ENTERPRISE INFORMATION SYSTEMS
2019, VOL. 13, NOS. 7–8, 935–958
https://doi.org/10.1080/17517575.2019.1597385

Taylor & Francis
Taylor & Francis Group

Check for updates

Crowdsourcing-based semantic relation recognition for natural language questions over RDF data

Xin Hu [a], Jiangli Duan [b] and Depeng Dang [a]

[a]College of Information Science and Technology, Beijing Normal University, Beijing, China; [b]Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, China

ABSTRACT

Natural language query systems over RDF data need to rely on the semantic relations in query. First, we propose the new crowdsourcing model that used to produce semantic relations dataset. The model not only inherits completeness of the iterative model and accuracy of the parallel model, but also saves human resources. Second, we mine the rules of semantic relation recognition from the correlations between dependency structures and semantic relations. Third, we propose an algorithm of semantic relation recognition for natural language query over RDF data, and experiments demonstrate that it can recognize more semantic relations than existing methods.

ARTICLE HISTORY
Received 24 December 2017
Accepted 17 March 2019

KEYWORDS
Semantic relation; crowdsourcing; natural language question answering; RDF; data mining

1. Introduction

Natural language question answering over RDF (Resource Description Framework) data have received widespread attention because natural language is easy to use and has strong expressive power. Most natural language question answering systems contain three stages: recognising semantic relations from a question, producing candidate mappings for semantic relations, and translating basic graph pattern (i.e., the combination of semantic relation mappings) to a SPARQL statement or searching a subgraph that contains basic graph pattern from an RDF data graph. It is obvious that the semantic relation is the key for natural language question answering over RDF data. However, existing state-of-the-art natural language question answering systems such as DEANNA (Yahya et al. 2013) and the graph data-driven approach (Zou et al. 2014) have not noticeably improved the answer ratio because the semantic relations in most of the natural language questions cannot be completely recognised.

Existing methods (Yahya et al. 2012a, 2012b, 2013; Yahya 2016; Zou et al. 2014; Liu et al. 2017) can recognise some semantic relations by the rules that are based on an artificial hypothesis (i.e., the relation phrases in semantic relations come from the verb phrases in questions), however, it is not always true. Yahya et al. (2012a, 2012b, 2013, 2016) maps verb phrase and noun phrase to predicate and type/entity, respectively, and then utilises their connection in RDF data graph to form semantic relation mappings. Zou et al. (2014) find all

CONTACT Depeng Dang ddepeng@bnu.edu.cn College of Information Science and Technology, Beijing Normal University, Beijing 100875, China

© 2019 Informa UK Limited, trading as Taylor & Francis Group

المنارة للاستشارات
www.

verb phrases which are common to the question and the paraphrase dictionary *D* which records the semantic equivalence between verb phrases and predicates and then finds two associated arguments of each verb phrase from the question according to heuristic linguistic rules. Finally, the verb phrase, together with its two associated arguments, forms a semantic relation <*arg*1, *rel, arg*2> (a more rigorous discussion is available in DEFINITION 5). However, if the relation phrase *rel* in the semantic relation is not a verb phrase, or the paraphrase dictionary *D* does not include the verb phrase in the question, they cannot recognise the semantic relations (e.g., Example 1). In addition, Liu et al. (2017) has used some dependency structures to recognise semantic relations, such as '*nsubj*', '*nmod:of*', '*amod*' and so on (De Marneffe et al. (2014) has given more rigorous discussion about dependency structures), but these dependency structures is not enough.

Example 1. For the question '*How many books by*[1] *Kerouac were published by Viking Press?*', the verb phrase '*published*' will most likely be found in the paraphrase dictionary *D* by chance, while the nonverb phrase '*by*[1]' is not. Therefore, they can recognise <*Kerouac, published, Viking Press*> and overlook <*books, by*[1]*, Kerouac*>.

To avoid over-reliance on the verb phrases, we expect to recognise the semantic relations by the rules which were mined from dataset rather than the rules which originate from an artificial hypothesis such as a verb phrase and so on, namely, the rules come from the correlations between dependency structures and semantic relations. The dependency structure dataset can be produced by the Stanford parser while the semantic relation dataset needs to be collected manually. Crowdsourcing is suitable for producing a semantic relation dataset because it is a human organisation model to help solve a wide variety of problems which are more difficult to address for computers, but humans can easily handle them (see related work for details). However, for current crowdsourcing models, we must make a choice between high accuracy (but low completeness) or high completeness (but low accuracy). Since the algorithms in data mining usually require much more qualified data, a new crowdsourcing model which has high accuracy/completeness is desperately needed.

In conclusion, we need a crowdsourcing model which is used to produce semantic relation dataset and has relatively high accuracy/completeness, an algorithm of mining the rules of semantic relation recognition from datasets, and an algorithm of recognising semantic relations from the natural language questions. To solve the above problems, we make the following contributions in this paper:

(1) We propose a new crowdsourcing model (i.e., the parallel-dominated iterative model with feedback) which is used to produce a semantic relation dataset. The model inherits accuracy and completeness from parallel model and iterative model, respectively, and saves human resources.
(2) We propose an algorithm of mining semantic association rules from the correlations between dependency structures and semantic relations. The semantic association rules include the association between the dependency structure and the semantic relation (i.e., *subject-like, object-like, triple-like*$\Rightarrow$*R*), and the association between the dependency structure combination and the semantic relation (e.g., *nsubj, nmod* $\Rightarrow$*R*).

المنارة للاستشارات
www.

(3) Based on the semantic association rules above, we propose an algorithm of semantic relation recognition for natural language question over RDF data, and experimental results demonstrate that the algorithm can recognise more semantic relations than existing methods for natural language questions over RDF data.

2. Related work

In the current Big Data era, semantics enables computer to understand and reason data (Salem, Boufares, and Correia 2014), which can be applied to analyse social media (Basili, Croce, and Castellucci 2017), economic news (Elshendy and Colladon 2017), multimedia resources (Hu et al. 2014) and so on. The Semantic web is a web of data, in which each metadata has specific semantics. It can be used to improve information retrieval (Li et al. 2014; Luo et al. 2015), web service (Chen et al. 2015), and business process management (Rico et al. 2015; Hoang, Jung, and Tran 2014). RDF has been widely used as a W3C standard to describe data in the Semantic Web. For the better effectively utilise RDF data, natural language question answering over RDF data have received widespread attention (semantic relation recognition is the core of understanding natural language question).

2.1. Natural language question answering over RDF data

Many natural language question answering systems improve the answer ratio and efficiency by limiting input, constructing templates, analysing sentence structure and disambiguation, rewriting input, evaluating a user's interaction and so on. Based on controlled natural languages, the approaches in (Ferré 2013, 2014; Mazzeo and Zaniolo 2016) consider a well-defined restricted subset of natural language that can be unambiguously interpreted. TBSL (Unger, Bühmann, and Lehmann 2012) is a template-based approach, which constructs some templates based on a linguistic analysis of the input question, so both the number of templates and the diversity of questions are limited. To tackle this problem, Zheng, Zou, and Lian (2015) and Abujabal et al. (2017) study how to generate templates automatically. Yahya et al. (2012a, 2012b, 2013, 2016) map verbal phrases to relations and noun phrases to either individual entities or semantic classes, and judiciously generates variables for target entities or classes to express joins between multiple triple patterns. Liu et al. (2017) propose a method for constructing directed acyclic graphs and triples, and the parsing for the modifier constraint greatly improves the conversion efficiency. Rozinajová and Macko (2016) propose a method based on a sentence structure and alternative word set, which produces an adjacent relation between entities by utilising dependencies between the words in a question, and then produces triples by an alternative word set. Dubey et al. (2016) propose a framework, called AskNow, where the question is first normalised into an intermediary canonical syntactic form (i.e., the sentence structure-based templates), and then translated into SPARQL statements. Zou et al. (2014) propose an entire-graph data-driven framework, which pushes down the disambiguation into the query evaluation stage. Shekarpour et al. (2017) propose a method for automatic rewriting of questions on graph-structured RDF data. Lopez and Motta (2004, 2005, 2006) propose an ontology-portable question answering system, which translates the question into a SPARQL statement by the user's interaction. Freitas and Curry (2014) present a distributional-

المنارة للاستشارات
www.

compositional semantics approach for natural language questions over heterogeneous linked data graphs.

In addition, there are some natural language question answering systems that pay attention to many other interesting research directions. Amsterdamer (2014), Amsterdamer, Kukliansky, and Milo (2015a, 2015b) develops NL2CM, a prototype system that queries general and individual knowledge, which translates the question into a well-formed crowd-mining query statement OASSIS-QL. Fader, Zettlemoyer, and Etzioni (2014), Sun et al. (2015), Balakrishna et al. (2016) and Tatu et al. (2016) propose the open question answering systems over curated and extracted knowledge bases. El-Ansari, Beni-Hssane, and Saadi (2017) present a question answering system which combines multiple knowledge bases. Scholten et al. (2016) and Hamon, Grabar, and Mougin (2017) propose the natural language question answering systems for medical linked data. Höffner, Lehmann, and Usbeck (2016) propose the question answering system on RDF data cubes. Moreover, because most of the methods cannot tell the user that the answer is right or not, Ngonga Ngomo et al. (2013) translates the SPARQL statement into natural language. Habernal and KonopíK (2013) present the Czech natural language question answering system, and Ray and Shaalan (2016) review the developments occurring in Arabic question answering systems as well as the challenges faced by researchers in developing Arabic question answering systems.

However, in the above-mentioned literature, only a few methods try to improve semantic relation recognition for the natural language questions. Yahya et al. (2012a, 2012b, 2013, 2016) map verbal phrases to relations and noun phrases to either individual entities or semantic classes, and then produces a semantic relation by combining them. On this basis, Zou et al. (2014) improve the ability of combining verbal phrases and noun phrases (i.e., finding two arguments for the verbal phrase by several heuristic rules). Moreover, Liu et al. (2017) propose some extraction rules of semantic relation such as '*nsubj*', '*nmod:of*', '*amod*' and so on.

In conclusion, although there is a considerable amount of systems or methods to answer natural language questions over RDF data, most of them (i.e., except the template-based approaches) need to rely on the semantic relations in the natural language questions because the semantic relation has the same form as the triple in RDF data. To date, however, the rules of semantic relation recognition originate from an artificial hypothesis (Yahya et al. 2012a, 2012b, 2013; Yahya 2016; Zou et al. 2014; Liu et al. 2017), so they have strong subjectivity and a low recognition rate. Therefore, in this paper, we mine the rules from a dataset, which can compensate for the lack of human experience and obtain more accurate rules.

2.2. Crowdsourcing

2.2.1. What is crowdsourcing?

In an article for Wired Magazine in 2006, Jeff Howe defined 'crowdsourcing' as 'an idea of outsourcing a task that is traditionally performed by an employee to a large group of people in the form of an open call' (Howe 2006). Prime examples include Wikipedia, Linux, Yahoo! Answers and Mechanical Turk-based systems (Doan, Ramakrishnan, and Halevy 2011). In recent years, with the rapid development and wide acceptance of the

المنارة للاستشارات

www.

Internet, crowdsourcing is no longer confined to a commercial production model and has become the model for handling distributed problems, which are difficult to be processed on a computer (Yuen, King, and Leung 2011). Crowdsourcing has three marked features: 1) Crowdsourcing can be applied to some problems, which are harder to address by computers, but humans can easily handle, such as image recognition (Yan, Kumar, and Ganesan 2010), text recognition (Robinson et al. 2012), video recognition (Verroios and Bernstein 2014), content analysis (Conley and Tosti-Kharas 2014) and reasoning (Demartini, Difallah, and Cudré-Mauroux 2013); 2) Many workers working together can easily complete massive tasks with a heavy workload, which can help to increase efficiency; 3) Workers can be paid a small amount of remuneration (Mason and Watts 2010; Horton and Chilton 2010) for tasks they are interested in. For example, YouTube is a key motivational tool that promotes video sharing among users without cost (Huberman, Romero, and Wu 2009). Studies regarding crowdsourcing can be classified into application research (Jagadeesan et al. 2009; Jorda, Sawaya, and Yeates 2014; Mellebeek et al. 2010), efficiency research (Koblin 2009; Warby et al. 2014; MacLean and Heer 2013), and high-quality model research.

For high-quality model research, Ipeirotis, Provost, and Wang (2010) propose that workers should be rated and found that adopting the results from high-level workers would improve the quality of the final results. Hsueh, Melville, and Sindhwani (2009) propose three selection criteria to filter high-quality annotation results. Sheng, Provost, and Ipeirotis (2008) explain that duplicate labels could improve the model result. Friess (2007) and Hafner (2007) suggest that each small area should be searched by a group of people to improve the quality during search and rescue. Maisonneuve and Chopard (2012) study two types of parallel and iterative redundancy mechanisms.

In these high-quality model studies, we must give a choice between quality and cost. In this study, we will explore a high-quality and low-cost crowdsourcing model and then use it to identify the semantic relations in the natural language questions.

2.2.2. Two common crowdsourcing models

As common crowdsourcing models, the iterative model (Sheng, Provost, and Ipeirotis 2008) and the parallel model (Friess 2007; Hafner 2007) can reduce the impact of insincere or malicious workers on crowdsourcing results.

Iterative model: *n* workers perform the same task in succession, that is, a worker performs a task based on the result of the previous worker, as shown in Figure 1. The process is executed until all workers finish the task, and the result of the last worker is considered as the result of the iterative model.

Parallel model: *n* workers complete the same task independently, as shown in Figure 2. There are *n* results that come from *n* workers after finishing the task, and the result of the parallel model can be obtained by combining the *n* results.

المنارة للاستشارات
www.

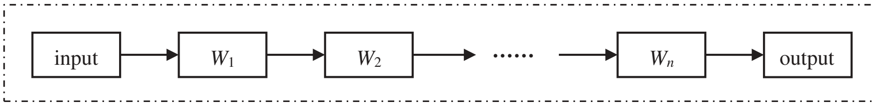

Figure 1. Schema of the iterative model.

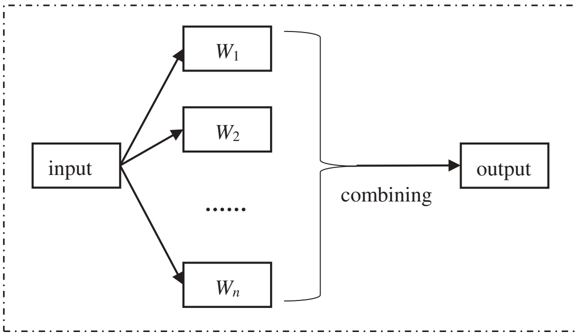

Figure 2. Schema of the parallel model.

3. Our crowdsourcing model

3.1. The parallel-dominated iterative model with feedback

Both the iterative model and the parallel model have their own merits and defects (Maisonneuve and Chopard 2012). The iterative model asks n workers to perform the same task in succession so that it can enable later workers to focus their attention on 'fresh' areas (i.e., relatively high completeness); however, exploiting previously discovered solutions can lead to a premature convergence on suboptimal solutions. The parallel model allocates n independent workers in parallel to do the same task, and the result comes from the high consensus of workers so that it has relatively high accuracy; meanwhile, the workers always focus on a general area rather than 'fresh' areas (i.e., relatively low completeness).

We propose a hybrid model, called the parallel-dominated iterative model with feedback, which inherits the advantages of two models and overcomes their defects. Namely, on the one hand, the hybrid model can generate a high completeness of solutions and constantly improve previous results. On the other hand, the iterative part of the hybrid model can avoid generating many wrong results, and the parallel part of the hybrid model can avoid leading to a premature convergence on suboptimal solutions. Furthermore, we also consider the characteristic that workers always care about their own mistakes pointed out by others rather than others' mistakes, so a unique group of workers performs the same task iteratively in the hybrid model. Therefore, the hybrid model improves the quality of the results and reduces the number of workers.

Parallel-dominated iterative model with feedback: 1) n workers form one group; 2) the unique group performs the same task many times, namely, one time represents one iteration in the iterative model and n workers complete the same task independently in one iteration; 3) after each iteration, the n worker's results are combined into the middleware result which is the consensus of most people, and the non-consensus part

المنارة للاستشارات
www.

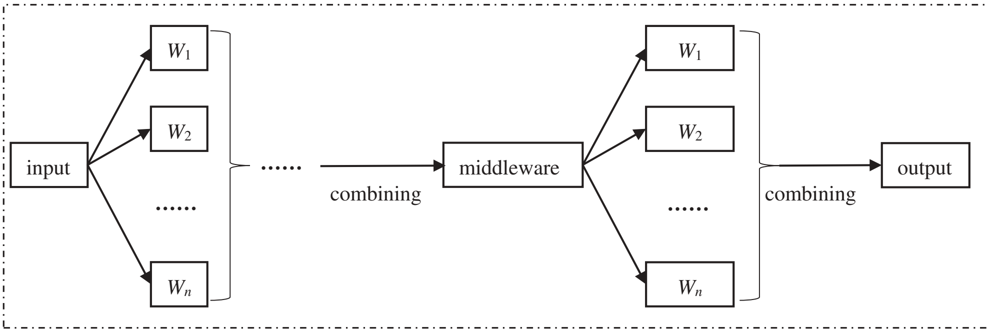

Figure 3. Schema of the parallel-dominated iterative model with feedback.

results are used to spark self-examination and enlighten other workers. The whole process is shown in Figure 3.

3.2. Evaluation criteria of the crowdsourcing model

Precision is used to evaluate the correct proportion of the recognised semantic relations; *Recall* is used to evaluate the recognition ratio of the semantic relations, and *F-measure* is the weighted measure of both *Precision* and *Recall*. Moreover, when combining the results of the parallel part, *Agreement* is a parameter that influences the quality of combined results.

Definition 1. (*Precision*). The ratio of the number of correct recognised semantic relations and the number of recognised semantic relations: $Precision = \frac{|E \cap V|}{|V|}$,

where E is the reference set identified by experts and V is the set recognised by crowdsourcing workers.

Definition 2. (*Recall*). The ratio of the number of correct recognised semantic relations and the number of semantic relations: $Recall = \frac{|E \cap V|}{|E|}$.

Definition 3. (*F-measure* (Baeza-Yates and Ribeiro-Neto 1999)). *F-measure* enables us to measure the trade-off between *Precision* and *Recall*: $F - measure = \frac{Precision * Recall}{Precision + Recall}$.

Definition 4. (*Agreement*). When m workers among the whole n workers approve the same semantic relations, then these semantic relations are recognised as the result, denoted as *Agreement*= m ($m \leq n$).

4. Mining semantic association rules

4.1. Producing dependency structures by the stanford parser

The Stanford dependency structure is a practical representation of English syntax, aims at natural language understanding applications and represents the relationship between

المنارة للاستشارات
www.

two words. De Marneffe et al. (2014) gives a more rigorous discussion about dependency structures. Some NLP (Natural Language Processing) literature suggests that the dependency structure is more stable for the relation extraction (Nakashole, Weikum, and Suchanek 2012), and the Stanford parser (http://nlp.stanford.edu: 8080/parser/) is a frequently used tool to obtain the dependency structures. Therefore, we apply the Stanford parser to obtain the dependency structures from the natural language questions. Figure 4 shows the dependency structures for question '*How many books by Kerouac were published by Viking Press?*' (e.g., the dependency structure '*amod(books-3, many-2)*' represents an adjectival modifier of a noun).

4.2. Marking semantic relations by our crowdsourcing model

There are many crowdsourcing platforms, such as AMT (Amazon Mechanical Turk) and MobileWorks, which provide APIs (application programming interfaces) for easily calling many workers to complete microtasks (called human intelligent tasks (HITs)). To mark semantic relations in the natural language questions by our crowdsourcing model (i.e., the parallel-dominated iterative model with feedback), we create a HIT, as shown in Figure 5, and publish it to the crowdsourcing platform AMT. For the HIT, the workers are required to write the semantic relation (i.e., subject, relation phrase, object) when they find semantic

advmod(many-2, How-1)
amod(books-3, many-2)
nsubjpass(published-7, books-3)
case(Kerouac-5, by-4)
nmod:by(books-3, Kerouac-5)
auxpass(published-7, were-6)
root(ROOT-0, published-7)
case(Press-10, by-8)
compound(Press-10, Viking-9)
nmod:by(published-7, Press-10)

Figure 4. The dependency structures of a sample question.

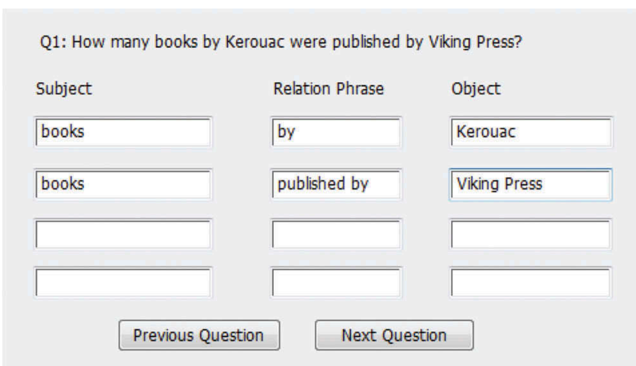

Figure 5. A sample HIT for marking semantic relations in a question.

المنارة للاستشارات
www.

relations in a question, and then address next question by button '*Next Question*' or modify previous question by the button '*Previous Question*'. After all, workers have completed all HITs in the form of the parallel-dominated iterative model with feedback, we obtain the dataset of the semantic relations in all questions.

DEFINITION 5. (*Semantic Relation, R*). A semantic relation is a triple that represents part of a user's query intention, denoted as *R*< *arg*1, *rel, arg*2>, where *rel* is a relation phrase and *arg*1 and *arg*2 are two associated arguments.

Example 2. For the question '*How many books by Kerouac were published by Viking Press?*', <*books, published, Viking_Press*> is a semantic relation, in which '*published*' is the relation phrase *rel*, and '*books*' and '*Viking_Press*' are two associated arguments *arg1* and *arg2*, respectively. We can also find another semantic relation <*books, by, Kerouac*> in the question.

4.3. Algorithm of mining semantic association rules

Definition 6. (*Semantic Association Rule*). A semantic association rule is an expression $X \Rightarrow Y$, where X is an itemset that consists of one or two dependency structure(s), and Y is a semantic relation in a question.

Example 3. For the question '*How many books by Kerouac were published by Viking Press?*', the dependency structures of the question can produce two semantic relations, so we can produce semantic association rules '{*nmod:by(books, Kerouac)*} ⇒ *R*< *books, by, Kerouac*>' and '{*nsubjpass(published, books), nmod:by(published, Press)*} ⇒ *R*< *books, published, Viking_Press*>' that can be abbreviated to '{*nmod*} ⇒ *R*' and '{*nsubjpass, nmod*} ⇒ *R*', respectively. In addition, the latter contains two sub-rules (i.e., '{*nsubjpass*} ⇒ *R*' and '{*nmod*} ⇒ *R*').

Definition 7. (*Neighborhood Relation*). A dependency structure and a semantic relation satisfy a neighborhood relation, only if all phrases in the dependency structure exist in the semantic relation.

Definition 8. (*Candidate Itemset*). A candidate itemset consists of one or two dependency structure(s) and one semantic relation, and each dependency structure and the semantic relation satisfy the neighborhood relation.

Example 4. For the sample question, '{*nmod, R*}' (i.e., '{*nmod:by(books, Kerouac), R*< *books, by, Kerouac*>}') is a candidate itemset while '{*amod, R*}' (i.e., '{*amod(books, many), R*< *books, by, Kerouac*>}') is not, because the latter does not satisfy the neighborhood relation (i.e., the phrase '*many*' does not exist in the semantic relation '*R*< *books, by, Kerouac*>').

Definition 9. (*Participation Ratio* (*PR*)/*Participation Index* (*PI*)). The participation ratio of a dependency structure (denoted as dep_i) in a candidate itemset (denoted as c) is defined as:,

$$\mathrm{PR}(dep_i, \mathrm{c}) = \frac{\left|\pi_{dep_i}(c)\right|}{\left|\pi_{dep_i}(U)\right|}$$

المنارة للاستشارات
www.

where π is the relational projection operation with duplication elimination and U is the universal set of various dependency structures. And the participation index of the candidate itemset c is defined as:.

$$PI(c) = \min_{i=1}^{n} PR(dep_i,\ c)$$

Definition 10. (*Frequent Itemset*). c is a frequent itemset, only if $PI(c)$ are greater than the given minimum prevalence threshold *min_prev*, and in dependency structure dataset, the occurrence times of any dependency structure dep_i of c are greater than the given minimum occurrence threshold *min_count*, where *min_prev* and *min_count* were designated by experts.

Example 5. For the candidate itemset {*dobj, R*} in Figure 6, the number of occurrences of '*dobj*' in the candidate itemset is 1 (i.e., {*dobj(have-6, employees-3), R< google, have, employees>*}), and the number of occurrences of '*dobj*' in dependency structure dataset is 2 (i.e., '*dobj(Give-1, websites-4)*' and '*dobj(have-6, employees-3)*'). Therefore, the

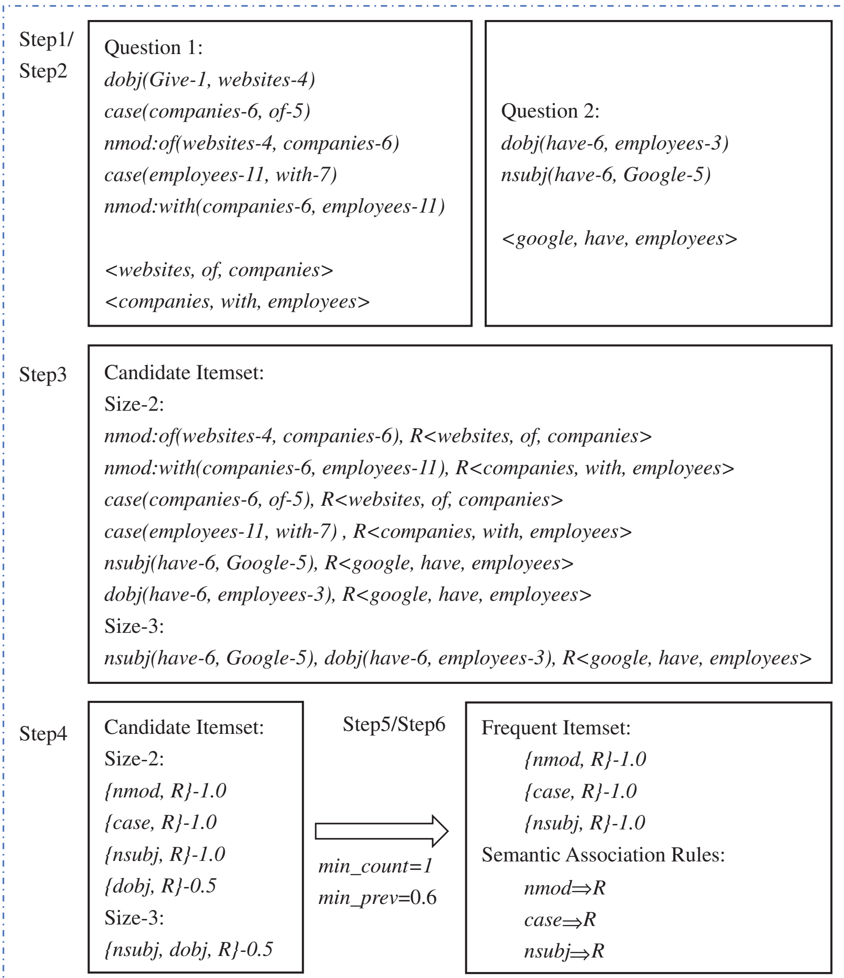

Figure 6. A sample for mining semantic association rules.

المنارة للاستشارات
www.

participation ratio of '*dobj*' in the candidate itemset c= {*dobj, R*} is 0.5. Furthermore, if *min_count*= 1 (2 > 1, true) and *min_prev*= 0.6 ($PI(c)$ = 0.5 < 0.6, false), the candidate itemset {*dobj, R*} is not a frequent itemset.

The semantic association rule in this paper is similar to, but not the same as traditional association analysis, so we only use the core idea of traditional association analysis and then design an algorithm according to the characteristics of semantic association rules.

We propose the algorithm of mining semantic association rules, as shown in Algorithm 1. The idea of the algorithm is as follows: first, we produce the dependency structure dataset and semantic relation dataset from the natural language questions by the Stanford parser and our crowdsourcing model, respectively (*step*1 and *step*2). Second, we obtain the candidate itemset from the above two datasets (*step*3), and then calculate the participation ratio and participation index for each candidate itemset (*step*4). Third, we obtain the frequent itemset (*step*5) and semantic association rules (*step*6). The example of Algorithm 1 is shown in Figure 6 and Example 6.

Example 6. For questions '*Question 1: Give me the websites of companies with more than 500,000 employees.*' and '*Question 2: How many employees does Google have?*', we can obtain dependency structures and semantic realtions from the two questions by the Stanford parser and our crowdsourcing model respectively (i.e., *step*1/*step*2 in Figure 6, where we omit some dependency structures that are irrelevant to Algorithm 1). In addition, then we produce a candidate itemset by the neighborhood relation (i.e., *step*3 in Figure 6). Afterwards, all candidate itemsets will be simplified, and *PR*/*PI* of them will be calculated (i.e., *step*4 in Figure 6). Finally, according to the given threshold *min_count* and *min_prev* by experts, we can obtain the frequent itemset and semantic association rules (i.e., *step*5/*step*6 in Figure 6).

4.4. Complexity analysis

Since we only use the core idea of traditional association analysis, and the maximum size of itemset is 3, the complexity in this paper is far lower than traditional association analysis. Moreover, to analyse the complexity, we suppose that there are N natural language questions, and the number of semantic relations and dependency structures in a question are R and D, respectively.

Algorithm 1. Mining Semantic Association Rules

Require: Input: the set of natural language questions δ_Q
minimum prevalence threshold *min_prev*
minimum occurrence threshold *min_count*
Output: semantic association rules δ_{SAR}

(1) δ_{dep}= Stanford_Parser(δ_Q)
(2) δ_R= Crowdsourcing(δ_Q)
(3) δ_c= Produce_candicate_itemset(δ_{dep}, δ_R)
(4) $\delta_{c\_with\_PI}$= Caculate_PR_and_PI(δ_c)
(5) $\delta_{frequent\_c}$= Produce_frequent_itemset($\delta_{c\_with\_PI}$, *min_prev*, *min_count*)
(6) δ_{SAR}= Produce_rules($\delta_{frequent\_c}$)

المنارة للاستشارات
www.

Time complexity. As we know (Cer et al. 2010; Zou et al. 2014), the time complexity of the Stanford parser for each question is $O(R^3)$, so the time complexity in *step*1 is $O(N^*R^3)$. In *step*2, for each question, four crowdsourcing workers recognise semantic relations twice, whose time complexity is $O(8^*N) = O(N)$. In *step*3, for each dependency structure, we analyse whether semantic relations contain it or not; therefore, its time complexity is $O(N^*R^*D)$. In *step*4, we count the number of occurrences of all dependency structures ($O(N^*D)$) and that of all candidate itemsets ($O(N^*R^*3) = O(N^*R)$, because each semantic relation can associate with three candidate itemsets), and then calculate *PR* and *PI* for each candidate itemset ($O(N^*R)$). In *step*5, we delete all candidate itemsets that are not frequent ($O(N^*R)$). In *step*6, we produce semantic association rules for all frequent itemsets ($O(N^*R)$). In conclusion, the time complexity of the algorithm of mining semantic association rules is $O(N^*R^3) + O(N^*R^*D)$.

Space complexity. Obviously, the space complexity of storing natural language questions, semantic relations and dependency structures are $O(N)$, $O(R)$ and $O(D)$, respectively. Moreover, since each semantic relation can associate with three candidate itemsets and the number of frequent itemsets is less than that of candidate itemsets, the space complexity of storing *PR*/*PI* for each candidate itemset and semantic association rules for each frequent itemset are $O(3^*R^*2) = O(R)$ and $O(3^*R) = O(R)$, respectively. In conclusion, the space complexity of the algorithm of mining semantic association rules is $O(N+ R+ D)$.

5. Semantic relation recognition

5.1. Algorithm of semantic relation recognition

Based on the semantic association rules, we propose an algorithm of semantic relation recognition for natural language questions over RDF data.

Definition 11. (*subject-like, object-like, triple-like*). There are three kinds of Stanford dependency structures that can be used to produce a semantic relation. If two phrases in a dependency structure are *arg*1 (or *arg*2) and *rel* in a semantic relation, then the dependency structure belongs to *subject-like* (or *object-like*). Similarly, if two phrases in a dependency structure are *arg*1 and *arg*2 in a semantic relation, the dependency structure belongs to *triple-like*.

Example 7. For the semantic association rule '{*nsubjpass, nmod*}⇒*R*' (e.g., '{*nsubjpass(published, books), nmod:by(published, Viking_Press)*} ⇒*R*< *books, published, Viking_Press*>'), '*nsubjpass*' and '*nmod*' belong to *subject-like* and *object-like*, respectively. For the semantic association rule '{*nmod*}⇒*R*' (e.g., '{*nmod:by(books, Kerouac)*}⇒*R*< *books, by, Kerouac*>'), '*nmod*' belongs to *triple-like*.

After mining semantic association rules, we need to determine how to obtain a semantic relation by dependency structures. From the experimental result in section 6.2, we obtain three categories of dependency structures as shown in Table 1. Before mining semantic association rules, many dependency structures have a clear category, '*subj, nsubj, nsubjpass, csubj, csubjpass, xsubj*' and '*obj, pobj, dobj, iobj*' belong to *subject-like* and *object-like*, respectively. Besides, we find some new dependency structures

المنارة للاستشارات
www.

Table 1. Three categories of dependency structures.

Categories	Dependency structures
$\delta_{subject-like}$	*subj, nsubj, nsubjpass, csubj, csubjpass, xsubj, acl*
$\delta_{object-like}$	*obj, pobj, dobj, iobj, dep, advmod, nmod*
$\delta_{triple-like}$	*nmod*

which can be used to produce semantic relations (i.e., '*acl*' and '*dep/advmod*' belong to *subject-like* and *object-like* respectively, and '*nmod*' belongs not only to *object-like* but also to *triple-like*).

Based on the three categories in Table 1, we have determined the combination order of dependency structures in the process of producing the semantic relation. At first, we combine the dependency structures in $\delta_{subject-like}$ and $\delta_{object-like}$. Then, we combine the dependency structures in $\delta_{subject-like}$ and $\delta_{triple-like}$ (i.e., '*nmod*') because some instances of '*nmod*' belong to *object-like*. Finally, we produce semantic relations from $\delta_{triple-like}$ (i.e., the remaining instances of '*nmod*'). The detailed combination rules are as follows:

(1) $R(s,p,o)=\mathrm{f}(\delta_{subject-like} \wedge \delta_{object-like})$
(2) $R(s,p,o)=\mathrm{f}\left(\delta_{subject-like} \wedge \delta_{triple-like}\right)$
(3) $R(s,p,o)=\mathrm{f}(\delta_{triple-like})$

Moreover, a phrase in a semantic relation may contain more than one word, such as the entity '*Viking Press*' and the relation phrase '*official_langauge*'; therefore, we need to combine these words by a dependency structure '*compound*'. Consequently, we proposed the algorithm of semantic relation recognition as follows:

5.2. Complexity analysis

To analyse the complexity, we consider that the number of semantic relations and dependency structures in a question are *R* and *D*, respectively.

Time complexity. As we know (Cer et al. 2010; Zou et al. 2014), in *step*1, the time complexity of the Stanford parser for one question is $O(R^3)$. From *step*2 to *step*4, obtaining and updating the dependency structure requires scanning all dependency structures twice, and the number of phrases that need to be compounded is approximately three, so that the time complexity is $O(D*2*3) = O(D)$. From *step*5 to *step*7,

Algorithm 2. Semantic Relation Recognition

Require: Input: A natural language question *N*
Output: the set of semantic relations δ_R

(1) δ_{dep}= Stanford_Parser(*N*)
(2) $\delta_{compound}$= Get_dependency_structures(δ_{dep})
(3) $\delta_{compound\_phrase}$= Combine_words($\delta_{compound}$)
(4) δ_{dep}= Update_dependency_structures($\delta_{compound\_phrase}$, δ_{dep})
(5) $\delta_{subject-like}$= Get_dependency_structures(δ_{dep})
(6) $\delta_{object-like}$= Get_dependency_structures(δ_{dep})
(7) $\delta_{triple-like}$= Get_dependency_structures(δ_{dep})
(8) δ_R= Combine($\delta_{subject-like}$, $\delta_{oject-like}$)
(9) δ_R=δ_R+ Combine($\delta_{subject-like}$+$\delta_{triple-like}$)

(1) δ_R=δ_R+ Combine($\delta_{triple-like}$)

المنارة للاستشارات
www.

classifying dependency structures only needs to scan all dependency structures once, so the time complexity is $O(D)$. From *step*8 to *step*10, for the worst case scenario, any two of all dependency structures can be combined in *step*8 and *step*9, and all dependency structures can produce semantic relations in *step*10, so the time complexity is $O(D^*D^*2 + D) = O(D^2)$. In conclusion, the time complexity of the algorithm of semantic relations recognition is $O(R^3) + O(D^2)$.

Space complexity. Obviously, the space complexity of storing original dependency structures, dependency structures that have been classified and semantic relations are $O(D)$, $O(D)$ and $O(R)$, respectively. Therefore, the space complexity of the algorithm of semantic relations recognition is $O(D+ R)$.

6. Experiments of crowdsourcing model and mining semantic association rules

First, we compare our proposed crowdsourcing model (i.e., the parallel-dominated iterative model with feedback) with two common crowdsourcing models (i.e., the iterative model and the parallel model) over a small number of questions, and experimental results demonstrate that our model improves the quality of results and reduces the number of workers. Second, we mine semantic association rules from the correlations between dependency structures and semantic relations, where the former and the latter be generated from the natural language questions by the Stanford parser and our proposed crowdsourcing model, respectively. Third, we compare our algorithm of semantic relation recognition with existing methods, and experimental results demonstrate that it can recognise more semantic relations than existing methods.

6.1. Crowdsourcing model comparison

In the comparative experiments of three crowdsourcing models, the task of the crowdsourcing model is to recognise semantic relations in 20 natural language questions that can be selected randomly from QALD-3 questions. The reference data is provided by RDF query experts.

We recruited eight workers to recognise semantic relations by the iterative model and the parallel model, and experimental results are shown in Figures 7 and 8, respectively. For the iterative model, the subsequent workers will explore a new answer (occasionally insincere or malicious workers may appear), so the *Recall* curve rises slightly (i.e., it has relatively high completeness). For the parallel model, the result reflects the high

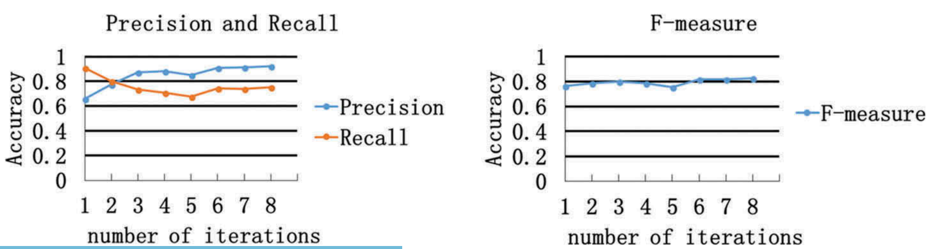

Figure 7. The capability of the iterative model.

المنارة للاستشارات
www.

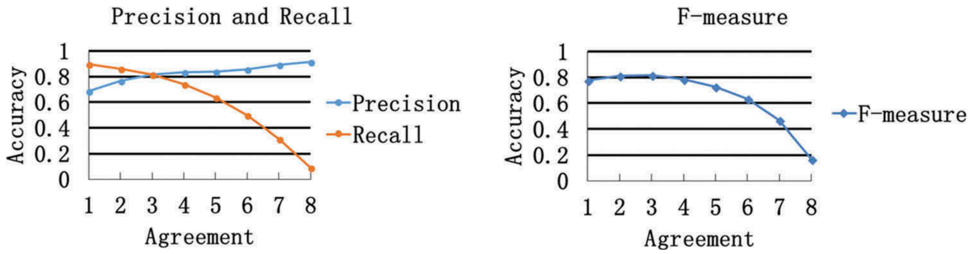

Figure 8. The capability of the parallel model.

consensus (i.e., it has relatively high accuracy), and both the *F-measure* and the *Recall* curves fall as *Agreement* changes from one to eight because it is getting more difficult to forge consensus for more workers.

To ensure the comparability, we recruited four workers (i.e., they form a unique group) to recognise semantic relations two times (i.e., two iterations) by our crowdsourcing model (i.e., the parallel-dominated iterative model with feedback), and experimental results are shown in Figure 9. For our model, the result in the second iteration is more accurate and stable than the result in the first iteration. Furthermore, compared with two common crowdsourcing models (i.e., the iterative model and the parallel model), as shown in Figure 10, our model improves the quality of the results and reduces the number of human resources (i.e., four workers) to complete the same task.

6.2. Mining semantic association rules

In addition to common sense dependency structures, we find some new ones. Before mining semantic association rules, many dependency structures have a clear category (i.e., '*subj, nsubj, nsubjpass, csubj, csubjpass, xsubj*' and '*obj, pobj, dobj, iobj*' belong to *subject-like* and *object-like*, respectively). After mining semantic association rules, we find some new dependency structures as shown in Table 2 (i.e., '*acl*' belongs to *subject-like*, '*case/dep*' belongs to *object-like*, and '*nmod*' belongs to *object-like* and *triple-like*), where *min_prev* and *min_count* were designated by experts. We also show *PI*(*PR*) of each candidate itemset in Table 2 (there is only one dependency structure in each itemset so that *PI* and *PR* are equal). For example, '*nsubjpass*-0.97(31/32)' means that there are 32 instances of '*nsubjpass*' in the dependency structure dataset, and 31 instances belong to *subject-like*, so that $PI(c)$ = $PR(nsubjpass,\ c)$ = 0.97, where the candidate itemset c= {*nsubjpass, R*}.

In addition, we also find some dependency structure combinations that can be used to produce semantic relations, as shown in Table 3. Because the aim of analysing these combinations is to understand the combinational rules from the dependency structure to the semantic relation, both *min_prev* and *min_count* have not been set, and whether a combination is frequent or not depends on each dependency structure in it. We also show *PI*(*PR*) of each candidate itemset in Table 3, for example, '*nsubj, dobj*-0.41(56/136| 56/108)' means that 56 instances of the combination '*nsubj, dobj*' can produce semantic relations, and there are 136 and 108 instances of '*nsubj*' and '*dobj*' in the dependency structure dataset, respectively, so *PRs* of '*nsubj*' and '*dobj*' in the candidate itemset *c*=

المنارة للاستشارات
www.

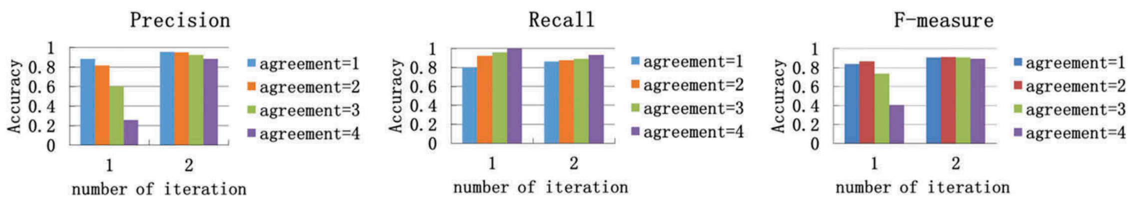

Figure 9. The capability of the parallel-dominated iterative model with feedback.

www.

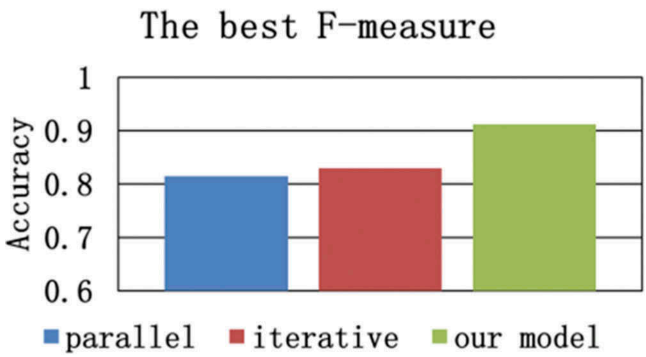

Figure 10. The best F-measure of three models.

Table 2. Three kinds of Stanford dependency structures.

	subject-like	PI(PR)	object-like	PI(PR)	triple-like	PI(PR)
Frequent	*nsubjpass*	0.97(31/32)	*case*	0.57(95/166)	*nmod*	0.52(89/172)
(*min_prev*=0.2	*nsubj*	0.73(99/136)	*dobj*	0.56(61/108)		
and	*acl*	0.89(17/19)	*dep*	0.38(6/16)		
min_count>15)			*nmod*	0.36(62/172)		
			advmod	0.21(9/42)		
Non-Frequent	*expl*	0.33(1/3)	*expl*	0.33(1/3)	*nsubj*	0.06(8/136)
	dep	0.19(3/16)	*xcomp*	0.33(2/6)		
	case	0.03(5/166)	*cop*	0.12(5/43)		
	cop	0.07(3/43)	*acl*	0.05(1/19)		
	dobj	0.01(1/108)	*iobj*	0.02(1/42)		
	nmod	0.01(1/172)				

Table 3. Dependency structure combinations.

	combinations	PI(PR)
Frequent	*nsubj, dobj*	0.41(56/136|56/108)
	nsubj, nmod	0.16(28/136|28/172)
	nsubjpass, nmod	0.13(22/32|22/172)
	nsubjpass,advmod	0.10(4/32|4/42)
	acl, nmod	0.08(14/19|14/172)
	nsubj, dep	0.04(5/136|4/16)
	nsubj, advmod	0.04(5/136|5/42)
	nsubjpass, dep	0.03(1/32|1/16)
	acl, dobj	0.02(2/19|2/108)
	nsubj, iobj	0.01(1/136|1/42)
	nsubjpass, dobj	0.01(1/32|1/108)
Non-Frequent	*case, case*	0.03(5/166|5/166)
	nsubjpass, acl	0.03(1/32|1/39)
	nsubjpass, xcomp	0.03(1/32|1/16)
	dep, nmod	0.01(2/16|2/172)
	nmod, dobj	0.01(1/172|1/108)
	dep, dobj	0.01(1/16|1/108)
	nsubj, expl	0.01(1/136|1/3)
	nsubj, xcomp	0.01(1/136|1/6)

{*nsubj, dobj, R*} are '56/136' and '56/108', respectively, and *PI* of the candidate itemset *c* is 0.41 (i.e., 56/136).

المنارة للاستشارات
www.

For the dependency structure '*case*' in Table 2, although it belongs to *object-like*, its instances cannot produce a semantic relation or can be replaced by a dependency structure '*nmod*'. First, there are 156 instances of '*case*' that belong to *object-like*, but all of them cannot produce a semantic relation (i.e., there is no another dependency structure that can be combined with instance of '*case*'), and their corresponding semantic relations can be produced by '*nmod*', such as the semantic relation '*R< cities, in, New_Jersey>*', as shown in Table 4. Second, there are five instances of '*case*' that cannot be replaced by '*nmod*', but all of them also cannot produce a semantic relation, such as the semantic relation '*R< games, by, GMT>*', as shown in Table 4. Third, there are five instances of the semantic association rule '*case, case*□ *R*', as shown in Table 3, but all instances can be replaced by '*nmod*' such as the semantic relation '*R< cinemas, in, Netherlands>*', as shown in Table 4.

For the dependency structure '*iobj*' in Table 2, it is simply a coincidence that only one instance of '*iobj*' belongs to *object-like*, because there are 41 questions, such as '*Give me ……*', so that most of its instances are '*iobj(Give, me)*' which does not belong to *object-like*. Namely, with the exception of '*iobj(Give, me)*', all other instances of '*iobj*' belong to *object-like*.

For the dependency structure '*nmod*' in Table 2, it is the most dramatic finding for semantic relation recognition. On the one hand, 89 semantic relations can be produced by the dependency structure '*nmod*', as shown in *triple-like* in Table 2. On the other hand, 50 semantic relations can be produced by combining the dependency structure '*nmod*' and others (i.e., '*nsubj, nsubjpass, acl*'), as shown in Table 3.

6.3. Semantic relation recognition comparison

We compare our method with other methods that try to improve semantic relation recognition, as shown in Table 5, and the benchmark question dataset originates from QALD-3 training questions and testing questions, which contain 198 questions. Our method can recognise 221 sematic relations of 256 semantic relations, and 176 questions (i.e., all semantic relations in the questions can be recognised) of 198 questions. It is obvious that our method is better than other methods (Yahya et al. 2012a, 2012b, 2013; Yahya 2016; Zou et al. 2014; Liu et al. 2017), because their methods rely on the

Table 4. A sample of the dependency structure '*case*'.

	case	*nmod*	Semantic realtion	Count
object-like	*case(New_Jersey, in)*	*nmod:in(cities, New_Jersey)*	*<cities, in, New_Jersey>*	*156*
	case(GMT, by)	*nothing*	*<games, by, GMT>*	*5*
case, case ⇒R	*case(cinemas, in)& case(Netherlands, in)*	*nmod:in(cinemas, Netherlands)*	*cinemas, in, Netherlands*	*5*

Table 5. Comparison of several algorithms about semantic relation recognition.

Algorithm	*R*	questions	Core competitiveness
Our method	221(256)	176(198)	rules that be mined from data
DEANNA	68	46	verbal phrases, noun phrases and RDF data
Zou et al. (2014)	109	69	verbal phrases, noun phrases and heuristic linguistic rules
Liu et al. (2017)	126	96	*nsubj, nsubjpass, dobj, iobj, nmod:of, nmod:pos, nmod:by, amod*

المنارة للاستشارات
www.

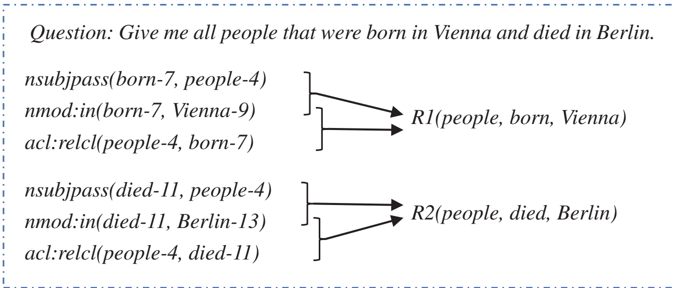

Figure 11. A sample for producing a semantic relation from two methods.

rules that come from an artificial hypothesis while our method relies on the rules that can be mined from data.

In addition, there are 139 frequent dependency structure combinations in Tables 3 and 89 instances of '*nmod*' that belong to *triple-like* in Table 2; apparently, all of them can produce 228 semantic relations. The reason our method only produce 221 semantic relations in Table 5 is that some semantic relations can be produced from two methods.

Example 7. For the question '*Give me all people that were born in Vienna and died in Berlin.*', there are dependency structures '*nsubjpass(born-7, people-4)/acl:relcl(people-4, born-7)*' and '*nmod:in(born-7, Vienna-9)*'. Both the dependency structure combinations '*nsubjpass, nmod*' and '*acl, nmod*' can produce the semantic relation *R< people, born, Vienna>* as well as semantic relations *R< people, died, Berlin>*, as shown in Figure 11.

7. Conclusion and future work

We propose a new crowdsourcing model (i.e., the parallel-dominated iterative model with feedback) which improves the quality of the results by inheriting the completeness/accuracy of the iterative/parallel model and saves human resources, and the experimental results demonstrate that our model is better than two existing models. For the iterative model, *n* workers perform the same task in succession, and the later workers will continue to explore a new result so that it has relatively high completeness. For the parallel model, *n* workers complete the same task independently; the result is achieved by the high consensus of workers so that it has relatively high accuracy. Compared with two existing models, on the one hand, our model is a blend of the iterative model and the parallel model so that the parallel part of our model can avoid generating many wrong results (i.e., the defect of the iterative model), and the iterative part of our model can avoid missing results (i.e., the defect of the parallel model). Therefore, our model can generate a greater completeness of solutions and constant improvement of the accuracy of the results. On the other hand, we consider the characteristic that workers always care about their own mistakes pointed out by others rather than others' mistakes; therefore, a unique group of workers performs the same task iteratively in our model, which improves the ability of workers to obtain better results and saves human resources. Moreover, compared with the iterative/parallel model as shown in Figure 10, our

المنارة للاستشارات
www.

model improves the quality of the results and reduces the number of human resources (i.e., four workers) to complete the same task.

We mine the semantic association rules from the correlations between dependency structures and semantic relations. First, we obtain all dependency structures from the natural language questions by the Stanford parser and obtain all semantic relations from the natural language questions by our crowdsourcing model (i.e., the parallel-dominated iterative model with feedback). Then, based on the core idea of association analysis and the characteristics of semantic association rules, we propose an algorithm of mining the semantic association rules from the dependency structures dataset and the semantic relations dataset. Finally, we obtain many meaningful semantic association rules, such as the association between the dependency structure and the semantic relation (i.e., *subject-like, object-like, triple-like* ⇒*R*), as shown in Table 2, and the association between the dependency structure combination and the semantic relation (e.g., *nsubj, nmod* ⇒*R*), as shown in Table 3.

According to the rules above, we propose an algorithm of semantic relation recognition for natural language questions over RDF data, and the experimental results demonstrate that it is better than existing methods. First, it produces semantic relations by combining the dependency structures in *subject-like* with the dependency structures in *object-like*. Second, it produces semantic relations by combining the dependency structures in *subject-like* or *object-like* with the dependency structures in *triple-like* (i.e., '*nmod*') because some instances of '*nmod*' belong to *object-like*. Third, it produces semantic relations from the remaining dependency structures in *triple-like* (i.e., the rest of the instances of '*nmod*'). Because the rules are mined from data while the rules in existing methods are derived from an artificial hypothesis, we can recognise more semantic relations than existing methods for natural language questions over RDF data, as shown in Table 5.

Overall, we proposed crowdsourcing-based semantic relation recognition for natural language questions over RDF data, which mine semantic association rules from the dependency structure dataset that is produced by the Stanford parser and the semantic relation dataset that is produced by our proposed crowdsourcing model (i.e., the parallel-dominated iterative model with feedback), and then recognise the semantic relations in a natural language question by the rules above. Comparing the rules of semantic relation recognition in existing methods that originate from an artificial hypothesis, our rules are derived from data so that they are more reliable and can recognise the semantic relations in most of the natural language questions.

There are some related issues that are worth studying in the future. 1) Although the dependency structure '*compound*' represents that the phrase (i.e., relation/entity/class) may contain more than one word, '*compound*' does not appear in some cases, so the accuracy of the Stanford parser still needs to be improved. 2) Not all query intentions in a natural language question can be represented by a semantic relation; therefore, it will be necessary to find a method to address this case. 3) Although our method has recognised most of the semantic relations, there are still many problems that need to be solved for the translation from a semantic relation to a SPARQL statement, such as mapping and disambiguation.

المنارة للاستشارات
www.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Natural Science Foundation of China [Project No. 61672102, No. 60940032, No. 61073034 and No. 61370064]; the Program for New Century Excellent Talents in University of Ministry of Education of China under Grant No. NCET-10–0239; the National Key Technology Research and Development Program of China [No.2017YFB1002804].

ORCID

Xin Hu http://orcid.org/0000-0002-4258-6746

References

Abujabal, A., M. Yahya, M. Riedewald, and G. Weikum. 2017. “Automated Template Generation for Question Answering over Knowledge Graphs.” Proceedings of the 26th International Conference on World Wide Web, 1191–1200. doi:10.1145/3038912.3052583.

Amsterdamer, Y., S. B. Davidson, T. Milo, S. Novgorodov, and A. Somech. 2014. “OASSIS: Query Driven Crowd Mining.” Proceedings of the 2014 ACM SIGMOD international conference on Management of data, 589–600. doi:10.1145/2588555.2610514.

Amsterdamer, Y., A. Kukliansky, and T. Milo. 2015a. “A Natural Language Interface for Querying General and Individual Knowledge.” *Proceedings of the VLDB Endowment* 8 (12): 1430–1441. doi:10.14778/2824032.2824042.

Amsterdamer, Y., A. Kukliansky, and T. Milo. 2015b. “NL2CM: A Natural Language Interface to Crowd Mining.” Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 1433–1438. doi:10.1145/2723372.2735370.

Baeza-Yates, R., and B. Ribeiro-Neto. 1999. *Modern Information Retrieval*, ch.3, 81–82. New York: ACM press.

Balakrishna, M., S. Werner, M. Tatu, T. Erekhinskaya, and D. Moldovan. 2016. “K-Extractor: Automatic Knowledge Extraction for Hybrid Question Answering.” Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on, 390–391. doi:10.1109/ICSC.2016.30.

Basili, R., D. Croce, and G. Castellucci. 2017. “Dynamic Polarity Lexicon Acquisition for Advanced Social Media Analytics.” *International Journal of Engineering Business Management* 9: 1–18. doi:10.1177/1847979017744916.

Cer, D. M., M. C. D. Marneffe, D. Jurafsky, and C. D. Manning. 2010. “Parsing to Stanford Dependencies: Trade-Offs between Speed and Accuracy.” International Conference on Language Resources and Evaluation (LREC), 1628–1632.

Chen, F., M. Li, H. Wu, and L. Xie. 2015. “Web Service Discovery among Large Service Pools Utilising Semantic Similarity and Clustering.” *Enterprise Information Systems* 11 (3): 452–469. doi:10.1080/17517575.2015.1081987.

Conley, C., and J. Tosti-Kharas. 2014. “Crowdsourcing Content Analysis for Managerial Research.” *Management Decision* 52 (4): 675–688. doi:10.1108/MD-03-2012-0156.

De Marneffe, M. C., T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, and C. D. Manning. 2014. “Universal Stanford Dependencies: A Cross-Linguistic Typology.” International Conference on Language Resources and Evaluation (LREC), 4585–4592.

Demartini, G., D. E. Difallah, and P. Cudré-Mauroux. 2013. “Large-Scale Linked Data Integration Using Probabilistic Reasoning and Crowdsourcing.” *The VLDB Journal* 22 (5): 665–687. doi:10.1007/s00778-013-0324-z.

المنارة للاستشارات

www.

Doan, A., R. Ramakrishnan, and A. Y. Halevy. 2011. "Crowdsourcing Systems on the World-Wide Web." *Communications of the ACM* 54 (4): 86–96. doi:10.1145/1924421.1924442.
Dubey, M., S. Dasgupta, A. Sharma, K. Höffner, and J. Lehmann. 2016. "Asknow: A Framework for Natural Language Query Formalization in SPARQL." International Semantic Web Conference, 300–316. doi:10.1007/978-3-319-34129-3_19.
El-Ansari, A., A. Beni-Hssane, and M. Saadi. 2017. "A Multiple Ontologies Based System for Answering Natural Language Questions." Europe and MENA Cooperation Advances in Information and Communication Technologies, 177–186. doi:10.1007/978-3-319-46568-5_18.
Elshendy, M., and A. F. Colladon. 2017. "Big Data Analysis of Economic News: Hints to Forecast Macroeconomic Indicators." *International Journal of Engineering Business Management* 9: 1–12. doi:10.1177/1847979017720040.
Fader, A., L. Zettlemoyer, and O. Etzioni. 2014. "Open Question Answering over Curated and Extracted Knowledge Bases." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 1156–1165. doi:10.1145/2623330.2623677.
Ferré, S. 2013. "Squall2sparql: A Translator from Controlled English to Full SPARQL 1.1." Work. Multilingual Question Answering over Linked Data (QALD-3).
Ferré, S. 2014. "SQUALL: The Expressiveness of SPARQL 1.1 Made Available as a Controlled Natural Language." *Data & Knowledge Engineering* 94: 163–188. doi:10.1016/j.datak.2014.07.010.
Freitas, A., and E. Curry. 2014. "Natural Language Queries over Heterogeneous Linked Data Graphs: A Distributional-Compositional Semantics Approach." Proceedings of the 19th international conference on Intelligent User Interfaces, 279–288. doi:10.1145/2557500.2557534.
Friess, S. 2007. "50,000 Volunteers Join Distributed Search for Steve Fossette." *Wired News*, September 11. 10.1094/PDIS-91-4-0467B
Habernal, I., and M. KonopíK. 2013. "SWSNL: Semantic Web Search Using Natural Language." *Expert Systems with Applications* 40 (9): 3649–3664. doi:10.1016/j.eswa.2012.12.070.
Hafner, K. 2007. "Silicon Valleys High-Tech Hunt for Colleague." *New York Times*, February 3. 10.1094/PDIS-91-4-0467B
Hamon, T., N. Grabar, and F. Mougin. 2017. "Querying Biomedical Linked Data with Natural Language Questions." *Semantic Web* 8 (4): 581–599. doi:10.3233/SW-160244.
Hoang, H. H., J. J. Jung, and C. P. Tran. 2014. "Ontology-Based Approaches for Cross-Enterprise Collaboration: A Literature Review on Semantic Business Process Management." *Enterprise Information Systems* 8 (6): 648–664. doi:10.1080/17517575.2013.767382.
Höffner, K., J. Lehmann, and R. Usbeck. 2016. "CubeQA – Question Answering on RDF Data Cubes." International Semantic Web Conference, 325–340. doi:10.1007/978-3-319-46523-4_20.
Horton, J. J., and L. B. Chilton. 2010. "The Labor Economics of Paid Crowdsourcing." Proceedings of the 11th ACM conference on Electronic commerce, 209–218. doi:10.1145/1807342.1807376.
Howe, J. 2006. "The Rise of Crowdsourcing." *Wired Magazine* 14 (6): 1–4.
Hsueh, P. Y., P. Melville, and V. Sindhwani. 2009. "Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria." Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing, 27–35.
Hu, C., Z. Xu, Y. Liu, L. Mei, L. Chen, and X. Luo. 2014. "Semantic Link Network-Based Model for Organizing Multimedia Big Data." *IEEE Transactions on Emerging Topics in Computing* 2 (3): 376–387. doi:10.1109/TETC.2014.2316525.
Huberman, B. A., D. M. Romero, and F. Wu. 2009. "Crowdsourcing, Attention and Productivity." *Journal of Information Science* 35 (6): 758–765. doi:10.2139/ssrn.1266996.
Ipeirotis, P. G., F. Provost, and J. Wang. 2010. "Quality Management on Amazon Mechanical Turk." Proceedings of the ACM SIGKDD workshop on human computation, 64–67. doi:10.1145/1837885.1837906.
Jagadeesan, A. P., A. Lynn, J. R. Corney, X. T. Yan, J. Wenzel, A. Sherlock, and W. Regli. 2009. "Geometric Reasoning via Internet Crowdsourcing." 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling, 313–318. doi:10.1145/1629255.1629296
Jorda, J., M. R. Sawaya, and T. O. Yeates. 2014. "CrowdPhase: Crowdsourcing the Phase Problem." *Acta Crystallographica Section D: Biological Crystallography* 70 (6): 1538–1548. doi:10.1107/S1399004714006427.

المنارة للاستشارات

www.

Koblin, A. M. 2009. "The Sheep Market." Proceedings of the seventh ACM conference on Creativity and cognition, 451–452. doi:10.1145/1640233.1640348.
Li, J., J. J. Yang, C. Liu, Y. Zhao, B. Liu, and Y. Shi. 2014. "Exploiting Semantic Linkages among Multiple Sources for Semantic Information Retrieval." *Enterprise Information Systems* 8 (4): 464–489. doi:10.1080/17517575.2013.879923.
Liu, J., W. Li, L. Luo, J. Zhou, X. Han, and J. Shi. 2017. "Linked Open Data Query Based on Natural Language." *Chinese Journal of Electronics* 26 (2): 230–235. doi:10.1049/cje.2016.11.003.
Lopez, V., and E. Motta. 2004. "Ontology-Driven Question Answering in AquaLog." International Conference on Application of Natural Language to Information Systems, 89–102. doi:10.1007/978-3-540-27779-8_8.
Lopez, V., E. Motta, and V. Uren. 2006. "PowerAqua: Fishing the Semantic Web." The Semantic Web: Research and Applications, 3rd European Semantic Web Conference(ESWC), 393–410. 10.1016/j.ultsonch.2006.09.003
Lopez, V., M. Pasin, and E. Motta. 2005. "Aqualog: An Ontology-Portable Question Answering System for the Semantic Web." The Semantic Web: Research and Applications, Second European Semantic Web Conference(ESWC), 546–562.
Luo, J., B. Meng, C. Quan, and X. Tu. 2015. "Exploiting Salient Semantic Analysis for Information Retrieval." *Enterprise Information Systems* 10 (9): 959–969. doi:10.1080/17517575.2015.1080301.
MacLean, D. L., and J. Heer. 2013. "Identifying Medical Terms in Patient-Authored Text: A Crowdsourcing-Based Approach." *Journal of the American Medical Informatics Association* 20 (6): 1120–1127. doi:10.1136/amiajnl-2012-001110.
Maisonneuve, N., and B. Chopard. 2012. "Crowdsourcing Satellite Imagery Analysis: Study of Parallel and Iterative Models." International Conference on Geographic Information Science, 116–131. doi:10.1007/978-3-642-33024-7_9.
Mason, W., and D. J. Watts. 2010. "Financial Incentives and the Performance of Crowds." *ACM SigKDD Explorations Newsletter* 11 (2): 100–108. doi:10.1145/1809400.1809422.
Mazzeo, G. M., and C. Zaniolo. 2016. "Answering Controlled Natural Language Questions on RDF Knowledge Bases." 19th International Conference on Extending Database Technology (EDBT), 608–611. doi:10.5441/002/edbt.2016.60.
Mellebeek, B., F. Benavent, J. Grivolla, J. Codina, M. R. Costa-Jussa, and R. Banchs. 2010. "Opinion Mining of Spanish Customer Comments with Non-Expert Annotations on Mechanical Turk." Proceedings of the NAACL HLT 2010 workshop on Creating speech and language data with Amazon's mechanical turk, 114–121.
Nakashole, N., G. Weikum, and F. Suchanek. 2012. "PATTY: A Taxonomy of Relational Patterns with Semantic Types." Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 1135–1145.
Ngonga Ngomo, A. C., L. Bühmann, C. Unger, J. Lehmann, and D. Gerber. 2013. "Sorry, I Don'T Speak SPARQL: Translating SPARQL Queries into Natural Language." Proceedings of the 22nd international conference on World Wide Web, 977–988. doi:10.1145/2488388.2488473.
Ray, S. K., and K. Shaalan. 2016. "A Review and Future Perspectives of Arabic Question Answering Systems." *IEEE Transactions on Knowledge and Data Engineering* 28 (12): 3169–3190. doi:10.1109/TKDE.2016.2607201.
Rico, M., M. L. Caliusco, O. Chiotti, and M. R. Galli. 2015. "An Approach to Define Semantics for BPM Systems Interoperability." *Enterprise Information Systems* 9 (3): 324–347. doi:10.1080/17517575.2013.767381.
Robinson, W. N., A. Akhlaghi, T. Deng, and A. R. Syed. 2012. "Discovery and Diagnosis of Behavioral Transitions in Patient Event Streams." *ACM Transactions on Management Information Systems (TMIS)* 3 (1): 4:1–4:28. doi:10.1145/2151163.2151167.
Rozinajová, V., and P. Macko. 2016. "Using Natural Language to Search Linked Data." Semanitic Keyword-based Search on Structured Data Sources, 179–189. doi:10.1007/978-3-319-53640-8_16.
Salem, A. B., F. Boufares, and S. Correia. 2014. "Semantic Recognition of a Data Structure in Big-Data." *Journal of Computer and Communications* 2 (09): 93–102. doi:10.4236/jcc.2014.29013.

المنارة للاستشارات
www.

Scholten, P., J. Ji, H. Chen, and Y. Song. 2016. "A Natural Language Based Knowledge Representation Method for Medical Diagnosis." SAI Computing Conference (SAI), 32–37. doi:10.1109/SAI.2016.7555959.

Shekarpour, S., E. Marx, S. Auer, and A. P. Sheth. 2017. "RQUERY: Rewriting Natural Language Queries on Knowledge Graphs to Alleviate the Vocabulary Mismatch Problem." Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 3936–3943.

Sheng, V. S., F. Provost, and P. G. Ipeirotis. 2008. "Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 614–622. doi:10.1145/1401890.1401965.

Sun, H., H. Ma, W. Yih, C. T. Tsai, J. Liu, and M. W. Chang. 2015. "Open Domain Question Answering via Semantic Enrichment." Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 1045–1055. doi:10.1145/2736277.2741651.

Tatu, M., M. Balakrishna, S. Werner, T. Erekhinskaya, and D. Moldovan. 2016. "Automatic Extraction of Actionable Knowledge." Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on, 396–399. doi:10.1109/ICSC.2016.29.

Unger, C., L. Bühmann, and J. Lehmann. 2012. "Template-Based Question Answering over RDF Data." Proceedings of the 21st international conference on World Wide Web, 639–648. doi:10.1145/2187836.2187923.

Verroios, V., and M. S. Bernstein. 2014. "Context Trees: Crowdsourcing Global Understanding from Local Views." Second AAAI Conference on Human Computation and Crowdsourcing, Pittsburgh, PA, November 2-4, 210–219.

Warby, S. C., S. L. Wendt, P. Welinder, E. G. Munk, O. Carrillo, H. B. Sorensen, and E. Mignot. 2014. "Sleep-Spindle Detection: Crowdsourcing and Evaluating Performance of Experts, Non-Experts and Automated Methods." *Nature Methods* 11 (4): 385–392. doi:10.1038/nmeth.2855.

Yahya, M. 2016. "Question answering and query processing for extended knowledge graphs." PhD Thesis, Universitat des Saarlandes.

Yahya, M., K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum. 2012a. "Deep Answers for Naturally Asked Questions on the Web of Data." Proceedings of the 21st international conference on World Wide Web, 445–449. doi:10.1145/2187980.2188070.

Yahya, M., K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum. 2012b. "Natural Language Questions for the Web of Data." Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 379–390.

Yahya, M., K. Berberich, S. Elbassuoni, and G. Weikum. 2013. "Robust Question Answering over the Web of Linked Data." Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, 1107–1116. doi:10.1145/2505515.2505677.

Yan, T., V. Kumar, and D. Ganesan. 2010. "Crowdsearch: Exploiting Crowds for Accurate Real-Time Image Search on Mobile Phones." Proceedings of the 8th international conference on Mobile systems, applications, and services, 77–90. doi:10.1145/1814433.1814443.

Yuen, M. C., I. King, and K. S. Leung. 2011. "A Survey of Crowdsourcing Systems." 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 766–773. doi:10.1109/PASSAT/SocialCom.2011.203.

Zheng, W., L. Zou, and X. Lian. 2015. "How to Build Templates for Rdf Question/Answering: An Uncertain Graph Similarity Join Approach." Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 1809–1824. doi:10.1145/2723372.2747648.

Zou, L., R. Huang, H. Wang, J. X. Yu, W. He, and D. Zhao. 2014. "Natural Language Question Answering over RDF: A Graph Data Driven Approach." Proceedings of the 2014 ACM SIGMOD international conference on Management of data, 313–324. doi:10.1145/2588555.2610525.

المنارة للاستشارات
www.

Copyright of Enterprise Information Systems is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.

www.manaraa.com