# Marie and BERT - A Knowledge Graph Embedding based Question Answering System for Chemistry

Xiaochi Zhou<sup>1</sup>, Shaocong Zhang<sup>2</sup>, Mehal Agarwal<sup>2</sup>, Jethro Akroyd<sup>1,2,4</sup>, Sebastian Mosbach<sup>1,2,4</sup>, Markus Kraft<sup>1,2,3,4,5</sup>

released: May 19, 2023

 Department of Chemical Engineering and Biotechnology University of Cambridge Philippa Fawcett Drive Cambridge, CB3 0AS United Kingdom

 <sup>3</sup> School of Chemical and Biomedical Engineering Nanyang Technological University 62 Nanyang Drive Singapore, 637459

> <sup>5</sup> The Alan Turing Institute London United Kingdom

<sup>2</sup> CARES Cambridge Centre for Advanced Research and Education in Singapore 1 Create Way CREATE Tower, #05-05 Singapore, 138602

<sup>4</sup> CMCL Innovations Sheraton House Cambridge CB3 0AX United Kingdom

Preprint No. 307



Keywords: KGQA, KG Embedding, BERT

## Edited by

Computational Modelling Group Department of Chemical Engineering and Biotechnology University of Cambridge Philippa Fawcett Drive Cambridge, CB3 0AS United Kingdom

E-Mail: mk306@cam.ac.uk World Wide Web: https://como.ceb.cam.ac.uk/



#### Abstract

This paper presents a novel Knowledge Graph Question Answering (KGQA) system for chemistry implemented on hybrid knowledge graph embeddings. Unlike other existing designs, the system operates on multiple embedding spaces, which use various embedding methods, and queries the embedding spaces in parallel. With the answers returned from multiple embedding spaces, the system leverages a score alignment model to adjust the answer scores and re-rank the answers. Further, the system implements an algorithm to derive implicit multi-hop relations to handle the complexities of deep ontologies and improve multi-hop question answering. The system also implements a BERT-based bi-directional entity-linking model to enhance the robustness and accuracy of the entity-linking module. The system uses a joint numerical embedding model to efficiently handle numerical filtering questions. Further, it is capable of invoking semantic agents to perform dynamic calculations autonomously. Finally, the KGQA system handles numerous chemical reaction mechanisms using semantic parsing supported by a Linked Data Fragment server. This paper evaluates the accuracy of each module within the KGQA system with a chemistry question dataset.



#### **Highlights**

- A novel design of a QA system that operates on top of multiple embedding spaces which utilize different embedding methods.
- An algorithm to efficiently derive implicit multi-hop relations within deep ontologies.
- A novel embedding method combining TransR model with joint numerical embedding.

# Contents

1	Intr	oduction	4
2	Rela	ited work	5
	2.1	The World Avatar Knowledge Graph	5
	2.2	Chemistry Ontologies	6
		2.2.1 TWA Chemistry Ontologies	6
		2.2.2 Wikidata chemistry	7
	2.3	Knowledge Graph Question Answering System	8
		2.3.1 Semantic Parsing	8
		2.3.2 Information Retrieval	9
	2.4	Entity linking	9
	2.5	Knowledge Graph Embeddings for KGQA	10
		2.5.1 TransE	11
		2.5.2 TransEA	11
		2.5.3 ComplEx	12
		2.5.4 TransR	12
3	Data	asets	12
	3.1	Entity Linking Dataset	12
	3.2	Relation Prediction Dataset	13
	3.3	Score Alignment Dataset	13
4	Desi	gn	14
	4.1	Overview	14
	4.2	Entity Extraction	14
		4.2.1 SMILES Translation	15
		4.2.2 Joint ER and EL	17
	4.3	Hybrid Knowledge Graph Embeddings	17
		4.3.1 Implicit Multi-hop Relation Derivation	18
5	Imp	lementation/Method	18
	5.1	QA Engine Workflow	18
	5.2	Information Derivation	19

	5.3	Numerical Questions	19
	5.4	OntoKin Chemical Reactions	21
	5.5	Semantic Agents for Dynamic Calculations	22
	5.6	Score Alignment Model	24
6	Eva	luation	24
	6.1	Entity linking	24
	6.2	QA engine	25
	6.3	Language Models	27
7	Rest	ılts	27
8	Con	clusion	28
No	omeno	lature	32
	Refe	erences	42

# **1** Introduction

In the rapidly evolving digital world, the chemistry sector is producing an enormous amount of data, which is becoming increasingly complex to handle. The traditional methods of information storage and retrieval are inadequate to manage this vast quantity of information. In this context, Knowledge Graphs (KGs) provide a flexible and powerful framework for semantic retrieval, enabling efficient storage and retrieval of complex and interconnected data. KGs can also uncover new relationships between different entities in the data. By conducting a deep exploration of the chemical space using knowledge graphs, researchers can effectively navigate the vast and complex chemical space to uncover previously unknown relationships between chemical entities, further expanding their understanding of the chemical space and its potential applications. However, KGs are usually very large and not easily accessible for users as they need to know a query language as well as the structure and relations in the KG.

Knowledge Graph Question Answering (KGQA) [11, 44, 48, 64] systems bridge human users to the complex knowledge within KGs through natural language queries by providing a natural language interface for querying KGs and accessing relevant information. This allows users to perform more complex and nuanced searches than would be possible using traditional search methods. Thus, the application of a deep exploration of the chemical space using KGQA systems has the potential to revolutionize the information discovery process and pave the way for more efficient and effective utilization of chemical data across a wide range of fields.

A Knowledge Graph (KG) is a form of data representation, that consists of collections of descriptions of entities: events, concepts, or objects in the physical world, where the entities are interconnected with each other via relations. It is a directed graph with the entities as nodes and their relations as edges. Each directed edge in this graph, along with its head and tail entities, constitutes a triple, i.e., (head entity, predicate, tail entity). The description of entities has formal semantics provided by ontologies. As a result, machines can process information in KGs in an unambiguous manner. Major existing KGs include the Wikidata KG [51], the DBpedia KG[3], and the Google KG [18].

KGQA systems have emerged as a prominent research topic in the KG community and attract massive attention. In particular, KGQA in the chemistry domain is a promising area of research owing to the rapid growth of chemistry-related KGs and the potential advantages of a deep search of the chemical space. There are various implementation methods of KGQA systems. One such rule-based method is Semantic Parsing (SP) which transforms natural language queries into machine-readable logical forms that can be processed by KGs. In the past, chemistry KGQA systems such as Marie [66] utilized the templatebased SP method to interpret questions and generate SPARQL queries by filling query templates with Internationalized Resource Identifiers (IRIs) for question-answering. The 2 key advantages of such template-based SP methods lie in its lower reliance on training data, allowing the system to be implemented quickly with limited training material and the higher accuracy of answers.

However, SP-based KGQA systems are faced with several limitations due to the increasing complexity of constructing precise queries making them prone to errors. The chemistry domain consists of non-shallow ontologies such as PubChem, OntoSpecies, necessitating multiple steps in a SPARQL query to retrieve the correct answer, contributing to the increasing difficulty of the QA system to construct the appropriate query. The nature of formal representations like SPARQL queries make them vulnerable to semantic or syntactic errors, which can lead to inaccurate results or query failure. Furthermore, the chemistry domain has multiple ontologies with different schemas, which creates scalability and robustness issues for SP-based KGQA systems which are incapable of handling the heterogeneity of such KGs. Hence, there is a need for a KGQA system that can overcome the aforementioned challenges.

In order to bridge this gap, this paper explores, Information Retrieval (IR)-based KGQA systems [24, 38, 57] which have become popular due to their ability to generate limited answer candidates and rank them in accordance with the question, thereby overcoming the constraints of template-based SP approaches.

The purpose of this paper is to present Marie and BERT, an IR-based KGQA system designed to facilitate an extensive exploration of the chemical space. The proposed design of Marie and BERT aims to handle heterogeneous multi-ontology KGs and alleviate the costs of creating SPARQL templates by using hybrid KG embeddings. It aims to recognize entity references within natural language questions through a BERT-based Entity Linking (EL) model. The system also proposes the application of a predicate prediction model to ascertain the relationship within the question. Furthermore, the system is designed to include a score alignment model to handle multiple answer candidates from various Chemistry KGs. In order to address the issues posed by complex non-shallow ontologies, this paper proposes an implicit multi-hop relation derivation mechanism to create triples for training KG embedding models. Further, Marie and BERT aims to learn and infer information from the existing knowledge in the KG. It aims to invoke semantic agents that operate over TWA KG to perform dynamic calculations. Finally, the system intends to handle a vast number of chemical reaction mechanisms utilizing a combination of SP with Linked Data Fragment when addressing chemical reaction-related questions.

# 2 Related work

### 2.1 The World Avatar Knowledge Graph

The modern world is composed of intricate and complex systems such as industrial symbioses, chemical plants, and cities. These systems are made up of diverse components such as power generators, storage tanks, and abstract industrial operations. However, integrating the relevant data, knowledge, and models from these components to achieve complex tasks such as running simulations and optimizations, and coordinating multiple components, poses a challenge due to communication friction resulting from the use of heterogeneous conventions across domains.

TWA project is aimed at creating a comprehensive virtual representation of the physical world, with the goal of facilitating seamless integration and interoperability across diverse domains. The project builds on the concept of Digital Twins, which involves creating

virtual representations of entities in industrial processes and takes it to the next level by extending it to cover all aspects of the physical world. The aim is to enable uniform integration between not only devices, but also between devices and operations, thereby upgrading the Internet of Things to the Internet of Services.

Through the J-Park Simulator (JPS) [15], one specific implementation of TWA project, a data management common ground for these components has been provided to enable semantic interoperability. The JPS is now fully integrated into TWA KG, which is a large-scale, dynamic KG that integrates multiple ontologies from different domains. By using Semantic Web technology, information is represented in a machine-readable way, where concepts, entities, and the relations between them are formally defined and connected. This connection enables the retrieval and navigation of related data within a KG, and interconnects previously isolated datasets by linking knowledge from different domains.

To update and maintain the large-scale KG over time, a number of agents are part of TWA KG, which perform functions like data retrieval, simulation, and data update on top of the knowledge layer. In the TWA KG, agents are crucial for its dynamic functionality. These agents are web services deployed in a distributed manner and can be accessed through HTTP requests. Their semantic descriptions are stored in the KG, and they are semantically described by OntoAgent [65]. This allows for the implementation of an agent composition framework, which enables automated discovery, composition, and invocation of agents. The TWA KG hosts a wide range of agents, including the Thermodynamic Data agent (STDC agent) and the Power Conversion Efficiency agent (PCE agent), which are specific to the chemistry domain.

## 2.2 Chemistry Ontologies

Marie and BERT is a KGQA system developed for chemistry, which operates on top of the chemistry ontologies in TWA KG and the Wikidata KG.

#### 2.2.1 TWA Chemistry Ontologies

TWA KG is a large-scale, cross-domain, and dynamic knowledge graph (dKG) that follows linked data principles and integrates ontologies from various domains including chemistry. TWA integrated and interconnected a number of ontologies specifically developed for representing chemical data such as OntoKin [16], OntoCompChem [29], OntoSpecies [37], and OntoMOPs [27] ontologies.

OntoKin is an ontology that represents chemical kinetic reaction mechanisms. These mechanisms involve the set of stochiometric reactions among different chemical species, described through various thermodynamic and transport model concepts and identified by OntoSpecies IRIs [16]. It employs Description Logic (DL) to provide a semantic representation of chemical data within reaction mechanisms, which offers various benefits, such as interoperability between chemical kinetic systems, automated comprehension of chemical mechanisms by agents, and the capability to perform complex semantic queries on mechanisms in a web environment [17]. OntoKin includes specific details about where the reaction takes place, such as in gas or on a surface, and offers a range of common re-

action rate models for gas-phase and surface reactions. Additionally, OntoKin models the reversibility of a reaction using the reaction order, and it allows for easy detection of inconsistencies in thermodynamic, transport, and reaction data across mechanisms.

OntoCompChem is an ontology designed to enhance the semantics of chemical data in computational chemistry calculations, with a focus on density functional theory (DFT) for molecular systems [28]. It extends the Gainesville Core and CompChem ontologies and employs DL-based semantics to enable interoperability between quantum chemistry software, reduce computational resource consumption via calculation reuse, and aid automated agents in understanding such calculations. OntoCompChem represents various aspects of a calculation, including its objective, software, theoretical level, charge, spin polarization, and calculated frontier orbitals and self-consistent field energy. It also stores optimized geometries and computed vibrational frequencies for geometry optimizations and frequency calculations, respectively, linking them back to their corresponding calculations.

The OntoMOPs ontology reflects concepts and relationships relevant for the rational design of metal-organic polyhedra (MOPs) [27]. These concepts involve chemical and spatial factors that are used in chemical and spatial evidence based reasoning respectively. MOPs are assemblies made of organic and metal-based chemical building units (CBUs) that resemble regular polyhedra. To facilitate the design of new MOPs, OntoMOPs encodes assembly models (AMs) and generic building units (GBUs) as mental blueprints that guide the selection of CBUs from available sets. Additionally, OntoMOPs employs the OntoSpecies ontology to instantiate the CBUs as species.

OntoSpecies [37] is a fundamental chemistry ontology in TWA KG. This ontology contains the IRIs of about 36,000 and is constantly growing. The ontology also covers the basic chemical and physical properties of species. It is used as a foundation and has been expanded to encompass a diverse collection of identifiers, classifications and uses of chemical species, as well as spectral data, in addition to information indicating its origins and attribution.

#### 2.2.2 Wikidata chemistry

The Wikidata chemistry ontology is a subset of the Wikidata KG [51, 53], focused exclusively on chemical species. This ontology was created by generating a customized dump that includes only instances classified under the sub-classes of the class "group or class of chemical substances". The Wikidata chemistry ontology currently comprises 33,061 distinct chemical species and approximately 50,000 triples.

The Wikidata chemistry ontology primarily captures chemical and physical properties of the included species, such as "mass", "chemical structure", "chemical formula", "refractive index", "ionization energy", and "autoignition temperature". In addition, it also includes the identifiers of these species in other databases, such as PubChem CID, Chem-Spider ID, and CAS Registry Number. This feature allows for easy linking and integration of Wikidata chemical data with external databases, facilitating cross-database searches and interoperability.

#### 2.3 Knowledge Graph Question Answering System

Knowledge Graph Question Answering (KGQA) [11] is a major focus in both the KG community and the Natural Language Processing (NLP) community, as KGs are rich sources of semantic and structured data, and question answering represents one of the biggest challenges for NLP. In general, KGQA systems answer questions using two methods [60]: Semantic Parsing-based (SP-based) method [48, 64] and Information Retrieval-based (IR-based) method [24, 57].

#### 2.3.1 Semantic Parsing

In the SP-based method, questions are parsed into formal representations such as SPARQL queries [39],  $\lambda$ -DCS [33], or FunQL [26]. These formal representations are then executed against the KG to find answers. There are three different approaches for SP-based methods: ranking methods, coarse-to-fine methods, and generation methods.

**Ranking method:** In the ranking method, the QA system first generates a list of candidate formal representations through a process known as candidate enumeration. For example, Abujabal et al., Bast and Haussmann, Berant and Liang create candidate queries by filling pre-specified query or machine-generated query templates with IRIs and other specific arguments. Another approach for candidate enumeration is to traverse the relation paths and neighbours linked to the topic entity in the question, and add query operations including constraints or aggregation functions to form candidate queries.

Once the candidate queries are formed, the ranking method employs semantic matching techniques to select the top-ranked candidates. Neural models such as CNN [62], LSTM [63], and Pre-trained Language Models (PLM) [20, 61] are commonly used to score question-query pairs.

**Coarse-to-fine method:** The coarse-to-fine method comprises two steps. First, the QA system predicts a rough skeleton that focuses only on the high-level structure of the query. Recent works often predict rough queries using encoder-decoder models. For example, Ravishankar et al. utilized a transformer-based SEQ2SEQ model initialized with BERT to generate the SPARQL skeleton corresponding to the question text. Sun et al. use a pipeline of sub-tasks including question split and span prediction for skeleton parsing. Das et al. use the pretrained T5 encoder-decoder model to directly produce a coarse skeleton.

With the rough skeleton in place, the QA system then populates it with details and creates the final queries. For example, Ding et al. use an attention-based BiLSTM network to link possible entities and relations in the questions, and come up with all combinations of entities and relations for each skeleton. Hu et al. use BERT as a binary classifier to map attributes in the question to semantic relations.

**Generation method:** The generation method features two paradigms: graph search and encoder-decoder. In the works that leverage graph search, formal representations are created by traversing the graph from the topic entity in the question. For example, Lan et al. iteratively searches through reachable relation paths starting from the topic entity and ranks the relation paths in the context of the question. The encoder-decoder paradigm aims to directly translate the questions to their formal representations. Numer-

ous works [8, 10, 19, 40] experiment with various encoder-decoder models to improve the correctness and well-formedness.

#### 2.3.2 Information Retrieval

The IR-based method, given a question, extracts a question-specific subgraph, which ideally includes all entities and relations related to the question, and applies a ranking algorithm to find the answer within the subgraph.

One of the most prevalent IR-based KGQA methods is the KG embedding-based method. First, all entities and relations are represented within a vector space, where their semantic relations are preserved. With the KG embedding, the likelihood or relative likelihood of a triple (h, r, t) can be measured by a scoring function  $\phi(\hat{e}_h, \hat{e}_r, \hat{e}_t)$ . In a KG embeddingbased KGQA system, the system translates the question into an embedding vector which serves as the relation embedding  $\hat{e}_r$ . Meanwhile, the system typically extracts an entity as the head entity and looks up its embedding  $\hat{e}_h$ . In addition, all the neighbours within an n-hop distance from the head entity are extracted as candidate answers  $A = \{a_1, ..., a_n\}$ . To rank all the candidates, for each  $a' \in A$ , the likelihood of the hypothetical tripe,  $\phi(h, r, a')$ is calculated.

Huang et al. first proposed the KG embedding-based method which leverages KG embedding to answer simple questions. BiLSTM is used to convert the question into a vector representing the predicate, while TransE and TransR are applied to embed the KG. Saxena et al. then improved the KG embedding-based method by using the RoBERTa model for question embedding and applying the Complex embedding method for KG embedding. Shang et al. further improved the method for answering time-sensitive questions over temporal KGs.

The KG embedding-based method has been applied in several domain-specific KGQA systems. For instance, in the biomedical field, Rao et al. proposed a KGQA system over the Hetionet dataset, which leverages RoBERTa and BioBERT for question embedding and applies Complex embedding for KG embedding.

## 2.4 Entity linking

Entity Linking (EL) is a crucial component of KGQA systems as it bridges the gap between human language and KG representation. EL involves identifying entity references, called 'mentions', in unstructured text and mapping them to corresponding entities in the KG. Traditional EL approaches use a pipeline structure consisting of three sub-tasks [2]: Named Entity Recognition (NER) for identifying the mention boundaries in text, Candidate Generation for generating pre-filtered candidate lists for mentions, and Entity Ranking for ranking the candidates based on certain criteria.

Recent studies have demonstrated that PLMs like BERT [13] significantly outperform traditional rule-based systems in EL [45]. Wu et al. [55] use BERT-based encoders in a two-stage fashion for EL, achieving the state-of-the-art (SOTA) performance in local settings, while Yamada et al. [58] consider global contextual information and achieve the

highest SOTA performance in global settings. These studies highlight the advantages of using pre-trained BERT for EL. However, most of these studies evaluate their approaches exclusively on general-topic KGs such as DBpedia [3] or Freebase [6], limiting their application to laboratory setups instead of actual KGs in operation. Further research is needed to transfer these methodologies to specific scientific disciplines, such as chemistry, or specific downstream applications, such as QA.

EL for QA requires a specific approach since it involves short questions that have less context information compared to general EL that focuses on lengthy and properly formed documents like Wikipedia pages or news articles. Li et al. [32] leveraged BERT-embedding for EL and designed it specifically for QA scenarios. However, their evaluation is based on DBpedia entities, limiting their study's applicability to scientific domains. The vocabulary sets and domain-related characteristics of scientific domains present additional challenges that need to be addressed.

The biomedical domain has presented some experience in EL, with the community developing BERT models based on domain-specific corpus, such as BioBERT [31] and ClinicalBERT [47] in addition to the original BERT models trained on Wikipedia and BookCorpus [59]. Ji et al. [25] fine-tune pre-trained BERT models for the sentence pair classification task and achieve the SOTA performance, demonstrating the advantages of using domain-adapted BERT models over the original BERT model. Chen et al. [9] proposed a simpler model with a specific feature suited for biomedical terms. However, these approaches primarily focus on disease terms and clinical records, which are not directly applicable to the chemical QA scenario.

## 2.5 Knowledge Graph Embeddings for KGQA

KGs are widely used to represent structured information in various domains. One of the key challenges in working with KGs is to effectively manipulate and reason with the large amount of data they contain. KG embedding is a popular technique that can address this challenge by representing entities and relations in a continuous low-dimensional vector space. This technique is widely used in tasks such as KG inference, relation prediction, KG completion, and in supporting KGQA.

A KG is represented as a set of triples  $\mathscr{G} = \{(h, r, t)\}$ , where each triple contains a head entity *h*, a tail entity *t*, and a relation *r* between them. The sets of entities and relations are denoted as *N* and *M*, respectively. To learn the embeddings, a loss function L(h, r, t)is defined and the embeddings are iteratively updated to minimize the loss.

Over the years, a wide variety of KG embedding algorithms with unique loss functions have been developed, each capable of capturing different features of the KG. Choosing the appropriate embedding algorithm is crucial for achieving optimal efficiency and accuracy in embedding the KG, as the scale and structure of the KG can vary widely across domains.

#### 2.5.1 TransE

TransE [7] is one of the simplest and most effective embedding methods for KGs that have only 1-to-1 relations. For a triple  $(h, r, t) \in \mathcal{G}$ , it models the process of predicting the tail entity, *t* given the head entity, *h* and the relation, *r* by translating *h* to *t* via *r*.

Given a triple  $(h, r, t) \in \mathscr{G}$ , the TransE embedding in  $\mathbb{R}^d$  results in three *d*-dimensional vectors denoted as  $\hat{e}_h, \hat{r}, \hat{e}_t \in \mathbb{R}^{1 \times d}$ . The TransE embedding is constructed such that  $\hat{e}_h + \hat{r} \approx \hat{e}_t$  if (h, r, t) holds. To achieve this, the TransE score function  $\phi(\hat{e}_h, \hat{r}, \hat{e}_t)$  is introduced, which measures the distance between the sum of the vectors of the embedded head and the embedded relation and the embedded tail entity, defined as:

$$\phi(\hat{e}_h, \hat{r}, \hat{e}_t) = ||\hat{e}_h + \hat{r} - \hat{e}_t||_{1/2}.$$
(1)

The training of TransE embedding is done by a pairwise method, which creates a fake triple  $(h', r, t') \notin \mathscr{G}$  for each  $(h, r, t) \in \mathscr{G}$  by replacing the head or tail entity with a random head or tail. The loss function, *L* is a margin ranking function, which takes the embeddings of both the true and fake triples,  $(\hat{e}_h, \hat{r}, \hat{e}_t')$  and  $(\hat{e}_h, \hat{r}, \hat{e}_t)$  as inputs and calculates the difference between the scores of the triple pair with a margin  $\gamma$ , defined as:

$$L(\hat{e},\hat{r}) = \max(0,\gamma + \phi(\hat{e}_{h},\hat{r},\hat{e}_{t}) - \phi(\hat{e}_{h}',\hat{r},\hat{e}_{t}')).$$
(2)

The training algorithm updates the embedding of all entities  $\hat{e} \in E_e$  and  $\hat{r} \in E_r$  with respect to the gradient of  $L(\hat{e}, \hat{r})$  iteratively to minimize  $L(\hat{e}, \hat{r})$ .

#### 2.5.2 TransEA

TransEA [56] is an extension of the TransE embedding method that can handle numerical literals in addition to entity triples,  $(h, r, t) \in \mathcal{G}$ . Some knowledge graphs, such as Wikidata, contain both entity triples and attribute triples,  $(h, l, v) \in \mathcal{G}$ , where *l* is the attribute and *v* is a literal value associated with the entity. In TransE, attribute triples are ignored.

In the TransEA method, a loss function for the embedding of numerical literals,  $L_l(\hat{e}, \hat{l}, \hat{b})$  is added to the TransE loss function for entity triples,  $L_r(\hat{e}, \hat{r})$ . For the embedding of a triple with a numerical literal  $(\hat{e}_h, \hat{l}, v)$ , the loss function is defined as:

$$L_{l}(\hat{e}_{h},\hat{l},\hat{b}_{l}) = ||\hat{l}^{\top} \cdot \hat{e}_{h} + \hat{b}_{l} - v||_{1/2}$$
(3)

where  $\hat{l} \in \mathbb{R}^{1 \times d}$  is the embedding of the attribute  $l, \hat{b}_l \in \mathbb{R}^1$  is the bias for the attribute  $\hat{l}$ , and  $v \in \mathbb{R}$  is the numerical value of the entity attribute.

The TransEA loss function  $L_r(\hat{e}, \hat{r})$  for an entity triple (h, r, t) is identical to that of TransE. To model both entity triples and attribute triples, TransEA sums the two loss functions with an adjustable factor  $\alpha$ . The combined loss function  $L(\hat{e}, \hat{r}, \hat{l}, \hat{b}_l)$  is defined as:

$$L(\hat{e},\hat{r},\hat{l},\hat{b}_l) = (1-\alpha) \cdot L_r(\hat{e},\hat{r}) + \alpha \cdot L_l(\hat{e},\hat{l},\hat{b}_l).$$

$$\tag{4}$$

#### 2.5.3 ComplEx

ComplEx [50] is a tensor factorization approach that represents the entities and relations in the KG in the complex space as complex-valued vectors, which enables the use of complex algebra to model semantic relationships among them.

For each triple  $(h, r, t) \in \mathcal{G}$ , ComplEx generates  $(e_h, e_r, e_t) \in \mathbb{C}$  and defines a scoring function:

$$\phi(h, r, t) = Re(\langle e_h, e_r, \overline{e_t} \rangle)$$
  
=  $Re(\sum_{k=1}^d e_h^k e_r^k \overline{e}_t^k)$  (5)

such that  $\phi(h, r, t) > 0$  for all true triples, and  $\phi(h, r, t) < 0$  for false triples. The real and imaginary parts of the embeddings capture different aspects of the semantics of the entities and relations, respectively, and their combination in the scoring function allows for the modeling of more complex and nuanced relationships among them.

#### 2.5.4 TransR

TransR [34] is an extension of TransH [54], which encodes entities and relations into distinct spaces. For each triple (h, r, t), entities embeddings are  $h, t \in \mathbb{R}^k$  and relation embeddings are  $r \in \mathbb{R}^d$ . A projection matrix  $M_r \in \mathbb{R}^{k \times d}$  is learned for each relation r, that can project an entity to different relationship semantic spaces [34]. Each of these spaces captures a different aspect of an entity that is related to a distinct relationship. The projected vectors of entities are denoted as:

$$h_r = hM_r \tag{6}$$

$$t_r = tM_r \tag{7}$$

The loss function is defined as:

$$f_r(h,t) = ||h_r + r - t_r||_2^2$$
(8)

# **3** Datasets

#### **3.1 Entity Linking Dataset**

To train an effective EL model, a high-quality EL dataset is required, which consists of text labelled with mentions and their corresponding IRIs. The EL dataset is designed to support the training of EL models that can recognize mentions of these entities in questions and link them to their corresponding IRIs in the KG. The EL dataset focuses

on several domains, including chemical formula, name, SMILES, and InChI, as well as various classes such as MOPs and CBUs.

The data in the EL dataset is collected from Wikipedia pages related to chemistry and chemical species. These pages contain not only textual information about the chemical species but also their Wiki IDs. By utilizing the information provided by Wikidata, we map these Wiki IDs to their corresponding PubChem Compound Identifiers (CIDs).

For data not available on Wikipedia, such as classes like MOPs and CBUs from a specific domain ontology, we employ scripts to automatically generate training data from the ontology. These scripts traverse the domain ontology, gathering IRIs and labels. Subsequently, the scripts generate text with labeled mentions by applying pre-defined templates and rules.

The input to the dataset is text labelled with mentions, and the output is the IRI of the mention.

## **3.2 Relation Prediction Dataset**

The relation prediction dataset is designed to train a model for predicting relation embeddings given a question. The purpose of this dataset is to enable the development of models that can accurately predict relation embeddings between entities in a KG based on the input question.

The input to the dataset is a question, and the output is the relation embedding. The dataset covers both explicit and implicit relations within all chemistry ontologies in this project's scope, providing a comprehensive range of relation types to support the development of relation prediction models.

The data for this dataset is also generated using scripts. These scripts traverse the KG and gather both explicit and implicit relations along with their labels. Pre-defined question templates are then populated with the relation labels to generate questions. For instance, a question template might be "How much does it weigh?" where the entity is omitted. Implicit relations are pre-labeled manually before the data generation process.

#### **3.3** Score Alignment Dataset

The score alignment dataset is designed to train a model for re-ranking the answers returned by different domain Question Answer (QA) engines based on the input question and the ontologies with which it is affiliated. The purpose of this dataset is to enable the development of models that can accurately align scores across different QA engines and improve the accuracy of question answering.

The input to the dataset is a question and a list of ontologies with which the question is affiliated. The output is a re-ranked list of answers returned by different domain QA engines, based on the alignment of their scores. The dataset covers possible questions from all chemistry ontologies within this project's scope, providing a comprehensive range of question types to support the development of the score alignment model.

The training dataset is created using a method similar to that employed for the relation prediction dataset.

# 4 Design

#### 4.1 Overview

The Cross Graph Question Answering (CGQA) engine, as depicted in 1, serves as the primary interface for the QA engines developed for each ontology within the Marie and BERT system. The CGQA engine incorporates seven distinct QA engines, with each one designed for a specific chemistry ontology in the TWA KG. Each QA engine operates on the KG embeddings associated with its particular ontology.

When the CGQA engine receives a natural language question, it distributes the query to all QA engines which operate in parallel. In response, every QA engine generates a collection of answer candidates and ranks them using a score function. This function assesses the likelihood of each candidate answer's existence.

Subsequently, the CGQA re-ranks the answers provided by the individual QA engines and presents a final set of the most suitable answers within its context.

## 4.2 Entity Extraction

It is crucial to first obtain the IRI of the head entity referred to in the input question before a QA engine can predict the answers. For a KG  $\mathscr{G}$ , a set of entities  $\mathscr{E}$  and question text q, the Entity Extraction module aims to find the corresponding IRI h of the head entity  $e \in \mathscr{E}$ referred to in q. The occurrence of an entity in a natural language sentence is referred to as a 'mention'. The conventional workflow involves first identifying a mention span in the question and then scoring entities based on their similarity to the mention span text. This process is divided into two steps: Entity Recognition (ER) followed by Entity Linking (EL). However, we propose a different workflow for three main reasons:

- 1. Mention boundaries of entities are irrelevant in Question Answering.
- 2. Scientific notations like SMILES differ significantly from natural language, making them difficult to process with a single unified language model.
- 3. A stand-alone ER system infers solely from the question text and does not utilize the information in the KG. Once the span is marked, additional information in the question text is discarded and not utilized in the follow-up EL. Moreover, the question text being short and lacking context, ER as a standalone step might be error-prone and become the bottleneck.

Instead, we implement a joint ER and EL approach similar to [32]. Additionally, we include a SMILES translation module to extract any SMILES names and convert them to



**Figure 1:** The overall design of the Marie QA system: for each ontology, an independent set of entity embeddings is trained, where  $E_e$  are indexed set of entity embeddings and  $\mathcal{E}$  is the indexed set of entity IRIs, and an independent QA engine is implemented. Given a question q, each QA engine will return a set of answers together with their scores S and the encoded label L of the corresponding embedding space  $E_e$ . Given the scores S and the embedding space labels, and the question q, the score alignment model will adjust the scores with respect to the inputs and re-rank the answers returned from different embedding spaces and QA engines.

IUPAC names, which are more accessible for a language model to make predictions. We propose the following two-stage workflow, as shown in 2:

#### 4.2.1 SMILES Translation

We recognize SMILES names as unique surface forms in the chemical domain that are difficult to process with a unified language model. Consequently, we design a SMILES to IUPAC name translation module as a pre-processing step. For this purpose, we fine-tuned a *BERT*<sub>BASE</sub> model  $T_{SMILES}$  along with an additional linear classifier *C* on the token classification task, following standard practices. Each token in the question text is classified as



**Figure 2:** The workflow of the Entity Extraction module is completed in two stages, Left: the SMILES-Translation stage: if a SMILES substring exists in the question q, it is identified and translated through an off-the-shelf SMILES to IUPAC translation module. The question is then preprocessed by replacing the SMILES string with its IUPAC translation. Right: the joint ER and EL stage: tokenlevel embedding of the question text,  $v_q$  is obtained from the question encoder. The mention classifier then outputs the possibility of a mention span [i, j] in the question being the true mention span P([i, j]). Its embedding is the average embeddings of the tokens in the span,  $v_{mention}$ . Entity embeddings  $v_e$  are generated by the Entity Encoder and cached before runtime.  $P(v_e|v_{mention})$  is the probability of an entity being the gold entity given a mention.The mention entity pairs are eventually scored based on the joint probability  $P[i, j]P(v_e|v_{mention})$ .

either belonging to a SMILES string or not. The training procedure is detailed in 9.

Given a question text q, the trained model can now identify SMILES substrings. Once identified, we translate the SMILES string to its common IUPAC name and replace it in the question text before feeding it into the Joint EL and ER model for entity extraction. For translation, we employ the off-the-shelf library STOUT [41].

#### 4.2.2 Joint ER and EL

A joint ER and EL system considers all possible spans of the question text as mention span candidates and measures the joint probability of a span being the mention span and the mention matching a candidate entity in the KG. The design for the model follows the method of [32]. The model consists of three trainable components:

- 1. The question encoder, which encodes the question text.
- 2. The entity encoder, which generates entity embeddings from the KG.
- 3. The mention classifier, which outputs the probability of all mention spans being the true span.

The model is trained using the dataset described in 3.1. We follow the methodology in [32] and train the model in a two-step fashion. First, we use the EL dataset with the given mention spans to train a plain bi-encoder. Then, we freeze the entity encoder to train the new question encoder and the mention classifier for the joint ER and EL task. Details of the training procedure can be found in 7 and 8. After the training is completed, the models are frozen, and the entity embeddings for all KG entities are cached for faster computation.

During runtime, a question is first fed into the question encoder to obtain the question embedding. Using the question embedding, the mention classifier outputs the probability of each mention span being the true span. Given a candidate mention span, the mention embedding is calculated as the average of the textual embeddings of all tokens in the mention span. The probability of each entity being the true entity given this mention span is then calculated by comparing the mention embedding and the question embedding.

The final joint probability of an entity-mention span pair is computed as the product of two probabilities: the probability of the mention span being the true span, and the probability of the entity being the true entity given this mention span. For all entity-mention span pairs, we select the top K pairs with the highest joint probability.

#### 4.3 Hybrid Knowledge Graph Embeddings

The KG embedding module aims to represent entities and relations within a KG in vector space. Several KG embedding methods are available for KGs of varying complexities and structures. Due to the heterogeneous and cross-domain nature of the TWA KG, we implemented a novel system wherein different ontologies are embedded using distinct embedding methods. The candidate embedding methods include TransE, TransEA, TransR, TransRA, and Complex embeddings.

OntoSpecies, Wikidata chemistry, and OntoMOPs contain 1 to N relations, in which one head node can connect to multiple tail nodes via the same relation. The majority of the information contained in these ontologies is numerical rather than semantic.

Consequently, the TransRA embedding method is chosen for its ability to model 1 to N relations, compared to the TransE embedding method [34]. Additionally, experiments

conducted to analyze the performance of different embedding methods on the task of inferring knowledge from OntoSpecies reveal two main findings. First, the addition of joint numerical embedding significantly improves the inference performance on OntoSpecies, which is primarily numerical. Second, the experimental results also demonstrate that joint numerical embedding is significantly less effective when applied to Complex embedding.

As a result, the TransRA embedding method is selected as the embedding method for OntoSpecies, Wikidata chemistry, and OntoMOPs. OntoCompChem and OntoKin, on the other hand, contain less numerical data. However, these two ontologies have non-shallow structures, where the answer nodes can be multiple relations away from the head node. Therefore, Complex embedding is chosen for its capability to capture implicit relations [50].

#### 4.3.1 Implicit Multi-hop Relation Derivation

Ontologies like OntoCompChem are non-shallow and of complicated structures, as illustrated in 3. In these ontologies, a property of a species can be three or more hops away from the species itself. Consequently, the formal representation for querying such KGs can be complex and therefore prone to errors. This complexity also poses challenges for IR-based methods. Although Complex embedding can capture implicit relations, the complicated structure of OntoCompChem limits its capability and accuracy of relation prediction.

To address this issue, we adopt an implicit multi-hop relation derivation algorithm as described in algorithm 6. This approach involves deriving possible implicit relations in advance and creating a limited set of triples with these implicit relations for training the KG embedding models. This method does not require significant effort but greatly enhances the accuracy of relation prediction.

# 5 Implementation/Method

#### 5.1 QA Engine Workflow

Figure 4 illustrates the workflow of a QA engine operating on the embedding space for one KG,  $\mathscr{G}$ , where the set of all entity IRIs is denoted as  $\mathscr{E}$ , the set of all relation IRIs as  $\mathscr{R}$ , the embeddings of entities as  $E_e$ , and the embeddings of relations as  $E_r$ .

Each QA Engine receives a question string, q, as input. The question is passed to the EL module, which extracts the head entity in the question and links it to its IRI, h. The subgraph agent then extracts the candidate answers,  $A' = \{a_1', ..., a_j'\} \subset \mathcal{E}$ , that are within n-hops from h. For each  $a' \in A'$ , we look up its embedding to create  $E_a' = \{e_{a1}^{\prime}, ..., e_{aj}^{\prime}\}$ , and similarly, we look up the embedding  $\hat{e}_h$  of h.

In the meantime, given the question q, the relation prediction model maps the relation in q to a relation embedding,  $\hat{e}_q$ . For each candidate answer  $\hat{e}_a' \in E_a'$ , the embeddings of a hypothetical triple  $(\hat{e}_h, \hat{e}_q, \hat{e}_a')$  are created. All hypothetical triples are then scored using



Figure 3: The illustration of the structure of the OntoCompChem ontology. As shown in this figure, the head entity, which is a species, is 3-hops away from the candidate answers. In addition, the first two hops from the head entity to the candidate answers are identical.

the scoring function  $\phi(\hat{e}_h, \hat{e}_q, \hat{e}_a')$  of the embedding method being used for  $\mathscr{G}$ . A set of candidate answer embeddings  $E_a = \{\hat{e}_{a1}, ..., \hat{e}_{ak}\}$  with the highest scores is formed and converted to their IRI forms  $A = \{a_1, ..., a_k\}$ , where  $k \leq j$ . If  $a \in A$  is linked to numerical values, the numerical values are then retrieved by the lateral lookup agent.

## 5.2 Information Derivation

Inference in KGs refers to the process of deducing new information from the existing knowledge represented in the graph. The use of inference can enhance the accuracy and completeness of the information represented in the graph and can be used for a variety of tasks, such as link prediction, question answering, and recommendation systems.

The information derivation task can take several forms such as tail, relation or head inference. Marie and BERT employs the TransRA embedding methodolody to achieve the inference task on OntoSpecies. The performance of TransRA compared to the other embedding methods is shown in Table 1.

Information derivation offers several applications. One potential application is to infer the use of a particular species represented in OntoSpecies.

#### **5.3 Numerical Questions**

Marie and BERT's CGQA is capable of handling numerical questions in the chemistry domain. For example, they can answer questions like, "Which species have a molecular weight less than 50 gm/mol?" Based on the filtering criteria specified in such questions,



**Figure 4:** The working flow of the QA engine for one embedding space, where the indexed set of entity embedding is denoted as  $E_e$  and the indexed set of entity IRIs  $\mathscr{E}$ : q represents the question string and  $\hat{e}_q$  represents the predicted embedding vector of q, h denotes IRI of the head entity within the question while  $\hat{e}_h$  denotes the embedding of h in the vector space, A' denotes the IRIs of a set of candidate answers while  $\hat{e}_a'$  denotes the embeddings of a' in the vector space. For each  $e_a'$ ,  $\phi(\hat{e}_h, \hat{e}_q, \hat{e}_a')$  is calculated to score the hypothetical triple  $(\hat{e}_h, \hat{e}_q, \hat{e}_a')$ .

such as "larger," "smaller," or "close to," the set of candidate species is narrowed down by predicting their numerical attributes.

To handle numerical questions, the QA engine employs two rounds of filtering operations. In the first round, a set of candidate species is obtained based on their predicted numerical values, which must align with the numerical filter specified in the question. Next, the actual numerical attribute values for this subset of species are determined using their em-

	f_mrr	f_hit_1	f_hit_5	f_hit_10
TransE	0.4055	0.3396	0.4528	0.6038
TransEA	0.4821	0.3396	0.6603	0.7169
TransR	0.3911	0.2452	0.6038	0.7547
TransRA	0.5134	0.3962	0.6226	0.6981
Complex	0.3736	0.1698	0.6603	0.6981
Complex_Numerical	0.233	0.1509	0.3396	0.415

**Table 1:** Performance of different embedding methods on OntoSpecies tail inference task.

beddings. Finally, the second round of filtering produces a final list of answer candidates whose actual numerical attribute values adhere to the numerical filter in the question.

This methodology is time-efficient, which is crucial for effectively answering numerical questions. We evaluated the accuracy of numerical filtering mechanism by evaluating its recall, precision, and F1 score. Two separate tests are conducted, the filtered test is conducted given the true relation and true numerical operators in the question while the unfiltered test is conducted without the ground truth information. Table 3 shows the results of the two tests.

## 5.4 OntoKin Chemical Reactions

The OntoKin Chemical Reactions ontology comprises 89,780 unique reactions, 9,964 unique species, and 679,737 triples. Several experiments were conducted to embed this ontology, however, the embedding faces two major challenges. First, the large size of the ontology results in a high embedding cost. Moreover, upon analysing the structure of the ontology, it becomes evident that all nodes within the ontology are connected, and no subgraph can be separated without breaking connections. Second, the ontology structure is highly imbalanced. Some very common species are connected to a significantly larger number of reactions compared to some rare species. For example, the species Hydrogen is connected to 7,316 reactions, whereas  $C_9H_7$  is only connected to 76 reactions. Due to this imbalance, the embedding becomes less effective [21], which further complicates the embedding process for the OntoKin Chemical Reaction ontology.

To address the aforementioned challenges, we adopted a more cost-effective SP-based approach for querying OntoKin Chemical Reactions. This approach involves the use of the NLTK semantic parsing module to analyse the grammatical dependencies in a question and setting up a Linked Data Fragment (LDF) server to support reaction querying. The semantic parsing module leverages probabilistic dependency grammar to parse the question into a tree structure, illustrating the grammar dependencies among different components within the question. We then identify the reactants and/or products within the question. Figure 5 illustrates an example of the grammatical dependency in a reaction query question.

Nonetheless, due its large scale, hosting the ontology using a traditional gragh database, such as Blazegraph or RDF4J, can be computationally expensive. The LDF [52] approach

offers a solution for querying semantic data. This method enhances the scalability and availability of the query endpoint through partitioning strategies and caching mechanisms. Consequently, we implemented an LDF server to host the OntoKin Chemical Reactions ontology.



**Figure 5:** *Grammatical dependency structure of question "When H2 reacts with OH, what is the final outcome of the reaction".* 

## 5.5 Semantic Agents for Dynamic Calculations

In order to perform calculations on the fly, Marie and BERT accesses the dynamic components of the TWA KG: the agents. Agents are web services that are deployed in a distributed manner and accessible via HTTP requests, with their semantic descriptions stored in the KG. In TWA KG, an agent is semantically described using OntoAgent [65]. The typical OntoAgent description of an agent has been revised and simplified to include a detailed description of its input/output (I/O) signatures, its URL, and associated question templates, as illustrated in Figure 6.

The TWA KG encompasses a wide range of agents. Within the chemistry domain, these agents consist of the Thermodynamic Data agent (STDC agent) and the Power Conversion Efficiency agent (PCE agent). The STDC agent calculates the gas-phase thermodynamic properties of a chemical species as a function of temperature T and pressure P [35], while the PCE agent computes the power conversion efficiency of an organic solar cell, given the SMILES string of the donor molecule of the cell [67].

In the previous Marie QA system, the process of agent discovery relied on SPARQL queries, which are prone to errors. Additionally, using a simple classifier for agent discovery proved inadequate for distinguishing between very similar agents. To address these challenges, we have implemented an agent interface in the Marie and BERT CGQA system that is independent of all QA engines. This agent interface operates on the OntoAgent embeddings for chemistry-domain agents, with concepts in the OntoAgent description being embedded using TransR.

The embeddings of the input/output configurations for each agent are utilized to create matrices representing each semantic agent. Given a question, we employ a BERT-based relation prediction model to predict the relation embedding present in the question. In this approach, a pre-trained BERT model is connected to two fully connected layers, which





transforms the question into a relation embedding. Subsequently, we conduct similarity matching between this relation embedding and all available semantic agents using their respective matrices to determine the question-agent affiliation.

Furthermore, we employ a chemical Entity Linking (EL) model to identify the key components within the question. For instance, in the question "What is the heat capacity of benzene at 100 K?", the term "benzene" is labeled as "species". This labeling is achieved through a fuzzy set search within a pre-defined vocabulary list containing a large number of instance and class labels, including species. The label with the highest similarity to the mention is then looked up in a pre-defined vocabulary dictionary, where labels are mapped to their types. For example, "benzene" is mapped to the type "species".

Qualifiers, including numerical values and their units, are identified using regular expres-

sions. For example, in the aforementioned question, the qualifier is identified as "100 K" and labeled as "temperature". However, in the proof-of-concept implementation, we did not handle unit conversion and assumed that all temperature units are in Kelvin (K) and pressure units are in Pascal (Pa).

If a suitable agent is identified, the key components are passed to an agent invocation interface, which generates the HTTP request, invokes the appropriate agent, and returns the answer to the score alignment model for further processing.

## 5.6 Score Alignment Model

By design, the QA engine for each ontology is implemented separately, utilizing independent embedding spaces. However, there is also overlap between domains. For instance, both Wikidata chemistry and OntoSpecies can answer questions about the basic physical and chemical properties of some species. Consequently, correct answers might come from multiple QA engines. Moreover, since different embedding methods are applied to the ontologies, the scale of the answer scores varies, rendering it impossible to directly compare the scores of answers originating from different QA engines.

To address this issue, we implemented a score alignment model that adjusts the scores from multiple QA engines, using the question and the domains of the scores as inputs. The intuition behind this model is to predict a bias for the score associated with each answer based on the question-ontology affiliation and the domain from which the score is derived. Algorithm 4 outlines the training process for this model.

Table 5 presents the results of an ablation test for the score alignment model. In the absence of the score alignment model, the set of scores from each ontology is normalized by dividing each score by the highest score within the set. Subsequently, the scores from different domains are re-ranked together. The results indicate that the score alignment model effectively improves the accuracy of the final answers.

# 6 Evaluation

## 6.1 Entity linking

Due to the lack of available chemical QA datasets, we collected chemistry-related natural language questions from https://socratic.org/chemistry. However, the original questions exhibited a significant imbalance in the distribution of entities. To address this, we manually selected 172 template questions and generated our training and test sets by replacing the original mentions in the question templates with entity mentions sampled from the KG. These entities were randomly sampled from a combination of the first 5000 Pub-Chem compounds and our chemical ontologies. For PubChem entities, the mention could randomly be either the IUPAC name, chemical formula, or any of its aliases.

To demonstrate our model's ability to understand unseen question structures, we ensured that the training and testing templates were separate and had no overlap. We trained our model on 20,000 questions generated from 32 templates and tested it on 5,000 questions generated from 140 templates. For SMILES questions, we randomly selected 500 SMILES expressions from the PubChem compounds list and generated questions from the 140 test templates. We measured the K - 1 accuracy of our predictions.

We employed a traditional pipeline denoted by ChemData+fuzzysearch as a baseline for evaluation. This pipeline first recognizes possible mentions using ChemDataExtractor[49], a popular toolkit for chemical entity recognition using random fields combining custom dictionaries and rule-based NLP techniques. The extracted mention is then linked by a fuzzy name search, including IUPAC names, chemical formulas, and all known aliases.

To compare the performance of our translation pipeline to training with SMILES expressions directly, we trained a second set of models following the same procedure, except with an additional 4,000 training questions generated from SMILES expressions. In this case, the extra translation module was not employed, and the SMILE expression was passed through and compared by the bi-encoder directly. The K - 1 accuracy of all settings are shown in Table 2.

Model / Test Dataset	Template Questions	SMILES Questions	
ChemData+FuzzySearch	0.20	-	
Ours	0.88	0.484	
Ours w/o translation	0.88	0.290	

**Table 2:** Evaluation of Entity Linking.

## 6.2 QA engine

In order to assess the accuracy of question answering, we first evaluated the QA engine implemented for each ontology using an evaluation dataset specifically created for the ontology. Table 4 presents the evaluation results for each ontology, which includes four metrics: hit 1 rate, hit 5 rate, hit 10 rate, and mean reciprocal ranking (MRR). The hit n rate indicates the percentage of instances where the true answer is found within the top-n answers returned by the QA engine. MRR is defined as follows:

$$MRR = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{rank_i},\tag{9}$$

where  $rank_i$  refers to the rank of the true answer and Q represents the total number of questions used for evaluation.

For each ontology, we calculated a filtered and an unfiltered set of accuracy results. Filtered accuracy measurements were obtained by providing the QA engines with the true head entity, while unfiltered accuracy measurements were calculated without providing the true head entity. Consequently, due to errors in EL, the filtered accuracy results tend to be higher than the unfiltered accuracy results.

Test	Recall	Precision	F1 score	Operator accuracy	Mean value error
Unfiltered	0.8367	0.7177	0.7726	1.0	0.4583
Filtered	0.8367	0.9684	0.8978	1.0	0.5519

**Table 3:** Accuracy of numerical filtering questions.

	hit 1 rate	hit 5 rate	hit 10 rate	mrr
	Int I fate	Int 5 fate	Int To Tate	
Pubchem	0.8122	0.8155	0.8155	0.8132
Pubchem filtered	0.9449	0.9544	0.9544	0.9996
Ontocompchem	0.3366	0.39	0.39	0.7971
Ontocompchem filtered	0.8147	0.9621	0.9621	0.8542
Ontokin	0.88	0.885	0.885	0.6767
Ontokin filtered	0.9943	1.0	1.0	0.8755
OntoSpecies	0.7907	0.8210	0.8210	0.7784
OntoSpecies filtered	0.9168	0.9519	0.9519	0.8920
Wikidata	0.4910	0.5731	0.5731	0.5255
Wikidata filtered	0.8182	0.9364	0.9364	0.8673
OntoMOPs	0.7356	0.8451	0.9132	0.8736
OntoMOPs filtered	0.8612	0.8933	0.9245	0.8952

**Table 4:** Evaluations of separate QA engines.

**Table 5:** Ablation test for score alignment.

	hit 1 rate	hit 5 rate	hit 10 rate	mrr
w/o score alignment	0.4762	0.9452	0.9469	0.5677
with score alignment	<b>0.9822</b>	<b>0.9879</b>	<b>0.9903</b>	<b>0.9784</b>

## 6.3 Language Models

Large Language models (LLMs) are primarily designed to process and generate text, which can lead to challenges when handling numbers and numerical operations. A clear example of this limitation can be observed in the performance of ChatGPT when dealing with numerically-focused questions. Although ChatGPT may provide accurate information in certain instances, such as identifying the density of benzene as  $0.87 \ g/cm^3$ , it can also generate incorrect or self-contradictory answers.

what is the density of benzene and find solvents that have less density than benzene
The density of benzene is approximately 0.879 grams per cubic centimeter (g/cm<sup>3</sup>).
There are several solvents that have lower density than benzene. Some examples include:

Acetone: Density of approximately 0.791 g/cm<sup>3</sup>.
Diethyl ether: Density of approximately 0.713 g/cm<sup>3</sup>.
Toluene: Density of approximately 0.866 g/cm<sup>3</sup>.
Ethanol: Density of approximately 0.789 g/cm<sup>3</sup>.

Dichloromethane (methylene chloride): Density of approximately 1.326 g/cm<sup>3</sup>.
Hexane: Density of approximately 0.660 g/cm<sup>3</sup>.

# Figure 7: Response from ChatGPT [36] to question "What is the density of benzene and find solvents that have less density than benzene". We observe the result incorrectly includes solvent with density greater than that of benzene.

For instance, when asked to find solvents with a lower density than benzene, ChatGPT incorrectly lists chloroform as one of the options, even though its density is 1.49  $g/cm^3$ , as illustrated in Figure 7. This example highlights the limitations of language models like ChatGPT when it comes to accurately processing and answering numerical questions.

# 7 Results

This section showcases the performance of Marie and BERT through a series of screenshots that provide a clear and insightful view of the KGQA system's capabilities in producing accurate responses. For instance, Figure 8 demonstrates the ability of Marie and BERT to answer a question from the Metal Organic Polyhedras (MOPs) domain. Figure 9 demonstrates the ability of the system to answer a question dynamically by invoking the Thermodynamic Data agent (STDC agent). The response includes a visual representation of the heat capacity of Cyclopropanone at constant pressure and constant volume. Lastly, Figure 10 illustrates Marie and BERT's proficiency in answering numerical questions. Users can interact with Marie and BERT by following this link https://como.ceb.cam.ac.uk/people/mab999/. However, this system is still under development and the accuracy of the results will increase with further refinement of the underlying ontologies.

<image><image><image><image><image>

#### Results

#### From the World Avata

Result	Assembly Model	Metal Organic Polyhedra
1	OCTAHEDRON   (4-PLANAR)6(3-PYRAMIDAL)8(OH)	[Cu2]6[(C6H3)(OC6H4)3(CO2)3]8
2	OCTAHEDRON   (4-PLANAR)6(3-PYRAMIDAL)8(OH)	[Cu2]6[C6(CH3)3(CH2)3(CO2)3]8
3	OCTAHEDRON   (4-PLANAR)6(3-PYRAMIDAL)8(OH)	[Cu2]6[(C6H3)(CONHC6H4)3(CO2)3]8
4	OCTAHEDRON   (4-PLANAR)6(3-PYRAMIDAL)8(OH)	[Cu2]6[C6H3(CONHCHCH3)3(CO2)3]8

**Figure 8:** Response from Marie and BERT to question "List the MOPs with assembly model (3-planar)4(3-pyramidal)4(Td)."

# 8 Conclusion

In this paper, we have introduced a novel IR-based KGQA system that employs hybrid knowledge graph embeddings to accurately answer natural language questions in the chemistry domain. Our system improves the robustness of chemistry KGQA by utilizing IR-based methods, which are less sensitive to errors compared to SP-based KGQA

# 

Explore Contact



#### Marie

This website presents a proof-of-concept search engine system for accessing chemical data from the World Avatar Knowledge Graph and the Wikidata Knowledge Graph. The Knowledge Graphs offers inter-connected data from chemical kinetics to chemical and physical properties of species and many other domains. We trained a question type classification model and an entity extraction model to interpret chemistry-related questions of interest. The system has a novel design which applies a topic model to identify the question-to-ontology affiliation to improve its accuracy.

To use the Marie search engine, type a question into the field below, or select one of the provided sample questions.



#### Results



Figure 9: Response from Marie and BERT to question "What is the heat capacity of C3H4O".

systems.

To address the heterogeneous structure of the TWA chemistry KG, we have applied different embedding strategies for various chemistry ontologies based on their complexity and structure. We have also proposed a score alignment model that re-ranks results from different embedding spaces by providing a score bias based on the question-ontology affiliation. This enables the production of a final list of inter-ontology answers scored on a uniform scale.

In order to manage the large volume of numerical data in the TWA KG, we have combined the translation embedding method, TransR, with a joint numerical embedding model to form TransRA. This model enhances the quality of KG embedding and, consequently, improves the accuracy of the information derivation task. Additionally, we have implemented a numerical filtering mechanism to reduce the time and space complexity of answer selection for numerical questions.

	End all apprica with hei			Find all species with boiling point above 0 celsius
	Pind all species with boi	ing point above o ceisius	~	
K				Certainly! Here are some examples of species (substances) with boiling points above 0 degrees Celsius:
				1. Hydrogen (H2) - Boiling point: -252.87 degrees Celsius at standard atmospheric pressure.
	lt.			2. Oxygen (O2) - Boiling point: -183 degrees Celsius at standard atmospheric pressure.
Resu	lits			3. Nitrogen (N2) - Boiling point: -195.79 degrees Celsius at standard atmospheric pressure.
From th	e World Avatar			4. Argon (Ar) - Boiling point: -185.85 degrees Celsius at standard atmospheric pressure.
				5. Carbon dioxide (CO2) - Boiling point: -78.5 degrees Celsius at standard atmospheric
	Result	result	value	pressure.
	1	C6H3CIN2O4	588.15	6. Methane (CH4) - Boiling point: -161.5 degrees Celsius at standard atmospheric pressure.
				7. Propane (C3H8) - Boiling point: -42.1 degrees Celsius at standard atmospheric pressure.
	2	C2H7N	280.53888	8. Butane (C4H10) - Boiling point: -0.5 degrees Celsius at standard atmospheric pressure.
	3	CH2O	374.2611	9. Ethylene (C2H4) - Boiling point: -103.7 degrees Celsius at standard atmospheric pressure.
	4	CHN	352.15	<ol> <li>Ammonia (NH3) - Boiling point: -33.34 degrees Celsius at standard atmospheric pressure.</li> </ol>
	5	H2	20.372223	
	6	H2O2	423.37222	These boiling points are approximate values at standard atmospheric pressure (1 atmosphere). It's important to note that the boiling points of substances can vary under
	7	C5H11NO2S	573.15	different pressure conditions.
	8	C3H4O2	345.37222	

**Figure 10:** Side-by-side responses from Marie and BERT (left) and ChatGPT [36] (right) to question "Find all species with boiling point above 0 celsius". The red box on the right indicates that the results returned by ChatGPT incorrectly includes species with boiling point less than 0 celsius. The response shown from Marie demonstrates a subset of the results captured for the purpose of this figure.

Furthermore, we have developed an implicit multi-hop relation derivation algorithm to handle non-shallow ontologies, such as OntoCompChem, which increases the QA accuracy. Marie and BERT is also capable of answering questions that require on-the-fly calculations by invoking semantic agents for dynamic calculations in a robust, scalable, and autonomous manner.

In addition, we have implemented a cost-effective SP-based method to handle numerous chemical reactions, supported by a Linked Data Fragment service, and to answer questions related to chemical reactions.

Finally, our EL module is designed to identify SMILES strings and translate them into their respective IUPAC names. As language models like BERT struggle with handling SMILES strings for similarity comparisons, this translation significantly improves the accuracy of identifying species represented by SMILES strings in a question.

However, the lack of a proper Chemistry QA dataset that is diverse and varied presents a challenge. The task requires both chemistry domain knowledge expertise and the understanding of the underlying KG structure, which poses difficulties.

Future work includes efforts to reduce the cost of training embedding models and making the system more robust. Furthermore, expanding the system to integrate the knowledge of other domains could also be a fruitful avenue for future research.

# Acknowledgements

This project is supported by CMCL Innovations and the National Research Foundation(NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise(CREATE) programme. Part of this work was also supported by Towards Turing 2.0 under the EPSRC Grant EP/W037211/1. Markus Kraft gratefully acknowledges the support of the Alexander von Humboldt foundation.

# Nomenclature

- AM Assembly Model
- BERT Bidirectional Encoder Representations from Transformers
- CBU Chemical Building Unit
- CGQA Cross Graph Question Answering
- CID Compound Identifier
- **DFT** Density Functional Theory
- dKG dynamic Knowledge Graph
- **DL** Description Logic
- EL Entity Linking
- **ER** Entity Recognition
- **GBU** Generic Building Unit
- **IRI** Internationalized Resource Identifier
- **IR** Information Retrieval
- **IUPAC** International Union of Pure and Applied Chemistry
- JPS J-Park Simulator
- LDF Linked Data Fragment
- LLM Large Language Model
- MOP Metal Organic Polyhedra
- NER Named Entity Recognition
- NLP Natural Language Processing
- PCE Power Conversion Efficiency
- PLM Pre-trained Language Model
- SMILES Simplified molecular-input line-entry system
- SPARQL SPARQL Protocol and RDF Query Language
- **SP** Semantic Parsing
- TWA The World Avatar

# Notations

Notations	Explanations
Ē	indexed set of all entities in $\mathscr{G}$ in IRI forms
$\overline{\mathscr{R}}$	indexed set of all relations in $\mathscr{G}$ in IRI forms
$\overline{\mathscr{G} = \{(h, r, t) : h, t \in \mathscr{E}, r \in \mathscr{R}\}}$	a knowledge graph
$\overline{(h,r,t)}$	a triple in the form of IRIs
$\overline{(\hat{e_h},\hat{r},\hat{e_t})}$	a triple in the form of embedding representations
n	the total number of entities in $\mathscr{G}$
m	the total number of relations in $\mathscr{G}$
d	the dimension of the embedding representations
$\overline{q}$	the question in text
$\overline{E_r \in \mathbb{R}^{m  imes d}}$	indexed embedding representations of all predicates in $\mathscr{G}$
$\overline{E_e \in \mathbb{R}^{n  imes d}}$	indexed embedding representations of all entities in ${\mathscr G}$
$\overline{\hat{e}_q \in \mathbb{R}^{1  imes d}}$	predicted relation embedding
$\overline{\hat{e_a}' \in \mathbb{R}^{1  imes d}}$	the embedding of a candidate answer for $q$
j	the number of entities within 3 hops distance from $h$
$\overline{A' = \{a_1',, a_j'\}}$	entities within 3 hops distance from h
$\overline{E_a = \{\hat{e_{a1}}, \dots, \hat{e_{aj}}\}}$	embeddings of entities within 3 hops distance from h
k	number of final answers
$\overline{A = \{a_1, \dots, a_k\}}$	the top-k answers in IRI forms ranked by their scores
$\overline{\phi(\hat{e_h},\hat{e_q},\hat{e_a}')}$	scoring function for a triple
η	learning rate for model training
$\overline{L(\cdot)}$	a loss function
$\odot$	element-wise product e.g. $\begin{bmatrix} a_1 & a_2 \end{bmatrix} \bigcirc \begin{bmatrix} b_1 & b_2 \end{bmatrix} = \begin{bmatrix} a_1b_1 & a_2b_2 \end{bmatrix}$
	concatenation operator e.g. $\begin{bmatrix} a_1 & a_2 \end{bmatrix} $ $\begin{bmatrix} b_1 & b_2 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 & b_2 \end{bmatrix}$
EL	Entity linking model
BERT <sub>BASE</sub>	The pretrained BERT model

# **Table 6:** Explanation of notations.

## Algorithms

Algorithm 1: Training of TransE embedding **Data:** set of all entity IRIs  $\mathscr{E}$ , set of all relation IRIs  $\mathscr{R}$ , training set  $S = \{(h, r, t) : h, t \in \mathcal{E}, r \in \mathcal{R}\} \in \mathcal{G}, \text{ margin } \gamma, \text{ embeddings dim } d, \text{ number of } d$ entities n, number of relations m, learning rate  $\eta$ , batch size b, number of epoches epoch\_num **Result:** Embeddings of entities  $E_e = \{\hat{e}_1, ..., \hat{e}_n\}$ , relations  $E_r = \{\hat{r}_1, ..., \hat{r}_m\}$ 1 initialization: 2  $E_e \leftarrow \{\hat{e}_i \in \mathbb{R}^d : \hat{e}_i = \text{uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}), 0 < i \le n\}$ 3  $E_r \leftarrow \{\hat{r}_i \in \mathbb{R}^d : \hat{r}_i = \text{uniform} \frac{-6}{\sqrt{d}}, \frac{\sqrt{a}}{\sqrt{d}}, 0 < i \le m \}$ 4  $\hat{r} \leftarrow \frac{\hat{r}}{||\hat{r}||}$  for each  $\hat{r} \in E_r$ ; s Ent\_embed(*e*) ←  $\hat{e}$  for each  $e \in \mathscr{E}$  and  $\hat{e} \in E_e$ ; 6 Rel\_embed(r)  $\leftarrow \hat{r}$  for each  $r \in \mathscr{R}$  and  $\hat{r} \in E_r$ ; while *epoch* < *epoch\_num* do 7  $\hat{e} \leftarrow \frac{\hat{e}}{||\hat{e}||}$  for each  $\hat{e} \in E_e$ ; 8  $S_{\textit{batch}} \gets ext{sample}(S,b); \textit{// sample}$  a batch of triples of size b9  $T_{batch} \leftarrow \emptyset$ ; 10 for  $(h, r, t) \in S_{batch}$  do 11 while  $(h', r, t') \in \mathscr{G}$  do 12  $h' \leftarrow \text{sample}(\mathscr{E}, 1);$ 13  $t' \leftarrow \operatorname{sample}(\mathscr{E}, 1);$ 14  $\hat{e}_h \leftarrow \text{Ent\_embed}(h) ; \hat{e}_t \leftarrow \text{Ent\_embed}((t) ;$ 15  $\hat{e_h}' \leftarrow \operatorname{Ent\_embed}(h')$ ;  $\hat{e_t}' \leftarrow \operatorname{Ent\_embed}(t')$ ;  $\hat{r}' \leftarrow \operatorname{Rel\_embed}(r)$ ; 16  $T_{batch} \leftarrow T_{batch} \cup \{((\hat{e}_h, \hat{r}, \hat{e}_t), (\hat{e}_h', \hat{r}, \hat{e}_t'))\};$ 17  $L(E_{e}, E_{r}) = \sum_{((\hat{e}_{h}^{\prime}, \hat{r}, \hat{e}_{t}^{\prime}), (\hat{e}_{h}, \hat{r}, \hat{e}_{t})) \in T_{barch}} [\max(0, \gamma + \phi(\hat{e}_{h}, \hat{r}, \hat{e}_{t}) - \phi(\hat{e}_{h}^{\prime}, \hat{r}, \hat{e}_{t}^{\prime}))];$ 18  $\hat{e} \leftarrow \hat{e} - \eta imes 
abla L(\hat{e}), \forall \hat{e} \in E_e; //$  update the parameters of entity 19 embeddings  $\hat{r} \leftarrow \hat{r} - \eta imes 
abla L(\hat{r}), orall \hat{r} \in E_r;$  // update the parameters of relation 20 embeddings  $epoch \leftarrow epoch + 1$ ; 21 22 return  $E_{e}, E_{r}$ ;

#### Algorithm 2: Training of TransEA embedding

```
Data: set of all entity IRIs \mathscr{E}, set of all relation IRIs \mathscr{R}, set of all attributes IRIs \mathscr{L},
                    set of all numerical laterals \mathscr{V}, training set
                    S = \{(h, r, t), (h, l, v) : h, t \in \mathscr{E}, r \in \mathscr{R}, l \in \mathscr{L}, v \in \mathscr{V}\} \in \mathscr{G}, \text{ margin } \gamma,
                    embeddings dim d, number of entities n, number of relations m, number of
                    attributes j, learning rate \eta, batch size b, number of epochs epoch_num
      Result: Embeddings of entities E_e = \{\hat{e}_1, ..., \hat{e}_n\}, relations E_r = \{\hat{r}_1, ..., \hat{r}_m\}, lateral
                        attributes E_l = \{\hat{l}_1, ..., \hat{l}_i\}, and lateral attributes biases E_b = \{\hat{b}_1, ..., \hat{b}_i\}
 1 initialization:
 2 E_e \leftarrow \{\hat{e}_i \in \mathbb{R}^d : \hat{e}_i = \operatorname{uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}), 0 < i \le n\}

3 E_r \leftarrow \{\hat{r}_i \in \mathbb{R}^d : \hat{r}_i = \operatorname{uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}), 0 < i \le m\}

4 E_l \leftarrow \{\hat{l}_i \in \mathbb{R}^d : \hat{l}_i = \operatorname{uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}), 0 < i \le j\}
 s E_b \leftarrow \{\hat{b}_i \in \mathbb{R}^1 : \hat{b}_i = \text{uniform}(\frac{-6}{\sqrt{d}}, \frac{6}{\sqrt{d}}), 0 < i \leq j \}
 6 \hat{r} \leftarrow \frac{\hat{r}}{||\hat{r}||} for each \hat{r} \in E_r;
 7 Ent_embed(e) \leftarrow \hat{e} for each e \in \mathscr{E} and \hat{e} \in E_e;
 8 Rel_embed(r) \leftarrow \hat{r} for each r \in \mathscr{R} and \hat{r} \in E_r;
 9 Attr_embed(l) \leftarrow \hat{l} for each l \in \mathscr{L} and \hat{l} \in E_l;
10 Bias_embed(l) \leftarrow \hat{b} for each l \in \mathscr{L} and \hat{b} \in E_b;
      while epoch < epoch_num do
11
              \hat{e} \leftarrow \frac{\hat{e}}{||\hat{e}||} for each \hat{e} \in E_e;
12
              S_{\textit{batch}} \leftarrow 	ext{sample}\left(S,b
ight); // sample a batch of triples of size b
13
              T_{batch} \leftarrow \emptyset;
14
              for (h, r, t), (h, l, v) \in S_{batch} do
15
                      while (h', r, t') \in \mathscr{G} do
16
                             h' \leftarrow \text{sample}(\mathscr{E}, 1);
17
                            t' \leftarrow \text{sample}(\mathscr{E}, 1);
18
                      \hat{e}_h \leftarrow \operatorname{Ent\_embed}(h); // \text{lookup embeddings with IRI}
19
                      \hat{e}_t \leftarrow \text{Ent\_embed}((t);
20
                     \hat{e_h}' \leftarrow \text{Ent\_embed}(h');
21
                     \hat{e}_t' \leftarrow \text{Ent\_embed}(t');
22
                      \hat{r}' \leftarrow \text{Rel\_embed}(r);
23
                     \hat{l} \leftarrow \text{Attr embed}(l);
24
                     \hat{b} \leftarrow \text{Bias\_embed}(l);
25
                    T_{batch} \leftarrow T_{batch} \cup \{((\hat{e}_h, \hat{r}, \hat{e}_t), (\hat{e}_h', \hat{r}, \hat{e}_t'), (\hat{e}_h, \hat{l}, v, \hat{b}))\};
26
             L_{r}(E_{e}, E_{r}) = \sum_{((\hat{e}_{h}', \hat{r}, \hat{e}_{r}'), (\hat{e}_{h}, \hat{r}, \hat{e}_{r})) \in T_{hatch}} [\max(0, \gamma + \phi(\hat{e}_{h}, \hat{r}, \hat{e}_{t}) - \phi(\hat{e}_{h}', \hat{r}, \hat{e}_{t}'))];
27
             L_l(E_l,E_b) = \sum_{(\hat{e}_h,\hat{l},v,\hat{b})\in T_{batch}} ||\hat{l}^{	op}\cdot\hat{e}_h + \hat{b} - v||_{1/2};
28
             L(E_e, E_r, E_l, E_b) = (1 - \alpha) \cdot L_r(E_e, E_r) + \alpha \cdot L_l(E_l, E_b);
29
              \hat{e} \leftarrow \hat{e} - \eta \times \nabla L(\hat{e}), \forall \hat{e} \in E_e; \hat{r} \leftarrow \hat{r} - \eta \times \nabla L(\hat{r}), \forall \hat{r} \in E_r;
30
                \hat{l} \leftarrow \hat{l} - \eta \times \nabla L(\hat{l}), \forall \hat{r} \in E_r; \hat{b} \leftarrow \hat{b} - \eta \times \nabla L(\hat{b}), \forall \hat{b} \in E_b; epoch \leftarrow epoch + 1;
31 return E_e, E_r, E_l, E_b;
```

Algorithm 3: Training of Complex embedding

**Data:** set of all entity IRIs  $\mathscr{E}$ , set of all relation IRIs  $\mathscr{R}$ , training set  $S = \{(h, r, t) : h, t \in \mathcal{E}, r \in \mathcal{R}\} \in \mathcal{G}, \text{ embeddings dim } d = 2l \text{ where } l \in \mathbb{Z} \text{ and}$  $l \ge 1$ , number of entities *n*, number of relations *m*, learning rate  $\eta$ , batch size b, number of epochs epoch\_num **Result:** Embeddings of entities  $E_e = \{\hat{e}_1, ..., \hat{e}_n\}$  and relations  $E_r = \{\hat{r}_1, ..., \hat{r}_m\}$ 1 initialization:; 2  $E_{re} \leftarrow \{ \hat{e}_i \in \mathbb{R}^l : \hat{e}_i = \text{xavier\_uniform}(l), 0 < i \le n \}$  $E_{im} \leftarrow \{ \hat{e}_i \in \mathbb{R}^l : \hat{e}_i = \text{xavier\_uniform}(l), 0 < i \leq n \}$  $R_{re} \leftarrow \{\hat{r}_i \in \mathbb{R}^l : \hat{r}_i = \text{xavier\_uniform}(l), 0 < i \le m\}$  $R_{im} \leftarrow \{\hat{r}_i \in \mathbb{R}^l : \hat{r}_i = \text{xavier\_uniform}(l), 0 < i \le m\}$ 3 Re\_ent(*e*) ←  $\hat{e}_{re}$  for each  $e \in \mathscr{E}$ ,  $\hat{e}_{re} \in E_{re}$ ; 4 Im\_ent(e)  $\leftarrow \hat{e}_{im}$  for each  $e \in \mathscr{E}$ ,  $\hat{e}_{im} \in E_{im}$ ; 5 Re\_rel(r)  $\leftarrow \hat{r}_{re}$  for each  $r \in \mathscr{R}$ ,  $\hat{r}_{re} \in R_{re}$ ; 6 Im\_rel(r)  $\leftarrow \hat{r}_{im}$  for each  $r \in \mathscr{R}$ ,  $\hat{r}_{im} \in R_{im}$ ; 7 while epoch < epoch\_num do</pre>  $S_{batch} \leftarrow \text{sample}(S, b) / / \text{ sample a batch of triples of size } b$ 8  $T_{batch} \leftarrow \emptyset$ ; 9 for  $(h, r, t) \in S_{batch}$  do 10 while  $(h', r, t') \in \mathscr{G}$  do 11  $h' \leftarrow \text{sample}(\mathscr{E}, 1);$ 12  $t' \leftarrow \operatorname{sample}(\mathscr{E}, 1);$ 13  $y \leftarrow 1, T_{batch} \leftarrow T_{batch} \cup (h, r, t, y);$ 14  $y \leftarrow 0, T_{batch} \leftarrow T_{batch} \cup (h', r, t', y);$ 15  $R_{batch} \leftarrow \emptyset$ ; 16 for  $(h, r, t, y) \in T_{batch}$  do 17  $\phi(h, r, t) = \text{Re}_\text{ent}(h) \odot \text{Re}_\text{ent}(t) \odot \text{Re}_\text{rel}(r) +$ 18 Im ent(h)  $\odot$  Im ent(t)  $\odot$  Re rel(r) + 19  $\text{Re}_{ent}(h) \odot \text{Im}_{ent}(t) \odot \text{Im}_{rel}(r)$  -20  $Im_ent(h) \odot Re_ent(t) \odot Im_rel(r)$ 21  $x = \sigma(\phi(h, r, t)); / / \sigma$ : plain sigmoid function 22  $R_{batch} \leftarrow R_{batch} \cup (x, y)$ 23  $L(E_{re}, E_{im}, R_{re}, R_{im}) = \sum_{(x,y) \in R_{batch}} [y \cdot \log(x) + (1-y) \cdot \log(1-x)];$ 24  $\hat{e}_{re} \leftarrow \hat{e}_{re} - \eta \times \nabla L(\hat{e}_{re}), \forall \hat{e}_{re} \in E_{re};$ 25  $\hat{e}_{im} \leftarrow \hat{e}_{im} - \eta \times \nabla L(\hat{e}_{im}), \forall \hat{e}_{im} \in E_{im};$ 26  $\hat{r}_{re} \leftarrow \hat{r}_{re} - \eta \times \nabla L(\hat{r}_{re}), \forall \hat{r}_{re} \in R_{re};$ 27  $\hat{r}_{im} \leftarrow \hat{r}_{im} - \eta \times \nabla L(\hat{r}_{im}), \forall \hat{r}_{im} \in R_{im};$ 28  $epoch \leftarrow epoch + 1$ 29 30  $\hat{e} \in \mathbb{R}^d \leftarrow \hat{e}_{re} \ \widehat{e}_{im}, orall \hat{e}_{re}, \hat{e}_{im} \in E_{re}, E_{im}$  // concatenate real and imaginary parts 31  $\hat{r} \in \mathbb{R}^d \leftarrow \hat{r}_{re} \frown \hat{r}_{im}, \forall \hat{r}_{re}, \hat{r}_{im} \in R_{re}, R_{im}$ 32 return  $E_e, E_r$ 

Algorithm 4: Training of the scoring model

**Data:** embedding dimension  $d, S = \{(q, r) : r \in \mathcal{R}, r \in \mathbb{R}^d\}$ , learning rate  $\eta$ , batch size *b*, number of epoches *epoch\_num*, pretrained BERT base model *BERT*<sub>BASE</sub>

**Result:** Weights of fully connected layer  $w \in \mathbb{R}^{768 \times d}$ 

1 while epoch < epoch\_num do

 $S_{batch} = \text{sample}(S, b);$  $for (q, r) \in S_{batch} do$  $L_q \leftarrow \text{tokenize}(q);$  $e_q \in \mathbb{R}^{768} \leftarrow BERT_{BASE}(t_q);$  $L(w) = \sum_{(q,r) \in S_{batch}} ||\hat{e}_q - \hat{r}||_1;$  $w \leftarrow w - \eta \times \nabla L(w)$ 9 return w

Algorithm 5: Training of score alignment model

**Data:**  $S = \{((q, S_t, l_t), (q, S_f, l_f))\}$ , where *q* is the question,  $S_t$  is the set of scores of the true answer,  $S_f$  is the set of scores of false answers returned by another embedding space,  $l_t, l_f \in \mathbb{Z}^2$  are embedding space labels for the true or false answer, batch size *b*, learning rate  $\eta$ , loss margin  $\gamma$ , number of epochs *epoch\_num* 

- pretrained BERT model *BERT<sub>BASE</sub>* **Result:** Wights of fully connected layer w<sub>1</sub> ∈ ℝ<sup>768×2</sup>, Weights of fully connected layer w<sub>2</sub> ∈ ℝ<sup>4×1</sup>
- 2 for epoch < epoch\_num do

```
S_{batch} = \text{sample}(S, b);
 3
              for ((q, S_t, l_t), (q, S_f, l_f)) \in S_{batch} do
 4
                      t_q \leftarrow \text{tokenize}(q);
 5
                      \hat{q} \in \mathbb{R}^2 \leftarrow w_1^T \cdot BERT_{BASE}(t_q);
 6
                      for s_t, s_f \in S_t, S_f do
 7
                              s_t \leftarrow s_t / \max(S_t);
 8
                            s_{f} \leftarrow s_{f} / \max(S_{f});

s_{t} \leftarrow s_{t} + w_{2}^{T} \cdot (\hat{q}^{\frown}l_{t});

s_{f} \leftarrow s_{f} + w_{2}^{T} \cdot (\hat{q}^{\frown}l_{f});
 9
10
11
             L(w_1, w_2) = \sum_{((q, S_t, l_t), (q, S_f, l_f)) \in T_{batch}} [\max(0, \gamma + S_f - S_t)];
12
              w_1 \leftarrow w_1 - \boldsymbol{\eta} \times \nabla L(w_1);
13
             w_2 \leftarrow w_2 - \eta \times \nabla L(w_2);
14
```

15 return  $w_1, w_2$ 

#### Algorithm 6: Algorithm for deriving implicit relations

**Data:** The target KG  $\mathscr{G} = \{(h, r, t)\}$ , sample size for derived triples *s*, set of entities with labels  $\mathcal{E}_l$ **Result:** The set the derived triples S' = (h, r', t)1  $\mathscr{R}' \leftarrow \emptyset, \mathscr{G}' \leftarrow \emptyset, S' \leftarrow \emptyset;$ 2 for  $h \in \mathscr{E}_l$  do 3 *hop*  $\leftarrow$  0;  $H \leftarrow \{h\};$ 4  $R(h,h) \leftarrow \emptyset$ ; 5 while hop < 3 do 6  $H' \leftarrow \emptyset$ ; 7 for  $h' \in H$  do 8 for  $(h', r, t) \in \mathscr{G}$  do 9  $R(h',t) \leftarrow \{r\};$ 10  $H' \leftarrow H' \cup \{t\};$ 11  $R(h,t) \leftarrow R(h,h') \cup R(h',t);$ 12  $\mathscr{G}' \leftarrow \mathscr{G}' \cup \{(h, R(h, t), t)\};$ 13  $H \leftarrow H'$ ; 14  $hop \leftarrow hop + 1$ ; 15 16 for  $(h, R(h, t), t) \in \mathscr{G}'$  do  $R' \leftarrow R(h,t);$ 17 **if** |R'| > 1 **then** 18  $r' \leftarrow \operatorname{hash}(R')$ ; 19  $S' \leftarrow S' \cup \{(h, r', t)\}$ 20 21  $S' \leftarrow \text{sample}(S', s)$ ; 22 return S'

Algorithm 7: Fine-tuning of plain EL Entity-Question Bi-encoder on BERT

**Data:** Training set  $S = \{(e, q, q_m) : e \in \mathscr{E}\}$ , question text q, mention text  $q_m$ , BERT<sub>BASE</sub> transformers  $T_e$ ,  $T_{q1}$ , the pretrained parameters of  $T_e$ ,  $T_{q1}$ :  $\theta_e$ ,  $\theta_{q1}$ , hyperparameters: learning rate  $\eta$ , epoch\_num

**Result:** Transformers  $T_e$ ,  $T_{q1}$  with updated parameters  $\theta_e$  and  $\theta_{q1}$  respectively 1 while *epoch* < *epoch\_num* **do** 

 $S_{\textit{batch}} \leftarrow ext{sample}\left(S,b
ight)$  // sample a batch of triples of size b 2 3 for  $(e,q) \in S_{batch}$  do 4  $e' \leftarrow \text{tokenize}(e)$ 5  $q' \leftarrow \text{tokenize}(q)$ 6  $v_e = F(T_e(e'))$ , F taking the output of [CLS] token 7  $v_q = F(T_{q1}(q'))$ , F taking the output of [CLS] token 8  $s(e',q') = v_e \cdot v_q$ 9  $loss = -s(e',q') + \log\left(\sum_{j=1}^{b} \exp(s(e'_j,q'))\right)$ 10  $egin{aligned} & heta_e \leftarrow heta_e - \eta \cdot 
abla loss \ & heta_{q1} \leftarrow heta_{q1} - \eta \cdot 
abla loss \end{aligned}$ 11 12

13 return  $\theta_e$ ,  $\theta_{q1}$ 

#### Algorithm 8: Fine-tuning of joint EL and ER Entity-Question Bi-encoder on BERT

- **Data:** Training set  $S = \{(e, q, q_m) : e \in \mathscr{E}\}$ , question text q, true mention text  $q_m$ , BERT<sub>BASE</sub> transformers  $T_{q2}$  with pretrained parameters  $\theta_{q2} = [\theta_1 \dots \theta_L]$ , L the number of layers in BERT<sub>BASE</sub>, fine-tuned entity encoder  $T_e$  with parameters  $\theta_e$ , mention classifier with parameters { $w_{start}, w_{end}, w_{mention}$ }, hyperparameters: learning rate  $\eta$ , epoch\_num
- **Result:** Transformer  $T_{q2}$  with updated parameters  $\theta_{q2}$  and mention classifier with updated parameters  $\{w_{start}, w_{end}, w_{mention}\}$

```
1 while epoch < epoch_num do
```

```
S_{batch} \leftarrow 	ext{sample}\left(S,b
ight) // sample a batch of size b
  2
  3
              loss \leftarrow 0
  4
              for (e,q) \in S_{batch} do
  5
                      e' \leftarrow \text{tokenize}(e)
  6
                      v_e = F(T_e(e')), F taking the output of [CLS] token
  7
                      q' \leftarrow \text{tokenize}(q)
  8
  9
                      v_{q1}\ldots v_{qn} = T_{q2}(q')
                      for [i, j], \forall 1 \leq i < j \leq n do
10
                              v_{i,j} = \frac{1}{(j-i+1)} \sum_{t=i}^{j} v_{qt}
11
                           p[i, j] = \sigma(w_{start}^{T} v_{qi} + w_{end}^{T} v_{qj} + \sum_{t=i}^{j} w_{mention}^{T} v_{qt})
s(e, [i, j]) = v_{e} \cdot v_{i,j}
p(e|[i, j]) = \frac{exp(s(e, [i, j]))}{\sum_{e' \in \mathscr{E}} exp(s(e', [i, j]))}
loss = loss + \frac{1}{N} CrossEntropy(\mathbb{1}([i, j] = a), p[i, j]) - logp(e|[i, j])
12
13
14
15
                      \theta_{q2} \leftarrow \theta_{q2} - \eta \cdot \nabla loss
16
                      w_{start} \leftarrow w_{start} - \eta \cdot \nabla loss
17
                      w_{end} \leftarrow w_{end} - \eta \cdot \nabla loss
18
                      w_{mention} \leftarrow w_{mention} - \eta \cdot \nabla loss
19
20 return \theta_{q2}
```

#### Algorithm 9: Fine-tuning of SMILES Name Recognition on BERT

**Data:** Training set S = (q, l), q the question text of token length *w*, l the ground truth type sequence for q, of length *w*. BERT<sub>*BASE*</sub> transformers  $T_{SMILES}$ , with pretrained model parameters  $\theta_T = [\theta_{qT,1}...\theta_{qT,L}]$ , Linear Classifier *C* with model parameters  $\theta_C = [\theta_{qC,1}...\theta_{qC,L'}]$ , *L*, *L'* the respective number of model layers, hyperparameters: learning rate  $\eta$ , epoch\_num

**Result:** Transformer  $T_{SMILES}$ , Classifier C with updated parameters  $\theta_T$ ,  $\theta_C$  respectively

1 while *epoch* < *epoch\_num* do

```
2 S_{batch} \leftarrow \text{sample}(S, b) / / \text{ sample a batch of size } b
```

 $\begin{array}{cccc} \mathbf{3} & & ; \\ \mathbf{4} & & \mathbf{for} \ (q,l) \in S_{batch} \ \mathbf{do} \\ \mathbf{5} & & & \\ \mathbf{6} & & & \\ \mathbf{7} & & & \\ \mathbf{7} & & & \\ \mathbf{8} & & & \\ \mathbf{9} & & & \\ \mathbf{9} & & & \\ \mathbf{10} & & & \\ \mathbf{7} & & \\ \mathbf$ 

11 return  $\theta_T, \theta_C$ 

# References

- [1] A. Abujabal, M. Yahya, M. Riedewald, and G. Weikum. Automated Template Generation for Question Answering over Knowledge Graphs. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1191–1200, Perth Australia, 2017. International World Wide Web Conferences Steering Committee. doi:10.1145/3038912.3052583.
- [2] T. Al-Moslmi, M. G. Ocana, A. L. Opdahl, and C. Veres. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, 8:32862–32881, 2020. doi:10.1109/ACCESS.2020.2973928.
- [3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *DBpedia: A Nucleus for a Web of Open Data*, pages 722–735. Springer, 2007.
- [4] H. Bast and E. Haussmann. More Accurate Question Answering on Freebase. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 1431–1440. ACM, 2015. doi:10.1145/2806416.2806472.
- [5] J. Berant and P. Liang. Semantic Parsing via Paraphrasing. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1415–1425. Association for Computational Linguistics, 2014. doi:10.3115/v1/P14-1133.
- [6] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase. In Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, page 1247. ACM Press, 2008. doi:10.1145/1376616.1376746.
- [7] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.
- [8] S. Cao, J. Shi, Z. Yao, X. Lv, J. Yu, L. Hou, J. Li, Z. Liu, and J. Xiao. Program Transfer for Answering Complex Questions over Knowledge Bases. arXiv e-prints, 2021. doi:10.48550/ARXIV.2110.05743.
- [9] L. Chen, G. Varoquaux, and F. M. Suchanek. A lightweight neural model for biomedical entity linking. *arXiv preprint arXiv:2012.08844*, 2020.
- [10] S. Chen, Q. Liu, Z. Yu, C.-Y. Lin, J.-G. Lou, and F. Jiang. ReTraCk: A Flexible and Efficient Framework for Knowledge Base Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 325–336, Online, 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.acl-demo.39.
- [11] W. Cui, Y. Xiao, H. Wang, Y. Song, S.-w. Hwang, and W. Wang. KBQA: learning question answering over QA corpora and knowledge bases. *Proceedings of the VLDB Endowment*, 10(5):565–576, 2017. doi:10.14778/3055540.3055549.

- [12] R. Das, M. Zaheer, D. Thai, A. Godbole, E. Perez, J. Y. Lee, L. Tan, L. Polymenakos, and A. McCallum. Case-based Reasoning for Natural Language Queries over Knowledge Bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9594–9611, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.emnlp-main.755.
- [13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 10 2018.
- [14] J. Ding, W. Hu, Q. Xu, and Y. Qu. Leveraging Frequent Query Substructures to Generate Formal Queries for Complex Question Answering. arXiv preprint arXiv:1908.11053, 2019. doi:10.48550/ARXIV.1908.11053.
- [15] A. Eibeck, M. Q. Lim, and M. Kraft. J-Park Simulator: An ontology-based platform for cross-domain scenarios in process industry. *Computers & Chemical Engineering*, 131:106586, 2019. doi:10.1016/j.compchemeng.2019.106586.
- [16] F. Farazi, J. Akroyd, S. Mosbach, P. Buerger, D. Nurkowski, M. Salamanca, and M. Kraft. OntoKin: An Ontology for Chemical Kinetic Reaction Mechanisms. *Journal of Chemical Information and Modeling*, 60(1):108–120, 2020. doi:10.1021/acs.jcim.9b00960.
- [17] F. Farazi, M. Salamanca, S. Mosbach, J. Akroyd, A. Eibeck, L. K. Aditya, A. Chadzynski, K. Pan, X. Zhou, S. Zhang, et al. Knowledge graph approach to combustion chemistry and interoperability. *ACS omega*, 5(29):18342–18348, 2020.
- [18] D. Fensel, U. Şimşek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, A. Wahler, D. Fensel, et al. Introduction: what is a knowledge graph? *Knowledge graphs: Methodology, tools and selected use cases*, pages 1–10, 2020.
- [19] Y. Gu and Y. Su. ArcaneQA: Dynamic Program Induction and Contextualized Encoding for Knowledge Base Question Answering. arXiv preprint arXiv:2204.08109, 2022. doi:10.48550/ARXIV.2204.08109.
- [20] Y. Gu, S. Kase, M. Vanni, B. Sadler, P. Liang, X. Yan, and Y. Su. Beyond I.I.D.: Three Levels of Generalization for Question Answering on Knowledge Bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488, Ljubljana Slovenia, 2021. ACM. doi:10.1145/3442381.3449992.
- [21] C. Guo, C. Zhang, X. Han, and Y. Ji. AWML: adaptive weighted margin learning for knowledge graph embedding. *Journal of Intelligent Information Systems*, 53(1): 167–197, 2019. doi:10.1007/s10844-018-0535-2.
- [22] S. Hu, L. Zou, and X. Zhang. A State-transition Framework to Answer Complex Questions over Knowledge Base. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2098–2108, Brussels, Belgium, 2018. Association for Computational Linguistics. doi:10.18653/v1/D18-1234.

- [23] X. Huang, J. Zhang, D. Li, and P. Li. Knowledge Graph Embedding Based Question Answering. In Proc. Twelfth ACM Int. Conf. Web Search Data Min., pages 105–113, New York, NY, USA, 2019. ACM. doi:10.1145/3289600.3290956.
- [24] X. Huang, J. Zhang, D. Li, and P. Li. Knowledge Graph Embedding Based Question Answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 105–113, Melbourne VIC Australia, 2019. ACM. doi:10.1145/3289600.3290956.
- [25] Z. Ji, Q. Wei, and H. Xu. Bert-based ranking for biomedical entity normalization. *arXiv preprint arXiv:1908.03548*, 2019.
- [26] R. J. Kate, Y. W. Wong, and R. J. Mooney. Learning to transform natural to formal languages. In *Proceedings of the 20th National Conference on Artificial Intelligence* - *Volume 3*, AAAI'05, page 1062–1068. AAAI Press, 2005.
- [27] A. Kondinski, J. Bai, S. Mosbach, J. Akroyd, and M. Kraft. Knowledge engineering in chemistry: From expert systems to agents of creation. *Accounts of Chemical Research*, pages 11713–11728, 2022.
- [28] N. Krdzavac, S. Mosbach, D. Nurkowski, P. Buerger, J. Akroyd, J. Martin, A. Menon, and M. Kraft. An ontology and semantic web service for quantum chemistry calculations. *Journal of chemical information and modeling*, 59(7):3154–3165, 2019.
- [29] N. Krdzavac, S. Mosbach, D. Nurkowski, P. Buerger, J. Akroyd, J. Martin, A. Menon, and M. Kraft. An Ontology and Semantic Web Service for Quantum Chemistry Calculations. *Journal of Chemical Information and Modeling*, 59(7): 3154–3165, 2019. doi:10.1021/acs.jcim.9b00227. Number: 7.
- [30] Y. Lan, S. Wang, and J. Jiang. Multi-hop Knowledge Base Question Answering with an Iterative Sequence Matching Model. In 2019 IEEE International Conference on Data Mining (ICDM), pages 359–368, Beijing, China, 2019. IEEE. doi:10.1109/ICDM.2019.00046.
- [31] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pretrained biomedical language representation model for biomedical text mining. *arXiv* preprint arXiv:1901.08746, 2019. doi:10.1093/bioinformatics/btz682.
- [32] B. Z. Li, S. Min, S. Iyer, Y. Mehdad, and W. tau Yih. Efficient one-pass end-to-end entity linking for questions. *arXiv preprint arXiv:2010.02413*, 2020.
- [33] P. Liang. Lambda Dependency-Based Compositional Semantics, Sept. 2013. arXiv:1309.4408 [cs].
- [34] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

- [35] S. Mosbach, A. Menon, F. Farazi, N. Krdzavac, X. Zhou, J. Akroyd, and M. Kraft. Multiscale cross-domain thermochemical knowledge-graph. *Journal of Chemical Information and Modeling*, 60(12):6155–6166, 2020.
- [36] OpenAI. ChatGPT, 2023. URL https://chat.openai.com/. Last access: 2023-05-16.
- [37] L. Pascazio, S. Rihm, A. Naseri, S. Mosbach, J. Akroyd, and M. Kraft. A chemical species ontology for data integration and knowledge discovery, 2023. URL https: //como.ceb.cam.ac.uk/preprints/306/. Submitted for publication.
- [38] M. Petrochuk and L. Zettlemoyer. SimpleQuestions Nearly Solved: A New Upperbound and Baseline Approach. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 554–558, Brussels, Belgium, 2018. Association for Computational Linguistics. doi:10.18653/v1/D18-1051.
- [39] J. Pérez, M. Arenas, and C. Gutierrez. Semantics and Complexity of SPARQL. In *The Semantic Web - ISWC 2006*, volume 4273, pages 30–43. Springer, 2006. doi:10.1007/11926078\_3. Series Title: Lecture Notes in Computer Science.
- [40] Y. Qiu, K. Zhang, Y. Wang, X. Jin, L. Bai, S. Guan, and X. Cheng. Hierarchical Query Graph Generation for Complex Question Answering over Knowledge Graph. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1285–1294, Virtual Event Ireland, 2020. ACM. doi:10.1145/3340531.3411888.
- [41] K. Rajan, A. Zielesny, and C. Steinbeck. Stout: Smiles to iupac names using neural machine translation. *Journal of Cheminformatics*, 13, 2021. doi:10.1186/s13321-021-00512-4.
- [42] D. J. Rao, S. S. Mane, and M. A. Paliwal. Biomedical Multi-hop Question Answering Using Knowledge Graph Embeddings and Language Models. arXiv preprint arXiv:2211.05351, 2022. doi:10.48550/ARXIV.2211.05351. URL https://arxi v.org/abs/2211.05351.
- [43] S. Ravishankar, J. Thai, I. Abdelaziz, N. Mihidukulasooriya, T. Naseem, P. Kapanipathi, G. Rossiello, and A. Fokoue. A Two-Stage Approach towards Generalization in Knowledge Base Question Answering. *arXiv preprint arXiv:2111.05825*, 2021. doi:10.48550/ARXIV.2111.05825. Publisher: arXiv Version Number: 2.
- [44] A. Saxena, A. Tripathi, and P. Talukdar. Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings. In *Proceed*ings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4498–4507, Online, 2020. Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.412.
- [45] O. Sevgili, A. Shelmanov, M. Arkhipov, A. Panchenko, and C. Biemann. Neural entity linking: A survey of models based on deep learning. *arXiv preprint arXiv:2006.00575*, 2020. doi:10.3233/SW-222986.

- [46] C. Shang, G. Wang, P. Qi, and J. Huang. Improving Time Sensitivity for Question Answering over Temporal Knowledge Graphs. arXiv preprint arXiv:2203.00255, 2022. doi:10.48550/ARXIV.2203.00255. URL https://arxiv.org/abs/2203 .00255. Publisher: arXiv Version Number: 1.
- [47] Y. Si, J. Wang, H. Xu, and K. Roberts. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26:1297–1304, 2019. ISSN 1067-5027. doi:10.1093/jamia/ocz096.
- [48] Y. Sun, L. Zhang, G. Cheng, and Y. Qu. SPARQA: Skeleton-Based Semantic Parsing for Complex Questions over Knowledge Bases. *Proceedings of the AAAI Conference* on Artificial Intelligence, 34(05):8952–8959, 2020. doi:10.1609/aaai.v34i05.6426.
- [49] M. C. Swain and J. M. Cole. Chemdataextractor: a toolkit for automated extraction of chemical information from the scientific literature. *Journal of chemical information and modeling*, 56(10):1894–1904, 2016.
- [50] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.
- [51] T. Van Veen. Wikidata. Information technology and libraries, 38(2):72-81, 2019.
- [52] R. Verborgh, M. Vander Sande, P. Colpaert, S. Coppens, E. Mannens, and R. Van de Walle. Web-scale querying through Linked Data Fragments. In C. Bizer, T. Heath, S. Auer, and T. Berners-Lee, editors, *Proceedings of the 7th Workshop on Linked Data on the Web*, volume 1184 of *CEUR Workshop Proceedings*, 2014.
- [53] D. Vrandečić. Wikidata: A new platform for collaborative data collection. In Proceedings of the 21st International Conference on World Wide Web, WWW '12 Companion, page 1063–1064, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2187980.2188242.
- [54] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge Graph Embedding by Translating on Hyperplanes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), 2014. doi:10.1609/aaai.v28i1.8870. URL https://ojs.aaai.org/ind ex.php/AAAI/article/view/8870.
- [55] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*, 2019.
- [56] Y. Wu and Z. Wang. Knowledge Graph Embedding with Numeric Attributes of Entities. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 132–136. Association for Computational Linguistics, 2018. doi:10.18653/v1/W18-3017.
- [57] Y. Xiao, L. Zhang, J. Huang, L. Zhang, and J. Wan. An Information Retrieval-Based Joint System for Complex Chinese Knowledge Graph Question Answering. *Electronics*, 11(19):3214, 2022. doi:10.3390/electronics11193214.

- [58] I. Yamada, K. Washio, H. Shindo, and Y. Matsumoto. Global entity disambiguation with bert. In *Proceedings of the 2022 Conference of the North American Chapter* of the Association for Computational Linguistics: Human Language Technologies, pages 3264–3271, 2022.
- [59] W. Yao and R. Huang. Temporal event knowledge acquisition via identifying narratives. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.
- [60] X. Yao, J. Berant, and B. Van Durme. Freebase QA: Information Extraction or Semantic Parsing? In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 82–86, Baltimore, MD, 2014. Association for Computational Linguistics. doi:10.3115/v1/W14-2416.
- [61] X. Ye, S. Yavuz, K. Hashimoto, Y. Zhou, and C. Xiong. RNG-KBQA: Generation Augmented Iterative Ranking for Knowledge Base Question Answering. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6032–6043, Dublin, Ireland, 2022. Association for Computational Linguistics. doi:10.18653/v1/2022.acl-long.417.
- [62] W.-t. Yih, M.-W. Chang, X. He, and J. Gao. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1321–1331, Beijing, China, 2015. Association for Computational Linguistics. doi:10.3115/v1/P15-1128.
- [63] H. Zafar, G. Napolitano, and J. Lehmann. Formal Query Generation for Question Answering over Knowledge Bases. In A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, editors, *The Semantic Web*, volume 10843, pages 714–728. Springer International Publishing, Cham, 2018. doi:10.1007/978-3-319-93417-4\_46.
- [64] W. Zheng, J. X. Yu, L. Zou, and H. Cheng. Question answering over knowledge graphs: question understanding via template decomposition. *Proceedings of the VLDB Endowment*, 11(11):1373–1386, 2018. doi:10.14778/3236187.3236192.
- [65] X. Zhou, A. Eibeck, M. Q. Lim, N. B. Krdzavac, and M. Kraft. An agent composition framework for the j-park simulator-a knowledge graph for the process industry. *Computers & Chemical Engineering*, 130:106577, 2019.
- [66] X. Zhou, D. Nurkowski, S. Mosbach, J. Akroyd, and M. Kraft. Question Answering System for Chemistry. *Journal of Chemical Information and Modeling*, 61(8):3868– 3880, 2021. doi:10.1021/acs.jcim.1c00275. Number: 8.
- [67] X. Zhou, D. Nurkowski, A. Menon, J. Akroyd, S. Mosbach, and M. Kraft. Question answering system for chemistry—A semantic agent extension. *Digital Chemical Engineering*, 3:100032, 2022. doi:10.1016/j.dche.2022.100032.