Generating Consensus Ontologies among Socially Interacting Agents

Ergun Biçici

Koç University
Rumelifeneri Yolu 34450
Sariyer Istanbul, Turkey

Abstract. This paper presents an approach for building consensus ontologies from the individual ontologies of a network of socially interacting agents. Each agent has its own conceptualization of the world. The interactions between agents are modeled by sending queries and receiving responses and later assessing each other's performance based on the results. This model enables us to measure the *quality* of the societal beliefs in the resources which we represent as the *expertise* in each domain. The dynamic nature of our system allows us to model the emergence of consensus that mimics the evolution of language. We present an algorithm for generating the consensus ontologies which makes use of the authoritative agent's conceptualization in a given domain. As the expertise of agents change after a number of interactions, the consensus ontology that we build based on the agents' individual views evolves. We evaluate the consensus ontologies by using different heuristic measures of similarity based on the component ontologies.

1 Introduction

Language evolves over time and similarly terminologies evolve as well. The non-existence of a shared global conceptualization of a domain, which we can refer to when resolving misunderstandings, requires us to develop methods to find specialized and task oriented solutions. In this vein, several special purpose ontologies have been developed for different domains. However, access to most of these ontologies is not straightforward and they are proprietary [LGP+90].

Semantic Web's dream is to allow machines to share, exploit, and understand knowledge on the web [BLHL01]. An *ontology* is a thesaurus [Sco86], which answers the question of "what there is" [Qui86] in a domain. Ontologies present a structure over the language we use to represent the world.

The existence of a single ontology that can cover all the required conceptual information for reaching semantic understanding is questionable because it would presume an agreement among all ontology experts. Therefore, semantic agreement among heterogeneous ontologies is not always possible. In the most extreme case, different ontologies may not even share lexicons; hence making communication impossible.

Another problem is that there exists various ontologies that conceptualize the same domain but it is hard to decide which one provides the best conceptualization. The quality of the statements can also vary within each ontology. Thus, there is a need to

find models of building consensus among diverse sources of heterogeneous statements. In this paper, we address the problem of building consensus ontologies which represent the consensus from multiple heterogeneous ontologies belonging to a number of agents interacting with each other.

Motivation. Forming a consensus ontology is important for two reasons. First, it provides us with a vocabulary to which agents can refer to when they encounter misunderstandings in communication. Second, it provides a unified world view supported by the members, which facilitates distributed knowledge management. Any information system that makes use of different sources of knowledge needs to deal with the management of heterogeneous representations and conflicting statements. There is also the need for checking the validity of resources.

Some issues we need to address are:

– How can we resolve conflicting conceptualizations of the world?
– How can we relate concepts that are conceptualized or named differently?
– How can we evaluate the goodness of the consensus ontology?

The impossibility of a shared ontology is not only because of the difficulty of imposing a standard on ontologies but also on reaching an agreed upon conceptualization among different sources. A study by Stephens and Huhns [SH01] show the difficulties in reaching agreement even for a general domain like “humans” (an example ontology from the Stephens and Huhns data is given in Figure 1).

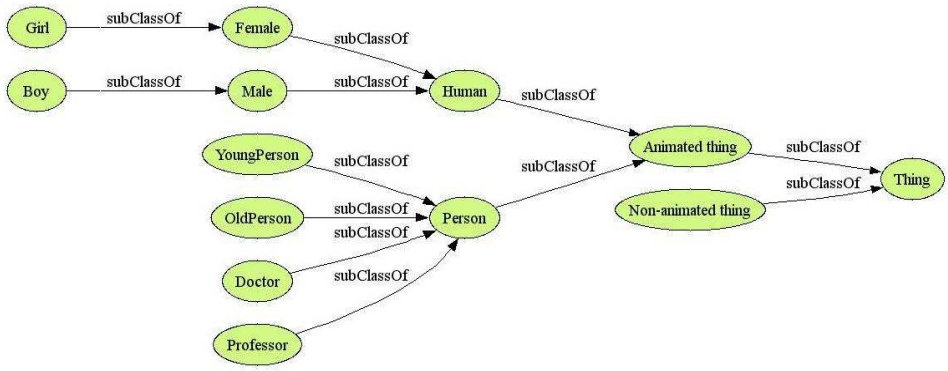

Fig. 1. Sample ontology from the data set.

Technical Challenges. Our goal is to reach semantic agreement among different world views shared by different agents. There are a number of technical difficulties that we need to address.

– *Conceptual mapping:* A concept belonging to the ontology of an agent need not be present in other ontologies due to the heterogeneity of conceptualizations. Therefore, we need to be able to find mappings between different ontologies.

- *Conflict resolution:* Finding consensus among sets of statements is not easy since they may conflict with each other. As Arrow's *social choice impossibility theorem* [Arr63] states, there can be no general method for reaching a global preference order that will obey all of the preferences specified by the members of a society.
- *Consensus formation:* What is a good way to form the consensus ontology?
- *Consensus evaluation:* Measuring the goodness of the final consensus is not easy since each agent represents a different world view.

Contributions. The interactions in a social network enable us to model the societal beliefs in the *quality* of resources as *expertise* in a given domain. Our approach for building the consensus ontology is based on combining the beliefs of experts in each domain where expertise is gained by agents through social interactions. The framework that we use is based on the social interactions of agents in a referral based multiagent system. The system collaboratively builds the consensus ontology based on the evolving values for the expertise in each domain.

The system that we have developed has the following contributions. First, we are able to model the emergence of consensual agreements among socially interacting agents. Second, we developed heuristics measures for evaluating the consensus ontology based on three different perspectives. Third, we present a method of concept mapping based on the conceptual structures in the ontologies.

Related Work. The naive approach would assign each resource (which can be computationally represented as an RDF triplet) can be assigned equal weight from each agent such that the statements with the majority of the votes win. This statistical reinforcement approach was taken by Stephens and Huhns [SH01]. This method is likely to result with conflicting and non realistic set of statements. Aberer *et. al.* [KA03] present a framework for query transformation and a method for detecting semantic agreements in which peers transform queries based on their local schema and their already existing mapping functions between schemas. Campbell and Shapiro [CS98] attempt to find algorithms for finding the meanings of unfamiliar words by asking questions. Their approach is to resolve terminological mismatches by an ontological mediator. Noy [Noy04] discusses techniques for finding correspondences between ontologies. Building consensus ontologies facilitates knowledge sharing and has applications in service composition [WPB03].

Sections. The next section introduces the formal presentation of the problem of building consensus. We discuss several abstractions for comparing ontologies such as lexical, conceptual, or information retrieval. We also discuss methods for mapping concepts. Section 3 introduces social networks of agents and how they communicate and collaborate with each other from the perspective of building consensus. In Section 4, we present our methods for building consensus ontologies and in Section 5, we present our experiments and results. Section 6 discusses possible directions of future work and in the last section we conclude.

2 Formal Definitions

In this section, we give a formal introduction and definitions to the problem of finding consensus among a given set of ontologies.

$SM(L_i, L_j) := max\left(0, \frac{min(\lvert L_i\rvert,\lvert L_j\rvert) - ed(L_i,L_j)}{min(\lvert L_i\rvert,\lvert L_j\rvert)}\right)$ (1)	$precision(\mathcal{O}, \mathcal{O}_C) = \frac{\lvert elements(\mathcal{O}) \cap elements(\mathcal{O}_C)\rvert}{elements(\mathcal{O})}$ (8)
$\overline{SM}(\mathcal{L}_1, \mathcal{L}_2) := \frac{1}{\lvert\mathcal{L}_1\rvert} \sum_{L_i \in \mathcal{L}_1} max_{L_j \in \mathcal{L}_2} SM(L_i, L_j)$ (2)	$recall(\mathcal{O}, \mathcal{O}_C) = \frac{\lvert elements(\mathcal{O}) \cap elements(\mathcal{O}_C)\rvert}{elements(\mathcal{O}_C)}$ (9)
$\overline{\overline{SM}}(\mathcal{L}_C, \mathbf{L}) = \frac{1}{\lvert\mathbf{L}\rvert} \sum_{\mathcal{L}_i \in \mathbf{L}} \frac{\overline{SM}(\mathcal{L}_C,\mathcal{L}_i) + \overline{SM}(\mathcal{L}_i,\mathcal{L}_C)}{2}$ (3)	$FMeasure(\mathcal{O}, \mathcal{O}_C) = \frac{2 \times recall(\mathcal{O}) \times precision(\mathcal{O})}{recall(\mathcal{O}) + precision(\mathcal{O})}$ (10)
$AS(C_i, <^i_{\mathcal{C}}) := \{C_j \in \mathcal{C} \mid C_i <^i_{\mathcal{C}} C_j \vee C_i = C_j\}$ (4)	$\overline{Precision}(\mathcal{O}_C) = \frac{1}{\lvert\mathbf{O}\rvert} \sum_{\mathcal{O}_i \in \mathbf{O}} precision(\mathcal{O}_i)$ (11)
$TS(L, \mathcal{O}_i, \mathcal{O}_j) := \begin{cases} TS_1(L, \mathcal{O}_i, \mathcal{O}_j), & \text{if } L \in \mathcal{L}_j \\ TS_2(L, \mathcal{O}_i, \mathcal{O}_j), & \text{if } L \notin \mathcal{L}_j \end{cases}$ (5)	$\overline{Recall}(\mathcal{O}_C) = \frac{1}{\lvert\mathbf{O}\rvert} \sum_{\mathcal{O}_i \in \mathbf{O}} recall(\mathcal{O}_i)$ (12)
$\overline{TS}(\mathcal{O}_i, \mathcal{O}_j) := \frac{1}{\lvert\mathcal{L}_i\rvert} \sum_{L \in \mathcal{L}_i} TS(L, \mathcal{O}_i, \mathcal{O}_j)$ (6)	$\overline{FMeasure}(\mathcal{O}_C) = \frac{1}{\lvert\mathbf{O}\rvert} \sum_{\mathcal{O}_i \in \mathbf{O}} FMeasure(\mathcal{O}_i)$ (13)
$\overline{\overline{TS}}(\mathcal{O}, \mathbf{O}) = \frac{1}{\lvert\mathbf{O}\rvert} \sum_{\mathcal{O}_i \in \mathbf{O}} \frac{\overline{TS}(\mathcal{O},\mathcal{O}_i) + \overline{TS}(\mathcal{O}_i,\mathcal{O})}{2}$ (7)	$O^{\cap} = < \bigcap_{i=1}^{n} \mathcal{C}_i, <_{\cap} >$ (14)

Table 1. Heuristic measures for evaluating ontologies and their elements.

2.1 Problem Formulation

We define an ontology as a 2-tuple $< \mathcal{C}, <_{\mathcal{C}} >$ where $\mathcal{C}$ represents the set of concepts and $<_{\mathcal{C}}$ is the "subClassOf" relation which relates two concepts having the subclass of relation. $C_1 <_{\mathcal{C}} C_2$ denotes that C_1 is a subconcept of C_2. A multiagent system (MAS) is a set of agents, $\mathcal{A} = \{A_1 \ldots A_n\}$, where agents interact by asking each other questions and evaluating the answers they receive. Each agent A_i has an ontology $\mathcal{O}_i =< \mathcal{C}_i, <_{\mathcal{C}_i} >$ and a lexicon, $\mathcal{L}_i$, which defines the set of allowable terms.

We use $O^{\cap}$ to denote the ontology which represents the intersection of a given set of ontologies, $\mathbf{O}$, where $O_i \in \mathbf{O}$ and $<_{\cap}$ defines an ordering consistent with all of the given set of orderings (equation 14, see Table 1).

In Table 1, we define a number of heuristics for evaluating ontologies and their elements. We base some of our heuristics on Maedche's [Mae02] representation. We compare ontologies at three different levels of abstraction: lexical (equations 1, 2, and 3), conceptual (equations 4, 5, 6, and 7), and based on the measures of information retrieval (equations 8, 9, 10, 11, 12, and 13).

We compare two ontologies at *the lexical level*, by averaging over the syntactic similarities of their lexicon (equation 2). The string matching heuristic that we use, SM, is defined based on the edit distance, *ed* (equation 1). The $|.|$ operator used in the equations corresponds to the length of the lexical term or the size of the lexicon depending on the context. The similarity of the lexicon of the consensus ontology to the lexicon of component ontologies, $\mathbf{L}$, can be computed by averaging over all component ontologies (equation 3). Since $\overline{SM}$ is asymmetric, we take the arithmetic mean.

At *the conceptual level*, we use the similarity between the conceptual taxonomies of two given ontologies. The conceptual similarity between two concepts C_i and C_j is approximated by calculating the similarity between their ancestor sets (AS) (equation 4). Based on AS, we calculate the taxonomic similarity (TS) between two conceptual hierarchies $<^i_{\mathcal{C}}$ of $\mathcal{O}_i$ and $<^j_{\mathcal{C}}$ of $\mathcal{O}_j$ for a given lexical term (equation 5). When there exists a lexical entry L that is in $\mathcal{L}_i$ but not in $\mathcal{L}_j$, then we search for the maximum

$$UC(C_i, <_{\mathcal{C}}) := \{C_j \in \mathcal{C} \mid C_i <_{\mathcal{C}} C_j) \vee C_i = C_j\} \quad (15) \qquad LCM(L_1, \mathcal{O}_1, L_2, \mathcal{O}_2) := CM(\mathcal{F}(L_1), \mathcal{O}_1, \mathcal{F}(L_2), \mathcal{O}_2) \quad (16)$$

$$CM(C_1, \mathcal{O}_1, C_2, \mathcal{O}_2) := \frac{|\mathcal{F}_1^{-1}(UC(C_1, <_{\mathcal{C}}^1)) \cap \mathcal{F}_2^{-1}(UC(C_2, <_{\mathcal{C}}^2))|}{|\mathcal{F}_1^{-1}(UC(C_1, <_{\mathcal{C}}^1)) \cup \mathcal{F}_2^{-1}(UC(C_2, <_{\mathcal{C}}^2))|} \quad (17)$$

$$OUC(C_i, <_{\mathcal{C}}) := \{C_j \in \mathcal{C} \mid C_i <_{\mathcal{C}} C_j) \vee C_i = C_j\}_{\leqslant_{<_{\mathcal{C}}}} \quad (18) \quad OM(A_{\leqslant_A}, B_{\leqslant_B}) = \sum_{i=1}^{n-1} a_i \leqslant_A a_{i+1} \Leftrightarrow \mathbf{m}(a_i) \leqslant_B \mathbf{m}(a_{i+1}) \quad (19)$$

$$OCM(C_1, \mathcal{O}_1, C_2, \mathcal{O}_2) := \begin{cases} \frac{OM(OUC(C_1, <_{\mathcal{C}}^1), OUC(C_2, <_{\mathcal{C}}^2))}{|OUC(C_1, <_{\mathcal{C}}^1)|}, & if\ |OUC(C_1, <_{\mathcal{C}}^1)| < |OUC(C_2, <_{\mathcal{C}}^2)| \\ \frac{OM(OUC(C_1, <_{\mathcal{C}}^1), OUC(C_2, <_{\mathcal{C}}^2))}{|OUC(C_2, <_{\mathcal{C}}^2)|}, & otherwise \end{cases} \quad (20)$$

Table 2. Methods for mapping concepts.

overlap among all those lexicon of $\mathcal{L}_j$ (TS_2). We define the average taxonomic similarity between two ontologies, $\overline{TS}$ (equation 6), and compute the average similarity of the taxonomy of the consensus ontology compared to the taxonomies of component ontologies by averaging over all component ontologies (equation 7).

We can view building the consensus ontology task within the scope of information retrieval, where there exists a set of target elements that we are trying to retrieve, the consensus ontology, and a larger set that we choose from, the set of component ontologies. Equations (8-13) give the definitions for our information retrieval measures where the function $elements(\mathcal{O})$ returns the set of class lexicon in ontology $\mathcal{O}$. In this sense, precision corresponds to the proportion of selected lexicon that the system got right (equation 8) whereas recall corresponds to the proportion of the lexicon that the system selected (equation 9). Equations 11, 12, and 13 calculate the averages for precision, recall, and F-Measure values correspondingly. The closer the measures we use are to 1, the better they are.

2.2 Mapping Concepts

This section presents our method of mapping concepts from different ontologies. Given two ontologies $\mathcal{O}_i$ and $\mathcal{O}_j$ with lexicons $\mathcal{L}_i$ and $\mathcal{L}_j$, let $L_i \in \mathcal{L}_i$ and $L_j \in \mathcal{L}_j$. A *mapping* function, $\mathbf{m}$, between $L_i \in \mathcal{L}_i$ and $L_j \in \mathcal{L}_j$ is a function whose domain is $\mathcal{L}_i$ of $\mathcal{O}_i$ and whose range is $\mathcal{L}_j$ of $\mathcal{O}_j$. Then, under the mapping $\mathbf{m}$, we can use L_j whenever we use L_i.

Our method for concept mapping is given in Algorithm 1. The function OCM returns the level of *ordered conceptual match* between two concepts corresponding to the lexical entries in their respective ontologies. This function is based on the taxonomic similarity that we have defined. We have set the threshold levels for the concept mapping as 0.6, 0.3, and 0.5 for α_1, α_2, and α_3 correspondingly. Our experiments verify that this selection gives us good results. $\mathbf{m}(L_1) = L_2$ states that concept topic names L_1 and L_2 match with the mapping function $\mathbf{m}$.

In Table 2, we list definitions for concept matching. We adopt the mathematical representation used in [Mae02] for formal ontology. The relation $\mathcal{F} \subseteq \mathcal{L}_{\mathcal{C}} \times \mathcal{C}$ denotes *references* for concepts. Based on $\mathcal{F}$, let for $L \in \mathcal{L}_{\mathcal{C}}$:

$$\mathcal{F}(L) = \{C \in \mathcal{C} \mid (L, C) \in \mathcal{F}\} \text{ and for } \mathcal{F}^{-1}(C) = \{L \in \mathcal{L}_{\mathcal{C}} \mid (L, C) \in \mathcal{F}\}.$$

We define abstractions for upwards cotopy (UC, equation 15), lexical concept match (LCM, equation 16), concept match (CM, equation 17), ordered upwards cotopy (OUC, equation 18), ordering match (OM, equation 19), and ordered concept match (OCM, equation 20). LCM ignores the depth of the hierarchy considered in different ontologies. Highly specialized ontologies might use various levels when representing the same hierarchical composition. To give an example, given two hierarchical structures of two ontologies,

$$\{C_1 \rightarrow B; B \rightarrow A\} \subseteq <^1_{\mathcal{C}} \text{ and } \{C_2 \rightarrow Y; Y \rightarrow B; B \rightarrow X; X \rightarrow A\} \subseteq <^2_{\mathcal{C}},$$

the concept match between C_1 and C_2 becomes: $CM(C_1, \mathcal{O}_1, C_2, \mathcal{O}_2) = \frac{3}{5}$. This discrepancy might increase when comparing two hierarchical structures belonging to two different agents with different expertise levels.

One way to overcome this is to define similarity based on the compliance of the hierarchical order in which concepts are positioned in the two hierarchies. Based on such a measure, C_1 and C_2 should have a perfect match. Thus, we define an ordered concept set as: an *ordered set* is an n-tuple, denoted by $\{a_1, a_2, \ldots, a_n\}_{\leqslant}$, such that there exists a total order, $\leqslant$, defined on the elements of the set. Based on this ordered set, we can define a new type of mapping, *monotone mapping*: a mapping $\mathbf{m} : \mathcal{L}_1 \rightarrow \mathcal{L}_2$, whose domain is the lexicon of $\mathcal{O}_1$ and range is the lexicon of $\mathcal{O}_2$, is monotone or order-preserving, if for $L_1, L_2 \in \mathcal{L}_1$, $L_1 \leqslant L_2$ implies $\mathbf{m}(L_1) \leqslant \mathbf{m}(L_2)$, where $\mathbf{m}(L_1), \mathbf{m}(L_2) \in \mathcal{L}_2$.

Ordered concept match (OCM) is based on order-preserving mappings. $\leqslant_{<_C}$ term in OUC definition (equation 18) represents the total order based on the taxonomic hierarchy of concepts. Various techniques of representing order in RDF is presented by Melnik and Decker [MD01]. The overlap between two ordered sets is given by the ordering match (OM), where $A_{\leqslant_A} = \{a_1, a_2, \ldots, a_n\}_{\leqslant_A}$, $B_{\leqslant_B} = \{b_1, b_2, \ldots, b_n\}_{\leqslant_B}$, and $\mathbf{m}$ is a mapping whose domain is $A_{\leqslant_A}$ and range is $B_{\leqslant_B}$. Simplest such mapping is the lexicographic equivalence function, which can be defined as: $\mathbf{m} = \{(x, y) \mid x \in A_{\leqslant_A},\ y \in B_{\leqslant_B},\ Lex(x) = Lex(y)\}$ where $Lex()$ is a function from set elements to lexical entities which signifies the element.

Algorithm 1: Concept mapping.

Given: *Two lexical entries L_1 and L_2 belonging to ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ correspondingly, find out if their concepts do match with the parameters α_1, α_2, and α_3 used as thresholds.*

if $SM(L_1, L_2) \geqslant \alpha_1$ **then**
  if $OCM(L_1, \mathcal{O}_1, L_2, \mathcal{O}_2) \geqslant \alpha_2$ **then**
    $\mathbf{m}(L_1) = L_2$
else if $OCM(L_1, \mathcal{O}_1, L_2, \mathcal{O}_2) \geqslant \alpha_3$ **then**
  $\mathbf{m}(L_1) = L_2$
else
  $\mathbf{m}(L_1) \neq L_2$

3 Social Networks

A referral system is a multi-agent system in which agents cooperate by using referrals where a referral corresponds to a link to another agent stored by the models of agents. A social network refers to a set of agents which socially interact with each other by using queries and answers [YS03]. Agents in our system have a number of policies to learn models of other agents that they interact with. These models store information about their expertise, the projected ability to produce correct answers, and their sociability, the projected ability to produce correct referrals.

The system differentiates between each agent's interests and expertise since these two aspects do not necessarily overlap. This enables us to model the change in each agent's expertise as they develop new interests and update their expertise correspondingly. Each agent poses a query based on its own interests. These queries are first sent to potentially expert agents in the neighborhood of an agent. Agents receiving a query may answer the query based on their confidence in their answer or refer to another agent that is more appropriate. The received answers are used for evaluating the expertise of the answering agent. We represent queries, answers, and interests as sets of $\langle term,\ expertiseValue \rangle$ tuples when we calculate the similarities between them.

Definition 1 (Similarity). *Given two sets of term-value mappings, a query Q and expertise E, the similarity of Q to E is found as follows:*

$$Q \diamond E = \frac{\sum_i q_i \times e_j}{\sqrt{n \sum_{i=1}^{n} q_i^2}},$$

where n is the number of terms in the query, $q_i \in Q$ is a term in Q, and $e_j \in E$ is a term in E such that $\mathbf{m}(q_i) = e_j$.

Definition 1 is similar to the cosine similarity measure that weighs expertise vectors with higher magnitude more. Each agent has an expertise level in a concept term from its ontology, defined in the range $[0, 1]$. Expertise levels are learned dynamically by the social network through query-answer interactions and assessments of the answers. As the interests of agents change, the contents of the questions they ask change and this, in advance, causes the expertise levels and the consensual structure to evolve over time. Thus, the system we have developed can be characterized as a dynamically evolving semantic system based on the social interactions.

Agent Communication. When two agents, A_i and A_j, communicate, they may experience misunderstandings based on the discrepancies in their intended meanings. Given a lexical term L_i from $\mathcal{O}_i$ being used by A_i to communicate with A_j, we might observe that L_i is not present in $\mathcal{O}_j$. In that case, we need to find the best matching concept from $\mathcal{O}_j$. In another case, two lexicon L_i and L_j can be syntactically equivalent but conceptually different. We accept that two agents can reach a shared understanding when the lexical terms they use to communicate share the same meaning where the meaning is based on the terms themselves and their corresponding conceptual structures. We resolve these issues by using our concept mapping algorithm (Algorithm 1).

4 Building Consensus Based on Domain Expertise

We present an algorithm based on the observation that an agent who is expert in a domain will likely be able to conceptualize the underlying structure of the domain better than others.

Algorithm 2: Building consensus based on domain expertise.

Given: *A set of agents, $\mathcal{A}$, sharing a set of ontologies, $\mathbf{O}$, find the consensus ontology, $\mathcal{O}_C$, represented by a consistent set of statements such that it represents a consensus for the MAS.*

$\mathcal{O}_C = \bigcap_{i=1}^{n} O_{A_i}$
while $newLeafSetSize \neq LeafSetSize$ **do**
    $LeafSet = getLeaves(\mathcal{O}_C)$
    $LeafSetSize = |LeafSet|$
    for $C_{subj} \in LeafSet$ **do**
        $A_{expert} = getDomainExpert(\mathbf{O}, C_{subj})$
        $expansionSet = getDomainConceptualization(\mathcal{O}_{A_{expert}}, C_{subj})$
        for $C_{obj} \in expansionSet$ **do**
            $C'_{obj} = getBestMatchingConcept(\mathbf{O}, C_{obj})$
            if $C'_{obj} \neq \emptyset$ **then**
                $add(\mathcal{O}_C, C_{subj}, C'_{obj})$
            else
                $add(\mathcal{O}_C, C_{subj}, C_{obj})$
        end
        $newLeafSet = getLeaves(\mathcal{O}_C)$
        $newLeafSetSize = |newLeafSet|$
    end
end

In Algorithm 2, we first initialize the consensus ontology to the intersection of the component ontologies. This forms the upper ontology model accepted by all agents in the MAS. For each concept in the leaf set, that is the set of concepts that are considered as leaves when the ontology is seen as a tree, we determine the expert agent in that domain. Given the set of agent ontologies from the MAS and a concept, the $getDomainExpert$ function returns the agent, A_{exp}, which is the expert in the domain corresponding to the concept. Based on A_{exp}'s conceptualization of the domain, we find an expansion set, $expansionSet$, which contains the set of concepts that are subclasses of the domain. For each concept C_{obj} in the set, we try to find a matching concept from the component ontologies which has a higher expertise level. For a given set of component ontologies and a concept, the $getBestMatchingConcept$ function returns the best matching concept, C'_{obj}, from all ontology models which has the best expertise level greater than the expertise level of C_{obj}. If the expertise level of C'_{obj} is not greater than the expertise level of C_{obj}, then this function returns the empty set.

The retrospective approach assumes that an expert agent chosen for a given concept term is likely to be good in its subconcepts. This assumption might not be true in all cases. For instance, an expert in Java programming might not necessarily be good in programming itself.

4.1 Building Consensus By Simulated Annealing

In this section, we present a method based on heuristic search in the space of RDF statement triples for finding the consensus ontology as local agreement among multiple component ontologies. We seek to find the best consensus ontology, $\mathcal{O}_C$, by adding statements to the initial consensus, which is set to $\mathcal{O}^{\cap}$. To prevent local minima, we use an approach based on randomized algorithms in which we can randomize the statement selection up to a level so that we are allowed to make bad moves.

Our general approach to consensus building is based on simulated annealing [RN95]. In the inner loop, we pick a random statement and check to see if it improves the heuristic value. If it does, we add the statement to our current consensus ontology. Otherwise, with some probability, $p = e^{\frac{\Delta E}{T}}$, we add the statement. p decreases exponentially with the badness of the move, ΔE. Also, the parameter T determines the likelihood of us allowing bad moves. $schedule$ determines the value of T based on a function of the number of cycles that has already been completed.

Algorithm 3: Building consensus by simulated annealing.

Given: *A set of ontologies,* $\mathbf{O}$*, find the consensus ontology represented by a consistent set of statements,* $\mathcal{O}_C$*, such that it has maximum heuristicValue.*

$\mathcal{O}_C = \bigcap_{i=1}^{n} \mathcal{O}_i$ (*Initialization*)
$\mathbb{S} = \bigcup_{k=1}^{n} \mathcal{O}_k \ - \ \mathcal{O}_C$
$t = 0$; *(Temperature)*
$e = 0$; *(Energy)*
for $t \leftarrow 1$ *to* ∞ **do**
    $T \leftarrow schedule[t]$
    if $T = 0$ **then**
        return $\mathcal{O}_C$
    $S_k = randneighbor(\mathcal{O}_C, \mathbb{S})$
    $\Delta E = heuristicValue(\mathcal{O}_C \cup S_k, \mathbf{O}) \ - \ heuristicValue(\mathcal{O}_C, \mathbf{O})$
    if $\Delta E > 0$ **then**
        $\mathcal{O}_C = \mathcal{O}_C \cup S_k$
    else
        $\mathcal{O}_C = \mathcal{O}_C \cup S_k$ with probability $e^{\frac{\Delta E}{T}}$
end

In Algorithm 3, $heuristicValue$ function is any heuristic measure that estimates the level of overlap based on the given component ontologies. We choose to use the taxonomic overlap measure, $\overline{\overline{TO}}$, which corresponds to the taxonomic overlap among

its arguments. This is due to our data set which contains mostly taxonomic relations and due to the fact that taxonomic relations are more important than non-taxonomic ones. $randneighbor(\mathcal{O}_C, \mathbb{S})$ is a function which returns a randomly chosen neighboring statement, $S_k \in \mathbb{S}$, of the current consensus ontology such that $\mathcal{O}_C \cup S_k$ is consistent. By neighboring statements to an ontology, we mean the set of statements that can be added to extend a given ontology such that the consistency is preserved.

5 Experiments and Results

We have experimented with a number of agents ranging from 5 to 1000 having various numbers of differing ontologies ranging from 2 to 53. The expertise levels of agents are initialized to a measure of the depth of the domain within each agent's ontology. The results of our experiments are given in Table 3. We evaluate a consensus ontology based on how well it fits with the component ontologies of the given agents. The evolving nature of the consensus ontology that is generated among 500 agents using 53 different ontologies can be seen in Figure 2, which are ordered according to F-Measure.

In our experiments, we attempted to address the variance in the performance of the consensus ontology with respect to the number of agents involved and the number of differing ontologies used. We present our results in Table 3 where $AvgSynSim$ and $AvgTaxSym$ corresponds to average syntactic and taxonomic similarity scores correspondingly. The resulting graph when the results are plotted in 3D is given in Figure 3. The results show that the performance increases some as we decrease the number of agents collaborating towards the consensus and it increases greatly as we decrease the number of different ontologies taking role.

We have also experimented with the threshold values used in the similarity measures to find the best setting for building consensus. Under the setting with 50 agents sharing 5 different ontologies, we have found that α values of 0.6, 0.3, and 0.5 for α_1, α_2, and α_3 correspondingly gave the best results for the syntactic and taxonomic match measures. F-Measure is maximized when α_1, α_2, and α_3 is set to 0.5, 0.2, and 0.3. We chose to use 0.6, 0.3, and 0.5 for the presented experiments which gave good results overall. All concept mapping algorithms need to balance the weights given for the lexicon, which may be regarded as the pointers to the real concepts, and the weights given for the conceptual structures themselves.

6 Future Work

In the current version of the system, only the consensus ontology is allowed to evolve whereas individual agents' ontologies remain unchanged. Allowing each agent to change its ontology based on queries might be a better alternative for simulations.

The final consensus ontology that is built can be refined based on some heuristics. One such heuristic is the *coherence continuum*. If the expert agent chosen for domains that are consecutively ordered based on the subclass relation, such as in $C_1 \subseteq C_2 \subseteq C_3$, alternate, then we can choose to refine the consensus ontology so that all the domains are chosen from the alternating agent's recommendation. For example, if A_1 is the expert

for domains C_1 and C_3 and A_2 is the expert for C_2, then to preserve the coherence we can discard the intermediate agent, A_2's conceptualization.

Another refinement can be done in choosing good domain experts. We can choose to store domain expert histories which can later be used as to select experts from when the expertise of the best agent in the current domain is not as good as the agents who are experts in the upper levels of the consensus ontology. This retrospective approach assumes that an expert agent chosen for a given concept term is likely to be good in its subconcepts. In the real world, this assumption might not be true in all cases. For instance, an expert in Java programming might not necessarily be good in programming itself.

7 Conclusion

We studied the generation of consensus ontologies among agents having differing ontologies within the multiagent system framework. The system that we have developed has the capability of modeling the emergence of consensual agreements among socially interacting agents. We developed heuristics measures for evaluating the consensus ontology based on three different perspectives and methods for conceptual processing. We presented a method of concept mapping based on the conceptual structures in the ontologies.

We expect that this research will help us understand and formalize the tradeoffs between approaches to building consensus which can later determine inference mechanisms that can be in place.

Acknowledgements

The research reported here was supported in part by North Carolina State University. The author would like to thank Munindar P. Singh, James Lester, and Jon Doyle for helpful discussions and for their guidance and support during the term of this research.

References

[Arr63] Kenneth J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, New York, second edition, 1963.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic Web. *Scientific American*, 284(5):34–43, May 2001.

[CS98] Alistair Campell and Stuart C. Shapiro. Algorithms for ontological mediation. In Sanda Harabagiu, editor, *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 102–107. Association for Computational Linguistics, Somerset, New Jersey, 1998.

[KA03] M. Hauswirth K. Aberer, P. Cudr-Mauroux. Start making sense: The chatty web approach for global semantic agreements. *Journal of Web Semantics*, 1(1), 2003.

[LGP+90] D. B. Lenat, R. V. Guha, K. Pittman, D. Pratt, and M. Shepherd. Cyc: Toward programs with common sense. *Communications of the ACM, CACM*, 33(8):30–49, August 1990.

[Mae02] Alexander Maedche. *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers, Boston, 2002.

[MD01] Sergey Melnik and Stefan Decker. Representing order in RDF, 2001.

[Noy04] Natalya Fridman Noy. Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, 33(4):65–70, 2004.

[Qui86] W. V. O. Quine. *Philosophy of Logic*. Harvard University Press, second edition, 1986.

[RN95] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1995.

[Sco86] Dana S. Scott. Capturing concepts with data structures. In *Proc.DS-2, IFIP TC-2 Conference on Knowledge and Data, Portugal*, November 1986.

[SH01] Larry M. Stephens and Michael N. Huhns. Consensus ontologies: Reconciling the semantics of web pages and agents. *IEEE Internet Computing*, 5(5):92–95, 2001.

[WPB03] Andrew B. Williams, Anand Padmanabhan, and M. Brian Blake. Local consensus ontologies for b2b-oriented service composition. In *AAMAS*, pages 647–654, 2003.

[YS03] Pınar Yolum and Munindar P. Singh. Dynamic communities in referral networks. *Web Intelligence and Agent Systems*, 1(2):105–116, 2003.

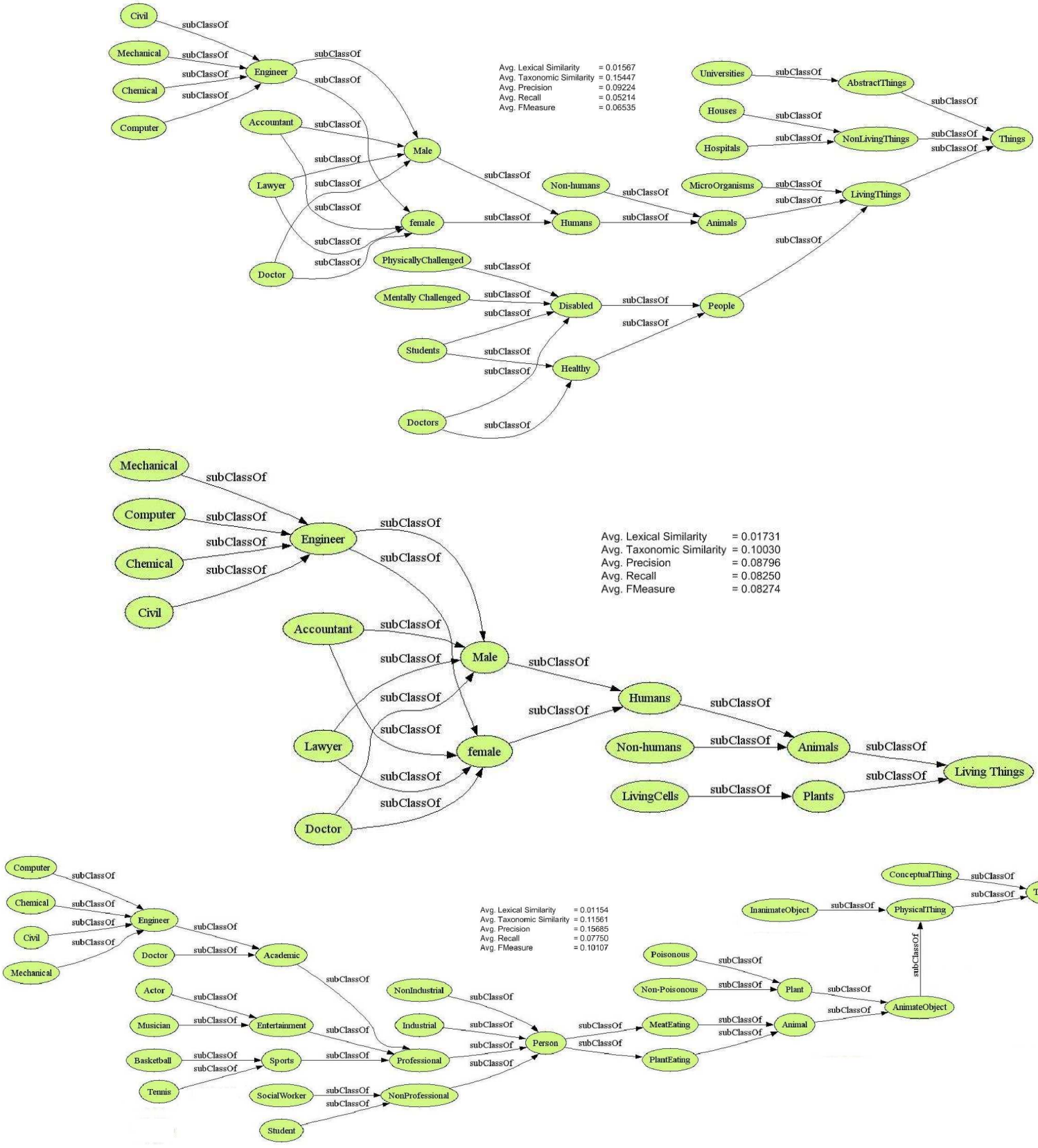

Fig. 2. Evolution of consensus among 500 agents using 53 different ontologies ordered according to their FMeasure performance.

		5	10	25	50	100	250	500	1000
2	$AvgSynSim$		0.3856		0.3856	0.3856	0.3856	0.3856	0.3856
	$AvgTaxSim$		0.2890		0.2890	0.2890	0.2890	0.2890	0.2890
	$FMeasure$		0.5417		0.5417	0.5417	0.5417	0.5417	0.5417
5	$AvgSynSim$	0.1258			0.1249	0.1231	0.1267	0.1267	0.1240
	$AvgTaxSim$	0.2011			0.1997	0.1970	0.2025	0.2025	0.1984
	$FMeasure$	0.2433			0.2472	0.2550	0.2393	0.2393	0.2511
10	$AvgSynSim$		0.0710		0.0783	0.0783	0.0759	0.0979	0.0963
	$AvgTaxSim$		0.1666		0.1678	0.1678	0.1674	0.1962	0.1777
	$FMeasure$		0.2234		0.1893	0.1893	0.2006	0.1993	0.2384
25	$AvgSynSim$			0.0266	0.0264	0.0265	0.0266	0.0261	0.0262
	$AvgTaxSim$			0.1278	0.1289	0.1283	0.1278	0.1305	0.1300
	$FMeasure$			0.1239	0.103	0.1135	0.1239	0.0716	0.0821
53	$AvgSynSim$				0.0162	0.0141	0.0131	0.0144	0.0141
	$AvgTaxSim$				0.1181	0.1188	0.1164	0.1281	0.1188
	$FMeasure$				0.0794	0.0884	0.0938	0.0831	0.0884

Table 3. Evaluation results for the consensus built.

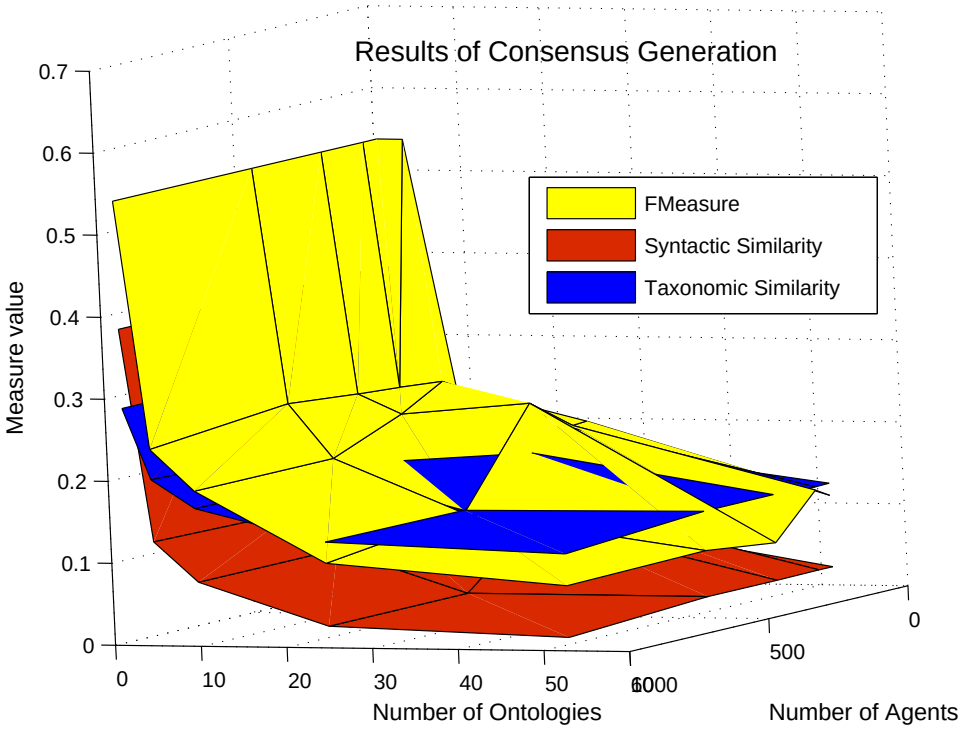

Fig. 3. Results plotted in 3D.