Hybrid Question Answering System based on Natural Language Processing and SPARQL Query

Mickael Rajosoa, Rim Hantach, Sarra Ben Abbes, Philippe Calvez
CSAI LAB ENGIE, France
Rajosoa.Mickael@gmail.com, Rim.Hantach@external.engie.com,
Sarra.BEN-ABBES@external.engie.com, Philippe.Calvez1@engie.com

Abstract

Chatbot is a conversational agent that communicates with users based on natural language. It is founded on a question answering system which tries to understand the intent of the user. Several chatbot methods deal with a model based template of question answering. However, these approaches are not able to cope with various questions and can affect the quality of the results. To address this issue, we propose a new semantic question answering approach combining Natural Language Processing (NLP) methods and Semantic Web techniques to analyze user's question and transform it into SPARQL query. An ontology has been developed to represent the domain knowledge of the chatbot. Experimentations show that our approach outperforms state of the art methods.

1 INTRODUCTION

Nowadays, the huge amount of data has been increased which makes the task of information retrieval more difficult. To overcome this problem, several approaches in different domain areas have been proposed based on question answering system (QA) [BAB12, YD14a] where the aim is to understand natural language questions and extract relevant information. QA is a computer science discipline designed to generate an answer of question posed by human in natural language. This system is based on Natural Language Processing methods and Information Retrieval (IR) [CLR13, Fer16]. NLP helps computer to understand and answer user's query through IR. Furthermore, QA systems are divided into two categories: an open domain and a closed domain. The open domain deals with questions related to several topics and the closed domain deals with questions related to a specific domain. In order to represent knowledge and facilitate the information retrieval, Semantic Web techniques are required. Ontology is one of these techniques. It's a formal model that allows to describe concepts and relations between them in a domain. However, recent works in the literature have not highly developed these techniques, they rely on keywords to identify the context and answers for user's question. These methods involve the removal of stop words. Nevertheless, the removal of these words can lead to the loss of the sentence's meaning. From this point, we found some improvement areas which motivate us to combine NLP methods and Semantic Web techniques. The main objective of our system is to get user's intent based on syntactic dependency relationships. In this paper, we review the related work in section 2. Then, in section 3, we highlight the use of the syntactic relationships to translate user's question into triple patterns and build the SPARQL queries. We conduct a comparative study with previous related work to prove the effectiveness of our approach in section 5. Finally, in section 6, we provide some conclusions.

2 Related Work

Several chatbot approaches have been addressed in the literature where most of them are based mainly on the preparation of a question answer template. In [A+18], authors suggest a chatbot leaded and controlled by template questions. Therefore, when user asks a question and it is present in the file that contains all the templates for questions and answers (AIML), the bot can provide an answer based on the question template. However, this approach, as shown in many works, has revealed a number of unexpected problems and weaknesses related to the reliability of answers. In practise, it is impossible to list all the possible questions that a user may have. Therefore, if a user's question is not in the dataset, the bot can't supply an answer.

Researchers proposed tools to transform user's question into SPARQL query language in order to find the answer. They created an application called Quepy [Mac18][BC14]. The purpose of this application is to transform a question (in natural language) into SPARQL in order to query linked open data such as DBpedia or Freebase [ABK+07, YD14b]. In this application, they highlight the use of NLP methods to identify named entities (i.e. human entities, places, organizations) and question templates in regular expression form to generate the SPARQL query. However, such an approach is doomed to be ineffective because it is based on a prepared template. To resolve these limitations, researchers advanced more in-depth approaches. For example, C. S. Kulkarni et al. [KBPK17] propose a new NLP and machine learning approach to cope with agent conversational system problems. First, a dataset of questions/answers has been prepared in order to train the model. Then, to categorize users question, authors suggested a non supervised classification algorithm where a cosine similarity measure has been used to classify new users questions. However, this approach, while quite efficient, does not deal with semantic relationships between questions.

In [BDNM18], a new approach has been proposed based on the combination of Semantic and NLP methods to enhance chatbots ability. Giving a question, keywords and named entities have been extracted. Therefore, the keywords express the intent of the question and help in the construction of the SPARQL request. In addition, authors propose to use WordNet [Mil95] to establish a synonym list and perform mapping. Nevertheless, removing the stop words and extracting only the keywords in order to identify questions intent can severely limit its effectiveness and induce erroneous answers. In [AKS17], A. Albarghothi et al. combine as well a linguistic approach with Semantic Web processing. They use NLP functions (normalization, tokenization, removing stop words, stemming, tagging) to translate natural language into triple patterns and query an ontology. This approach is limited because it suffers from semantic rules. In [APMG12], authors establish an ontology and AIML categories to reply users question. After a classic processing of the users question, they convert properties and relations between concepts into AIML categories to supply a complete sentence for the user. Their approach requires improvements to be deemed in industry. In [SWRR14], a simple Knowledge Organization System (SKOS) and Spin rules have been used to translate natural language into SPARQL. Nevertheless, this method is not yet applicable under industrial conditions.

In [NNBU+13], a new approach has been established to transform a SPARQL request into a natural language. To do this, different rules (these rules look like patterns) have been defined to build the sentences. Unfortunately, this approach suffers from the syntactic and semantic aspects. In fact, generated sentences may poorly be tuned due to the superimposition of rules. Thus, we obtain as results, sentences having no sense or without correct grammar rules..

Different state of the art approaches suffer from semantic and syntactic relationships to understand the intent of the question. To overcome these limitations, we propose a new semantic chatbot based on the dependency relationships and the SPARQL query. The originality of our approach lies in the definition of new rules to deal with question answering system.

3 Proposed approach based on Linguistics and Semantics rules

The approach is based on the combination of NLP methods and Semantic Web techniques. The purpose of this combination is to understand user's question. Here, "understand" means to get user's intent (what is he looking for behind his question?) in order to supply a correct answer. Our approach requires an ontology to represent information and relationships related to the topics. The main step, is to analyze the words of a sentence. It should be emphasized that words part of speech is not sufficient to understand the meaning of a sentence. It is also necessary to deal with the syntactic function of different words and detect the named entities which can help us to perceive the meaning of the question and discern the intention. Thus, we use the NLP tools to extract the syntactic structure of the sentence. Then, this linguistic information will be used to build the rules. Finally, these rules will be transformed into SPARQL queries to request the triple store. In addition to that, external

resources such as WordNet and DBpedia have been used to enhance and strengthen the reliability of our chatbot. Figure 1 shows an illustration of the proposed approach.

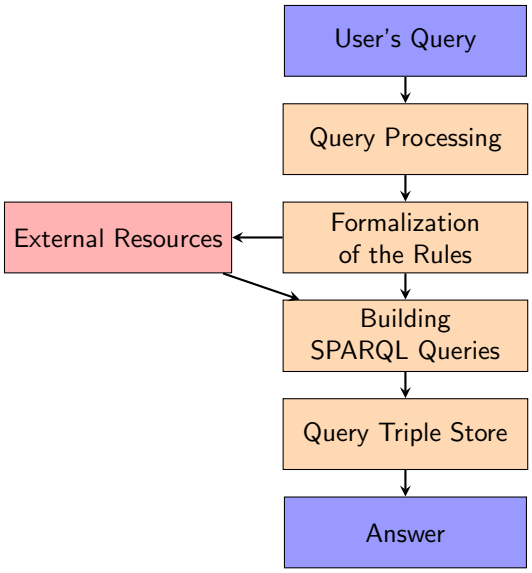

Figure 1: Process of the proposed approach

We start by the query processing (syntactic and linguistic annotations of the question), then, rules' formalization in order to transform the question into SPARQL query and finally, we query the triple store to get the answer.

3.1 Query Processing

The aim of the query processing is to analyze the linguistic and syntactic structure of the question. By focusing on the structure, we can know: what is the subject of the question? What is the core of the question or the main verb? What are the complements (object, noun)? Does the sentence have any specific feature for example coordinating conjunctions, a question with a copula (an intransitivity verb which links a subject to a noun phrase, adjective or other constituent which expresses the predicate)? Moreover, we start by extracting the part of speech of each word which allows us to know the subject of the sentence, the main verb that symbolizes the action of the sentence and the complements for specific cases. Then, we extract the named entities to identify the context of the sentence. Finally, the dependency relationships have been identified in order to obtain the syntactic functions and get relations between words in a sentence.

3.2 Formalization of the rules

During this step, we develop a generalized rules model based on syntactic relationships in a question. The defined rules could be adapted to any question (who, what, where, when, how, etc.), they are presented as follows:

- 1st rule: when the question contains a possessive phrase. This means that the possessive will give additional information to his syntactic head. Thus, this syntactic head will become the secondary predicate.

  Example : What is the name of Michelle Obama's daughters ?

  "name" represents the main predicate and the apostrophe "s" is the possessive of "Michelle Obama". The syntactic head of this possessive is "daughters". Therefore, the latter become the secondary predicate.

- 2nd rule: when the question contains coordinating conjunctions such as "and", "or". This means that all named entities in the question are on the same level. In other words, they share the same main predicate.
  Example: Who are the daughters of Michelle Obama and Barack Obama ?
  "daughters" is the main predicate and we have two named entities which are separated by a coordinating conjunction "and" which means that we want to know the daughter of both.

- 3rd rule: when the question contains a non-verbal predicate, for example the verb "to be" and it is preceded by the main predicate and it's an interrogative adverb type, then, the intent of the question will be a nominal subject (in most cases it's a noun).

  Example : How is Eagle's syndrome ?

  In this example, we have an interrogative adverb "How" followed by a non-verbal predicate. Consequently, we focus on the nominal subject of the question to get the intent. "syndrome" is the subject of the question. So, it becomes the main predicate.

- 4th rule: other questions that have a subject, a main predicate and a complement.

  Example : Who wrote Harry Potter ?

  "wrote" is the main predicate of the question and "Harry Potter" is a named entity and a complement. In this case, we are looking for the subject of the question. In other words, the writer of the novel.

These rules can be perfectly combined. If a sentence or a question follows a rule, it does not exclude the other rules. The main issue of syntactic dependencies is the identification of user's intent. Indeed, sometimes NLP methods are not able to correctly identify the intent. That's why, external resources have been used to deal with semantic relations (section 3.5).

3.3 Building SPARQL Queries

After listing all the rules (section 3.2), we now associate each of these rules to a SPARQL request. The core of the question will be considered as the main predicate. In other words, it's the main property that will connect a resource A to a resource B, in each question there is always a main predicate. For the modifiers or complements, their syntactic heads will be considered as the secondary property. This property will link a resource C to a resource A.

For the specific cases: first, we know that the named entities in a user's question share the same property when we have a coordinating conjunction in the sentence. In other words, user's intent is exactly the same for these entities. Therefore, we use *UNION* structure because it's useful for concatenating solutions from two possibilities. Second, if the heart of the question is an interrogative adverb then we consider that the main predicate will be the nominal subject of this sentence. SPARQL queries for the different rules are defined as follows:

- 1st rule:

```
SELECT DISTINCT ?a
WHERE
    {?ans onto:main_predicate ?a
     ?x onto:secondary_predicate ?ans}
```

- 2nd rule:

```
SELECT DISTINCT ?ans_label
WHERE
    {?x onto:main_predicate ?ans
     ?y onto:main_predicate ?ans
     ?ans rdfs:label ?ans_label.
    {?x rdf:type onto:Person
     ?x rdfs:label 'X'.}
UNION
    {?y rdf:type onto:Person
     ?y rdfs:label 'Y'.}}
```

- 3rd rule:

```
SELECT DISTINCT ?ans_label
WHERE
    {?x onto:nominal_subject ?ans
     ?ans rdfs:label ?ans_label.}
```

- 4th rule:

```
SELECT DISTINCT ?a
WHERE
    {?ans onto:main_predicate ?a}
```

3.4 Query the Triple Store

The knowledge graph is stored in a triple store called GraphDB [NAJ14]. We chose a triple store because the data will be structured as a triplet and it will be easy to update it using SPARQL. We use a wrapper service to query the repository of our knowledge Graph and get answers to our questions (Figure 2).

User: Who is Camille Dupond's father?
Bot: Paul Dupond.
User: Where did Chantal come from?
Bot: Berlin.

Figure 2: Conversation between the bot and the user

3.5 External Resources

We employ external resources to reduce ambiguities. In fact, the user may use specific terms in his question and NLP tools are not able to identify the user's intent or extract the named entities. These problems are solved through two external resources: WordNet and DBpedia.

3.5.1 WordNet

WordNet [Mil95] is a lexical database developed by Princeton University. Once the intention of the question has been identified, a list of synonyms must be drawn up to promote mapping on the ontology. The integration of this resource allows our system to avoid ambiguities.

3.5.2 DBpedia

DBpedia is used to find the type of named entities. Indeed, NLP tools may omit to extract some entities. Thus, thanks to grammatical analysis which gave us upstream the different proper nouns of the sentence. DBpedia is able to identify the type of each proper name. It's a knowledge base that standardizes the content of Wikipedia. Each Wikipedia page is browsed by a set of extractors and these extractors will identify elements of the page and generate data. We map the proper noun with his label, then we try to get his type with a SPARQL request.

4 ILLUSTRATIVE WITH EXAMPLE

For our approach, we use Stanford Core NLP [Cor19] as NLP tools. In Table 1, we mention the main syntactic relations that interest us. As we can see on the left of the table, the annotation used by Stanford and on the right, names of syntactic functions in dependency grammar. We illustrate our approach using the example below.

Example: What is Genghis Khan's real name?

4.1 Query Processing

During this step, user's query follows four processing : tokenization, parsing, dependency parsing and named entity recognition (NER). Tokenization is the task of cutting it up into pieces, called tokens. Parsing gives the parts of speech of each word and the structure syntagmatics of sentence. Dependency Parsing analyzes the grammatical structure of a sentence, and establishes relationships between "head" words and words which modify those heads. NER classifies named entities that are present in a question into predefined categories like *person*, *organization*, *location*, etc..

Table 1: Main dependency relationships for our system

Dependency Parsing	
Annotation by Stanford	Name of the Function
ROOT	Predicate (core of sentence)
nsubj	nominal subject
nmod	nominal modifier
cop	copula
cc	coordinating conjunction
conj	conjunct
avdmod	adverbial modifier

```
tokenization: ['What', 'is', 'Genghis',
      'Khan', ''s', 'real', 'name', '?']

parsing:
(ROOT
  (SBARQ
    (WHNP (WP What))
    (SQ (VBZ is)
     (NP
     (NP (NNP Genghis) (NNP Kan) (POS 's))
     (JJ real) (NN name)))
    (. ?)))

dependency parsing:
[('ROOT',0,1),('cop',1,2),('compound',4,3)
,('nmod',7,4),('case',4,5),('amod',7,6),
('nsubj',1,7),('punct',1,8)]

NER:
[('What', 'O'),('is', 'O'),('Genghis',
'PERSON'), ('Khan', 'PERSON'),(''s', 'O'),
('real', 'O'),('name', 'O'),('?', 'O')]
```

4.2 Formalization of the rules into SPARQL

From Query Processing results, we establish a SPARQL request that queries the triple store. Parsing indicates that there is a noun phrase (NP) in the question. The latter composed of two proper names "Genghis" & "Khan". Dependency Parsing gives more information about the function of these words. Indeed, it tells us that "Genghis" is a compound word and his syntactic head is "Khan". Thus, these two proper names are linked together. Named entity recognition (NER) indicated the type of these two names which is "PERSON" type. Then, when we look back on dependency relationships, we see that the main kernel is "What".

Nevertheless, in dependency relationships, we can't have "ROOT" type of interrogative pronoun. In addition, "ROOT" is followed by a copula. In this case, according to the third rule, the intention of a question is symbolized by the nominal subject. The nominal subject in this question is "name" so it becomes "ROOT" of the question. Therefore, it is considered as the main predicate. These annotations are expressed in the following request:

```
SELECT DISTINCT ?reponse
WHERE {?x onto:name ?reponse.
       ?x rdf:type onto:Person.
       ?x rdfs:label 'Genghis Khan'.}
```

4.3 SPARQL into Answer

After transforming user's question into a SPARQL request, we query the triple store to get the answer corresponding to the question. In order to do this, we use a SPARQL Wrapper [LZB17] , it's a wrapper around a SPARQL service that allows us to query the URI of our triple store.

User: What is Genghis Khan's real name?
Bot: Temujin.

5 Evaluation

5.1 Background

To evaluate the performance of our approach, an ontology has been used that symbolizes the concept *person* Figure 3. The ontology represents personal information about a person x, such as date of birth, place of birth, profile description, job, etc. Therefore, the ontology contains classes (*Person*, *Organization*, *Occupation*, and *Location*, etc), object properties (*wasBorn*, *isLocated*, *hasOccupation*, etc) and data properties (*born*, *cost*, *description*, etc).

Precision, Recall and F-measure have been used to compare our approach with A. Bouziane et al [BDNM18]. The main purpose of this evaluation is to evaluate chatbot's ability to identify the user's intent. In order to do this, we submit a dataset of questions related to the ontology *person*. These questions were built by Yassine Benajiba [RBL06]. In this study case, the questions will be asked in order to extract personal information related to a person, his family, his job, etc.

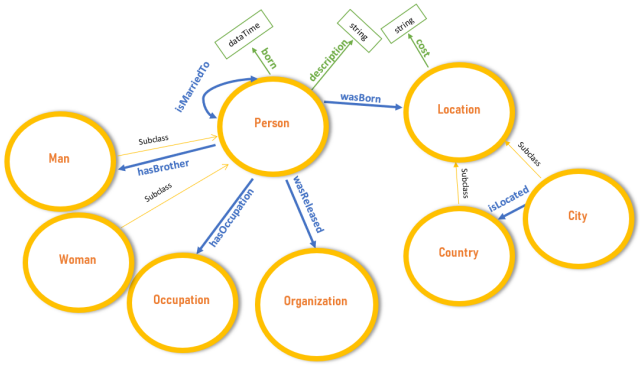

Figure 3: Ontology used during evaluations

$$Precision(P) = \frac{\text{Number of Correct Answer}}{\text{Total Number of Answer}}$$

$$Recall(R) = \frac{\text{Number of Correct Answer}}{\text{Total Number of Correct Answer}}$$

$$F{-}measure = 2 \times \frac{\text{(P x R)}}{\text{(P + R)}}$$

5.2 Results

A comparison was made between our system approach and an Arabic Question Answering System [BDNM18]. Their system attained respectively 0.71, 0.66, 0.68 for the precision, recall and F-measure. However, our system successfully achieves, as you can see in Table 2, 0.88, 0.86 and 0.87 in terms of Precision, Recall and F-measure.

The main challenges in our approach were the formalization of the rules and the building of the ontology. In fact, the rules are formalized manually and it requires significant corpus of questions to elaborate generic rules. These rules demand a permanent renewal as soon as specific cases arise. In addition, in order to deal with the questions present in the dataset, it is necessary to create a same domain ontology than [BDNM18]

Table 2: Evaluation results, expressed in Precision, Recall and F-measure.

Results		
Measure Performance	System Approach	A. Bouziane et al.
Precision	**0.88**	0.71
Recall	**0.86**	0.66
F-measure	**0.87**	0.68

with all the concepts, relations and instances. The implementation of these two things may take time but our method is fruitful because the results show that the system is very efficient to find the user's intent and they are much higher than [BDNM18] approach. This is because we have essentially used dependency relationships to understand user's intent.

Indeed, these relationships have helped us to understand not only the meaning of user's question but also the structure of his question. This linguistic information are then managed by the rules that we have formalized in order to be translated into triple patterns and find the answer to the question. While [BDNM18] put forward the stop words removal to identify the user's intent. This method is too drastic and not applicable to certain number of questions. In fact, by removing the empty words, this can affect or destroy the meaning and especially the structure of the question. Therefore, their system may provide an incorrect answer which clearly distorts the results of their approach.

We believe that our method represent a significant improvement of the state of the art QA systems due to the development of a generic method that deals with different cases and different ways that the questions were asked. Indeed, using the NLP methods helps us to identify user's intent, however there is still room for improvement. For example: dealing with several intents in a question. Actually, our system can only detect one intent at a time. Then, we can make automatic rules generation.

6 Conclusion & Future Works

One of the biggest challenges in the development of question answering system is to advance a conversational system or chatbot that is not based on preparation in upstream of questions and answers template. This paper presents a question answering system based on NLP methods and Semantic Web techniques to provide answers to questions expressed in natural language. In our approach, we use NLP methods to process user's question. In this processing, we use syntactic dependency relationships to view the semantic and syntactic structure of the question. These relationships are very important to understand and correctly answer user's question. Then, we transform them into SPARQL queries by means of rules in order to query our triple store. Indeed, the knowledge of the chatbot has been represented using an ontology. The evaluation of our approach shows that our method is good as our system can be adapted to a large number of questions. This shows that our approach constitutes a significant step in the question answering field.

However, the proposed approach requires improvement in future works. First, we will have to rework the intent of the bot. Indeed, the bot can respond and process one intention at a time for now. Secondly, the formalization of the rules is still manual and it will be better to take into account this issue. Then, we will introduce the ontology alignment in our method in order to deal with open domain and ameliorate results. Finally, we can customize our bot, by adding vocal conversations. This implies implementation of machine learning and advanced algorithms.

References

[A+18] A. ARAIN et al. Artificial intelligence mark-up language based written and spoken academic chatbots using natural language processing. *Sindh University Research Journal-Science Series*, 50:153–158, mar 2018.

[ABK+07] Sren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. volume 6, pages 722–735, 01 2007.

[AKS17] Ali Albarghothi, Feras Khater, and Khaled Shaalan. Arabic question answering using ontology. *Procedia Computer Science*, 117:183–191, dec 2017.

[APMG12] Agnese Augello, Giovanni Pilato, Alberto Mach, and Salvatore Gaglio. An approach to enhance chatbot semantic power and maintainability: Experiences within the frasi project. In *IEEE International Conference on Semantic Computing*, pages 186–193, sep 2012.

[BAB12] Raju Barskar, Gulfishan Ahmed, and Nepal Barskar. An approach for extracting exact answers to question answering (qa) system for english sentences. *Procedia Engineering*, 30:1187–1194, dec 2012.

[BC14] Ritika Bansal and Sonal Chawla. An approach for semantic information retrieval from ontology in computer science domain. *International Journal of Engineering and Advanced Technology*, 4:58–65, dec 2014.

[BDNM18] Abdelghani Bouziane, Bouchiha Djelloul, Doumi Noureddine, and Malki Mimoun. Toward an arabic question answering system over linked data. *Jordanian Journal of Computers and Information Technology*, may 2018.

[CLR13] L. Chiticariu, Yunyao Li, and F.R. Reiss. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 827–832, Seattle, Washington, USA, oct 2013.

[Cor19] Stanford CoreNLP Natural Language software, https://stanfordnlp.github.io/CoreNLP/index.html, 2019.

[Fer16] Sbastien Ferr. Sparklis: An expressive query builder for sparql endpoints with guidance in natural language. *Semantic Web*, 8:405–418, dec 2016.

[KBPK17] Chaitrali S. Kulkarni, Amruta U. Bhavsar, Saviata R. Pingale, and Satish S. Kumbhar. Bank chat bot - an intelligent assistant system using nlp and machine learning. *International Research Journal of Engieering and Technology*, 4:2374–2376, may 2017.

[LZB17] Maxime Lefrançois, Antoine Zimmermann, and Noorani Bakerally. A SPARQL extension for generating RDF from heterogeneous formats. In *Proc. Extended Semantic Web Conference (ESWC'17)*, Portoroz, Slovenia, May 2017.

[Mac18] Machinalis. Quepy's project documentation, 2018.

[Mil95] George A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41, 1995.

[NAJ14] Lucas Fonseca Navarro, Ana Paula Appel, and Estevam Rafael Hruschka Junior. Graphdb – storing large graphs on secondary memory. In *New Trends in Databases and Information Systems*, pages 177–186, Cham, 2014. Springer International Publishing.

[NNBU+13] Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. Sorry, i don't speak sparql: Translating sparql queries into natural language. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 977–988. ACM, 2013.

[RBL06] Paolo Rosso, Yassine Benajiba, and Abdelouahid Lyhyaoui. Towards an arabic question answering system. *Proc. of SRO4*, jan 2006.

[SWRR14] Malte Sander, Ulli Waltinger, Mikhail Roshchin, and Thomas A. Runkler. Ontology-based translation of natural language queries to sparql. In *AAAI Fall Symposia*, 2014.

[YD14a] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, volume 1, pages 956–966, jun 2014.

[YD14b] Xuchen Yao and Benjamin Van Durme. Information extraction over structured data: Question answering with freebase. In *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, volume 1, pages 956–966, jun 2014.