Domain Ontology-Driven Knowledge Graph Generation from Text

ZHIXIN MENG, SHAOXIONG ZHAN, and RUIQING XU, Huazhong Agricultural University, Wuhan, China
WOLFGANG MAYER, UniSA STEM, University of South Australia, Adelaide, Australia
YE ZHU, Centre for Cyber Resilience and Trust, Deakin University, Burwood, Australia
HONG-YU ZHANG, Huazhong Agricultural University, Wuhan, China
CHUAN HE, Huazhong University of Science and Technology, Wuhan, China
KEQING HE, Wuhan University, Wuhan, China
DEBO CHENG, STEM, University of South Australia, Adelaide, Australia
ZAIWEN FENG, College of Informatics, Huazhong Agricultural University, Wuhan, China

A knowledge graph serves as a unified and standardized representation for extracting and representing textual information. In the field of knowledge extraction and representation research, named entity recognition and relation extraction provide effective solutions for knowledge graph generation tasks. However, it is a challenge that lies in extracting domain-specific knowledge from the rich and general textual corpora and generating corresponding domain knowledge graphs to support domain-specific reasoning, question-answering, and decision-making tasks. The hierarchical domain knowledge representation model (i.e., domain ontology) provides a solution for this problem. Therefore, we propose an end-to-end approach based on domain ontology embedding and pre-trained language models for domain knowledge graph generation from text, which incorporates domain node recognition and domain relation extraction phases. We evaluated our domain ontology-driven model on the Wikidata-TekGen dataset and the DBpedia-WebNLG dataset, and the results indicate that our approach based on the pre-trained language models with fewer parameters compared with the baseline models has significantly contributed to the domain knowledge graph generation without prompts.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Information systems** → *Information retrieval*;

Zhixin Meng, Shaoxiong Zhan, and Ruiqing Xu contributed equally to this research.

This research project was supported in part by National Key Research and Development Program of China under Grant 2023YFF1000100, and the Key Research and Development Program of Hubei Province of China under Grant 2024BBB055, and in part by the National Natural Science Foundation of China under Grant 62072206, and in part by the Fundamental Research Funds for the Chinese Central Universities under Grants 2662023XXPY004 and 2662024SZ006.

Authors' Contact Information: Zhixin Meng, Huazhong Agricultural University, Wuhan, China; e-mail: 13930435862@163.com; Shaoxiong Zhan, Huazhong Agricultural University, Wuhan, China; e-mail: jasaxion@gmail.com; Ruiqing Xu, Huazhong Agricultural University, Wuhan, China; e-mail: 1009925804@qq.com; Wolfgang Mayer, UniSA STEM, University of South Australia, Adelaide, Australia; e-mail: Wolfgang.Mayer@unisa.edu.au; Ye Zhu, Centre for Cyber Resilience and Trust, Deakin University, Burwood, Australia; e-mail: ye.zhu@ieee.org; Hong-Yu Zhang, Huazhong Agricultural University, Wuhan, China; e-mail: zhy630@mail.hzau.edu.cn; Chuan He, Huazhong University of Science and Technology, Wuhan, China; e-mail: 3109631383@qq.com; Keqing He, Wuhan University, Wuhan, China; e-mail: hekeqing@whu.edu.cn; Debo Cheng (corresponding author), STEM, University of South Australia, Adelaide, Australia; e-mail: chedy055@mymail.unisa.edu.au; Zaiwen Feng (corresponding author), College of Informatics, Huazhong Agricultural University, Wuhan, China; e-mail: Zaiwen.Feng@mail.hzau.edu.cn.

ACM 2836-8924/2025/12-ART21
https://doi.org/10.1145/3708478

Additional Key Words and Phrases: Domain Knowledge Graph, Domain Ontology, Ontology Embedding, Domain Node Recognition, Domain Relation Extraction

ACM Reference format:
Zhixin Meng, Shaoxiong Zhan, Ruiqing Xu, Wolfgang Mayer, Ye Zhu, Hong-Yu Zhang, Chuan He, Keqing He, Debo Cheng, and Zaiwen Feng. 2025. Domain Ontology-Driven Knowledge Graph Generation from Text. *ACM Trans. Probab. Mach. Learn.* 1, 4, Article 21 (December 2025), 18 pages.
https://doi.org/10.1145/3708478

1 Introduction

There is a substantial corpus of textual information encompassing diverse domains, including technology, medicine, finance, and culture [12]. Embedded within these texts lies a wealth of domain-specific knowledge [1]. However, within specialized domains, general knowledge often faces issues of knowledge redundancy, absence, or unreliability. Consequently, the extraction and application of domain knowledge assume paramount significance [7, 30, 31]. Nevertheless, the sheer magnitude and dispersion of information present formidable challenges in effectively distilling useful domain knowledge. Hence, the development of a methodology capable of extracting domain knowledge from general text holds immense promise in enhancing information resource utilization and maximizing its value [19].

Knowledge graphs [7], as a unified representation of knowledge, facilitate interoperability between different data sources, thereby promoting knowledge integration and sharing. As a representative of structured knowledge, knowledge graphs have been widely applied in various fields such as finance, biology, and medicine [1]. Traditional methods of constructing knowledge graphs mainly involve named entity recognition and relation extraction, but in recent years, with the development of **Pre-Trained Language Models (PLM)**, automated knowledge graph construction [28] has garnered attention. Ontologies [18], as a form of abstract representation of knowledge, can encompass the necessary domain knowledge categories, thereby guiding the generation of knowledge graphs.

This study aims to extract the domain knowledge end-to-end from unstructured text and represent it in the form of a knowledge graph. Driven by domain ontology, the constructed domain knowledge graph can systematically organize and represent knowledge within the domain, helping to eliminate knowledge fragmentation, enhance the consistency of domain knowledge, and thus better assist users in understanding and utilizing information within the domain.

In summary, the major contributions of our work are as follows:

- —We propose a novel task scenario for domain knowledge graph generation without focusing on domain-irrelevant information.
- —The proposed end-to-end domain ontology-driven method employs ontology embedding and PLMs to effectively recognize domain nodes and extract relations between them.
- —We evaluate our model on the Wikidata-TekGen dataset and the DBpedia-WebNLG dataset, comparing the existing baselines, and the experimental results validate the effectiveness of our method.

The remainder of this article is organized as follows. We first introduce the related work in Section 2. Section 3 demonstrates the motivation and an illustrative example. Section 4 describes the details of the novel end-to-end domain ontology-driven method for knowledge graph generation from text. Section 5 presents the experimental results, and conclusions are drawn in Section 6.

2 Related Work

The construction of knowledge graphs aims to obtain entity nodes and relationships from text. Traditional construction methods are mainly divided into two parts [7]: entity and relationship extraction. CRF [27] utilizes correlations between features, effectively modeling sequence labeling tasks by learning conditional probability models. On this basis, BiLSTM-CRF [6] integrates **Bidirectional Long Short-Term Memory (BiLSTM)** Networks, enhancing the modeling ability of input sequences, better capturing contextual information within the sequences, thus improving the accuracy and generalization ability of entity recognition and relationship extraction tasks. Additionally, BERT [4] learns semantic representations of text using a pre-trained deep Bidirectional Transformer model, which is then applied to tasks such as entity linking and relationship extraction, thereby enhancing the semantic understanding and information extraction capabilities of knowledge graph construction.

The current popular methods for constructing knowledge graphs mainly fall into two categories: sequence-based methods and graph-based methods [24]. Sequence-based methods typically convert text into structured representations and then use PLMs to generate or expand knowledge graphs. These methods include models based on PLMs, such as T5 [20], BART [9], and others. MLBiNet [15] mentions using the T5 architecture to convert relation extraction into a Seq2Seq task, simplifying the process of knowledge graph construction. MaMa [22] uses PLMs to match entities and relationships in text, then maps these matching results to existing knowledge graphs to generate new triples. Grapher [16] proposes a new end-to-end automatic method for constructing knowledge graphs, addressing the issue of needing pre-defined entity and relationship patterns. On the other hand, graph-based methods utilize pre-trained graph neural network models to learn semantic relationships between entities and between entities and relationships. These models can capture complex associations and semantic information between entities through graph-structured representations, gradually expanding the entire knowledge graph. These methods include GraphRNN [25], CycleGAN [32], DualTKB [5], and others.

Domain knowledge graphs are widely utilized in both academia and industry. Clinical decision-making knowledge graphs [10] and medical recommendation graphs [14] constructed based on traditional entity and relationship extraction play crucial roles in the medical field. In the realm of the industrial **Internet of Things (IoT)**, IoT knowledge graphs proposed by Xie et al. [23] and graph of things proposed by Le-Phuoc et al. [8] also play significant roles. Meanwhile, the concept of ontology system reshaping proposed by Zhou et al. [29] aims to generate knowledge graphs that completely cover data, which has been applied to the Bosch welding case.

Regarding the automatic construction of knowledge graphs based on ontology, Text2KGBench [17] has proposed the basic process of ontology-based knowledge graph construction as well as related evaluation schemes.

Currently, traditional methods typically extract knowledge graphs from general texts, but these approaches have very limited effectiveness in extracting knowledge graphs specific to particular domains. Our proposed method introduces domain ontology, using it as a reference for filtering nodes within the domain, thereby achieving the extraction of domain knowledge graphs from general texts. This approach enhances the capability of constructing domain knowledge graphs and provides technical support for domain researchers and professionals.

3 Motivation and Illustrative Example

Knowledge graphs convert unstructured textual data into structured representations by utilizing entities and relationships, facilitating various downstream applications, including reasoning, question-answering, decision-making, and so on. Currently, most researches focus on named entity

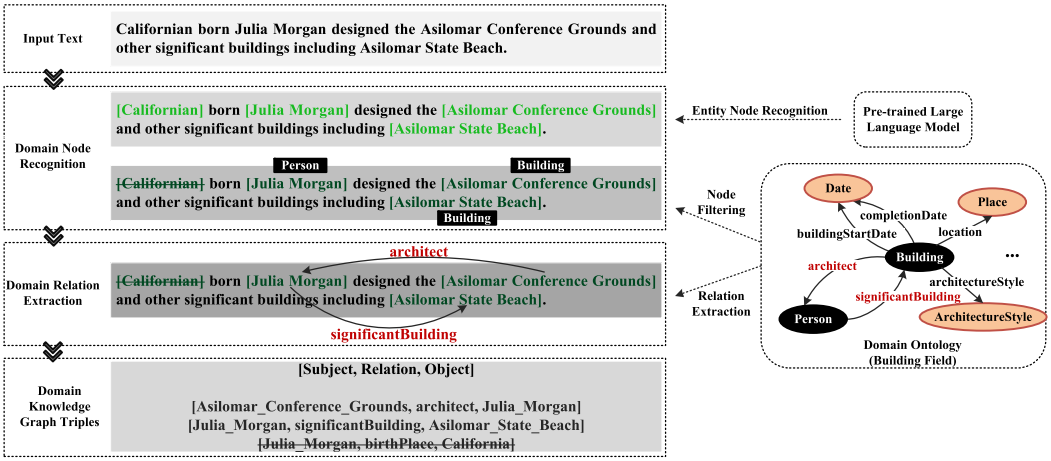

Fig. 1. The overall pipeline of domain knowledge graphs generation driven by domain ontologies.

recognition and relation extraction tasks to extract knowledge from given texts and construct knowledge graphs. However, in light of the exponential growth of textual information, the key challenge lies in effectively extracting desired domain-specific knowledge to build specialized domain knowledge graphs.

Domain ontology, as a formalized knowledge representation model, defines and describes a rich set of concepts, properties, and relationships within a specific domain. During knowledge graph generation, the domain ontology acts as a shared semantic model that enables unified understanding and representation of knowledge, ensuring the consistency and accuracy of the resulting knowledge graph. Besides, ontology-driven knowledge graph generation with good scalability can adapt to the requirements of different domains and tasks, and support incremental knowledge updates and maintenance. Therefore, we propose a novel task scenario for domain-specific knowledge graph generation driven by the domain ontology and define the problem as follows. Given the textual corpus $T = \{s_1, s_2, ..., s_m\}$, where each sentence $s_i = \{t_{i1}, t_{i2}, ..., t_{in}\}$, $t_{ij} =< sub_{ij}, rel_{ij}, obj_{ij} >$ represents the jth triple knowledge in s_i, our objective is to filter out the irrelevant knowledge using the domain ontology $DO = \{C_{DO}, R_{DO}\}$ and obtain the domain-specific knowledge graph DKG consisting of triple knowledge t_{ik} within the domain, where C_{DO} and R_{DO} represent the sets of classes and relationships in DO, respectively.

As shown in Figure 1, the overall pipeline mainly includes domain ontology embedding, domain node recognition, and domain relationship extraction. Specifically, based on the building-related domain ontology, for the input text "*Californian born Julia Morgan designed the Asilomar Conference Grounds and other significant buildings including Asilomar State Bea*," which implies knowledge of <Julia Morgan, birthPlace, California>, <Asilomar Conference Grounds, architect, Julia Morga>, and <Julia Morgan, significantBuilding, Asilomar State Beach>, we expect to extract building-related knowledge without focusing on other information (e.g., person-related knowledge <Julia Morgan, birthPlace, California>). The first step is to recognize the entity nodes such as "*Californian*," "*Julia Morgan*," "*Asilomar Conference Grounds*," and "*Asilomar State Beach*" from the input text by using the PLM. Then, we can obtain the domain nodes except the entity node "*Californian*" after node filtering based on the building-related ontology. Finally, under the guidance of the relationships from the building-domain ontology, identifying the semantic relationships for the generated domain nodes to construct the building-domain knowledge graph.

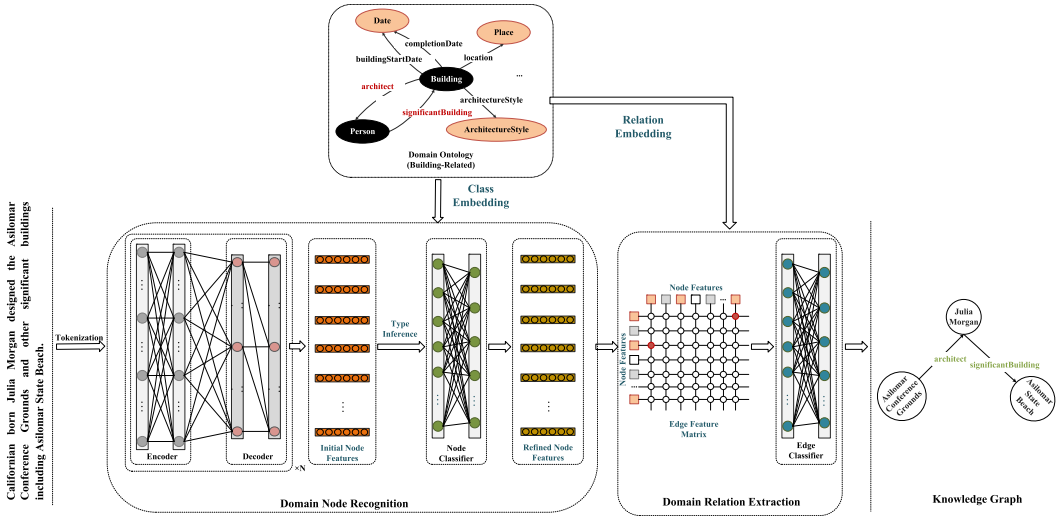

Fig. 2. Framework of the end-to-end domain knowledge graph generation based on ontology embedding and the PLM.

4 Method

The procedure for constructing domain knowledge graphs mainly involves two steps as illustrated in Figure 2: domain node recognition and domain relation extraction, where the domain ontology plays a crucial role in filtering out domain-irrelevant information. Specifically, we obtain the entity nodes by using the PLM (i.e., T5) first. Subsequently, entity nodes are classified based on the mapping of class nodes in the domain ontology to identify the nodes belonging to the specific domain. Furthermore, we introduce the corresponding class information and their contextual semantic relationships of the generated domain nodes to refine the scope of relation extraction, thereby enhancing the accuracy of domain relation extraction among domain nodes.

We describe the process of domain ontology embedding in Section 4.1, and then introduce the details on recognizing the domain nodes and extracting relations between them driven by class nodes and relations of the domain ontology in Sections 4.2 and 4.3, respectively.

4.1 Domain Ontology Embedding

Ontology is a high-level abstract knowledge representation model that describes knowledge information in a structured manner. And domain ontologies formalize and systematize domain knowledge representation by defining concepts, properties, and their relationships within a specific domain. Given a domain ontology $DO = \{C_{DO}, R_{DO}\}$, where $C_{DO} = \{c_1, c_2, ..., c_p\}$ represents the set of class nodes corresponding to the concepts c_i in DO, and $R_{DO} = \{rel_1, rel_2, ..., rel_q\}$ represents the set of relations rel_i in DO. The mappings $\phi(e_i) \longrightarrow c_i$ of entity nodes e_i and class nodes c_i are used to infer and classify the domain of entity nodes. For example, entity nodes "*Julia Morgan*" and "*Asilomar Conference Grounds*" are associated with class nodes *Person* and *Building* in the given building-related ontology, respectively, whereas domain-irrelevant entity node "*California*" fails in establishing a mapping between each other. Additionally, the contextual relationship information of class nodes in DO assists in refining domain relation extraction between entity nodes. In summary, the introduction of a domain ontology enables to filter out the domain-irrelevant information in knowledge extraction from text, ensuring the accuracy and domain consistency of extracted knowledge, thereby generating a high-quality domain knowledge graph.

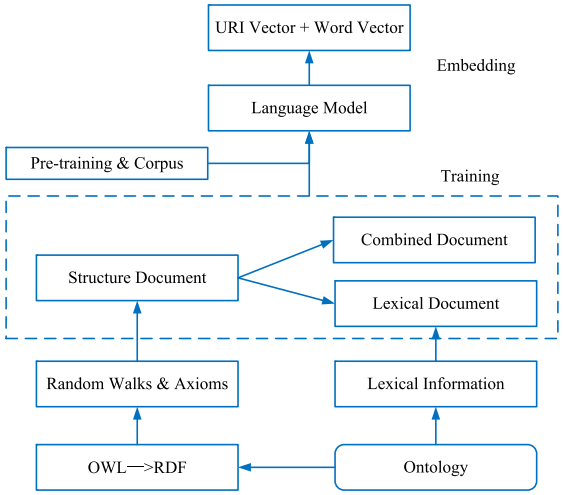

Fig. 3. OWL2Vec* algorithm pipeline [3].

Ontology embedding is an effective method for representing the rich semantics of class nodes and relations in ontologies as vectors. OWL2Vec* model [3], a technique based on random walks and word embeddings, integrates the graph structure, lexical information, and logical constructs of ontologies to encode implicit semantics, as shown in Figure 3. In this article, we fine-tuned the pre-trained OWL2Vec* model to learn feature vector representations of class nodes and relations, providing additional rich semantic guidance for subsequent domain node recognition and domain relation extraction, respectively.

$$\begin{aligned} E_c &= \texttt{OWL2Vec*}(DO, C_{DO}), \\ E_{rel} &= \texttt{OWL2Vec*}(DO, R_{DO}), \end{aligned} \tag{1}$$

where $E_c = \{e_c^i | i = 1, ..., p, e_c^i \in \mathbb{R}^d\}$ is a set of d-dimensional feature vectors e_c^i for class nodes c_i. Similarly, $E_{rel} = \{e_{rel}^j | j = 1, ..., q, e_{rel}^j \in \mathbb{R}^d\}$ is the set of feature vectors e_{rel}^j for relations rel_j.

4.2 Domain Node Recognition

There are two steps in the domain node recognition module, involving entity node generation and node type inference. In the first step, we utilize the PLM (i.e., T5 [20]) to identify and extract entity nodes from the input text. In the next step, based on the learned class node embeddings, a classifier is employed to filter out domain-irrelevant nodes from the set of entity nodes.

Specifically, we model the entity node recognition task as a sequence-to-sequence problem, where for the given input text T after tokenization, a language model L is utilized to obtain the corresponding entity node sequences ENS: $L(T) \longrightarrow ENS$, where $ENS = \{en_1, en_2, ..., en_i | 0 \leq i \leq m\}$. Moreover, in a generated node sequence $en_i = \{e_{i1}, e_{i2}, ..., e_{ij} | 1 \leq j \leq u\}$, the <$NODE\_SEP$> label and <$NO\_NODE$> label are, respectively, viewed as the separator and padding token in the generated node sequence. Consequently, the ultimate node sequence takes the form of <PAD> $NODE_1$ <$NODE\_SEP$> $NODE_2$ ... <$NODE\_SEP$> <NO_NODE> </S> as in Melnyk et al. [16], where <PAD> and </S> represent the start and end tokens of the sequence, respectively. Besides the node sequences, we derive the d-dimensional initial feature vectors $E_{node}^i \in \mathbb{R}^{N \times d}$ of entity nodes for each sentence s_i after the decoder, where N is the number of the generated

nodes in s_i:

$$E^i_{node} = \texttt{T5}(tokenizer(s_i), s_i \in T). \tag{2}$$

Finally, the cross-entropy loss function is utilized to calculate the loss between the generated node sequence ENS_h and the real node sequence ENS_r, denoted as $loss_1$:

$$\begin{aligned} loss_1 &= CELoss(ENS_h, ENS_r) = \{l_1, \ldots, l_{BS}\}^T \\ l_i &= -log\frac{exp(ENS_{h_i}, ENS_{r_i})}{\sum_{c=1}^{C} exp(ENS_{h_i}, c)}, \end{aligned} \tag{3}$$

where C represents the total number of dictionaries, and BS stands for the batch size.

To filter out irrelevant entity nodes, we introduce the class nodes in domain ontology and their corresponding feature vectors learned in Section 4.1 for node classification, which is also named type inference. The mapping class node denoted as $\phi(e_{ij}) \longrightarrow \{c_{ij} | c_{ij} \in C_{DO} \cup c_{no\_class}\}$ for each entity node e_{ij} is obtained via a linear projection and normalization:

$$\phi(e_{ij}) = \texttt{norm}(W_e e_{ij} + b_e), \tag{4}$$

where W_e and b_e are adaptive parameters, c_{no_class} is an additional special class node and we consider the entity nodes mapped with c_{no_class} as domain-irrelevant nodes. In the end, the loss between the predicted category C_h and the real category C_r of the entity nodes is calculated using the cross-entropy loss function denoted as $loss_2$:

$$\begin{aligned} loss_2 &= CELoss(C_h, C_r) = \{l_1, \ldots, l_{BS}\}^T \\ l_i &= -log\frac{exp(C_{h_i}, C_{r_i})}{\sum_{c=1}^{N_c} exp(C_{h_i}, c)}, \end{aligned} \tag{5}$$

where N_c represents the length of the set $\{C_{DO} \cup c_{no\_class}\}$.

In addition, to avoid the impact of domain-irrelevant nodes on subsequent domain relation extraction tasks, based on the classification results, the initial node features E^i_{node} will be refined to some extent. Specifically, zero-padding is applied to refine the features of domain-irrelevant nodes for obtaining the valid refined node features $E^i_{node*} \in \mathbb{R}^{N \times d}$.

4.3 Domain Relation Extraction

We aim at extracting relations between the generated domain nodes using the refined node feature vectors E^i_{node*} to construct the domain knowledge graph in this module.

The knowledge graph is a directed graph, where <Sub_1,Rel_1,Obj_1> and <Obj_1,Rel_1,Sub_1> represent distinct semantics. Therefore, it is important to consider the directionality between node pairs in the representation of edge relationship vectors to extract accurate relations. Inspired by Melnyk et al. [16], we calculate the difference between node feature vectors to effectively represent the edge relationship feature between node pairs and obtain the global edge feature matrix E_{edge} as shown in Figure 2. Specifically, $E_{edge}(i \longrightarrow j) = E_{node*}(:i) - E_{node*}(:j)$ represents the semantic relation between node en_i as the subject and node en_j as the object.

Moreover, we introduce the relation features E_{rel} learned in Section 4.1 to refine the initial edge features by using the mean operation as aggregating function $f_{agg}(E_{edge}, E_{rel})$ and locate the scope of domain relation extraction:

$$E_{edge*} = \texttt{norm}(f_{agg}(E_{edge}, E_{rel})). \tag{6}$$

Subsequently, we extract domain relationships between domain node pairs using a linear edge classifier and generate the desired domain knowledge graph, which consists of triples <Asilomar

Conference Grounds, architect, Julia Morga> and <Julia Morgan, significantBuilding, Asilomar State Beach>.

Additionally, we observe that there is a class imbalance issue in the relationship categories. To prevent the model biasing towards selecting the more numerous relationship categories during the training process, the Focal loss [13] function is employed in this stage instead of the traditional cross-entropy loss function, and weights are assigned to each relationship category, as illustrated in Equation (7), to enhance the generalization performance of the model on the imbalanced dataset.

$$\begin{aligned} &CELoss(x, y) = \{l_1, \ldots, l_{BS}\}^T, \\ &l_n = -\omega_{y_n} log p_n, \\ &\omega_{y_n} = \frac{\max(num_{y_1}, \ldots, num_{y_R})}{num_{y_n}}, \\ &p_n = \frac{exp(x_n, y_n)}{\sum_{c=1}^{C} exp(x_n, c)}, \\ &Focal = (1 - p_n)^{\gamma} \times CELoss, \end{aligned} \tag{7}$$

where $\gamma \geq 0$ is an adjustable hyper-parameter. Finally, we utilize the Focal loss function to calculate the loss between the predicted relation E_h and the true relation E_r, denoted as $loss_3$.

$$loss_3 = Focal(E_h, E_r). \tag{8}$$

5 Experiments

5.1 Datasets

In the experimental section, we utilized two benchmark datasets proposed by Text2KGBench [17]: the Wikidata-TekGen dataset and the DBpedia-WebNLG dataset for relevant evaluation tasks. The two datasets are described as follows:

(i) *Wikidata-TekGen Dataset.* The Wikidata-TekGen dataset was constructed based on the TekGen dataset [2]. The TekGen dataset is a synthesized corpus generated by converting triples from the Wikidata knowledge graph into natural language text, which aims to integrate the structured knowledge graph and the natural language corpus to enhance the generalization capabilities of PLMs. Based on the TekGen dataset, Text2KGBench constructed 10 domain ontologies based on concept and relation pairs in Wikidata. Given that the TekGen dataset is generated based on distant supervision, it may contain noise, thus, Text2KGBench further refined the dataset by manually analyzing it to accurately align the text on the dataset, and in this way constructed the Wikidata-TekGen dataset.

(ii) *DBpedia-WebNLG Dataset.* The DBpedia-WebNLG dataset was constructed on top of the WebNLG dataset [21]. Initially, the WebNLG dataset was used for the WebNLG Natural Language Generation Challenge, which involved mapping the triple set to text and mapping text to the set triple. The WebNLG dataset consists of triples describing facts and their corresponding natural language texts, with a total of 19 categories. Text2KGBench constructed a domain ontology for each category separately and constructed the DBpedia-WebNLG dataset.

However, we find certain quality issues in the domain ontologies of both datasets, such as ontology concept redundancy, object attribute redundancy, and the existence of concepts and relationships that are irrelevant to the specific domain. Responding to these issues, for each domain ontology, we made some modifications, such as pruning certain categories, modifying the domain and range of relationships, and so on, thus creating differences between domain-specific and non-domain-specific knowledge at the ontology level. For example, concerning the Building domain ontology, Figure 4(a) shows the partial view of the original Building domain ontology, which

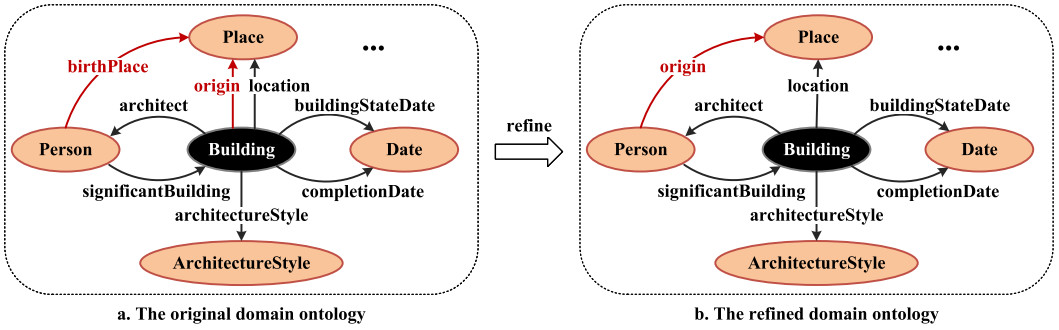

Fig. 4. Building-related domain ontology modification.

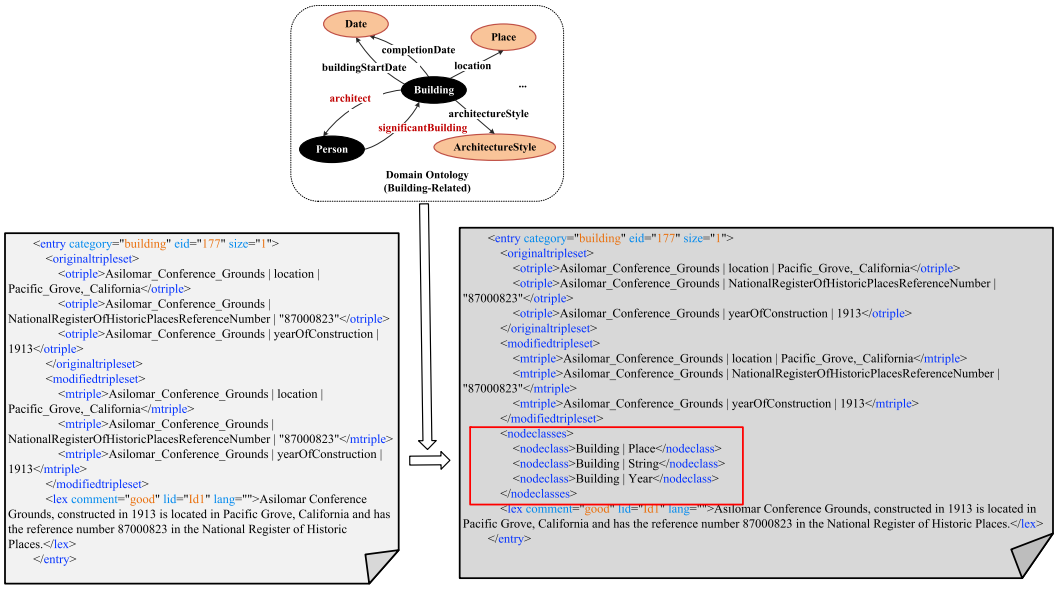

Fig. 5. The annotation of nodes.

includes the relation triple (Building, origin, Place) and the relation triple (Person, birthPlace, Place). Nevertheless, we found that the relationship "birthPlace" is not highly relevant to the Building domain ontology, and in the relation triple (Building, origin, Place), the subject should be "Person" instead of "Building." Therefore, we made corrections by removing the relation triple (Person, birthPlace, Place) and modifying the triple (Building, origin, Place) to (Person, origin, Place). The refined Building domain ontology is shown in Figure 4(b).

Subsequently, we annotated the head and tail nodes of the triples on the Wikidata-TekGen dataset and DBpedia-WebNLG dataset based on the modified domain ontology, as shown in Figure 5. The pseudocode for data annotation is presented as Algorithm 1. Then, the Wikidata-TekGen dataset is divided into train set, valid set, and test set, with the statistical results shown in Table 2. The DBpedia-WebNLG dataset is divided into a train set and a test set, with the statistical results shown in Table 1.

Algorithm 1: Data Annotation Algorithm

```
function DataAnnotation(dataset_directory, ontology_directory)
    data_files ← Get all files in the dataset directory (dataset_directory)
    ontology_files ← Get all files in the ontology directory (ontology_directory)
    for all datafile, ontofile in data_files , ontology_directory do
        data_path ← Get the full path of the file (datafile)
        ontology_path ← Match the ontology file (ontofile)
        ontology_info ← Parse the ontology file (ontology_path)
        data_content ← Parse the data file (data_path)
        xml ← new XML
        for all triple in data_content.triples do
            domain_relation ← Find the relation in the ontology (triple.relation, ontology_info)
            if domain_relation is undefined then
                Mark the relation as invalid (triple.relation)
            else
                domain ← Get the domain of the relation (domain_relation)
                range ← Get the range of the relation (domain_relation)
            end if
            Append triple information to xml (triple, domain, range)
        end for
        Write the xml to a new file (write_file)
    end for
end function
```

Table 1. DBpedia-WebNLG Dataset with Intra and Inter-Domain Knowledge Differences

Category	Domain Ontology		Train Set		Test Set	
	Concepts	Relations	Domain Texts	Domain-Irrelevant Texts	Domain Texts	Domain-Irrelevant Texts
University	12	35	103	6	46	1
Musicalwork	15	22	173	30	78	9
Airport	15	30	153	61	54	38
Building	11	27	130	62	58	25
Athlete	12	28	135	70	56	32
Politician	17	33	178	45	77	19
Company	12	20	94	13	39	7
Celestialbody	8	20	107	29	48	10
Astronaut	11	21	76	32	40	6
Comicscharacter	8	16	57	14	23	8
Meanoftransportation	18	47	154	66	59	35
Monument	12	13	20	44	9	19
Food	11	17	188	91	77	42
Writtenwork	13	27	146	79	61	36
Sportsteam	9	14	142	22	68	3
City	10	20	235	9	103	1
Artist	15	19	180	90	78	38
Scientist	10	19	117	64	53	25
Film	12	25	159	26	74	5

Table 2. Wikidata-TekGen Datasets with Knowledge Differences within and between Domains

Category	Domain Ontology		Train Set		Valid Set		Test Set	
	Concepts	Relations	Domain Texts	Domain-Irrelevant Texts	Domain Texts	Domain-Irrelevant Texts	Domain Texts	Domain-Irrelevant Texts
Movie	9	12	1,437	243	477	83	479	81
Music	8	11	1,293	53	432	16	434	15
Sport	8	7	826	138	275	47	278	43
Book	8	10	938	148	310	52	307	55
Military	6	4	408	14	134	7	134	7
Computer	5	8	428	18	141	7	146	3
Space	9	5	390	10	127	6	132	1
Politics	6	5	379	6	128	0	128	0
Nature	8	8	690	221	234	70	231	72
Culture	5	3	307	1	103	0	102	1

Table 3. Parameter Scales for the T5 Model

Model	Parameter Scale	Number of Layers	Embedding Dimension	Hardware Equipment
T5-small	61 M	12	512	RTX A4000 16 G
T5-base	223 M	24	768	A100-PCIE-40 GB
T5-large	738 M	48	1,024	A100-PCIE-40 GB

5.2 Experimental Setup

In the selection of PLMs, we utilized three different scales of language model from T5: T5-small, T5-base, and T5-large. The model parameters, number of layers, embedding dimension d, and hardware devices used for training are shown in Table 3.

In the domain ontology embedding stage, we set the random walk step to 4, the window size to 5, and the epoch to 100 in the OWL2Vec* model, and introduce negative sampling techniques to enhance the generalization of the domain ontology embedding representations. In the domain node recognition stage, we set the maximum number of entity nodes **N** to 6, meaning that when constructing knowledge graphs, each graph can contain at most six entity nodes. In the node type inference stage and domain relation extraction stage, we adopt the ReLU activation function and introduce the dropout technique, with a rate set to 0.5, to enhance the generalization ability of the model. Furthermore, for the hyper-parameter γ in the Focal loss function, we set it to 3, aiming to solve the category imbalance problem. Additionally, we set the learning rate to 10^{-4}, the batch size to 11, and the number of training rounds epoch to 100. To improve the efficiency of model training, the Adaw optimizer is used for optimization.

5.3 Baseline Model

To verify the validity of the models proposed in this article, we selected two baseline models: Vicuna and Alpaca-LoRA proposed by Text2KGBench. A brief introduction to these two baseline models is provided below, and the model parameter scales are shown in Table 4:

(i) *Vicuna Model.* Vicuna [11] is an open source **Large Language Model (LLM)** obtained by fine-tuning the pre-training model LLaMA using conversation transcripts containing 70K user shares. Vicuna-13B claims to have achieved 90% of the performance level of OpenAI ChatGPT and Google Bard [26]. In this evaluation, the authors utilized a metric termed "Relative Response Quality," employing robust LLMs, such as GPT4, as judges to assess the model's responses to open-ended questions.

(ii) *Alpaca-LoRA Model.* Alpaca-LoRA is obtained by fine-tuning the pre-trained LLaMA model by using a self-instruct approach and 52K instructions generated from Open AI's text-davinci-003

Table 4. Parameter Scales of the Baseline Models

Benchmark Model	Parameter Scale
Vicuna	13 B
Alpaca-LoRA	13 B

model. Alpaca-LoRA further employs low-rank adaptation by freezing the weights of the pre-trained model to introduce a trainable rank decomposition matrix into each transformer layer to significantly reduce the number of trainable parameters.

5.4 Evaluation Metrics

In this section, we referred to the evaluation metrics proposed by WebNLG 2020 Challenge [21] to assess the performance of generating knowledge graphs from text. The WebNLG 2020 Challenge proposed four kinds of matching on the element level. For each element in the triple, *Strict*: The candidate string and the reference must be exactly matched, and the element type (subject, relation, object) must also match the reference; *Exact*: The candidate string and the reference is required to be exactly matched, but the element type (subject, relation, object) is not required; *Partial*: The candidate string should match at least partially with the reference string, yet the element type (subject, relation, object) is irrelevant; *Type*: The candidate string should partially match the reference string, and the element type (subject, relation, object) should also match the reference.

In this article, based on the *Partial* matching level, we use *precision (P)*, *recall (R)*, and *F1 score* evaluation metrics to measure the knowledge extraction accuracy by the model-driven the domain ontology. We compare the triples generated by the model with the target triplets. Precision (P) is calculated by dividing the number of correct triples generated by the model by the number of triplets. The calculation method for recall (R) is dividing the number of correct triples generated by the model by the number of in-domain triplets. The F1 score is calculated as the harmonic mean of precision (P) and recall (R).

Additionally, we also adopt the evaluation metric **Ontology Conformance (OC)** [17] to measure whether the knowledge triples extracted by the model are affiliated to that domain. In this article, OC refers to the percentage of domain triples among all triples extracted by the model. The domain triple is defined as if the relationship in this triple is within a specific domain ontology, then this triple is an in-domain triple, otherwise, it is a non-domain triple.

In summary, higher values of the four evaluation metrics indicate better performance of the model. Since the order of the triples generated by the model and the true triples is not significant, we search for the optimal alignment between each generated triple and the true triples by considering all possible permutations of the true triples, to ensure the accuracy of the evaluation.

5.5 Experimental Results

To validate the effectiveness of the proposed model, we trained and evaluated the model against the two baseline models on the modified Wikidata-TekGen dataset and DBpedia-WebNLG dataset, and the experimental results on the Wikidata-TekGen dataset are shown in Tables 5, 6 and 9, and the experimental results on DBpedia-WebNLG dataset are shown in Tables 7, 8, and 10.

Combining the evaluation results of the knowledge extraction accuracy of the proposed model on the datasets as shown in Tables 5 and 7, it can be observed that the performance of the proposed model varies on different domain ontologies. However, as the complexity of the model increases, the knowledge extraction accuracy of the proposed model is enhanced. Overall, the average performance

Table 5. The Compared Experimental Results of T5 Models with Different Scales on the Wikidata-TekGen Dataset

Category	Our Model (T5-Small)			Our Model (T5-Base)			Our Model (T5-Large)		
	Pre	Re	F1	Pre	Re	F1	Pre	Re	F1
Movie	0.21	0.23	0.22	0.24	0.26	0.25	0.27	0.29	0.28
Music	0.23	0.24	0.24	0.25	0.26	0.26	0.29	0.30	0.29
Sport	0.48	0.54	0.51	0.48	0.53	0.50	0.50	0.56	0.53
Book	0.27	0.30	0.28	0.29	0.32	0.30	0.35	0.29	0.37
Military	0.37	0.39	0.38	0.39	0.41	0.40	0.39	0.41	0.40
Computer	0.26	0.27	0.26	0.30	0.31	0.30	0.33	0.34	0.33
Space	0.49	0.51	0.50	0.49	0.51	0.50	0.52	0.54	0.53
Politics	0.34	0.34	0.34	0.36	0.36	0.36	0.42	0.42	0.42
Nature	0.35	0.44	0.39	0.37	0.48	0.42	0.42	0.53	0.47
Culture	0.42	0.42	0.42	0.44	0.44	0.44	0.45	0.45	0.45
Average	0.30	0.33	0.31	0.32	0.35	0.34	0.36	0.39	0.37

Table 6. The Compared Experimental Results between the Proposed Model and the Baseline Models on the Wikidata-TekGen Dataset

Category	Vicuna			Alpaca-LoRA			Our Model (T5-Large)		
	Pre	Re	F1	Pre	Re	F1	Pre	Re	F1
Movie	0.33	0.23	0.25	0.28	0.14	0.17	0.27	0.29	0.28
Music	0.42	0.28	0.32	0.33	0.19	0.22	0.29	0.30	0.29
Sport	0.57	0.52	0.52	0.51	0.44	0.45	0.50	0.56	0.53
Book	0.31	0.25	0.26	0.25	0.17	0.19	0.35	0.29	0.37
Military	0.24	0.25	0.24	0.21	0.21	0.21	0.39	0.41	0.40
Computer	0.38	0.35	0.35	0.39	0.27	0.29	0.33	0.34	0.33
Space	0.68	0.67	0.66	0.58	0.55	0.55	0.52	0.54	0.53
Politics	0.34	0.32	0.33	0.22	0.22	0.21	0.42	0.42	0.42
Nature	0.25	0.27	0.25	0.28	0.26	0.27	0.42	0.53	0.47
Culture	0.31	0.32	0.31	0.15	0.16	0.15	0.45	0.45	0.45
Average	0.38	0.35	0.35	0.32	0.26	0.29	0.36	0.39	0.37

of the model in terms of knowledge extraction accuracy on both the Wikidata-TekGen dataset and DBpedia-WebNLG dataset does not exceed 40%, indicating that there is significant room for improvement in the model proposed in this article.

By comparing the results shown in Tables 6 and 8, we can evaluate the performance of the proposed model with the benchmark models in terms of knowledge extraction accuracy. On the Wikidata-TekGen dataset, the proposed model is not the best in terms of accuracy, but it has improved recall and F1 score by 4% and 2%, respectively, compared to the benchmark models. More notably, on the DBpedia-WebNLG dataset, the proposed model outperforms the benchmark models in all three aspects: accuracy, recall, and F1 score, with improvements of 5%, 13%, and 8%, respectively. These results indicate that the model proposed in this article can extract more accurate domain knowledge.

Table 7. The Compared Experimental Results of T5 Models with Different Scales on the DBpedia-WebNLG Dataset

Category	Our Model (T5-Small)			Our Model (T5-Base)			Our Model (T5-Large)		
	Pre	Re	F1	Pre	Re	F1	Pre	Re	F1
University	0.363	0.363	0.363	0.370	0.370	0.370	0.378	0.378	0.378
Musicalwork	0.339	0.348	0.343	0.378	0.387	0.382	0.391	0.401	0.396
Airport	0.398	0.398	0.398	0.397	0.397	0.397	0.400	0.400	0.400
Building	0.409	0.409	0.409	0.398	0.398	0.398	0.398	0.398	0.398
Athlete	0.378	0.380	0.379	0.384	0.386	0.385	0.384	0.385	0.385
Politician	0.373	0.373	0.373	0.391	0.391	0.391	0.388	0.388	0.388
Company	0.321	0.327	0.324	0.345	0.351	0.348	0.352	0.358	0.355
Celestialbody	0.371	0.371	0.371	0.379	0.379	0.379	0.383	0.383	0.383
Astronaut	0.361	0.364	0.362	0.381	0.383	0.382	0.391	0.394	0.393
Comicscharacter	0.337	0.337	0.337	0.333	0.333	0.333	0.323	0.323	0.323
Meanoftransportation	0.371	0.371	0.371	0.387	0.387	0.387	0.402	0.402	0.402
Monument	0.289	0.289	0.289	0.314	0.314	0.314	0.370	0.370	0.370
Food	0.323	0.323	0.323	0.336	0.336	0.336	0.337	0.337	0.337
Writtenwork	0.360	0.360	0.360	0.365	0.365	0.365	0.382	0.382	0.382
Sportsteam	0.319	0.319	0.319	0.327	0.327	0.327	0.336	0.336	0.336
City	0.460	0.460	0.460	0.478	0.478	0.478	0.496	0.496	0.496
Artist	0.373	0.376	0.375	0.387	0.391	0.389	0.392	0.396	0.394
Scientist	0.367	0.383	0.375	0.374	0.391	0.382	0.385	0.402	0.393
Film	0.376	0.381	0.379	0.386	0.391	0.389	0.386	0.391	0.389
Average	0.369	0.371	0.370	0.381	0.383	0.382	0.389	0.390	0.389

By comparing the results shown in Tables 9 and 10, we can evaluate the performance of the proposed model in terms of OC with the benchmark models. Compared with the benchmark models, although the proposed model does not perform best in certain specific domains, such as Music and Space, and so on, it demonstrates the best overall performance, with an average performance of around 90%. Specifically, on the Wikidata-TekGen dataset and DBpedia-WebNLG dataset, compared with the benchmark models, the proposed model improves the OC by 2% and 2%, respectively. Driven by domain ontology, these results indicate that the proposed model can accurately extract the in-domain knowledge and refrain from extracting knowledge out-of-domain.

In addition, the benchmark models Vicuna and Alpaca-LoRA are LLMs with GPT architecture, which not only have larger parameter scales but also need to be trained under the guidance of Prompts. However, the model proposed in this article not only significantly reduces the scale of parameters but also eliminates the need for Prompt guidance during the training process, and can obtain the same or better results than the benchmark models.

We also conduct an error analysis of the experimental results to investigate the reasons for the model's mistakes. The possible reasons include the following: (i) *Ontology quality issue*. Although we have made some corrections to the domain ontologies on the original Wikidata-TekGen dataset and DBpedia-WebNLG dataset, inevitably, there are still some other problems. For example, the ontology having a limited domain coverage and poor quality, which affects the type annotation of the head and tail entity nodes in the triples, thereby affecting the model's performance; (ii) *Data quality issue*. Quality issues in the original dataset, such as pronoun ambiguity, where only pronouns exist in the text without the entities they refer to, may cause the model to make errors in the

Table 8. The Compared Experimental Results between the Proposed Model and the Baseline Models on the DBpedia-WebNLG Dataset

Category	Vicuna			Alpaca-LoRA			Our Model (T5-Large)		
	Pre	Re	F1	Pre	Re	F1	Pre	Re	F1
University	0.31	0.19	0.23	0.29	0.16	0.20	0.378	0.378	0.378
Musicalwork	0.20	0.18	0.18	0.18	0.15	0.15	0.391	0.401	0.396
Airport	0.33	0.24	0.27	0.28	0.18	0.20	0.400	0.400	0.400
Building	0.48	0.33	0.38	0.35	0.27	0.29	0.398	0.398	0.398
Athlete	0.33	0.26	0.29	0.38	0.30	0.32	0.384	0.385	0.385
Politician	0.39	0.28	0.32	0.39	0.27	0.30	0.388	0.388	0.388
Company	0.49	0.37	0.41	0.35	0.21	0.25	0.352	0.358	0.355
Celestialbody	0.48	0.37	0.41	0.52	0.44	0.47	0.383	0.383	0.383
Astronaut	0.40	0.28	0.32	0.34	0.21	0.25	0.391	0.394	0.393
Comicscharacter	0.41	0.41	0.40	0.52	0.41	0.45	0.323	0.323	0.323
Meanoftransportation	0.22	0.17	0.18	0.13	0.09	0.10	0.402	0.402	0.402
Monument	0.04	0.05	0.05	0.11	0.07	0.08	0.370	0.370	0.370
Food	0.43	0.39	0.39	0.38	0.30	0.31	0.337	0.337	0.337
Writtenwork	0.40	0.34	0.36	0.39	0.35	0.36	0.382	0.382	0.382
Sportsteam	0.52	0.38	0.42	0.41	0.28	0.31	0.336	0.336	0.336
City	0.12	0.12	0.12	0.10	0.09	0.09	0.496	0.496	0.496
Artist	0.30	0.21	0.23	0.35	0.22	0.26	0.392	0.396	0.394
Scientist	0.52	0.43	0.46	0.42	0.33	0.34	0.385	0.402	0.393
Film	0.23	0.19	0.20	0.18	0.14	0.15	0.386	0.391	0.389
Average	0.34	0.27	0.30	0.32	0.23	0.25	0.389	0.390	0.389

Table 9. The Compared Experimental Results between the Proposed Model and the Baseline Models on the Wikidata-TekGen Dataset for OC

Category	Vicuna	Alpaca-LoRA	Our Model (T5-Small)	Our Model (T5-Base)	Our Model (T5-Large)
Movie	0.89	0.92	0.73	0.74	0.75
Music	0.94	0.91	0.86	0.89	0.91
Sport	0.85	0.96	0.89	0.89	0.90
Book	0.92	0.94	0.95	0.96	0.96
Military	0.80	0.93	0.86	0.87	0.87
Computer	0.85	0.83	0.85	0.86	0.86
Space	0.93	0.90	0.85	0.87	0.88
Politics	0.92	0.90	0.88	0.90	0.91
Nature	0.68	0.93	0.86	0.88	0.89
Culture	0.59	0.54	0.89	0.91	0.93
Average	0.84	0.87	0.87	0.88	0.89

process of entity recognition, leading to a decline in the overall performance of the model; (iii) *Data distribution issue*. The imbalanced distribution of relationships in the dataset is another challenge. Although we introduced the Focal loss function as well as relationship weights to alleviate this problem, the model still tends to select the category with a large number of samples, resulting in poor performance on categories with few samples.

Table 10. The Compared Experimental Results between the Proposed Model and the Baseline Models on the DBpedia-WebNLG Dataset for OC

Category	Vicuna	Alpaca-LoRA	Our Model (T5-Small)	Our Model (T5-Base)	Our Model (T5-Large)
University	0.92	0.89	0.91	0.93	0.94
Musicalwork	0.89	0.85	0.91	0.92	0.93
Airport	0.92	0.94	0.94	0.95	0.96
Building	0.98	0.93	0.94	0.95	0.96
Athlete	0.92	0.94	0.93	0.94	0.95
Politician	0.89	0.92	0.94	0.96	0.96
Company	1.00	0.95	0.85	0.88	0.90
Celestialbody	0.97	0.96	0.91	0.93	0.94
Astronaut	0.87	0.87	0.89	0.91	0.92
Comicscharacter	0.97	0.93	0.82	0.86	0.88
Meanoftransportation	0.94	0.97	0.94	0.95	0.96
Monument	0.94	0.97	0.77	0.82	0.85
Food	0.94	0.92	0.95	0.96	0.97
Writtenwork	0.92	0.90	0.94	0.95	0.96
Sportsteam	0.91	0.90	0.93	0.95	0.95
City	0.98	0.96	0.95	0.96	0.97
Artist	0.89	0.83	0.94	0.95	0.96
Scientist	0.95	0.92	0.87	0.89	0.90
Film	0.94	0.82	0.91	0.93	0.94
Average	0.93	0.91	0.92	0.94	0.95

6 Conclusion

In this study, we have presented an end-to-end approach for generating domain knowledge graphs from unstructured text, driven by domain ontology. Our methodology integrates ontology embedding and PLMs to facilitate domain node recognition and relation extraction, resulting in the construction of coherent and domain-specific knowledge graphs. The evaluation of the Wikidata-TekGen dataset and the DBpedia-WebNLG dataset demonstrates our method's accuracy in constructing domain knowledge graphs.

Our work contributes to addressing the challenge of extracting domain-specific knowledge from textual corpora and represents a significant step towards supporting domain-specific reasoning, question-answering, and decision-making tasks. By systematically organizing and representing knowledge within specialized domains, our approach offers potential benefits in eliminating knowledge fragmentation and enhancing the consistency of domain knowledge.

6.1 Limitations

Despite our method accurately extracting domain knowledge graphs related to the ontology, it still has some limitations that warrant consideration: (i) The effectiveness of our method heavily relies on the quality and coverage of the domain ontology. Incomplete or inaccurate domain ontologies may lead to sub-optimal results in domain node recognition and relation extraction. (ii) The scalability of our approach may be limited when dealing with large-scale or rapidly evolving domains, as ontology maintenance and update processes could become cumbersome. (iii) The performance of our method may vary across different domains, depending on the availability of domain-specific textual corpora and the complexity of domain knowledge structures.

6.2 Future Work

To address the limitations identified, several avenues for future research can be explored. (i) Enhancing the quality and coverage of domain ontologies through automated ontology learning and enrichment techniques could improve the performance of our approach across diverse domains.

(ii) Investigating techniques for incremental ontology updates and dynamic adaptation of knowledge graphs to evolving domains would enhance the scalability and adaptability of our method. (iii) Exploring the integration of multimodal information sources, such as images and videos, could enrich the representation of domain knowledge graphs and enable a more comprehensive understanding and utilization of domain-specific information. (iv) Extending the evaluation of our method to additional datasets and real-world applications would provide further insights into its robustness and practical utility in various domains and tasks.

References

[1] Bilal Abu-Salih. 2021. Domain-specific knowledge graphs: A survey. *Journal of Network and Computer Applications* 185 (2021), 103076.

[2] Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. arXiv:2010.12688. Retrieved from https://arxiv.org/abs/2010.12688

[3] Jiaoyan Chen, Pan Hu, Ernesto Jimenez-Ruiz, Ole Magnus Holter, Denvar Antonyrajah, and Ian Horrocks. 2021. OWL2Vec*: Embedding of OWL ontologies. *Machine Learning* 110, 7 (2021), 1813–1845.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805. Retrieved from https://arxiv.org/abs/1810.04805

[5] Pierre L. Dognin, Igor Melnyk, Inkit Padhi, Cicero Nogueira dos Santos, and Payel Das. 2020. DualTKB: A dual learning bridge between text and knowledge base. arXiv:2010.14660. Retrieved from https://arxiv.org/abs/2010.14660

[6] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv:1508.01991. Retrieved from https://arxiv.org/abs/1508.01991

[7] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S. Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems* 33, 2 (2021), 494–514.

[8] Danh Le-Phuoc, Hoan Nguyen Mau Quoc, Hung Ngo Quoc, Tuan Tran Nhat, and Manfred Hauswirth. 2016. The graph of things: A step towards the live knowledge graph of connected things. *Journal of Web Semantics* 37 (2016), 25–35.

[9] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv:1910.13461. Retrieved from https://arxiv.org/abs/1910.13461

[10] Linfeng Li, Peng Wang, Jun Yan, Yao Wang, Simin Li, Jinpeng Jiang, Zhe Sun, Buzhou Tang, Tsung-Hui Chang, Shenghui Wang, et al. 2020. Real-world data medical knowledge graph: Construction and applications. *Artificial Intelligence in Medicine* 103 (2020), 101817.

[11] Wei-Lin Chiang Lianmin Zheng, Ying Sheng and Lisa Dunlap. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. Retrieved November 7, 2024 from https://lmsys.org/blog/2023-03-30-vicuna/

[12] Jinjiao Lin, Yanze Zhao, Weiyuan Huang, Chunfang Liu, and Haitao Pu. 2021. Domain knowledge graph-based research progress of knowledge representation. *Neural Computing and Applications* 33 (2021), 681–690.

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988.

[14] Zhihuang Lin, Dan Yang, and Xiaochun Yin. 2020. Patient similarity via joint embeddings of medical knowledge graph and medical entity descriptions. *IEEE Access* 8 (2020), 156663–156676.

[15] Dongfang Lou, Zhilin Liao, Shumin Deng, Ningyu Zhang, and Huajun Chen. 2021. MLBiNet: A cross-sentence collective event detection network. arXiv:2105.09458. Retrieved from https://arxiv.org/abs/2105.09458

[16] Igor Melnyk, Pierre Dognin, and Payel Das. 2022. Knowledge graph generation from text. arXiv:2211.10511. Retrieved from https://arxiv.org/abs/2211.10511

[17] Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F. Enguix, and Kusum Lata. 2023. Text2KGBench: A benchmark for ontology-driven knowledge graph generation from text. In *Proceedings of the International Semantic Web Conference*. Springer, 247–265.

[18] Fabian Neuhaus. 2018. What is an ontology? arXiv:1810.09171. Retrieved from https://arxiv.org/abs/1810.09171

[19] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review* 56, 11 (2023), 130071–13102.

[20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[21] Claire Gardent Thiago Castro Ferreira, Chris van der Lee Nikolai Ilinykh, Diego Moussallem Simon Mille, and Anastasia Shimorina. 2020. GitHub - WebNLG/WebNLG-Text-to-Triples: WebNLG+ Challenge 2020: The Automatic Evaluation Script for the Text-to-RDF Task — github.com. Retrieved February 3, 2024 from https://github.com/WebNLG/WebNLG-Text-to-triples

[22] Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. arXiv:2010.11967. Retrieved from https://arxiv.org/abs/2010.11967

[23] Cheng Xie, Beibei Yu, Zuoying Zeng, Yun Yang, and Qing Liu. 2020. Multilayer Internet-of-Things middleware based on knowledge graph. *IEEE Internet of Things Journal* 8, 4 (2020), 2635–2648.

[24] Hongbin Ye, Ningyu Zhang, Hui Chen, and Huajun Chen. 2022. Generative knowledge graph construction: A review. arXiv:2210.12714. Retrieved from https://arxiv.org/abs/2210.12714

[25] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. 2018. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *Proceedings of the International Conference on Machine Learning*. PMLR, 5708–5717.

[26] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 36, 46595–46623.

[27] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 1529–1537.

[28] Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2023. A comprehensive survey on automatic knowledge graph construction. arXiv:2302.05019. Retrieved from https://arxiv.org/abs/2302.05019

[29] Dongzhuoran Zhou, Baifan Zhou, Zhuoxun Zheng, Ahmet Soylu, Gong Cheng, Ernesto Jiménez-Ruiz, Egor Kostylev, and Evgeny Kharlamov. 2022. Ontology reshaping for knowledge graph construction: Applied on Bosch welding case. In *Proceedings of the International Semantic Web Conference*. Springer, 770–790.

[30] Dongzhuoran Zhou, Baifan Zhou, Zhuoxun Zheng, Ahmet Soylu, Ognjen Savkovic, Egor V. Kostylev, and Evgeny Kharlamov. 2022. ScheRe: Schema reshaping for enhancing knowledge graph construction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 5074–5078.

[31] Guangtong Zhou, Selasi Kwashie, Yidi Zhang, Michael Bewong, Vincent M. Nofong, Junwei Hu, Debo Cheng, Keqing He, Shanmei Liu, and Zaiwen Feng. 2023. FastAGEDs: Fast approximate graph entity dependency discovery. In *Proceedings of the International Conference on Web Information Systems Engineering*. Springer, 451–465.

[32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2223–2232.

Received 9 March 2024; revised 16 August 2024; accepted 9 November 2024