_g for generating new entities. However, in chain-of-thought reasoning, we prompt the LLM to initially generate potential fine-grained class names of the positive seed entities, along with the positive attributes that share the same value. This information is then integrated into Prompt_g to guide subsequent entity generation. The chain-of-thought reasoning stimulates the reasoning capability of LLM and benefits it for perceiving the semantic classes of seed entities and the concerned attributes.

6. Experiments

6.1. Experiment Setup

Compared Methods. We compare with three categories of models. The initial category encompasses traditional statistical probability-based methods, notably **SetExpan** [18] and **CasE** [36]. The second category comprises methods based on the pre-trained language model BERT, including **CGExpan** and **ProbExpan**, with ProbExpan representing the prior state-of-the-art method. Lastly, the third category involves methods based on the generative LLM, **GPT-4**. We devised prompt templates incorporating both positive and negative seed entities to guide the model to generate target entities.

Evaluation Metrics. The primary objective of ultra-fine-grained ESE is to expand the entity list in descending order based on their similarity with positive seed entities. Following prior research, PosMAP@K and PosP@K are

TABLE 2. WE REPORT THE MAP AND P SCORES CORRESPONDING TO POS/NEG/COMB ON ULTRAWIKI. THE HIGHEST SCORE IS BOLDED.

Metric Type	Method Type	Method	MAP				P				Avg
			@10	@20	@50	@100	@10	@20	@50	@100	
Pos ↑	Probability Based	SetExpan	13.41	11.83	10.66	10.79	20.10	19.97	21.88	26.39	16.88
		CaSE	16.72	13.74	11.60	10.91	24.58	23.28	26.86	30.53	19.78
	Retrieval Based	CGExpan	21.64	19.72	19.11	20.22	30.61	31.24	38.39	50.03	28.87
		ProbExpan	21.86	22.11	22.80	23.89	38.08	39.41	47.02	62.71	34.74
	Generation Based	GPT4	37.20	35.37	35.49	35.59	47.12	48.87	55.31	62.22	44.65
	Retrieval Based (Ours)	RetExpan	41.73	39.53	38.55	39.91	54.58	58.03	66.76	77.23	52.04
		RetExpan + Contrast	47.45	44.68	43.63	**44.20**	59.83	62.02	**69.36**	**77.92**	**56.14**
	Generation Based (Ours)	GenExpan	46.79	45.00	42.89	40.80	59.77	62.15	66.26	66.57	53.78
		GenExpan + CoT	**50.39**	**47.80**	**43.67**	40.06	**62.74**	**64.45**	64.06	60.38	54.19
Neg ↓	Probability Based	SetExpan	4.06	3.77	3.71	4.20	7.66	8.10	10.92	17.44	7.48
		CaSE	5.33	4.32	3.63	3.50	10.22	10.10	12.79	15.96	8.23
	Retrieval Based	CGExpan	6.15	6.54	8.03	9.96	12.29	16.37	27.38	41.72	16.06
		ProbExpan	6.72	8.16	10.85	13.47	15.12	19.92	34.51	56.48	20.65
	Generation Based	GPT4	6.04	6.61	8.03	8.35	10.40	15.06	24.57	33.63	14.09
	Retrieval Based (Ours)	RetExpan	8.77	9.04	10.65	13.29	16.44	21.04	34.78	56.54	21.32
		RetExpan + Contrast	8.02	8.98	10.89	13.05	14.83	21.23	35.47	55.12	20.95
	Generation Based (Ours)	GenExpan	7.25	8.28	8.72	8.01	15.21	21.31	27.33	28.58	15.59
		GenExpan + CoT	7.79	9.29	8.15	6.89	15.97	22.66	23.52	21.90	14.52
Comb ↑	Probability Based	SetExpan	54.67	54.03	53.48	53.30	56.22	55.93	55.48	54.48	54.70
		CaSE	55.69	54.71	53.99	53.70	57.18	56.59	57.03	57.28	55.77
	Retrieval Based	CGExpan	57.75	56.59	55.54	55.13	59.16	57.44	55.50	54.15	56.41
		ProbExpan	57.57	56.98	55.97	55.21	61.48	59.75	56.25	53.12	57.04
	Generation Based	GPT4	65.58	64.38	63.73	63.62	68.36	66.90	65.37	64.29	65.28
	Retrieval Based (Ours)	RetExpan	66.48	65.25	63.95	63.31	69.07	68.50	65.99	60.34	65.36
		RetExpan + Contrast	69.72	67.85	66.37	65.57	72.50	70.39	66.95	61.40	67.59
	Generation Based (Ours)	GenExpan	69.77	68.36	67.08	66.40	72.28	70.42	69.47	68.99	69.10
		GenExpan + CoT	**71.30**	**69.25**	**67.76**	**66.58**	**73.39**	**70.90**	**70.27**	**69.24**	**69.84**

employed as evaluation metrics. PosMAP@K is computed as follows:

$$\text{PosMAP@}K = \frac{1}{|Q|} \sum_{q \in Q} \text{PosAP}_K \left(L_q, \mathcal{P}\right), \tag{9}$$

where Q stands for the set of all queries, $\text{PosAP}_K \left(L_q, \mathcal{P}\right)$ denotes the average precision at position K with the ranked list L_q and ground-truth list $\mathcal{P}$. PosP@K is the precision of the top-K entities.

Notably, another crucial objective of ultra-fine-grain ESE is to prevent the intrusion of negative entities in the candidate entity list that satisfy the negative attribute (entities in $\mathcal{N}$). Hence, we symmetrically define NegMAP@K and NegP@K, whereby computing them only requires substituting $\mathcal{P}$ with $\mathcal{N}$. A well-built model should also keep negative metrics as low as possible. The combined metrics $\text{CombMAP@}K = (\text{PosMAP@}K + 100 - \text{NegMAP@}K)/2$, and $\text{CombP@}K = (\text{PosP@}K + 100 - \text{NegP@}K)/2$ are also calculated to comprehensively reflect model capacities. These combined metrics are normalized to range from 0 to 1, with higher values indicating better performance. In summary, our evaluation metric can be expressed as $xy@K$, where x={Pos, Neg, Comb}, y={MAP, P}, K={10, 20, 50, 100}.

6.2. Main Experiments

The results of the main experiment are presented in Table 2, from which we can observe that:

(1) Regarding the average values of the comprehensive metrics CombMAP and CombP, our proposed GenExpan achieves optimal and sub-optimal performances with the incorporation of the chain-of-thought reasoning strategy and the entity-based retrieval augmentation strategy, respectively. Although the backbone model of GenExpan, LLaMA, has only 7b parameters, it surpasses the GPT-4, which comprises more than 1T parameters (over 200 times larger than the GenExpan). We attribute this to two factors. Firstly, our LLaMA undergoes further pretraining on the provided corpus focusing on entities, thereby reinforcing the model's ultra-fine-grained semantic understanding of entities. Secondly, GenExpan ensures that the generated entities belong to the candidate entities by constraining the decoding process through the prefix tree. This mitigates the negative impact of introducing irrelevant entities on PosMAP and PosP. It can also be seen from the results that GenExpan outperforms GPT-4 mainly in the Pos class metrics. This constraint strategy is particularly valuable for ESE in user-customized scenarios. It ensures that the entities output by GenExpan are within user's focus domain.

(2) Our proposed RetExpan model also consistently outperforms the leading retrieval-based model, ProbExpan, and the advanced generation-based model, GPT-4, as measured

by the Comb metrics. RetExpan and ProbExpan are similar in their overall frameworks but have a large performance gap. The primary reason for this difference is that RetExpan utilizes the hidden state of the trained BERT for entity representation, while ProbExpan relies on the probability distribution of candidate entities at the [MASK] token. We believe that the hidden state, as a continuous vector in the feature space, captures the semantics of entities with finer granularity. In contrast, the probability distribution, as a discrete metric in the probability space, inherently offers relatively coarser granularity due to its limited capacity to store entity information. Consequently, ProbExpan underperforms in Ultra-ESE. This insight prompts future research into devising better representations to express the ultra-fine-grained semantics of entities, such as decoupling the base semantics of entities from the ultra-fine-grained attribute semantics, similar to the Mix-of-Expert (MoE) approach, where distinct features represent different perspectives of the semantics.

The ablation experiments presented in Table 3 further validate our hypothesis regarding the factors why RetExpan and GenExpan surpass the previous state-of-the-art models. Both methods exhibit remarkable degradation after removing the corresponding modules.

(3) Comparing the two enhancement strategies, ultra-fine-grained contrastive learning provides the most substantial improvement in the Pos class metrics, averaging 4.10 points (from 52.04 to 56.14). Conversely, contrastive learning brings smaller gains in Neg class metrics. This discrepancy is primarily attributed to the implementation of contrastive learning, where both positive and negative target entities are pulled away from normal negative entities originating from other semantic classes to ensure underlying semantics. This dilutes the pulling away of positive and negative target entities from each other, as they are both in the denominator of the contrastive loss and are assigned the same weight.

Furthermore, we found that directly increasing the weights of negative terms formed by positive and negative target entities is ineffective. This is because positive and negative target entities are determined by GPT-4 and inevitably contain errors. Therefore, this inspires us to devise more precise contrastive data mining methods in the future to amplify the penalty for hard negative terms formed by positive and negative target entities.

(4) The conventional probability-based methods SetExpan and CaSE attain quite low scores on both NegMAP and NegP metrics, which indicates that negative target entities are not excessively introduced. However, this does not imply that these methods exhibit a high degree of negative semantic awareness. The Pos class metrics of SetExpan and CaSE are low in similarity, indicating their limited understanding of fine-grained semantic classes overall. Consequently, they struggle to recall entities with the given seed entities, leading to low scores in both Pos and Neg class metrics. We further evaluated the MAP@100 of CaSE at the fine-grained semantic class level, which only reaches 21.43, significantly lower than other methods such as 82.08 of RetExpan.

TABLE 3. ABLATION EXPERIMENTS FOR EACH MODULE OF RETEXPAN AND GENEXPAN. EACH MODULE CONTRIBUTES TO THE OVERALL PERFORMANCE.

Method	MAP				Avg
	@10	@20	@50	@100	
RetExpan	66.48	65.25	63.95	63.31	64.75
- Entity prediction	63.94	62.27	61.32	60.48	62.00
GenExpan	69.79	68.35	67.07	66.38	67.90
- Prefix constrain	57.03	56.64	56.33	56.1	56.53
- Further pretrain	68.58	66.67	65.23	64.23	66.18

(5) Without further fine-tuning and modifications to the model structure, GPT-4 achieves excellent results, which proves that generative LLMs have great potential for addressing Ultra-ESE tasks. However, GPT-4 still fails to outperform the GenExpan, which is also generative but based on smaller LLaMA-7B. Analysis of cases reveals that there are two primary problems with GPT-4. On the one hand, it performs poorly on long-tail problems, such as U.S. national monuments and mobile phone brands, which contain a considerable number of low-frequency entities with limited information available on the Internet. GPT-4 only achieves single-digit PosMAP on these semantic classes. In contrast, GenExpan performs better, benefiting from the given contextual corpus. On the other hand, GPT-4 is prone to haphazardly generate non-existent entities (e.g., fake mobile phone brands), which is referred to hallucination problem in recent work[37, 38]. We are currently unable to solve this issue by simple output post-processing. While the results of GPT-4 are impressive, there remains a lot of space for improvement.

6.3. Analysis of Negative Entities

We conducted analytical experiments from various perspectives regarding negative seed entities to answer the following questions:

Whether the introduction of negative seed entities is effective for representing ultra-fine-grained semantic classes? The negative seed entity-based entity re-ranking modules were removed from RetExpan and GenExpan. Thanks to the high scalability, it was also integrated into ProbExpan. Results of the three comparison experiments are presented in Table 4. We can observe that:

(1) After discarding the input of negative samples, RetExpan and GenExpan show a consistent decrease and increase in Pos class metrics and Neg class metrics, respectively. ProbExpan, on the other hand, demonstrates a rise of 0.33 points on average in the Comb class metrics after equipping the entity re-ranking module. This strongly suggests that the introduction of negative seed entities effectively mitigates the intrusion of negative target entities while raising the ranking of positive target entities, i.e., portraying the semantic classes at a finer granularity.

(2) Further analyzing the Pos and Neg metrics, we observe that the addition of entity re-ranking module to

TABLE 4. ABLATION EXPERIMENTS ON ENTITY RE-RANKING MODULE WITH NEGATIVE SEED ENTITIES.

Method	Metric Type	MAP				P				Avg
		@10	@20	@50	@100	@10	@20	@50	@100	
ProbExpan	Pos	21.86	22.11	22.80	23.89	38.08	39.41	47.02	62.71	34.74
	Neg	6.72	8.16	10.85	13.47	15.12	19.92	34.51	56.48	20.65
	Comb	57.57	56.98	55.97	55.21	61.48	59.75	56.25	53.12	57.04
+ Neg Rerank	Pos	23.63	23.65	24.13	25.20	37.36	39.41	46.79	62.71	35.36
	Neg	6.82	8.22	10.89	13.47	14.79	19.92	34.48	56.48	20.63
	Comb	58.41	57.72	56.62	55.87	61.29	59.75	56.20	53.12	57.37
△	Pos	1.77	1.54	1.33	1.32	-0.72	0.00	-0.23	0.00	0.63
	Neg	0.10	0.07	0.04	0.00	-0.33	0.00	-0.13	0.00	-0.03
	Comb	0.83	0.74	0.64	0.66	-0.19	0.00	-0.05	0.00	0.33
RetExpan (Ours)	Pos	41.73	39.53	38.55	39.91	54.58	58.03	66.76	77.23	52.04
	Neg	8.77	9.04	10.65	13.29	16.44	21.04	34.78	56.54	21.32
	Comb	66.48	65.25	63.95	63.31	69.07	68.50	65.99	60.34	65.36
- Neg Rerank	Pos	40.39	38.33	37.31	38.70	54.09	58.03	66.62	77.23	51.34
	Neg	9.31	9.68	11.44	13.99	16.81	21.04	35.53	56.54	21.79
	Comb	65.54	64.33	62.94	62.36	68.64	68.50	65.55	60.34	64.78
△	Pos	-1.34	-1.20	-1.24	-1.21	-0.49	0.00	-0.14	0.00	-0.70
	Neg	0.54	0.64	0.79	0.70	0.37	0.00	0.75	0.00	0.47
	Comb	-0.94	-0.92	-1.01	-0.95	-0.43	0.00	-0.44	0.00	-0.59
GenExpan (Ours)	Pos	46.79	45.00	42.89	40.80	59.77	62.15	66.26	66.57	53.78
	Neg	7.25	8.28	8.72	8.01	15.21	21.31	27.33	28.58	15.59
	Comb	69.77	68.36	67.08	66.40	72.28	70.42	69.47	68.99	69.10
-Neg Rerank	Pos	46.08	44.31	42.17	40.10	59.26	62.15	66.17	66.57	53.35
	Neg	7.85	8.71	9.20	8.45	15.70	21.31	27.44	28.58	15.90
	Comb	69.12	67.80	66.49	65.83	71.78	70.42	69.37	68.99	68.98
△	Pos	-0.71	-0.69	-0.72	-0.70	-0.51	0.00	-0.09	0.00	-0.43
	Neg	0.60	0.43	0.48	0.44	0.49	0.00	0.11	0.00	0.32
	Comb	-0.65	-0.56	-0.59	-0.57	-0.50	0.00	-0.10	0.00	-0.37

ProbExpan conversely resulted in a notable decline in PosP metrics. It's essential to clarify the distinction between MAP and P metrics: P@K is concerned solely with number of target entities in the top-K entity list, regardless of their positions, while MAP@K is rank-aware, with higher values indicating that target entities are positioned closer to the top of the list. Consequently, this indicates that the incorporation of negative seed entities in ProbExpan makes a mixed impact, propelling some positive target entities to higher ranks and simultaneously causing others to be ranked lower.

(3) For RetExpan, negative seed entities have a similar impact on metrics with different K values, whereas for GenExpan, negative seed entities have a greater impact on metrics with a smaller K. This may be due to the fact that RetExpan is a one-time expansion [36, 39, 40] model and GenExpan is an iterative expansion [18, 41, 42] model, so entities expanded later (corresponding to a larger K) are more likely to deviate from the original ground-truth semantic class. During these stages, re-ranking using negative seed entities also doesn't work. Addressing the issue of semantic drift [25, 27, 43] in iterative expansion models like GenExpan is a longstanding challenge in the ESE field.

What are the roles played by negative seed entities? We first assessed the efficacy of RetExpan and its enhancement strategies by comparing scenarios where positive and negative attributes are identical or different. As shown in Table 5, we can find that:

(1) When positive and negative attributes are identical, there is no overlap between positive and negative target entities, which simplifies semantic classes and yielding higher performance. In such cases, negative seed entities play a supportive role, primarily emphasizing user-concerned attributes and mitigating ambiguity. Conversely, when positive and negative attributes are different, the potential overlap between entities satisfying these attributes poses a significant challenge to semantic understanding, leading to lower overall performance. Here, negative seed entities play a crucial role in conveying unwanted semantics and are indispensable.

(2) The benefits derived from contrastive learning are more pronounced in semantic classes where positive and negative attributes are identical. This is because, in such scenarios, there is no overlap between positive and negative

TABLE 5. COMPARISON EXPERIMENTS WHEN POSITIVE AND NEGATIVE ATTRIBUTES ARE THE SAME AND DIFFERENT.

Method	Metric Type	MAP				Avg
		@10	@20	@50	@100	
$\mathcal{A}^{\mathbf{pos}} = \mathcal{A}^{\mathbf{neg}}$						
RetExpan	Pos	43.14	41.74	41.93	42.89	42.43
	Neg	5.54	6.44	8.71	11.56	8.06
	Comb	68.80	67.65	66.61	65.67	67.18
RetExpan + Contrast	Pos	49.68	48.73	48.77	49.50	49.17
	Neg	5.76	6.51	8.64	11.62	8.13
	Comb	71.96	71.11	70.06	68.94	70.52
$\mathcal{A}^{\mathbf{pos}} \neq \mathcal{A}^{\mathbf{neg}}$						
RetExpan	Pos	41.73	39.53	38.55	39.91	39.93
	Neg	8.77	9.04	10.65	13.29	10.44
	Comb	66.48	65.25	63.95	63.31	64.75
RetExpan + Contrast	Pos	46.44	43.69	42.51	43.14	43.95
	Neg	8.22	9.25	11.17	13.38	10.51
	Comb	69.11	67.22	65.67	64.88	66.72

TABLE 6. COMPARISON EXPERIMENTS ON SEMANTIC CLASSES WITH DIFFERENT NUMBERS OF POSITIVE AND NEGATIVE ATTRIBUTES.

$(\|\mathcal{A}^{pos}\|, \|\mathcal{A}^{neg}\|)$	Metric Type	MAP				Avg
		@10	@20	@50	@100	
(1, 1)	Pos	41.32	39.18	38.36	39.58	39.61
	Neg	8.61	9.14	10.88	13.43	10.52
	Comb	66.36	65.02	63.74	63.07	64.55
(1, 2)	Pos	39.76	37.65	41.49	43.44	40.59
	Neg	0.47	0.98	2.44	5.57	2.37
	Comb	69.64	68.33	69.52	68.93	69.11
(2, 1)	Pos	38.84	32.85	36.39	41.33	37.35
	Neg	2.39	3.29	5.38	9.81	5.22
	Comb	68.22	64.78	65.51	65.76	66.07

target entities, making the contrastive training pairs mined based on the similarity of positive and negative seed entities more reliable.

(3) When positive and negative attributes differ, the primary focus lies in evaluating the model's capability to comprehend the semantics expressed by negative seed entities. We observe that contrastive learning demonstrates less effective in this situation. This phenomenon suggests that the understanding of negative seed entities demands more additional knowledge compared to positive seed classes. This notion is further verified in the analysis of chain-of-thought reasoning in Section 6.4.

What are the differences in challenges posed by semantic classes with varying numbers of positive and negative attributes? We analyze the performance of RetExpan on semantic classes characterized by varying quantities of positive and negative attributes in Table 6. To maintain control over variables, we exclusively present semantic classes for three combinations of attribute quantity: (1,1), (1,2), and (2,1). Imposing stricter constraints on the attributes results in a reduction in the number of associated target entities, rendering it more challenging to discern these entities accurately. Consequently, a decrease in the Pos (Neg) metrics is observed, correlating with an increase in the number of positive (negative) attributes. It's important to emphasize that although the Neg metrics decrease with more negative attribute constraints, this does not imply a reduced difficulty in distinguishing negative target entities. Rather, identifying these negative target entities requires more comprehensive attribute information about negative seed entities.

6.4. Analysis of Enhancement Strategy

6.4.1. Analysis of Contrastive Learning. We conducted ablation experiments on each part of the training data for ultra-fine-grained contrastive learning to explore their impact on overall performance. As illustrated in Table 7, based on the full training data, the last three rows sequentially remove the hard negative samples comprising pairs of positive and negative target entities, the normal negative samples consisting of pairs of positive and negative target entities along with entities from other semantic classes, and the positive samples composed of entity pairs within the same ultra-fine-grained semantic class.

TABLE 7. ABLATION EXPERIMENTS ON CONTRASTIVE LEARNING.

Method	Metric Type	MAP				Avg
		@10	@20	@50	@100	
RetExpan	Pos	41.73	39.53	38.55	39.91	39.93
	Neg	8.77	9.04	10.65	13.29	10.43
	Comb	66.48	65.25	63.95	63.31	64.75
RetExpan +Contrast	Pos	47.45	44.68	43.63	44.20	44.99
	Neg	8.02	8.98	10.89	13.05	10.24
	Comb	69.72	67.85	66.37	65.57	67.38
- Neg from (L_{pos}, L_{neg})	Pos	46.34	43.60	42.69	43.36	43.99
	Neg	8.71	9.23	11.16	13.30	10.60
	Comb	68.82	67.18	65.76	65.03	66.70
- Neg from $(L_{pos}, \overline{L_0})$ & $(L_{neg}, \overline{L_0})$	Pos	46.89	44.14	43.04	43.86	44.48
	Neg	8.42	9.00	11.14	13.26	10.46
	Comb	69.24	67.57	65.95	65.30	67.01
- Pos from (L_{pos}, L_{pos}) & (L_{neg}, L_{neg})	Pos	46.95	43.55	42.77	43.36	44.16
	Neg	8.531	9.103	10.996	13.185	10.45
	Comb	69.211	67.221	65.8875	65.0875	66.85

It can be seen that hard negative samples consisting of pairs of positive and negative target entities contribute the most to the overall performance, which is in line with our intuition of proposing ultra-fine-grained contrastive learning. We set both $|L_{pos}|$ and $|L_{neg}|$ to 10 to mitigate the introduction of excessive noise. Consequently, it's evident that this part of training data wields more influence when K is small. Exploring how to mitigate the noise introduced by automatic GPT-4 labeling when $|L_{pos}|$ and $|L_{neg}|$ increase is a promising direction for future exploration.

Additionally, we observe that normal negative samples boost the overall performance. They ensure that positive and negative target samples do not deviate excessively from each other at the fine-grained semantic level, thereby preventing semantic collapse. Nevertheless, the integration of normal negative samples compromises the efficacy of ultra-fine-grained (hard) negative samples. This arises from the fact that the presence of normal negative samples indirectly diminishes the penalization applied to the distance of hard negative samples by the loss function, as they are all in the denominator of the loss function and have the same weights. Specifically, the incorporation of normal negative samples diminishes the average enhancement of hard negative samples on the Comb metric from 67.01 - 64.75 = 2.26 (line 4 minus line 1) to 67.38 - 66.70 = 0.68 (line 2 minus line 3). Moreover, positive samples, composed of entities belonging to the same ultra-fine-grained semantic class, similarly contribute positively to the overall performance.

6.4.2. Analysis of Chain-of-thought Reasoning. Experiments in Table 8 examine the chain-of-thought with varying reasoning depths and assess the impact of reasoning precision on model performance. From the results, we observe that: (1) Deeper reasoning proves to be more advantageous for Ultra-ESE, as it facilitates a more explicit understanding of the implicit ultra-fine-grained semantics of seed entities.

(2) Surprisingly, the use of manually labeled ground-truth class names is not superior to those derived through

TABLE 8. COMPARISON EXPERIMENTS ON CHAIN-OF-THOUGHT REASONING. THE IMPACT OF REASONING DEPTH AND REASONING PRECISION ON PERFORMANCE IS INVESTIGATED. GT: GROUND-TRUTH. GEN: GENERATED. CN: CLASS NAME. POS/NEG: POSITIVE/NEGATIVE ATTRIBUTE INFORMATION.

Method	Metric Type	MAP				Avg
		@10	@20	@50	@100	
GenExpan	Pos	46.84	44.99	42.88	40.79	43.88
	Neg	7.27	8.29	8.74	8.02	8.08
	Comb	69.79	68.35	67.07	66.38	67.90
GenExpan + CoT (GT CN)	Pos	49.20	46.84	43.67	41.04	45.19
	Neg	7.43	9.15	9.17	8.36	8.53
	Comb	70.89	68.84	67.25	66.34	68.33
GenExpan + CoT (Gen CN)	Pos	49.33	46.97	43.51	40.82	45.15
	Neg	7.37	8.97	8.74	7.89	8.24
	Comb	70.98	69.00	67.38	66.46	68.46
GenExpan + CoT (Cen CN & Gen Pos)	Pos	50.39	47.80	43.67	40.06	45.48
	Neg	7.79	9.29	8.15	6.89	8.03
	Comb	71.30	69.25	67.76	66.58	68.72
GenExpan + CoT (Cen CN & GT Pos)	Pos	50.77	48.25	44.38	40.67	46.02
	Neg	7.68	9.16	8.12	6.83	7.95
	Comb	71.55	69.54	68.13	66.92	69.04
GenExpan + CoT (Cen CN & Gen Pos & Gen Neg)	Pos	49.78	47.24	42.68	39.01	44.68
	Neg	7.60	8.97	7.98	6.75	7.82
	Comb	71.09	69.14	67.35	66.13	68.43
GenExpan + CoT (Cen CN & GT Pos & GT Neg)	Pos	51.02	49.19	44.86	41.40	46.62
	Neg	7.30	8.70	7.59	6.53	7.53
	Comb	71.86	70.25	68.63	67.43	69.54

reasoning with the LLaMA-7B. The case study of the generated class names reveals that LLaMA-7B can produce class names encapsulating positive attribute information based on the given positive seed entities. For instance, when given entities about U.S. airports with the "location" attribute, GenExpan accurately infers more specific semantic classes that reflect positive attribute information, e.g., "Airports in Michigan". This suggests that reasoning results are not necessarily inferior to manually annotated "ground-truth" labels, as chain-of-thought reasoning induces LLMs to attend to details that human ignores.

(3) Incorporating negative attribute information obtained from LLaMA-7B's reasoning leads to a decline in model performance. Reasoning about negative attributes poses greater challenges than positive attributes due to constraints from two aspects: first, the negative seed entities are identical on negative attributes; second, the positive seed entities pose different values on negative attributes compared to the negative seed entities. Our analysis reveals that the current LLaMA-7B model struggles to deduce the relevant negative attributes by comparing positive and negative seed entities. This difficulty is particularly pronounced for long-tailed entities (such as ancient musical instruments) and semantic classes with diverse attributes, which indicates that enhancing the ultra-fine-grained perception of entities within LLMs is a promising avenue for future research.

6.5. Explore Experiments on Different Paradigm Model Interactions

We are curious whether retrieval-based and generation-based frameworks can interact and mutually reinforce each other. Actually, in the process of enhancing the retrieval-based RetExpan, we have leveraged the generative LLMs GPT-4. At the same time, when enhancing the generation-based GenExpan, we have utilized the retrieval-based approach to retrieve extra entity knowledge. In this section, we explore the facilitative relationship between the two paradigms at a higher level. Specifically, for the A+B model in Table 9, we first allow Model A to obtain 1000 entities from the complete candidate entities, which guarantees high entity recall. Then, model B executes on the obtained 1000 entities. To our surprise, no matter whether RetExpan or GenExpan is deployed first, both of them enhance the performance on all metrics considerably. These two different paradigm frameworks have their own advantages. The retrieval-based framework measures the semantic relation between candidate and seed entities by feature similarity, which inherently fits the nature of the retrieval task. Thus even when utilizing smaller models, retrieval-based frameworks can achieve competitive results, and be easily transferred to specific domains. Generation-based frameworks, on the other hand, typically rely on LLMs, which offer a broader knowledge base and greater semantic awareness, but the LLMs tend to be less efficient. Therefore, how to make the two types of models learn collaboratively, promote each other, and then achieve automatic evolution is a promising direction for solving Ultra-ESE.

TABLE 9. EXPERIMENTS ON THE INTERACTION OF RETEXPAN AND GENEXPAN.

Method	Metric Type	MAP				Avg
		@10	@20	@50	@100	
RetExpan	Pos	41.73	39.53	38.55	39.91	39.93
	Neg	8.77	9.04	10.65	13.29	10.44
	Comb	66.48	65.25	63.95	63.31	64.75
RetExpan + GenExpan	Pos	45.93	44.37	42.96	42.35	43.90
	Neg	7.03	7.51	9.04	9.48	8.27
	Comb	69.45	68.43	66.96	66.43	67.82
GenExpan	Pos	46.84	44.99	42.88	40.79	43.88
	Neg	7.27	8.29	8.74	8.02	8.08
	Comb	69.79	68.35	67.07	66.38	67.90
GenExpan + RetExpan	Pos	48.91	46.09	44.13	42.49	45.41
	Neg	6.24	6.80	7.56	7.84	7.11
	Comb	71.33	69.64	68.29	67.32	69.15

7. Conclusion

In conclusion, our study addresses the challenge of ultra-fine-grained Entity Set Expansion (Ultra-ESE) by introducing negative seed entities alongside positive ones, mitigating ambiguity and facilitating the expression of "unwanted" attributes. The creation of UltraWiki, a tailored dataset for Ultra-ESE, enables rigorous evaluation and future research. Through RetExpan and GenExpan frameworks, we demonstrate the effectiveness of large language models in Ultra-ESE from retrieval-based and generation-based perspectives. Our proposed strategies enhance models' comprehension of ultra-fine-grained entity semantics. While our findings are promising, they also highlight the substantial scope for improvement in Ultra-ESE, advancing the field and paving the way for future exploration.

Acknowledgement

This research is supported by National Natural Science Foundation of China (Grant No.62276154), Research Center for Computer Network(Shenzhen)Ministry of Education, the Natural Science Foundation of Guangdong Province (Grant No.2023A1515012914), Basic Research Fund of Shenzhen City (Grant No.JCYJ20210324120012033 and JSGG20210802154402007), the Major Key Project of PCL for Experiments and Applications (PCL2023A09), and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (HW2021008).

References

[1] Y. Li, S. Huang, X. Zhang, Q. Zhou, Y. Li, R. Liu, Y. Cao, H.-T. Zheng, and Y. Shen, "Automatic context pattern generation for entity set expansion," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12 458–12 469, 2023.

[2] Y. Zhang, J. Shen, J. Shang, and J. Han, "Empower entity set expansion via language model probing," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8151–8160.

[3] Y. Li, Y. Li, Y. He, T. Yu, Y. Shen, and H.-T. Zheng, "Contrastive learning with hard negative entities for entity set expansion," in Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1077–1086.

[4] Y. Li, T. Lu, H.-T. Zheng, Y. Li, S. Huang, T. Yu, J. Yuan, and R. Zhang, "Mesed: A multi-modal entity set expansion dataset with fine-grained semantic classes and hard negative entities," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 8, 2024, pp. 8697–8706.

[5] Y. Wang, H. Huang, and C. Feng, "Query expansion with local conceptual word embeddings in microblog retrieval," IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1737–1749, 2019.

[6] S. Seo, B. Oh, E. Jo, S. Lee, D. Lee, K.-H. Lee, D. Shin, and Y. Lee, "Active learning for knowledge graph schema expansion," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 12, pp. 5610–5620, 2021.

[7] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using neighborhood-inflated seed expansion," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 5, pp. 1272–1284, 2016.

[8] S. Pei, L. Yu, and X. Zhang, "Set-aware entity synonym discovery with flexible receptive fields," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 1, pp. 891–904, 2021.

[9] R. Kohita, I. Yoshida, H. Kanayama, and T. Nasukawa, "Interactive construction of user-centric dictionary for text analytics," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 789–799.

[10] Z. Huang, B. Cautis, R. Cheng, Y. Zheng, N. Mamoulis, and J. Yan, "Entity-based query recommendation for long-tail queries," ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 12, no. 6, pp. 1–24, 2018.

[11] G. Jacucci, P. Daee, T. Vuong, S. Andolina, K. Klouche, M. Sjöberg, T. Ruotsalo, and S. Kaski, "Entity recommendation for everyday digital tasks," ACM Transactions on Computer-Human Interaction (TOCHI), vol. 28, no. 5, pp. 1–41, 2021.

[12] E. Choi, O. Levy, Y. Choi, and L. Zettlemoyer, "Ultra-fine entity typing," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 87–96.

[13] N. Li, Z. Bouraoui, and S. Schockaert, "Ultra-fine entity typing with prior knowledge about labels: A simple clustering based strategy," in Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 11 744–11 756.

[14] T. Komarlu, M. Jiang, X. Wang, and J. Han, "Ontotype: Ontology-guided and pre-trained language model assisted fine-grained entity typing," in Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 1407–1417.

[15] X. Zhao, L. Zhang, Z. Ding, L. Xia, J. Tang, and D. Yin, "Recommendations with negative feedback via pairwise deep reinforcement learning," in Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 1040–1048.

[16] C. Wu, F. Wu, Y. Huang, and X. Xie, "Neural news recommendation with negative feedback," CCF Transactions on Pervasive Computing and Interaction, vol. 2, pp. 178–188, 2020.

[17] R. Xie, C. Ling, Y. Wang, R. Wang, F. Xia, and L. Lin, "Deep feedback network for recommendation," in Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 2519–2525.

[18] J. Shen, Z. Wu, D. Lei, J. Shang, X. Ren, and J. Han, "Setexpan: Corpus-based set expansion via context feature selection and rank ensemble," in Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10. Springer, 2017, pp. 288–304.

[19] S. Pradhan, A. Moschitti, N. Xue, H. T. Ng, A. Björkelund, O. Uryupina, Y. Zhang, and Z. Zhong, "Towards robust linguistic analysis using ontonotes," in Proceedings of the Seventeenth Conference on Computational Natural Language Learning, 2013, pp. 143–152.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[21] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar et al., "Llama: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971, 2023.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in International conference on machine learning. PMLR, 2020, pp. 1597–1607.

[23] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," arXiv preprint arXiv:2104.08821, 2021.

[24] J. Xiao, M. Elkaref, N. Herr, G. D. Mel, and J. Han, "Taxonomy-guided fine-grained entity set expansion," in Proceedings of the 2023 SIAM International Conference on Data Mining (SDM). SIAM, 2023, pp. 631–639.

[25] J. R. Curran, T. Murphy, and B. Scholz, "Minimising semantic drift with mutual exclusion bootstrapping," in Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics, vol. 6. Citeseer, 2007, pp. 172–180.

[26] P. Jindal and D. Roth, "Learning from negative examples in set-expansion," in 2011 IEEE 11th International Conference on Data Mining. IEEE, 2011, pp. 1110–1115.

[27] B. Shi, Z. Zhang, L. Sun, and X. Han, "A probabilistic co-bootstrapping method for entity set expansion," in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 2280–2290.

[28] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," arXiv preprint cs/0306050, 2003.

[29] J. L. Fleiss, "Measuring nominal scale agreement among many raters." Psychological bulletin, vol. 76, no. 5, p. 378, 1971.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[32] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," in International Conference on Learning Representations (ICLR), 2021.

[33] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," arXiv preprint arXiv:1807.03748, 2018.

[34] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," Advances in neural information processing systems, vol. 27, 2014.

[35] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou et al., "Chain-of-thought prompting elicits reasoning in large language models," Advances in Neural Information Processing Systems, vol. 35, pp. 24 824–24 837, 2022.

[36] P. Yu, Z. Huang, R. Rahimi, and J. Allan, "Corpus-based set expansion with lexical features and distributed representations," in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1153–1156.

[37] Z. Ji, T. Yu, Y. Xu, N. Lee, E. Ishii, and P. Fung, "Towards mitigating llm hallucination via self reflection," in Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 1827–1843.

[38] X. Cheng, Z. Zhu, W. Xu, Y. Li, H. Li, and Y. Zou, "Accelerating multiple intent detection and slot filling via targeted knowledge distillation," in The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.

[39] G. Kushilevitz, S. Markovitch, and Y. Goldberg, "A two-stage masked lm method for term set expansion," arXiv preprint arXiv:2005.01063, 2020.

[40] J. Mamou, O. Pereg, M. Wasserblat, A. Eirew, Y. Green, S. Guskin, P. Izsak, and D. Korat, "Term set expansion based nlp architect by intel ai lab," arXiv preprint arXiv:1808.08953, 2018.

[41] J. Huang, Y. Xie, Y. Meng, J. Shen, Y. Zhang, and J. Han, "Guiding corpus-based set expansion by auxiliary sets generation and co-expansion," in Proceedings of The Web Conference 2020, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2188–2198. [Online]. Available: https://doi.org/10.1145/3366423.3380284

[42] X. Rong, Z. Chen, Q. Mei, and E. Adar, "Egoset: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion," in Proceedings of the Ninth ACM international conference on Web search and data mining, 2016, pp. 645–654.

[43] T. McIntosh, "Unsupervised discovery of negative categories in lexicon bootstrapping," in Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, 2010, pp. 356–365.

[44] C.-H. Wei, H.-Y. Kao, and Z. Lu, "Gnormplus: an integrative approach for tagging genes, gene families, and protein domains," BioMed research international, vol. 2015, no. 1, p. 918710, 2015.

[45] C.-H. Wei, L. Luo, R. Islamaj, P.-T. Lai, and Z. Lu, "Gnorm2: an improved gene name recognition and normalization system," Bioinformatics, vol. 39, no. 10, p. btad599, 2023.

Appendix A. Prompts

When constructing contrastive learning training data, we use GPT's classification to obtain L_{pos} and L_{neg}. For each entity in the top-T of L_0, determine whether its attributes are consistent with S^{pos} (S^{neg}). If it is consistent, GPT should output 1; otherwise, GPT should output 0. Next, these entities that GPT deems consistent with the attributes of S^{pos} (S^{neg}) will be merged with S^{pos} (S^{neg}) to form L_{pos} (L_{neg}). The prompt to classify entity is shown in Table 12.

In GenExpan, each round uses three example entities to construct $Prompt_g$, which is used to generate entities that are semantically similar to them. $Prompt_g$ is as shown in Table 13.

In the process of enhancing GenExpan with Chain of Thought prompting, we employ LLM to generate fine-grained class names. The prompt, $Prompt_c$, is shown in Table 14.

Appendix B. Implementation Details of RetExpan

In RetExpan, sentences are tokenized using the WordPiece tokenizer and then fed into a 12-layer Transformer initialized with $BERT_{BASE}$ weights. To optimize training efficiency and preserve semantic knowledge learned by BERT, we freeze the first 11 layers of the encode, thus only the last layer is fine-tuned. When tokenizing, to ensure that mentions of entities in long sentences are not truncated, we have implemented entity focus, guaranteeing the presence of entities within the sentence. During expansion, we use one-shot expansion to directly obtain the preliminary expansion results L_0. In RetExpan with contrastive learning strategy, each epoch during training alternates between computing and optimizing entity prediction loss and contrastive learning loss.

To ensure that the encoder acquires knowledge from the corpus, we trained it for 20 epochs on 8 RTX 3090 GPUs. The hyperparameters for training learning rate, batch size, weight decay, label smoothing factor η were set to 4e-5, 128, 1e-2, and 0.075 respectively.

Appendix C. Implementation Details of GenExpan

In GenExpan, we use LLaMA-7b as the base model. Initially, we train for 1 epoch on the corpus using 6 A100 GPUs. The training hyperparameters learning rate, batch size, gradient accumulation steps, weight decay, gradient clipping are set to 1e-5, 4, 8, 1e-4, and 1.0 respectively.

During entity generation in GenExpan, we utilize prefix-constrained beam search with a beam size of 40 to generate 40 entities at a round. For entity selection, we compute the positive similarity score for each entity and select those whose scores are in the top 0.7 as the results for the current round. When no new entity is generated for 20 consecutive rounds, the generation process will end and move to re-ranking.

TABLE 10. TYPES OF ULTRA-FINE-GRAINED SEMANTIC CLASSES. CLS.: SEMANTIC CLASS

$\lvert\mathcal{A}^{pos}\rvert$	$\lvert\mathcal{A}^{neg}\rvert$	#Ultra-fine-grained CLS.
1	1	238
1	2	5
2	1	9
2	2	7
3	3	2

Appendix D. Details of UltraWiki

UltraWiki contains 10 fine-grained semantic classes, which fall under 5 coarse-grained semantic categories: Organization, Location, Product, Person, and Miscellaneous. The number of entities in each fine-grained semantic class ranges from 45 to 952. Each fine-grained semantic class has 2 to 3 attributes, and depending on the combination of attributes, several ultra-fine-grained semantic classes can be derived from a single fine-grained class. Detailed data is displayed in Table 11.

The number of combinations from positive and negative attributes varies significantly, leading to substantial differences in the number of ultra-fine-grained semantic classes. The count of UltraWiki's positive and negative attributes and their corresponding ultra-fine-grained classes are presented in Table 10.

Appendix E. Discussion of Generalization of Dataset Construction Pipeline

UltraWiki is built through a multi-step pipeline involving:

(1) **Identifying fine-grained semantic classes and entities**
(2) **Extracting entity-relevant sentences**
(3) **Annotating entity attributes**
(4) **Generating ultra-fine-grained semantic classes** via positive and negative attribute constraints

While our pipeline utilizes Wikipedia's structured data (e.g., entity lists, hyperlinks, and Wikidata properties), its core principle, leveraging attribute constraints to define ultra-fine-grained entity sets, is domain-agnostic. The key steps (finding entities, tagging them in context, characterizing their attributes, and forming semantic classes) can be implemented using alternative techniques to replace Wikipedia's built-in structure. For instance, in biomedicine, a similar Ultra-ESE dataset can roughly be constructed as follows:

TABLE 11. FINE-GRAINED SEMANTIC CLASSES DETAIL. CLS.: SEMANTIC CLASS

Coarse CLS.	Fine-grained CLS.	#Entities	#Ultra-fine-grained CLS.	Attributes
Organization	Canada universities	99	10	<Loc-Province>, <Type>
Location	China cities	675	50	<Province>, <Prefecture>
	Countries	190	68	<Continent>, <Driving-Side>, <Per-Capita-Income>
	US airports	370	74	<Role>, <Loc-State>
	US national monuments	112	12	<Loc-State>, <Agency>
Product	Mobile phone brands	159	7	<Loc-Continent>, <Status>
	Percussion instruments	128	10	<Type>, <Source-Continent>
Person	Nobel laureates	952	11	<Prize>, <Gender>
	US presidents	45	5	<Party>, <Birth-State>
Miscellaneous	Chemical elements	118	14	<Period>, <Phase-at-R.T.>

TABLE 12. THE PROMPT USED TO SELECT THE ENTITIES THAT ARE CONSISTENT WITH THE ATTRIBUTES OF S^{pos} (S^{neg}) IN THE TOP-T ENTITIES OF L_0 TO CONSTRUCT L_{pos} (L_{neg}).

I have a task that involves classifying candidate entities based on their alignment with a seed entity set. The seed entities are grouped together because they share certain attributes, referred to as seed attributes. I will provide a list of seed entities along with their seed attributes. Additionally, I have a list of candidate entities that are similar to the seed entities but may not necessarily share the same seed attributes. I need you to identify the seed attributes and use them to classify each candidate entity into one of two categories: 1) consistent with the seed entity set in terms of seed attributes, or 0) inconsistent with the seed entity set in terms of seed attributes. For the given N candidate entities, please output N values, each being 1 or 0, indicating whether each candidate is consistent (1) or inconsistent (0) with the seed entity set based on the seed attributes.

Input:
Seed entities: [Mark Twain, Ernest Hemingway, F. Scott Fitzgerald]
Candidate entities: [J.K. Rowling, Stephen King, Agatha Christie, John Steinbeck, Harper Lee, Charles Dickens, Virginia Woolf], total 7 entities
Output:
"result": [0,1,0,1,1,0,0]

Input:
Seed entities: [Golden Retriever, German Shepherd, Labrador Retriever]
Candidate entities: [Bengal Tiger, Beagle, Siberian Husky, African Elephant, Pug], total 5 entities
Output:
"result": [0,1,1,0,1]

Input:
Seed entities: [{Entity1}, {Entity2}, {Entity3}]
Candidate entities: [{Entity1'}, {Entity2'}, {Entity3'}, ...], total {} entities
Output:

TABLE 13. THE PROMPT USED TO GENERATE ENTITIES THAT ARE SEMANTICALLY SIMILAR TO 3 GIVEN ENTITIES.

iron, copper, aluminum and zinc.
math, physics, chemistry and biology.
{Entity1}, {Entity2}, {Entity3} and _____

(1) **Identifying Fine-Grained Entity Categories**

- *Domain-Specific Resources:* Ontologies like MeSH (Medical Subject Headings)[2] or Gene Ontology (GO)[3] provide structured biomedical concepts, serving as alternatives to Wikipedia for obtaining fine-grained categories and entity lists.

(2) **Collecting Entity Contexts**

- *Scientific Literature:* Context sentences can be extracted from PubMed[4] abstracts or full-text articles. Tools like GNormPlus [44, 45] can identify and normalize gene/protein mentions to ensure precise entity linking.

(3) **Annotating Entity Attributes**

- *Structured Databases:* Domain-specific resources like UniProt[5] can offer detailed annotations for protein properties (e.g., protein function, subcellular

2. https://www.ncbi.nlm.nih.gov/mesh/
3. https://geneontology.org/
4. https://pubmed.ncbi.nlm.nih.gov/
5. https://www.uniprot.org/

TABLE 14. THE PROMPT USED TO GENERATE A CLASS NAME THAT COVERS THE GIVEN ENTITIES.

Generate a class name that accurately represents the following entities. This class name should encompass all the given entities and reflect their shared characteristics. Examples: [Tiger, Lion, Cheetah] → Big Cats [Shakespeare, Tolstoy, Hemingway] → Famous Authors [Mercury, Venus, Mars] → Planets in the Solar System [{Entity1}, {Entity2}, {Entity3}] → _____

location). Manual curation can also be introduced to ensure benchmark quality.

(4) **Defining Ultra-Fine-Grained Classes**

- *Attribute Constraints:* Formulate ultra-fine classes by combining positive and negative attribute constraints. Our negative-aware semantic class generation algorithm can be directly applied at this step.

These steps can be adapted to different domains, while the essential goal is still to construct ultra-fine-grained semantic classes using attribute constraints. Certainly, manual curation remains inevitable; however, the rise of LLMs like ChatGPT can reduce human effort. Exploring how LLMs can facilitate high-quality, automated dataset creation remains a promising research direction.

Appendix F. Parameter Analysis

To explore the sensitivity of each key hyperparameter in RetExpan and GenExpan, we conducted a comprehensive analysis of the smoothing factor η and the re-ranking segment length l in RetExpan, as well as the number of positive and negative entities mined in the contrastive learning strategy $|L_{pos}|$ and $|L_{neg}|$, the Top-p of entity selection module in GenExpan, and the re-ranking segment length l, respectively. The results are depicted in Figure 7, from which we can observe that:

(1) The label smoothing η serves to mitigate the penalty for entities possessing similar semantics to ground-truth entities. In Ultra-ESE, there are fewer entities exhibiting similarity to ground-truth entities at the ultra-fine-grained semantic level, thereby diminishing the utility of label smoothing. Consequently, the fluctuation in model performance due to parameter variations is relatively not significant.

(2) Excessive segment length l in both RetExpan and GenExpan equates to degrading the segmented re-ranking strategy to a naive re-ranking approach, thereby introducing noise from irrelevant entities, as mentioned in the main text. Hence, there is a general decrease in both PosMAP@K as the segment length increases, with larger K values leading to greater declines.

(3) Regarding the ultra-fine-grained contrastive learning strategy in RetExpan, we observe that mining too many or too few positive and negative entities using GPT-4 results in performance degradation. Insufficient positive and negative entities fail to fully stimulate the potential of contrast

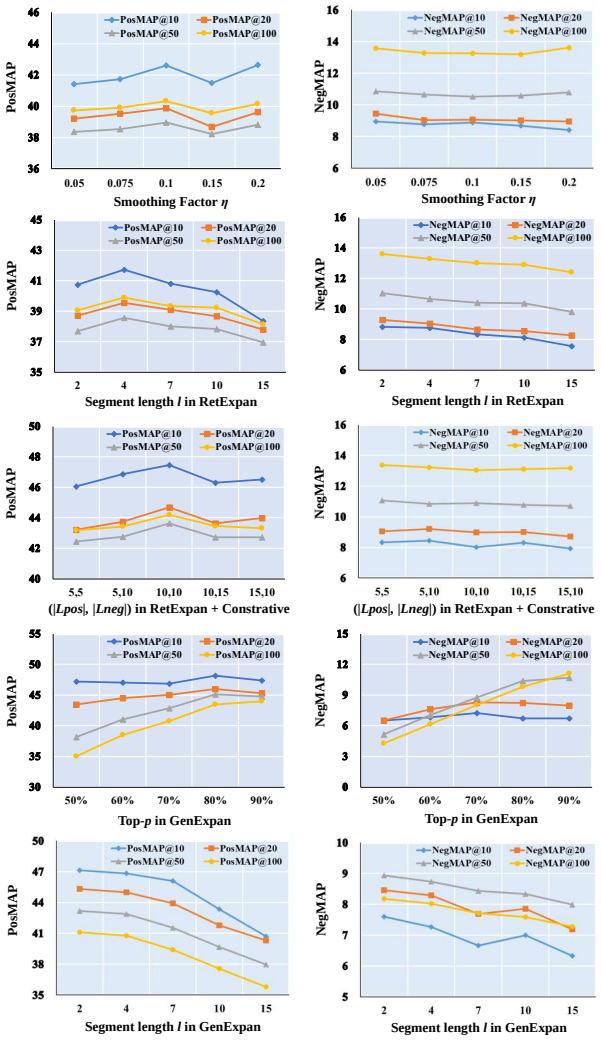

Figure 7. Parameter analysis experiments.

learning, whereas an excess of such entities introduces noisy entities. Therefore, as the number of entities to be mined increases gradually, PosMAP initially rises and then falls, and NegMAP exhibits a slight initial decline followed by a rise. However, overall, these two parameters demonstrate robustness, with even the worst-performing set of values outperforming the original RetExpan.

(4) A trade-off is also necessary for the Top-p. A too

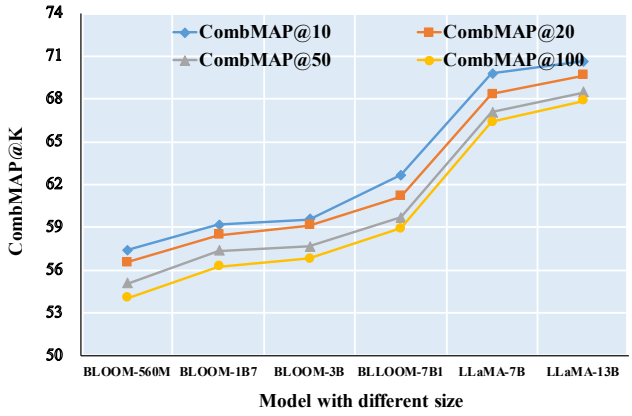

Figure 8. Comparison experiments on different LLM families and sizes.

small Top-p may fail to ensure the diversity of newly generated entities, while a too large Top-p may introduce an excessive number of irrelevant entities during iterative expansion. Consequently, we observe a decrease in PosMAP and a notable increase in NegMAP when Top-p is increased from 80% to 90%, which indicates that there is considerable space for improving the current rejection strategy for negatively targeted entities.

Appendix G. Analysis of Model Size

We utilize various families and sizes of LLMs as backbone model of GenExpan, including 560M, 1B7, 3B, 7B1 for BLOOM, as well as 7B and 13B for LLaMA. The experiment results depicted in Figure 8 align well with the expectation. Both the BLOOM and LLaMA families satisfy the scaling law: larger models tend to be more effective in Ultra-ESE, yielding better results. Moreover, LLaMA-7B outperforms BLOOM-7B1 at the same scale.

Appendix H. Impact of Negative Seeds on Performance

As shown in Tables 4 and 6 (in main body of paper), we have analyzed how the number of negative attributes affects model performance. Here, we further investigate how the number of negative seed entities impacts model performance, conducting experiments on the RetExpan model.

As shown in Figure 9, we progressively remove negative seed entities from the input. The results clearly indicate that as the number of negative seeds decreases, overall model performance declines. Specifically, PosMAP decreases, while NegMAP increases accordingly. Additionally, we observe that negative seeds have a greater impact on the Negative metric, and as the number of negative seeds increases, the performance gain gradually diminishes (diminishing marginal effect).

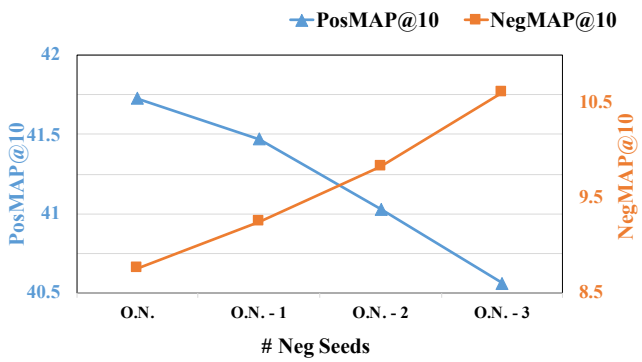

Figure 9. The figure shows the performance curve as the number of negative seeds varies. The x-axis, labeled "# Neg Seeds", represents the number of negative seed entities in input. O.N. means original number of negative seed entities.

Appendix I. Why GenExpan Omits Negative Seeds in Entity Generation

The primary reason for not using negative seeds during the entity generation stage is to maximize recall in this initial phase, ensuring the retrieval of as many entities belong to fine-grained class as possible. This intuition aligns with the first-stage retrieval (coarse ranking) in recommendation and retrieval systems.

In our preliminary experiments, we compared results with and without negative seeds, using the negative-seed-enhanced prompt described in Table 16. As reported in Table 15, we evaluated both the first-stage recall (coarse recall in entity generation) and end-to-end model performance (ComMAP). The results show that introducing negative seeds harms recall, which in turn leads to a decline in overall MAP performance.

TABLE 15. APPLYING NEGATIVE SEEDS TO THE ENTITY GENERATION PHASE (SIMPLY NOTED AS PHASE 1) OF GENEXPAN.

Method	Metric	@10	@20	@50	@100
GenExpan	Recall in Phase 1	93.58	92.43	74.14	52.30
	ComMAP	69.79	68.35	67.07	66.38
+ Neg. seeds in Phase 1	Recall in Phase 1	95.27	83.65	48.52	27.24
	ComMAP	65.91	63.79	61.05	60.02

Appendix J. Enhancing Performance of RetExpan and GenExpan Using Entity Descriptions

Leveraging seed entity descriptions can further enhance LLM's reflection on its own generation. On one hand, the contextual information of entities primarily comes from the provided corpus D, which may not be sufficient for capturing ultra-fine-grained semantic distinctions. Incorporating

TABLE 16. PROMPT USED IN THE GENERATION PHASE OF GENEXPAN WITH NEGATIVE SEEDS.

similar to iron, copper, aluminum, rather than wood, plastic, glass, it is zinc. similar to math, physics, chemistry, rather than history, art, music, it is biology. similar to {pos_ents}, rather than {neg_ents}, it is _____

TABLE 17. WE REPORT THE MAP AND P SCORES CORRESPONDING TO POS/NEG/COMB ON ULTRAWIKI. THE HIGHEST SCORE IS BOLDED.

Metric Type	Method Type	Method	MAP				P				Avg
			@10	@20	@50	@100	@10	@20	@50	@100	
Pos ↑	Retrieval Based (Ours)	RetExpan	41.73	39.53	38.55	39.91	54.58	58.03	66.76	77.23	52.04
		RetExpan + Contrast	47.45	44.68	43.63	**44.20**	59.83	62.02	**69.36**	**77.92**	**56.14**
		RetExpan + RA	44.74	42.23	40.66	40.27	58.31	61.36	66.58	73.97	53.52
	Generation Based (Ours)	GenExpan	46.79	45.00	42.89	40.80	59.77	62.15	66.26	66.57	53.78
		GenExpan + CoT	**50.39**	**47.80**	43.67	40.06	**62.74**	**64.45**	64.06	60.38	54.19
		GenExpan + RA	48.68	46.26	**43.80**	41.27	61.12	62.64	66.49	65.65	54.49
Neg ↓	Retrieval Based (Ours)	RetExpan	8.77	9.04	10.65	13.29	16.44	21.04	34.78	56.54	21.32
		RetExpan + Contrast	8.02	8.98	10.89	13.05	14.83	21.23	35.47	55.12	20.95
		RetExpan + RA	5.53	5.88	6.56	7.44	11.70	16.70	25.81	37.05	14.58
	Generation Based (Ours)	GenExpan	7.25	8.28	8.72	8.01	15.21	21.31	27.33	28.58	15.59
		GenExpan + CoT	7.79	9.29	8.15	6.89	15.97	22.66	23.52	21.90	14.52
		GenExpan + RA	7.16	8.54	8.63	7.70	14.79	21.18	26.67	26.84	15.19
Comb ↑	Retrieval Based (Ours)	RetExpan	66.48	65.25	63.95	63.31	69.07	68.50	65.99	60.34	65.36
		RetExpan + Contrast	69.72	67.85	66.37	65.57	72.50	70.39	66.95	61.40	67.59
		RetExpan + RA	69.60	68.18	67.05	66.42	73.31	**72.33**	**70.39**	68.46	69.47
	Generation Based (Ours)	GenExpan	69.77	68.36	67.08	66.40	72.28	70.42	69.47	68.99	69.10
		GenExpan + CoT	**71.30**	**69.25**	**67.76**	66.58	**73.39**	70.90	70.27	69.24	**69.84**
		GenExpan + RA	70.76	68.86	67.58	**66.78**	73.16	70.73	69.91	**69.41**	69.65

entity descriptions introduces additional external knowledge. On the other hand, entity-specific information can be diluted during training, especially for long-tail entities. Explicitly including entity descriptions in the prompt can help activate long-tail knowledge during inference.

To explore this, we propose a entity-based retrieval-augmented method (RA). Specifically, we retrieve entity introduction texts from Wikidata to enrich entity knowledge. During both training and inference, these entity descriptions serve as prefixes for all sentences, enhancing the model's understanding of ultra-fine-grained entity semantics. As shown in Table 17, this simple retrieval-augmentation strategy significantly improves performance for both RetExpan and GenExpan. Notably, for RetExpan, this strategy improves the comprehensive Comb metrics by an average of 4.11 points. The strategy primarily optimizes the Neg class metrics, leading to a significant reduction in the intrusion of negative target entities. However, its effect on Pos class metrics exhibits instability. For instance, the PosP@50 and PosP@100 of RetExpan+RA are instead lower compared to the original RetExpan. This phenomenon can be attributed to the fact that the extra introductions corresponding to each entity are currently fixed and do not adapt to changes in the entity context. Consequently, this increases the homogeneity of the entity embedding between irrelevant entities (neither positive nor negative target entities) and positive seed entities under the same fine-grained semantic class to some extent. Our point is further supported by the observation that RetExpan's MAP@100 for fine-grained semantic classes grows from 82.08 to 87.17 after the introduction of the retrieval augmentation strategy. This suggests that a dynamic and more fine-grained knowledge retrieval strategy needs further investigation.

Appendix K. Dynamic Segmentation

The re-ranking strategy uses the uniform segmentation method to reduce noise. To further enhance the model performance, we propose **Confidence-based Dynamic Segmentation (CDS)**, an algorithm that partitions a set of candidate indices T into segments based on a confidence threshold p. The candidates are ranked in descending order by their similarity sim^+ to a set of positive seeds P. As shown in Algorithm 1, the algorithm first computes the similarity scores sim^+ between candidates and positive seeds. It then iteratively normalizes these scores within the remaining candidate pool and accumulates them until the sum exceeds the threshold p, creating a segment. This process repeats until all candidates are segmented. The resulting segments reflect regions of high confidence in the candidate list, ensuring that each segment maintains a consistent level of similarity to the positive seeds.

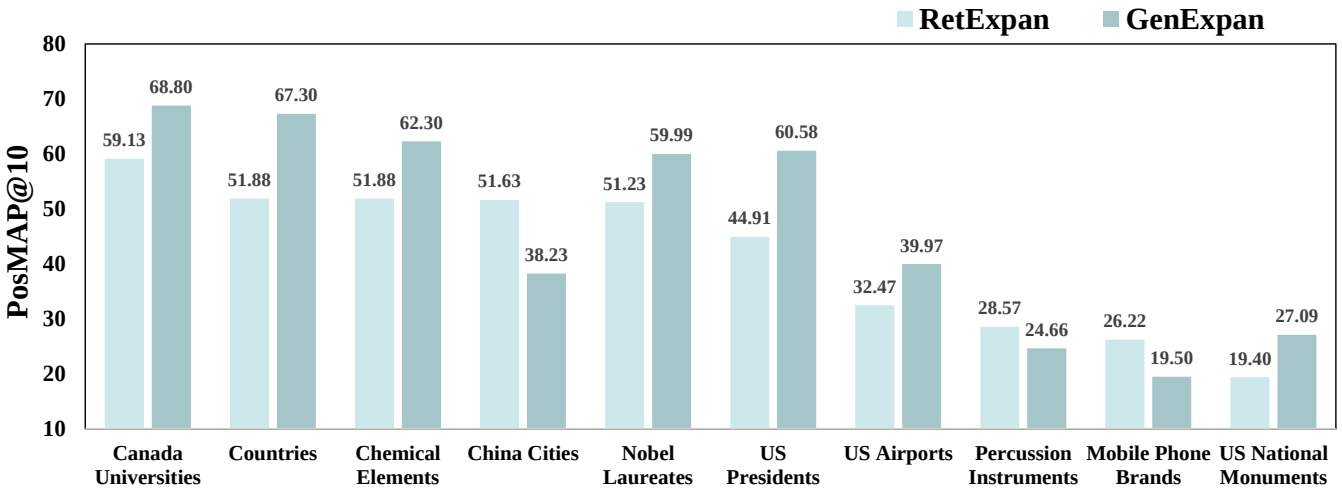

Figure 10. This figure shows the average PosMAP@10 of ultra-fine-grained semantic classes within each fine-grained class for RetExpan and GenExpan.

Algorithm 1 Confidence-based Dynamic Segmentation

Input:
  T: Candidate indices (sorted by sim^+ in descending order).
  P: Positive seed indices.
  p: Confidence threshold for segmentation ($0 < p \leq 1$).
Output: Segmented candidate indices.

Compute $sim^+ = \text{mean\_similarity}(T, P)$.
Initialize $i = 1$, segments $S = []$.
while $i \leq |T|$:
  Normalized scores $s_j \leftarrow \frac{sim_j^+}{\sum_{k=i}^{|T|} sim_k^+}$ for $j \in [i, |T|]$.
  Find the smallest $j > i$ such that $\sum_{k=i}^{j} s_k > p$.
  Add new segment $S \leftarrow S \cup \{T[i : j-1]\}$.
  Set $i = j + 1$.
return S.

TABLE 18. PERFORMANCE OF RETEXPAN AND THE RETEXPAN USING CONFIDENCE-BASED DYNAMIC SEGMENTATION, WHICH IS SIMPLY NOTED AS RETEXPAN (CDS).

Metric Type	Method	MAP				Avg
		@10	@20	@50	@100	
Pos	RetExpan	41.73	39.53	38.55	39.91	39.93
	RetExpan (CDS)	41.97	39.75	38.63	39.94	40.07
Neg	RetExpan	8.77	9.04	10.65	13.29	10.43
	RetExpan (CDS)	7.91	8.29	10.04	12.69	9.73
Comb	RetExpan	66.48	65.25	63.95	63.31	64.75
	RetExpan (CDS)	67.03	65.73	64.30	63.63	65.17

Appendix L.
Challenges with Long-Tail / Fine-Grained Entities

For classes with high semantic overlap, we take **Non-Small Airports in Florida** and **Non-Large Airports in Florida** as an example. The target entity overlap between these two ultra-fine-grained classes reaches 63.64%. Even the best-performing GenExpan model struggles with this distinction, showing a performance drop of nearly 40 points compared to the average (69.8 → 30.3). Additionally, we observe entity "intrusion" between these two classes, where 9% and 13% of target entities that should belong exclusively to one class appear in the top-100 expanded entities of the other class.

TABLE 19. STATISTICAL TABLE OF THE AVERAGE NUMBER OF ACTUAL EXPANSIONS (EXPANDED) AND EXPECTED EXPANSIONS (EXPECTED) FOR SELECTED NON-LONG-TAIL ENTITIES AND LONG-TAIL ENTITIES.

Entity Type	Expectation	Expansion	
		not expanded	expanded
Non-long-tail	not expected	0.45	11.09
	expected	0.00	15.45
Long-tail	not expected	19.64	1.74
	expected	4.00	1.62

For long-tail entities, we report the PosMAP@10 performance for each fine-grained semantic class in Figure 10. From the figure, we observe that the model performs worse on semantic classes with a higher proportion of long-tail entities, which aligns with intuition. To illustrate this more clearly, we randomly selected 50 entities from the "US National Monuments" class, where 78% were long-tail entities and 22% were non-long-tail entities. Here, long-tail entities are defined as those with fewer than 1,000 words in their Wikipedia entries. Furthermore, we analyzed a confusion matrix of expansion results, categorizing entities into expanded, not expanded, expected to be expanded, and not expected to be expanded, as shown in Table 19. The results indicate that non-long-tail entities were expanded significantly more frequently than long-tail entities. Specifically, non-long-tail entities appeared 26.5 times on average in expansions, whereas long-tail entities had an expansion count of 3.4 under the same conditions.

This suggests that long-tail entities, due to their limited available information, are harder for the model to capture, leading to poorer expansion performance. This phenomenon further highlights the performance bottleneck for long-tail entities in fine-grained semantic classes, especially in cases like "US National Monuments", where long-tail entities dominate.

Appendix M. Ultra-fine-grained Semantic Classes Visualization

To better illustrate the distribution of entities across ultra-fine-grained semantic classes divided by different attributes, we have visualized them with t-SNE. In Figure 11, based on the semantic class of **Nobel Laureates**, we divided them into four semantic classes according to four attribute values [Physics, Chemistry, Physiology or Medicine, Literature] under the **Prize** attribute. Entities with different prizes show distinct distribution differences, and after refining entity representations through our ultra-fine-grained contrastive learning, the clustering effect becomes more optimal. In Figure 12, based on two attribute values each from the **Prize** and **Gender** attributes, we divided the entities into four semantic classes, each constrained by two attributes. From the figure, it can be observed that entities with the same prize tend to cluster better than those with the same gender. After refining representations through contrastive learning, entities with different genders can also be well distinguished, although the effect is not as good as in the case of single-attribute constraints. This also indicates that ultra-fine-grained semantic classification with multiple attribute constraints places high demands on the precision of semantic understanding, and that contrastive learning is highly effective.

Appendix N. Case Study

Figure 13 presents some case studies of GenExpan and the chain-of-thought reasoning technique. Overall, it is evident that GenExpan predominantly avoids expanding entities that do not belong to the same semantic class as the positive and negative seed entities. This emphasizes the relative tractability of traditional ESE tasks compared to the significant challenges posed by Ultra-ESE for existing LLMs, where the model cannot exclude the expansion of negative target entities and irrelevant entities (belonging to the same fine-grained semantic class).

For chain-of-thought reasoning, an interesting example is shown in Figure 13. We find that the ground-truth positive target semantics is "low-income countries", but the model erroneously infers the positive semantic class as "African countries", which encompasses relatively more low-income countries, and thus improves the recall of the positive target entities. Of course, some high-income African countries such as "Seychelles" are also incorrectly introduced. In

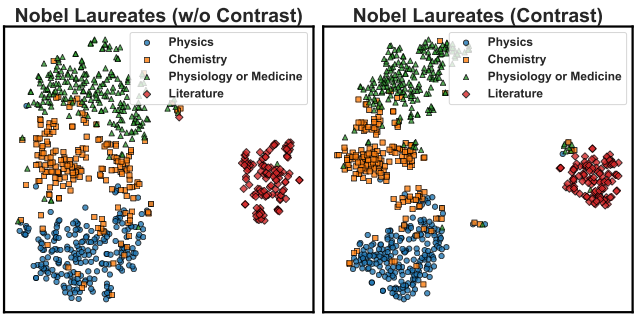

Figure 11. Comparison of clustering effects before and after contrastive learning on Nobel Laureates' **Prize** attribution (choosing 4 values [Physics, Chemistry, Physiology or Medicine, Literature]).

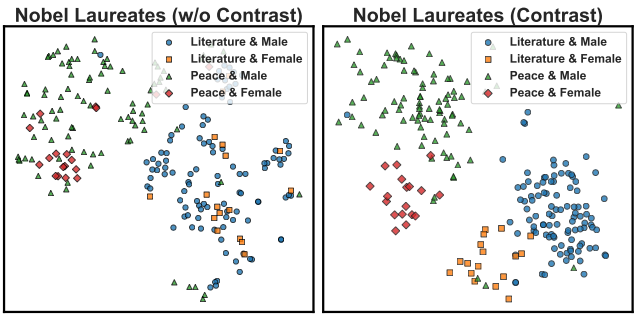

Figure 12. Comparison of clustering effects before and after contrastive learning on Nobel Laureates' **Prize** and **Gender** attributions ([Literature, Peace] × [Male, Female]).

fact, this error could have been avoided since the provided positive seed entity contains the non-African country "Saint Vincent and the Grenadines". This also shows the potential for improvement in the chain-of-thought reasoning strategy.

For retrieval augmentation proposed above, more positive target entities are introduced after fetching the additional knowledge. However, for some negative target entities that fulfill positive attributes, the retrieved knowledge may contain the introduction about its positive attributes, conversely misleading the model to overlook its negative attributes. For example, for the negative target entity "Xinxiang" in the second column, it also belongs to Henan province, i.e., it satisfies the requirement of positive attributes. Therefore, after concatenating the retrieved entity introduction "Xinxiang is a prefecture-level city *in northern Henan province*, China.", it is incorrectly introduced instead.

China Cities				Countries			
GenExpan		GenExpan + RA		GenExpan		GenExpan + CoT	
1	Changge +++	1	Xinmi +++	1	Saint Lucia +++	1	Saint Lucia +++
2	Linqing !!!	2	Gongyi +++	2	Grenada +++	2	Seychelles !!!
3	Gongyi +++	3	Yongcheng +++	3	Saint Kitts and Nevis !!!	3	São Tomé and Príncipe +++
	.....		.....		.....		.....
15	Linyi ---	15	Xinzheng +++	31	Cape Verde +++	31	Sierra Leone +++
16	Tongchuan ---	16	Xinxiang ---	32	Seychelles !!!	32	Guinea-Bissau +++
17	Linxia !!!	17	Xingyang +++	33	Barbados !!!	33	Liberia +++
18	Hanzhong ---	18	Changge +++	34	Trinidad and Tobago !!!	34	Comoros +++
19	Xingyang +++	19	Dengfeng +++	35	Cape Verde +++	35	Mauritius +++
	.....		.....		.....		.....

Positive Seeds	Negative Seeds	Positive Seeds	Negative Seeds
Xiangcheng	Shuozhou	Saint Vincent and the Grenadines	Paraguay
Linzhou	Zhangye	Guinea	Colombia
Yanshi	Lanzho	Libya	Bolivia
Weihui	Pu'er		
Mengzhou			
Positive Attributes		**Positive Attributes**	
<Province>	Henan	<Per_Capita_Income>	Low
Negative Attributes		**Negative Attributes**	
<Prefecture>	Perfecture-level city	<Continent>	South America

Figure 13. Case studies of GenExpan and the chain-of-thought reasoning technique. +++ and - - - stand for positive and negative target entities, respectively. !!! represents irrelevant entities belonging to the same fine-grained semantic class as the seed entities.