Discovering Latent Facts from Context to Construct Richer Open Knowledge Graphs

Jinpeng Li[1,2] Hang Yu[3]*, Ziqi Ma[3], Peng Qi[4]
[1]Deep Sea Science Data Center, National Deep Sea Center, Qingdao, China
[2]College of Environmental Science and Engineering, Ocean University of China, Qingdao, China
[3]School of Computer Engineering and Science, Shanghai University, Shanghai, China
[4]College of Electronics and Information Engineering, Tongji University, Shanghai, China
lijinpeng@ndsc.org.cn, {yuhang, ziqi_ma}@shu.edu.cn, pqi@tongji.edu.cn

Abstract

Knowledge graph construction (KGC) aims to extract valuable information from text and organize it into structured knowledge graphs (KGs). Recent methods have leveraged the strong generative capabilities of large language models (LLMs) to improve the generalization and reduce the labor costs. However, constrained by the input length of LLMs, existing methods mainly focus on extracting knowledge within individual texts and lack the capability to discover latent knowledge across texts. To fill this gap, we propose a novel method for open knowledge graph construction, termed KG-DLF. The core idea of this method is to enhance the knowledge graph construction process by discovering new facts that are consistent with the underlying contextual logic. Specifically, we first design a knowledge extractor to extract knowledge from the text. Then, a knowledge normalizer performs schema alignment on the extracted knowledge. Next, we explore a knowledge discoverer based on a clue search strategy, which leverages the logical consistency of context to mine latent facts. Finally, we design a counterfactual-based knowledge corrector, enabling the model to purify knowledge and reduce factual errors. Experimental results show that KG-DLF is capable of extracting comprehensive knowledge in open-world scenarios across three KGC benchmarks.

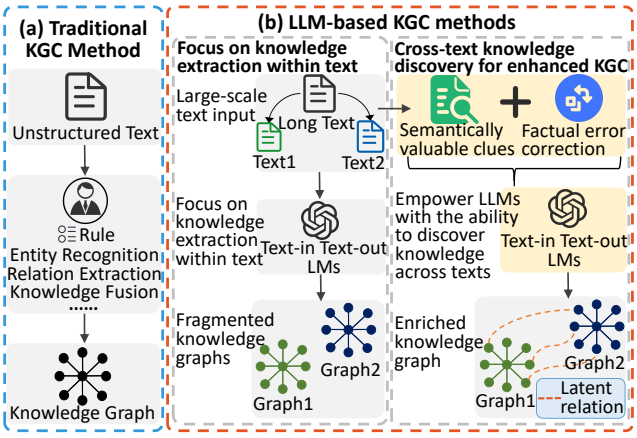


Figure 1: An example of enriched KGC: (a) Traditional methods rely on manual rules and feature engineering. (b) Example of LLM-based KGC: left graph — limited input leads to fragmented knowledge; right graph — semantically valuable clues and factual error correction empower the model to uncover latent cross-text knowledge.

Introduction

Knowledge Graph Construction (KGC) aims to extract, integrate, and organize information from various text to generate structured knowledge graphs (KGs) (Ji et al. 2021; Mesquita et al. 2019; Zhang et al. 2020). Since KGC requires strong syntactic and semantic understanding, traditional methods often follow a multi-step pipeline—including entity recognition (Cocchieri et al. 2025), relation extraction (Li et al. 2025), and knowledge fusion (Zhang et al. 2024)—as shown in Figure1(a). These methods rely on rules and feature engineering, which limits their generalization, increases error propagation, and leads to high construction costs—making them less suitable for large-scale open-domain scenarios.

Recently, large language models (LLMs) have demonstrated strong performance across various natural language processing tasks (Fang et al. 2024), making them the mainstream approach for KGC (Ye et al. 2022). By unifying heterogeneous tasks into a text-to-text format (Yao, Mao, and Luo 2019; Kim et al. 2020) and embedding extensive real-world knowledge from pretraining (Gouidis et al. 2024; Wang et al. 2023), LLMs offer significant advantages for KGC. Therefore, various innovative methods based on LLMs for KGC have been proposed, such as multi-turn dialogue (Wei et al. 2023), code generation (Bi et al. 2024), and pattern construction (Zhang and Soh 2024), to generate entity-relation facts that represent KGs.

However, as shown on the left side of Figure 1 (b), due to the limited input length that LLMs can process in a single inference, current methods for handling large-scale texts typically adopt a segmentation strategy—dividing a long text into multiple shorter segments (e.g., splitting it into Text1 and Text2) and extracting knowledge from each segment individually to construct corresponding subgraphs (Graph1 and Graph2). Although this strategy improves extraction efficiency, the independent processing of each paragraph overlooks semantic associations across texts, resulting in a lack of necessary connections between the con-

*Corresponding authors.

structed subgraphs. Consequently, the overall structure of the knowledge graph becomes fragmented and discontinuous. This phenomenon of knowledge fragmentation is particularly pronounced in cross-text scenarios.

We argue that discovering latent knowledge across texts relies on two key factors: semantically valuable clues and factual error correction, as shown on the right side of Figure 1 (b). The former helps reduce hallucination in model reasoning, while the latter ensures the plausibility of the inferred latent knowledge. Incorporating these two factors enhances LLMs' ability to uncover cross-text knowledge, enabling the inference of latent relations—such as those between Graph1 and Graph2 highlighted by the orange dashed lines in the figure—thus alleviating knowledge fragmentation caused by input length limitations.

Based on the above motivation, we propose KG-DLF, a method to enhance **K**nowledge **G**raph construction by **D**iscovering **L**atent **F**acts that are logically consistent with the context. It consists of four main components: a knowledge extractor, a knowledge normalizer, a knowledge discoverer, and a knowledge corrector. The knowledge extractor is responsible for extracting structured knowledge from raw text and defining its schema. The knowledge normalizer aligns schemas through semantic similarity to mitigate terminological ambiguity. The knowledge discoverer leverages the real-world knowledge modeling capabilities of LLMs and a clue retrieval strategy to guide the generation of contextually coherent latent facts. The knowledge corrector performs automatic correction of knowledge through counterfactual comparison. The main contributions of our work are summarized as follows:

- We introduce the KG-DLF model, which is capable of extracting intra-text knowledge as well as uncovering latent inter-text knowledge. To the best of our knowledge, it is the first model designed to enhance the KGC process through the discovery of latent knowledge.
- We propose a knowledge corrector that emulates human reflective behavior through counterfactual comparison, allowing for autonomous correction of knowledge.
- Extensive experiments verify the effectiveness of each module and its superiority over state-of-the-art baselines.

Related Work

Traditional Knowledge Graph Construction

Traditional knowledge graph construction methods typically follow a modular, pipeline-based architecture, where the process is divided into a sequence of subtasks, including entity extraction, relation extraction, and knowledge fusion. Entity extraction aims to identify entities such as persons, organizations, and locations from raw text. Early studies relied on rule-based templates (Li, Li, and Gao 2014) and statistical machine learning models (Lu et al. 2016). With the advancement of deep learning, methods based on BiLSTM-CRF (Greenberg et al. 2018) and pretrained language models like BERT (Li et al. 2021) have significantly improved the accuracy of entity recognition. Relation extraction focuses on identifying semantic relationships between entity pairs. Traditional approaches include feature-based supervised models (Zeng et al. 2014), which automatically aligns text with knowledge bases to generate training data. However, distant supervision suffers from noisy labels. To address this, joint models (Huo et al. 2023; Pu et al. 2024) attempt to learn entities and relations simultaneously, though most still follow a sequential learning pipeline. Knowledge fusion involves resolving redundant or conflicting information, primarily through entity resolution and ontology matching. Tools like OpenRefine (Ham 2013) and frameworks such as HMAN (Yang et al. 2019) and embedded networks (He et al. 2021; Jin et al. 2021) use similarity-based heuristics to align entities across heterogeneous sources. While effective to some extent, these methods often rely on manual rules and lack scalability (Jin et al. 2023). Despite their interpretability, pipeline-based KGC systems suffer from error propagation, lack end-to-end optimization, and require extensive task-specific annotation, limiting scalability and generalization in open-domain settings.

LLM-based Knowledge Graph Construction

To overcome the limitations of traditional pipeline-based methods, which heavily rely on manual rules and feature engineering, recent research has begun leveraging generative models for KGC. Thanks to advances in pre-trained generative language models (e.g., T5 (Raffel et al. 2020) and BART (Lewis 2019)), more recent works frame KGC as a sequence-to-sequence problem and generate relational facts in an end-to-end manner by fine-tuning moderately-sized models, such as REGEN (Dognin et al. 2021) and GenIE (Josifoski et al. 2022). The success of LLMs has pushed this paradigm further: current methods directly prompt the LLMs to generate facts in a zero/few-shot manner. For example, EDC+R (Zhang and Soh 2024) extracts facts by framing the task as a multi-turn question-answering problem, SAC-KG (Chen et al. 2024) introduces a multi-agent paradigm to construct hierarchical knowledge graphs, while AutoSchemaKG (Bai et al. 2025) formulates the task as a co-evolution problem. As mentioned earlier, due to LLM input length limits, existing models often split large texts into paragraphs and extract knowledge independently from each. This ignores semantic links between texts, causing the resulting subgraphs to lack connections and leading to a fragmented knowledge graph. KG-DLF addresses this by discovering latent cross-text knowledge without fine-tuning the base LLM, producing a more coherent and integrated graph.

Preliminaries

Let $\mathcal{T}$, $\mathcal{F}$, $\mathcal{E}$ and $\mathcal{R}$ denote a finite set of texts, facts, entities, and relations, respectively. The task of KGC is to extract a valid set of facts $\mathcal{F}$ from a given text $t \in \mathcal{T}$. A fact list $\mathcal{F} = \{f_1, \ldots, f_n\} = \{(s, r, o)_1, ..., (s, r, o)_n\}$ where s, $o \in \mathcal{E}$ are the subject and object entities, and $r \in \mathcal{R}$ is the relation between them. The LLMs have transformed the representation of KGC (Zhang and Soh 2024), which can be viewed as $\mathrm{P}(y|x) = \prod_{i=1}^{|y|} \mathrm{P}(y_i|y_{<i}, x)$, where x is the input text and y represents the extracted facts. It is noteworthy that both x and y are flattened textual representations of

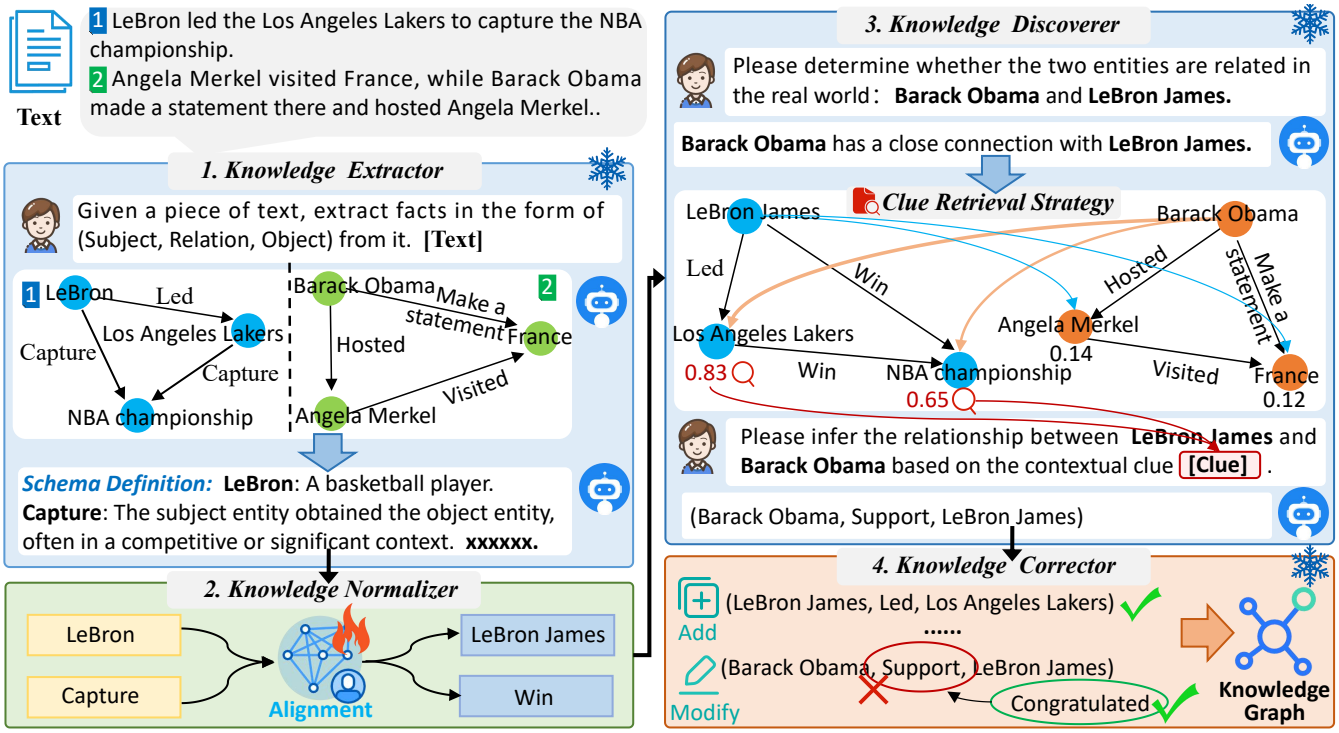


Figure 2: An overview of the KG-DLF. The knowledge extractor extracts knowledge; the knowledge normalizer standardizes knowledge; the knowledge discoverer uncovers latent cross-text knowledge; and the knowledge corrector enables self-modification of knowledge.

knowledge (Li et al. 2024). Based on this, we extended our research on using LLMs for the KGC task.

Methodologies

Figure 2 shows the overview of KG-DLF. Unlike traditional KGC methods, it not only effectively extracts and normalizes knowledge, but also discovers latent cross-text knowledge and performs automatic correction.

Knowledge Extractor

To help LLMs quickly adapt to the extraction task, we provide a few examples to guide their understanding of the task requirements. Specifically, given a text and a few examples, prompts guide the LLMs to extract facts. We designed a prompt template:

> *Given a piece of text, extract facts in the form of (Subject, Relation, Object) from it.*
> *Here are some examples:* `[EXP]`.
> *Please extract facts from the following text:* `[t]`.

In this prompt, `[EXP]` represents a placeholder for a few examples of ICL (In-Context Learning (Rubin, Herzig, and Berant 2022)), and `[t]` represents a placeholder for the text to be extracted. An example: *Text: LeBron led the Los Angeles Lakers to capture the NBA championship. Facts: [('LeBron', 'Led', 'Los Angeles Lakers'), ('Los Angeles Lakers', 'Capture', 'NBA championship')].*

Next, we define the schema for each element (entity or relation) of the fact:

> *Given a piece of text and a list of elements, your task is to write a description for each element based on the text.*
> *Text:* `[t]`.*Elements:* `[ELE]`.

In this prompt, `[ELE]` is a placeholder for the element in the facts. It is worth noting the initially extracted knowledge fails to capture facts that span across multiple texts.

Knowledge Normalizer

We designed a knowledge normalizer to ensure consistency in knowledge descriptions. To achieve this, we first establish a predefined set of schemas $S = (s_1, s_2, \ldots, s_m)$, where m represents the total number of schemas, which include entity and relation types. All extracted data is then structured according to these schemas, which guarantees uniformity and standardization across the knowledge graph.

Here, we leverage a BERT-based embedding model $\Phi(\cdot)$ to obtain the representations of the schemas:

$$\mathbf{h}_{\hat{s}} = \Phi(\hat{s}), \mathbf{h}_{s_i} = \Phi(s_i), \quad (1)$$

where $\hat{s}$ and $s_i \in S$ represent the newly extracted schema and the predefined schema, respectively. $\mathbf{h} \in \mathbb{R}^d$ represents a d-dimensional embedding vector.

The semantic similarity between the new schema and each cluster is calculated based on cosine similarity:

$$\text{sim}(\hat{s}, s_i) = \cos(\mathbf{h}_{\hat{s}}, \mathbf{h}_{s_i}) = (\mathbf{h}_{\hat{s}}^\top \mathbf{h}_{s_i}) / (\|\mathbf{h}_{\hat{s}}\| \, \|\mathbf{h}_{s_i}\|). \quad (2)$$

The schema that is semantically closest to $\hat{s}$ is obtained through the argmax function:

$$s^* = \arg\max_{s_i \in S} \text{sim}\left(\hat{s}, s_i\right), \tag{3}$$

where s^* is the schema in the predefined set that has the highest semantic similarity to $\hat{s}$. if $\text{sim}\left(\hat{s}, s^*\right) > \lambda$, then align $\hat{s}$ with s^*. Here, $\lambda > 0$ denotes the similarity threshold.

To better capture the fine-grained semantic differences between schemas, we train the embedding model $\Phi\left(\cdot\right)$ using a contrastive loss:

$$\begin{aligned}\mathcal{L} = \max\big(0, \text{sim}(\Phi(s_i), \Phi(s_j^-)) \\ - \text{sim}(\Phi(s_i), \Phi(s_j^+)) + \alpha\big),\end{aligned} \tag{4}$$

where $s_j^+ \in S$ is the positive schema sample of s_i, while $s_j^- \in S$ is the negative schema sample of s_i. We set the margin $\alpha = 0.2$ following Garg, Vu, and Moschitti (2020), and use the method proposed by Zhang et al. (Zhang and Soh 2024) for filtering positive and negative samples.

Knowledge Discoverer

The goal is to discover potential knowledge associations in order to enrich the knowledge graph. However, not all entities are inherently related. Here, we prompt LLMs to identify entities with objective connections in the real world:

Please determine whether the entities are related in the real world.`[SUB].[OBJ].`

In this prompt, `[SUB]` and `[OBJ]` denote the subject along with its schema and the object entity along with its schema, respectively. Next, we perform completion on the entity pairs identified as relevant by the LLM. Unlike previous methods that rely on reasoning over a predefined relation set, we use the LLM's real-world knowledge to directly generate possible relationships. However, in the absence of reasoning conditions, the relations directly generated by the LLM may be inconsistent with the contextual logic of the existing KG.

To enhance the logical structure of KGs, we propose a clue retrieval strategy that extracts relevant paths for a given entity pair, providing contextual constraints to guide LLM reasoning. Specifically, given a subject entity and an object entity, we compute the relevance between the subject entity and each node within the l-hop neighborhood of the object entity. Similarly, the relevance between the object entity and each node within the l-hop neighborhood of the subject entity is also computed. The relevance scores are generated by prompting LLM to evaluate the semantic associations between the entities, with the following prompt:

Please provide the relevance scores between the two entities within the range of 0 to 1, where scores closer to 0 indicate lower relevance, and scores closer to 1 indicate higher relevance. `[SUB].[OBJ].`

The score of each path p_i is the average relevance score of all the entities within the path, as follows:

$$\text{conf}_X(p_i) = (1/l) \sum_{j=1}^{l} \psi(X, e_j),\ e_j \in \mathcal{N}_l(Y), \tag{5}$$

where $X \in \{s, o\}$ and $Y \in \{s, o\}$ are the corresponding entities in an entity pair. $\text{conf}_X(p_i)$ denotes the confidence score of path p_i, and $\psi\left(\cdot\right)$ is the relevance scoring function computed by the LLM. $\mathcal{N}_l\left(Y\right)$ represents the set of all entities within an l-hop neighborhood of entity Y and $l = 2$. e_j represents a neighbor of entity Y.

Then, select the Top-k paths with the highest average relevance scores as contextual clues $\mathcal{P}_X^{(k)}$, as follows:

$$\mathcal{P}_X^{(k)} = \text{Top}-k\left(\{\text{conf}_X(p_i) \mid p_i \in \mathcal{P}_X\}\right). \tag{6}$$

Finally, based on the retrieved contextual clues, the LLM is guided to perform relation reasoning:

Please infer the relationship between `[SUB]` *and* `[OBJ]` *based on the contextual clue.* `[CLUE].`

In this prompt, `[CLUE]` represents a placeholder for the contextual clues. This allows the LLM to generate relations that are coherent with the surrounding context.

Knowledge Corrector

To evaluate the plausibility of knowledge, inspired by counterfactual thinking (Roese 1994), we designed a knowledge corrector. It constructs a set of counterfactuals for each fact and assesses plausibility by comparing the original fact with its counterfactuals. It is worth noting that the counterfactuals are generated by replacing the relation in the original fact.

Based on this, given a fact $f_i = (s, r, o)$ and its corresponding counterfactual set $\hat{\mathcal{F}} = \left\{\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n\right\}$, under the premise that the subject s and object o in f_i necessarily exists, we can derive the following two inferences:

(1) **If the plausibility of f_i is higher than all the counterfactual facts in $\hat{\mathcal{F}}$, then f_i is considered correct.**

The derivation is:

$\text{P}\left(f_i\right) = \max\left(\text{P}\left(f_i\right), \text{P}\left(\hat{f}_1\right), \text{P}\left(\hat{f}_2\right), \ldots, \text{P}\left(\hat{f}_n\right)\right) \wedge \exists r \in f_i \Rightarrow \exists f_i \wedge f_i = \top$, where $\text{P}\left(\cdot\right)$ represents the probability of a fact being reasonable, with higher values indicating higher plausibility;

(2) **If f_i is incorrect, then at least one counterfactual fact in $\mathcal{F}$ will have higher plausibility than f_i.**

The derivation is:

$\exists r \in f_i \wedge f_i = \bot \wedge \text{P}\left(f_i\right) < \text{P}\left(\hat{f}_j \in \mathcal{F}\right) \Rightarrow \exists \hat{f}_j \wedge \hat{f}_j = \top$, where $\top$ and $\bot$ represent correct and incorrect, respectively.

For the plausibility of the facts, we guide the LLMs to make judgments through prompts, as follows:

Your task is to select the most reasonable fact from the list based on the context. Context: `[C]`*.Facts:* `[F].`

In this prompt, `[C]` and `[F]` represent placeholders for the context and the facts (i.e., the set composed of facts and counterfactuals), respectively. Notably, extracted facts use the original text as context, while discovered facts rely on the top-k retrieved relevant paths.

Experiments

In this section, we evaluate KG-DLF on three public datasets. We first describe the experimental setup and baselines, then present the results. Ablation studies are conducted to assess the contribution of different components, and real-world scenarios cases illustrate its effectiveness in cross-text knowledge discovery and correction.

Datasets

We evaluate KG-DLF on the task of behavior prediction using three benchmark datasets, as follows:

- **WebNLG*** (Ferreira et al. 2020). The WebNLG (Web Natural Language Generation) dataset offers DBpedia facts paired with descriptive text, focusing on converting structured data into natural language. We use the test split from WebNLG+2020 (v3.0), which includes 1,165 text-fact pairs, 4,001 facts, 345 unique entities, and 159 unique relations.
- **REBEL*** (Cabot and Navigli 2021). The REBEL (Relation Extraction By End-to-end Language Generation) dataset is a benchmark for extracting complex relation facts with long-distance dependencies. From its 105,516 test entries, we randomly sample 1,000 long-text fact pairs, containing 4,000 facts, 3,643 unique entities, and 196 unique relations.
- **Wiki-NRE*** (Distiawan et al. 2019). The Wiki-NRE (Wikipedia-based Neural Relation Extraction) dataset pairs Wikipedia text with knowledge graph relations to extract entity relationships from unstructured text, including long-tail cases. From its 29,619 test entries, we randomly sampled 1,000 cross-text fact instances composed of more than two texts, covering 2,335 unique entities and 45 unique relations.

Baseline Methods

We select two types of baseline models for comparison:

- Some general large models (GLMs) utilize their own understanding capabilities to extract facts present in the text through a question-and-answer interaction method, with deduplication added to avoid fact redundancy. We selected five popular GLMs, including **Mistral-7BInstruct** (Jiang et al. 2023), **LLaMA3-8B-Instruct** (Vavekanand and Sam 2024), **ChatGPT-3.5** (Brown et al. 2020), **ChatGPT-4.0**[1].
- Recent popular generative construction methods (GCMs) guide generative models to perform extraction tasks by designing various prompt and adapt to specific datasets by incorporating certain fine-tuning techniques, including **REGEN** (Dognin et al. 2021), **GenIE** (Josifoski et al. 2022), **EDC+R** (Zhang and Soh 2024), **SAC-KG** (Chen et al. 2024) and **AutoSchemaKG** (Bai et al. 2025).

Evaluation Metrics and Environment Settings

We used the WEBNLG evaluation script (Ferreira et al. 2020), which calculates the F1 score, exact precision, partial precision, and strict precision of the output facts against the ground truth in a token-based manner.

- **F1**: The extracted facts are required to completely match the ground truth, and the F1 score is calculated based on precision and recall.
- **Exact**: Requires an exact match between extracted facts and ground truth, ignoring element types.
- **Partial**: Allows for at least a partial match between the extracted facts and the ground truth, disregarding the element types.
- **Strict**: Demands an exact match between the extracted facts and the ground truth, including the element types.

In our experiments, we implemented KG-DLF using PyTorch (Paszke et al. 2019) in collaboration with HuggingFace (Wolf et al. 2020) and evaluated its performance on a 22-core CPU (AMD EPYC 7T83) and two GPUs (RTX-4090) with 24GB of memory each. Please note that all experimental results are averaged over three runs.

Experimental Results

The experimental results are shown in Table 1. For the datasets WebNLG*, REBEL*, and Wiki-NRE*, the best models based on LLMs were selected for the experiments. The aim is to observe whether KG-DLF can adapt to KGC tasks in different scenarios. Overall, KG-DLF achieved state-of-the-art performance and significantly enhanced the KGC performance of LLMs (LLaMA3-8B-Instruct, chatgpt-3.5, chatgpt-4.0). This confirms that discovering cross-text knowledge is a key factor in improving the quality of KGs.

Specifically, on the single-text WebNLG* dataset, KG-DLF improved the F1 score by 1.9% (from 0.826 to 0.845) compared to the best baseline model, EDC+R. This confirms that KG-DLF is not only designed for cross-text scenarios but also demonstrates strong competitiveness in non-cross-text settings. Although the improvements in extraction-related metrics such as F1, Partial, and Exact are limited, we observed a notable gain in the Strict metric, indicating that the discovery of latent knowledge contributes to a more complete and structurally sound knowledge graph.

On the REBEL* dataset, which involves complex long-range dependencies, KG-DLF outperformed the best baseline model, EDC+R, by a margin of 4.6% in F1 score (from 0.602 to 0.648), showing a more significant improvement than on WebNLG*. The Strict metric analysis indicates that long-range relations overlooked in the initial LLM extraction were effectively recovered through the knowledge discovery process, demonstrating the model's capacity to complete missing information.

Surprisingly, On the cross-text Wiki-NRE* dataset, which features a long-tail relation distribution, KG-DLF achieved the largest improvement over the best baseline, AutoSchemaKG, increasing the F1 score by 8.6% (from 0.714 to 0.800). This demonstrates that KG-DLF effectively mitigates the challenges of cross-text knowledge extraction under limited input. Moreover, during the process of self-correction, the introduction of counterfactuals inadvertently

[1]https://openai.com/index/gpt-4-research/

	Models	WebNLG*				REBEL*				Wiki-NRE*			
		F1	Partial	Strict	Exact	F1	Partial	Strict	Exact	F1	Partial	Strict	Exact
GLMs	Mistral-7B-Instruct	0.216	0.222	0.212	0.222	0.222	0.344	0.333	0.333	0.292	0.411	0.402	0.410
	LLaMA3-8B-Instruct	0.531	0.697	0.661	0.680	0.459	0.738	0.703	0.726	0.616	0.678	0.674	0.677
	ChatGPT-3.5	0.736	0.712	0.675	0.667	0.305	0.692	0.660	0.680	0.588	0.704	0.700	0.702
	ChatGPT-4.0	0.720	0.733	0.708	0.715	0.415	0.621	0.754	0.699	0.572	0.692	0.691	0.694
GCMs	REGEN	0.723	0.714	0.755	0.713	0.443	0.476	0.415	0.488	0.493	0.684	0.571	0.652
	GenIE	0.673	0.706	0.692	0.710	0.471	0.453	0.447	0.510	0.505	0.675	0.562	0.664
	SAC-KG	0.725	0.681	0.675	0.697	0.463	0.460	0.435	0.467	0.655	0.639	0.612	0.633
	EDC+R	0.826	0.794	0.753	0.772	0.602	0.559	0.516	0.529	0.687	0.647	0.638	0.640
	AutoSchemaKG	0.815	**0.802**	0.761	0.770	0.597	0.534	0.508	0.536	0.714	0.693	0.685	0.657
	KG-DLF$_{Mistral-7B}$	0.211	0.220	0.204	0.205	0.369	0.680	0.636	0.652	0.494	0.594	0.583	0.490
	KG-DLF$_{LLaMA3-8B}$	0.782	0.746	0.688	0.713	**0.648**	0.700	0.700	0.700	0.556	0.500	0.500	0.500
	KG-DLF$_{ChatGPT-3.5}$	0.813	0.795	0.773	0.762	0.630	0.736	0.743	0.747	**0.800**	0.754	0.731	**0.762**
	KG-DLF$_{ChatGPT-4.0}$	**0.845**	0.798	**0.782**	**0.787**	0.639	**0.785**	**0.782**	**0.793**	0.771	**0.769**	**0.768**	0.742

Table 1: The main experimental results are shown. The best results are in bold; the second-best are underlined.

enriches the semantics of long-tail relations from a global perspective.

Additionally, the choice of LLMs is particularly important, as it indicates that some LLMs struggle to meet the basic requirements for KGC, such as: Mistral-7B-Instruct, which leads to the generation of hallucinated results.

Ablation Study

In this section, we explore different combinations of model components, including the Knowledge Extractor (KE), the Knowledge Normalizer (KN), the Knowledge Discoverer (KD) and the Knowledge Corrector (KC), and apply them to different construction scenarios. It is worth noting that we built KG-DLF based on the ChatGPT-4.0, as it is well-suited for various KGC tasks.

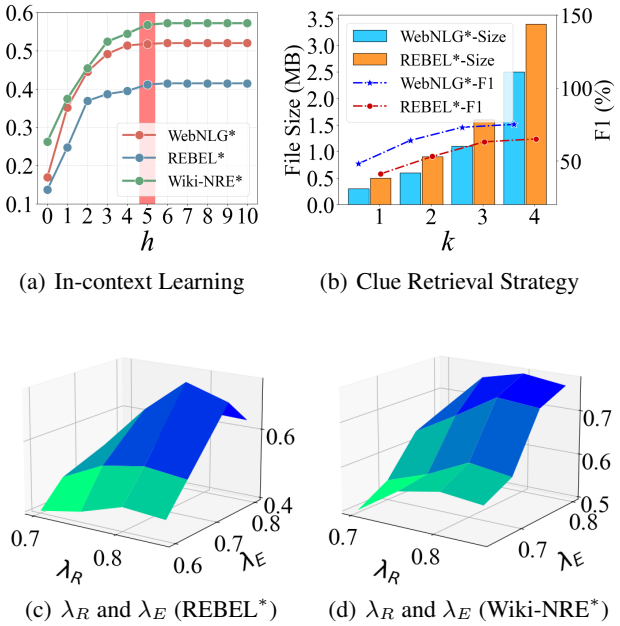
The experimental results are shown in Table 2. These three components are crucial to the entire framework, with KE being indispensable, as it forms the foundation of the whole KGC process. We observed that the F1 score of KN is higher than that of KD and KC because it leverages the predefined schema to ensure term consistency, while also retaining the core extraction capabilities of KE. Additionally, we found that KD outperforms KN in certain structural metrics, such as Partial, Strict, and Exact, particularly when handling the complex semantics of the REBEL* and Wiki-NRE* datasets. This indirectly suggests that although KN enhances the overall semantic consistency of the knowledge graph to a certain extent, it still lacks the ability to assess the quality of knowledge, thereby leaving room for KC to further demonstrate its advantages.

Model Analysis

In this section, we provide a detailed analysis of the effectiveness of KG-DLF in terms of model hyperparameter settings, few-shot ICL, and the number of clues.

Firstly, we validated the impact of the number of ICL examples on the performance of LLM in knowledge extraction, using the F1 score as the evaluation metric, as shown in Figure 3 (a). Generally, as the number of ICL examples h increases, the extraction performance of the model gradually improves. However, after reaching a certain point, performance peaks, and adding more examples no longer makes a difference. We were surprised to find that the optimal number of ICL examples did not vary with dataset characteristics. For all three datasets—WebNLG*, REBEL*, and Wiki-NRE*—setting h=5 tends to stabilize performance. This indicates that the LLMs only require a small number of ICL examples to guide the unification of extraction structures.

Secondly, the number of clues k provided by the knowl-



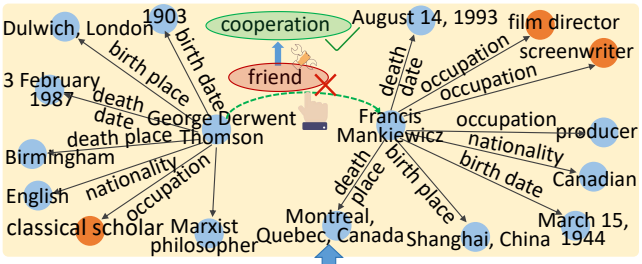
(a) In-context Learning (b) Clue Retrieval Strategy

(c) λ_R and λ_E (REBEL*) (d) λ_R and λ_E (Wiki-NRE*)

Figure 3: The effects of different parameter settings on performance. We use ChatGPT-4.0 as the base model and F1 as the evaluation metric.

Ablation	KN	KD	KC	WebNLG*				REBEL*				Wiki-NRE*			
				F1	Partial	Strict	Exact	F1	Partial	Strict	Exact	F1	Partial	Strict	Exact
1	✓			0.720	0.733	0.708	0.715	0.415	0.621	0.754	0.699	0.572	0.692	0.691	0.694
2		✓		0.424	0.305	0.367	0.465	0.381	0.585	0.583	0.582	0.416	0.480	0.484	0.482
3			✓	0.515	0.501	0.632	0.575	0.412	0.554	0.579	0.581	0.442	0.567	0.518	0.556
4	✓	✓		0.725	0.720	0.774	0.733	0.633	0.735	0.775	0.782	0.766	0.750	0.754	0.735
5	✓		✓	0.842	**0.823**	0.778	**0.792**	0.618	0.755	0.763	0.785	0.761	0.744	0.738	0.732
6		✓	✓	0.520	0.482	0.466	0.535	0.435	0.655	0.631	0.602	0.516	0.580	0.524	0.568
KG-DLF	✓	✓	✓	**0.845**	0.798	**0.782**	0.787	**0.639**	**0.785**	**0.782**	**0.793**	**0.771**	**0.769**	**0.768**	**0.742**

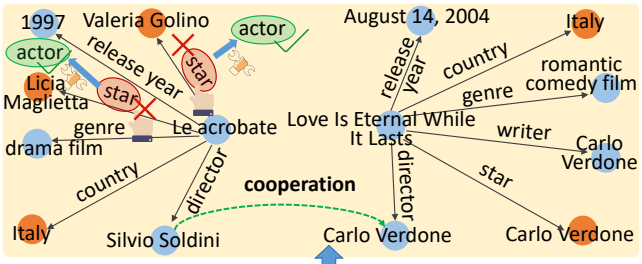
Table 2: Experimental results of the ablation study. The foundational LLM for KG-DLF is ChatGPT-4.0.



Original Unstructured Text

1. Francis Mankiewicz (March 15, 1944 in Shanghai, China – August 14, 1993 in Montreal, Quebec, Canada) was a Canadian film director, screenwriter and producer. 2. George Derwent Thomson was an English classical scholar and Marxist philosopher.

(a) A real-world case from REBEL*



Original Unstructured Text

1. Love Is Eternal While It Lasts is a 2004 Italian romantic comedy film written, directed and starred by Carlo Verdone. 2. Le acrobate is a 1997 Italian drama film directed by Silvio Soldini and starring Licia Maglietta and Valeria Golino.

(b) A real-world case from Wiki-NRE*

Figure 4: Case studies of latent knowledge discovery and knowledge correction in REBEL* and Wiki-NRE*.

edge discoverer to the LLM directly affects its reasoning capability, as shown in Figure 3 (b). Notably, as k increases, the F1 score for discovering latent knowledge also improves. However, supplying more clues imposes a heavier burden on the LLM. We set k to 3, as it achieves a good balance between performance and memory usage. Although $k = 4$ yields slightly better accuracy, it results in exponential growth in storage requirements, averaging 3.3MB across the three datasets. In contrast, $k = 3$ averages only 1.5MB, making it a more practical choice.

Finally, we evaluated the impact of the similarity threshold λ on KGC performance (measured by F1) across REBEL* and Wiki-NRE*, as shown in Figure 3 (c)–(d).Two thresholds were considered—entity schema (λ_E) and relation schema (λ_R)—which were tuned jointly. Results show that performance improves with increasing thresholds up to a critical point, after which it declines. Experimental results on the REBEL* and Wiki-NRE* datasets suggest a common optimal configuration for λ_E and λ_R across different datasets, with $\lambda_E = 0.80$ and $\lambda_R = 0.75$ emerging as consistently effective settings.

Case Study

The real-world case studies shown in Figure 4 illustrate the two core capabilities of KG-DLF in Knowledge Graph Construction (KGC): discovering latent knowledge and correcting erroneous knowledge. In the REBEL* case, the knowledge discoverer identified a strong connection between *George Derwent Thomson* and *Francis Mankiewicz*, inferring a friend relationship based on contextual clues (orange nodes). However, the knowledge corrector determined that *cooperation* better reflects their actual relationship, thereby revising the extracted knowledge accordingly. In the Wiki-NRE* case, the discoverer successfully complemented the knowledge graph with a *cooperation* relation between *Silvio Soldini* and *Carlo Verdone*. Nevertheless, the corrector detected two errors in the constructed graph, primarily caused by misinterpreting *star* as a relational link between a person and a work.

Conclusion

For the task of open-domain KGC, we propose KG-DLF. Unlike previous methods, we not only effectively extract and normalize knowledge, but also discover latent cross-text knowledge and automatically correct it. Specifically, we first design a knowledge extractor to extract knowledge from the text. Second, a knowledge normalizer performs schema alignment on the extracted knowledge. Next, we design a knowledge discoverer, which leverages the logical consistency of context to mine latent facts. Finally, we design a knowledge corrector, enabling the model to purify knowledge. Experimental results show that KG-DLF can significantly enrich the knowledge graph across three KGC benchmarks without requiring the LLM to be trained.

Acknowledgments

This work is supported by National Natural Science Foundation of China (GrantNo.62302287) and projects of the Shanghai Committee of Science and Technology, China (GrantNo.23ZR1423500).

References

Bai, J.; Fan, W.; Hu, Q.; Zong, Q.; Li, C.; Tsang, H. T.; Luo, H.; Yim, Y.; Huang, H.; Zhou, X.; et al. 2025. AutoSchemaKG: Autonomous Knowledge Graph Construction through Dynamic Schema Induction from Web-Scale Corpora. *arXiv e-prints*, arXiv–2505.

Bi, Z.; Chen, J.; Jiang, Y.; Xiong, F.; Guo, W.; Chen, H.; and Zhang, N. 2024. Codekgc: Code language model for generative knowledge graph construction. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(3): 1–16.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *CoRR*, abs/2005.14165.

Cabot, P.-L. H.; and Navigli, R. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2370–2381.

Chen, H.; Shen, X.; Lv, Q.; Wang, J.; Ni, X.; and Ye, J. 2024. SAC-KG: Exploiting Large Language Models as Skilled Automatic Constructors for Domain Knowledge Graph. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4345–4360.

Cocchieri, A.; Galindo, M. M.; Frisoni, G.; Moro, G.; Sartori, C.; and Tagliavini, G. 2025. ZeroNER: Fueling Zero-Shot Named Entity Recognition via Entity Type Descriptions. In *Findings of the Association for Computational Linguistics: ACL 2025*, 15594–15616.

Distiawan, B.; Weikum, G.; Qi, J.; and Zhang, R. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 229–240.

Dognin, P.; Padhi, I.; Melnyk, I.; and Das, P. 2021. ReGen: Reinforcement Learning for Text and Knowledge Base Generation using Pretrained Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1084–1099.

Fang, X.; Xu, W.; Tan, F. A.; Hu, Z.; Zhang, J.; Qi, Y.; Sengamedu, S. H.; and Faloutsos, C. 2024. Large Language Models (LLMs) on Tabular Data: Prediction, Generation, and Understanding - A Survey. volume 2024.

Ferreira, T. C.; Gardent, C.; Ilinykh, N.; Van Der Lee, C.; Mille, S.; Moussallem, D.; and Shimorina, A. 2020. The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (webnlg+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*.

Garg, S.; Vu, T.; and Moschitti, A. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 7780–7788.

Gouidis, F.; Papantoniou, K.; Papoutsakis, K.; Patkos, T.; Argyros, A.; and Plexousakis, D. 2024. Fusing domain-specific content from large language models into knowledge graphs for enhanced zero shot object state classification. In *Proceedings of the AAAI Symposium Series*, volume 3, 115–124.

Greenberg, N.; Bansal, T.; Verga, P.; and McCallum, A. 2018. Marginal likelihood training of BiLSTM-CRF for biomedical named entity recognition from disjoint label sets. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2824–2829.

Ham, K. 2013. OpenRefine (version 2.5). http://openrefine.org. Free, open-source tool for cleaning and transforming data. *Journal of the Medical Library Association: JMLA*, 101(3): 233.

He, D.; Wang, T.; Zhai, L.; Jin, D.; Yang, L.; Huang, Y.; Feng, Z.; and Yu, P. S. 2021. Adversarial representation mechanism learning for network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 35(2): 1200–1213.

Huo, C.; Jin, D.; Li, Y.; He, D.; Yang, Y.-B.; and Wu, L. 2023. T2-gnn: Graph neural networks for graphs with incomplete features and structure via teacher-student distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4339–4346.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Philip, S. Y. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2): 494–514.

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

Jin, D.; Feng, B.; Guo, S.; Wang, X.; Wei, J.; and Wang, Z. 2023. Local-global defense against unsupervised adversarial attacks on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8105–8113.

Jin, D.; Yu, Z.; He, D.; Yang, C.; Yu, P. S.; and Han, J. 2021. GCN for HIN via implicit utilization of attention and meta-paths. *IEEE Transactions on Knowledge and Data Engineering*, 35(4): 3925–3937.

Josifoski, M.; De Cao, N.; Peyrard, M.; Petroni, F.; and West, R. 2022. GenIE: Generative Information Extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4626–4643.

Kim, B.; Hong, T.; Ko, Y.; and Seo, J. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, 1737–1743.

Lewis, M. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Li, B.; Miao, Y.; Wang, Y.; Sun, Y.; and Wang, W. 2021. Improving the efficiency and effectiveness for bert-based entity resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 13226–13233.

Li, J.; Yu, H.; Luo, X.; and Liu, Q. 2024. COSIGN: Contextual Facts Guided Generation for Knowledge Graph Completion. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 1669–1682.

Li, L.; Li, J.; and Gao, H. 2014. Rule-based method for entity resolution. *IEEE Transactions on Knowledge and Data Engineering*, 27(1): 250–263.

Li, Y.; Miao, X.; Zhou, S.; Xu, M.; Ren, Y.; and Qian, T. 2025. Enhancing Relation Extraction via Supervised Rationale Verification and Feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 24521–24529.

Lu, J.-L.; Kato, M. P.; Yamamoto, T.; and Tanaka, K. 2016. Entity identification on microblogs by CRF model with adaptive dependency. *IEICE TRANSACTIONS on Information and Systems*, 99(9): 2295–2305.

Mesquita, F.; Cannaviccio, M.; Schmidek, J.; Mirza, P.; and Barbosa, D. 2019. Knowledgenet: A benchmark dataset for knowledge base population. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 749–758.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 8026–8037.

Pu, R.; Li, Y.; Zhao, J.; Wang, S.; Li, D.; Liao, J.; and Zheng, J. 2024. A joint framework with heterogeneous-relation-aware graph and multi-channel label enhancing strategy for event causality extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18879–18887.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.

Roese, N. J. 1994. The functional basis of counterfactual thinking. *Journal of personality and Social Psychology*, 66(5): 805.

Rubin, O.; Herzig, J.; and Berant, J. 2022. Learning To Retrieve Prompts for In-Context Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2655–2671.

Vavekanand, R.; and Sam, K. 2024. Llama 3.1: An In-Depth Analysis of the Next-Generation Large Language Model.

Wang, L.; Xu, W.; Lan, Y.; Hu, Z.; Lan, Y.; Lee, R. K.-W.; and Lim, E.-P. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2609–2634.

Wei, X.; Cui, X.; Cheng, N.; Wang, X.; Zhang, X.; Huang, S.; Xie, P.; Xu, J.; Chen, Y.; Zhang, M.; et al. 2023. Zero-Shot Information Extraction via Chatting with ChatGPT. *arXiv e-prints*, arXiv–2302.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Association for Computational Linguistics.

Yang, H.-W.; Zou, Y.; Shi, P.; Lu, W.; Lin, J.; and Sun, X. 2019. Aligning Cross-Lingual Entities with Multi-Aspect Information. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4431–4441.

Yao, L.; Mao, C.; and Luo, Y. 2019. KG-BERT: BERT for Knowledge Graph Completion. *arXiv e-prints*, arXiv–1909.

Ye, H.; Zhang, N.; Chen, H.; and Chen, H. 2022. Generative Knowledge Graph Construction: A Review. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 1–17.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, 2335–2344.

Zhang, B.; and Soh, H. 2024. Extract, Define, Canonicalize: An LLM-based Framework for Knowledge Graph Construction. *arXiv e-prints*, arXiv–2404.

Zhang, N.; Deng, S.; Bi, Z.; Yu, H.; Yang, J.; Chen, M.; Huang, F.; Zhang, W.; and Chen, H. 2020. Openue: An open toolkit of universal extraction from text. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 1–8.

Zhang, S.; Pan, L.; Zhao, J.; and Wang, W. Y. 2024. The Knowledge Alignment Problem: Bridging Human and External Knowledge for Large Language Models. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, 2025–2038. Association for Computational Linguistics.