Preprint Cambridge Centre for Computational Chemical Engineering ISSN 1473 – 4273

Question Answering System for Chemistry – a semantic agent extension

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

released: December 16, 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
Cambridge
CB3 0AX
United Kingdom

Preprint No. 286

UNIVERSITY OF CAMBRIDGE

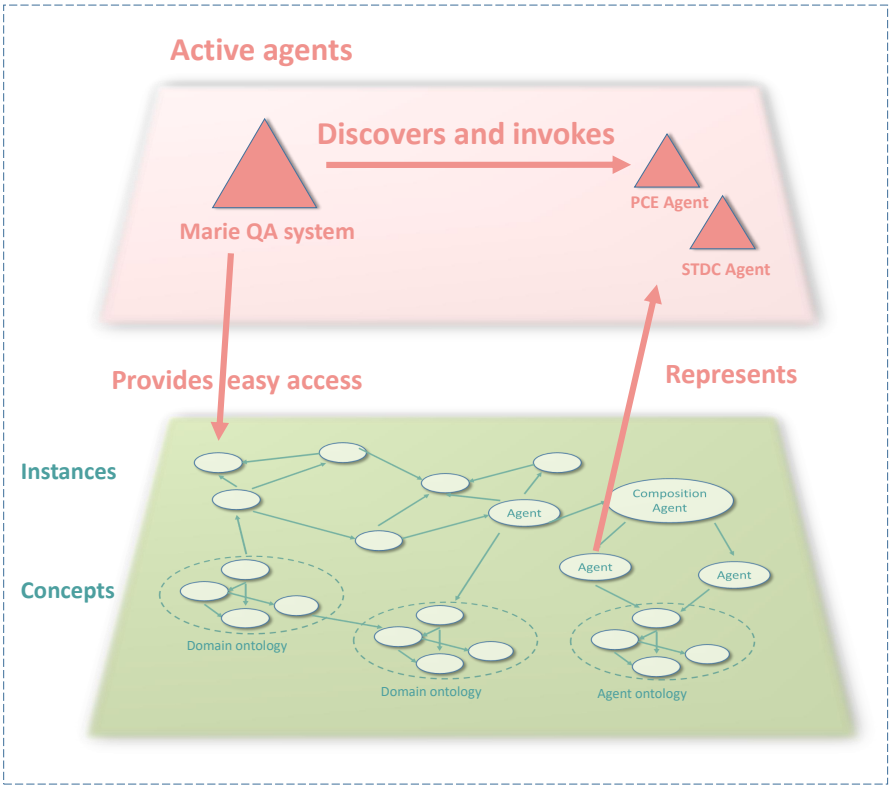
Keywords: Question answering, Semantic Agent, Knowledge Graph

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 introduces an extension of a previously developed question answering (QA) system for chemistry, operating on a knowledge graph (KG) called Marie. This extension enables the automatic invocation of semantic agents to answer questions when static data is absent from the KG. The agents are semantically described using the agent ontology, OntoAgent, to enable automated agent discovery and invocation. The natural language processing (NLP) models of the QA system need to be trained in order to interpret questions to be answered by new agents. For this purpose, we extend OntoAgent so that it becomes possible to automatically create training material for the NLP models. We evaluate the extended QA system with two example chemistry-related agents and an evaluation question set. The evaluation result shows that the extension allows the QA system to discover the suitable agent and to invoke the agent by automatically constructing requests from the semantic agent description, thereby increasing the range of questions the QA system can answer.



Highlights

- The Marie QA system is extended to invoke semantic agents to answer questions.
- Extended OntoAgent ontology to provide information for the creation of training material for NLP models.
- Implemented a mechanism to automatically include semantic agents into the QA system.

Contents

1 Introduction

Knowledge graphs and the Semantic Web technologies are rapidly playing larger roles in data storage, sharing, and manipulation across a variety of domains, including the field of chemistry. A knowledge graph is a form of data representation, made up by collections of descriptions of entities: events, concepts, or objects in the physical world, where the entities are interconnected with each other via relations. A collection of entities of a certain domain is referred to as an ontology [30]. A pair of connected entities and the connecting relation form a triple, which is the basic unit for representing data in a KG. In a KG, entities and relations are represented by Internationalized Resource Identifiers (IRIs) [11], which serve as both unique identifiers and locators. With the interconnected structure, a common data ground is established for previously isolated datasets, enabling cross-domain information retrieval. Major existing KGs include the Wikidata KG [31], the DBpedia KG [20], and the Google KG [16].

Semantic queries are the main tools for accessing data in KGs, one prominent example is the SPARQL query language [33]. By specifying triple patterns and other criteria, a SPARQL query can perform complex data retrival on top of a KG. However, to formulate a SPARQL query, one requires not only the knowledge on the syntax of the SPARQL language but also the knowledge on how the specific data is represented in the KG. For example, to retrieve the molecular mass value of a species, one must know the IRIs of the species and the relation between the species and its molecular mass value. In addition, in many cases, there can be one or more nodes placed in between the entities and their attribute and therefore a more complex triple pattern is required for the query. As a result, the barrier for general users to accessing data in a KG is high.

Establishing Knowledge Base Question Answering (KBQA) systems [8, 35] is an effective solution to the barrier of accessing data from KGs. In fields that have rich but complex data resources, for example, chemistry, the role of QA systems is even more important. A typical KBQA system translates natural language query systems into formal representations that machine can understand, including SPARQL queries, logic forms, and subgraph embedding [4]. As a result, users can perform complex and precise data retrieval on top of KGs without the knowledge of query language syntax and the structure of data. In addition, a QA system, different from a search engine, aims to provide direct answers for a query instead of a set of relevant information [5].

Marie [39] is a QA system for chemistry, which operates on top of a KG. It is, to the best of our knowledge, the only existing QA system for chemistry so far. The Marie QA system operates on top of the chemistry ontologies in the World Avatar (TWA) KG [12]. TWA KG is a dynamic knowledge graph (dKG) that follows Linked Data principles and integrates ontologies from various domains. The ontologies include OntoCAPE [22] for process engineering, OntoEIP [36] for eco-industrial park (EIP), OntoPowSys [9] for power systems, and OntoCityGML [12] for city and landscape models. For chemistry, TWA KG includes OntoCompChem [19] for quantum chemistry calculations, OntoSpecies [15] for chemical species, and OntoKin [14] for chemical kinetic reaction mechanisms. It also integrates the OntoAgent [37] ontology to describe semantic agents, which update the content and the structure of the KG over time.

Marie is a template-based QA system [7], which uses natural language processing (NLP) tools to interpret questions, converts question components into their semantic representations (*i.e.* IRIs), and constructs SPARQL queries by filling query templates with IRIs. In the Marie system, the NLP tools include a question classification model [34], which assign query templates to questions, and a named entity recognition model [25], which extracts and labels key components in the questions.

The NLP models are trained via supervised learning, where the training material are questions automatically generated and annotated from chemistry ontologies in TWA KG. As a result, the NLP models Marie employs are domain-specific for chemistry and hence so is the Marie QA system.

The Marie QA system has been evaluated with a set of chemistry questions. The system is also compared to two state-of-art KBQA systems, QAnswer [10] and Platypus [26], and two widely used search engines, the Google search engine and the Wolfram Alpha engine. In the aspect of the percentage of correct answers returned, the Marie system outperforms the two KBQA systems in all 11 types of evaluation questions and outperforms the two search engines in 8 out of the 11 types of evaluation questions.

However, currently, the Marie QA system can only access the static part of TWA KG, and the incompleteness of the static part of TWA KG hinders the Marie QA system from answering more questions. In addition, some real-time data, for example, sensor data, needs to be retrieved from more dynamic data sources than the static components of the KG. Also, due to the substantial number of chemical species and reactions in the KG, it is not as efficient to calculate all of their properties, for example, thermodynamic properties, and store the results in the KG as to make some calculations on-demand.

The Wolfram Alpha engine [18] is an example of a QA system integrating dynamic data sources to answer questions. The Wolfram Alpha engine invokes functions according to the user's questions. For example, Wolfram Alpha invokes the "Plus" function if "what is 1 + 1" is asked. This feature makes the Wolfram Alpha engine one of the most versatile QA systems [32].

As a result, to further extend the range of questions that the Marie QA system can answer, one solution is to make the dynamic components of the KG accessible to the Marie QA system. In TWA KG, on top of the Marie QA system, the dynamic parts are semantic agents. A typical semantic agent is a web service accessible through requests and has semantic descriptions of its function, request format, quality of service (QoS), *etc*. With their functions described semantically, semantic agents that fit specific tasks can be automatically discovered via semantic queries. Also, the semantic description specifies the request format of the agents, so that valid requests to the agents can be automatically constructed. These features make semantic agents ideal for producing information dynamically in a knowledge graph.

Therefore, the purpose of this paper is to propose an extension that enables access to semantic agents in TWA KG for the Marie QA system. This extension allows the QA system to locate and invoke semantic agents to answer questions, when the QA system fails to answer the question with the static data in the KG. In addition, this paper introduces an extension of the agent ontology OntoAgent [37] to enable automated training of the NLP models in Marie to interpret new questions enabled by the semantic agents.

The rest of this paper is organised as follows. Section 2 introduces the existing Marie QA system together with the underlying KG. Section 3 discusses the implementation of the extension on the Marie QA system in detail. Section 4 analyses the evaluation results of the extended QA system and section 5 provides an error analysis. Section 6 concludes this paper. In addition, section 7 provides information about data and software availability of the system implemented.

2 Background

In this section, we will introduce the World Avatar (TWA) Knowledge graph together with its semantic agent system, and the existing Marie QA system, on top of which the extension is implemented.

2.1 The World Avatar KG

The World Avatar Knowledge Graph (TWA) [12] is a dynamic knowledge graph (dKG), which integrates multiple ontologies from different domains. Knowledge graphs, for example TWA KG, utilise Semantic Web technology [1, 17] to represent information in a machine-readable way, where concepts, entities, and the relations between them are formally defined and connected. Through the links between instances, it is convenient to retrieve and navigate through related data within a KG. In addition, by applying the Linked Data principles [2] and linking knowledge from different domains, KGs interconnect previously isolated datasets. For example, in TWA KG, the instance of a power plant is connected to a city instance via an "isLocatedIn" location, while another connection between the instance of natural gas can be connected to this power plant by the "hasPrimarilyFuel" relation. Further, the physical and chemical properties of natural gas, such as its molecular weight, can be also connected to the instance of natural gas. As a result, a KG provides a common ground for accessing data from different domains or multiple levels [24] and guarantees that related data can be easily queried.

TWA integrates geospatial data [6], datasets for quantum calculation [19], datasets for chemical kinetic reaction mechanisms [14], datasets for chemical species [15], power systems [9], and descriptions for semantic agents [37].

2.2 Marie QA system

The existing Marie QA system is a template-based QA system for chemistry [39]. It uses NLP models to interpret questions and construct SPARQL queries from SPARQL query templates to retrieve information from the KG. The NLP models include a topic model, a question classification model, and a named entity recognition (NER) model. The topic model is a Latent Dirichlet allocation (LDA) [3] topic model, which derives abstract topics from ontologies and identifies the affiliation between a question and an ontology. The question classification model is based on text-embedding on top of the StarSpace embedding model [34]. This model assigns the suitable SPARQL query template for questions.

The NER model is a Conditional Random Field (CRF) [29] model, which extracts and labels the key components in a question. For example, in the question "What is the heat capacity of benzene", the NER model extracts "heat capacity" and labels it as "attribute" and extracts "benzene" and labels it as "species". All the models are specifically customized for the chemistry domain as they are trained on top of chemistry-related training material.

One of the highlights of the Marie QA system is an automated mechanism to generate training material for the NLP models. This mechanism leverages the rich taxonomy and hierarchy of information in the ontologies to create and label training questions. As a result, the Marie QA system is able to integrate new ontologies easily.

2.3 Semantic agent system

Agents make up the dynamic part of the knowledge graph. In TWA KG, agents are web services deployed in a distributed way and accessible via HTTP requests, where their semantic descriptions are stored in the KG.

In TWA KG, an agent is semantically described by OntoAgent [37]. A typical OntoAgent description of an agent contains the detailed description of its input/output (I/O) signatures, its URL, its Quality of Service (QoS) [38], and the reference to its quality, which is stored in a public blockchain. The I/O signatures are represented by concepts from ontologies. For example, one of the inputs for a weather agent is city. In the OntoAgent description, the signature of this input is represented by the IRI of the semantic concept "https://dbpedia.org/ontology/City" from the DBpedia KG. With the semantic description, an agent composition framework is implemented, which enables the automated discovery, composition, and invocation of the agents.

The TWA KG contains a wide range of agents. In the chemistry domain, the agents include the thermodynamic data agent (STDC agent) and power conversion efficiency agent (PCE agent).

2.4 Example semantic agents

In this section, we will introduce the two example semantic agents integrated to the Marie QA system.

2.4.1 PCE agent

The purpose of this agent is to compute the power conversion efficiency (PCE) of an organic solar cell [28] given the SMILES string of the donor molecule of the cell. It is assumed that the solar cell is of hetero-junction type and its acceptor molecule is fullerene-based. Internally the agent invokes a support vector regression (SVR) machine learning model [27] optimised and trained on the HOPV15 dataset [21]. The HOPV15 dataset consists of 350 experimentally measured PCEs on variety of different solar cell architectures. A detailed description of the HOPV15 dataset and how the SVR model was created

is provided in our previous work [13], thus only a short explanation is given below.

The created SVR model transforms the input SMILES strings of a donor molecule into the FS-bit long Morgan fingerprints for a given radius FR, where both FS and FR are the hyperparameters. The model passes the computed fingerprints into the radial basis function kernel (RBF) and calculates the Tanimoto distance between them:

$$K(x,x') = \exp\left[\gamma(1 - T(x,x'))\right], \tag{1}$$

where $K(x,x')$ is the RBF kernel, x and x' are the two fingerprint bit vectors of size FS and $T(x,x')$ denotes the Tanimoto similarity index defined as:

$$T(x,x') = \frac{\sum_{i=1}^{\mathrm{FS}} (x_i \wedge x'_i)}{\sum_{i=1}^{\mathrm{FS}} (x_i \vee x'_i)}. \tag{2}$$

The SVR model parameters were tuned in a 5-fold cross-validation loop and then their optimal values were used in the model re-training step to produce the final model, which then predicts the PCE of the cell. The created model was then used to build the PCE web-agent interface. For the OntoAgent description of this agent, the data type of the input it receives is the species concept from the OntoSpecies ontology, "ontospecies:Species".

2.4.2 Thermodynamic data agent

The purpose of this agent is to calculate the gas-phase thermodynamic properties of a chemical species as a function of temperature T and pressure P. This agent is described in more detail in a previous paper [24], but for ease of reference, a brief summary is included here. The calculated properties are the species molar entropy S, enthalpy H, internal energy U, Gibbs energy G change and heat capacities at constant volume C_v and pressure C_p. Internally, the agent uses standard statistical thermodynamics equations to derive the results from the species molecular properties [23]:

$$S = Nk_{\mathrm{B}} \left[\frac{\partial (T \ln q)}{\partial T} - \ln N + 1 \right] \tag{3}$$

$$C_{\mathrm{v}} = Nk_{\mathrm{B}} T \frac{\partial^2 (T \ln q)}{\partial^2 T} \tag{4}$$

$$C_{\mathrm{p}} = C_{\mathrm{v}} + Nk_{\mathrm{B}} \tag{5}$$

$$\Delta U = U(T) - U(0) = Nk_{\mathrm{B}} T^2 \frac{\partial \ln q}{\partial T} \tag{6}$$

$$\Delta H = H(T) - H(0) = \Delta U(T) + Nk_{\mathrm{B}} T \tag{7}$$

$$\Delta G = \Delta H - TS \tag{8}$$

where, N is the Avogadro's number, k_{B} is the Boltzmann constant and q is the molecular partition function defined in equation (9). Note that equations (6), (7) and (8) provide only

energy differences as opposed to absolute values. In order to obtain meaningful absolute values, a known reference state for one of the energies needs to be provided. The standard enthalpy of formation at 298.15 K has been selected as the reference state for enthalpy, which in turn, is used to reference internal and Gibbs energies.

The molecular partition function, appearing in the thermodynamic properties equations above, is defined as follows:

$$q = q_{\mathrm{t}} q_{\mathrm{r}} q_{\mathrm{v}} q_{\mathrm{e}}, \tag{9}$$

whose components are the translational (q_{t}), rotational (q_{r}), vibrational (q_{v}) and electronic (q_{e}) partition functions respectively. These components are derived from the standard statistical mechanics expressions under the rigid-rotor-harmonic-oscillator (RRHO) approximation:

$$q_{\mathrm{t}} = \left(\frac{m k_{\mathrm{B}} T}{2\pi\hbar^2} \right)^{3/2} \frac{k_{\mathrm{B}} T}{P} \tag{10}$$

$$q_{\mathrm{v}} = \prod_{i=1}^{N_v} \frac{\exp\left(-\frac{-2\pi\hbar\nu_i}{k_{\mathrm{B}} T} \right)}{1 - \exp\left(-\frac{-2\pi\hbar\nu_i}{k_{\mathrm{B}} T} \right)} \tag{11}$$

$$q_{\mathrm{r}} = \begin{cases} \dfrac{2TIk_{\mathrm{B}}}{\sigma_r \hbar^2} & \text{linear molecule} \\ \dfrac{\left(8\pi I_x I_y I_z\right)^{\frac{1}{2}} \left(k_{\mathrm{B}} T\right)^{\frac{3}{2}}}{\sigma_r \hbar^3} & \text{nonlinear molecule} \end{cases} \tag{12}$$

$$q_{\mathrm{e}} \approx g_0^E, \tag{13}$$

where m is the mass of a chemical species, g_0^E is the degeneracy of the ground electronic state, $\hbar$ is the reduced Planck's constant, N_v is the number of vibrational modes and ν_i is the i^{th} vibrational mode value, σ_r is the rotational symmetry number and I or I_x, I_y, I_z are the rotational moments of inertia around specified axes.

In order for the agent to calculate the thermodynamic properties of a chemical species, it must receive a species IRI via an HTTP request. Temperature and pressure are the optional inputs and, if not provided, default values of 298.15 K and 1 atm are used. The agent then queries the Knowledge Graph for the species molecular data and the energy reference point and runs the thermodynamic calculations. The OntoAgent instance describing this agent also uses the concept "ontospecies:Species" from the OntoSpecies ontology for the data type of the input of the agent.

3 Extension of the QA system

This extension of the Marie QA system is implemented in addition to the existing components, where the existing components remain the same. The extension includes the training of a new question classification model, the extension on the OntoAgent ontology, the training of a new named entity recognition (NER) model, and the implementation of a

mechanism for agent query and request construction. Figure 1 demonstrates the workflow of the extended module of the QA system.

When the attempt of answering a question using the existing Marie system fails, the Marie system will pass the question to this extended module.

When a question is passed to the extended module for invoking agents, the named entity recognition agent, which is built on top of the NER model trained, will extract and label the key components in the question. For example, in the question “What is the heat capacity of benzene at 100 K”, the term “heat capacity” will be labelled as “attribute”, “benzene” as “species”, and "100 K” as “qualifier”.

Then the question will be fed to the agent lookup agent, which is built on top of the question classification model. Given the question, the agent lookup agent will provide the IRI of the most suitable agent to answer this question, by identifying the question-agent affiliation. Given the agent IRI, the agent query agent then retrieves the information about how the agent should be invoked.

Lastly, the extracted and labelled components and the information about agent invocation will be passed to the HTTP request construction agent, which constructs the HTTP request to invoke the agent and invoke the agent to answer the question. For some inputs, terms extracted from the question will be converted into IRIs via the ontology lookup agent.

The rest of this section will introduce the aforementioned components in detail.

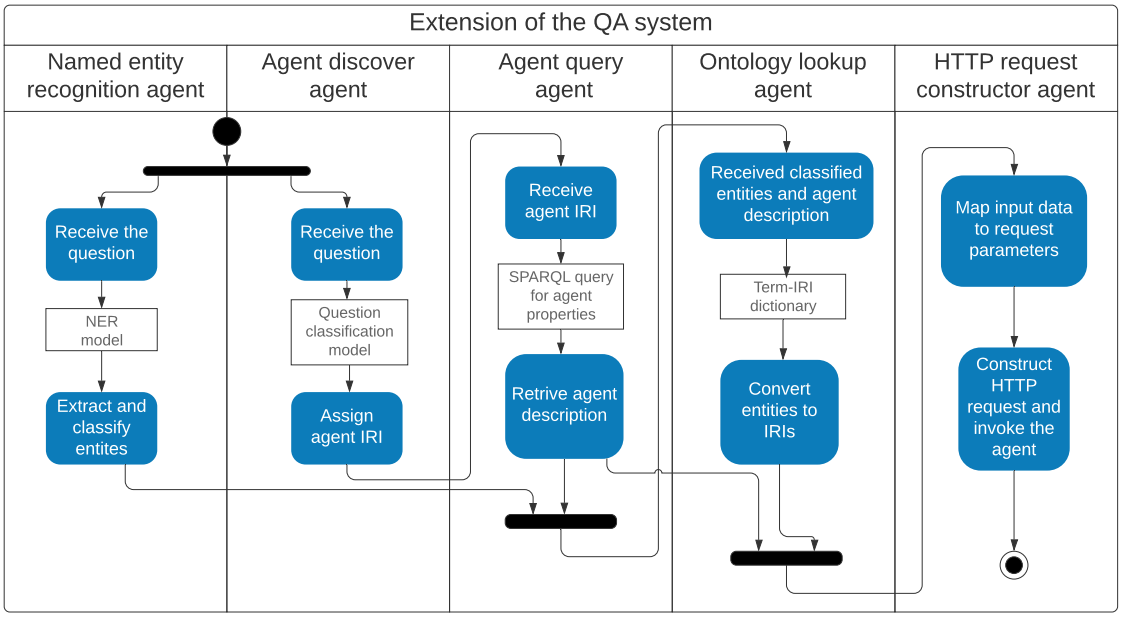


Figure 1: *The work flow of the extension of the QA system.*

3.1 Extension of OntoAgent

In this extension, the semantic agent description based on OntoAgent has a new role: to provide information for automated creation of training questions for the NER model and

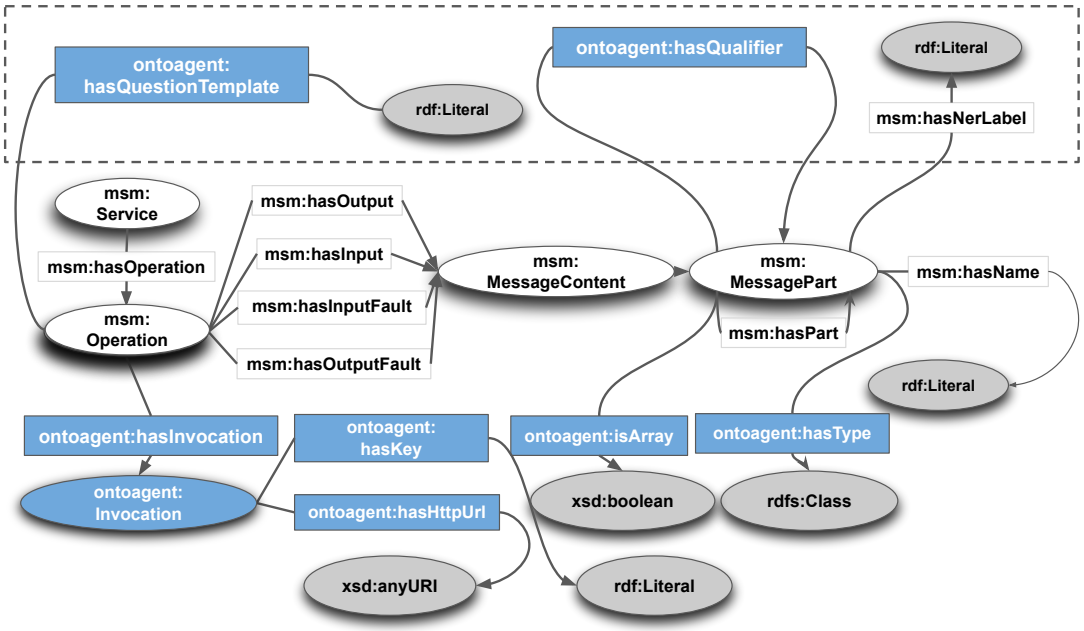


Figure 2: *The schema of the extended OntoAgent ontology: within the dotted box are the two newly added properties.*

the question classification model.

As a result, the OntoAgent T-Box (the schema of the ontology) is extended to provide vocabulary to describe the new extra information. The extension is shown in figure 2.

Firstly, a property named "hasQualifier" is added, which connects the MessagePart instance that represents an output of the agent and their qualifiers, which is also a MessagePart instance. This property is added because many agents require qualifiers of outputs to further refine the outputs. For example, temperature is the qualifier for enthalpy of a species and pressure is the qualifier for the boiling point of a species.

Secondly, a property named "hasQuestionTemplate" is created. It serves as the predicate between the Operation instance and a question template. This question template represents the expected structure of the questions to be answered by this agent. For example, the question template of the STDC agent can be "<enthalpy> of <species> at <temperature>" or "<species>'s <enthalpy> at <temperature>". A question generated by this template may be "What is the enthalpy of benzene at 100K?".

Appendix A shows the example of the description of the STDC agent and elaborates how the new properties in OntoAgent are used.

3.2 Creation of training material for NLP models

When a new agent is included in the QA system, the range of the questions that the QA system can answer will be expanded. As a result, the NLP models of the system need to be retrained to interpret the new questions that the QA system can now answer. Therefore, an automated mechanism is implemented to generate the training material on top of the

OntoAgent description of agents. The workflow of the automated creation of the training material is described in Algorithm 1.

For the NER model, on which the named entity recognition agent is implemented, its training requires a set of questions where their key components are highlighted and labelled. For example, in question "What is the heat capacity of benzene", "heat capacity" and "benzene" should be highlighted and labelled as "attribute" and "species".

The training of the question classification, which is used to build the agent discovery agent, requires questions labelled with the IRI of the agents that should be used to answer them. For example, the question "What is the heat capacity of benzene" is labelled with the IRI of the STDC agent, "OntoAgent:STDC_Agent".

The training material for the two NLP models are generated by the same script and stored in the same document, leveraging the information in the KG, especially the information from the agent descriptions.

The script will first query the property "hasQuestionTemplate" of the agent to retrieve the question template of this agent. Then the script will iterate through the inputs, outputs, and the qualifier of outputs of the agent and retrieve their information. For outputs, the scripts retrieves their "hasType" values. For example, the "hasType" value of the output of the STDC agent is the IRI "ontokin:Enthalpy". As a result, the script can determine from this query result that the STDC agent outputs enthalpy. Then the script will retrieve the label and alternative labels of "ontokin:Enthalpy", which are "enthalpy" and "molar enthalpy".

This output also has a qualifier, of which the "hasType" value is "ontokin:Temperature". The script will query this class and find its units and numerical ranges. According to the units and numerical range, the scripts generates a set of possible values with their units, *e.g.* 300K and 1000K. The input of the STDC agent is "species", which is represented by "ontospecies:Species". In questions, for example, "show the enthalpy of benzene at 100K", the QA system will look up the ontospecies:Species instance that matches benzene.

As a result, the script will query the KG to retrieve a set of the labels, formula, and other identifiers of instances of "ontospecies:Species", such as "benzene", "C6H6", "c1ccccc1". Then, the script will query the agent description again to get the "hasNerLabel" label of the input or output, for example, the "hasNerLabel" label for "benzene" is "species". This label serves as the annotation label for the training material of the NER model. With the information about the I/O signatures retrieved from the KG, the script will fill the information into the question templates. An example of the generated question for STDC agent is "show me the [enthalpy](attribute) of [c1ccccc1](species) at [100K](temperature)". The collection of these labelled questions are stored in one document for training both the NER model and the question classification model.

3.3 Named entity recognition agent

The Named entity recognition (NER) agent extracts and labels the key components in the question. This agent is built on top of the NER model, which is a a conditional random

Algorithm 1 Automated creation of training material

```
1: Q_training ← ∅
2: for All agent from KG do
3:     for All template from agent[hasTemplate] do
4:         for All input from agent[hasInputs] do
5:             if input[hasType] is Class then
6:                 candidates = instances ∈ input[hasType] in the KG
7:                 input_content = random_sample(candidates)[label]
8:             end if
9:             if input[hasType] is Numerical value then
10:                input_content = Random_number + input[hasType][unit][label]
11:            end if
12:            for All output from agent[hasOutputs] do
13:                output_content = output[hasType][label]
14:                for All q from output[hasQualifer] do
15:                    if q[hasType] is Class then
16:                        candidates = instances ∈ q[hasType]
17:                        q_content = random_sample(candidates)[label]
18:                    end if
19:                    if q[hasType] is Numerical value then
20:                        q_content = Random_number + q[hasType][unit][label]
21:                    end if
22:                    question = fill template with input_content, output_content, and
   q_content
23:                end for
24:            end for
25:        end for
26:    end for
27:    Q_training.append(question)
28: end for
29:
```

field (CRF) model [29] trained on questions created from the agent descriptions. The NER agent takes a question as the input and produces a list of key-value pairs, where the keys are the key components extracted from the question and the values are their labels. For example, for the question "What is the heat capacity of benzene", the output is "heat capacity" paired with "attribute" and "benzene" paired with "species". The result of this agent will be passed to the ontology lookup agent and HTTP request construction agent.

Since the NER model in the extension is trained only on material generated from agent descriptions, this NER agent is restricted to interpret questions that can be answered by the collection of semantic agents integrated to the QA system. In addition, the NER model will be automatically retrained once new semantic agents are included.

3.4 Agent discovery agent

Given the question, the agent discovery agent predicts the most likely agent that could answer this question. The agent discovery agent is built on the question classification model trained. As mentioned before, the question classification model is trained based on pairs of questions, automatically generated from agent descriptions, and agent IRIs. The text-embedding and training of the question classification model are based on the StarSpace [34] model, which is a general-purpose neural model for entity embedding learning. In this case, two separate sets of questions are generated for the two agents and each question is labelled with the URI of the agent which they are generated from.

As a result, given a question, the question classification model returns the IRI of the agent that most likely can answer this question. For example, the result returned from the agent discovery agent for the question "What is the heat capacity of CO2 at 100 K?" is the IRI of the STDC agent "OntoAgent:STDC_AGENT". The result of this agent will be passed to the agent query agent. The underlying model of this agent is also automatically retrained once a new semantic agent is introduced to the system.

3.5 Agent query agent

With the agent IRI as the input, the Agent query agent use SPARQL queries to query the knowledge graph and retrieves the semantic description of the agent, particularly the description of the I/O signature of the agent and request format of the agent. In this case, the description of the two selected agents include the "hasNerLabel", "hasName", and "hasType" properties of the inputs of the agents, and the HTTP URL of the agents. The result will be passed to both the ontology lookup agent and the HTTP request construction agent.

3.6 Ontology lookup agent

Some semantic agents receive IRIs as inputs. For example, the STDC agent and the PCE agent, as defined in their semantic agent description, accept the IRI of the species instead of other forms of the species. As a result, we implemented a mechanism to convert terms

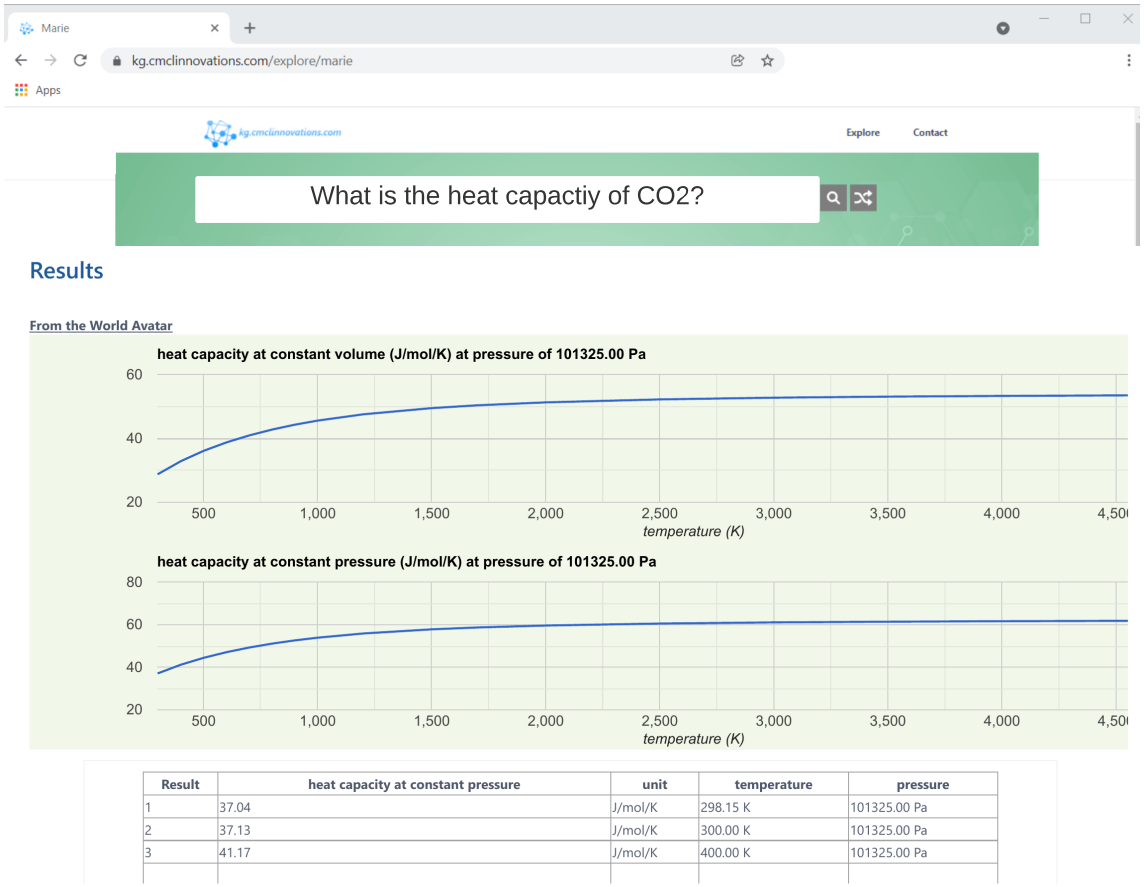
extracted and labelled by the named entity recognition agent into IRIs. The ontology lookup agent firstly identifies the inputs that need to be transformed into IRIs based on the agent description. Secondly, the ontology lookup agent looks up the IRI of the term to convert in a dictionary [39] from the existing Marie QA system. The keys of the dictionary are terms, for example "CO2" and the values are the respective IRIs, for example "ontospecies:CO2". To look up the IRI of a term, the ontology lookup agent calculates the string similarity between the term and each key of the dictionary and finds the key with the highest string similarity. Then the agent retrieves the IRI of this key. The resulting IRIs will be passed to the HTTP request construction agent.

3.7 HTTP request construction agent

To invoke the agents, it is necessary to construct requests based on the inputs collected from the interpretation of the question. For the two example agents, they are accessible via HTTP requests and receive the inputs through key-value pairs encoded in the HTTP requests. With the agent description retrieved from the knowledge graph and the inputs interpreted from the question, Marie makes a mapping between the entities extracted and labelled by the NER model and inputs that the invocation requires. The HTTP request construction script iterates through the inputs and qualifiers in the agent description and find the entities extracted and labelled by the NER model with a label that matches the "hasNerLabel" property of the input. A mapping between the name of the key, which is described by "hasName" property, and the entity value in the question is created. For example, "temperature" and "100K". The algorithm is described in Algorithm 2. With the mapping between the keys and the input values, the key-value pairs are generated.

However, before encoding the key-value pairs into the URL, the species, for example "benzene", will be transformed into their IRIs in the knowledge graph as required by the description of the semantic agents. The transformation is conducted by an ontology lookup agent, which transforms natural language terms to their IRIs.

For the ontology lookup agent, a dictionary that maps different forms of representations of species to their IRIs in the OntoSpecies ontology is made in advance. The keys of this dictionary include conventional names, International Chemical Identifiers (InChI), Simplified molecular input line entry specification (SMILES), and chemical formulae. Via string similarity comparison and ranking, the ontology lookup agent finds the most suitable IRI representing the given species. Then the key-value pairs, where the species are transformed into IRIs are encoded into the URL retrieved from the agent description. The HTTP request can then be executed to provide an answer to the asked question. Figure 3 shows a screenshot of the answers returned by the QA system for the question "What is the heat capacity of CO2?".



Result	heat capacity at constant pressure	unit	temperature	pressure
1	37.04	J/mol/K	298.15 K	101325.00 Pa
2	37.13	J/mol/K	300.00 K	101325.00 Pa
3	41.17	J/mol/K	400.00 K	101325.00 Pa

Figure 3: *Results from Marie: answer to the question "plot the heat capacity of CO2". An online demonstration of the extended QA system is accessible through https://kg.cmclinnovations.com/explore/marie.*

Algorithm 2 Input mapping

$D_{input} \leftarrow \emptyset$
for All *input* from agent description **do**
    for All *entity* extracted by the NER model **do**
        if $input.hasNerLabel == entity.label$ **then**
            $D_{input}[input.hasName] \leftarrow entity.value$
        end if
    end for
end for

$D_{qualifier} \leftarrow \emptyset$
for All *output* from agent description **do**
    for All *qualifiers* from *output* **do**
        for All *entity* extracted by the NER model **do**
            if $qualifier.hasNerLabel == entity.label$ **then**
                $D_{qualifier}[qualifier.hasName] \leftarrow entity.value$
            end if
        end for
    end for
end for

4 Results

4.1 Evaluation questions

To evaluate the performance of the two NLP models trained and the overall performance of the QA system in answering questions, we created 100 evaluation questions. 50 of them are questions asking the power conversion efficiency of different species, for example, "What is the power conversion efficiency of OPF with donor of styryltrimethylsilane?". The other 50 of the questions ask about the thermodynamic data about various species, for example, "Show me CC1=C(C)CCC1's heat capacity at constant pressure at 150 Kelvin.".

The questions are generated semi-automatically. 100 species are randomly selected from TWA knowledge graph via the built-in Python function "random.sample()", represented in different forms including chemical formulae, names, InChI, and SMILES strings. Based on the species, questions are manually formulated with variations in structures and expression. For the thermodynamic data questions, conditions such as temperatures and pressures are also randomly generated by scripts via the built-in Python function "random.randint()" and "random.random()". Table 4 shows some examples from the evaluation question set.

Table 1: *Performance of models.*

	Question classification	Named Entity Recognition
F1	1.0	0.9456
Recall	1.0	0.9567
Precision	1.0	0.9349

4.2 Model evaluation

The question classification model is evaluated to see whether it can, given a question, select the most suitable agent to answer the question. The 50 questions asking power conversion efficiency and the 50 questions asking thermodynamic data are fed to the question classification model and the results are examined. For the power conversion questions, the correct classification result should be the IRI of the PCE agent and for the thermodynamic data questions the correct result is the IRI of the STDC agent. As shown in Table 1, the question classification model returned the correct result for all the evaluation questions.

Then the Named Entity Recognition (NER) model is evaluated to test how accurately this model can extract and label key components in the questions. The key components of each evaluation have already been separated out and labelled manually. The 100 evaluation questions are fed to the NER model and the predicted results are compared to the expected results. The true positive (TP) count is the number of words that appear in both the predicted and expected results, the false positive (FP) count is the number of words that appear only in the predicted results, and the false negative (FN) count is the number of words that appear only in the expected results. Table 1 shows the three scores: F1, recall, and precision, equation 16, 15, and 14 show how the three scores are calculated on TP, FP, and FN.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{16}$$

4.3 Answer evaluation

To examine the overall ability of this extension to call agents to answer questions, we also had an overall test on the QA system. The 100 evaluation questions are asked of the extended QA system. For each question, we first manually checked whether the correct request to the agent is constructed, in other words, whether the HTTP request format is

Table 2: *Overall performance of the extension on the QA system.*

	PCE agent	STDC agent	overall
Correct requests constructed	88%	78%	83%
Answer returned	84%	78%	81%
Previously unavailable answers	84%	70%	77%

correct and whether the correct input parameters are encoded into the request. Secondly, we automatically checked whether the agent returned any calculation result. However, whether the answer is correct or not is not examined since the purpose of this paper is to present the proof-of-concept integration between semantic agents and the QA system. Lastly, we counted the number of questions that the QA system could not answer before this extension. Table 2 shows the percentage of questions under the aforementioned three cases out of the 100 evaluation questions: correct requests constructed, answer returned, and previously unavailable answers returned.

5 Error analysis

Based on the results from the model evaluation and the answer evaluation, we categorise the errors occurred in answering the evaluation questions into three types: species lookup failure, NER failure, and out-of-scope species for agents. Table 3 gives the percentage of each type of failures in all failures.

The species lookup failure happens when the ontology lookup agent does not have the record of a species and hence failed to return the correct IRI of the species. This type of failure makes up 36.8% of all failures. For example, the species "MgCl2" was absent in the ontology lookup dictionary. The NER failure means the NER model failed to correctly extract and label all the key components and caused the failure to construct a correct request. This type of failure makes up 31.6% of all failures. One example is in question "What is the entropy of n-Butyraldehyde under -23 degree Celsius 101.3 Kpa?", "101.3 Kpa" is wrongly identified as a species. The out-of-scope species failure means although the correct request to the agent is constructed, the agent still failed to return an answer. This type of failure stems from the species being out of the scope for the agent, for example, the quantum calculation job of this species is not available for the STDC agent.

6 Conclusions

This paper introduces an extension of a question answering (QA) system for chemistry. This extension allows the QA system to automatically invoke semantic agents when the KG failed to provide information to answer the question. The use of an OntoAgent-

Table 3: *Percentage of different types of failures.*

Type of failure	Percentage
Lookup failure	36.8%
NER failure	31.6%
Out-of-scope species failure	31.6%

based description of the agents allows the QA system to use an agent lookup service to locate the agents based on their input/output (I/O) signatures. The OntoAgent ontology is also extended to provide information for the QA system to generate training material for NLP models automatically. The automated generation of training material consequently enables automated integration of new agents. From the evaluation results, we conclude that the automatically generated training material is sufficient to train natural language processing (NLP) models to handle new questions to be answered by agents. Also, the evaluation results show that this extension of the QA system does expand the range of questions by enabling access to dynamic components of the KG for the QA system.

From the error analysis, we identified tree types of failures: lookup failures, out-of-scope species failures, and NER failures. Lookup failures and out-of-scope failures are both caused by the absence of species in the knowledge graph. NER failure is caused by the lack of variation in the training material. Also, there are only two semantic agents integrated into the QA system, which limits the range of questions the QA system can answer. As a result, work on including more chemical species and more semantic agents in the knowledge graph is in progress. This will enable further progress towards the goal of Marie in lowering the barrier for general users to access the knowledge graph for chemistry.

7 Data and Software Availability

The training data, evaluation data, evaluation results are available in the GitHub repository https://github.com/cambridge-cares/TheWorldAvatar under subdirectory JPS_Chatbot. An online demonstration is available at https://kg.cmclinnovations.com/explore/marie.

7.1 Training data

The training data for the question classification model and the named entity recognition model are available in the nlu.md files under "UI/source/Agent_Query/training/data" directory. The data can be reproduced by running the script "create_nlu_for_ontoagent.py".

7.2 Evaluation data

The evaluation questions and their results are provided in repository doi:10.17863/CAM.78870.

7.3 Software

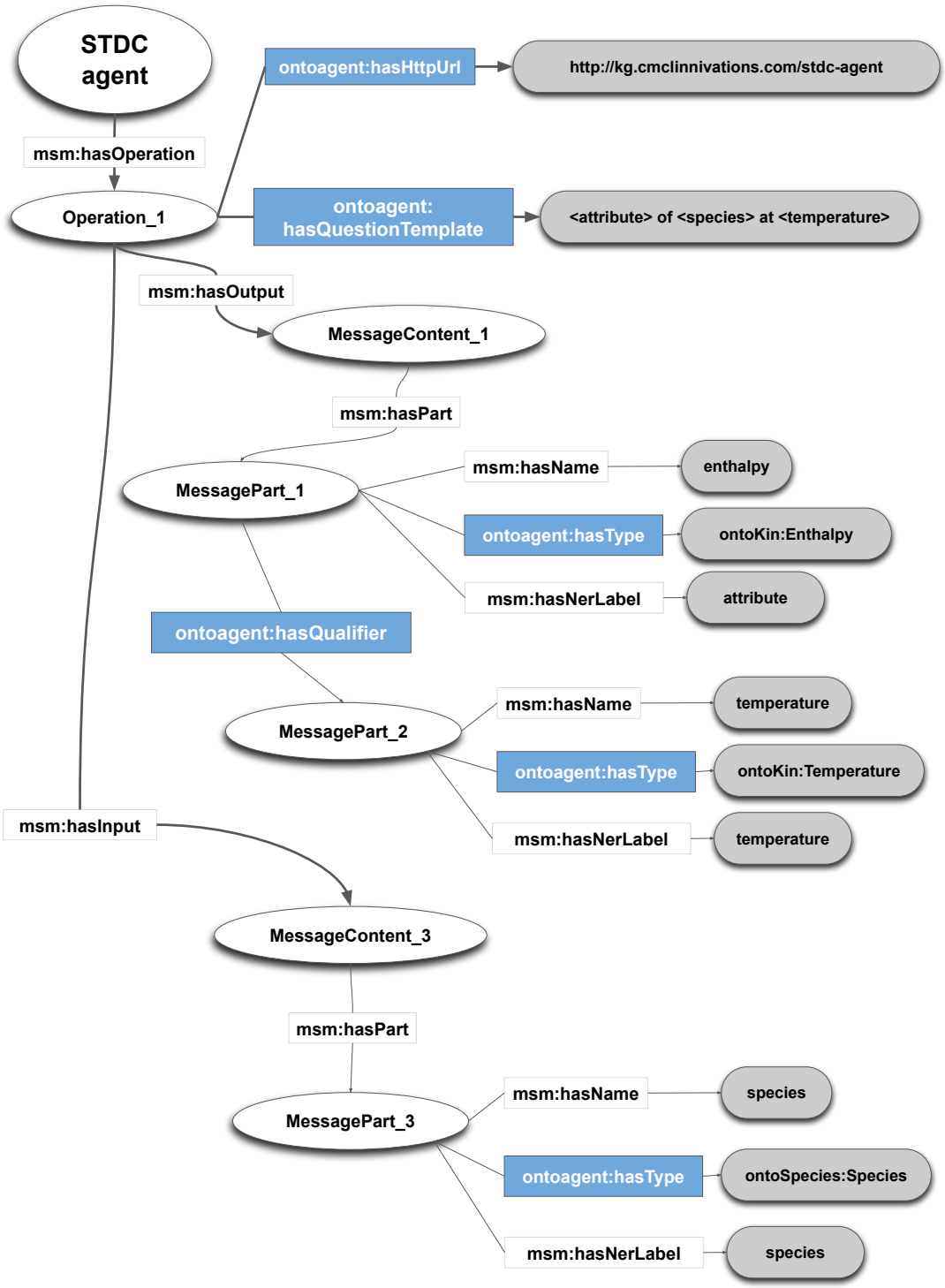
All the third-party software used in this system are freely available.

The Python environment suitable for operating the QA system is Python3.7 and all the Python libraries required and their versions are listed in the file "JPS_Chatbot/requirements.txt". All the packages from NLTK 3.5 also need to be downloaded and installed.

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 Agent description



B Example questions

Table 4: *Example questions from evaluation set.*

Example questions the QA system can answer
What is the power conversion efficiency of OPF with donor of styryltrimethylsilane?
What is enthalpy of C3H5N3O at the temperature of 294.62 degree Celsius?
What is C6H7NSe's enthalpy at 181.09 Fahrenheit?
What is C2H3IO's entropy at 230.84 kelvin and 1.01325 bar?
What is CH3's heat capacity at 61.11 degrees in temperature?
What is heat capacity at constant pressure of C6H11O3 at room temperature?
What is internal energy of NH4OH at -95 F?
What is heat capacity of InChI=1/C7H5N/c8-6-7-4-2-1-3-5-7/h1-5H under 30 C?
What is COC1CC1=C(C)C's entropy at 162 Fahrenheit?
What is pce of InChI=1/C7H12/c1-4-5-6-7(2)3/h1?
What is pce of OPF with donor of C=CC(C)=O?
What is power conversion efficiency of C2H6B4?
What is power conversion efficiency of CH3COCHO?
What is pce of OPF with donor of (CH3)3C-CN?
What is power conversion efficiency of nicaethan?
Show me CC1=C(C)CCC1's heat capacity at constant pressure at 150 kelvin

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. Semantic web. *Sci. Am.*, 284(5):34–43, 2001. doi:10.1038/scientificamerican0501-34. URL http://www.jstor.org/stable/26059207.

[2] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *Int. J. Semant. Web Inf. Syst.*, 5(3):1–22, 2009. doi:10.4018/jswis.2009081901.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] A. Bordes, S. Chopra, and J. Weston. Question Answering with Subgraph Embeddings. In *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process.*, pages 615–620, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics. doi:10.3115/v1/D14-1067.

[5] A. Bouziane, D. Bouchiha, N. Doumi, and M. Malki. Question Answering Systems: Survey and Trends. *Procedia Comput. Sci.*, 73:366–375, 2015. doi:10.1016/j.procs.2015.12.005.

[6] A. Chadzynski, N. Krdzavac, F. Farazi, M. Q. Lim, S. Li, A. Grisiute, P. Herthogs, A. von Richthofen, S. Cairns, and M. Kraft. Semantic 3D City Database — An enabler for a dynamic geospatial knowledge graph. *Energy AI*, 6:100106, 2021. doi:10.1016/j.egyai.2021.100106.

[7] W. Cui, Y. Xiao, and W. Wang. KBQA: An online template based question answering system over freebase. In *IJCAI*, pages 4240–4241, 2016. URL http://www.ijcai.org/Abstract/16/640.

[8] C. Deng, G. Zeng, Z. Cai, and X. Xiao. A Survey of Knowledge Based Question Answering with Deep Learning. *Journal on Artificial Intelligence*, 2(4):157–166, 2020. doi:10.32604/jai.2020.011541.

[9] A. Devanand, G. Karmakar, N. Krdzavac, R. Rigo-Mariani, Y. Foo Eddy, I. A. Karimi, and M. Kraft. OntoPowSys: A power system ontology for cross domain interactions in an eco industrial park. *Energy AI*, 1:100008, 2020. doi:10.1016/j.egyai.2020.100008.

[10] D. Diefenbach, P. H. Migliatti, O. Qawasmeh, V. Lully, K. Singh, and P. Maret. QAnswer: A Question Answering prototype bridging the gap between a considerable part of the LOD cloud and end-users. In *World Wide Web Conf.*, pages 3507–3510, New York, NY, USA, 2019. ACM. doi:10.1145/3308558.3314124.

[11] M. Dürst and M. Suignard. Internationalized resource identifiers (IRIs). Technical report, RFC 3987, 2005.

[12] 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.

[13] A. Eibeck, D. Nurkowski, A. Menon, J. Bai, J. Wu, L. Zhou, S. Mosbach, J. Akroyd, and M. Kraft. Predicting power conversion efficiency of organic photovoltaics: Models and data analysis. *ACS Omega*, 6:23764–23775, 2021. doi:10.1021/acsomega.1c02156.

[14] 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.

[15] 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.

[16] D. Fensel, U. Şimşek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, and A. Wahler. *Introduction: What Is a Knowledge Graph?*, pages 1–10. Springer International Publishing, 2020. doi:10.1007/978-3-030-37439-6_1.

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

[18] M. B. Hoy. Wolfphram|Alpha: A Brief Introduction. *Med. Ref. Serv. Q.*, 29(1): 67–74, 2010. doi:10.1080/02763860903485225.

[19] 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.

[20] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015. doi:10.3233/SW-140134.

[21] S. A. Lopez, E. O. Pyzer-Knapp, G. N. Simm, T. Lutzow, K. Li, L. R. Seress, J. Hachmann, and A. Aspuru-Guzik. The Harvard organic photovoltaic dataset. *Scientific Data*, 3(1):1–7, 2016. doi:10.1038/sdata.2016.86.

[22] W. Marquardt, J. Morbach, A. Wiesner, and A. Yang. *OntoCAPE*. RWTHedition. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-04655-1.

[23] D. A. McQuarrie and J. D. Simon. *Molecular thermodynamics*. University Science Books, Sausalito CA, USA, 1999.

[24] 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. doi:10.1021/acs.jcim.0c01145.

[25] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investig.*, 30(1):3–26, aug 2007. doi:10.1075/li.30.1.03nad.

[26] T. Pellissier Tanon, M. D. de Assunção, E. Caron, and F. M. Suchanek. Demoing Platypus – A Multilingual Question Answering Platform for Wikidata. In *Semant. Web ESWC 2018 Satell. Events*, pages 111–116. Springer International Publishing, 2018. doi:10.1007/978-3-319-98192-5_21.

[27] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004. doi:10.1023/B:STCO.0000035301.49549.88.

[28] H. J. Snaith. Perovskites: the emergence of a new era for low-cost, high-efficiency solar cells. *The Journal of Physical Chemistry Letters*, 4(21):3623–3630, 2013. doi:10.1021/jz4020162.

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

[30] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. doi:10.1006/knac.1993.1008.

[31] T. Van Veen. Wikidata. *Information Technology and Libraries*, 38(2):72–81, 2019. doi:10.6017/ital.v38i2.10886.

[32] E. Weisstein. Computable Data, Mathematics, and Digital Libraries in Mathematica and Wolfram|Alpha. In *CICM 2014 Intell. Comput. Math.*, pages 26–29, 2014. doi:10.1007/978-3-319-08434-3_3.

[33] 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 December 16, 2021.

[34] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston. Starspace: Embed all the things!, 2017.

[35] P. Wu, X. Zhang, and Z. Feng. A Survey of Question Answering over Knowledge Base. In *China Conf. Knowl. Graph Semant. Comput.*, pages 86–97. Springer Singapore, 2019. doi:10.1007/978-981-15-1956-7_8.

[36] 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:10.1016/j.compchemeng.2018.07.010.

[37] 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.

[38] 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.

[39] 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.