Received: 8 April 2019 | Revised: 16 May 2019 | Accepted: 18 May 2019
DOI: 10.1002/cpe.5406

WILEY

SPECIAL ISSUE PAPER

Ontology negotiation: Knowledge interchange between distributed ontologies through agent negotiation

Junwu Zhu[1] | Ling Teng[1] | Huimin Lu[2] | Jieke Shi[1] | Bin Li[1]

[1]College of Information Engineering, Yangzhou University, Yangzhou, China
[2]Electrical Engineering and Electronics, Kyushu Institute of Technology, Japan

Correspondence
Junwu Zhu, College of Information Engineering, Yangzhou University, Yangzhou 225009, China.
Email: jwzhu@yzu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 61872313; Key Research Projects in Education Informatization in Jiangsu Province, Grant/Award Number: 20180012; Postgraduate Research and Practice Innovation Program of Jiangsu Province, Grant/Award Number: KYCX18_2366; Yangzhou Science and Technology, Grant/Award Number: YZ2017288 and YZ2018209; Yangzhou University Jiangdu High-end Equipment Engineering Technology Research Institute Open Project, Grant/Award Number: YDJD201707

Summary
With the proliferation of knowledge source on the internet as well as the widely professional agents, the knowledge interchange is drawing much attention. Ontology is recognized as the crucial technology due to their nature of sharing, formalization, and conceptualization to integrate and share the knowledge. In this paper, by interpreting and negotiating the communication content, a unified understanding of knowledge is formed; then, we can realize the interoperability between ontologies. We have developed the ontology automatic negotiation by agent elect protocol (AEP) to elect optimal participants and encourage agents to obey the protocol, concept mapping protocol (CMP) to find the corresponding concept mappings with the highest relevancy, and in addition, agent negotiation protocol (ANP). In ANP, we define the simultaneous negotiation protocol and agents' strategies to combine distributed ontologies interchange with agent negotiation. Finally, the implementation and preliminary results are given to verify the validity of the proposed ontology negotiation.

KEYWORDS
concept relevance, knowledge interchange, negotiation protocol, ontology negotiation

1 | INTRODUCTION

Consider a situation where there are several strangers with distinct languages want to share candies. Because strangers may not have the concepts to understand what others do, they need a shared vocabulary to help them understand each other. With the shared vocabulary, they now can negotiate with each other on how to distribute candies for maximum benefit. We develop this metaphor to introduce our motivation of knowledge interchange among distributed ontologies through agent negotiation.

Background. Knowledge interchange in coalitions is very important because it could provide the valuable information for collaborators and improve team's efficiency when they complete tasks[1] However, with the rapid development of the Internet, it is becoming clear that a fundamental problem that hampered the knowledge sharing in multiagent system is resulted from the distributed systems and the heterogeneity of the agents' knowledge. To solve these problems, description logic and ontological models are widely adopted to integrate and share the knowledge.[2-4] Ontology refers to the a formal, explicit, and detailed description of the specific domain, it is the abstraction of the essence of domain entities. Ontology emphasizes the definition of the categories and properties as well as the relations between the concepts or entities. Because of that, it has been a common method to represent the knowledge and is proved to be practicable and effective.[5,6] Therefore, we try to realize the knowledge interchange by operations of ontologies. While, agents may differ in the way they model their domain, this leads to a challenge of semantic interoperability. Various approaches have been proposed to overcome the problem.[7-9] We will discuss some of them in the succeeding section.

Motivations. To spark a deep interaction among distributed systems, we intend to realize the knowledge interchange by designing the ontology negotiation mechanism through agent negotiation and concept mappings. On the one hand, we introduce the automated negotiation protocol to reduce manual intervention, and the extension and intension of concepts are allowed to be transmitted during the negotiation process to improve the quality of communication. On the other hand, we hope the concept mapping protocol (CMP) can help the mechanism adapt to the open environment where new concepts can appear. Besides, to elect optimal participants and encourage agents to obey the protocol, we also add the agent elect protocol (AEP) into the ontology negotiation mechanism.

Our approaches and results. In this paper, we combine the ontology negotiation with the agent communication and design the protocol as well as algorithms to enable the knowledge that can be shared among distributed system. First, we present an ontology automatic negotiation model with the AEP, agent negotiation protocol (ANP), and CMP, and the conversion conditions between these protocols to achieve the goal of knowledge interchange. A method based on trust-reputation is present in AEP to elect negotiation participants, and we have proved that agents who always obey the agreement have more chance to take part in negotiation. In CMP, a cosine similarity-based approach is used to find the corresponding concept mappings. In the worst case, the proposed mapping algorithm can produce a result with the semantic relevance and have the same time complexity as a brute-force search. Then in ANP, we formalize the protocol as a tuple and require the agents involved in negotiation to execute in accordance with the protocol. To make the negotiation efficient and enable the lossless communication, we also define the lossless communication in the view of the receiver agents and different types of messages and actions for sender agents. The detection algorithm can enable the lossless transmission via theoretical proofs.

The structure of this paper is organized as follows. In Section 2, we discuss the related work on ontologies and knowledge interchange. In Section 3, we give an overview of the ontology negotiation mechanism among multiple agents including the definitions underlying the model. In Section 4, we explain the detailed protocols and algorithm to enable ontology negotiation as well as agent communication, and in Section 5, we present the implementation and preliminary results when it is applied to the news domain. We conclude the paper with the limitation and problems and describe the future work in Section 6.

2 | RELATED WORK

To meet the demand of interoperability between heterogeneous and distributed information system, researchers have developed plenty of approaches such as ontology alignment and ontology merging. Ontology alignment focuses on knowledge sharing between two objects while ontology merging can benefit several systems.

Ontology alignment. The goal of ontology alignment is to find a set of meaningful mappings between two ontologies.[10] To improve the performance of ontology alignment, Xue et al[11] proposed a novel approach based on compact genetic algorithm, which is able to reduce the time and memory consumption as well as to ensure the completeness and correctness. An ontology knowledge mining method for ontology alignment is put forward by Idoudi R[12] to address the problem complexity. Such method allows the knowledge granularity to be analyzed between ontologies and facilitates several ontology engineering techniques. The readers can look into the work of Dimitrieski et al[13] for more details and critical reviews of healthcare ontologies and existing ontology alignment approaches. However, in most cases (such as commercial organization, etc), the ontology alignment approaches rely on the third parties.[10] It is not suitable for the situations where the ontology includes some private or confidential concepts.

Ontology merging. Ontology merging is a straightforward way to address the communication problem among heterogeneous ontologies. The aim of ontology merging is to establish a common ontology that can be approved by all agents. Porello and Endriss[14,15] proposed to view the ontology merging as a problem of social choice and models the ontology using description logic. Then, they applied several social choice procedures in this problem and defined some desirable properties. Fahad et al[16] contributed to reduce the human intervention by recognizing the semantic inconsistencies automatically of ontology merging. Although the ontology merging method sounds simple and effective, it is very difficult to obtain a common ontology in dynamic multiagent system since the ontology always updates because of the multiplication of knowledge. Moreover, when all agents adopt the common ontology, they must change their original view of the word; this is not what we expected.

Similarity measures. The most crucial component in ontology mapping and merging is a concept matcher. There is a number of similarity measure for concept matcher, such as edit distance,[17,18] WordNet-based similarity algorithm,[19] Jaccard similarity coefficient,[20] similarity flooding,[21] and so on. The similarity method of edit distance is simple but it may produce wrong mappings because it ignores the semantic relation between homographic strings. The WordNet is not complete and may be incorrect in the specific domain. For Jaccard similarity coefficient, the frequency of occurrence of items in the set is not considered, when the items among ontologies are very distinct, the calculated similarity results are easily distorted. In this paper, we take the semantic relation as well as the frequency of instances into consideration and use cosine similarity[22] to find the concept correspondences between ontologies.

3 | OVERVIEW FOR ONTOLOGY NEGOTIATION BETWEEN MULTIPLE AGENTS

3.1 | Preliminary

Ontology emphasizes the definition of the categories and properties as well as the relations between the concepts or entities. It can be regarded as the two-tuples $O =< Tbox, Abox >$ where the terminology set *Tbox* is often used to define the concepts and their relations. For example, equivalence relation $C_1 \equiv C_2$ and inclusion relationship $C_1 \subseteq C_2$. The assertion set *Abox* is used to describe specific individuals in the universe. Assume Δ represents the finite set of all individuals in the domain; then, we give the extensional meaning of the concept C in the ontology O by defining $Abox(C) = \{a \in \Delta \mid O \vDash a\}$, that is, the object a belongs to the concept C.

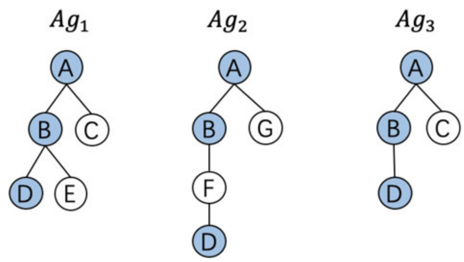


FIGURE 1 Three heterogeneous ontologies with shared concepts

Then, we will introduce local ontology and shared lexicon and their differences. **Local ontology** refers to the private knowledge base of the negotiation agent. The local ontology of agent i is abbreviated into O_i, and the set of concepts or terminologies is denoted by Φ_i. **Shared lexicon** refers to the common knowledge between the negotiation agents. The set of concepts in shared lexicon is denoted by Φ_{sl}. Another important issue in ontology negotiation is translating the concept by code and decode functions. The **Code function** is the function $CF : \Phi_i \rightarrow \Phi_{sl}$, which maps a concept in local ontology of the speaker to the concept in shared lexicon. The **Decode function** is the inverse function of CF, that is, $DF : \Phi_{sl} \rightarrow \Phi_i$, which maps a concept in shared lexicon to the concept in local ontology of the hearer. The following definitions will also be used in the model.

Definition 1 (Nearest shared super concept).
The nearest shared super concept of C is the concept $C^{'}$, which satisfies the following:

1. $O_i \vDash C \subseteq C' \wedge CF(C') \neq \emptyset$.
2. $C'', CF(C'') \neq \emptyset$ does not exist, such that $O_i \vDash C'' \subset C' \wedge C \subset C''$.

By the first definition above, C' is the super concept of C and has an equivalence concept in the shared lexicon, while the second definition states that C' is more specific than other super concepts of C.

Lossless communication refers to the knowledge that the hearer receives, and it preserves the original meaning of what the speaker intends to convey.

Definition 2 (Lossless communication).
In this paper, a negotiation communication is lossless for the hearer when the received message C_j is the most specific super concept of the concept C_i that the speaker intends to convey, that is, $O_j \vDash C_i \subseteq C_j$ and C_j' does not exist such that $O_j \vDash C_j' \subset C_j \wedge C_i \subset C_j'$.

Example 1. As shown in Figure 1, there are several agents with heterogeneous ontologies. The upper letters represent the concepts. Circles marked in blue are shared concepts, while the white ones are local concepts. The line between circles indicates the parent-child relation between concepts. The local ontology of ag_1 can be represented as a set of atomic formulas $O_1 = \{B \perp C, D \perp E, \{B, C\} \subseteq A, \{D, E\} \subseteq B\}$, and the set of concepts of O_1 is denoted by $\Phi_1 = \{A, B, C, D, E\}$. The shared lexicon in blue is $\Phi_{sl} = \{A, B, D\}$, where A is the top concept that all individuals in Δ belongs to and B, D with the relationship $D \subseteq B$. Moreover, the super concepts of E in O_1 are $\{A, B\}$ but the nearest shared super concept is B. The code function can translate the concept $C \in \Phi_1$ into the concept $A \in \Phi_{sl}$ while decode function can translate the concept $A \in \Phi_{sl}$ into his subconcept $G \in \Phi_2$ in ag_2's local ontology.

3.2 | The model

The model of ontology negotiation through agents' communication, which is depicted in Figure 2, is composed of three parts: AEP, ANP, and CMP. **AEP** is the preparatory stage of ontology negotiation: each agent can play the role of a proposer or a participant. The agent who acts as a negotiation proposer first proposes the negotiation subject, and after receiving the participant's response, the agent determines the negotiation participants according to his strategy. After that, all agents leave AEP and enter into ANP. In this part, we have the motivation of encouraging agents to obey their agreements; therefore, we put forward a trust-reputation model in which the agents with the more trust degree or reputation have the higher chance to take part in negotiations.

The most important part in this model is **ANP**. Multiple agents communicate for knowledge or a task in a specific domain to achieve interoperability of domain ontology. However, because of various reasons, for example the confidential or private property and differences in the way they model their domain, the distributed ontologies cannot exchange information directly. Therefore, we propose ANP with shared lexicon. This shared lexicon can be regarded as a vocabulary that can be understood by both parties when an agent interacts; it can help them share common knowledge and improve the negotiation efficiency. When the shared lexicon is insufficient for communication, agents will switch to CMP for help. By means of negotiation protocol and agent strategies, the knowledge can be conveyed and shared among agents.

The ontology builder may update their knowledge base in the dynamic open environment, to extend the shared lexicon and ensure the meaningful communication, **CMP** is presented. CMP uses a cosine similarity algorithm that finds the corresponding concept mappings between ontologies. When the shared lexicon is sufficiently to enable the communication, the agents leave CMP and return to ANP.

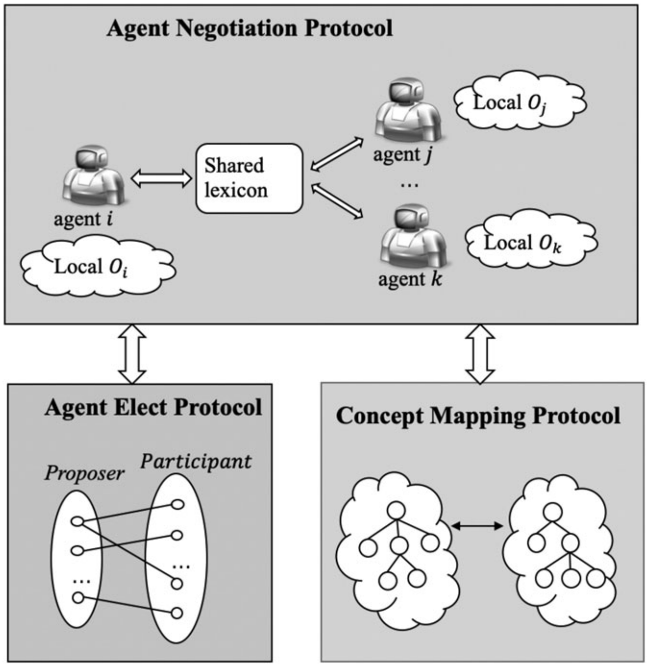


FIGURE 2 The model for ontology negotiation via agents' communication

The goal of the proposed model can be clarified as three points:

1. Design the AEP to choose the trustworthy negotiation participants and encourage them to obey and update their knowledge base in line with agreement.
2. Design the ANP that can standardize agent actions and enable the lossless communication.
3. Design the CMP to extend the shared lexicon when the agents are incapable of continuing to negotiate.

4 | ONTOLOGY NEGOTIATION PROCESS

Agent negotiation can be applied in e-commerce, resource allocation, and other fields. In this paper, agent negotiation is used to deal with knowledge conflicts of agents, so as to standardize the beliefs of agents.

4.1 | Agent elect protocol

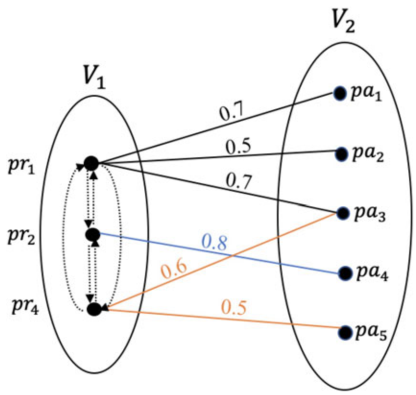
Faced with a large number of negotiation participants, proposers should choose appropriate participants according to their own needs. It is of great significance to solve the problem of negotiation agent selection before negotiation begins. Now, suppose that the set of all agents participate in negotiation is $Ag = Pr \cup Pa$, where $Pr = \{pr_1, pr_2, \dots pr_n\}$ is the set of negotiation proposers, while $Pa = \{pa_1, pa_2, \dots pa_m\}$ is the set of negotiation participants. The proposer needs to choose the participants who it trusts and can bring it higher benefits according to the trust and reputation of agents. Therefore, to improve negotiation efficiency and benefit, the AEP based on the agent's trust and reputation is established. After AEP is completed, the negotiation proposer and winner participants leave AEP and enter ANP jointly. The trust between the proposer and the participant refers to the proposer's belief in the participant's reliability and honesty. The reputation of an agent means that the proposer gets the belief about the participant from other proposers. The higher the reputation and trust are, the greater the probability of participating in the negotiation will be. Now, we will present the formal expression of reputation and trust:

Definition 3 (Trust degree between agents).

The trust degree between negotiation proposer $pr_i \in Pr$ and participant $pa_j \in Pa$ is denoted by T_{ij}.

$$T_{ij} = \begin{cases} \frac{n_j}{N_{ij}} & \text{case 1} \\ 0 & \text{case 2} \end{cases}$$

In case 1, where pr_i and pa_j had negotiation deals before, N_ij is the total number of negotiation times that they had been completed and n_j is the number of times that pa_j obey his agreement. This will encourage negotiation participant to follow the commitments.



	pr_1	pr_2	pr_3
pa_1	$T_{11} = 0.7$	$R_{21} = 0.35$	$R_{31} = 0.35$
pa_2	$T_{12} = 0.5$	$R_{22} = 0.25$	$R_{32} = 0.25$
pa_3	$T_{13} = 0.7$	$R_{23} = 0.65$	$T_{33} = 0.6$
pa_4	$R_{14} = 0.4$	$T_{24} = 0.8$	$R_{34} = 0.4$
pa_5	$R_{15} = 0.25$	$R_{25} = 0.25$	$T_{35} = 0.5$

FIGURE 3 The trust degree and reputation among negotiation agents

Definition 4 (Reputation between agents).
The reputation of negotiation participant pa_j for proposer pr_i is denoted by R_{ij}; it is resulted from others proposers' trust degree, that is,

$$R_{ij} = \sum r_{i \to -i} \cdot T_{-ij},$$

where $r_{i \to -i}$ is the reliability that pr_i trust $pr_{-i} \in Pr \backslash \{pr_i\}$.

According to the above definition, if there is negotiation history between agents, the negotiation proposer always chooses the participant that it trusts most; if there is no negotiation history between agents, the proposer always chooses the agent with the highest reputation as the negotiation object. The set of successful participants of negotiation proposer pr_i is represented as W_i; then,

$$W_i = \left\{ pa_j \mid pa_j = \text{argmax} \left(T_{ij} \vee \left(R_{ij} \bigwedge_{pa_j \in Pa} T_{ij} = 0 \right) \right) \right\}$$

which has the explanation of "choose the participant with the maximal trust degree or the reputation when it had no history negotiation with others."

Example 2. Assume a negotiation setting with three proposers $Pr = \{pr_1, pr_2, pr_3\}$ and five participants $Pa = \{pa_1, \dots pa_5\}$. We use a trust-reputation graph (see Figure 3) to represent their relationships: nodes in V_1 are the proposers, and V_2 are the participants. The solid lines (v_i, v_j) indicate there is a trust relation between $v_i \in V_1$ and $v_j \in V_2$, while the dotted lines express the reliability between the proposers in V_1. Here, we suppose that each reliability $r_{i \to -i}$ is 0.5. The table in the right of Figure 2 shows the trust degree and the reputation of each pr_i. Then, $W_1 = \{pa_1, pa_3\}, W_2 = \{pa_4\}, and W_3 = \{pa_3\}$.

4.2 | Agent negotiation protocol

The ANP is composed with the negotiation protocol as well as a negotiation strategy. The negotiation protocol is the basis of negotiation, which is used to deal with the interaction among the participants and to stipulate what kind of actions the agent should take in a certain time. The negotiation strategy includes actions and strategy selection functions, which can be used to select appropriate behavior for the negotiating agents.

Definition 5 (Negotiation protocols).
Negotiation protocol gives the basic framework of negotiation and rules to be followed in the process of agent interaction. Now, we define the negotiation protocol as a multituple $P =< N, A, S, L, u, Agr, C >$, where

$N = \{ag_1, \dots, ag_{|N|}\}$ is the set of agents who have qualified to enter the negotiation state because of the AEP.

$A = \{A_1, \dots, A_{|N|}\}$ is the set of actions where A_i is the set of actions of ag_i. In this paper, $A_i = \{propose, accept, reject, noop\}$.

S is the set of all possible states in the protocol that correspond to performances of actions by a particular agent. Especially, we use s_0 to denote the only initial state and T to be the set of all terminal states.

$L = \{L_1, \dots, L_{|N|}\}$ is a tuple where $L_i : S \backslash T \to 2^{A_i}$ is the legality function for ag_i.

$u : (S \backslash T) \times A_1 \dots \times A_{|N|} \to S$ is the update function that maps each nonterminal state to a new state. Notice that every asynchronous negotiation can be regarded as the simultaneous action when we add a dummy-move, eg, 'noop action' because it has no effect on update function.

Agr is the agreement space that result from different terminal state, and we represent the conflict deal as η, that is, the agents do not come to any agreement.

$C : T \to Agr$ is the commitment function that maps each terminal state to an agreement.

Definition 6 (Strategy of agents).
The strategy $\mathcal{S}_i$ for agent ag_i is the set of legal actions that map each nonterminal state to actions, that is, $\mathcal{S}_i : S \backslash T \rightarrow 2^{A_i}$. In other words, the strategy of agent in negotiation protocol is determined by transitions between possible states.

The strategy $\mathcal{S}_i$ is **deterministic** when, for agent ag_i, there is $\mathcal{S}_i(S \backslash T) \neq \emptyset$ and $|\mathcal{S}_i(S \backslash T)| = 1$.

During the negotiation process, there may be several "bargaining" rounds, so a negotiation deadline $s_{\max}$ is set; it can also be understood as the maximum round of negotiations. When the negotiation exceeds the maximum round $s_{\max}$, the negotiation is terminated regardless of whether an agreement is reached or not. Then, the terminal states of negotiation can be expressed as

$$T = \{s \in S \mid \mathcal{S}_i(s-1) = \{accept\} \cup \mathcal{S}_i(s-1) = \{reject\} \cup s = s_{\max}\}.$$

In the negotiation process, we assume that the action set of all agents including a proposer and a participant is $A_i = \{propose, accept, reject, noop\}$. In the initial state s_0, when ag_i intends to convey a concept or an instance, the agent will send the proposal in the form of $< sid, hid, mty, mco >$, where *sid* is speaker's ID and *hid* is hearer's ID, *mty* refers to the type of message, and *mco* is the content of negotiation, that is a set of atomic propositions. There are several types of messages.

Tell message. Tell message refers to the message that the speaker ag_i can translate from what the speaker wants to negotiate in its local concept Φ_i into a concept Φ_{sl} in the shared lexicon. This is stated in the following formula:

$$C_{\mathrm{sl}} = \begin{cases} CF(C_i) & CF(C_i) \neq \emptyset \\ CF(C_i') & \text{otherwise} \end{cases}.$$

The concept C_{sl} function as a vehicle to transmit the knowledge. When $CF(C_i) \neq \emptyset$, the tell message translated is what the speaker wants to convey into an equivalent knowledge in shared lexicon. When it does not exist, an equivalent relationship between what the speaker wants to convey and the concept in the shared lexicon, that is, $CF(C_i) = \emptyset$, then the tell message is translated to the nearest shared super concept of C_i into an equivalent knowledge in the shared lexicon. Notice that in the extreme cases, the speaker may translate the local concept into the top concept that all individuals in Δ belongs to. To prevent this meaningless communication, the speaker will switch to CMP to extend the shared lexicon.

Accept message. Once the hearer ag_j receives the message from the speaker, the hearer first makes a judgement on whether there is an information loss in the communication. When the hearer detects a lossless and consistent communication and finds that the decode function can align that message to hearer's local ontology, then the hearer updates its knowledge base with that information and returns an accept message "OK." The negotiation process ends

Request message. In most cases, the phenomenon of information loss is common in ANP. Hence, the hearer must request further detail explication of the received message by a request message when the hearer cannot determine a lossless communication. The request explication will initiate a new communication for concept negotiation.

Clarification message. The aim of clarification is to convey the extensional or the internal meaning of the message. When the speaker receives the request explication message from the hearer, the speaker will explain the concept by clarifying the relationship with other terminology in the shared lexicon. If the concepts in shared lexicon are insufficient to explain the meaning of that message, the speaker will turn to the CMP.

Reject message. The goal of the reject message is to tell others that the agent will no longer take part in this negotiation when the agent finds an inconsistency if the agent adds the received message.

Since lossless communication is an important point in this paper, Algorithm 1 shows how to detect the lossless information for the received agent ag_j.

Algorithm 1: Algorithm for detecting lossless information

Input: the shared lexicon Φ_{sl} and the ontology O_j
Output: *Flag* represents the lossless communication

$Bound \leftarrow \emptyset, Flag \leftarrow 0$
for *each* $C \in \Phi_{sl}$ **do**
  if $C \subseteq C_{sl}$ *and does not exist* $C' \in \Phi_{sl}$ *such that* $C' \subseteq C_{sl} \wedge C \subseteq C'$ **then**
    $Bound = Bound \cup \{C\}$
  end
end
for *each* $b \in Bound$ **do**
  if $d \in \Phi_j$ *and exist* $d \subseteq C_j$ *such that* $b \subseteq d$ **then**
    $Flag = 1$
  end
end
return *Flag*

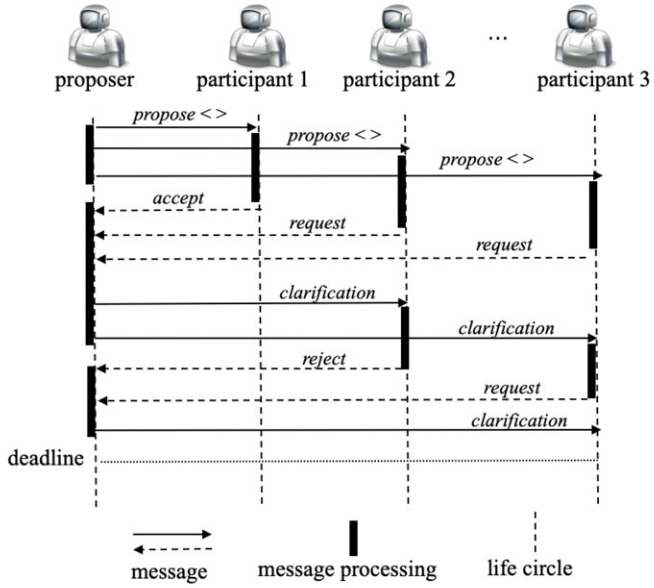


FIGURE 4 The sequence diagram for negotiation process

Theorem 1. *The ANP ensures the lossless communication.*

Proof. From the tell message, ag_j knows that $C_i \subseteq C_{\text{sl}}$ and does not exist $C'_{\text{sl}} \in \Phi_{\text{sl}}$ such that $C_i \subseteq C'_{\text{sl}} \subseteq C_{\text{sl}}$ due to Definition 3. Because ag_j always translates C_{sl} to the equivalent knowledge C_j, then we have $C_i \subseteq C_j$. According to Definition 4, we also need to show that there does not exist concept $C'_j \in \Phi_j$ such that $C_i \subseteq C'_j \subseteq C_j$. Next, we try to use reduction to absurdity and find a contradiction to prove it. Assume that there exists concept $C'_j \in \Phi_j$ such that $C_i \subseteq C'_j \subseteq C_j$. Then, in Algorithm 1, ag_j detects lossless communication when he finds that $C'_j \subseteq C'_{\text{sl}} \subseteq C_{\text{sl}} \wedge C'_{\text{sl}} \in Bound$. By combining these two formulas, we can get $C_i \subseteq C'_j \subseteq C_{\text{sl}} \equiv C_j$, which is contradictory with the fact that does not exist $C'_{\text{sl}} \in \Phi_{\text{sl}}$ such that $C_i \subseteq C'_{\text{sl}} \subseteq C_{\text{sl}}$. Therefore, the theorem has been proven. □

Now, we will introduce the negotiation process based on the negotiation protocol and agents' actions. Interaction between agents may result in the following cases shown in Figure 4. First, the proposer sends a proposal message to all agents who participate in the negotiation. Participants select action to execute in the legal actions set and then send a return message to the proposer. If participants execute an accept or reject message, or the state reaches the maximum round s_{max}, the negotiation ends.

Case 1. ag_i sends tell message to ag_j; then ag_j determines whether the atomic concept is understood by the shared lexicon. First, ag_j detects the lossless information via Algorithm 1; if the agent finds an information loss, then it switches to case 3. Otherwise, the agent will judge whether the content is consistent with the local knowledge base. If there is a conflict, the agent switches to case 2 or returns the accept message to ag_i and updates its local ontology; the negotiation process ends.

Case 2. ag_i sends tell message to ag_j, and ag_j finds lossless information but the content is in conflict with his local ontology; then, the agent can return a reject message and terminate the negotiation or make a compromise to modify beliefs and return an accept message after the amendment is completed.

Case 3. ag_i sends tell message to ag_j, when ag_j detects an information loss and cannot understand the atomic concept; then, the agent returns a request message for more details. After that, ag_i sends clarification message again to explain the relations between the shared concept, and the interactive state returns to case 1.

Example 3 (cont.).

We will continue to use the simple illustration in Section 3 to elaborate the process of the ontology negotiation among agents. The lowercase letters represent the instances in the set Δ. The actions of participants are either to propose a message of concept, to accept the proposal, to do nothing, or to reject this negotiation, that is, $A_1 = A_2 = A_3 = \{propose < message >, accept, noop, reject\}$. Under the initial state s_0, ag_1 wants to share the knowledge $E(a)$ with others, which means that the instance a belongs to the concept E. In this example, ag_1 can translate concept E into the nearest shared super concept B in the shared lexicon. Then,

$$L_1(s_0) = \{propose < ag_1, (ag_{2,3,}), tell, B(a) >\}$$

$$L_2(s_0) = L_3(s_0) = \{\text{noop}\}.$$

The update function is defined as $u(s_0, propose, noop, noop) = s_1$. The shared lexicon in state s_1 remains unchanged; the legal actions of the agents are as follows:

$$L_1(s_1) = \{\text{noop}\}$$
$$L_j(s_1) = A_j \backslash \{\text{noop}\} \quad (j = \{2, 3\}).$$

Since that ag_3 understands that $DF(B) = B$ and finds lossless communication, the agent updates its local ontology with $B(a)$ and returns the accept message (case 1). While ag_2 needs further detail explication because of the subconcept $F(D \subseteq F \subseteq B)$, the agent asks for more details about knowledge $B(a)$ by a request message. Then, $u(s_1, noop, request, accept) = s_2$, ie,

$$L_1(s_2) = A_1 \backslash \{\text{noop}\}$$
$$L_j(s_2) = \{\text{noop}\} \quad (j = \{2, 3\}).$$

When the state comes to s_2, the negotiation between ag_1 and ag_3 ends, and ag_3 can take part in other ontology negotiations. After receiving the request message, ag_1 clarifies the relation with other terminologies in shared lexicon by $E \subset B \wedge D \perp E$. So, $u(s_2, propose, noop) = s_3$. ag_2 then adds the concept E in its local ontology and returns the accept message because the knowledge is consistent (case 3). The state is updated to s_4, and all the negotiations are ended due to the accept action.

4.3 | Concept mapping protocol

When the speaker does not have any shared lexicon to explain the concept that the speaker wants to convey, the speaker can switch to CMP. To enable meaningful ontology negotiation, the agent should make correspondences between concepts in heterogeneous ontologies, which map concepts from one ontology to the mapping concept in another ontology.

First, the speaker will add shared concepts and related attributes or instances into the shared lexicon. Then, the hearer finds the synonymous mapping in the local ontology of that concept according to the similarity relations: the higher the similarity of the concepts is, the closer the semantics are. In this paper, the cosine similarity is used to calculate the similarity of two concept nodes according to the instances of the concept. For any two concepts $< A, B \mid A \in O_1, B \in O_2 >$, the calculation method of similarity is as follows. We use I to denoted the set of all instances, the I^A and I^B represent the instances of concept A and B, respectively. Then, according to the frequencies of instances that belongs to I in I^A and I^B, we can get the instance frequency vectors IV^A and IV^B of concepts A and B. Finally, the formula

$$sim(A, B) = \frac{IV^A \cdot IV^B}{\|IV^A\| * \|IV^B\|} = \frac{\sum_{i=1}^{|I|} (IV_i^A * IV_i^B)}{\sqrt{\sum_{i=1}^{|I|} (IV_i^A)^2} * \sqrt{\sum_{i=1}^{|I|} (IV_i^B)^2}}$$

can provide us with the similarity. For example, the concept A has the instance $\{a, b, c, d\}$ and the concept B has the instance $\{b, d, e, f\}$; then, the similarity between these two concepts is $sim(A, B) = \frac{IV^A \cdot IV^B}{\|IV^A\| * \|IV^B\|} = (1 + 1)/(2 * 2) = 0.5$. To reduce the unnecessary computing and communication overhead, this paper uses Algorithm 2 to find the synonym mapping instead of the brute-force search method.

Algorithm 2: Synonym concept mapping algorithm

```
Input: the concept A ∈ O_1
Output: the concept B ∈ O_2
1  h ← 1, root ← C^h
2  while sub(root) ≠ ∅ do
3      for each concept node B in sub(root) do
4          b = sim(A, root)
5          C^{h+1} = argmaxsim(A, B)
6          if sim(A, C^{h+1}) ≤ b then
7              return root
8          end
9          root ← C^{h+1}
10     end
11 end
12 return root
```

In Algorithm 2, the variable *root* is used to stores the concept node with the highest similarity of concept A among other concepts in the current depth h in ontology tree O_2 (line 1). $sub(root)$ represent all the child nodes contained in the concept node *root*. When *root* is a leaf node (line 2) or the similarity of concepts in $sub(root)$ are less than that in *root* (line 6), the algorithm ends and returns *root* with the highest similarity to concept A.

Theorem 2. *The synonym concept mapping algorithm is better than ontology merging method on time complexity.*

Proof. The first step in ontology merging is to find the corresponding relation among concepts. We assume that there are only two ontologies. Since that the concept A in O_1 is known, if we adopt ontology merging method, then $|O_2|$ calculations are required to find the most similar concept where $|O_2|$ is the number of concept nodes in O_2. However, the worst-case time complexity of our algorithm is $O(|O_2|)$, which is the same as the ontology merging method. In other words, the synonym concept mapping algorithm is just as well as the ontology merging method in the worst cases. □

5 | IMPLEMENTATION AND RESULTS

5.1 | Implementation of ontology negotiation

We demonstrate the system by using the news case from the work of Van Diggelen et al.[8] We allow negotiations to be taken place at the same time as long as the agents stay in the 'noop' state. The four agents shown in Figure 5 represent news publishers BBC, Moreover, Reuters, and Yahoo, respectively. The shared concepts are marked in blue. In the initial state s_0, the shared lexicon $\Phi_{sl} = \{Article\}$, ag_4 intends to inform all the agents about the knowledge *ScienceNews*. According to the ANP, the current shared lexicon insufficiently conveys the knowledge; they switch to CMP. The internal working of agents in CMP will not been displayed, but the results will be shown. The most similar concept of *ScienceNews* for ag_1 is *Science/Nature*, for ag_2 is *Article*, and for ag_3 is *Science*. After that, the agents exit CMP and return to ANP.

Then, the initial state s_0 is updated to state s_1 due to the update function as well as the strategies of other agents, the shared lexicon $\Phi_{sl} = \{Article, ScienceNews\}$. In state s_1, when ag_4 sends the tell message $< ag_4, ag_{1,2,3}, tell, ScienceNews >$, ag_1, ag_3 both accept the proposal and update their local ontology, and return the accept message to ag_4. However, the ag_2 detects the information loss and proposes a request message to ask more details on *ScienceNews*.

The state is update to s_2. Since that $L_1(s_2) = L_2(s_2) = L_3(s_2) = \{noop\}$, they can start new negotiations between each other, while ag_4 must propose a clarification message that states the relationship of *ScienceNews* with other terminologies, $< ScienceNews \subset Article, ScienceNews \perp SportsNews >$. Notice that *SportsNews* is not a shared concept, so the agent proposes to enter the CMP. Then, ag_2 learns that there is a semantic mapping between *SportsNews* and *Basketball*, and adds *ScienceNews* as the subconcept of *Article* and disjoint concept of *Basketball*. When CMP ends, ag_4 sends the knowledge *ScienceNews* to ag_2 again, and ag_2 knows the concept and returns an accept message. After all agents update their local ontologies, we can trigger another negotiation. We will no longer go into the painful details of the negotiation process. The final local ontology and shared lexicon are depicted in Figure 6.

5.2 | Experimental results and analysis

We use evaluation criteria in information retrieval, which include precision and recall to test the performance of ontology negotiation. The precision ratio is an index to measure the signal-to-noise ratio of a retrieval system, which is defined as

$$\text{Precision} = \frac{A}{A+B} * 100\%.$$

The recall is defined as

$$\text{Recall} = \frac{A}{A+C} * 100\%.$$

A indicates the correct matching results discerned by the method, B indicates the false matching results discerned by the method, and C represents correct results not recognized by the method. We selected three ontologies in the conference track from the Ontology Alignment Evaluation Initiative to measure the performances of our method. The three ontologies and their information are given in Table 1. The results are shown in Table 2.

We compare our method with ontology merging based on social choice (OMSC) and show their results in Figure 7. Obviously, ontology negotiation mechanism performs better than OMSC on recall but worse on precision. This is because of these two following reasons. One is that OMSC needs to consider the preferences of the majority of local ontologies and ask them to abandon their original view of the word. This

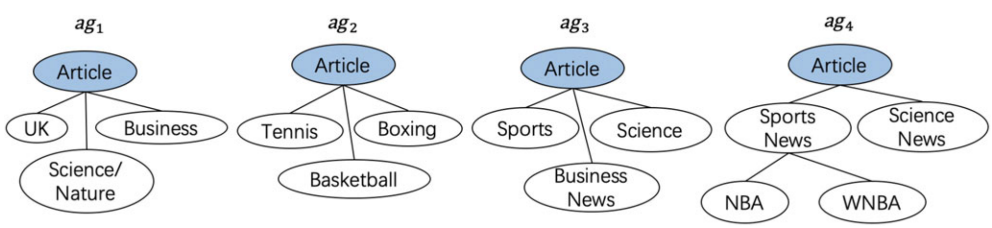


FIGURE 5 Four heterogeneous ontologies with shared concepts

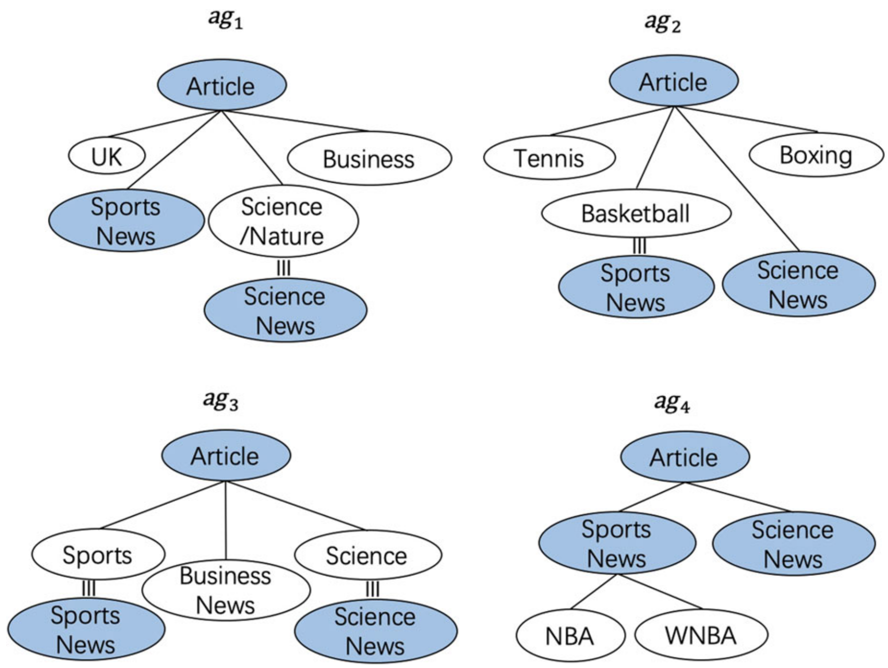


FIGURE 6 The results after ontology negotiation

TABLE 1 Ontologies and their information

Name	Number of classes	Number of datatype properties	Number of object properties
Cmt	36	10	49
Ekaw	74	0	33
Sigkdd	49	11	17

TABLE 2 Experimental results via ontology negotiation(%)

Data	Precision	Recall	F_1-measure
Cmt-Ekaw	60.00	54.55	57.14
Cmt-Sigkkd	83.33	83.33	83.33
Ekaw-Sigkkd	64.29	81.82	72.00

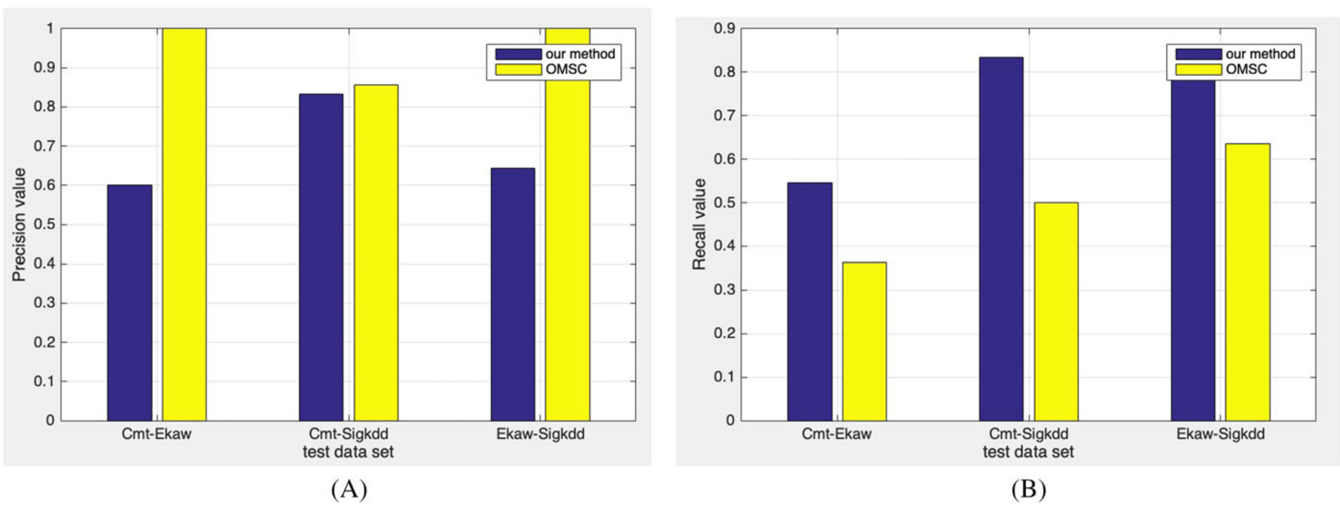


FIGURE 7 Comparison of experimental results. A, Results on precision rate; B, Results on recall rate

will result in loss of some details in local ontology, which will then lead to a low recall rate. Another one is that we introduce the CMP during the negotiation process to help local ontology learn new concepts and properties. In other words, ontology negotiation helps local ontology to extend their original knowledge in the specific domain. During dynamic environment, with the continuous expansion of terminologies, the recall rate of mechanism will continue to improve. However, at the same time, the precision rate also decreases.

6 | CONCLUSION AND PROSPECT

In this paper, we combine the agent automatic negotiation with knowledge interchange to solve the problem of semantic interoperability among distributed ontologies. First, we propose an AEP that integrates the confidence levels and reputation of agents to prevent the phenomenon of noncompliance with agreements. Then, to make the negotiation communication efficient and lossless, we define the protocol and agents' strategies in ANP as well as the cosine similarity-based approach in CMP. We consider situations where there are more than two negotiation agents by sending the message with agent's ID. So, the multiple negotiation processes can be conducted simultaneously.

However, we do not take the quality of shared lexicon into consideration. With the increase in ontologies in open environment, the shared lexicon will become larger to enable the meaningful communication. This leads us to reduce redundant concepts and produce a minimal but effective shared lexicon in future work.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under grant no. 61872313; by the Key Research Projects in Education Informatization in Jiangsu Province under grant 20180012; by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under grant KYCX18_2366; and by Yangzhou Science and Technology under grant YZ2017288 and YZ2018209; and by Yangzhou University Jiangdu High-end Equipment Engineering Technology Research Institute Open Project under grant YDJD201707.

ORCID

Junwu Zhu https://orcid.org/0000-0001-5810-613X

REFERENCES

1. Jiang G, Zhang D, Perrussel L. Knowledge sharing in coalitions. Paper presented at: 28th Australasian Joint Conference on Artificial Intelligence; 2015; Canberra, Australia.
2. Meyer T, Lee K, Booth R. Knowledge integration for description logics. Paper presented at: The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference; 2005; Pittsburgh, PA.
3. Kopena JB. Description logic-ground knowledge integration and management. Paper presented at: The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference; 2005; Pittsburgh, PA.
4. Hao G, Ma S, Sui Y, Lv J. Dynamic description logic model for data integration. *Front Comput Sci China*. 2008;2(3):306-330.
5. Liu J, Zheng BJ, Luo LM, Zhou JS, Zhang Y, Yu ZT. Ontology representation and mapping of common fuzzy knowledge. *Neurocomputing*. 2016;215:184-195.
6. Liu JNK, He YL, Lim EHY, Wang XZ. A new method for knowledge and information management domain ontology graph model. *IEEE Trans Syst Man Cybern Syst*. 2013;43(1):115-127.
7. Bailin SC, Truszkowski W. Ontology negotiation between intelligent information agents. *Knowl Eng Rev*. 2002;17(01):7-19.
8. Van Diggelen J, Beun RJ, Dignum F, Van Eijk RM, Meyer JJ. ANEMONE: an effective minimal ontology negotiation environment. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems; 2006; Hakodate, Japan.
9. Van Diggelen J, Beun RJ, Dignum F, Van Eijk RM, Meyer JJ, . Ontology negotiation: goals, requirements and implementation. *Int J Agent Oriented Softw Eng*. 2007;1(1):63-90.
10. Krysta P, Li M, Payne TR, Zhi N. Mechanism design for ontology alignment. In: Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems; 2017; São Paulo, Brazil.
11. Xue X, Liu J, Tsai PW, Zhan X, Ren A. Optimizing ontology alignment by using compact genetic algorithm. Paper presented at: 2015 11th International Conference on Computational Intelligence and Security (CIS); 2016; Shenzhen, China.
12. Idoudi R, Ettabaa KS, Solaiman B, Hamrouni K. Ontology knowledge mining for ontology alignment. *Int J Comput Intell Syst*. 2016;9(5):876-887.
13. Dimitrieski V, Petrović G, Kovačević A, Luković I, Fujita H. A survey on ontologies and ontology alignment approaches in healthcare. In: *Trends in Applied Knowledge-Based Systems and Data Science: 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August 2-4, 2016, Proceedings*. Cham, Switzerland: Springer International Publishing; 2016. *Lecture Notes in Computer Science*.
14. Porello D, Endriss U. Ontology merging as social choice. In: *Computational Logic in Multi-Agent Systems: 12th International Workshop, CLIMA XII, Barcelona, Spain, July 17-18, 2011. Proceedings*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2011.
15. Porello D, Endriss U. Ontology merging as social choice: judgment aggregation under the open world assumption. *J Log Comput*. 2014;24(6):1229-1249.
16. Fahad M, Moalla N, Bouras A. Detection and resolution of semantic inconsistency and redundancy in an automatic ontology merging system. *J Intell Inf Syst*. 2012;39(2):535-557.
17. Bille P. A survey on tree edit distance and related problems. *Theor Comput Sci*. 2005;337(1-3):217-239.
18. Ristad ES, Yianilos PN. Learning string-edit distance. *IEEE Trans Pattern Anal Mach Intell*. 1996;20(5):522-532.

19. Lu W, Cai Y, Che X, Shi K. Semantic similarity assessment using differential evolution algorithm in continuous vector space. *J Vis Lang Comput*. 2015;31:246-251.

20. Santisteban J, Tejada-Cárcamo J. Unilateral weighted Jaccard coefficient for NLP. Paper presented at: 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI); 2016; Cuernavaca, Mexico.

21. Melnik S, Garcia-Molina H, Rahm E. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In: Proceedings of the 18th International Conference on Data Engineering; 2002; San Jose, CA.

22. Zhao W, Martha VS, Chen G, Xu X. Fast information retrieval and social network mining via cosine similarity upper bound. Paper presented at: International Conference on Social Computing; 2013; Alexandria, VA.

How to cite this article: Zhu J, Teng L, Lu H, Shi J, Li B. Ontology negotiation: knowledge interchange between distributed networks through agent negotiation. *Concurrency Computat Pract Exper*. 2019;e5406. https://doi.org/10.1002/cpe.5406