Machine Learning Module to Improve Communication between Agents in Multi-agent System

Conference Paper · December 2012

DOI: 10.1109/ICMLA.2012.235

3 authors:

Shimaa Elsherif
Benha University

Behrouz H. Far
University of Calgary

Armin Eberlein
University of Regina

Machine learning module to improve communication between agents in Multi-Agent System

Shimaa M. El-Sherif[1] , Behrouz Far[2]
Department of Electrical and Computer Engineering
University of Calgary
Calgary, Canada
e-mail: [1]smmelshe@ucalgary.ca
[2]far@ucalgary.ca

Armin Eberlein
Department of Computer Science & Engineering
American University of Sharjah
Sharjah, UAE
e-mail: eberlein@ucalgary.ca

Abstract— **Distributed knowledge has attracted more and more attention as a way to improve knowledge sharing across the world using the Internet. This paradigm enables many systems to interact with each other and share their knowledge while keeping their own ontology. Several researchers have worked on this topic with different strategies but they all argue that the main issue is to make sure that the other systems understand the concepts of its domain correctly. In order to be sure that they understand each other, systems use concept learning to learn the meaning of concepts they communicate with. In this paper, we try to overcome this complexity by suggesting a system that enables agents to learn new concepts from several different agents at the same time and each agent has its own ontology. We use social networks paradigm to communicate between agents to enhance the accuracy of learning process.**

Keywords- machine learning; distributed knowledge management; concept learning; multi-agent system; social network; ontology

I. INTRODUCTION

The World Wide Web (WWW) has enabled users to access information from all over the world. With the increase of the amount of information available, the burden on the user to search, filter and select the desired information increases drastically. One possible solution is to enhance the WWW infrastructure by adding semantics to the search. This is the motivation for the semantic web [1] which inherits the decentralized architecture of the traditional Internet. On one hand, decentralization of the web has advantages as it scales up easily and there is no single point of failure. On the other hand, this decentralization causes several problems, such as understandability and heterogeneity of ontologies. This can be overcome using Multi-Agent Systems (MAS).

In real life, it is better to get the opinion of several people rather than listening to only one person. Different people have different point of views for a given subject, so the problem that faces the learner is to compromise all these opinions and came out with its own point of view. In MAS, each agent creates its own ontology that suitable for its own problem solving strategy and may vary from others' ontologies even if they describe the same domain. For an agent (learner agent), in order to learn a new concept from several other agents (teacher agents), each teacher agent sends some positive and negative examples that describe the required concept to the learner agent. These examples agree with its ontology and may differ from examples sent by other teachers. The problem that faces the learner agent is to try to compromise all those results to reach a suitable definition to the new concept. The higher the learning accuracy the better the definition is.

In this paper we introduce a semantic search technique based on ontological concept learning. Our system based on Multi-Agent System (MAS) that can handle semantic search and learn new concepts and at the same time hide the search and learning complexity from the user. The aim of this paper is to show that teacher agents in our system can teach a learner agent new concepts even if this is not a standalone concept in their ontologies. Also, we aim at proving that using social networks in communicating between agents in concept learning system improves the accuracy of the learning process.

II. BACKGROUND

In this section we define some fundamental terms that are essential to understand our work.

A. Agent:

Agents can be defined as: "*Anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. For example a software agent receives keystrokes, file contents and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files and sending network packets*" [2].

In our framework, we use the architecture of the agent that depends on three parameters, $Ag = (Sit, Act, Dat)$. *Sit* is the set of situations the agent can be in, which represents the effectors on the agent. *Act* is the set of actions the agent can perform. *Dat* is the set of internal data areas of the agent. There are functions for the agent f_{Ag}: $Sit \times Dat \rightarrow Act$, which enable the agent to select the most suitable action to perform according to the current situation and the internal knowledge of the agent [3].

B. Multi-Agent System (MAS)

MAS is defined as: "*a loosely coupled network of problem solvers (agents) that interact to solve problems which are beyond the individual capabilities or knowledge of each problem solver*" [3]. Therefore, MAS is a collection of heterogeneous agents, each of which with its own problem solving strategy, that are able to interact and coordinate with each other [4]. When an agent is implemented in an MAS, it interoperates with other agents, therefore the architecture of agent must be able to distinguish between its own internal data Dat_{own} and the internal data of other agents Dat_{other} it communicates with [5]. At the same time it must be able to understand other agents even if they use different knowledge representations.

C. Concept

A concept is a unit of thought consisting of two parts, extension and intension. Extension covers all objects belonging to this concept and intension comprises all features valid for all those objects. With each concept there are some documents which their major topic is this concept. These documents are considered positive examples for this concept.

D. Ontology

Ontology is widely used in AI, knowledge engineering, education, e-commerce, semantic web, etc. There are several definitions of ontology. We will use Daconta's definition [6] which is relevant to our work, "*Ontology defines the common words and concepts (meanings) and their relationships used to describe and represent an area of knowledge, and so standardize the meanings*."

E. Social networks

By social networks we do not mean Facebook, Twitter or other related web services. In this work, a social network is a set of actors (human, agent, document repository, etc.) and relationships between them. It can be represented as a set of nodes that have one or more kinds of relationships (ties) between them [7]. Using social networks gives us flexibility in dealing with concepts in ontologies. It allows agents to understand the meaning of the same concept even though its definition might be slightly different in each agent's ontology.

Earlier social networks only consider the existence of a tie (i.e. relationship) between two actors. Recently, more attention has been paid to the tie strength. The strength of a tie is affected by several factors. Granovetter [8 - 10] proposed four dimensions that may affect tie strength: the duration of the relationship; the intimacy between the two actors participating in the relationship; the intensity of their communication with each other; and the reciprocal services they provide to each other. Other factors, such as socioeconomic status, educational level, political affiliation, race and gender are also considered to affect the strength of ties [11]. Structural factors, such as network topology and information about social circles, may affect the tie strength [12]. Gilbert et al [13] suggest quantitative measures (variables) for the tie strength including intensity variable, days passed since the last communication and duration [13]. Another variable that may affect the strength of the tie is the neighborhood overlap variable [14] which refers to the number of common friends the two actors have. Petróczi et al [15] introduced mutual confidence between the actors of social networks. We propose in [16] a new methodology to calculate the strength of ties between agents in a social network using Hidden Markov Models (HMM) [17]. In this paper we illustrate that tie strength depends on several factors: Closeness factor: by measuring how close two agents are to each other (i.e. the degree of similarity between the two ontologies used by the two agents participating in the relationship); Time-related factor: it combines all time factors that affect the strength of the relationship (e.g. duration of the relationship, frequency of communication between the two agents, time passed since the last communication ... etc); Mutual confidence factor: clarifying the nature of the relationship under measure, if it is a one-sided relationship or a mutual relationship. Then we built an HMM model to measure the strengths of ties between agents in a social network using those factors.

III. SYSTEM ARCHITECTURE

Most researchers consider both common language and same fully understanding of used concepts. In real world, each MAS is designed and implemented by a different group of developers and each group uses different ontology in representing knowledge bases. MAS can handle distribution and decentralization of information at the expense of ontology diversity. In order to overcome the difficulty of communication between agents with diverse ontologies, we suggest integrating semantic search with concept learning to enable agents to learn concepts from each other and therefore understand each other better. Ontological concept learning helps an agent to understand new concepts from peer agents. We use social networks to manage communication between agents and to improve the learning process by resolving the conflicts that may occur during learning new concepts from several agents. In order to achieve our goal, several works had been done. In [22], methods of concept learning verification are explained in details. In [23], we started to utilize this learning mechanism into the semantic search application, and enrich its practicability with some modifications. In [18], [20] and [21], a conceptual model for semantic interoperation between concept learning and semantic search has been proposed.

Our system consists of a group of MASs, each controls a repository of knowledge. The knowledge in each repository is represented by different ontology (we are not restricted to a single ontology in representing our knowledge bases). The MASs can communicate with each other and try to understand each other. MASs are communicating via a social network. Strengths of ties between agents in the social network represent how close/far each two agents. The tie strengths are updated dynamically after each communication between agents.

Each MAS has two major modules; semantic search module and concept learning module [18][19]. In semantic search module [20], one of the agents, in this case it is called local agent, accepts a search query from a user and tries to search it locally in its local repository. In order to improve the search results, the local agent sends the same search query to all other agents; they are called remote agents in this case. Each remote agent searches its local repository and sends the results back to the local agent. The local agent collects all results and sends it back to the user.

The second major module in our system is concept learning module [21]. During semantic search process, the local agent may discover that it does not know a certain concept and needs to learn it. In this case, the local agent became a learner agent and sends a learning request to all remote agents (teacher agents in this case) to learn this new concept.

IV. SEMANTIC SEARCH PROCESS

We have devised a spiral workflow to incorporate both search and concept learning in the semantic search process, see figure 1. This process was introduced in [21].

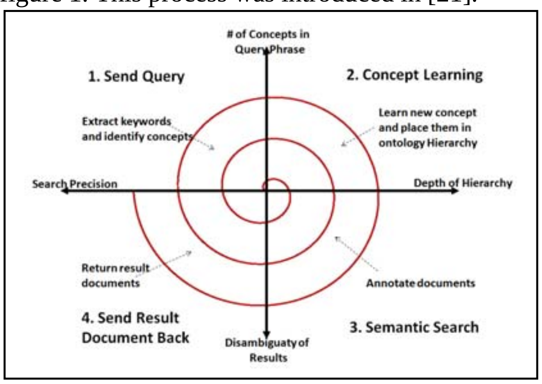

Figure 1. Spiral workflow between semantic search and concept learning

The process started when a user sends a search query to our search engine (Send Query phase). The system tries to extract keywords sent in the search query and identify the concepts in it. Each concept is compared to those defined in the ontology of the local repository. In this stage, local agent may identify new concepts that do not exist in its local repository. After finding new concepts, the local agent sends a learning request to all remote agents to learn this new concept. This step represents the initialization of the concept learning process (Concept Learning phase). The concept learning mechanism is continued until the local agent learns the new concepts adequately. After learning all new concepts, the concept hierarchy in local repository is reorganized to add the new concepts in their proper position. After learning all new concepts, the annotation procedure is then performed on the fly on all the concepts in the local repository, old and newly learned concepts (Semantic Search phase). UIMA [24] is used to enable search and classification within repository. At the same time, the local agent broadcasts the search query to all remote agents to search their local repositories and send back the search result. Finally, the local agent collects the returned results from remote agents and ranks them, depending on social networks, before sending them to the user (Send Result Documents Back phase).

V. CONCEPT LEARNING MODULE

The learner agent (Ag_L) in the concept learning module initiates the learning process by sending a learning request to all peer agents (teacher agents in this case), as described in section IV. This request contains a query with all available information about the required concept C_{goal} to be learned (e.g. concept name, keywords list, annotation information, feature/contextual information).

After receiving the initial query, each teacher agent (Ag_T) finds a concept C_{best} that best matches the information in the initial query it receives. After finding C_{best}, Ag_T chooses positive and negative examples representing C_{best} based on the similarity between C_{best} features and information sent in the initial query (see section VI below). The number of positive and negative examples chosen from each teacher agent depends on how socially close this teacher agent is to the learner agent (i.e. the strength of tie between Ag_L and each Ag_T), because tie strength reflects how much Ag_L trusts and depends on Ag_T. After selecting positive and negative example sets, each Ag_T sends its own set to Ag_L.

Finally, Ag_L measures the closeness between its updated ontology and ontologies of all teacher agents. Based on this closeness and interactions that occurred during the whole learning process, the strength of ties between Ag_L and all teacher agents is updated [16].

VI. SELECTING POSITIVE AND NEGATIVE EXAMPLES

After receiving the initial query, the teacher agents (Ag_T) tries to find a concept C_{best} that best matches the information in the initial query in order to answer the learning request sent. After finding C_{best}, Ag_T chooses some positive and negative examples representing C_{best} based on the similarity between C_{best} features and information sent in the initial query. In choosing positive and negative examples, we follow the strategy proposed in [19]. If the similarity between C_{best} features and sent information is higher than a selected threshold, then all examples representing C_{best} in Ag_T's ontology can be used as a source of positive examples and examples of siblings can be used as a source of negative examples, as shown in figure 2. If the similarity value is less than the selected threshold, then positive example source is only these examples represent the information sent and negative examples source is the rest, as shown in figure 3. Number of positive and negative examples chosen from each teacher agent depends on how close this teacher agent to the learner agent Ag_L (i.e. the strength of tie between Ag_L and Ag_T), because tie strength reflects how much does Ag_L trust Ag_T and depend on it. After selecting positive and negative example set, each Ag_T sends its own set of examples to Ag_L.

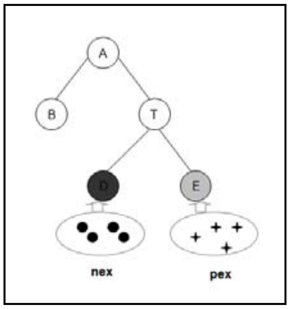

Figure 2. Selection of negative examples from external concept [19]

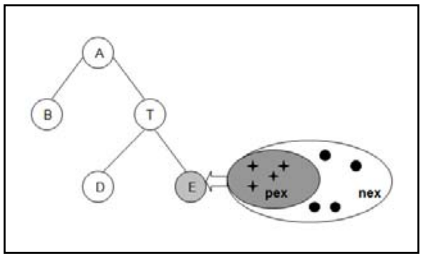

Figure 3. Selection of negative examples internally from the same concept examples [19]

VII. KNOWLEDGE MANAGEMENT

In order to manage ontology, each agent needs to perform document classification. Document classification is the task of assigning each document to one or more categories using a data mining technique. Document classification is used to organize explicit knowledge. In order to do knowledge classification, there are two major steps to follow: document preprocessing; document categorization

A. Document preprocessing:

In order to classify a document, we first need to generate a feature vector for this document. Document preprocessing consists of two tasks: feature vector extraction and document representation.

1) Feature vector extraction

The purpose of feature vector is to get a list of words that best describe the document sufficiently. To get feature vector, documents are tokenized to get a list of all words in the documents. Afterwards, a feature reducing technique is applied to reduce the list of words to the most significant words that describe the documents. In our system, we use Stop Words removal to reduce words of each document. Stop Words is used to remove words that usually do not contribute to the semantics of documents, such as numbers, prepositions or common English words such as: a, an, the … etc. we also create our own list of words that are not affecting document categorization. The next step is to use a common Natural Language Processing (NLP) method to reduce feature vector length. In our system, we use Word Stemming; Snowball technique. This technique deals with words, that have the same stem, as one word (e.g. mathematics and mathematical). Finally, a statistical technique is used to determine the most significant features for each document. In our system, we use Term Frequency plus Inverse Document Frequency (TFxIDF) technique. This technique calculates the relevance of each feature to each category.

2) Document representation

A document can be represented as a vector of a sequence of features and their weights. Feature weight can be of any type (starting from binary values to complex weighting schemes such as TF or IDF values). We select the top n words created for each category with the highest scores accompanied with their weight as a feature vector for each category. In our experiment, we set n to 20 as we noticed that in almost all documents, all relevant document features can be expressed with less than or equal to 20 words.

B. Document categorization

This is the learning process. In this stage, we assign each group of documents to a certain category and calculate the accuracy of learning technique used. In this paper, we use three different learning techniques: K-Nearest Neighbor (K-NN); Naive Bayes; and Support Vector Machine (SVM).

In our experiment, we adopt the RapidMiner (http://rapid-i.com/content/view/181/190) tool to perform document classification. It is a free data mining software tool that offers variety of methods to do documents preprocessing and classifications.

VIII. CASE STUDY

In our case study, we have data set consists of structured hierarchy ontologies of course syllabi in four universities; Cornell University Ag_C, University of Michigan Ag_M, University of Washington Ag_W and University of Calgary Ag_G. In order to test our approach, we use three MASs as teacher agents; Ag_C, Ag_M and Ag_W. We consider Ag_G as learner agent to learn some new concepts from the three teacher agents at the same time. Each MAS controls a repository that contains one of the four ontologies for the four universities course syllabi. In those ontologies, each concept represents a unit program in the taxonomy of each university. The learner agent tries to learn a new concept from the three teacher agents at the same time. The concept to be learnt is “Programming Language”. This concept is not a standalone concept in ontologies of all teacher agents but the teacher agents have to search for the best match for it to teach it to the learner agent.

In order to test our system, we have two experiments.

Experiment 1: In this experiment, there are no social networks introduced in the systems. All teacher agents have the same effect on the learner agent. Ag_G tries to learn a new concept “Programming language” from all teacher agents at the same time. Ag_G sends a learning request to all teacher agents containing the name of the new concept to be learnt “Programming Language” and the annotation that defines this concept. The annotation used is: “programming language” or “C++” or “Java” which means if the terms program and language or C++ or java are found in the document, then this document is one of our targets. Finally, we measure the accuracy of the learning processes by the three learning techniques used.

Experiment 2: in this experiment, we introduce a social network in defining relationships between the learner agent Ag_G and all teacher agents; Ag_C, Ag_M and Ag_W; in our system. In order to measure the strength of ties between learner agent Ag_G and each teacher agent Ag_C, Ag_M and Ag_W, we measure the closeness between ontology used by Ag_G and ontologies used by each teacher agent. The values of the closeness are considered as initial values of tie strengths between learner agent and each teacher agent. In this experiment, the learner agent tries also to learn the new concept "Programming Language". Ag_G sends a learning request to all teacher agents containing the name of the new concept to be learnt, "Programming Language", and the annotation that defines this concept, "programming language" or "C++" or "Java". Afterwards, we measure the accuracy of the learning processes using the three learning techniques used. Finally, we compare results obtained in both experiments to show the effect of using social networks in the overall accuracy of learning process.

In order to measure the accuracy of correctly identified learned concept, we use the confusion matrix, shown in table I, to measure the proportion of true results (i.e. true positive and true negative).

TABLE I. CONFUSION MATRIX

True positive	**False positive**
False negative	**True negative**

$$\text{Overall accuracy} = \frac{\text{True positive} + \text{True negative}}{\text{True positive} + \text{False positive} + \text{False negative} + \text{True negative}}$$

IX. RESULTS

A. Experiment 1

In this experiment, no social networks are introduced in defining relationships between agents. The learner agent, Ag_G, needs to learn a new concept "Programming Language". It sends the annotation ("programming language" or "C++" or "Java") to all teacher agents; Ag_C, Ag_M and Ag_W. Teacher agents use this annotation query in searching their ontologies for the best matched concept C_{best}.

TABLE II. NUMBER OF RETURNED DOCUMENTS AND THE SIMILARITY VALUES OF SEARCH RESULTS.

University/department	K	Similarity of K
Cornell University		
Computer Science	25	0.29
University of Michigan		
Electrical engineering and computer science	11	0.06
School of music/ ensemble	2	0.06
University of Washington		
Computer science and engineering	13	0.14

K: is the number of returned documents

From table II, we can see that the best matched concepts are:

- "Computer Science" from Cornell University
- "Electrical Engineering and Computer Science" from University of Michigan
- "Computer Science and Engineering" form University of Washington

As the similarity values of all chosen concepts are low then all negative examples are chosen internally from same concept. In this experiment, no social network is defined, so all teacher agents have equal effect on the learner agents, so the number of positive and negative examples from each teacher agent should be the same. We use 11 positive examples and 11 negative examples from each ontology to send them to Ag_G to learn this new concept. We use three machine learning techniques to get the accuracy of the learning process.

1. Using K-NN for learning:

	True non PL	True PL
Pred. Non PL	18	3
Pred. PL	15	30

Overall accuracy = 72.75%

2. Using Naive Bayes for learning:

	True non PL	True PL
Pred. Non PL	20	12
Pred. PL	13	21

Overall accuracy = 62.2%

3. Using SVM for learning:

	True non PL	True PL
Pred. Non PL	21	12
Pred. PL	12	21

Overall accuracy = 63.74%

We can notice here that the learning accuracy is not high enough, but that is due to the low number of positive and negative examples used in the learning process. In next experiment, we aim at improve these accuracy values via using a social network in communicating between agents.

B. Experiment 2

In this experiment, we set a social network between MAS controls repositories (learner and teacher agents). We measure closeness between ontologies of learner agent Ag_G and teacher agents Ag_C, Ag_M and Ag_W and use these closeness values as initial values of tie strengths between them. The initial values of tie strengths between the learner agent and each teacher agent are shown in table III.

TABLE III. INITIAL TIE STRENGTHS VALUES BETWEEN LEARNER AGENT AND EACH TEACHER AGENT

Teacher agent	Tie strength value
Cornell University Ag_C	0.255
University of Michigan Ag_M	0.52
University of Washington University Ag_W	0.28

Using the same annotation used in experiment 1 we get the following concepts as the best matched concepts from teacher agents

- "Computer Science" from Cornell University
- "Electrical Engineering and Computer Science" from University of Michigan
- "Computer Science and Engineering" form University of Washington

Based on tie strengths calculated in table III, we used 12 positive examples and 12 negative examples from University of Michigan (Ag_M), 8 positive examples and 8 negative examples from Cornell University (Ag_C) and 7 positive examples and 7 negative examples from University of Washington (Ag_W). We apply the same learning techniques used before to learn the new concept "Programming Language". We got the following results:

1. Using K-NN for learning:

	True PL	True Non PL
Pred. PL	26	7
Pred. Non PL	1	20

Overall accuracy = 85.27%

2. Using Naive Bayes for learning:

	True PL	True Non PL
Pred. PL	24	3
Pred. Non PL	3	24

Overall accuracy = 88.73%

3. Using SVM for learning:

	True PL	True Non PL
Pred. PL	26	6
Pred. Non PL	1	21

Overall accuracy = 87.27%

We can notice also that the accuracy is highly improved using all three learning techniques than the accuracies of experiment 1 where no social networks are defined in the system. That means using social networks in our system affects positively in overall accuracies of the learning process.

X. CONCLUSION

In this paper, we present a prototype system for semantic search using concept learning and social networking. This system utilizes ontological concept learning mechanism to resolve ontological diversity and social networks to improve the learning and search resolution. The system depends on a multi-agent infrastructure to hide the complexity of communication and search processes from the users. This concept learning technique depends on sending positive and negative examples to identify the asked concept from teacher agents to learner agent. Through a detailed experiment we showed that teacher agents in our system are able to teach a learner agent new concepts even if they do not have a full definition of these concepts. Also we introduce social networks paradigm in communicating between agents and we showed that it improves the overall accuracy of the learning process. Although in this experiment we focused on a network of four nodes to illustrate the results, the method is general enough and can be applied to networks of larger number of nodes.

REFERENCES

[1] T. Berners-Lee, "Weaving the Web," Harper, San Francisco, 1999.

[2] S.J. Russell, P. Norving, "Artificial Intelligent: A Modern Approach," Prentice Hall, second Edition, 2003.

[3] E.H. Durfee, V. Lesser, "Negotiating Task Decomposition and Allocation Using Partial Global Planning, Distributed Artificial Intelligent," Issue 2, Vol 2, pp 229-244, 1989.

[4] Z. Wang, H. Tianfield, "A Study of Multi-Agent System for Network Management, Proceedings of the International Conference on High Performance Computing and Applications," pp 489–493, August 2004.

[5] J. Denzinger, S. Ennis, "Being the New Guy in an Experienced Team-Enhancing Training on the Job," AAMas'02, No. 1519, Bologna, Italy, 2002.

[6] Daconta, M.C., Obrst, L.J., Smith, K.T, "The semantic Web. A Guide to the Future of XML, Web Services and Knowledge Management," Indianapolis (IN), USA 2003.

[7] Robert A. Hanneman, Mark Riddle, "Introduction to social networks methods", 2005.

[8] Granovetter, M., "The strength of weak ties: A network theory revisited. sociological theory", pp. 201 – 233, 1983.

[9] Granovetter, M. S., "The strength of weak ties." The American Journal of Sociology, vol. 78, no. 6, 1973.

[10] Granovetter, M., "Getting a job: A study of contacts and careers," University Of Chicago Press, 1974.

[11] Lin, N., Ensel, "Social resources and strength of ties: Structural factors in occupational status attainment," American Sociological Review, vo;. 46, pp. 393 – 405, 1981.

[12] Burt, R., "Structural holes: The social structure of competition," Harvard University Press, 1995.

[13] Eric Gilbert and Karrie Karahalios, "Predicting tie strength with social network," Proceedings of the 27th international conference on Human factors in computing systems, Boston, MA, USA, pp. 211-220, 2009.

[14] J. P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, A. L. Barabasi, "Structure and tie strengths in a mobile communication network," Proceedings of the National Academy of Science of the United States of America, vol. 104, no. 18, pp 7332 – 7336, 2007.

[15] Andrea Petróczi, Tamás Nepusz and Fülöp Bazsó, "Measuring tie-strength in virtual social networks," Connections, vol. 27, no. 2, pp. 39 - 52. INSNA, 2007.

[16] Shimaa M. El-Sherif, Behrouz Far, Armin Eberlein, "Calculating the Strength of Ties of a Social Network in a Semantic Search System Using Hidden Markov Models," International Conference on Systems, Man and Cybernetics SMC2011.

[17] Olivier Capp_e, Eric Moulines and Tobias Ryd_en, "Inference in Hidden Markov Models," Springer, 2007.

[18] B. H. Far, C. Zhong, Z. Yang and M. Afsharchi, "Realization of semantic search using concept learning and document annotation agents," The 21th Int. Conf. on Software Engineering and Knowledge Engineering SEKE 2009, 2009.

[19] Zilan (Nancy) Yang, "A practical ontology-based concept learning in MAS," master thesis, University of Calgary,Department of Electrical and Computer Engineering, 2010.

[20] Shimaa El-Sherif, Behrouz Far and Armin Eberlein, "Using social networking in resolving conflicts of concept learning process," Proceedings of IEEE CCECE, 2010.

[21] Shimaa El-Sherif, Behrouz Far and Armin Eberlein,"Semantic search based on multi-agent system and social networking," Proceedings of the sixth IASTED International Conference, ACSE, pp. 103-110, 2010.

[22] B.H. Far, A.H. Elamy, N. Houari and M. Afsharchi, "Adjudicator: A Statistical Approach for Learning Ontology Concepts from Peer Agents," The 19th Int. Conf. on Software Engineering and Knowledge Engineering SEKE 2007, 2007.

[23] Z. Yang, C. Zhong, B. H. Far, "A Practical Ontology-Based Concept Learning in MAS," IEEE CCECE'08: Symposium on Computer Systems and Applications, 2008.

[24] A. Lally, K. Verspoor, E. Nyberg, "Unstructured Information Management Architecture (UIMA) Version 1.0," OASIS, 2008.