LKD-KGC: Domain-Specific KG Construction via LLM-driven Knowledge Dependency Parsing

Jiaqi Sun[1⋆], Shiyou Qian[1] (✉), Zhangchi Han[1], Wei Li[2], Zelin Qian[2], Dingyu Yang[3], Jian Cao[1], and Guangtao Xue[1]

[1] Shanghai Jiao Tong University, Shanghai, China jotaro@sjtu.edu.cn, qshiyou@sjtu.edu.cn, hanzc_2358@sjtu.edu.cn, caojian@sjtu.edu.cn, xue-gt@cs.sjtu.edu.cn

[2] Alibaba Group, Hangzhou, Zhejiang, China jianhao@taobao.com, zelin.qzl@alibaba-inc.com

[3] Zhejiang University, Hangzhou, Zhejiang, China yangdingyu@zju.edu.cn

Abstract. Knowledge Graphs (KGs) structure real-world entities and their relationships into triples, enhancing machine reasoning for various tasks. While domain-specific KGs offer substantial benefits, their manual construction is often inefficient and requires specialized knowledge. Recent approaches for knowledge graph construction (KGC) based on large language models (LLMs), such as schema-guided KGC and reference knowledge integration, have proven efficient. However, these methods are constrained by their reliance on manually defined schema, single-document processing, and public-domain references, making them less effective for domain-specific corpora that exhibit complex knowledge dependencies and specificity, as well as limited reference knowledge. To address these challenges, we propose LKD-KGC, a novel framework for unsupervised domain-specific KG construction. LKD-KGC autonomously analyzes document repositories to infer knowledge dependencies, determines optimal processing sequences via LLM-driven prioritization, and autoregressively generates entity schema by integrating hierarchical inter-document contexts. This schema guides the unsupervised extraction of entities and relationships, eliminating reliance on predefined structures or external knowledge. Extensive experiments show that compared with state-of-the-art baselines, LKD-KGC generally achieves improvements of 10%–20% in both precision and recall rate, demonstrating its potential in constructing high-quality domain-specific KGs.

Keywords: KG Construction · Large Language Model · Domain-Specific KG · Triple Extraction

1 Introduction

Knowledge Graph (KG) refers to a semantic network architecture that organizes information through graph-structured data made up of nodes and edges, formally characterizing attributes and relationships among real-world entities [17,

[⋆] This work was completed during his internship at Alibaba Group.

18]. It employs a triple-based representation of subject-predicate-object for systematic knowledge organization, which supports downstream applications that require complex inference. For instance, domain-specific graph-based retrieval-augmented generation (Graph RAG) systems enable mapping natural language with relevant entities, providing structured knowledge support for question-answering systems [10].

Knowledge graph construction (KGC) involves extracting relational triples and their attributes from data sources, followed by organizing them into a coherent graph. However, manually constructing KGs proves to be inefficient due to the massive volume and high complexity of data [17]. Domain-specific corpora further pose three unique challenges: (1) Complex knowledge dependency: Domain knowledge progresses hierarchically from foundational to advanced concepts, requiring cross-referential analysis; (2) Domain specificity: Containing dense usage of technical terminology, domain-specific abbreviations, and intricate implementation details; (3) Limited reference knowledge: Private technical documentation and operational case studies are usually unavailable through external knowledge sources. These features place significant demands on the constructor's domain expertise [2].

Automated KGC methods are increasingly being developed to help knowledge extraction [26]. Early works based on traditional machine learning models [4, 14] and earlier deep learning models [31, 19] either suffer from complex feature engineering or limited transferability across diverse tasks [15]. Recent methodologies propose leveraging large language models (LLMs) as automated KGC agents, which demonstrate strong capability in identification and structuring of complex relationships and entities from unstructured text [18]. LLMs overcome these limitations by leveraging self-supervised pre-training on massive text corpora to autonomously learn deep semantic patterns, while enabling zero/few-shot transfer across tasks through In-Context Learning (ICL) ability [6].

Current research in LLM-based KGC primarily diverges into three directions: model fine-tuning, schema guidance and reference knowledge integration. **Model Fine-tuning:** Utilizing fine-tuned language models to generate structured triples directly from text, reducing manual efforts and improving contextual inference [11, 5]. **Schema Guidance:** Employing predefined or dynamically generated schemas (metadata frameworks defining KG elements) to guide triple extraction and alignment [32, 29, 16, 30]. **Reference Knowledge Integration:** Enhancing KGC by integrating external resources (search results, public KGs, domain documents) during extraction [34, 7, 20].

Despite advances in prior research, current LLM-based KGC methodologies still face challenges in processing domain-specific corpora, primarily in three aspects: the need for manually predefined schema, the dependence on external ground-truth knowledge, and the ignorance of knowledge dependency. For instance, the RAP framework requires a predefined schema to build schema-aware reference in prompt [29], which is usually absent in real-world texts. AutoKG's real-time web retrieval mechanism becomes ineffective when handling internal system specification unavailable from public access [34]. Finally, while the EDC framework is effective at generating schema from individual documents, it is short at identifying cross-document entity relationships [32].

To address these challenges, there is a growing need for frameworks that can handle the complexities of real-world data [30]. Inspired by the principle that knowledge advances from superficial to profound in real-world, we propose the LKD-KGC framework (**L**LM-dirven **K**nowledge **Dependency** Parsing for Domain-specific **K**nowledge **G**raph **C**onstruction), which fully utilizes hierarchical knowledge by considering knowledge dependencies. LKD-KGC basically consists of three components: Dependency Evaluation, Schema Definition, and Triple Extraction. LKD-KGC first recursively accesses all directories and documents within the input repository, automatically evaluates document comprehension priorities through LLM to determine the optimal access order for the knowledge base, and generates context-aware document abstracts by incorporating this ordered contextual knowledge. Subsequently, it extracts all entity types and definitions from both original documents and their corresponding abstracts. Finally, based on the entity schema, LKD-KGC identifies valid entities and their relationships within the original text, finally constructing a high-quality KG. The code implementation for LKD-KGC can be accessed via the following anonymous repository: *https://anonymous.4open.science/r/KAD-KGC-DA17*. Experimental results across three domain-specific datasets and two base LLMs demonstrate that compared with state-of-the-art baselines, our method generally achieves improvements of 10%–20% in both precision and recall rate.

2 Related Work

As previously established, KGs present inherent construction challenges, while LLMs exhibit strong contextual semantic reasoning capabilities due to cross-domain generalizability from massive pre-training corpora [18]. This complementary synergy drives three LLM-based KGC methodologies: model fine-tuning, schema guidance, and reference knowledge integration, which all exhibit limitations when applied to real-world domain-specific corpora.

2.1 KGC based on Fine-tuned Language Models

Previous KGC methods transition from multi-stage pipelines (entity discovery, coreference resolution, relation extraction) [18] to end-to-end generation via fine-tuned LLMs. COMET [5] predicts tail entities given head entities and relations, while REBEL [11] employs self-supervised learning for direct text-to-triple prediction, both leveraging generative language models like BART [12]. Despite outperforming fully supervised models like BiLSTM-CRF [19] in generalization, these approaches still face domain adaptation limitations. Moreover, domain-specific corpora, such as operational cases in cloud systems, undergo rapid updates and lack sufficient annotation, further constraining the applicability of fine-tuned models. Consequently, emerging approaches increasingly leverage the ICL [6] capability of LLMs by incorporating diverse schemas and reference knowledge.

2.2 KGC based on Schema Guidance

Schemas (ontologies) formally define KG entities, relations, and attributes to ensure semantic coherence and quality in the constructed KG. Current approaches utilize LLMs' in-context learning through schema-guided prompts for triple extraction and alignment.

For instance, CoT-Ontology [16] combines domain ontologies with Chain-of-Thought (CoT) prompting for stepwise triples extraction. RAP proposes a schema-aware retrieval-augmented approach that dynamically incorporates structured schema knowledge and semantically relevant instances as contextual prompts [29]. AdaKGC introduces schema-enriched prefix instructors and dynamic decoding methods to handle evolving schemas without requiring retraining model [30]. The Extract-Define-Canonicalize (EDC) framework breaks KGC into 3 stages: open information extraction, schema definition, and schema canonicalisation, which can be applied to scenarios whether a pre-defined target schema is available or not [32].

These methods have limitations when processing domain-specific corpora, primarily manifested in two aspects: dependence on manually defined schemas and neglect of knowledge dependency. Specifically, both CoT-ontology and RAP depend on predefined schemas to refine prompts. While AdaKGC enables dynamic schema adjustment during KG construction or maintenance, it cannot create schemas from scratch. Although EDC can dynamically construct schemas based on input texts, its schema definition phase depends solely on relationships identified within individual documents, thereby missing inter-document conceptual dependencies.

2.3 KGC based on Reference Knowledge Integration

This methodology enhances KGC through multi-source knowledge integration (search engines, open-source KGs, domain corpora), equipping LLMs with contextualized domain awareness to improve triple extraction accuracy and fidelity.

For example, the SAC-KG framework employs a generator that retrieves relevant context and triples for specified entities using domain corpora and DBpedia [7]. TKGCon develops an automatic framework to produce fine-grained theme KGs using theme schemas derived from Wikipedia categories [27], thus reducing human effort and keeping up with changing knowledge [9]. AutoKG employs communicative intelligent agents assigned with different roles to work together on KGC, while incorporating real-time Internet retrieval systems [34]. Papaluca et al. propose the KBTE pipeline that dynamically gathers contextual information from a Knowledge Base (KB) derived from input corpora, both in the form of context triplets and of <sentence, triplets> pairs as few-shot examples [20].

These methods face limitations because of reliance on external knowledge. SAC-KG requires seed entities and reference triples from open-source KG, TKGCon employs Wikipedia taxonomies, AutoKG relies on web retrieval, and KBTE constructs its knowledge base from human-annotated ground truth triples. However, for private corpora such as internal enterprise system specifications or operational cases in cloud systems, significant challenges arise due to domain-specific

Table 1: Comparison between LKD-KGC and recent LLM-based KGC methods

Method	Independent of Predefined Schema	Independent of Ground Truth Knowledge	Global Knowledge Context	Knowledge Dependency Awareness
EDC [32]	✔	✔	✗	✗
AutoKG [34]	✔	✗	✗	✗
KBTE [20]	✔	✗	✗	✗
SAC-KG [7]	✔	✗	✔	✗
RAP [29]	✗	✗	✔	✗
CoT-Ontology [16]	✗	✔	✔	✗
LKD-KGC (Ours)	✔	✔	✔	✔

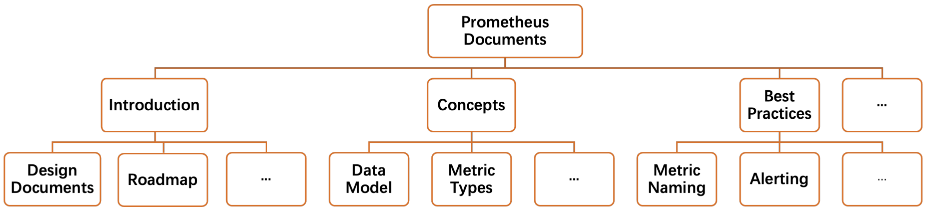

Fig. 1: The directory structure of Prometheus documents

terminologies, concepts, and abbreviations known only to internal personnel. These resources remain inaccessible through public networks and lack annotated training data with ground truth triples.

3 The LKD-KGC Framework

3.1 Overview

Design principles LKD-KGC uniquely prioritizes global knowledge dependencies through hierarchical knowledge access, inspired by the nature that human acquire knowledge from foundational to advanced concepts. This methodology processes corpora while leveraging contextual knowledge from preceding corpora segments. Table 1 demonstrates the comparison between LKD-KGC and recent LLM-based KGC methods, where only LKD-KGC does not rely on predefined schemas or external reference knowledge, while addressing Global Knowledge Context with knowledge dependency awareness.

Observation The design of LKD-KGC is derived from two critical observations pertaining to domain-specific corpora.

1. Domain corpora exhibit latent ideal accessing orders. Taking the popular open-source monitoring system Prometheus [21] as an example, as demonstrated in Figure 1, Prometheus' documentation [22], forms a directory tree with documents as leaves. When constructing knowledge graphs from this repository, initially processing the text within the *Introduction* directory helps understanding of the system's overall architecture. This should be followed by processing the content in *Concepts* to comprehend system components and terminology, before finally addressing various downstream application cases in *Best Practices*.
2. Text summarization aids in condensing entity types. By identifying and condensing key entities, it emphasizes semantically significant elements and relations while eliminating redundancy. For instance, when utilizing Llama-3.1-70B-Instruct [1] to summarize Prometheus' *Metric Types* document, it generates a summary as shown in the textbox below. By providing both the summary and the original text, LLMs can accurately summarize the four primary entity types: Counter, Gauge, Histogram, and Summary, while avoiding redundant types.

> **Example of Document Summarization for *Metric Types***
>
> Prometheus client libraries offer four core metric types: Counter, Gauge, Histogram, and Summary. These types are differentiated in client libraries and the wire protocol, but the Prometheus server currently flattens all data into untyped time series.

Workflow As depicted in Figure 2, the LKD-KGC framework comprises three modules for Dependency Evaluation, Schema Definition, and Triple Extraction. The modules sequentially implement a three-stage workflow: evaluating optimal knowledge access order → constructing entity schema → extracting triples under the guidance of established entity schema.

1. The Dependency Evaluation Module begins with a bottom-up traversal of the tree to generate summaries for documents or directories corresponding to each node. It then performs a top-down recursive ordering from the root node, assessing the optimal access order. Finally, it generates summaries for each document, incorporating the summaries of previously ordered documents as contextual knowledge.
2. For each document, the Schema Definition Module extracts entity types and their definitions from the original text and its summary. These entity types are embedded as semantic vectors and clustered. Similar entity types in each cluster are unified to retrieve relevant summaries and annotated with definitions.
3. The Triple Extraction Module extracts valid entities from the original text based on specified entity types and definitions, and subsequently identifies relationships among these valid entities to construct the final KG.

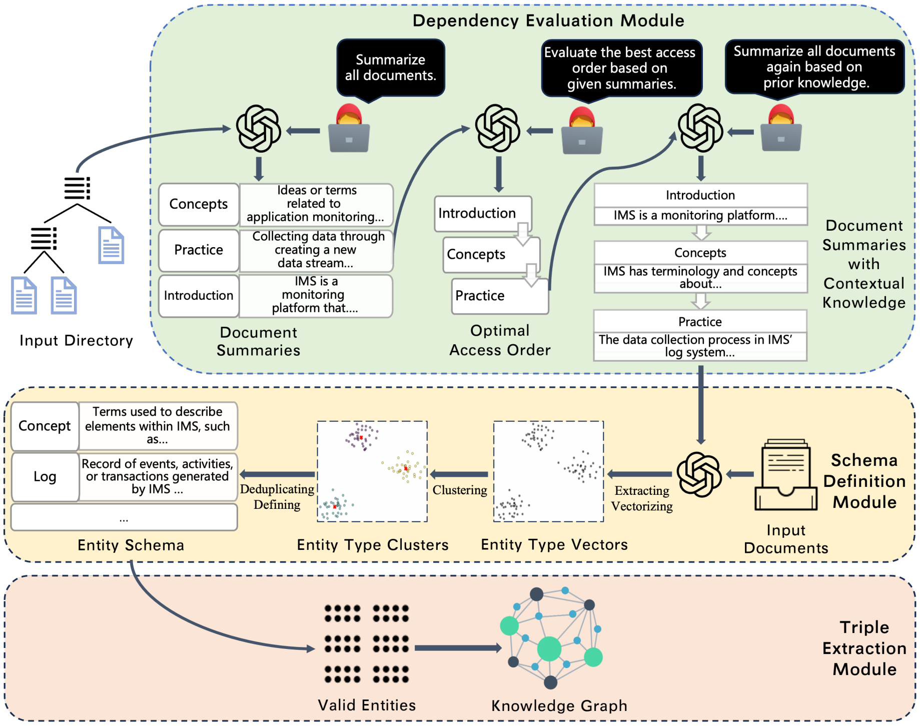
Dependency Evaluation Module
Summarize all documents.
Evaluate the best access order based on given summaries.
Summarize all documents again based on prior knowledge.
Input Directory
Concepts
Ideas or terms related to application monitoring...
Practice
Collecting data through creating a new data stream...
Introduction
IMS is a monitoring platform that....
Document Summaries
Introduction
Concepts
Practice
Optimal Access Order
Introduction
IMS is a monitoring platform....
Concepts
IMS has terminology and concepts about...
Practice
The data collection process in IMS' log system...
Document Summaries with Contextual Knowledge
Concept
Terms used to describe elements within IMS, such as...
Log
Record of events, activities, or transactions generated by IMS ...
...
Entity Schema
Deduplicating
Defining
Entity Type Clusters
Clustering
Entity Type Vectors
Extracting
Vectorizing
Input Documents
Schema Definition Module
Valid Entities
Knowledge Graph
Triple Extraction Module

Fig. 2: The framework of LKD-KGC. We employ the system specifications of an internal application monitoring system of a large Internet company as an illustrative example. To preserve author anonymity, we adopt the codename "IMS"(Internal Monitoring System) for referencing this corpus. There is almost no publicly available information regarding this system on the Internet.

3.2 System Components

Dependency Evaluation Module This module is designed to determine the optimal order for knowledge access and guide the generation of context-aware document summaries. When processing long unstructured text without a directory structure, we can split the text into chunks, treating each chunk as an individual document to establish a single-layer flattened structure. Therefore, we assume a more complex scenario where the input document repository contains multi-layered hierarchical structures.

When processing a document root directory, the Dependency Evaluation Module first traverses the entire directory tree and prompts the model to generate summaries for each document. This initial step aims to understand document content as the basis for ordering, requiring no specific access order and generating summaries without cross-document references. After obtaining summaries of each individual document, the module hierarchically generates directory summaries in a bottom-up sequence, finally establishing summaries containing key entities and relationships for each node in the directory tree.

Subsequently, the module performs a top-down traversal of the directory tree. At each hierarchical level, it constructs prompts using existing summaries to assess optimal access order. It recursively applies this evaluation process when encountering subdirectories. For leaf-node documents, it generates context-enhanced summaries by incorporating historical knowledge from previously accessed documents into prompts. This autoregressive approach ensures that each document summary is generated with reference to prior knowledge, progressively building a comprehensive knowledge repository through ordered knowledge integration.

In case of processing large volumes of data, including all prior document summaries in prompts becomes infeasible due to model context length constraints. To address this issue, we implement a vector retrieval functionality: each document summary is stored in a vector database with its embedding as the key, and the original summary text as the value. As processing subsequent documents, the content of new documents is vectorized as query vectors to retrieve the top-k related summaries ($k = 10$ by default) exhibiting the smallest cosine distance (i.e., highest semantic similarity) from the vector database. This RAG-like approach enables adaptation of referenced knowledge context length to LLM's context length.

Schema Definition Module This module extracts entity types from individual documents based on text summaries containing contextual information. It then performs entity type clustering and deduplication, ultimately generating definition for each entity type, as the final entity schema. Specifically, the module first constructs a prompt for each document (as illustrated below, taking *Prometheus Metric Types* as an example), instructing the LLM to summarize all meaningful entity types.

Example of Entity Type Extraction Prompt

Task: Some text and its summary will be provided. You need to refer to the summary and the content of the text to classify meaningful entities within the text, and return a list containing all the entity types.
Text: A counter is a cumulative metric that...
Summary: Prometheus client libraries offer four core metric types: Counter, Gauge, Histogram, and Summary...

LLM: [Counter, Gauge, Histogram, Summary]

However, the initially extracted entity types usually exhibit an excessive quantity with significant redundancy. To resolve coreference and deduplicate entity types, this module first filters out entity types that appear only once. Then, it vectorizes all entity types to embeddings and aggregates semantically similar ones through K-means clustering algorithm [13], as formalized below:

$$\arg\min_{S} \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2, \tag{1}$$

where $\mathbf{x}_i$ is the embedding of the *i-th* entity type, and $S = \{C_1, C_2, \ldots, C_k\}$ is the collection of all entity type clusters. Here cluster number k is a hyperparameter, and LKD-KGC automatically determines the optimal k by silhouette coefficient s [24], which is a quantitative measure that evaluates clustering quality. A higher s indicates better clustering quality, with stronger intra-cluster cohesion and clearer separation between clusters:

$$\arg\max_{k} \quad \frac{1}{N} \sum_{i=1}^{N} \frac{b(i) - a(i)}{\max(a(i), b(i))}, \tag{2}$$

where $a(i)$ = mean intra-cluster distance of entity type vector i (compactness); $b(i)$= mean nearest-cluster distance of entity type vector i (separation).

Only entity types remain insufficient to constitute a complete entity schema. Therefore, in the final stage, the Schema Definition Module presents each entity type cluster to LLM, directing it to deduplicate semantically equivalent types and formulate comprehensive definitions for all entity types. During this process, vector retrieval is performed with the summary knowledge base to recall summaries relevant to each cluster. These retrieved summaries serve as contextual knowledge to facilitate the model's entity definition as illustrated below. Finally, the Schema Definition Module produces a complete entity schema containing refined and meaningful entity types with their definitions.

Example of Entity Type Definition Prompt

Task: Here are some entity types. First, merge types that have the same meaning. Then, briefly define each merged type in one sentence. These entity types come from the same knowledge base; you can refer to the reference knowledge below.
Input entity types: [Configuration, Configures, ...]
Reference Knowledge: "guides/multi-target-exporter": The document provides a YAML configuration example for setting up Prometheus to scrape metrics from a Blackbox Exporter...

LLM: {"Configuration": Prometheus is configured by YAML files that define scrape targets, intervals, storage settings, and alerting rules, ...}

Triple Extraction Module This module prompts LLM to extract valid relation triples from the original text under the guidance of the entity schema. Given the large number of entity types (e.g., 66 entity types extracted from 62

Prometheus documentation articles using Llama-3.1-70B-Instruct [1]), the potential relationship types between entity types grow exponentially, making the definition of all relation types impractical. Therefore, the Triple Extraction Module does not define relation types. Instead, it first extracts valid entities and then extracts potential relations based on given entities as illustrated below, which finally form a comprehensive KG.

Example of Triple Extraction

Task: Here is a document. Please extract meaningful entities and their types which belongs to the given entity schema from the document.
Document content: This guide will introduce you to the multi-target exporter pattern...
Entity Schema: {"Configuration": Prometheus is configured by YAML files that define scrape targets, intervals, and alerting rules, ...}

LLM: {Configuration:[Blackbox.yml, Prometheus.yml], ...}

Task: Below is a document and some entities extracted from it. What entities do you believe have clear and meaningful relationships? Give all relation triples in the format of [(Subject Entity, Predicate, Object Entity)].
Document content: This guide will introduce you to the multi-target exporter pattern...
Entities {Configuration:[Blackbox.yml, Prometheus.yml], ...}
Entity Schema: {"Configuration": Prometheus is configured by YAML files that define scrape targets, intervals, and alerting rules, ...}

LLM: [(Prometheus.yml, Defines, Scrape_configs),...]

4 Experiments

In this section, we compare LKD-KGC with three LLM-based methods that unsupervisedly extract knowledge graphs from open texts. Experiments across three datasets and two base LLMs demonstrate that LKD-KGC outperforms state-of-the-art methods in both precision and recall metrics.

4.1 Experimental Setup

Datasets We select three domain-specific corpora, including two public corpus and one private corpus, as follows:

1. **Prometheus documentation [22]**: This corpus comprises 62 documents detailing the architecture, concepts, and tutorials of the Prometheus monitoring system [21], with extensive publicly available reference materials.

2. **Re-DocRED** [25]: A refined version of DocRED [28] (the most popular benchmark for document-level relation extraction), enhanced through re-annotation of omitted relation triples. We extracted a Windows-centric subset containing 15 documents, like *Windows Operating Systems* and *Windows Softwares*.
3. **IMS documentation**: Containing 46 internal technical documents from a large Internet company, covering product specifications, terminologies, and guidelines for a monitoring system. There are almost no reference knowledge about this system on the Internet.

Baselines We selected three LLM-based methods that extract knowledge graph without supervision as baselines: EDC [32], AutoKG [34], and KBTE [20]. All of these approaches do not rely on manually predefined schemas. Among them, KBTE requires constructing a knowledge base using ground truth tuples from the training set, which are unavailable in the open texts. Therefore, during reproduction, we build the knowledge base following KBTE's methodology based on tuples extracted from preceding documents.

Experiment Configurations We employ two base LLMs: Llama-3.1-70B-Instruct [1] and DeepSeek-R1-Distill-Qwen-14B [8]. Following previous work, the temperature parameter for LLM inference is set to 0.1 to control output stability [7]. The context length of vector retrieval is set to 10 summaries. Finally, the text embedding model used in all experiments is paraphrase-multilingual-MiniLM-L12-v2 [23].

Evaluation Metrics To evaluate the quality of KGs, we assess the precision and recall of extracted triples. Specifically, the Prometheus and IMS datasets lack manually annotated ground-truth triples, making direct validation of triple correctness infeasible. Given the substantial volume of triples, human evaluation also proves impractical. Following Zheng et al.'s finding that LLMs achieve human-comparable performance in fact assessment when combined with source text [33, 7], we adopt the LLM-as-judge approach from prior work, inputting extracted triples alongside their corresponding text into Qwen2-VL-72B [3] to evaluate precision. Calculating recall rate is infeasible due the absence of ground-truth tuples. Following Chen et al., we instead compute the number of true triples per document as recall metric, which aligns with real-world scenarios [7].

The Re-DocRED dataset provides ground-truth triples. However, due to the absence of training process, the outputs of LLMs fail to strictly align with the ground truth, exhibiting numerous entities and relations that share same semantics but differ in expressions (e.g., "Windows" vs. "Microsoft Windows"). We therefore employ the LLM-as-judge methodology again, utilizing LLMs to assess semantic equivalence between extracted triples and ground truth triples, subsequently computing precision, recall, and F1 score.

4.2 Evaluation Results

Public Corpora We first conducted evaluations on two public datasets. Experimental results shown in Table 2 demonstrate that LKD-KGC achieves significantly higher precision and greater numbers of recalled triples than all baselines

Table 2: Evaluation results on two public datasets: Prometheus Documentations(I) and Re-DocRED(II)

Model Dataset		Meta-Llama-3.1-70B-Instruct			DeepSeek-R1-Distill-Qwen-14B		
I	Method\Metric	Precision	Recall Number	Average Recall Number	Precision	Recall Number	Average Recall Number
	EDC	65.4%	3784	61	69.3%	3870	62.4
	AutoKG	65.7%	2678	43.2	66.3%	2138	34.5
	KBTE	75.6%	2216	35.7	63.0%	1503	24.2
	LKD-KGC	**83.4%**	**4561**	**73.6**	**84.7%**	**4047**	**65.3**
II	Method\Metric	Precision	Recall	F1 Socre	Precision	Recall	F1 Score
	EDC	17.7%	14.6%	16	10.3%	16.5%	12.6
	AutoKG	22.8%	20.9%	20.9	23.9%	16.9%	19.8
	KBTE	32.3%	18.8%	23.8	25.9%	22.0%	23.8
	LKD-KGC	**36.3%**	**25.1%**	**29.7**	**32.2%**	**27.1%**	**29.5**

on both Meta-Llama-3.1-70B-Instruct and DeepSeek-R1-Distill-Qwen-14B.

Prometheus Documentations(I) In terms of precision, the three baselines demonstrated comparable performance. However, LKD-KGC significantly reduces the false positive rate and achieves higher accuracy by first extracting valid entities according to entity schema, then discovering relationships based on these valid entities.

As for recall number, due to the highly specialized nature of Prometheus documentation that contains a vast number of technical terms, LKD-KGC extracted 6,580 valid entities, averaging 110.5 per document, thereby discovering more valid relations and achieving an advantage in recall number. Among the three baselines, EDC achieved significantly higher recall numbers. This difference primarily stems from the baselines' operational scopes: EDC performs open information extraction and relation schema definition at the sentence level, whereas AutoKG and KBTE operate at the document chunk level.

Re-DocRED (II) We conducted experiments on the "Microsoft Windows" subset within the Re-DocRED dataset. This dataset presents greater challenges by re-annotating false negative entities in the original DocRED dataset. As demonstrated by Zhu et al., LLMs fail to surpass fully supervised small models on such datasets, **usually achieving F1 scores around 20** [34]. The observed discrepancy can be attributed to the fundamental distinction that fully supervised approaches leverage human-annotated ground truth labels during training. In contrast, LLMs do not learn potential relation sets from training data in ad-

Table 3: Evaluation results on IMS Documentation dataset

Model	Meta-Llama-3.1-70B-Instruct			DeepSeek-R1-Distill-Qwen-14B		
Method\Metric	Precision	Recall Number	Average Recall Number	Precision	Recall Number	Average Recall Number
EDC	73.2%	1852	40.3	77.0%	2293	49.8
AutoKG	56.7%	2296	49.9	64.3%	1424	31.0
KBTE	80.4%	1637	35.6	74.0%	1067	23.2
LKD-KGC(Ours)	**89.6%**	**2672**	**58.1**	**88.9%**	**2764**	**60.1**

vance, thereby resulting in a broader spectrum of relation types. Despite this inherent difference, LKD-KGC demonstrates significantly superior performance in terms of precision and recall metrics compared to all baseline methods, notably achieving an F1 score approaching 30.

Private Corpora We further conducted evaluations on the IMS documentation dataset, with the results presented in Table 3. As previously established, there are little publicly available reference about this system on the Internet. Therefore, constructing a high-quality knowledge graph from this dataset necessitates full utilization of contextual knowledge within the documents.

Generally, consistent with the public dataset, LKD-KGC achieved superior precision and recall number. Compared to Prometheus documentation, the IMS documentation exhibits less extensive content and relatively simpler concepts, from which LKD-KGC identified 41 entity types.

In terms of accuracy, AutoKG exhibits significant performance degradation compared to public datasets, while the other three methods demonstrate improvements. This occurs because AutoKG retrieves numerous irrelevant concepts (e.g., hydrogen energy companies) when attempting to acquire IMS-related knowledge by online retrieval. These concepts unrelated to monitoring systems lead to model hallucinations, substantially reducing the accuracy of tuple extraction. Both EDC and KBTE incorporate mechanisms for context-aware knowledge assistance to facilitate factual comprehension, thereby achieving higher accuracy on these relatively simple corpora. LKD-KGC maintains superior accuracy by comprehensively evaluating the optimal order of document access, enabling more efficient utilization of knowledge contexts.

As for recall number, as previously noted, this dataset contains fewer knowledge concepts, resulting in generally lower average tuple recall numbers compared to the Prometheus dataset. Nevertheless, LKD-KGC extracted 4,984 valid entities, ensuring sufficient recalled triples.

5 Conclusion

This paper proposes LKD-KGC, a novel framework for unsupervised domain-specific KG construction. By leveraging LLM-driven knowledge dependency parsing and autoregressive schema generation, LKD-KGC autonomously infers hierarchical corpora processing order and integrates cross-document contexts to build entity schemas. Experimental results on public and private datasets demonstrate its superiority over state-of-the-art methods in both precision and recall, validating its capability to handle domain-specific complexities.

References

1. Aaron Grattafiori et. al.: The llama 3 herd of models (2024), https://arxiv.org/abs/2407.21783
2. Abu-Salih, B.: Domain-specific knowledge graphs: A survey. J. Netw. Comput. Appl. **185**, 103076 (2021). https://doi.org/10.1016/J.JNCA.2021.103076
3. An Yang, Baosong Yang, et. al.: Qwen2 technical report. CoRR **abs/2407.10671** (2024). https://doi.org/10.48550/ARXIV.2407.10671
4. Bikel, D.M., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: Fifth Conference on Applied Natural Language Processing. pp. 194–201. Association for Computational Linguistics, Washington, DC, USA (Mar 1997). https://doi.org/10.3115/974557.974586
5. Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y.: COMET: Commonsense transformers for automatic knowledge graph construction. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4762–4779. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1470
6. Brown, Tom B. et. al.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
7. Chen, H., Shen, X., Lv, Q., Wang, J., Ni, X., Ye, J.: SAC-KG: Exploiting large language models as skilled automatic constructors for domain knowledge graph. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 4345–4360. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024). https://doi.org/10.18653/v1/2024.acl-long.238
8. DeepSeek-AI, Daya Guo, et. al. : Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. CoRR **abs/2501.12948** (2025). https://doi.org/10.48550/ARXIV.2501.12948
9. Ding, L., Zhou, S., Xiao, J., Han, J.: Automated construction of theme-specific knowledge graphs (2024), https://arxiv.org/abs/2404.19146
10. Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R.O., Larson, J.: From local to global: A graph rag approach to query-focused summarization (2025), https://arxiv.org/abs/2404.16130
11. Huguet Cabot, P.L., Navigli, R.: REBEL: Relation extraction by end-to-end language generation. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 2370–2381. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.findings-emnlp.204

12. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.703
13. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–136 (1982). https://doi.org/10.1109/TIT.1982.1056489
14. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 188–191 (2003), https://aclanthology.org/W03-0430/
15. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Su, K.Y., Su, J., Wiebe, J., Li, H. (eds.) Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 1003–1011. Association for Computational Linguistics, Suntec, Singapore (Aug 2009), https://aclanthology.org/P09-1113/
16. Nie, J., Hou, X., Song, W., Wang, X., Jin, X., Zhang, X., Zhang, S., Shi, J.: Knowledge graph efficient construction: Embedding chain-of-thought into llms. In: Proceedings of Workshops at the 50th International Conference on Very Large Data Bases, VLDB 2024, Guangzhou, China, August 26-30, 2024. VLDB.org (2024), https://vldb.org/workshops/2024/proceedings/LLM+KG/LLM+KG-4.pdf
17. Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges. Commun. ACM **62**(8), 36–43 (Jul 2019). https://doi.org/10.1145/3331166
18. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. IEEE Trans. Knowl. Data Eng. **36**(7), 3580–3599 (2024). https://doi.org/10.1109/TKDE.2024.3352100
19. Panchendrarajan, R., Amaresan, A.: Bidirectional LSTM-CRF for named entity recognition. In: Politzer-Ahles, S., Hsu, Y., Huang, C., Yao, Y. (eds.) Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, PACLIC 2018, Hong Kong, December 1-3, 2018. Association for Computational Linguistics (2018), https://aclanthology.org/Y18-1061/
20. Papaluca, A., Krefl, D., Rodríguez Méndez, S., Lensky, A., Suominen, H.: Zero- and few-shots knowledge graph triplet extraction with large language models. In: Biswas, R., Kaffee, L.A., Agarwal, O., Minervini, P., Singh, S., de Melo, G. (eds.) Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024). pp. 12–23. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024). https://doi.org/10.18653/v1/2024.kallm-1.2
21. Prometheus Authors: prometheus (2025), https://prometheus.io/
22. Prometheus Authors: prometheus documents (2025), https://prometheus.io/docs/introduction/overview/
23. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), http://arxiv.org/abs/1908.10084
24. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987). https://doi.org/https://doi.org/10.1016/0377-0427(87)90125-7

25. Tan, Q., Xu, L., Bing, L., Ng, H.T., Aljunied, S.M.: Revisiting docred - addressing the false negative problem in relation extraction. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. pp. 8472–8487. Association for Computational Linguistics (2022). https://doi.org/10.18653/V1/2022.EMNLP-MAIN.580

26. Tang, J., Yang, Y., Wei, W., Shi, L., Su, L., Cheng, S., Yin, D., Huang, C.: Graphgpt: Graph instruction tuning for large language models (2024), https://arxiv.org/abs/2310.13023

27. Wikimedia Foundation: Wikipedia (2025), https://www.wikipedia.org/

28. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., Sun, M.: DocRED: A large-scale document-level relation extraction dataset. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 764–777. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1074

29. Yao, Y., Mao, S., Zhang, N., Chen, X., Deng, S., Chen, X., Chen, H.: Schema-aware reference as prompt improves data-efficient knowledge graph construction. In: Chen, H., Duh, W.E., Huang, H., Kato, M.P., Mothe, J., Poblete, B. (eds.) Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023. pp. 911–921. ACM (2023). https://doi.org/10.1145/3539618.3591763

30. Ye, H., Gui, H., Xu, X., Chen, X., Chen, H., Zhang, N.: Schema-adaptable knowledge graph construction (2023), https://arxiv.org/abs/2305.08703

31. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Hajic, J., Tsujii, J. (eds.) COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland. pp. 2335–2344. ACL (2014), https://aclanthology.org/C14-1220/

32. Zhang, B., Soh, H.: Extract, define, canonicalize: An llm-based framework for knowledge graph construction. In: Al-Onaizan, Y., Bansal, M., Chen, Y. (eds.) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024. pp. 9820–9836. Association for Computational Linguistics (2024), https://aclanthology.org/2024.emnlp-main.548

33. Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging llm-as-a-judge with mt-bench and chatbot arena. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html

34. Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., Zhang, N.: Llms for knowledge graph construction and reasoning: recent capabilities and future opportunities. World Wide Web (WWW) **27**(5), 58 (2024). https://doi.org/10.1007/S11280-024-01297-W

References

1. Aaron Grattafiori et. al.: The llama 3 herd of models (2024), https://arxiv.org/abs/2407.21783
2. Abu-Salih, B.: Domain-specific knowledge graphs: A survey. J. Netw. Comput. Appl. **185**, 103076 (2021). https://doi.org/10.1016/J.JNCA.2021.103076
3. An Yang, Baosong Yang, et. al.: Qwen2 technical report. CoRR **abs/2407.10671** (2024). https://doi.org/10.48550/ARXIV.2407.10671
4. Bikel, D.M., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: Fifth Conference on Applied Natural Language Processing. pp. 194–201. Association for Computational Linguistics, Washington, DC, USA (Mar 1997). https://doi.org/10.3115/974557.974586
5. Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y.: COMET: Commonsense transformers for automatic knowledge graph construction. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4762–4779. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1470
6. Brown, Tom B. et. al.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
7. Chen, H., Shen, X., Lv, Q., Wang, J., Ni, X., Ye, J.: SAC-KG: Exploiting large language models as skilled automatic constructors for domain knowledge graph. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 4345–4360. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024). https://doi.org/10.18653/v1/2024.acl-long.238
8. DeepSeek-AI, Daya Guo, et. al. : Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. CoRR **abs/2501.12948** (2025). https://doi.org/10.48550/ARXIV.2501.12948
9. Ding, L., Zhou, S., Xiao, J., Han, J.: Automated construction of theme-specific knowledge graphs (2024), https://arxiv.org/abs/2404.19146
10. Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R.O., Larson, J.: From local to global: A graph rag approach to query-focused summarization (2025), https://arxiv.org/abs/2404.16130
11. Huguet Cabot, P.L., Navigli, R.: REBEL: Relation extraction by end-to-end language generation. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 2370–2381. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.findings-emnlp.204
12. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.703
13. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–136 (1982). https://doi.org/10.1109/TIT.1982.1056489

14. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. pp. 188–191 (2003), https://aclanthology.org/W03-0430/
15. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Su, K.Y., Su, J., Wiebe, J., Li, H. (eds.) Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 1003–1011. Association for Computational Linguistics, Suntec, Singapore (Aug 2009), https://aclanthology.org/P09-1113/
16. Nie, J., Hou, X., Song, W., Wang, X., Jin, X., Zhang, X., Zhang, S., Shi, J.: Knowledge graph efficient construction: Embedding chain-of-thought into llms. In: Proceedings of Workshops at the 50th International Conference on Very Large Data Bases, VLDB 2024, Guangzhou, China, August 26-30, 2024. VLDB.org (2024), https://vldb.org/workshops/2024/proceedings/LLM+KG/LLM+KG-4.pdf
17. Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges. Commun. ACM **62**(8), 36–43 (Jul 2019). https://doi.org/10.1145/3331166
18. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. IEEE Trans. Knowl. Data Eng. **36**(7), 3580–3599 (2024). https://doi.org/10.1109/TKDE.2024.3352100
19. Panchendrarajan, R., Amaresan, A.: Bidirectional LSTM-CRF for named entity recognition. In: Politzer-Ahles, S., Hsu, Y., Huang, C., Yao, Y. (eds.) Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, PACLIC 2018, Hong Kong, December 1-3, 2018. Association for Computational Linguistics (2018), https://aclanthology.org/Y18-1061/
20. Papaluca, A., Krefl, D., Rodríguez Méndez, S., Lensky, A., Suominen, H.: Zero- and few-shots knowledge graph triplet extraction with large language models. In: Biswas, R., Kaffee, L.A., Agarwal, O., Minervini, P., Singh, S., de Melo, G. (eds.) Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024). pp. 12–23. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024). https://doi.org/10.18653/v1/2024.kallm-1.2
21. Prometheus Authors: prometheus (2025), https://prometheus.io/
22. Prometheus Authors: prometheus documents (2025), https://prometheus.io/docs/introduction/overview/
23. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), http://arxiv.org/abs/1908.10084
24. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987). https://doi.org/https://doi.org/10.1016/0377-0427(87)90125-7
25. Tan, Q., Xu, L., Bing, L., Ng, H.T., Aljunied, S.M.: Revisiting docred - addressing the false negative problem in relation extraction. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022. pp. 8472–8487. Association for Computational Linguistics (2022). https://doi.org/10.18653/V1/2022.EMNLP-MAIN.580
26. Tang, J., Yang, Y., Wei, W., Shi, L., Su, L., Cheng, S., Yin, D., Huang, C.: Graphgpt: Graph instruction tuning for large language models (2024), https://arxiv.org/abs/2310.13023

27. Wikimedia Foundation: Wikipedia (2025), https://www.wikipedia.org/
28. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., Sun, M.: DocRED: A large-scale document-level relation extraction dataset. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 764–777. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1074
29. Yao, Y., Mao, S., Zhang, N., Chen, X., Deng, S., Chen, X., Chen, H.: Schema-aware reference as prompt improves data-efficient knowledge graph construction. In: Chen, H., Duh, W.E., Huang, H., Kato, M.P., Mothe, J., Poblete, B. (eds.) Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023. pp. 911–921. ACM (2023). https://doi.org/10.1145/3539618.3591763
30. Ye, H., Gui, H., Xu, X., Chen, X., Chen, H., Zhang, N.: Schema-adaptable knowledge graph construction (2023), https://arxiv.org/abs/2305.08703
31. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Hajic, J., Tsujii, J. (eds.) COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland. pp. 2335–2344. ACL (2014), https://aclanthology.org/C14-1220/
32. Zhang, B., Soh, H.: Extract, define, canonicalize: An llm-based framework for knowledge graph construction. In: Al-Onaizan, Y., Bansal, M., Chen, Y. (eds.) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024. pp. 9820–9836. Association for Computational Linguistics (2024), https://aclanthology.org/2024.emnlp-main.548
33. Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging llm-as-a-judge with mt-bench and chatbot arena. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 (2023), http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html
34. Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., Zhang, N.: Llms for knowledge graph construction and reasoning: recent capabilities and future opportunities. World Wide Web (WWW) **27**(5), 58 (2024). https://doi.org/10.1007/S11280-024-01297-W