RESEARCH ARTICLE

Journal of Terminology and Knowledge Engineering
1: 20250007, 2026
https://doi.org/10.3724/jtke-20250007

Artificial Intelligence

ActiveIter-ED: An enhanced entity disambiguation method for knowledge graph construction based on active task disambiguation

Mohan Wang*, Hailong Cao* & Tiejun Zhao*

Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China

*Corresponding authors (emails: 15145082142@163.com (Mohan Wang); caohailong@hit.edu.cn (Hailong Cao); tjzhao@hit.edu.cn (Tiejun Zhao))

Received 25 December 2025; Revised 7 March 2026; Accepted 16 April 2026; Published online 9 July 2026

Abstract: Entity disambiguation is a core task in natural language processing that aims to link entity mentions in text to their corresponding entities in knowledge bases. Although dual-encoder architectures have achieved strong performance in this task, they still face significant challenges in handling ambiguous mentions, context dependencies, and negative sample selection. This paper proposes ActiveIter-ED, an enhanced entity disambiguation framework based on active task disambiguation. Our core innovation is a clarification sample augmentation mechanism that generates discriminative clarification questions via large language models (LLMs) and dynamically embeds them into entity representations, enabling the model to distinguish easily confusable entity pairs. Specifically, we design a three-stage pipeline: (1) hard negative sample discovery based on similarity retrieval, (2) LLM-driven clarification question generation that produces discriminative clarification questions for each positive-negative sample pair, and (3) probabilistic integration of clarification questions into entity representations during training to form augmented training samples.

Keywords: entity disambiguation, clarification sample augmentation, knowledge graph construction, terminology extraction, active learning

INTRODUCTION

Terminology extraction is a fundamental task in knowledge engineering that aims to identify, extract, and normalize domain-specific core terms from unstructured professional texts to construct domain knowledge systems [1]. With the rise of machine learning and large language model technologies, terminology management has evolved from traditional labor-intensive manual processes to efficient automated systems, enabling terminology databases to transform into dynamic, semantically rich, context-aware knowledge assets [1,2]. This transformation expands the scope of terminology extraction beyond mere term identification and extraction: more importantly, it addresses term ambiguity, establishes accurate mapping relationships between terms and their conceptual entities, and achieves semantic enhancement of terminology resources.

In the practice of professional domain knowledge engineering, term ambiguity is particularly prominent. In professional texts, the same term may denote entirely distinct conceptual entities across different contexts—a phenomenon especially prevalent in highly specialized fields like medicine, law, engineering, and geology

© The Author(s) 2026. Published by Science Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

[3]. Taking the oilfield development domain as an example, "water flooding" may refer to different levels of water drive conditions (e.g., initial flooding, late-stage flooding, complete flooding) in different paragraphs. Additionally, professional terms often involve complex scenarios such as abbreviations, symbolic expressions, and cross-lingual representations, further increasing the difficulty of term understanding. These challenges render traditional methods inadequate in handling the highly context-dependent semantic variations in professional texts. Particularly when processing long documents, cross-chapter references, and domain-specific abbreviations, the effectiveness of term identification and disambiguation declines significantly [4].

Entity linking (EL) is a core technical component of terminology extraction, aiming to accurately link entity mentions in text to their corresponding entities in knowledge bases (e.g., Wikidata, Wikipedia). It plays an irreplaceable role in modern terminology management systems that integrate terminology extraction with knowledge graphs [1,5]. With the advancement of pre-trained language models, dense retrieval methods based on dual encoders have become mainstream solutions due to their efficiency and scalability [6,7]. However, existing entity disambiguation systems still face two core challenges. First, task ambiguity poses a significant problem. Entity mentions in natural language often exhibit polysemy, and initial instructions or context may be insufficient to uniquely determine the target entity. For instance, the mention "Java" may refer to a programming language, an island, or a type of coffee, making accurate disambiguation difficult based solely on local context. Kobalczyk *et al*. [6] noted that large language models (LLMs) lack the ability to actively clarify ambiguous tasks, leading to outputs inconsistent with users' true intentions. Second, training efficiency and context modeling remain challenging [8]. Traditional training methods rely on static negative sampling, which struggles to capture semantic differences in complex contexts. More importantly, existing methods lack effective mechanisms to help models learn to distinguish entities that are semantically similar but conceptually distinct—resulting in high error rates in ambiguous mention scenarios.

Contributions. To address the above problems, we propose ActiveIter-ED (active iterative entity disambiguation), an enhanced entity disambiguation framework based on active task disambiguation. Our core innovation is the clarification sample augmentation mechanism, which generates discriminative clarification questions through large language models and embeds them into the training process, significantly improving the model's ability to distinguish similar entities.

In more detail, our contributions are as follows.

(1) We design a three-stage pipeline (hard negative sample discovery, LLM-driven clarification question generation, and dynamic embedding during training) that probabilistically appends clarification questions to entity representations, enabling the model to learn to distinguish easily confusable entity pairs. To our knowledge, this is the first method to systematically embed clarification questions into the entity disambiguation training process.

(2) Guided by Bayesian experimental design, the model generates clarification questions with high information gain, dynamically narrowing the candidate entity space.

(3) We validate the method's effectiveness on 8 Zero-shot Entity Linking Dataset (ZELDA) test sets, achieving improvements on 4 datasets—with the most significant gain on the AIDA-B dataset. Ablation studies confirm that clarification sample augmentation contributes the most to performance improvement, and we provide in-depth analysis of the impact of different design decisions.

(4) We verify the method's practical value in domain knowledge engineering through knowledge graph construction experiments in both oil & gas exploration and advertising compliance domains. Results demonstrate that ActiveIter-ED significantly improves entity linking accuracy (by 4.5% and 5.1%, respectively) and knowledge graph quality, providing effective technical support for terminology extraction and knowledge fusion in complex professional scenarios.

RELATED WORK

This section reviews related work in three areas relevant to our approach. We first discuss terminology extraction and the role of entity disambiguation within it; we then survey existing entity disambiguation methods and negative sampling strategies; finally, we examine active learning and task disambiguation, which together inspire our clarification sample augmentation mechanism.

Terminology extraction

As a fundamental task in knowledge engineering, terminology extraction focuses on identifying, extracting, and normalizing domain-specific core terms from unstructured professional texts to construct domain knowledge systems [1]. Within this process, entity disambiguation serves as a key technical component with three core functions. First, it resolves the polysemy problem in professional terminology, where terms with identical surface forms may refer to distinct conceptual entities across contexts—particularly in highly specialized fields such as medicine, law, engineering, and geology [3]. Entity disambiguation precisely links term mentions to the correct conceptual entities in knowledge graphs, thereby supporting the construction of high-quality domain knowledge graphs [9,10]. Second, it addresses the "one meaning, multiple words" problem. The same conceptual entity may appear in professional texts in various forms (e.g., full names, abbreviations, synonyms, cross-lingual expressions) [11]. By normalizing these diverse term representations to link to standardized entities, entity disambiguation enables unified terminology management and knowledge integration—playing a vital role in the construction and expansion of terminology knowledge bases [1,12]. Third, it provides a technical foundation for the development of professional terminology knowledge bases, acting as a key bridge in transforming unstructured text into structured knowledge [13]. By linking term mentions in text to standardized entities in knowledge bases, entity disambiguation lays the groundwork for subsequent relation extraction and knowledge graph construction, enhancing the usability and reusability of terminology resources [14].

However, entity disambiguation in terminology extraction scenarios faces unique challenges. Compared to general-domain entity disambiguation, terminology extraction requires handling deeper semantic ambiguity, stronger domain dependencies, and finer-grained distinction requirements. The semantic differentiation of professional terms often relies on deep domain knowledge, fine-grained contextual information, and complex domain ontology relationships. Particularly in professional texts, there are numerous term pairs that are semantically similar but conceptually distinct. These pairs are often clustered closely in the semantic representation space, making it difficult for traditional entity disambiguation methods to distinguish them effectively—leading to erroneous links and knowledge pollution [14]. Existing methods exhibit notable

shortcomings in addressing these challenges, which is the core problem this study aims to solve. In terminology extraction, entity disambiguation must address challenges that go beyond those in general-domain contexts. Terminological ambiguity often involves deeper semantic granularity. Secondly, professional terms frequently appear in abbreviated, symbolic, or cross-lingual forms, requiring normalization to canonical concepts. Thirdly, the distinction between semantically similar but conceptually distinct term pairs often hinges on subtle domain knowledge and fine-grained ontology relations, which are rarely encountered in open-domain entity linking. These challenges necessitate disambiguation methods that can leverage domain-specific knowledge and actively acquire discriminative features, which motivates our clarification sample augmentation mechanism.

Entity disambiguation methods and negative sampling

Entity linking, a core task in knowledge engineering, aims to link entity mentions in text to their corresponding entities in knowledge bases. In terminology extraction, entity disambiguation plays a crucial role by transforming unstructured text into structured knowledge by resolving term ambiguity, but also provides technical support for the knowledge representation of professional terms, term relation extraction, and domain ontology construction [3,11]. In recent years, dense retrieval methods based on pre-trained language models have become mainstream, achieving significant progress in general domains. However, entity disambiguation methods for terminology extraction still face unique challenges—particularly in handling domain-specific semantic ambiguity, terminology normalization, and cross-document knowledge fusion—where significant deficiencies remain.

Existing entity disambiguation methods can be broadly categorized into three types: classification-based, generative, and dense retrieval methods. Classification-based methods [8,15] make predictions on fixed candidate entity sets via Softmax classification heads, but their generalization ability is limited by predefined label sets, making it difficult to extend to new domain terms. Generative methods [16] treat entity disambiguation as a sequence generation task but suffer from high computational overhead, failing to meet real-time requirements. Dense retrieval methods—especially dual-encoder architectures—map mention contexts and entity descriptions to the same vector space through two independent encoders, achieving a favorable balance between efficiency and effectiveness [17–19]. This makes them a more attractive choice for terminology extraction scenarios.

Recent research, such as the VerbalizED model [20], has systematically evaluated and optimized key design decisions for dual encoders in entity disambiguation. This model uses dual encoders directly for final prediction, eliminating the time-consuming cross-encoder reranking step. The core improvements of VerbalizED include rich label verbalization (combining entity titles, descriptions, and category information), efficient hard negative sampling, document-level context utilization, and dynamic label embedding updates. These designs enable VerbalizED to achieve state-of-the-art performance on multiple benchmarks (including ZELDA) and provide an important baseline and methodological foundation for our work.

Hard negative sampling is critical for improving dual-encoder model performance. Traditional methods rely on in-batch negative sampling or surface-form-based hard sample retrieval. In terminology extraction, term pairs that are semantically similar but conceptually distinct often have similar surface forms, making it difficult for surface-form-based retrieval to identify truly hard negative samples. Existing methods primarily

focus on selecting hard negative samples but lack mechanisms to actively teach models how to distinguish these semantically similar entities. Our clarification sample augmentation mechanism addresses this gap: by generating clarification questions through LLMs that can clearly differentiate two similar entities and dynamically embedding these questions into the training process, we actively help the model learn key distinguishing features.

Active learning and task disambiguation

Traditional machine learning has extensively explored efficient data collection under model uncertainty, developing active learning strategies [21,22] that aim to reduce annotation costs by selecting the most informative samples for labeling [23]. To address the inherent ambiguity of natural language or intentional underspecification [24,25], active task disambiguation methods based on Bayesian experimental design [6] formalize task ambiguity as follows: when the feasible solution set H corresponding to task statement S= (R, C) (where R is the known requirement and C is the context information) is a proper superset of the true solution set H*, the task is considered ambiguous. In such cases, LLM agents should be able to gradually acquire additional task specifications by proposing clarification questions, narrowing the feasible solution space, and thereby reducing the risk of outputs inconsistent with users' true intentions.

The theoretical core of this method is the maximization of expected information gain (EIG). Specifically, for each candidate question q, its EIG is defined as

$$EIG(q) = \mathbb{H}[p^*(h|S)] - \mathbb{E}_{a\sim p^*(a|q,S)}[\mathbb{H}[p^*(h|S\cup(q,a))]],$$

where p^* represents the true preference distribution of the question proposer. To achieve efficient question selection, an explicit reasoning strategy is adopted: sample N candidate solutions and M candidate questions, generate pseudo-answers to estimate the EIG of each question, and finally select the question with the highest EIG. This method has been validated in code generation and "20 questions" games, significantly outperforming baseline methods that rely solely on implicit reasoning.

State-of-the-art LLM strategies have demonstrated remarkable success in logical reasoning [26,27], mathematical problem-solving [28,29], code generation [30,31], and creative writing [32,33]. Although they exhibit strong disambiguation capabilities across multiple tasks, their application in entity disambiguation remains unexplored. This paper draws on their core ideas, introducing a clarification question generation mechanism into the entity disambiguation training process to enhance the model's ability to distinguish ambiguous mentions.

ACTIVEITER-ED METHOD

This section presents the ActiveIter-ED framework in detail. We first describe the overall architecture and its three core components, then formalize the clarification sample augmentation mechanism through a three-stage pipeline—hard negative sample discovery, LLM-driven clarification question generation, and dynamic embedding during training—and finally introduce the training objective and optimization strategies that enable efficient learning.

Overall architecture

The ActiveIter-ED framework comprises three core components: the mention encoder, the entity encoder, and the clarification sample augmentation module. These components work synergistically to form an end-to-end entity disambiguation system based on a dual-encoder architecture. By introducing the clarification sample augmentation mechanism, the framework achieves efficient entity disambiguation for ambiguous mentions. The overall design follows a three-stage paradigm of "encoding-matching-augmentation", with the core innovation lying in the systematic embedding of LLM-generated clarification questions into the training process—enabling the model to distinguish easily confusable entity pairs.

Mention encoder. The mention encoder is responsible for encoding entity mentions and their contextual information in text into dense vector representations. Built on pre-trained Transformer models (e.g., Bidirectional Encoder Representations from Transformers, BERT), it captures contextual semantics around mentions. For a given training sample (m, c, e^+) (where m is the mention text, c is the context, and e^+ is the positive entity), the mention encoder first concatenates the mention and its context into an input sequence, encodes it via multi-layer Transformers, and ultimately generates a fixed-dimensional vector representation through pooling operations:

$$h_m = MentionEncoder([CLS]c[SEP]m[SEP]),$$

where h_m denotes the vector representation of the mention, and $[CLS]$ and $[SEP]$ are special token symbols used to identify the beginning and separation of sequences.

Entity encoder. The entity encoder encodes entity descriptions into vector representations. ActiveIter-ED adopts a structured label verbalization strategy, combining entity titles, descriptions, and category information into rich text descriptions. The entity encoder also leverages the Transformer architecture and may either share parameters with the mention encoder or operate independently, depending on the configuration. For entity e, its verbalization $v(e)$ is encoded as

$$h_e = EntityEncoder(v(e)),$$

where h_e represents the vector representation of the entity. A key feature of the entity encoder is its support for dynamic embedding updates—i.e., updating entity embeddings during training based on the current model state to maintain representation accuracy.

Clarification sample augmentation module. The clarification sample augmentation module is the core innovation of ActiveIter-ED, implementing three key functions: hard negative sample discovery, LLM clarification question generation, and dynamic embedding during training. During the training phase, the module first identifies candidate negative samples similar to positive entities via similarity computation, then uses LLMs to generate clarification questions that distinguish positive and negative samples, and finally probabilistically embeds these questions into entity representations to form augmented training samples. Specifically, the enhanced entity representation is expressed as

$$v_{enhanced}(e) = v(e) + X \cdot q_{clar}(e^+, e^-),$$

where X is an indicator variable controlling the probability of embedding clarification questions. This mechanism enables the model to learn both the basic semantics and distinguishing features of entities during training.

Interaction flow. The interaction flow between the three components is as follows. During the training

phase, the mention encoder encodes mention contexts from training samples into vectors, and the entity encoder encodes entity descriptions (which may include clarification question augmentation) into vectors, then computes the similarity between them. Similarity computation uses Euclidean distance or cosine similarity, and optimization is performed through contrastive learning, making the similarity of positive sample pairs higher than that of negative sample pairs. During the inference phase, the system first obtains the mention's vector representation through the mention encoder, then computes the similarity between this representation and all candidate entity representations, selecting the entity with the highest similarity as the prediction result.

The clarification sample augmentation module serves as a key bridge between hard negative sample mining and model training, providing additional discriminative signals to the model through high-quality clarification questions. This design enables ActiveIter-ED to significantly improve disambiguation performance for ambiguous mentions while retaining the efficiency of dual encoders—particularly excelling in handling semantically similar entity pairs.

Active task disambiguation mechanism

We formalize clarification question generation as an information gain maximization problem.

Problem definition. Given the current task statement $S = (R, C)$, where R is the known requirement, and C is the context information, our goal is to select a question q^* that maximizes the expected information gain:

$$q^* = argmax_{\{q \in Q\}} EIG(q),$$

where the expected information gain is defined as

$$EIG(q) = \mathbb{H}[p^*(h|S)] - \mathbb{E}_{a \sim p^*(a|q,S)}[\mathbb{H}[p^*(h|S \cup (q,a))]].$$

Here, h denotes candidate entities, and a is a possible answer to the question.

Implementation strategy. In practical implementation, we adopt the following steps to estimate *EIG*:

(1) Sample N candidate entities: $\{h_i\}_{\{i=1\}}^{N} \sim p_{\{\varphi_h\}}(\cdot|S)$;

(2) Sample M candidate questions: $\{q_j\}_{\{i=1\}}^{M} \sim p_{\{\varphi_q\}}(\cdot|S)$;

(3) For each question q_j and entity h_i, generate pseudo-answers $a_{\{i,k\}}$;

(4) Compute *EIG* estimates based on answer distributions.

For each candidate question q_j, we first collect all corresponding pseudo-answer sets $\{A_j\} = \{a_{\{i,j\}}\}_{\{i=1\}}^{N}$. Since different entities may produce the same answer to the same question, these answers may appear duplicated in the set. We construct the answer distribution by counting the frequency of each unique answer. Let the set of unique answers be $\{a_k\}_{\{k=1\}}^{n}$, where n is the number of unique answers ($n \leqslant N$), then the frequency of each answer a_k can be estimated as

$$p_k = \frac{Count(a_k, \{A_j\})}{N},$$

where $Count(a_k, \{A_j\})$ represents the number of times that answer a_k appears in set $\{A_j\}$. This frequency estimates the probability of obtaining answer a_k given question q_j and task statement S.

Based on this answer distribution, we compute the expected information gain estimate for the question q_j. From an information-theoretic perspective, information gain measures the degree to which answers reduce entity uncertainty. If the answer distribution is relatively concentrated (i.e., most entities produce the same or similar answers to the question), the answers provide less information; conversely, if the answer distribution is relatively uniform and dispersed, the answers can effectively distinguish different entities and provide more information. The *EIG* estimate can be expressed as the entropy of the answer distribution:

$$EIG(q_j) \approx -\Sigma^{n}_{\{k=1\}} p_k \log p_k.$$

When the answer distribution is highly concentrated (the probability of some answer approaches 1, while other answer probabilities approach 0), the entropy value is small, indicating that the question has weak discriminative ability. When the answer distribution is uniform (all answer probabilities are equal), the entropy reaches its maximum, indicating that the question has the strongest discriminative ability. Therefore, we select the question with the maximum entropy value as the optimal clarification question:

$$q^* = argmax_{\{q_j\}} EIG(q_j) = argmax_{\{q_j\}} \left(-\Sigma^{n}_{\{k=1\}} p_k \log p_k \right).$$

Clarification sample augmentation mechanism

Clarification sample augmentation is the core innovation of our method, designed to help the model distinguish easily confusable entities through LLM-generated clarification questions. This mechanism consists of three stages.

Hard negative sample discovery

Given a training sample (m, c, e^+), we first identify candidate negative samples similar to (but distinct from) e^+ via similarity computation. We use Jaccard similarity to measure the overlap between the word sets of the mention text and entity names:

$$Sim(e^+, e') = \frac{\left| W(m) \cap W(e') \right|}{\left| W(m) \cup W(e') \right|},$$

where $W(\cdot)$ represents converting text to a word set. We select the Top-K similar entities as the candidate negative sample set C_{neg}.

LLM clarification question generation

For each candidate negative sample $e^- \in C_{neg}$, we use large language models to generate clarification questions that effectively distinguish positive and negative sample entities. The core of this process is guiding LLMs to produce high-quality, discriminative questions that help the model learn key distinguishing features between entities during training.

Design principles for clarification question generation. The generation of clarification questions needs to meet several key requirements. First, questions must be discriminative and clearly highlight the key

differences between two similar entities in specific contexts. Second, questions should be concise and clear, with length controlled within a reasonable range to avoid redundant information interfering with model learning. Finally, questions should focus on contextually relevant features rather than only static attributes of entities themselves. These requirements together ensure that generated clarification questions can provide effective discriminative signals for the model.

Prompt template system. To meet these requirements, we design a hierarchical prompt template system. At the basic level, we adopt a general template structure containing core elements such as context information, mention text, and descriptions of positive and negative sample entities, guiding large language models to generate basic clarification questions. This general template is suitable for most scenarios and can handle general entity disambiguation needs. However, different datasets have different text types and domain characteristics, and general templates may not fully utilize dataset-specific information. Therefore, on the basis of general templates, we design dedicated templates for specific datasets. For example, for the NELDA dataset, which contains various text types such as news and social media, we design dedicated templates that can handle this diversity. These templates introduce keyword extraction mechanisms, extracting key distinguishing features from entity verbalizations, and organizing these features into prompts in a structured manner, guiding large language models to focus on these key discriminative pieces of information.

The generated clarification questions $q(e^+, e^-)$ are organized together with hard negative samples into tuple structures and saved as part of the training data. The complete hard negative sample set can be formally represented as

$$H = \{(c, m, e^+, e^-, q(e^+, e^-), v(e^+), v(e^-))\},$$

where $v(\cdot)$ denotes entity verbalization (including title, description, and category information). This structured storage enables efficient retrieval and use of clarification questions during training while maintaining data consistency and integrity.

Dynamic embedding during training

During training, we integrate the clarification sample augmentation mechanism into the entity disambiguation framework by extending the dual-encoder architecture. For each training sample (m, c, e^+), the system decides whether to augment the entity representation with probability p_{clar}. When augmentation is performed, the system retrieves the corresponding clarification question $q(e^+, e^-)$ from the pre-generated hard negative sample set H and appends it to the end of the original entity representation in a specific format. The formal representation of the enhanced entity representation $v_{enhanced}(e)$ is

$$v_{enhanced}(e) = v(e) + X \cdot q_{clar}(e^+, e^-),$$

where $q_{clar}(e^+, e^-)$ = "[clarification: "+ $q(e^+, e^-)$ +"]" represents the formatted clarification question string, and X is an indicator variable following a Bernoulli distribution $X \sim \text{Bernoulli}(p_{clar})$.

The probability control mechanism ensures an expected augmentation frequency of $E[X] = p_{clar}$—meaning that, on average, p_{clar} proportion of samples uses augmented entity representations, while $1 - p_{clar}$ uses original representations. This mixed strategy allows the model to learn discriminative features while retaining accurate understanding of original semantics, ensuring training efficiency.

From an information-theoretic perspective, the synergy between clarification questions and structured label

verbalization can be explained as follows. Let $I(v; y)$ denote the mutual information between verbalization v and label y, and $I(q; y|v)$ denote the conditional mutual information between clarification question q and y given v. The mutual information of the enhanced representation is

$$I(v_{enhanced}; y) = I(v; y) + I(q; y|v).$$

The information gain of the enhanced representation is

$$\Delta I = I(q; y|v).$$

This indicates that clarification questions provide additional discriminative information beyond the baseline verbalization. When question quality is high, $\Delta I \geqslant 0$, and clarification questions significantly improve discriminative ability.

Training objective and optimization

We adopt a contrastive learning framework, training the model by maximizing the similarity of positive sample pairs and minimizing the similarity of negative sample pairs. For a training sample (m, c, e^+), we obtain vector representations h_m and h_e through the mention encoder and entity encoder, respectively, using Euclidean distance $dist(m, e) = \|h_m - h_e\|_2$ as the similarity metric. The contrastive learning-based loss function can be expressed as

$$L = -\log\left(\frac{\exp(-dist(m, e^+))}{\exp(-dist(m, e^+)) + \Sigma_{\{e^- \in N\}}\exp(-dist(m, e^-))}\right),$$

where N is the hard negative sample set. The loss function normalizes through Softmax, maximizing the probability of positive sample pairs while minimizing the probability of negative sample pairs.

Hard negative sampling strategy. The hard negative sampling strategy adopts a two-stage mechanism. In the offline stage, for each training sample, we discover the Top-K similar entities based on similarity computation as candidate hard negative samples $N_{candidate} = \text{Top-}K_{\{e' \neq e^+\}} Sim(e^+, e')$. These candidate negative samples are pre-generated and stored together with clarification questions. In the online stage, we dynamically sample from the candidate set, prioritizing negative sample pairs that have generated clarification questions. The final hard negative sample set used for loss computation is $N = Sample(N_{candidate}, k, strategy = \text{hard})$.

Label embedding update mechanism. To balance efficiency and accuracy, we adopt a hybrid update strategy. Let $E = \{e_1, e_2, \ldots, e_M\}$ denote the set of all entities, and $H_E = \left\{h_{\{e_1\}}, h_{\{e_2\}}, \ldots, h_{\{e_M\}}\right\}$ denote the entity embedding cache. The hybrid update strategy consists of three components.

(1) **Periodic full update:** Update all entity embeddings every K training steps: $H_E^{t+K} = \{EntityEncoder(v(e_i)) \mid e_i \in E\}$.

(2) **Dynamic partial update:** Real-time update for entities used in the current batch $H_E^{t+1[e]} = EntityEncoder(v(e))$ for $e \in Entities(B)$.

(3) **High-frequency entity priority update:** Increase update frequency for frequently appearing entities.

This hybrid strategy reduces training time by 60%–80% while ensuring embedding accuracy, with negligible performance loss.

EXPERIMENTS

This section evaluates ActiveIter-ED through comprehensive experiments. We first describe the experimental setup, including datasets, evaluation metrics, baseline methods, and implementation details. We then report main results on standard benchmarks, followed by ablation studies that quantify the contribution of each component. We further demonstrate application effects in two real-world knowledge graph construction scenarios (geological exploration and advertising compliance), evaluate the quality of generated clarification questions, and conclude with a case study illustrating the discriminative ability of the proposed mechanism.

Experimental setup

Datasets. We conduct comprehensive evaluations on two widely used entity disambiguation benchmarks. First, we evaluate on ZELDA, a challenging benchmark that contains 8 diverse test sets spanning multiple domains: AIDA-B (news articles from Reuters), MSNBC (news broadcasts), AQUAINT (newswire text), ACE2004 (news and conversations), WNED-CWEB (web documents), WNED-WIKI (Wikipedia articles), REDDIT-POSTS (social media posts), and REDDIT-COMMENTS (social media comments). This diversity enables robust assessment of model generalization across different text types and domains. Second, we use AIDA-Yago, a standard benchmark derived from CoNLL 2003 data, which focuses specifically on news domain entity disambiguation and serves as a complementary evaluation for in-domain performance.

Evaluation metrics. We adopt standard entity disambiguation evaluation metrics: accuracy, F1 score, micro-averaged F1, and macro-averaged F1. For knowledge graph quality assessment, we use node correctness rate (proportion of correctly linked entity nodes to total nodes) and entity normalization rate (proportion of successfully normalized synonymous entity pairs).

Baseline methods. We compare ActiveIter-ED against three representative state-of-the-art entity disambiguation methods that exemplify different architectural paradigms. First, GENRE [34] represents the generative approach: it formulates entity linking as a sequence-to-sequence generation task, where an autoregressive Transformer directly produces entity names from mention contexts. Second, VerbalizED [20] embodies the dual-encoder retrieval paradigm and serves as our primary baseline. VerbalizED achieves current state-of-the-art performance through systematic optimisation of dual-encoder design decisions, including rich label verbalisation, efficient hard negative sampling, document-level context modelling, and dynamic label embedding updates. Third, we include LUKE_{FT} [35], a global entity disambiguation model that leverages entity-aware pre-training and an optional final epoch of fine-tuning with frozen entity embeddings. By comparing with these three methods, we demonstrate the effectiveness of our clarification sample augmentation mechanism across generative, dual-encoder, and entity-aware pre-training architectures.

Implementation details. We use BERT-base as the encoder backbone, with a batch size of 32 and a learning rate of 5×10^{-6}.

Clarification sample augmentation implementation details.

(1) **Hard negative sample discovery:** We use Jaccard similarity to compute word set similarity between mention text and entity names, selecting the Top-10 similar entities as candidate negative samples.

(2) **Clarification question generation:** LLM support: includes multiple large language models (Qwen, GPT-4, etc.). Dataset-specific templates: select prompt templates based on dataset names (NELDA, AIDA). Keyword extraction: for the NELDA dataset, extract key distinguishing features (type, domain, time, location, etc.) from verbalizations.

(3) **Embedding during training:** Probability control: clarification questions are dynamically embedded into entity representations with probability p_{clar}=0.5. Format specification: use the format 'original representation [Clarification: clarification question]' to maintain the integrity of original information.

(4) **Verbalization strategy:** We adopt the 'title + description + categories' strategy to construct entity representations, combining with clarification questions to form augmented training samples.

Main results

As shown in Table 1, ActiveIter-ED achieves excellent results on multiple datasets. We train the model on the NELDA dataset and evaluate performance on multiple test sets.

Overall performance improvement. ActiveIter-ED achieves an average accuracy of 80.4%, representing a 0.7-percentage-point improvement over the VerbalizED baseline. Improvements are observed on 4 datasets, primarily driven by the clarification sample augmentation mechanism.

Significantly improved datasets. AIDA-B shows an improvement of 4.5 percentage points, the largest across all datasets, demonstrating good cross-dataset generalization capability. REDDIT-POSTS shows an improvement of 3.5 percentage points, demonstrating strong performance on social media text. REDDIT-COMMENTS shows an improvement of 2.3 percentage points, effectively handling ambiguity even in short-text comment scenarios. WNED-CWEB shows an improvement of 1.3 percentage points, a stable gain on web text.

Outstanding performance on social media text. Accuracies of 92.7% (REDDIT-POSTS) and 88.5% (REDDIT-COMMENTS) indicate that the clarification sample augmentation mechanism effectively addresses ambiguity in informal text.

Challenging dataset analysis. SLINKS-SHADOW: Accuracy of 63.1% is slightly lower than the baseline. However, considering the challenging nature of this dataset (macro-averaged F1 of only 46.1%) and that our model is trained on NELDA while the baseline may be trained on more matched data, this result is still within a reasonable range. SLINKS-TAIL: Accuracy of 98.22% is slightly lower than the baseline (98.55%), a decrease of 0.33 percentage points, but it already approaches near-perfect performance.

As shown in Table 2, ActiveIter-ED performs exceptionally well in in-domain training. When trained and evaluated on the AIDA dataset, the model achieves an accuracy of 93.0%—demonstrating the superiority of

Table 1 Accuracy comparison (%) between ActiveIter-ED (trained on NELDA with clarification sample augmentation) and baseline methods across ZELDA test sets. Values in bold indicate the best performance.

Method	Dataset								Avg
	AIDA-B	TWEEKI	REDDIT-POSTS	REDDIT-COM	WNED-CWEB	SLINKS-TOP	SLINKS-TAIL	SLINKS-SHAD	
$GENRE_{ALL}$	72.4	75.9	88.8	83.9	66.5	43.5	95.3	38.7	70.6
$LUKE_{FT}$	81.2	77.8	81.5	78.5	70.3	51.8	98.0	22.5	70.2
VErbalizED	82.6	**78.9**	89.2	86.2	69.8	**67.1**	**98.6**	**65.3**	79.7
AcitveIter-ED	**87.1**	76.5	**92.7**	**88.5**	**71.1**	66.3	98.2	63.1	**80.4**

in-domain training and providing strong evidence for the effectiveness of the clarification sample augmentation mechanism.

Ablation studies

We conduct systematic ablation experiments to validate the contribution of each component, with results shown in Table 3.

The results demonstrate that clarification sample augmentation is an effective improvement method and contributes the most to performance: removing the clarification sample augmentation mechanism leads to a performance decrease of 2.5 F1 points, which is the largest contribution among all components. This proves the core role of clarification questions in helping the model distinguish similar entities.

Clarification questions vs. hard negative samples only: using only hard negative samples (without clarification questions) decreases by 0.9 F1 points compared to the complete model, indicating that clarification questions provide additional discriminative information beyond the negative samples themselves.

As shown in Table 4, p_{clar}=0.5 is the optimal configuration, achieving an average F1 score of 89.4% across three datasets. Too low (0.3) leads to insufficient model learning (88.6%), while too high (0.7) may cause overfitting (88.0%). This validates the necessity of the probability control mechanism.

Application effects in knowledge graph construction

Application in geological exploration knowledge graph: To validate ActiveIter-ED's practical value in knowledge engineering, we conduct knowledge graph construction experiments on geological exploration data. The experimental data includes reserve report texts from 9 well areas (≈15,000 sentences, 3200 entity mentions), covering 4 major categories and 12 subcategories of professional entities (blocks, well areas, oil layers, reserve parameters). The experimental process follows standard semantic knowledge graph construction methods [36], including entity linking, named entity extraction and validation, node construction, relation extraction, and graph quality assessment.

We adopt standard entity disambiguation evaluation metrics. For knowledge graph quality assessment, we compute node correctness rate (defined as the proportion of correctly linked entity nodes to total nodes) and

Table 2 Accuracy of ActiveIter-ED trained on the AIDA dataset.

Method	Accuracy (%)
VerbalizED	90.9
AcitveIter-ED	93.0

Table 3 Ablation study results (%) for the clarification sample augmentation mechanism and the negative sample mechanism. Values in bold indicate the best performance.

Method	Dataset			Avg
	AIDA-B	REDDIT-POSTS	REDDIT-COMMENTS	
ActiveIter-ED	87.1	92.7	88.5	**89.4**
ActiveIter-ED−clarified sample	83.1	91.2	86.5	86.9 (−2.5)
ActiveIter-ED−negative sample	86.6	92.1	86.7	88.5 (−0.9)

Table 4 Impact (%) of different configurations of clarification sample augmentation. Values in bold indicate the best performance.

Method	Dataset			Avg
	AIDA-B	REDDIT-POSTS	REDDIT-COMMENTS	
p_{clar} = 0.5	87.1	92.7	88.5	**89.4**
p_{clar} = 0.3	85.8	92.1	87.8	88.6 (−0.8)
p_{clar} = 0.7	85.2	90.8	88.0	88.0 (−1.4)

entity normalization rate (defined as the proportion of successfully normalized entity pairs to all synonymous entity pairs). All evaluation results are validated through manually annotated gold standards. The main quantitative results are summarized in Table 5.

Experimental results show that ActiveIter-ED achieves significant improvements in knowledge graph construction. Entity linking accuracy improves from 62.6% with the baseline method to 67.1% with ActiveIter-ED, directly driving knowledge graph node correctness from 58.3% to 64.7%. Additionally, entity normalization rate improves from 63.4% with the baseline method to 69.2%. The constructed knowledge graph contains 2847 entity nodes and 7203 relations. These improvements are primarily attributed to the clarification sample augmentation mechanism, which effectively reduces entity confusion and avoids knowledge pollution issues, providing a reliable technical foundation for domain knowledge management and decision support.

Application in advertising compliance knowledge graph: To further validate practical value in knowledge engineering, we conduct comprehensive knowledge graph construction experiments on advertising compliance data—a critical regulatory domain with complex ambiguity challenges. The experimental dataset comprises three interconnected sub-graphs: (1) illegal activities and legal evidence hierarchy, containing 2847 illegal activity nodes connected to 9896 legal evidence nodes across a three-level structure (articles, items, subitems); (2) application ecosystem, including 8432 application nodes, 1156 platform nodes, and 2387 influentials nodes representing social media accounts; and (3) multimodal regulatory classifications, comprising 4832 nodes across four core violation categories: inappropriate content, absolute and exaggerated claims, medical terminology misuse, and politically sensitive information. This multi-modal, hierarchical structure presents unique challenges for entity disambiguation, particularly in distinguishing between highly similar illegal activity categories and maintaining semantic coherence across different modalities.

As shown in Table 6, ActiveIter-ED achieves substantial improvements across all evaluation metrics in the advertising compliance domain. Entity linking accuracy reaches 78.6%, representing a 5.1-percentage-point improvement over the baseline. This achievement indicates that, through the clarification sample augmentation mechanism, the model can effectively handle complex ambiguity issues inherent in the advertising regulation domain, accurately distinguishing between highly similar illegal activity categories and their corresponding multi-level legal provisions. More critically, the node correctness rate increases from 74.5% to 77.7%. This improvement directly reflects the enhanced quality of knowledge graph construction, signifying a significant reduction in entity linking errors within the graph and effectively avoiding

Table 5 Application effects in geological knowledge graph construction.

Method	Node correctness rate (%)	Entity linking accuracy (%)	Entity normalization rate (%)
VerbalizED	58.3	62.6	63.4
AcitveIter-ED	64.7	67.1	69.2

Table 6 Application effects in advertising compliance knowledge graph construction.

Method	Node correctness rate (%)	Entity linking accuracy (%)	Entity normalization rate (%)
VerbalizED	74.5	73.5	70.2
AcitveIter-ED	77.7	78.6	75.6

"knowledge pollution". In high-risk regulatory domains like advertising compliance, the improvement in node correctness is decisive for ensuring the accuracy of subsequent regulatory judgments. High-quality node linking ensures that illegal activities are accurately mapped to corresponding applications, platforms, or legal provisions, providing precise data support for regulatory enforcement.

The consistent performance improvements across both geological exploration and advertising compliance domains provide strong evidence for ActiveIter-ED's versatility and effectiveness in diverse knowledge engineering scenarios. The achievement of 78.6% entity linking accuracy and 77.7% node correctness rate in the advertising compliance domain—along with the previously reported 67.1% accuracy in geological domain construction—demonstrates that the clarification sample augmentation mechanism generalizes well across distinct application scenarios with different domain characteristics, hierarchical structures, and data modalities. This further validates the method's practical value for real-world terminology extraction and knowledge fusion applications.

Clarification question quality evaluation

To ensure the reliability of the clarification sample augmentation mechanism, we conducted a systematic quality evaluation of the generated questions. We invited three natural language processing researchers to perform manual annotation, categorizing the questions into three quality levels.

High-quality (85.2%): Questions that (a) directly target the key distinguishing features between the positive and negative entities, (b) are relevant to the domain context of the mention, and (c) are answerable based on the given context. For example, for the entity pair "Table Talk", a high-quality question would be: "Does the term 'Table Talk' here refer to the syndicated American television talk show, or to the informal conversational genre exemplified by Martin Luther's collected dinner-table discussions". Medium-quality (10.3%): Questions that are relevant but are either too broad, too specific, or partially ambiguous. For example, for the entity pair "feral child", asking "Where did this child come from" might be too broad, while asking "Was this child discovered in 1798" might be too specific to uniquely distinguish between different feral child cases or legal definitions from different periods. Low-quality (4.5%): Questions that are (a) too generic and applicable to almost any entity (e.g., 'What is this entity'), (b) completely unrelated to the entity pair, (c) factually incorrect, or (d) unanswerable within the given context.

Among the 15% of medium or low-quality questions, we identified three common failure modes. First, overly generic questions (e.g., asking 'What is this') failed to provide distinguishing information, accounting for about 6% of all questions. Second, entity-specific but contextually irrelevant questions (e.g., asking "Did he create 'Snow White'" for the mention "Walt Disney" in a context discussing the history of animation) are relevant to the entity but cannot distinguish the author of Snow White from a car race event within the given context, accounting for about 5% of cases. Third, factually inconsistent or hallucinated questions (e.g., assigning incorrect attributes to an entity) accounted for the remaining 4%. These failures primarily stem from sparse or conflicting information in entity descriptions, highlighting the necessity of introducing a

quality filtering mechanism.

Case study

To concretely illustrate the advantage of the clarification sample augmentation mechanism over the VerbalizED baseline method, this paper presents a detailed case study involving the disambiguation of Putaohua Oil Layer in reserve report texts. This case represents a typical phenomenon of hierarchical semantic drift prevalent in geological terminology, where the same geological unit is annotated as different entity types across various contexts.

The entity mention "Putaohua Oil Layer" appears in the corpus with three distinct semantic roles: (1) geological feature, (2) target layer, and (3) oil-bearing layer. Although all three refer to the same geological formation, their conceptual boundaries in the knowledge graph differ significantly depending on contextual usage. For the positive-negative sample pair, the LLM-driven clarification module generates the following discriminative question with high information gain.

[Clarification: Is the current mention discussing the layer in the context of its stratigraphic position within the Yaojia Formation, or is it being referenced as a specific reservoir unit with oil-bearing parameters such as porosity and permeability]

This question is embedded into the entity representation during training with probability $p_{clar} = 0.5$, enabling the model to learn that the interpretation of resource attribute requires association with quantitative parameters, whereas the interpretation of geological feature emphasizes stratigraphic relationships.

On this specific ambiguity type, VerbalizED frequently conflates the three semantic roles, often misclassifying mentions of the oil-bearing layer as generic geological features when they appear in contexts involving reserve quantification, or erroneously interpreting stratigraphic position references as drilling targets. In contrast, ActiveIter-ED successfully disambiguates these hierarchical roles by leveraging ontology-aware clarification questions: it correctly identifies the resource attribute interpretation when the context involves quantitative parameters and associated reserve calculations, distinguishes the target layer interpretation in drilling operation contexts, and recognizes the geological feature interpretation in stratigraphic discussions. The clarification sample augmentation mechanism enables the model to learn these subtle contextual cues through discriminative questioning, effectively resolving semantic drift that traditional dual-encoder architectures fail to capture despite processing identical surface forms.

DISCUSSION

The current method has the following limitations.

(1) **Dependence on entity description quality:** For entities with incomplete or poor-quality descriptions, the quality of generated clarification questions also decreases, affecting disambiguation effectiveness.

(2) **Clarification question generation quality:** While 85% of generated questions are high-quality, the remaining 15% are of medium or low quality—requiring further optimization of generation strategies.

(3) **Multilingual generalization:** The method has only been evaluated on English data; its effectiveness in multilingual scenarios (particularly multilingual clarification question generation) remains unvalidated.

Future work directions include: improving clarification question generation, introducing quality assessment mechanisms to automatically filter low-quality questions. Dynamic clarification question generation: we will explore generating questions in real time during training (instead of pre-generation) to enhance adaptability. Multilingual support: we will extend the method to multilingual and cross-lingual entity disambiguation, with tailored multilingual clarification question generation strategies.

CONCLUSION

This paper proposes ActiveIter-ED, an enhanced entity disambiguation framework based on active task disambiguation. Our core innovation is the clarification sample augmentation mechanism, generating discriminative clarification questions via large language models and dynamically embedding them into the training process to enable the model to distinguish easily confusable entity pairs.

(1) **Clarification sample augmentation mechanism:** A three-stage pipeline (hard negative sample discovery, LLM-driven clarification question generation, dynamic embedding during training) is designed—the first method to systematically embed clarification questions into entity disambiguation training.

(2) **Significant performance improvements:** Across 8 ZELDA test sets, ActiveIter-ED achieves an average 0.7-percentage-point improvement over the baseline, with the most notable gain (4.5 percentage points) on the AIDA-B dataset.

(3) **Systematic experimental validation:** The method's effectiveness is verified on standard benchmarks (ZELDA, AIDA-Yago), with improvements on 4 out of 8 test sets. Detailed case analysis demonstrates the practical value of clarification questions.

(4) **Domain knowledge graph construction application validation:** Experiments on geological datasets show that ActiveIter-ED improves entity linking accuracy from 62.6% to 67.1% and knowledge graph node correctness from 58.3% to 64.7%—significantly enhancing domain knowledge graph quality and providing technical support for terminology extraction and knowledge fusion.

Our work not only presents an effective technical solution for entity disambiguation but also offers practical insights for system design via systematic ablation studies. In the future, we will explore applications in multilingual and zero-shot scenarios, while further optimizing clarification question quality and computational efficiency.

Funding

This work was supported by the National Key Research and Development Program of China (2023YFC3305003) and the Key Research and Development Program of Heilongjiang (2024ZXJ01A040).

References

1 Chatzitheodorou K. KnowledgeTB: Leveraging large language models for enhanced terminology extraction over knowledge graphs. In: *Proceedings of the 4th International Conference on Multilingual Digital Terminology Today*. 2025, 3990.

2 Janowicz K, Hitzler P, Li WW, *et al*. Know, Know Where, KnowWhereGraph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence. *AI Mag* 2022; **43**: 30–

39.
3 Temmerman R. *Towards New Ways of Terminology Description: The Sociocognitive-Approach*. Amsterdam: John Benjamins Pub, 2000.
4 Kim S, Sung M, Lee J, *et al*. Efficient terminology integration for LLM-based translation in specialized domains. In: *Proceedings of the Ninth Conference on Machine Translation*. Miami, 2024, 636–642.
5 Giguere J. Leveraging large language models to extract terminology. In: *Proceedings of the First Workshop on NLP Tools and Resources for Translation and Interpreting Applications*. Varna, 2023, 57–60.
6 Kobalczyk K, Astorga N, Liu T, *et al*. Active task disambiguation with LLMs. arXiv: 2502.04485.
7 Wang JX, Mousavi A, Attia O, *et al*. Entity disambiguation via fusion entity decoding. arXiv: 2404.01626.
8 Févry T, FitzGerald N, Soares LB, *et al*. Empirical evaluation of pretraining strategies for supervised entity linking. arXiv: 2005.14253.
9 Vrandečić D, Krötzsch M. Wikidata: A free collaborative knowledgebase. *Commun ACM* 2014; **57**: 78–85.
10 Liu XK, Liu Y, Zhang K, *et al*. OneNet: A fine-tuning free framework for few-shot entity linking via large language model prompting. arXiv: 2410.07549.
11 Hanh TTH, González-Gallardo CE, Delaunay J, *et al*. Is prompting what term extraction needs? In: *Proceedings of Text, Speech, and Dialogue —27th International Conference*. Brno, 2024, 17–29.
12 McCrae JP, Ojha AK, Chakravarthi BR, *et al*. Enriching a terminology for under-resourced languages using knowledge graphs. In: *Proceedings of eLex 2021*. 2021, 560–571.
13 Babaei Giglou H, D'Souza J, Auer S. LLMs4OL: large language models for Ontology learning. In: *Proceedings of the Sematic web–ISWC 2023*. 2023, 408–427.
14 Wilkinson MD, Dumontier M, Aalbersberg IJ, *et al*. The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* 2016; **3**: 160018.
15 Broscheit S. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning*. Hong Kong, 2019, 677–685.
16 Ganea OE, Hofmann T. Deep joint entity disambiguation with local neural attention. arXiv: 1704.04920.
17 Gillick D, Kulkarni S, Lansing L, *et al*. Learning dense representations for entity retrieval. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning*. Hong Kong, 2019, 528–537.
18 Wu L, Petroni F, Josifoski M, *et al*. Scalable zero-shot entity linking with dense entity retrieval. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 2020, 6397–6407.
19 Procopio L, Conia S, Barba E, *et al*. Entity disambiguation with entity definitions. In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Dubrovnik, 2023, 1297–1303.
20 Rücker S, Akbik A. Evaluating design decisions for dual encoder-based entity disambiguation. arXiv: 2505.11683.
21 Mackay DJC. Bayesian methods for adaptive models. Dissertation for Ph.D. Degree. Pasadena: California Institute of Technology, 1992.
22 Houlsby N, Huszár F, Ghahramani Z, *et al*. Bayesian active learning for classification and preference learning. arXiv: 1112.5745.
23 Settles B. Active Learning Literature Survey. Technical Report TR1648. 2009.
24 Stengel-Eskin E, Guallar-Blasco J, Zhou Y, *et al*. Why did the chicken cross the road? Rephrasing and analyzing ambiguous questions in VQA. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Toronto, 2023, 10220–10237.
25 Liu A, Wu ZF, Michael J, *et al*. We're afraid language models aren't modeling ambiguity. arXiv: 2304.14399.
26 Creswell A, Shanahan M, Higgins I. Selection-inference: Exploiting large language models for interpretable logical reasoning. arXiv: 2205.09712.
27 Lei B, Lin PH, Liao CH, *et al*. Boosting logical reasoning in large language models through a new framework: The graph of thought. arXiv: 2308.08614.
28 Romera-Paredes B, Barekatain M, Novikov A, *et al*. Mathematical discoveries from program search with large language

models. *Nature* 2024; **625**: 468–475.

29 Imani S, Du L, Shrivastava H. MathPrompter: Mathematical reasoning using large language models. arXiv: 2303.05398.

30 Liu JW, Xia CS, Wang YY, *et al*. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 2023, 21558–21572.

31 Zhang S, Chen ZF, Shen YK, *et al*. Planning with large language models for code generation. arXiv: 2303.05510.

32 Coenen A, Davis L, Ippolito D, *et al*. Wordcraft: A human-AI collaborative editor for story writing. arXiv: 2107.07430.

33 Chakrabarty T, Padmakumar V, Brahman F, *et al*. Creativity support in the age of large language models: An empirical study involving emerging writers. arXiv: 2309.12570.

34 De Cao N, Izacard G, Riedel S, *et al*. Autoregressive entity retrieval. arXiv: 2010.00904.

35 Yamada I, Asai A, Shindo H, *et al*. LUKE: Deep contextualized entity representations with entity-aware self-attention. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 2020, 6442–6454.

36 Bartalesi V, Coro G, Lenzi E, *et al*. A semantic knowledge graph of european mountain value chains. *Sci Data* 2024; **11**: 978.