

UniQ-Gen: Unified Query Generation across Multiple Knowledge Graphs

Daniel Vollmers¹[0000-0002-5324-4952], Nikit Srivastava¹[0009-0004-5164-4911],
Hamada M. Zahera¹[0000-0003-0215-1278], Diego
Moussallem¹[0000-0003-3757-2013], and Axel-Cyrille Ngonga
Ngomo¹[0000-0001-7112-3516]

Data Science Group, Paderborn University, Germany
{daniel.vollmers, nikit.srivastava, hamada.zahera, diego.moussallem,
axel.ngonga}@uni-paderborn.de

Abstract. Generating SPARQL queries is crucial for extracting relevant information from diverse knowledge graphs. However, the structural and semantic differences among these graphs necessitate training or fine-tuning a tailored model for each one. In this paper, we propose UniQ-Gen, a unified query generation approach to generate SPARQL queries across various knowledge graphs. UniQ-Gen integrates entity recognition, disambiguation, and linking through a BERT-NER model and employs cross-encoder ranking to align questions with the Freebase ontology. We conducted several experiments on different benchmark datasets such as LC-QuAD 2.0, GrailQA, and QALD-10. The evaluation results demonstrate that our approach achieves performance equivalent to or better than models fine-tuned for individual knowledge graphs. This finding suggests that fine-tuning a unified model on a heterogeneous dataset of SPARQL queries across different knowledge graphs eliminates the need for separate models for each graph, thereby reducing resource requirements.

Keywords: SPARQL Generation · Question Answering over Knowledge Graphs · Large Language Models · KGQA

1 Introduction

Large language models (LLMs) have recently shown significant performance in various NLP tasks, including answering questions on Knowledge Graphs (KGQA) [1]. These models are often fine-tuned on a domain-specific dataset (e.g., QALD-10) to convert natural text to corresponding logical forms like a SPARQL query [2] or S-expression [3]. However, training or fine-tuning LLMs is a resource-intensive process that requires a lot of computational resources such as extensive GPU hours [4]. Current approaches typically train or fine-tune an LLM on a single domain-specific dataset or knowledge graph [2]. However, these methods require further tuning when applied to new domains or knowledge graphs. This is due to knowledge graphs (e.g.g, Freebase [5], Wikidata [6], and DBpedia [7]) have

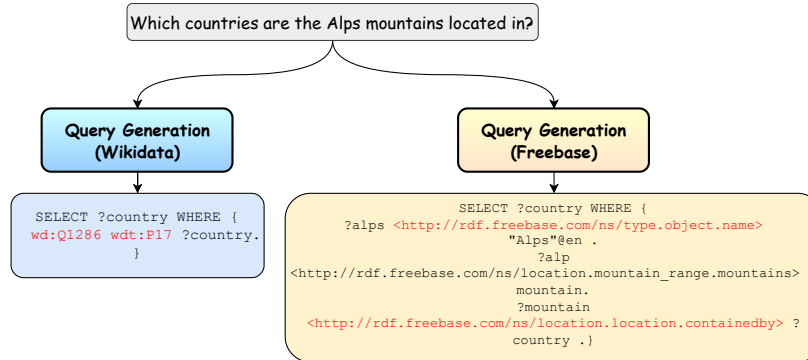


Fig. 1: An example of two individual models for generating SPARQL queries across different knowledge graphs.

significant variances in data representation. For instance, Wikidata represents *The Alps Mountains* with semantic identifiers like *wd:Q1286*, while Freebase uses encoded identifiers such as *?alps <http://rdf.freebase.com/ns/type.object.name> Alps @en* as shown in Figure 1. Moreover, Freebase hierarchically organizes semantic relations between entities, which differs from the structures in other knowledge graphs. These differences pose challenges in developing systems that are compatible with multiple knowledge graphs. Each knowledge graph requires a different entity linking and query generation components. Consequently, adapting to these differences needs re-training or fine-tuning of LLMs to ensure effective performance.

In this paper, we propose a unified approach to fine-tune a single large language model for generating SPARQL queries across different knowledge graphs. Our approach involves fine-tuning one LLM on multiple knowledge graphs rather than training separate models for each graph. To achieve this, we combine training data from multiple knowledge graphs in one dataset. This joint fine-tuning of one LLM allows to better generalization across different data representations and structures. As result, a single model for multiple KGs significantly reduces the resource requirements in a productive environment, as only one model needs to be deployed rather multiple models for different knowledge graphs. To evaluate the performance of our approach, we perform a comparative analysis between models tailored to each knowledge graphs and our unified model. In our experiments, we use two different knowledge graphs (Wikidata and Freebase), which have significant different in their SPARQL queries. Furthermore, we extract relevant information (e.g., entities, relations, types) from the knowledge graph and investigate which information has the most impact of the performance of query generation. The evaluation results demonstrate that our approach achieves performance comparable to, or the same as, single KG models, reducing the need for training or fine-tuning the model separately for each knowledge graph. We summarize the main contributions of this paper as follows:

- We propose a unified approach for generating SPARQL queries for multiple knowledge graphs.
- Our approach achieves equivalent or comparable performances as individual models (which tailored for each KG), eliminating the need for separate training or fine-tuning an LLM for each KG.
- Incorporating relevant information (e.g., entities, relations, and types) within the LLM prompt improves the performance of SPARQL query generation.
- The source code and datasets used in our experiments are publicly available.¹

2 Related Work

2.1 Query Generation on a Single Knowledge Base

Wikidata Recent approaches [2, 8] treat the query generation problem as a translation problem. The task is to translate a natural language question into a SPARQL query. As inputs these models use the natural language question itself, plus linked knowledge such as entities and relations from the knowledge base [2]. Applying fine-tuning techniques to language models also become increasingly popular over recent years. Other approaches use patterns for generating a set of candidate queries [9, 10]. Afterward, a ranking approach is applied to computing the final prediction. Commonly all approaches apply entity linking, which usually consists of a span detection step and a disambiguation step. The output of the disambiguation is either a ranked list of entities per span or only one entity per span in the case of an end-to-end entity linking setup.

Freebase On the Freebase knowledge base, semantic parsing is usually solved by iteratively predicting and ranking queries in the form of S-Expressions [2, 3, 11]. For example, the RnG-KBQA [3] framework uses a combination of ranking and query generation for predicting queries. Different from the translation approaches used for Wikidata question answering, candidate queries are generated and introduced in the generation model. Other approaches such as Pangu [12], construct queries, by iteratively extending and ranking a set of query sub-plans. All models share the characteristic of computing a large number of sub-plans, which demand significant resources in terms of GPU memory and time, making an end-to-end implementation usually unavailable.

2.2 Query Generation on Multiple Knowledge Graphs

Since the Semantic Web’s inception, various methods have been developed for semantic parsing over multiple or interlinked knowledge graphs. One early method, PowerAqua by Lopez et al. [13], uses semantic similarity between ontology terms and user queries to generate query triples, which an inference engine uses to retrieve answers. Its updated version [14] works on large KGs like DBpedia²

¹ <https://github.com/dice-group/KATRINA>

² <https://www.dbpedia.org/>

but struggles with scalability, performance, and effectiveness, especially with large-scale data, complex queries, and integrating data from different ontologies, requiring significant effort for updates and maintenance. SINA, introduced by Shekarpour and Auer [15], is a data-semantics-aware keyword search approach that converts natural language and keyword queries into SPARQL queries for accessing interlinked KGs within the Linked Open Data Cloud. It uses a hidden Markov model for query disambiguation and resource identification, leveraging Linked Data topology to construct federated SPARQL queries for information retrieval from multiple KGs. However, this process is computationally complex, and the keyword-based approach can overlook the question’s syntax.

OQA by Fader et al. [16] enhances answer accuracy and coverage by integrating curated KBs like Freebase with automatically extracted KBs, employing NLP for query translation and paraphrase-driven learning for query variability. In a similar manner, MULTIQUE by Bhutani et al. [17] uses semantic parsing through neural networks to handle complex queries by combining curated and extracted KBs. These approaches rely heavily on textual data, making it challenging to manage and computationally expensive to extract relevant information on-demand. Zhang et al. [18] employed a rule-based method to handle queries across multiple KGs by identifying resources, forming triple patterns, aligning variables, and performing joint inference to create accurate SPARQL queries. However, string matching for entity linking can cause mismatches and missed entities due to its inability to disambiguate similar names and handle naming variations accurately. Neelam et al. [19] introduced SYGMA, which streamlines query generation through KB-agnostic "Question Understanding" and KB-specific "Question Mapping & Reasoning." It uses abstract meaning representation to create a KB-agnostic lambda expression, refined with specific KB details before being converted into a SPARQL query using a rule-based system. SYGMA’s modular design aids generalization but is sensitive to individual module performance, particularly relation linking.

3 Approach

Figure 2 shows an overview of approach (UniQ-Gen) for generating SPARQL queries for multiple knowledge graphs (e.g., Wikidata and Freebase) using a single model. We achieve this by fine-tuning the T5 [20] model on a mixed dataset, containing training examples of (natural questions, SPARQL-for-Freebase) and (question, SPARQL-for-Wikidata). In this way, the fine-tuned T5 model learns how to generate SPARQL queries for both graphs, rather than fine-tuning two separate models for each graph. Accordingly, we reduce the computational and maintenance costs associated with fine-tuning and managing separate models for each knowledge graph. The following sections describes the details of each module in our approach, including *Knowledge Extraction* and *Query Generation*.

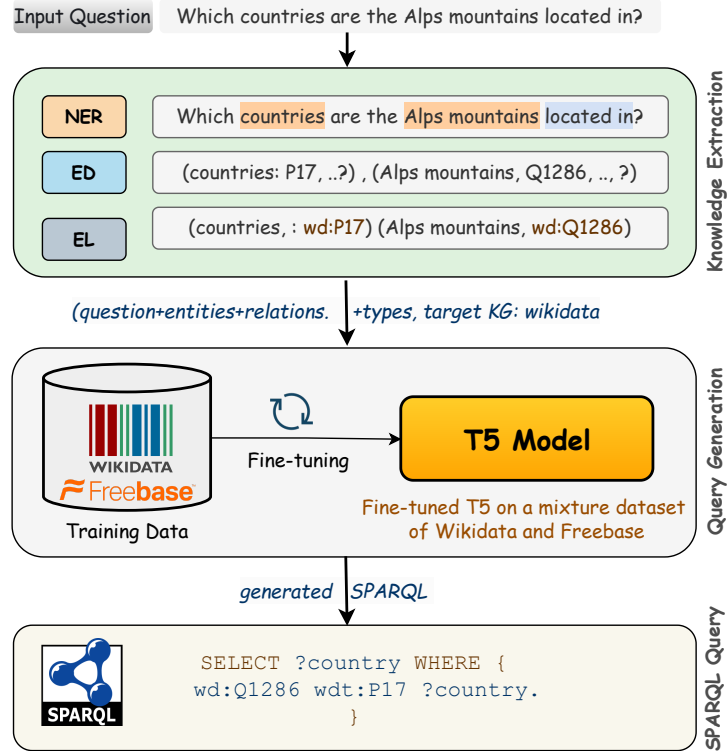
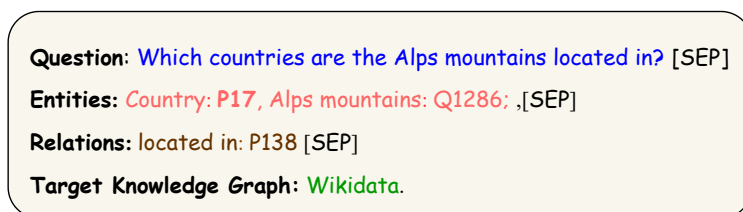


Fig. 2: Our approach (UniQ-Gen) for generating SPARQL queries using one model for multiple knowledge graphs.

3.1 Knowledge Extraction

The knowledge extraction process involves three main tasks: *named entity recognition* (NER), *entity disambiguation* (ED) and *entity linking* (EL). Named entity recognition identifies and classifies entity spans within the text[21], while entity disambiguation associates these spans with corresponding entities in the target knowledge graph. First, we extract relevant information (e.g., entities, relations, and types) from the input question. Noteworthy, This process varies between Wikidata and Freebase due to their different structures and representations of information.

Extracting Knowledge from Wikidata Many question-answering approaches rely on pre-built frameworks, such as DBpedia Spotlight, which generally yield satisfactory results, due to the significant resources required to develop an efficient linking systems. However, these frameworks often struggle with identifying



Question: Which countries are the Alps mountains located in? [SEP]
Entities: Country: P17, Alps mountains: Q1286; ,[SEP]
Relations: located in: P138 [SEP]
Target Knowledge Graph: Wikidata.

Fig. 3: An example input for generating SPARQL query with Wikidata

and categorizing relations and types. For instance, in a query like “Which mountains are located in the US?”, a typical NER framework would only recognize “US” as an entity. For accurate query generation, it is essential to also link the term “mountains” to the knowledge graph. Additionally, these frameworks usually fail to predict relationships between entities, a critical factor for enhancing QA system accuracy. To address these limitations, we employ the following methods:

- *Entity Recognition, Disambiguation, and Linking:* We use Flair [22], a state-of-the-art framework that employs an LSTM network with contextual string embeddings to accurately recognize and classify entity spans within text. For entity disambiguation, we use GENRE [23], which applies an autoregressive transformer architecture based on the pre-trained BART model with constrained decoding and beam search to predict Wikipedia titles. We link these titles with Wikidata using a dictionary, assuming each entity has a unique label mapping to a single Wikipedia title.
- *Extracting Relations and Types:* We use a fine-tuned T5 model to predict types and relations that are not directly mapped to spans in the input sequence. For example, in the question “Which is the highest mountain in the US?”, the relation “is located in” is needed but not directly present in the text. We extend target labels to include relations, e.g., “Who is the spouse of Obama[SEP]entities: Barack Obama, relations: spouse.”

We combine the outputs of these methods and use a dictionary to map Wikidata labels to their URIs, assuming each label is unique within the knowledge graph. These outputs are then used as input for the *Query Generation* module, as shown in Figure 2. The input includes: the *question*, *entities*, *relations* and a *target knowledge graphs*. All are concatenated using a separation token, *[SEP]* as shown in Figure 3.

Extracting Knowledge from Freebase Many types and relations in Freebase share identical labels due to its hierarchical ontology, making pre-built methods (e.g., Flair) for linking entities, relations, and types in Wikidata inadequate. Therefore, we adapt existing methods for extracting knowledge from Freebase as follows:

Table 1: Training Samples

Question	Entities	Relations	Target	SPARQL
Is Kevin Costner owner of Fielders Stadium?	wd:Q11930 wd:Q5447154	wdt:P1830	wikidata	ASK...
how many hadrons are in the family meson?	physics.hadron, m.04_rh	physics.particle. family	freebase	SELECT (COUNT...
What periodical literature does Delta Air Lines use as a moutpiece?	wd:Q188920 wd:Q1002697	wdt:P2813 wdt:P31	wikidata	SELECT...

- *Entity Recognition, Disambiguation, and Linking*: Our approach aligns with the RNG-KGQA framework [3], employing a BERT-NER model to accurately detect entity mentions within the text. For entity disambiguation, we use a pre-trained BERT-based model that leverages relation information linked with each entity, thereby improving the ranking of the target entity. For entity linking, our approach matches these mentions with surface forms from the Freebase KG, ranks them using popularity scores, and retains the top 5 candidates. The ranking model employed is a cross-encoder model, as described in Equation 1.
- *Types and Relations Linking*: Our approach follows the schema retrieval method from the TIARA Framework [11], using a cross-encoder ranker to rank relations and types from the Freebase Ontology. The score for question (x) and a schema (c) is computed as:

$$s(x, c) = \text{Linear}(\text{BertCls}([x; c])) \quad (1)$$

where BertCLS represents the *CLS* token from a BERT-encoder [11]. We use the top-5 relations and types as input for our query generation model.

3.2 Query Generation

We employ the T5 model in this module, which has demonstrated promising results in query generation task [2, 3]. In particular, we fine-tune the T5 model on diverse training dataset containing SPARQL queries from multiple knowledge graphs (Wikidata and Freebase). Table 1 show show some training examples of the dataset used to fine-tune the T5 model, including examples for question-to-SPARQL_(wikidata) and question-to-SPARQL_(freebase). One example includes: an input question, the (entities relations, and types), a target KG (e.g., Wikidata), and the ground-truth SPARQL query. During the training phase, these samples are shuffled into one batch and the T5 model is trained to generate SPARQL queries based on the input questions and the target knowledge graph. This process involves fine-tuning the model to accurately map the linguistic structures of the questions to the target knowledge graph.

Language models often encounter challenges with special tokens such as $\{$ or $\}$, which are integral to SPARQL queries. To address this issue, we replace

these tokens in the target strings as follows: `{` with `_cbo_` and `}` with `_cbc_`. We normalize variables by replacing the leading `?`-token. For example, a variable like `?uri` is replaced with `_var_<id>`, or `_result_<id>` if it is a part of the result set. Here, `<id>` represents a number, as a query can contain multiple variables. During the T5 model’s inference, we revert these substitutions to generate a valid SPARQL query. For variables, we only replace the leading underscore with the `?`-token. Note that types are included under the entity tag to shorten the input string, as entities and types are used similarly in SPARQL queries. For Freebase the approach is the same, except that the string `target:freebase` is appended at the end of the input string instead of the string

4 Experiments

We conducted our experiments to answer the following research questions:

- **RQ₁**: How well does our unified model perform compare to individual language models, trained on single knowledge graphs, for SPARQL query generation?
- **RQ₂**: How does our unified model perform compared to state-of-the-art baselines?
- **RQ₃**: What is the impact of incorporating knowledge such as entities, relations, and types on the performance of the query generation models?
- **RQ₄**: How does integrating knowledge from different resource extraction frameworks into the training datasets affect the performance of query generation models?

4.1 Datasets

In our experiments, we use different benchmark datasets, namely, LC-QuAD 2.0 [24], and QALD-10 [25] on Wikidata knowledge graph and GrailQA [26] on Freebase knowledge graph. We briefly describe these datasets as follows:

- **LC-QuAD 2.0** [24] is a large dataset with $30k$ questions in English, each paired with a corresponding Wikidata query. The dataset is divided into a training subset with around $24k$ questions and a test set with $6k$ questions.
- **QALD-10** [25] this dataset is manually annotated with 394 question-query pairs across different languages. It is an updated version of the QALD-9 dataset, referred to as QALD-9-plus. As the dataset is comparably small, we initially trained our model on the LC-QuAD 2.0 dataset as a pre-trained model (i.e., foundation model), then fine-tuned it on the QALD-9-plus dataset, following the same training strategy as [27].
- **GrailQA** [26] This dataset is a large, crowdsourced collection from Freebase KG, containing around $64k$ questions. The dataset provides not only SPARQL queries but also contains S-expressions as alternative logical representations. The dataset is divided into a train split with $44K$ questions a development with $6k$ and a test split with $13k$.

4.2 Experiment Setup

In our experiments, we use Nvidia-H100 GPUs for efficient models training. The T5-base model is used as a foundation model, since it is widely used in query generation research and ensures comparability with other methodologies. Each model is trained for a maximum 50 epochs with an early stopping mechanism to mitigate overfitting. For our unified model, we combine the LC-QuAD 2.0 [26] and GrailQA [24] datasets. Despite GrailQA containing approximately 20k more questions than LC-QuAD 2.0, our experiments show no significant impact on performance, indicating that data balancing is unnecessary. For the QALD-9-plus dataset, we fine-tuned our pre-trained LC-QuAD and Freebase model and supplemented the training data with an equivalent number of randomly selected entries from the GrailQA dataset to match the volume of the QALD-9-plus dataset.

4.3 Evaluation

We evaluated the performance of SPARQL query generation using the GERBIL-QA framework [28]. This framework is well-established with different benchmark datasets and evaluation metrics, including Micro-F1, Macro-F1, and Macro-F1 QALD, which are used in the QALD challenge.³ We adopted the same evaluation setting of Usbeck et al. [29], and also included metrics such as Macro Precision, Macro Recall, Macro F1, and Macro F1-QALD. The Macro F1 score is calculated per question and uses the geometric mean for the final score. For clarity, we refer to Macro Precision, Macro Recall, Macro F1, and Macro F1-QALD as Precision, Recall, F1, and F1-QALD, respectively. We set up the Virtuoso Triple Store for Freebase following instructions from the GrailQA repository⁴, and for Wikidata, we used the official Triple Store.⁵ Each query is generated regardless of whether the triple store returns an empty result set. We did not verify the correctness of the generated queries; therefore, improperly formatted queries result in an empty set of results. Finally, we compiled all outputs into a QALD-formatted JSON file and submitted it using the ‘*upload result file*’ function in the GERBIL-QA framework to calculate the final results.

4.4 Results and Discussion

Comparison of Unified Model and Single KG models (RQ₁) To answer this question, we implemented two different variants of models: the first model is a unified model which is fine-tuned on a heterogeneous dataset of SPARQL queries for Wikidata and Freebase. The other variants are single models tailored for Freebase and Wikidata, which are trained only on the train subsets for the respective KG.

³ <https://www.nliwod.org/challenge>

⁴ <https://github.com/dki-lab/GrailQA>

⁵ <https://query.wikidata.org/>

Table 2: Comparison of joint and single KG models **RQ₁**

Dataset	Experiment	Precision	Recall	F1	F1	QALD
LC-QuAD	Gold resources joint	0.88	0.87	0.88		0.92
	Gold resources only LC-QuAD data	0.88	0.88	0.89		0.92
	End-to-end joint	0.47	0.47	0.47		0.62
	End-to-end only LC-Quad data	0.42	0.43	0.42		0.59
GrailQA	Gold resources joint	0.51	0.59	0.54		0.67
	Gold resources only Grail QA data	0.54	0.62	0.57		0.68
	End-to-end joint	0.3	0.34	0.31		0.49
	End-to-end only Grail QA data	0.3	0.34	0.31		0.49
QALD-10	Joint gold resources	0.49	0.49	0.49		0.64
	Gold resource only QALD	0.47	0.47	0.47		0.62
	Joint end-to-end model	0.44	0.45	0.44		0.60
	End-to-end only QALD-10	0.45	0.45	0.45		0.61

We conducted two experiments using these models. The first one, referred as the *gold-resource experiment*, involved evaluating the models using high-quality input data derived from the test splits. This evaluation process involves extracting entities, types, and relations from the test split, then incorporated as additional information into the model’s input. In contrast, the second experiment, referred to as the *end-to-end experiment*, used knowledge acquired from our *Knowledge Extraction* module approach as direct input for the model. Our analysis on the GrailQA dataset shows that in the end-to-end configuration, the unified model performs equivalently to the single KG models. Conversely, in the gold-resource setup, the performance disparity between the unified and single KG models is minimal. Similarly, on the LC-QuAD dataset, the end-to-end performance of the unified model surpasses the single KG model. For the GrailQA dataset, the difference in performance between models trained and evaluated using gold resources was negligible. On the QALD-10 dataset, the unified model’s performance with gold-resource input slightly outperforms the single KG model. In the end-to-end experiment, the single KG model achieves a higher F-Measure by one percent compared to the unified model, though this difference is marginal, consistent with results from other datasets. Overall, the results show that the unified model achieves comparable or equivalent performance as single KG models across all experiments.

Comparison with state-of-the-art models (RQ₂)

Results on Wikipedia Datasets. We conducted two experiments on the LC-QuAD 2.0 dataset to compare the performance with state-of-the-art baselines in SPARQL query generation. First, we evaluated the performance of our system using golden entities and relations as inputs. Remarkably, the performance of our approach is aligned with the performance of Banerjee et al. [2], which uses also the T5-small model. This similarity in performance can be attributed to

Table 3: Baseline comparison on Wikidata datasets \mathbf{RQ}_2

Approach	F1 QALD	Approach	F1
Borroto et al. [8]	0.59	GPT 3.5. [1]	0.39
Diefenbach et al. [9]	0.58	Chat GPT [1]	0.42
Shivashanker et al. [30]	0.49	Joint model end-to-end	0.46
Baramiia et al. [31]	0.43	Single KG model end-to-end	0.42
Joint model end-to-end	0.60		
Single KG model end-to-end	0.61		

(b) End-to-end Results on the LC-QuAD 2.0 datasets

(a) Results on the QALD-10 dataset

Approach	F1 QALD
Banerjee et al. [2] (T5 base)	0.91
Banerjee et al. [2] (T5 small)	0.92
Banerjee et al. [2] (PGN-BERT)	0.86
Joint model gold knowledge	0.92
Single KG model gold knowledge	0.92

(c) Baseline comparison on the LC-QuAD dataset with gold knowledge

Table 4: Baseline comparison on the GrailQA dataset \mathbf{RQ}_2

Approach	Precision	Recall	F1	F1 QALD
Shu et al. 2022 [11]	0.59	0.71	0.62	0.71
Shu et al. 2024 [32]	0.59	0.71	0.62	0.71
Yu et al. 2023 [12]	0.64	0.79	0.68	0.72
Yu et al. 2024 [33]	0.62	0.79	0.67	0.71
Joint model end-to-end	0.3	0.34	0.31	0.49
Joint model gold	0.51	0.59	0.54	0.67

the similar inputs, despite the fact that our study employs a unified model for multiple KGs. In an end-to-end setup, our unified model achieves outperforming results compared to the model by Tan et al. [1] in terms of Macro F1-measure. For additional comparison, we refer to the results from the KGQA-leaderboard⁶. Furthermore, on the QALD-10 dataset, our approach achieves state-of-the-art results in terms of F1-QALD measure. Across all Wikipedia datasets, our unified model achieves the same or comparable results as single models.

Results on Freebase (GrailQA) dataset In the next step, we compare our results with the baseline models on the GrailQA dataset. It is important to note

⁶ <https://github.com/KGQA/leaderboard/>

Table 5: comparison of different input data **RQ₃**

Dataset	Experiment	Precision	Recall	F1	F1	QALD
LC-QuAD	Baseline	0.37	0.37	0.37		0.53
	Entities & types	0.7	0.71	0.71		0.81
	Golden resources	0.88	0.87	0.88		0.92
GrailQA	Baseline	0.2	0.24	0.21		0.39
	Golden entities & Golden types	0.33	0.4	0.35		0.54
	Golden resources	0.51	0.59	0.54		0.67

that the evaluation script used by GrailQA differs from ours, since it assesses queries in the form of S-expressions instead of SPARQL queries. To evaluate on the Gerbil-QA framework, we only included systems that provide their results in a format compatible with QALD. Our findings indicate that the results are not as strong as those achieved by the baseline models. This is mainly because the GrailQA dataset focuses on queries that require detailed knowledge of the Freebase structure, especially its hierarchical ontology. Existing methods usually generate and rank sub-queries, making it possible to learn the knowledge graph structure. However, these methods are resource intensive, as they need to generate and rank a large set of queries, resulting in a slow processing [11, 33], which may not be suitable for use in a production environment.

Influence of KG knowledge on the model performance (RQ₃) To answer this question, we conducted different experiments on extensive datasets, LC-QuAD 2.0 and GrailQA, to ensure that the models are trained on sufficient number of entities and relations. We carried out three experiments for each knowledge graph: i) The first experiment, referred to as a *baseline* experiment that used only the question as input. ii) The second experiment includes additional information such as entities and types as well as the question as an input. iii) The third experiment is referred a golden-resource experiment, where entities, types and relations are included with the question as an input.

Table 5 shows the evaluation results of all experiments, indicating that the model’s performance improved with the additional information (entities, relations, and types). Specifically, adding entities and types led to a significant performance boost. While relations also improved performance with less noticeable compared to entities. Overall, the LC-QuAD dataset demonstrates better results (F-measure of 0.88), with the additional information. In contrast, the GrailQA dataset reaches an F-measure of only 0.54. Therefore, future research should focus on adding more detailed information, such as the structural aspects of knowledge graphs.

Experiments with different training data (RQ₄) Typically, training data is enriched with additional information by adding golden-resource entities and relations from the target SPARQL queries. This method often causes the model

Table 6: Comparison of different training setups \mathbf{RQ}_4

Dataset	Experiment	Precision	Recall	F1	F1	QALD
LC-QuAD	Gold resources	0.88	0.87	0.88		0.92
	Gold resources inc. KE	0.84	0.85	0.84		0.90
	End-to-end	0.46	0.46	0.46		0.6
	End-to-end inc. KE	0.47	0.47	0.47		0.62
GrailQA	Gold resources	0.33	0.4	0.35		0.54
	Gold resources inc. KE	0.51	0.59	0.54		0.67
	End-to-end	0.12	0.15	0.12		0.26
	End-to-end inc. KE	0.3	0.34	0.31		0.49
QALD-10	Gold resources inc. KE	0.49	0.49	0.49		0.64
	Gold resources	0.48	0.49	0.48		0.64
	End-to-end inc. KE	0.44	0.45	0.44		0.60
	End-to-end model	0.27	0.28	0.28		0.43

to duplicate the input information without distinguishing between relevant and irrelevant data. We address this issue by extracting relevant information using our Knowledge Extraction module and include in the training data. To achieve this, we carried out several experiments per dataset, by training the model only with the golden-resource information from the dataset and additional input from knowledge extraction. Afterward, we performed the same experiments as in section 4.4, evaluating the models with both gold input and in an end-to-end setup.

Our findings indicate that including relevant information improves the model performance across all datasets in the end-to-end setup. However, on the LC-QuAD dataset, the performance improvement is minimal compared to training with gold-standard data. On the GrailQA dataset, we achieve a significant improvement, as the results improved from 0.26 to 0.49 in terms of the F1 QALD measure. Similarly, on the QALD dataset, the model’s performance improved from 0.43 to 0.6. These variations in performance across datasets can be attributed to different Knowledge Extraction methods in linking data. For instance, entity mentions in LC-QuAD closely align with those in Wikidata knowledge graph, whereas QALD-10 presents greater ambiguity. For example, the question: “*Do the princes William and Harry share the same mother?*”, where the entities are referred to only by their first names. In the evaluation setup, with golden-resource information, we observe a performance improvement only on the GrailQA dataset. This is not surprising, as incorporating golden information can introduce noise into the model inputs.

5 Conclusion and future work

This paper presents UniQ-Gen, a unified approach for fine-tuning a single model to generate SPARQL queries across different knowledge graphs. Our results demonstrate that training a unified model on a heterogeneous dataset (e.g., including samples from Wikidata and Freebase) achieves comparable performance

to single models for individual knowledge graphs, eliminating the need for separate models for each graph. Moreover, incorporating additional information such as entities, relations, and types, significantly enhances the performance of query generation models. While there are many effective solutions for entity linking, accurate and efficient relation linking remains a challenge in the field of knowledge graph question answering. However, our one-shot query generation approach lacks the incorporating of structural information about the knowledge graph. In our future work, we plan to address this limitation by including structural information (e.g., hierarchical relationships between entities) in our unified model. Furthermore, we will also adapt our approach to handle structural differences between knowledge graphs by integrating KG-specific structural knowledge.

Acknowledgement

This work has been supported by the German Federal Ministry of Education and Research (BMBF) within the projects, COLIDE (grant no 01I521005D), KIAM (grant no 02L19C115), the European Union’s Horizon Europe research and innovation programme (grant No 101070305), and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 – 438445824.

References

- [1] Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. Can chatgpt replace traditional kbqa models? an in-depth analysis of the question answering performance of the gpt llm family, 2023.
- [2] Debayan Banerjee, Pranav Ajit Nair, Jivat Neet Kaur, Ricardo Usbeck, and Chris Biemann. Modern baselines for sparql semantic parsing. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’22*. ACM, July 2022. <https://doi.org/10.1145/3477495.3531841>. URL <http://dx.doi.org/10.1145/3477495.3531841>.
- [3] Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering, 2022.
- [4] Junjie Yin, Jiahao Dong, Yingheng Wang, Christopher De Sa, and Volodymyr Kuleshov. Modulora: Finetuning 2-bit llms on consumer gpus by integrating with modular quantizers. *TMLR*, 2024.
- [5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD ’08*, page 1247–1250, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581026. <https://doi.org/10.1145/1376616.1376746>. URL <https://doi.org/10.1145/1376616.1376746>.

- [6] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014. ISSN 0001-0782. <https://doi.org/10.1145/2629489>. URL <https://doi.org/10.1145/2629489>.
- [7] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07*, page 722–735, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3540762973.
- [8] Manuel Alejandro Borroto, Francesco Ricca, and Bernardo Cuteri. A system for translating natural language questions into sparql queries with neural networks: Preliminary results *discussionpaper*. In *SEBD 2021: Italian Symposium on Advanced Database Systems*, pages 226–234, Aachen, Germany, 2021. RWTH Aachen.
- [9] Dennis Diefenbach, A. Both, K. Singh, and P. Maret. Towards a question answering system over the semantic web. *Semantic Web*, 11:421–439, 2020. <https://doi.org/10.3233/SW-190343>.
- [10] Daniel Vollmers, Richa Jalota, Diego Moussallem, Hardik Topiwala, Axel-Cyrille Ngonga Ngomo, and Ricardo Usbeck. *Knowledge Graph Question Answering Using Graph-Pattern Isomorphism*, pages 103–117. IOS Press, August 2021. <https://doi.org/10.3233/ssw210038>. URL <http://dx.doi.org/10.3233/SSW210038>.
- [11] Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje F. Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. Tiara: Multi-grained retrieval for robust question answering over large knowledge bases, 2022.
- [12] Yu Gu, Xiang Deng, and Yu Su. Don't generate, discriminate: A proposal for grounding language models to real-world environments, 2023.
- [13] Vanessa Lopez, Enrico Motta, and Victoria Uren. Poweraqua: fishing the semantic web. In *Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications, ESWC'06*, page 393–410, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540345442. https://doi.org/10.1007/11762256_30. URL https://doi.org/10.1007/11762256_30.
- [14] Vanessa Lopez, Miriam Fernández, Enrico Motta, and Nico Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semant. Web*, 3(3):249–265, jul 2012. ISSN 1570-0844.
- [15] Saeedeh Shekarpour and Sören Auer. "sina: semantic interpretation of user queries for question answering on interlinked data" by saeedeh shekarpour with prateek jain as coordinator. *SIGWEB Newsl.*, 2014(Summer), jul 2014. ISSN 1931-1745. <https://doi.org/10.1145/2641730.2641733>. URL <https://doi.org/10.1145/2641730.2641733>.
- [16] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, page 1156–1165, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569.

- <https://doi.org/10.1145/2623330.2623677>. URL <https://doi.org/10.1145/2623330.2623677>.
- [17] Nikita Bhutani, Xinyi Zheng, Kun Qian, Yunyao Li, and H. Jagadish. Answering complex questions by combining information from curated and extracted knowledge bases. In Ahmed Hassan Awadallah, Yu Su, Huan Sun, and Scott Wen-tau Yih, editors, *Proceedings of the First Workshop on Natural Language Interfaces*, pages 1–10, Online, July 2020. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.nli-1.1>. URL <https://aclanthology.org/2020.nli-1.1>.
- [18] Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. A joint model for question answering over multiple knowledge bases. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. <https://doi.org/10.1609/aaai.v30i1.10381>. URL <https://ojs.aaai.org/index.php/AAAI/article/view/10381>.
- [19] Sumit Neelam, Udit Sharma, Hima Karanam, Shajith Ikbal, Pavan Kapanipathi, Ibrahim Abdelaziz, Nandana Mihindukulasooriya, Young-Suk Lee, Santosh Srivastava, Cezar Pendus, Saswati Dana, Dinesh Garg, Achille Fokoue, G P Shrivatsa Bhargav, Dinesh Khandelwal, Srinivas Ravishankar, Sairam Gurajada, Maria Chang, Rosario Uceda-Sosa, Salim Roukos, Alexander Gray, Guilherme Lima, Ryan Riegel, Francois Luus, and L V Subramaniam. SYGMA: A system for generalizable and modular question answering over knowledge bases. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3866–3879, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.findings-emnlp.284>. URL <https://aclanthology.org/2022.findings-emnlp.284>.
- [20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL <http://jmlr.org/papers/v21/20-074.html>.
- [21] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30:3–26, 2007. URL <https://api.semanticscholar.org/CorpusID:8310135>.
- [22] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.
- [23] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=5k8F6UU39V>.
- [24] Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. Lc-quad 2.0: A large dataset for complex question answering

- over wikidata and dbpedia. In *The Semantic Web – ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II*, page 69–78, Berlin, Heidelberg, 2019. Springer-Verlag. ISBN 978-3-030-30795-0.
- [25] Ricardo Usbeck, Xi Yan, Aleksandr Perevalov, Longquan Jiang, Julius Schulz, Angelie Kraft, Cedric Möller, Junbo Huang, Jan Reineke, Axel-Cyrille Ngonga Ngomo, Muhammad Saleem, and Andreas Both. QALD-10 — The 10th Challenge on Question Answering over Linked Data. *Under review in the Semantic Web Journal*, 02 2023. URL <https://www.semantic-web-journal.net/system/files/swj3357.pdf>.
- [26] Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488. ACM, 2021.
- [27] Yucheng Zhou, Xiubo Geng, Tao Shen, Wenqiang Zhang, and Daxin Jiang. Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5822–5834, Online, June 2021. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.naacl-main.465>. URL <https://aclanthology.org/2021.naacl-main.465>.
- [28] Ricardo Usbeck, Michael Röder, Michael Hoffmann, Felix Conrads, Jonathan Huthmann, Axel-Cyrille Ngonga Ngomo, Christian Demmler, and Christina Unger. Benchmarking Question Answering Systems. *Semantic Web*, 10(2):293–304, 2019. <https://doi.org/10.3233/SW-180312>. URL <http://www.semantic-web-journal.net/system/files/swj1578.pdf>.
- [29] Ricardo Usbeck, Michael Röder, Michael Hoffmann, Felix Conrads, Jonathan Huthmann, Axel-Cyrille Ngonga Ngomo, Christian Demmler, and Christina Unger. Benchmarking Question Answering Systems. *Semantic Web*, 10(2):293–304, 2019. <https://doi.org/10.3233/SW-180312>. URL <http://www.semantic-web-journal.net/system/files/swj1578.pdf>.
- [30] K. Shivashankar, K. Benmaarouf, and N. Steinmetz. From graph to graph: Amr to sparql. In *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022)*, 2022.
- [31] N. Baramiia, A. Rogulina, S. Petrakov, V. Kornilov, and A. Razzhigaev. Ranking approach to monolingual question answering over knowledge graphs. In *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022)*, 2022.
- [32] Yiheng Shu and Zhiwei Yu. Distribution shifts are bottlenecks: Extensive evaluation for grounding language models to knowledge bases. In Neele Falk, Sara Papi, and Mike Zhang, editors, *Proceedings of the 18th Conference of*

the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 71–88, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.eacl-srw.7>.

- [33] Yu Gu and Yu Su. ArcaneQA: Dynamic program induction and contextualized encoding for knowledge base question answering. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1718–1731, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL <https://aclanthology.org/2022.coling-1.148>.