Preprint Cambridge Centre for Computational Chemical Engineering ISSN 1473 – 4273

A Question Answering System for Chemistry

Xiaochi Zhou[1], Daniel Nurkowski[4], Sebastian Mosbach[1,2], Jethro Akroyd[1,2], Markus Kraft[1,2,3]

released: January 25, 2021

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

[2] CARES
Cambridge Centre for Advanced Research and Education in Singapore
1 Create Way
CREATE Tower, #05-05
Singapore, 138602

[3] School of Chemical and Biomedical Engineering
Nanyang Technological University
62 Nanyang Drive
Singapore, 637459

[4] CMCL Innovations
Sheraton House
Castle Park, Castle Street
Cambridge, CB3 0AX
United Kingdom

Preprint No. 266

UNIVERSITY OF CAMBRIDGE

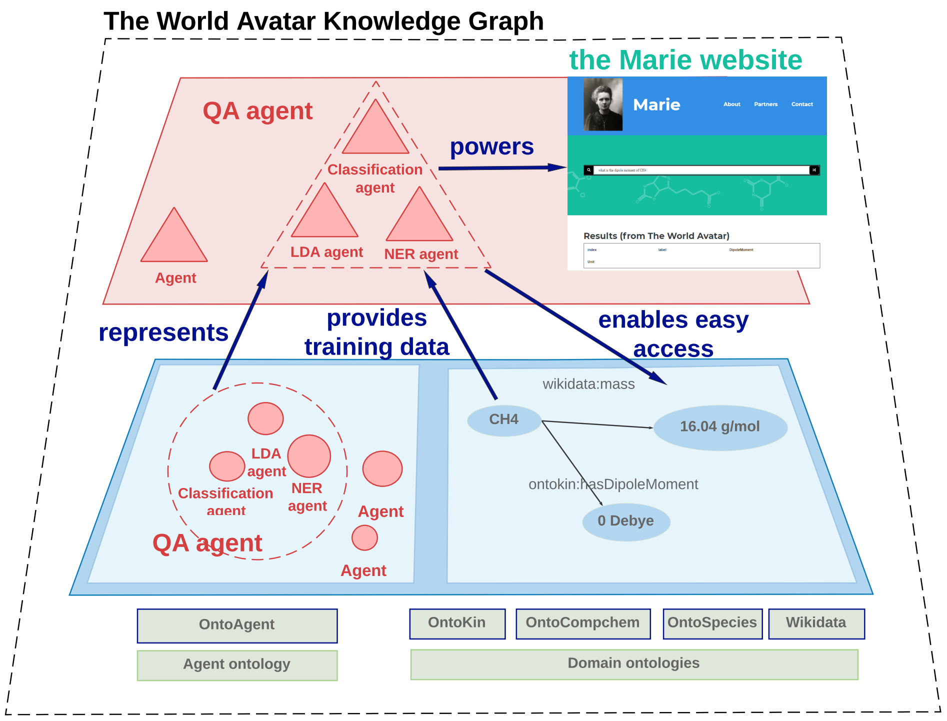
Keywords: Question Answering system, Natural language processing, Knowledge Graph

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 describes the implementation and evaluation of a proof-of-concept Question Answering system for accessing chemical data from knowledge graphs which offer data from chemical kinetics to chemical and physical properties of species. We trained a question type classification model and an entity extraction model to interpret chemistry questions of interest. The system has a novel design which applies a topic model to identify the question-to-ontology affiliation. The topic model helps the system to provide more accurate answers. A new method that automatically generates training questions from ontologies is also implemented. The question set generated for training contains 80085 questions under 8 types. Such a training set has been proven to be effective for training both the question type classification model and the entity extraction model. We evaluated the system using the Google search engine as the baseline. We found that it can answer 114 questions of interest that Google or Wolfram alpha can not give a direct answer to. Moreover, the application of the topic model was found to increase the accuracy of constructing the correct queries.



Highlights

- A proof-of-concept Question Answering system for chemical data is built.
- A novel design that integrates a topic model for better accuracy is investigated.
- A training set of 80085 questions is automatically generated.
- The training set is effective for training both the question classification and the entity extraction model.

Contents

1 Introduction

Proposals such as Industry 4.0 [26] and Eco-industrial park [25] aim to reduce energy consumption and carbon emission by improving the efficiency of the collaboration between industrial components across domains, such as power systems, transportation systems, and chemical plants. One criterion for the collaboration efficiency is the communication efficiency between the industrial components.

One solution to enable friction-free communication between components from different domains is to build ontologies as a common grounds for data representation and exchange [22]. In an ontology, entities, concepts, and relations are represented by Internationalized Resource Identifier (IRIs), and connected to each other in the form of triples, each triple includes two nodes and an edge where the nodes are entities or concepts and the edge is the relation between the nodes. The IRI representation enables explicit and unambiguous reference to concepts and entities and hence resolve the communication friction between systems. A collection of ontologies where their nodes are inter-connected with nodes from other ontologies forms a Knowledge Graph (KG). The inter-connected graph structure integrates the ontologies from different domains and allows more efficient automatic navigation of data. As a result, data exchange within KGs are efficient as well.

A knowledge graph contains data and information from various domains. Among them, chemistry-related data plays an important role for enabling cross-domain and multi-layer applications on top of a knowledge graph. For example, to simulate emissions from power plants, the combustion related data such as reaction mechanism is necessary.

One KG that includes chemical data is the World Avatar KG (WAKG), which is also referred to as the J-Park Simulator (JPS) KG in some earlier literature [30]. The WAKG is an attempt to create a general world model. It includes OntoKin [17] for chemical kinetics and reaction mechanisms, OntoCompChem [24] for quantum calculations, and OntoSpecies for chemical species [18]. It is also connected to a part of the Wikidata KG, which provides basic physical and chemical properties such as heat capacity and mass for chemical substances.

However, to access and utilize the data from a knowledge graph, it is necessary to formulate semantic queries, such as a SPARQL query. The semantic queries can fully utilized the advantages of KGs mentioned above and accurately retrieve information based on complex conditions. The barrier for formulating such semantic queries is high as it requires expertise on semantic web technologies and familiarity with the schema and content of the KG.

One solution to lower this barrier is to establish a Question Answering (QA) system [43]. A QA system allows end-users to pose natural language queries and accurately query information based on complex conditions. For example, a QA system for chemistry should aim to answer questions like "find the chemical structure of all the fatty acids having a molecular weight of more than 200".

A QA system utilizes machine learning models and natural language processing tools to interpret questions posed by end-users and convert the questions into machine-understandable queries. The common components of a QA system include classification of the questions, extraction of key components within the questions, mapping of the key components to their semantic representation, and query construction. The typical types of questions that can be answered include yes-or-no questions, factoid questions, list questions, and summary [42].

A number of projects have been proposed to build both open-domain and domain specific QA systems. For example, Question Answering over Linked Data (QALD) [32] is a series of evaluation campaigns on question answering over linked data. Since 2011, 9 challenges have been held and yielded many projects for open-domain QA systems. Some recent projects include WDAqua [14], ganswer2 [49], and QAKis [7]. A domain specific challenge for biomedical information QA systems is BioASQ [41]. Some prominent systems include HPI [36] and DeepMeSH [33].

However, to the best of our knowledge, there is no QA system built for chemistry data. To establish accurate named entity recognition of a domain-specific QA systems, it requires machine learning models specifically trained for the domain of interest. As a result, to build a QA system for chemistry data, it is necessary to train new named entity recognition models for handling chemistry question.

In addition, a large-scale KG such as WAKG, consists of many ontologies from various sources with different domains. However, it is common that there is overlapping of content between ontologies. Meanwhile, although similar information can be offered by different ontologies, the information quality may vary due to their different specializations. For example, the Wikidata KG stores some information about reactions. However, OntoKin, as an ontology specifically made for chemical reactions, the quality of information is significantly higher than the Wikidata KG. As a result, to guarantee the answer quality, it is necessary to have a mechanism to choose the preferred ontology to answer such questions.

Therefore, this paper introduces our implementation and investigation of a proof-of-concept QA system built on top of models specifically trained for chemistry-related questions. It also leverages topic modelling technology to identify the affiliation between questions and ontologies. A topic model is a statistical model for discovering the abstract "topics" that occur in a collection of documents. Such a model can also be used to analyse the composition of "topics" within a sentence such as a question.

Moreover, the establishment of named entity recognition models requires training data, usually in the form of labelled text and labelled questions. However, to the best of our knowledge, there is no such model or training data available for the chemistry domain. In addition, the manual creation of a sufficiently large training set is difficult and expensive, as it requires domain expertise. Therefore, we also propose a method to automatically generate and label training questions, leverage the rich information contained in ontologies.

Therefore, the purpose of this paper is to, firstly, develop a proof-of-concept design of a QA system for chemistry, and secondly, to study the effectiveness of the novel implementation of the topic model and the use of automatically generated questions.

2 Background

In this section we review ongoing research efforts that we found important when implementing a KG-based natural language query interface, including the KGs WAKG and Wikidata KG, as well as, natural language processing tools, and KG-based natural language query interfaces.

2.1 The World Avatar

Large cross-domain systems such as industrial symbioses, chemical plants, and cities are constituted by components such as power generators, storage tanks, and abstract industrial operations, which are from diverse domains. In order to achieve complex tasks including running simulations and optimizations, and coordination of multiple components, the relevant data, knowledge, and models must be integrated. However, the communication friction due to the heterogeneous conventions across domains hinders such an integration.

The World Avatar project aims to represent the physical world in all aspects. Such a concept is extended from the Digital Twin concept, which emphases on creating virtual representation of entities such as a device or an operation in industrial processes. The virtual representation will allow uniform integration between not only device and device but also devices and operations. Such uniform integration is similar to upgrading the Internet of Things to the Internet of Services and more.

One specific implementation of the World Avatar project is the J-Park Simulator (JPS) [15, 19, 46], which provides a data management common ground for those components and enable semantic interoperability so that cross-domain integration can be enabled. The JPS is now fully integrated in the WAKG. Figure 1 demonstrates the architecture of the JPS KG.

A number of ontologies from different domains, containing the definition and schema of concepts and relations as well as semantically described instances, form the terminology and instance layer of the WAKG. In addition, in order to update and maintain the large-scale KG over time, a number of agents, which carry out functions including data retrieval, simulation, and data update, are part of WAKG and operate on top of it.

2.1.1 Knowledge graph

A knowledge graph [8] is a collection of interconnected ontologies [16] and an ontology is a set of nodes including classes, properties, and instances denoted Internationalized Resource Identifiers (IRIs) [1]. For example, in the Wikidata knowledge graph, the node describing the chemical species methane is defined by `https://www.Wikidata.org/wiki/Q37129`. IRIs provide both access to the declaration of the nodes as well as a universally unique identifier for the node. As a result, referring to nodes with IRIs solves the ambiguity problem when referring to entities or concepts with natural language terms. For

[1]https://www.w3.org/International/O-URL-and-ident.html

instance, the word "methane" can also refer to a 1969 Italian movie but within the Wikidata ontology, the chemical species and the movie are assigned with a distinct IRI. Thus, due to the disambiguation, the data within a KGs are machine-readable and the conflicts between the convention of different data sources are eliminated. In addition, the nodes in KGs are connected and accessible through IRIs over the internet. All statements about the schema and the data are expressed in the form of semantic triples. A semantic triple is constituted by three entities, the subject, the predicate, and the object. For example, in the statement that declares the molecular weight of Benzene, the subject is "wd:Q2270", which is IRI of "Benzene". The predicate in the statement is "wdt:P2067", which is the IRI of "mass", while the object is the value 78.047. The object can also be a subject in another triple, connected to another entity such as the unit of the value. Therefore, a machine can navigate through the KGs and retrieve all the data related to a node with ease, even the nodes are stored distributed over the Internet, and such a feature enables complex queries over the KGs. Moreover, the ontologies not only share the common ground for data management, they are also interconnected following the Linked Data [5] principle, which means the entities in one ontology are connected to entities in other ontologies via IRIs if they are relevant. For example, a power plant node in the electricity system ontology can be connected to the node representing a city in where the plant is located within the Wikidata KG. Such as inter-connection between ontologies allows mutual enrichment between them. To access the data in KGs, the main tool is SPARQL Protocol and RDF Query Language (SPARQL) [45], which allows deep search of data within the KG by providing conditions on relations or values. SPARQL queries can also be used to update, delete, or insert information in the KG and the structure of these queries is almost identical to the queries for searching for information.

With the features of KGs mentioned above, different chemistry-related ontologies including OntoKin representing chemical kinetic reaction mechanisms, OntoCompChem representing quantum chemistry calculations, and OntoSpecies representing chemical species [17] are seamlessly integrated into the WAKG.

OntoKin is an ontology that captures the data and the semantics of chemical kinetic reaction mechanisms, where the mechanisms can be applied to simulate and understand the behaviour of chemical processes. One example is to support the simulation of the pollutant emission from internal combustion engines on top of the WAKG. The OntoKin ontology contains information including the phase of the substance, chemical reaction, reaction rate coefficients, thermodynamic models, and transport model. The OntoCompChem ontology covers features related to computational chemistry calculations of various properties of species, such as functional and basis set.

Besides the ontologies in the chemical domain, ontologies from any other domains can be integrated into the WAKG easily. The domain ontologies now in the WAKG include OntoPowerSys [13], which describes power systems, and OntoCityGML for city models, and so on. The ontologies not only share a common ground for data management, they are also interconnected to enrich each other.

2.1.2 Agents

Agents serve important roles in software environments nowadays. The main differences between an agent [21] and traditional software are, firstly, agents adopt a Service-Oriented Architecture [3]. From one aspect, agents are web services deployed on web servers that are accessible through the internet. As a result, the agents can be used in a distributed environment where digital resources are stored on different networked computers [37]. Secondly, different from web services, agents allow automatic management by computers including automatic discovery, composition, and execution [20, 27, 39]. The automatic management of agents is enabled by describing the details of the functions and protocols of agents with computer languages such as SOAP and WSDL.

Many agents have been implemented to maintain and update KGs. To better manage and utilize the agents, they are described in the KG with agent ontologies. For example, in the WAKG, the agents are described by an agent ontology OntoAgent [47] so that the functionality and communication standards can be understood by machines. In this the agents themselves are part of the KG. OntoAgent represents agents with their I/O signature by connecting the I/O parameters to classes of domain ontologies and also defines the restrictions on the I/O data. As a result, the agents can be automatically discovered and coordinated. With OntoAgent described agent documents, Zhou et al.[47] has implemented an agent composition framework on top of the JPS KG. The framework automatically puts the agents in sequence based on their I/O signatures so that a complex workflow containing multiple agents is created, forming a composite agent. As a result, the framework enables the KG to create new agents out of existing agents on demand. Zhou et al.[48] also implemented an agent marketplace on top of Blockchain-based Smart Contracts, which is tamper-proof and unbiased, to evaluate and monitor the performance of the agents.

2.2 Wikidata

Wikidata is a community-created knowledge graph of Wikipedia, where the semantic web technologies [4] are leveraged to build a structured database for Wikipedia content. Wikidata not only offers a machine-readable database for knowledge in almost all domains, it also includes a very detailed ontological class hierarchy for chemical species. For example, the entity of "Benzene" is not only under the class "chemical compound" but also "class IB flammable liquid","occupational carcinogen","male reproductive toxicant","developmental toxicant" and "carcinogen". Such a detailed classification for entities allows efficient selection of instances in their KG based on the classes, which is fairly useful for chemistry databases. In addition, the data comprehensiveness in Wikidata is also high for chemical species. Take the same example of Benzene, there are 212 properties connected to this node, including chemical formula, refractive index, molecular weight, ionization energy, autoignition temperature and so on. It also contains the identifiers of the species in other databases such as PubChem CID, ChemSpider ID, and CAS Registry Number, making the connection of the Wikidata chemical data to external databases.

However, to the best of our knowledge, the Wikidata only offers basic physical and chemical properties of species, more detailed data such as thermochemistry data, phase change data, and IR spectrum are not included.

2.3 KG-based natural language query

To lower the barrier of accessing data in the KGs and other similar structured databases, many efforts have been made to develop natural language query interfaces. For example, Kaufmann et al. [23] proposed an early implementation of a natural language interface to query ontologies, which converts questions into SPARQL queries. In addition, it asks the user for clarification when there is ambiguity in the question. Wang et al. [44] proposed PANTO system that also translates questions into SPARQL queries by leveraging a set of off-the-shelf parsers. Tablan et al. [40] also introduced their QuestIO system which converts natural language questions into SeRQL language or other queries. Later, they presented another system FREyA. It includes an ontology-based lookup service to map natural language term into IRIs. Al-Zubaide and Issa [1] proposed an ontology-based ChatBot, where Artificial Intelligence Markup Language (AIML) is used to create a mapping of questions and queries in the form of categories and store the resulted scenarios. Numerous other research efforts in this field are worth mentioning. For example, Mishra and Khilwani [31] has proposed the QUASE ontology-Based Domain Specific Natural Language Question Answering System, which includes a question classification module to increase the accuracy of the query construction. Saha et al. [35] proposed ATHENA, which is an ontology-driven system for natural language querying over relational data stores.

The aforementioned implementations share similar designs. The query interfaces first parse the questions and separate and tag different components using Natural Language Processing (NLP) tools. For example, in the question "show me all the chemical species with molecular weight larger than 200", an ideal parsing result will be "chemical species" as a "class", "molecular weight" as an "attribute", "larger" as a "comparison operator", and "200" as a "numerical value". The parsing result may also include the relation between the different components. Based on the parsing result, the next step is to convert the terms such as "molecular weight" into their Semantic representations, which are IRIs. The modules for mapping terms to IRIs are usually referred to as Ontology Lookup Service (OLS). One well-known implementation of the OLS system is proposed by Côté et al. [11]. In addition, several upgrades [10, 12] on the systems have been presented. The basic working mechanism is to create a mapping between the natural language labels of IRIs and the IRIs and return IRIs based on the string similarity between the IRI labels and the input terms. The last step of those systems is to fill the results obtained from the first two steps into SPARQL query templates. Several recalls may be repeated to return the answer.

2.4 Natural language processing toolkits

To support the interaction between computers and natural languages, many open-source natural language processing toolkits with different features have been developed. Existing toolkits include NLTK [28], OpenNLP [2], CoreNLP [29], Gensim [50], and Rasa framework [34]. To select the most suitable tools for each part of the query interface, these toolkits are evaluated in the context of chemistry and KGs. Some common functions offered by NLP toolkits are:

- **Tokenization**: to converts a sentence into a list of words for further processing;
- **Stop words filtering**: to filter out commonly used words such as "a" or "the";
- **Stemming**: to reduce inflected words to their root form. For example, to convert "products" and "produces" to "produc";
- **Part-of-speech (POS) tagging**: to mark the part of speech. For example, to tag "are" as "VBP (verb plural)";
- **Text parsing**: to analyse a sentence and convert it into a tree structure indicating the syntactic relation between its components;
- **Named entity recognition (NER)**: to extract and classify objects appeared in a sentence. For example, in the sentence "what is Benzene", some NER models can extract "Benzene" and classify it as a "species".

Natural Language Toolkit (NLTK) is an open-source Python-based platform with common NLP functions including tokenization, stop words filtering, stemming, part-of-speech (POS) tagging, sentence parsing. One of the most prominent advantages is that it provides convenient interfaces to many corpora and lexicons. Therefore, the NLTK can be used without training any model, therefore it is suitable for the pre-processing of training documents or other simple tasks. However, the disadvantage of NLTK is that it does not offer the function to train or customize the natural language models and does not offer NER functions.

The Apache OpenNLP is a Java library that supports functions including tokenization, POS tagging, NER, parsing, and coreference resolution, which is to identify the noun phrases in the same sentences that refer to the same entity. For example, in the sentence "We like cats because they are cute", coreference resolution can identify that "cats" and "they" are referring to the same entities. Similar to NLTK, the OpenNLP offers common NLP functions but only model training.

Stanford CoreNLP is another Java-based NLP framework. Besides the aforementioned common NLP tasks, Stanford CoreNLP offers advanced sentiment analysis. CoreNLP is recognized as one of the most accurate NLP tools for NER and has been widely applied in NLP-related applications. In addition, CoreNLP allows users to expand the NER module by adding labelled entities to the module.

Gensim is the state-of-the-art Python-based open-source library for topic modelling, which discovers the abstract topic within a collection of documents. Gensim's topic modelling is built on top of Latent Dirichlet Allocation [6] (LDA), which is a form of unsupervised machine learning. Gensim's topic modelling library helps users to create topic models. The users only need to provide a set of training documents and set up several training parameters, the library will extract a number of abstract topic from the documents and train topic models that can identify the probability that a word belongs to one or more topics.

Table 1: *Sample questions under the type "item_attribute_query"*

Question list
1. show me the [production method](attribute) of [C11H14N2S]((entity))
2. what are the [frvd](attribute) of [naloxone](entity)
3. what are [described in url](attribute) of [mizolastine](entity)
4. the [English Vikidia ID](attribute) of [C5H8O2](entity)
5. show the [picture](attribute) of [sec-butyllithium](entity)
6. what is the [treats](attribute) of [C7H15NO3](entity)
7. what are the [award won](attribute) of Itraconazole(entity)
8. what [part](attribute) does [C4H6O](entity) include
9. list [Kemler code](attribute) of [C4H4O4](entity)
10. what is [CaN2O6](entity) a [subsystem of](attribute)
11. what is the [hardness](attribute) of [stearophanate](entity)
12. what are the [IOR](attribute) of [sterogyl](entity)
13. what is [L-ornithine](entity) the [form of](attribute)
14. the [manufacturer](attribute) of [immunocytophyte](entity)

The Rasa framework is a conversational AI framework for building contextual assistants. Different from the aforementioned toolkits, the purpose of the Rasa framework is to provide all the tools necessary to build advanced Chatbots that naturally interact with human users. The Rasa framework contains four major modules: the interpreter module, the tracker module, the policy module, and the action module. The interpreter is built on top of NLP tools, to conduct tasks such as tokenization, text parsing, and NER. The result returned by the interpreter module contains the intent of the questions and the classified entities found in the sentence. The tracker module keeps track of the conversations and can provide information about the previous interactions. The policy module chooses the action from the Chatbot, such as ask the user for more information or query a database.

The Rasa interpreter module allows the users to train customized intent classification models and NER models. The users are required to provide questions where each question is labelled with their intent and each component in the sentence is highlighted and labelled with its classification. The intent classification models and NER models are built on top of the text embeddings technology.

3 Design and implementation

The overall working mechanism of our QA system is to convert a natural language question into a SPARQL query, which retrieves the desired information from a specific ontology in the KG. To achieve such purpose, there are several subsystems implemented:

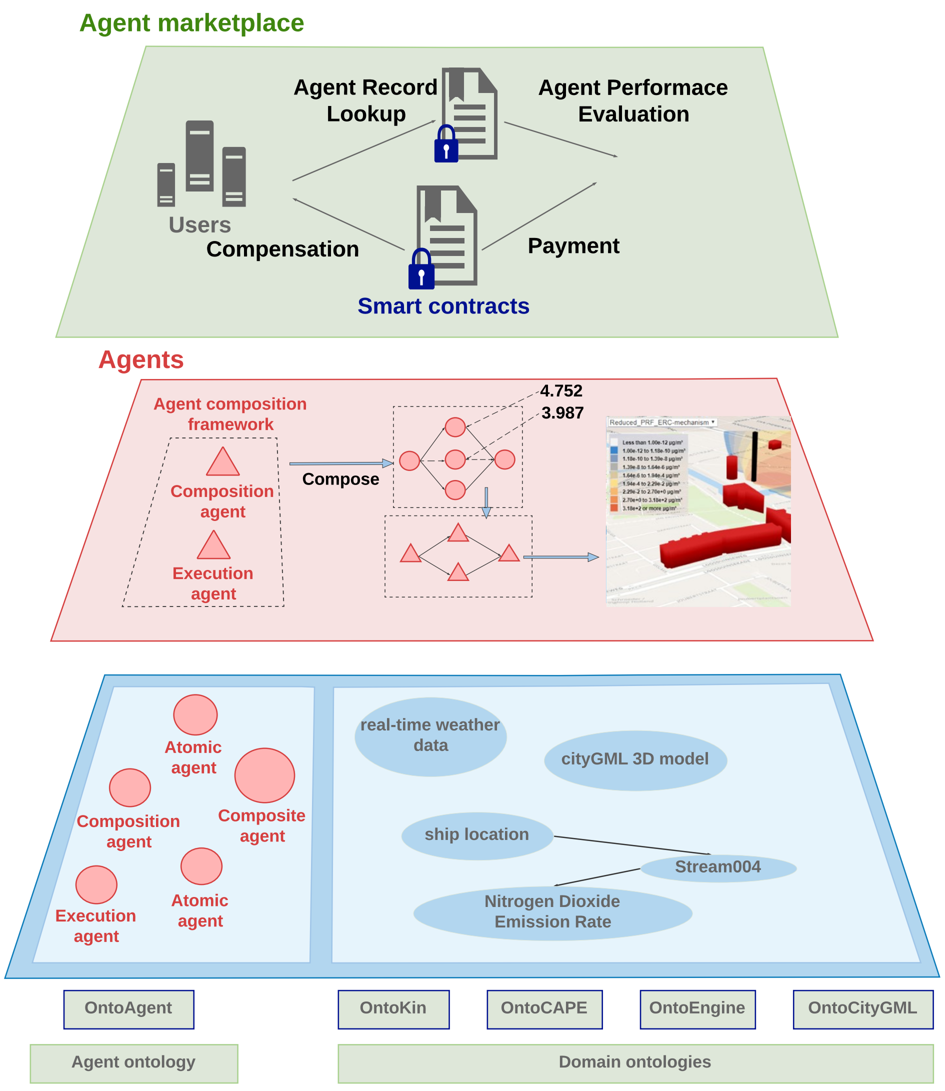


Figure 1: *The architecture of the J-Park Simulator KG as part of the WAKG. The green boxes on the bottom are the terminology parts of the ontologies while the blue layer contains the instances of the ontologies, including the red circles which are the semantic descriptions of agents. The red layer contains the operating agents which are red triangles. On top of the agent layer, an agent marketplace is monitoring and evaluating the performance of the agents.*

- **Topic model agent**: to identify the affiliation between the question and the domain ontologies.
- **Question classification agent**: to identify the type of the question and map the question to the according SPARQL query template;
- **Named entity recognition agent**: to extract the key components from the question;
- **Ontology lookup agent**: to convert the key components in the question into IRIs;

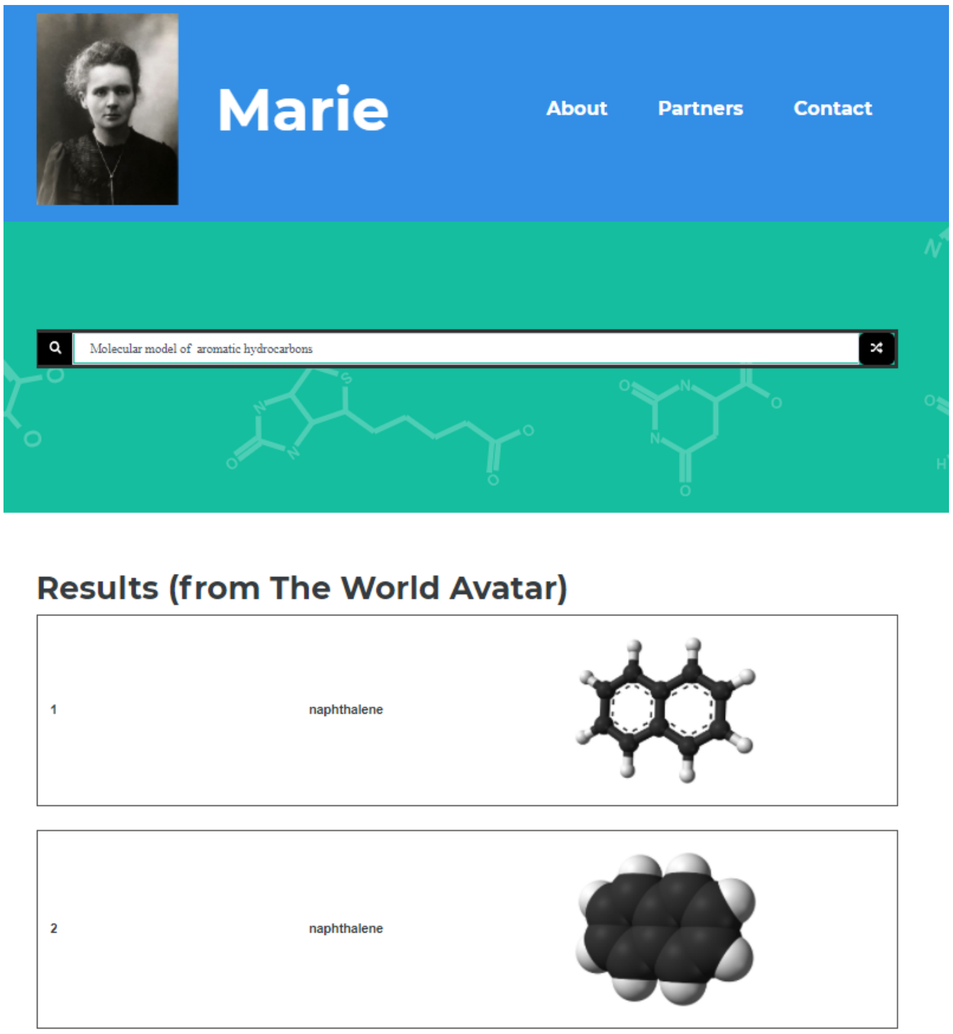

Marie
About
Partners
Contact
Molecular model of aromatic hydrocarbons
Results (from The World Avatar)
1
naphthalene
2
naphthalene

Figure 2: *A screenshot of the QA website "Marie" for chemistry data. The website is powered by the QA system presented in this paper.*

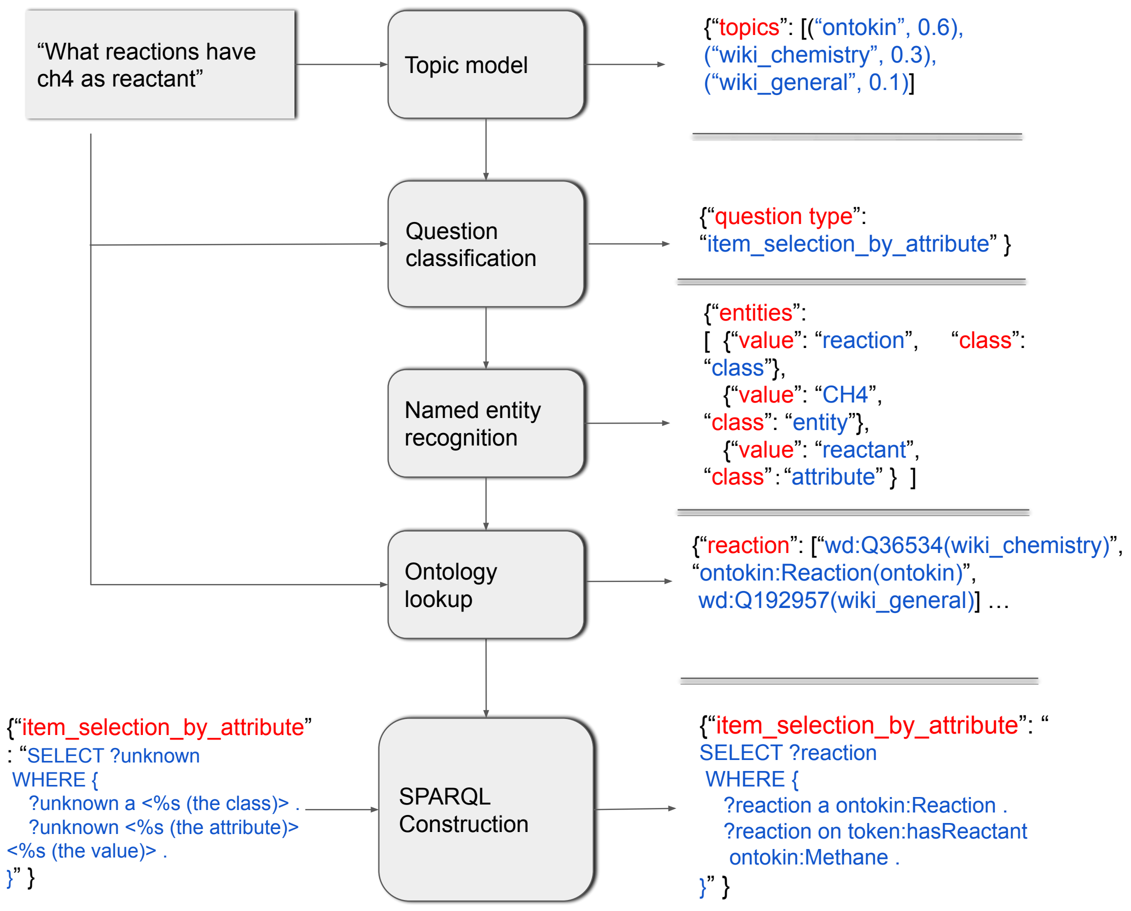

"What reactions have ch4 as reactant"
Topic model
{"topics": [("ontokin", 0.6), ("wiki_chemistry", 0.3), ("wiki_general", 0.1)]
Question classification
{"question type": "item_selection_by_attribute" }
Named entity recognition
{"entities": [{"value": "reaction", "class": "class"}, {"value": "CH4", "class": "entity"}, {"value": "reactant", "class": "attribute" }]
Ontology lookup
{"reaction": ["wd:Q36534(wiki_chemistry)", "ontokin:Reaction(ontokin)", wd:Q192957(wiki_general)] …
SPARQL Construction
{"item_selection_by_attribute" : "SELECT ?unknown WHERE { ?unknown a <%s (the class)> . ?unknown <%s (the attribute)> <%s (the value)> . }" }
{"item_selection_by_attribute": " SELECT ?reaction WHERE { ?reaction a ontokin:Reaction . ?reaction on token:hasReactant ontokin:Methane . }" }

Figure 3: *The general workflow of the query interfaces and the intermediate outputs of each module. The example of converting the question "what reactions have ch4 as reactant" into a SPARQL query is selected.*

3.1 Workflow

Figure 3 demonstrates the workflow of the QA system converting a natural language question into a valid SPARQL query.

When the QA system receives a natural language question, the question classification agent will identify its question type and a SPARQL query template will be assigned for each question type. If more than one possible question type are identified, a list of question types, which are ranked based on their confidence score, will be returned. Then the question classification agent passes the question and the question type to the named entity recognition agent. The named entity recognition agent will extract and classify the key components. Also, from the question classification result, the named entity recognition agent knows the numbers and the types of the key components within the question. Therefore, the named entity recognition agent can verify whether the recognition result matches the question type. If the match fails, the QA system will switch to the question type with

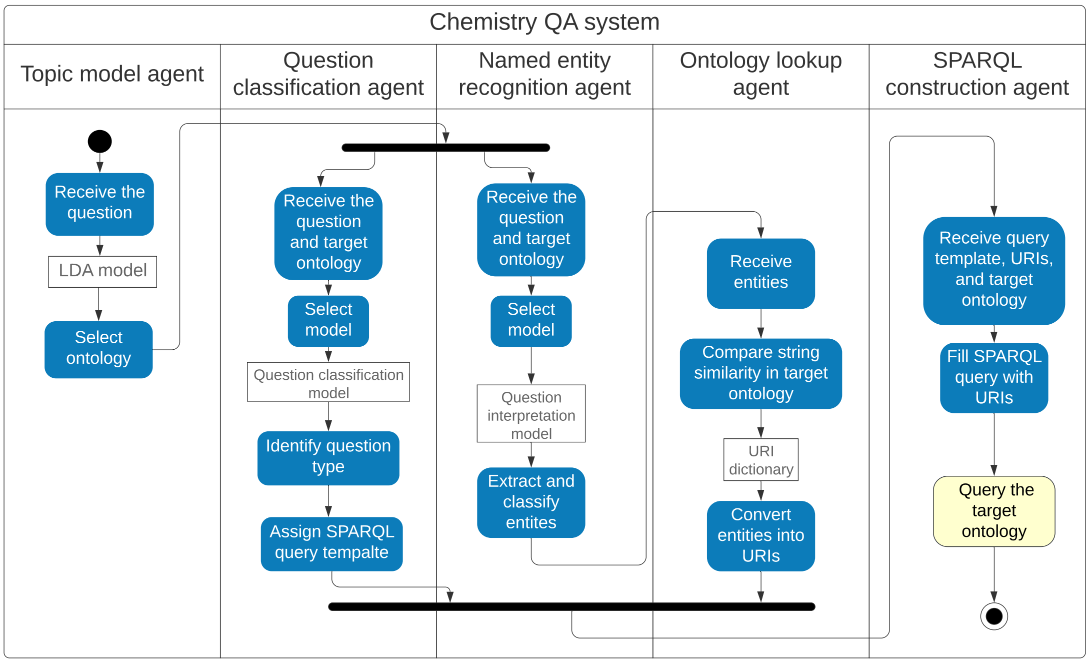

Chemistry QA system
Topic model agent
Question classification agent
Named entity recognition agent
Ontology lookup agent
SPARQL construction agent
Receive the question
LDA model
Select ontology
Receive the question and target ontology
Select model
Question classification model
Identify question type
Assign SPARQL query tempalte
Receive the question and target ontology
Select model
Question interpretation model
Extract and classify entites
Receive entities
Compare string similarity in target ontology
URI dictionary
Convert entities into URIs
Receive query template, URIs, and target ontology
Fill SPARQL query with URIs
Query the target ontology

Figure 4: *A UML activity diagram that demonstrates the activity of the agents included in the QA system.*

the next highest confidence.

Then the named entity recognition agent will pass the key components to the ontology lookup agent, which searches the IRIs in the predefined dictionary based on the key components. As the ontology lookup agent conducts a fuzzy string match to find the IRIs, a ranked list of IRIs will be returned for each component. Moreover, as some entities or classes exist in multiple ontologies, multiple IRIs from different ontologies with identical scores might be found in the dictionary. As a result, the topic classification agent will help the ontology lookup agent to find the best IRI match. The ontology lookup agent will pass the components and the question to the topic classification agent to identify the affiliation between the question and the ontologies. Based on the affiliation scores, the ontology lookup agent will further rank the IRIs based on the string similarity and the topic affiliation.

At the last step, the SPARQL construction agent will construct the SPARQL query by filling the IRIs into the assigned SPARQL query template. The constructed SPARQL query will be then sent to the according SPARQL query.

A website named "Marie" is also implemented as the graphical user interface of the QA system. Figure 2 shows a screenshot of the website.

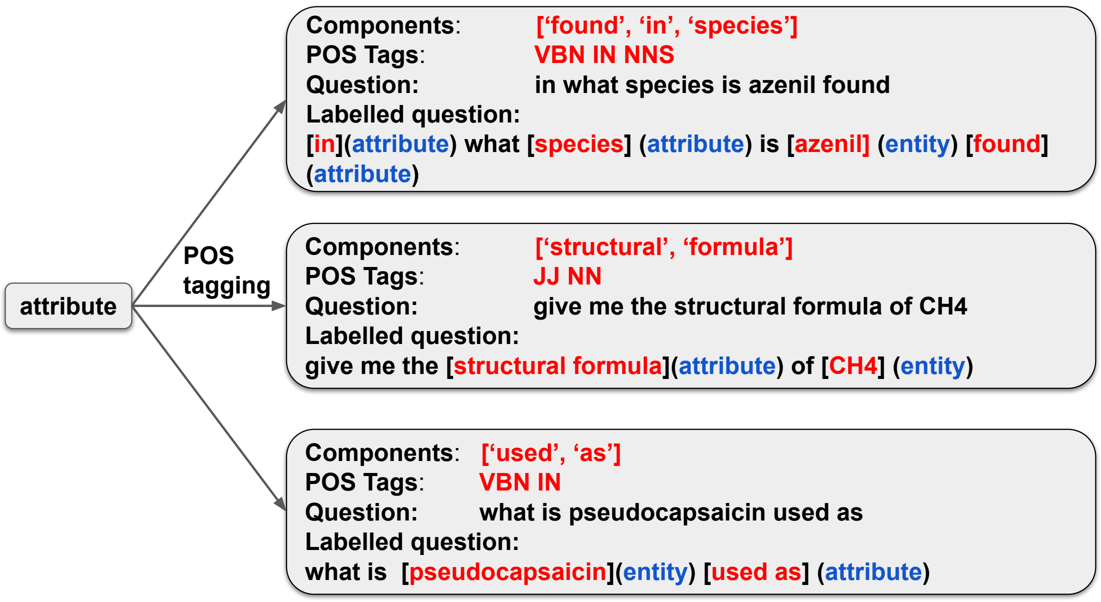

attribute
POS tagging
Components: ['found', 'in', 'species']
POS Tags: VBN IN NNS
Question: in what species is azenil found
Labelled question:
[in](attribute) what [species] (attribute) is [azenil] (entity) [found] (attribute)
Components: ['structural', 'formula']
POS Tags: JJ NN
Question: give me the structural formula of CH4
Labelled question:
give me the [structural formula](attribute) of [CH4] (entity)
Components: ['used', 'as']
POS Tags: VBN IN
Question: what is pseudocapsaicin used as
Labelled question:
what is [pseudocapsaicin](entity) [used as] (attribute)

Figure 5: *The process of generating questions of different grammatical structures based on the Part-of-Speech tagging result of the attribute. Each black box represents a question example, where the first line is the texts of the attribute, the second line is the POS tags of the texts, and the last line is the questions generated with their components tagged.*

3.2 Preliminary work

This section first describes the creation of training data and the training of three machine learning models on which the QA system is built: the question topic model, classification model, and the named entity recognition model. It then describes the creation of the term-IRI dictionary for the ontology lookup agent.

3.2.1 Data preparation

To train the question classification model, it requires a training set of questions labelled with their structural category. At the same time, the training of named entity recognition model requires a set of questions where their entities are underlined and labelled with their types. To train these two models, we created a set of training questions. Meanwhile, the entities in each question are underlined and labelled with their types. Table 1 shows some examples of the questions with highlighted entities.

Due to the high cost of manually creating the training set, we implemented a Python script to generate questions automatically. For each structural category, the script generates questions in a certain form. For example, the category "item_attribute_query", contains questions asking one attribute and one instance such as "what is CH4 used as". To create a question in this category, the script takes one instance and one of its relevant attribute in the ontologies and finds their natural language labels, for example, "used as" and "CH4".

Then the script applies Part-of-Speech (POS) tagging to the attribute "used as". Since the attribute is made up by a verb in past particle tense (VBN) and a preposition (IN), it is put behind the instance "CH4" to form the grammatically correct question "what is CH4 used as". Figure 5 gives examples of how questions of different grammatical structures are created under the "item_attribute_query" category using POS tagging. In addition, for training the named entity recognition model, the script also label "used as" as "attribute" and "CH4" as "entity". In total, we generated 80000 questions under 9 categories and the questions contain 6 types of entities. The question categories and the entity types are listed in Appendix A, are generated.

The training of the topic model requires a document for each ontology, which contains all the natural language words appearing in such ontology. We used SPARQL query to retrieve the labels and the comments of the nodes in the ontologies and put them in the training documents for topic model. In addition, some parts of the IRIs, such as OntoKin:ReactionMechanism, are also included in the documents after using regular expression to separate the words "reaction" and "mechanism". The documents are further processed with the Natural Language Toolkit (NLTK). The words are reduced into their stems and the stop words, which are the most frequently used English words, are removed.

3.2.2 Model training

The topic model is a Latent Dirichlet Allocation [6] (LDA), which is a form of unsupervised machine learning. The model is trained by the Gensim.models.ldamodel.LdaModel class provided by the Python-based Genism library. The documents are passed to the LdaModel function. There are four parameters for training the model: **"num_topic"**, **"passes"**, **"alpha"**, **"eta"**. **"num_topic"** is the number of topics expected. In our case, there are four ontologies: OntoCompChem (quantum calculation), OntoKin (reaction kinetic), OntoSpecies (chemical species), and Wikidata (chemical properties). The parameter **"passes"** defines the number of algorithm iterations. Due to the relatively small size of our training documents, **"passes"** is set to be 1000. The two parameters **"alpha"** and **"eta"** represent the dirichlet parameters for document-topic distribution and topic-word distribution. As the documents are assumed to follow dirichlet distribution,**"alpha"** and **"eta"** are set to be the default values 0.9 and 0.1.

The question classification model is a word embedding model [9]. The model embeds questions and the question types into the same space, where the questions are represented by a bag of words vector based on the term counts. The TensorFlow embedding module provided by the Rasa framework is used to train the question classification model. The configuration settings are default provided by the Rasa framework. Based on the training set of questions with type labels, the question classification model is then trained.

The named entity recognition (NER) model is a conditional random field model [38] and the training is handled by the NER_CRF module provided by the Rasa framework. The model is chosen as it is suitable for training customized NER models.

3.3 Dictionary creation

The ontology lookup agent requires a dictionary which maps the terms to the IRIs. As a result, we utilized the natural language labels in the ontologies to build such a dictionary. We used SPARQL query to extract the rdfs:label and skos:altLabel value of all the nodes. However, due to the substantial size of the Wikidata, we used WDumper, which is a tool to create custom RDF dumps, to created a sub-graph of chemistry-related nodes of the Wikidata knowledge graph. The skos:altLabel value provides the synonyms for the nodes. For example, the skos:altLabel for methane contains words including "CH4", "marsh gas","methyl hydride","tetrahydridocarbon", and "tetrahydrogen monocarbide". We created separated dictionaries for classes, attributes, and entities. In total, 33,219 words are included in the dictionaries.

4 Agent description

This section will describe the implementation and working detail of the agents. The agents include the topic model agent, the question classification agent, named entity recognition agent, the ontology lookup agent, and the SPARQL construction agent. Figure 4 illustrates the interaction between the agents of the QA system.

4.1 Topic model agent

The topic model agent is implemented on top of the LDA topic model described in Section 3.2.2. The Gensim LdaModel module is used to carry out the function of identifying the question-ontology affiliation. When the agent receives the question, it first tokenizes the words of the question and use PorterStemmer to reduce the words to their stem forms. Then the agent uses the pre-loaded LDA model to identify the topic distribution of this word list. The topic distribution is represented by a list of topics, each assigned with a confidence score. The topic model agent will rank the topics according to their scores and pass the ranked topics to the agents in the next step. However, if the scores are equal (0.25), it means no topic is identified. Then the topic model agent will conclude that the question is out of the scope of the QA system since the question is not chemistry-related.

4.2 Question classification agent

The core of the question classification agent is the question classification model mentioned in Section 3.2.2. Rasa TensorFlow embedding module is used to load and run such a model. A set of SPARQL query templates are made in advance for each type of questions, where the specific IRIs are replaced by placeholders. The classification agent also loads the templates when initiated. The types and numbers of the components that each SPARQL query template expects are also encoded into the question classification agent. For example, "item_attribute_query" questions expect one attribute and one instance. By mapping the question received to the question space of the question classification model, the question classification agent produces one or more question types with confidence scores. Then the agent passes both the question types identified and expected types and numbers of the components to the next agent, which is the named entity recognition agent.

4.3 Named entity recognition agent

The named entity recognition is implemented on top of the NER model described in Section 3.2.2. The agent takes the question and the expected types and numbers of components provided by the Question classification agent as inputs. Based on the inputs, the named entity recognition agent extracts and classifies the key components, including attributes, instances, and classes, using the Rasa framework rasa_nlu module, which loads and runs the NER model. For chemical reactions, the agent also identifies any symbols, such as "==>" or "–>", that separate reactants and products and hence categorizes the species into reactants and products. Regular expression is applied for this function. The output of this agent is the key components and their type. For example, for question "What is the heat capacity of CH4", the output will be "heat capacity" as "attribute" and "CH4" as "instance". In addition, this agent verifies whether the types and numbers of the components in the results fulfills the expectation given by the question classification agent. If not, the agent iterates to the next question type in the input.

4.4 Ontology lookup agent

The ontology lookup agent converts the key components passed by the named entity recognition agent into IRIs. When the ontology lookup agent receives a question component, it will iterate through all the terms in the dictionary mentioned in Section 3.2.2 and calculate the string similarity between the terms and the question component. A Python library Fuzzywuzzy is used to calculate the string similarity based on their Levenshtein distance. According to the string similarity, the ontology lookup agent will return a ranked list of IRIs found in the dictionary. However, it is common that different ontologies contain the same attribute, instances, or classes. Therefore, this agent only select IRIs from the target ontology, which is already identified by the topic model agent. Then the key components extracted by the named entity recognition agent are converted into IRIs. The IRIs and their types, for example "OntoKin:Reaction" and "class" will be passed to the SPARQL construction agent.

4.5 SPARQL construction agent

The SPARQL construction agent receives the SPARQL query templates and IRIs with their types. Then the agent will create a list of possible combinations of SPARQL query templates and IRIs and rank the combinations based on the confidence scores of both the SPARQL query template given by the question classification agent and the scores of the IRIs given by the ontology lookup agent.

Due to the low speed of SPARQL queries, the SPARQL construction agent will send

Table 2: *Comparison of performance between this QA system and Google/Wolfram Alpha search engines.*

	not answered by search engines	answered by search engine
correct	114	224
incorrect	98	88

multiple SPARQL queries to the SPARQL endpoints at the same time via multi-threading. The valid result with the highest score will be returned.

5 Evaluation

For evaluation, we collected 100 evaluation questions from researchers in different fields in the Department of Chemical Engineering and Biotechnology in the University of Cambridge. In addition, 424 questions were manually created by other members in our research group knowing information stored in the knowledge graph.

The baseline for evaluation are selected to be the Google search engine and Wolfram alpha, as they are the most popular tools for retrieving data, including chemistry data. Also, to the best of our knowledge, there is no existing QA system specific to chemistry data. Due to the lack of methods to automatically evaluate the results returned for these questions, the 524 question-answer pairs are examined manually. The results are put into four categories based on whether they are correct and whether they can be answered by Google or Wolfram alpha.

The evaluation result is shown in Table 2. We also present some typical questions that system can answer in Table 3. In addition, by investigating the evaluation result in detail, we have noticed that the queries over Wikidata ontologies have significantly better performance than the queries over the JPS ontologies. By analysing the errors, we conclude that the better performance of Wikidata ontologies is due to their shallow structure, comparing to the deeper structure of JPS ontologies.

6 Conclusion

This paper presents the novel design and the development of a proof-of-concept QA system for chemical data within KGs, which can answer a number of relatively complex chemistry-related questions. Question classification and topic modelling agents are integrated into the system to increase the accuracy of constructing SPARQL queries. In the process of developing such a query interface, a number of state-of-the-art NLP tools, including Stanford CoreNLP, NLTK, OpenNLP, Genism, and the Rasa framework are evaluated based on their features. For the intent recognition and named entity recognition functions, the Rasa framework is selected. For simple NLP tasks, NLTK is chosen and

Table 3: *Typical questions that the QA system can answer.*

Question list
Computational Quantum Chemistry:
1. Show me the vibration frequency of H2O2.
2. Find the gaussian files for C8H14.
3. What is the symmetry number of C8H14?
4. What is the spin multiplicity of C8H14?
5. Electronic energy of C2H2O2.
6. Show the formal charge of C3H6.
7. What is the geometry type of C2H2O2?
Kinetic and Thermodynamics
1. What is the lennard jones well depth of C2H2O2?
2. Give the polarizability of C2H2O2.
3. What is the dipole moment of C2H2O2?
4. Show the rotational relaxation collision number of C2H2O2.
Reactions and Mechanisms
1. What reaction produces H2 + OH?
2. Is the reaction H + H2O == H2 + OH reversible?
3. What reaction has CH4 as a reactant?
4. What mechanism contains CH4 + OH?
Class query
1. List the chemical formula of alkanol with heat capacity less than 15.
2. Show the mass of aromatic hydrocarbons with mass less than 170.
3. Aromatic hydrocarbons with mass less than 170.
4. Chemical formula of alkanol with heat capacity less than 15.
Query by SMILES
1. What is the molecular weight of C1CCCCC1?
2. Show me the molecular model of CH2=CHCHO.
3. Show me the ionization energy of C1=CC=CC=C1.
4. What is the heat capacity of C1=CC=CC=C1?

for Genism is used for topic modelling. The paper also demonstrates a use case of implementing the query interface on top of the WAKG and the Wikidata KG. By evaluating the performance of the query interface implemented on top of the WAKG and the Wikidata

KG with a set of test questions, the query interface demonstrates its capability to answer a number of question of various type and structure. Those questions also demonstrate the advantage of the query interface in querying attributes of instances while relatively complex conditions are applied. Additionally, the integration of intent recognition agent and topic modelling agent is proved to be effective in improving the accuracy of SPARQL template mapping and ontology lookup, and hence the accuracy of SPARQL construction. Although a set of questions of certain types can be answered, the query interface is a proof of concept implementation for evaluating the architecture. Also, the implementation helps the better understanding of applying the natural language query interface over semantic chemistry data. To make the query interface a practical tool for the industry or academia, further expansion of the training question set and more formal evaluation of the interface is required.

Acknowledgements

This research was supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. X.Zhou acknowledges financial support provided CMCL Innovations. M. Kraft gratefully acknowledges the support of the Alexander von Humboldt Foundation.

A Question categories and component types

Question category	**Purpose**
select_mechanism_by_reaction	find mechanism by specifying reaction
select_reaction_by_species	find reactions by reactants and/or products
query_reaction_property	find properties such as reaction rate of a reaction
query_quantum_chemistry	query computational quantum properties
query_thermodynamic	query kinetic and thermodynamic properties
batch_restriction_query	query an attribute of all instances of of one class
item_attribute_query	query an attribute of a specific instance
batch_restriction_query_numerical	find instances which meet a numerical restriction
batch_restriction_query_attribute	query an attribute of a specific instance
Component type	**Example**
attribute	electric dipole moment
class	aromatic hydrocarbons
entity	CH4
attribute	electric dipole moment
comparison	lower than
numerical_value	200

References

[1] H. Al-Zubaide and A. A. Issa. Ontbot: Ontology based chatbot. In *International Symposium on Innovations in Information and Communications Technology*, pages 7–12, 2011. doi:10.1109/ISIICT.2011.6149594.

[2] Apache Software Foundation. OpenNLP Natural Language Processing Library, 2014. `http://opennlp.apache.org/` Last accessed January 25, 2021.

[3] D. K. Barry. *Web services, service-oriented architectures, and cloud computing the savvy manager's guide*. The Savvy Manager's Guides. Morgan Kaufman, 2nd ed. edition, 2013.

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. URL `http://www.jstor.org/stable/26059207`.

[5] C. Bizer, T. Heath, and T. Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011. doi:10.1007/978-981-10-3168-7.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *Advances in neural information processing systems*, pages 601–608, 2002. doi:10.1162/jmlr.2003.3.4-5.993.

[7] E. Cabrio, J. Cojan, F. Gandon, and A. Hallili. Querying multilingual dbpedia with qakis. In P. Cimiano, M. Fernández, V. Lopez, S. Schlobach, and J. Völker, editors, *The Semantic Web: ESWC 2013 Satellite Events*, pages 194–198, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[8] H. Chen, H. Ji, L. Sun, H. Wang, T. Qian, and T. Ruan. *Knowledge graph and semantic computing : semantic, knowledge, and linked big data*. Communications in computer and information science ; 650. Springer, 2016. doi:10.1007/978-981-10-3168-7.

[9] B. Chiu, G. Crichton, A. Korhonen, and S. Pyysalo. How to train good word embeddings for biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 166–174, Berlin, Germany, 2016. Association for Computational Linguistics. doi:10.18653/v1/W16-2922.

[10] R. Côté, F. Reisinger, L. Martens, H. Barsnes, J. A. Vizcaino, and H. Hermjakob. The Ontology Lookup Service: bigger and better. *Nucleic Acids Research*, 38:W155–W160, 2010. doi:10.1093/nar/gkq331.

[11] R. G. Côté, P. Jones, R. Apweiler, and H. Hermjakob. The ontology lookup service, a lightweight cross-platform tool for controlled vocabulary queries. *BMC Bioinformatics*, 7:97–97, 2006. doi:10.1186/1471-2105-7-97.

[12] R. G. Côté, P. Jones, L. Martens, R. Apweiler, and H. Hermjakob. Ontology Lookup Service: more data and better tools for controlled vocabulary queries. *Nucleic acids research*, 36:W372–W376, 2008. doi:10.1093/nar/gkn252.

[13] A. Devanand, G. Karmakar, N. Krdzavac, R. Rigo-Mariani, Y. F. Eddy, I. A. Karimi, and M. Kraft. OntoPowSys: A Power System Ontology for Cross Domain Interactions in an Eco Industrial Park. *Energy and AI*, page 100008, 2020. doi:10.1016/j.egyai.2020.100008.

[14] D. Diefenbach, K. Singh, and P. Maret. WDAqua-core0: A question answering component for the research community. In M. Dragoni, M. Solanki, and E. Blomqvist, editors, *Semantic Web Challenges*, pages 84–89, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-69146-6_8.

[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, 2019. doi:10.1016/j.compchemeng.2019.106586.

[16] C. Eschenbach. *Formal Ontology in Information Systems*. Frontiers in artificial intelligence and applications Formal ontology in information systems. IOS Press, Amsterdam, 2008.

[17] 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.

[18] F. Farazi, N. B. Krdzavac, J. Akroyd, S. Mosbach, D. Nurkowski, A. Menon, and M. Kraft. Linking reaction mechanisms and quantum chemistry: An ontological approach. *Computers & Chemical Engineering*, 137:106813, 2020. doi:10.1016/j.compchemeng.2020.106813.

[19] F. Farazi, M. Salamanca, S. Mosbach, J. Akroyd, A. Eibeck, L. K. Aditya, A. Chadzynski, K. Pan, X. Zhou, S. Zhang, M. Q. Lim, and M. Kraft. Knowledge Graph Approach to Combustion Chemistry and Interoperability. *ACS Omega*, 5(29):18342–18348, 2020. doi:10.1021/acsomega.0c02055.

[20] D. Greenwood, P. Buhler, and A. Reitbauer. Web Service Discovery and Composition using the Web Service Integration Gateway. In *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, pages 789–790, 2005. doi:10.1109/EEE.2005.141.

[21] N. Griffiths and K.-M. Chao. *Agent-based service-oriented computing*. Springer, 2010. doi:10.1007/978-1-84996-041-0.

[22] T. Gruber. *Ontology*, pages 1963–1965. Springer US, Boston, MA, 2009. doi:10.1007/978-0-387-39940-9_1318.

[23] E. Kaufmann, A. Bernstein, and R. Zumstein. Querix: A natural language interface to query ontologies based on clarification dialogs. In *5th International Semantic Web Conference (ISWC 2006)*, pages 980–981, 2006.

[24] 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.

[25] A. Lambert and F. Boons. Eco-industrial parks: stimulating sustainable development in mixed industrial parks. *Technovation*, 22(8):471 – 484, 2002. doi:10.1016/S0166-4972(01)00040-2.

[26] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann. Industry 4.0. *Business & Information Systems Engineering*, 6(4):239–242, 2014. doi:10.1007/s12599-014-0334-4.

[27] B. Li, X. Tang, and J. Lv. The Research and Implematation of Services Discovery Agent in Web Services Composition Framework. In *2005 International Conference on Machine Learning and Cybernetics*, volume 1, pages 78–84, 2005. doi:10.1109/ICMLC.2005.1526923.

[28] E. Loper and S. Bird. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, page 63–70, USA, 2002. Association for Computational Linguistics. doi:10.3115/1118108.1118117.

[29] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014. doi:10.3115/v1/P14-5010.

[30] A. Menon, N. B. Krdzavac, and M. Kraft. From database to knowledge graph — using data in chemistry. *Current Opinion in Chemical Engineering*, 26:33 – 37, 2019. doi:
10.1016/j.coche.2019.08.004.

[31] V. Mishra and N. Khilwani. QUASE: AN Ontology-Based Domain Specific Natural Language Question Answering System. *International Journal of Recent Technology and Engineering (IJRTE)* , 2019. doi:10.35940/ijrte.d6773.118419.

[32] N. Ngomo. 9th challenge on question answering over linked data (QALD-9). *language*, 7(1), 2018.

[33] S. Peng, R. You, H. Wang, C. Zhai, H. Mamitsuka, and S. Zhu. DeepMeSH: deep semantic representation for improving large-scale MeSH indexing. *Bioinformatics*, 32(12):i70–i79, 06 2016. doi:10.1093/bioinformatics/btw294.

[34] Rasa Technologies. Open source conversational AI | Rasa - Rasa, 2020. `https://rasa.com/` Last accessed January 25, 2021.

[35] D. Saha, A. Floratou, K. Sankaranarayanan, U. F. Minhas, A. R. Mittal, and F. Özcan. Athena: an ontology-driven system for natural language querying over relational data stores. *Proceedings of the VLDB Endowment*, 9(12):1209–1220, 2016.

[36] F. Schulze, R. Schüler, T. Draeger, D. Dummer, A. Ernst, P. Flemming, C. Perscheid, and M. Neves. HPI question answering system in BioASQ 2016. In *Proceedings of the Fourth BioASQ workshop*, pages 38–44, 2016.

[37] J. Shamma. *Cooperative control of distributed multi-agent systems*. John Wiley & Sons, 2008.

[38] C. Sutton and A. McCallum. An introduction to conditional random fields. *Machine Learning*, 4(4):267–373, 2011. doi:10.1561/2200000013.

[39] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan. Automated discovery, interaction and composition of semantic web services. *Journal of Web Semantics*, 1 (1):27 – 46, 2003. doi:10.1016/j.websem.2003.07.002.

[40] V. Tablan, D. Damljanovic, and K. Bontcheva. A natural language query interface to structured information. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *The Semantic Web: Research and Applications*, pages 361–375, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[41] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, Y. Almirantis, J. Pavlopoulos, N. Baskiotis, P. Gallinari, T. Artieres, A. Ngonga, N. Heino, E. Gaussier, L. Barrio-Alvers, M. Schroeder, I. Androutsopoulos, and G. Paliouras. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138, 2015. doi:10.1186/s12859-015-0564-6.

[42] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M. R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, et al. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138, 2015. doi:10.1186/s12859-015-0564-6.

[43] C. Unger, A. Freitas, and P. Cimiano. An introduction to question answering over linked data. In *Reasoning Web International Summer School*, pages 100–140. Springer, 2014. doi:10.1007/978-3-319-10587-1_2.

[44] C. Wang, M. Xiong, Q. Zhou, and Y. Yu. Panto: A portable natural language interface to ontologies. In E. Franconi, M. Kifer, and W. May, editors, *The Semantic Web: Research and Applications*, pages 473–487, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[45] World Wide Web Consortium's RDF Data Access Working Group. SPARQL Query Language for RDF, 2008. https://www.w3.org/TR/rdf-sparql-query/Last accessed January 25, 2021.

[46] L. Zhou, C. Zhang, I. A. Karimi, and M. Kraft. An ontology framework towards decentralized information management for eco-industrial parks. *Computers & Chemical Engineering*, 118:49 – 63, 2018. doi:https://doi.org/10.1016/j.compchemeng.2018.07.010.

[47] 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. doi:10.1016/j.compchemeng.2019.106577.

[48] X. Zhou, M. Q. Lim, and M. Kraft. A smart contract-based agent marketplace for the J-Park Simulator - a knowledge graph for the process industry. *Computers & Chemical Engineering*, 139:106896, 2020. doi:10.1016/j.compchemeng.2020.106896.

[49] L. Zou, R. Huang, H. Wang, J. X. Yu, W. He, and D. Zhao. Natural Language Question Answering over RDF: A Graph Data Driven Approach. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, page 313–324, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2588555.2610525.

[50] R. Řehůřek and P. Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP*, pages 45–50, 05 2010. doi:10.13140/2.1.2393.1847.