MATHEMATIC MODEL AND ARGUMENTATION THEORY WITH MULTI AGENT BASED ARCHITECTURE FOR PATTERN RECOGNITION

Bosikashvili Zurab, Archvadze Giorgi
Georgian Technical University

Summary

When agents communicate they do not necessarily use the same vocabulary or ontology. For them to interact successfully they must find correspondences between the terms used in their ontologies. This paper describes our work constructing a formal framework for reaching agents' consensus on the terminology they use to communicate. Each agent can decide according to its interests whether to accept or refuse the candidate correspondence. We are trying to construct pattern recognition model and how to use different ways to recognize patterns, in this paper we are constructing agent model which consist of different sub model.

Keywords: Argumentation Framework. Pattern Recognition. Multi Agent System.

1. Introduction

Multi agent system with argumentation framework is major component in distributed system and in human intelligence. To communicate many agents when they are working for common problem we should use argumentation framework. For pattern recognition when many agents are making decision for successful communication we are using argumentation framework. Using this, agents can accept or refuse candidate opinion. When agents communicate they are not using same vocabulary, to interact successfully they should use arguments and find correspondences between the terms they are using.

Successful communication is main problem in the architecture where are working many agents for one solution. Agents should share information, respect or refuse another candidates opinion using corresponding argument. Main purpose is to make common solution depending on appropriate experience and knowledge. In this article is described formal model how agents can recognize object.

Also in this paper is shown formal algorithm how to use argumentation framework for agents communication, how they are accepting and refusing another candidates opinion and finally how they making decision.

2. Pattern Recognition

How we can represent object which we are going to recognize? We should find pattern which location from this object is minimal and also we should find corresponding class index. We can define our object as X and pattern as P then:

$$i^* = \min_{\substack{i,j \\ 1 <= i <= n, \\ 1 <= j <= k}} \mathrm{Dist}(X, P_{ij})$$

For distance calculation we can represent our object as a tree (graph). Root vertex is our Object and sub vertexes are patterns and so on. There are three ways for object recognition. They are:

1) Bottom – Up: First we should determine bottom layer object components and closest pattern from this object, this process goes recursively before current object is not head object.

$$M(x,Y) = \min_{t_{ij} \in \bigcup_{c_i \in Y} T_{c_i}} \left(Dist\left(g(x), g(t_{ij})\right) + \sum_{x_i \in S(x)} M(x_i, \sigma(c_i)) \right)$$

where

- $T_{c_i} = \{t_{i1}, t_{i2}, \dots, t_{ik}\}$ - patterns for c_i- class;
- $Dist\big(g(\ \ ), g(\ \ )\big)$ Evaluated two graphs similarity, this is a problem of NP-complexity;
- $\sigma(c_i) = \{c_{i1}, c_{i2}, \dots, c_{im}\}$ – set of classes composed c_i;
- $g(x)$- Graph constructed by properties and relation of x;
- Recursion starting condition: $M(o, C)$;
- where $C = \{c_1, c_2, \dots, c_m\}$ - set of learned classes;
- Recursion ending condition: $\sigma(c_i) = \emptyset$;
- $opt\,\{M(o,C)\} = M(o, \{c_i^*\})$ – Optimal value.

But if this object has many sub objects and current object has also many sub object and etc. To get result we need big resources and this way is impossible (Fig.1).

2) Top – Bottom: In this case on top layer we can assume what type object we are trying to recognize and after this on sub layer we should proof our assumption. This process goes also recursively. To use this way requires knowledge what type object should be on the head of the tree. To collect such knowledge is not easy and requires experience (Fig.1).

3) Combination Top- Bottom and Bottom – Up : In this case we are using both Top-Bottom and Bottom – Up together (Fig.2).

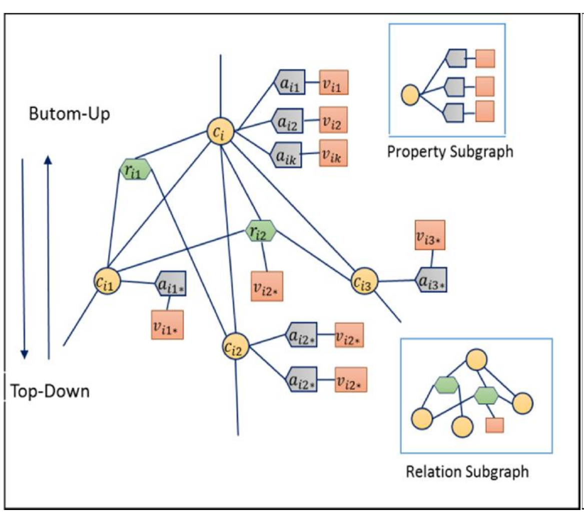

Fig.1

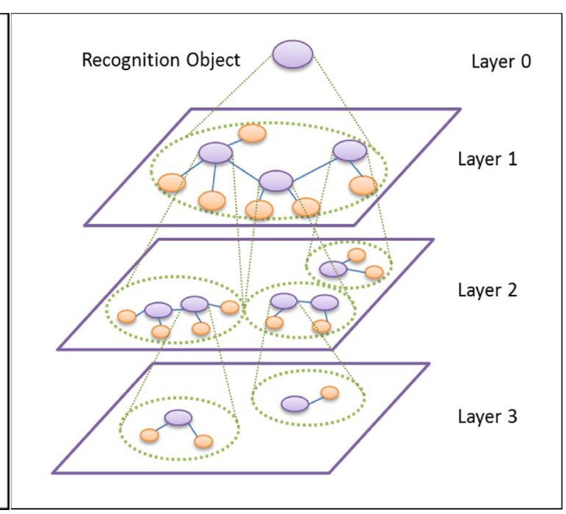

Fig.2

3. Agent Model

Many classical methods for representing and matching ontological knowledge in artificial intelligence (description logics, frame-based representations, semantic nets) are coming back into vogue, not least because of the “semantic web” initiative [1]. However, many problems remain when such approaches are applied to highly uncertain and ambiguous data of the sort that one is confronted with in computer vision and language processing. Much research remains to be done in fusing classical syntactic approaches to knowledge representation with modern factorized probabilistic modeling and inference frameworks [2].

4. Ontology Model

Ontology model can be represented as a tuple (Fig.3).

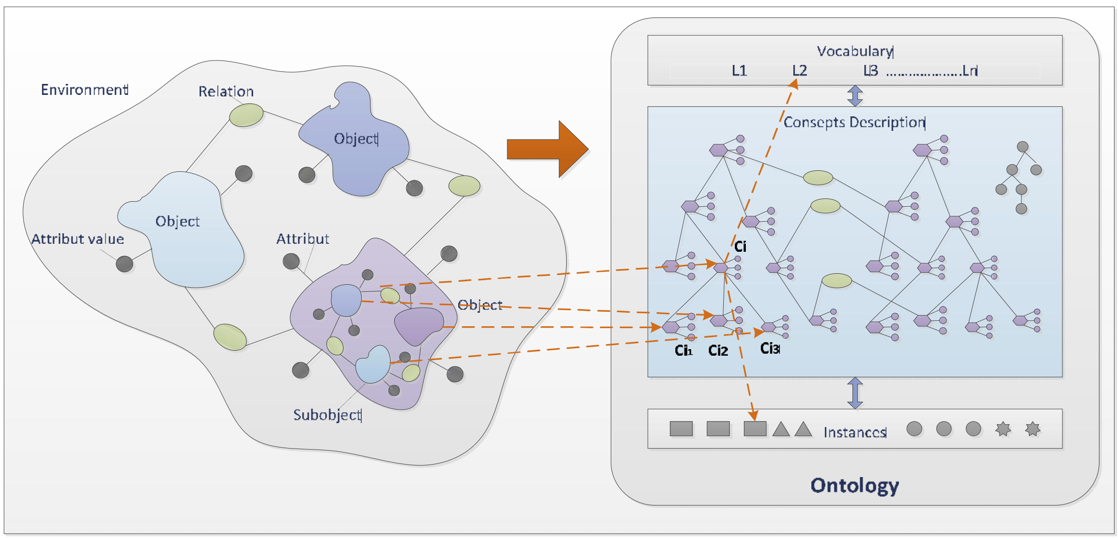

Fig.3

$$\Omega_a =< C_a, L_a, P_a^*, R_a^*, H_a, I_a, A_o > ,$$

where C_a- are concepts, H_a – concepts and attributes (properties and relations) hierarchy, L_a – label dictionary, I_a- copy of concepts set, A_o – set of axioms.

$$P_a^* =< \bar{P}, \bar{V}, Pr \quad > \text{- სადაც } \bar{P} = \{\bar{p} \mid \bar{p} \in \tilde{P} \subset \Re(P)\},$$

P – set of properties, $\Re(.)$- notes of all sub set. Every c_i or its copy has connection $\overline{p}$, this is a set of properties. For this import representation $f_p: \underset{\cdot}{C} \to \bar{P}$, this determines sequence of properties of the current class.

$$\text{E.g.} \quad f_p(Rectangle) = \{width, height\}.$$

Similarly define set of relations.

$R_a^* =< \bar{R}, \bar{V}, Pr \quad >$ - where $\bar{R} = \{\bar{r} \mid \bar{r} \ \in \tilde{R} \ \subset \Re(R)\}$, R is a set of relations, $f_r: C \rightarrow \bar{R}$, which determines current class relation.

E.g. $f_r(Line) = \{Cross, Vertical, Horizontal\}$

$\bar{V}$ – is set of property importance and $\bar{V} = \{\bar{v} \mid \bar{v} \ \in \tilde{V} \ \subset \Re(V)\}$, where V- is set of discrete value or $V \subset R^n$, $R -$ set of rational numbers.

E.g. $f_v(Width)= 5$, $f_v(Height)=[0,1]$, $f_v(Color) =$ {Black, White}.

$Pr = \{=, \neq >, <, \in, \notin, \geq, \leq, \subset, \not\subset, \subseteq, \nsubseteq, ..\}$ - is set of predicates which represents as an operator and specifies properties and relations importance.

E.g. Concept: "Rectangle", Property: "Width", Value: "<5"

Concept: "Rectangle", Property: "Width", Value: "=5"

5. Reasoning Model

A Reasoning model we can present as a tuple

$$\mathcal{R} = < S, Tr, Arg >$$

Of*, a semantic S, a trust Tr* and a *argumentation* model Arg.

6. Semantic

Commonly, the semantics is defined as an interpretation of concept by the real world objects and relations between them. In predicate logic, the interpretation of formulas is recursively define over the construction of the formula, in that we first assign constants, variables, and function symbols to objects in the real world [4]. We are interested in such formulas, which are valid for all interpretations or deductively follow from such formulas. Therefore, in our *formalism semantics* is expressed as

$$S = A_o \cup A_o^* \quad (.)$$

Where A_o is a *set of axioms* and A_o^* is a *deductive closure of* A_o .

7. Trust

As an Ontology define knowledge presentation model and more of knowledge is specified with uncertainty, the relationship between a statement and an experience cannot be properly captured by a binary truth-value anymore. Not only are binary truth-values not enough, even a multivalued logic may not be enough for adaptation if the truth-values are *qualitative*, rather than *quantitative*. Proposed solutions to this issue include various forms of *probabilistic logic* [Nilsson(1986); Adams (1998)] and fuzzy logic [Zadeh (1983)]. Therefore we must define and measure *trusting (evidence) for* a statement, given the system's experience [5].

Let f: $A^* \rightarrow R^k$ (k>0 , k∈N) mapping , then trust T is a set of $\{f_1,..,f_m\}$. Wang have defined the function $f_w: \rightarrow \{v,b\}$, $v,b \in [0,1] \subset R$, for evaluation evidence and confidence [6].

A_O^*- Construction traditionally are using methods of classical conclusion construction, when from existing facts, axioms and logical rules (Modus ponens, Modus tollens and etc.) are obtained new facts or verified is or not acceptable to get from existing system new rules. Because in all case we have open system with incompatible knowledge about external world. Maybe conclusions will be controversial or conclusion process will not be compatible in real time. In this case better way is to use mechanism of conclusion which humans are using in particular argumentation mechanism in multi agent system.

8. Argumentation Framework

- **Classical argumentation framework**

An argumentation framework is a pair AF= (AR, attacks), where AR is a set of arguments and attacks is a binary relation of AR. An attack (A, B) means that the argument A attacks the argument B. A set of arguments S attacks an argument B if B is attacked by an argument in S. The key question about the framework is whether a given argument A, A $\in$ AR, should be accepted. One reasonable view is that an argument should be accepted only if every attack on it is by an acceptable argument (Dung, 1995) [2,3].

Argumentation theory is an important field of Artificial Intelligence. In multi agent environment where agents try to purpose their own goals, cooperation cannot be taken for granted. To reach agreements and negotiation between self-motivated agents we should use argumentation framework. The purpose of argumentation is to resolve a conflict between agent's opinions. Argumentation framework has a big impact in multi agent environment.

- **Value-based Argumentation Framework**

We also can extend argumentation framework, namely, Value- based Argumentation Framework (VAF). The VAF allows determining which arguments are acceptable, with respect to the different audiences represented by different agents. Our agents apply different approaches and cooperate in order to exchange their local result (arguments).

The VAF is able to distinguish attacks from successful attacks, those which defeat the attacked argument, with respect to an ordering on the values that are associated with the arguments. It allows accommodate different audiences with different interests and preferences [7].

A Value-based Argumentation Framework (VAF) is a 5-tuple VAF = (*AR*, *attacks*, *V*, *val*, *P*) where (*AR*, *attacks*) is an argumentation framework, *V* is a nonempty set of values, *val* is a function

which maps from elements of *AR* to elements of *V* and *P* is a set of possible audiences. For each $A \in AR$, $val(A) \in V$.

9. Semantic Certainty Model

To construct certainty model, assume that object is concept and is represented as a finite number of layers in graph. On zero layer is located this object (concept) with its attributes (properties and relations) and all sub layer are located previous object sub object (concept sub concept) with its attributes [8].

Now we can describe certainty function:

$\varphi: g(O) \times g(T) \rightarrow R$, which describes how close our object from concept of pattern. Here $g: X \rightarrow \{G_1, G_2, \ldots\ldots\}$ are operators which reflect our object or class in their set of graph. Now we can define our relation case

1) Property Certainty:

$$\varphi\left(g^p\,(o), g^p\,(t)\right) \geq 0 \quad \text{If}$$

$$f_p\left(g(t)\right) \subseteq f_p\left(g(o)\right) \& \; \forall p\{\, p |\, p \in f_p\left(g(t)\right) \& \; (f_v(p_o) \subseteq f_v(p_t))\}\,)$$

Object property set completely includes respective set of patterns and for each pattern value performs defined condition with certain predicates.

E. g. If for pattern we have determined property “width<10” and value of object should be less than 10, E.g. “5” in this case we have complete certainty.

2) Relation certainty

$$\varphi(g^r\,(o), g^r\,(t)) \geq 0 \quad \text{If}$$

$$f_r\left(g(t)\right) \subseteq f_r\left(g(o)\right) \& \; \forall r\{if \;\; \exists p\left((r,p) \in g(t)\right) \& \; (f_v(p_o) \subseteq f_v(p_t))\} \;\&$$
$$Matchs\ Certainity\ (\sigma(t),\ \sigma(o)\}) \,,$$

Where σ- is an operator which every object connects set of sub object. Object relation are compatible to pattern relations if set of sub object is compatible is compatible to object sub object set and if pattern has properties and object sub objects can satisfactory pattern properties and values.

3) Other case:

$$\varphi(g(o), g(t)) < 0$$

In other case, when pattern is partially compatible to the object or not compatible we can say that we have uncertainty.

We are saying that object O is near to t_i pattern then t_j pattern, if $|\varphi(o, t_i)| < |\varphi(o, t_j)|$.

Import operator g_l^p, g_l^r , *which* describes l layer object properties and relations in sub graph, how clos is object from pattern when we can calculate with this framework:

$$\varphi(g(o), g(\,t_i)) = \sum_{l=1}^{k} [\,\alpha_l^p * \varphi\left(g_l^p(o), g_l^p(t_i)\right) + \alpha_l^r * \varphi(g_l^r(o), g_l^r(t_i))]$$

where α_l^p, α_l^r - are weight coefficients.

10. Qualitative Argument Generation by Agents

Qualitative argumentation generation starts when object properties compatibility is completed for generation arguments. We are using $\varphi\left(g_i^p(o), g_i^p(t)\right)$ function and are comparing their property sets. In Table 1 are represented C and H values. If we have pattern fully compatible to everything is clear this is certainty and another candidates cannot accept this compatibility.

Also when we have partial compatibility and compatibility function is positive then agent cannot accept this compatibility.

Tab.1

C	+(h)
Certainty	If all properties of object the o are similar to the properties of patterns t and all values of the properties are matched and $\varphi\left(g^{p}(o), g^{p}(t)\right) \geq 0$
Uncertainty	If some properties of the object o are similar to some properties of the pattern t or not all properties values are matched and $\varphi\left(g^{p}(o), g^{p}(t)\right) \geq 0$
C	-(h)
Uncertainty	Otherwise and $\varphi\left(g^{p}(o), g^{p}(t)\right) < 0$

11. Relative Argument Generation by Agents

Relative argumentation process starts when qualitative argumentation generation is completed and is attacked by agents and also is attacked every sub object and current object. In Table 1 are represented argument confidentiality and counterargument rules. Here to get arguments we are using function $\varphi(g^{r}(o), g^{r}(t))$

Tab.2

C	+(h)
Certainty	If all relations of object the o are similar to the relation of patterns t and pattern's all sub objects are matched to the sub objects of the object all relation property's values are matched and $\varphi(g^{r}(o), g^{r}(t)) \geq 0$
Uncertainty	If some relations of the object o are similar to some relations of the pattern and pattern's some sub objects are matched to some sub objects of the object or some relation property's values aren't matched and $\varphi(g^{r}(o), g^{r}(t)) \geq 0$
C	-(h)
Uncertainty	Otherwise and $\varphi(g^{r}(o), g^{r}(t)) < 0$

12. Structural Argument Generation by Agents

Structural agent's arguments generation reviewing as concepts in ontology context, when current object should be compatible any concept from this hierarchy. Of course argumentation generation starts from top to bottom and process must go as deep as possible. If on any layer we have incompatibility properties or relations and there is uncertainty then we should mark concepts which are located under current layer as unknown and we should stop generating arguments for them [6].

13. Preferred extension generation

Tab.3

N	Agent Designation	Code	Type	Count
1	Agent	AO	Slave	m
2	Property Value Calculation	AP	Slave	m+1
3	Object Property Compatibility to Pattern Property	APM	Slave	m+n
4	Object Relation Compatibility to Pattern Relation	ARM	Slave	m+n
5	Object Structure Compatibility to Pattern Property	ASM	Slave	m+n
6	Argument Generation	AA	Master	m+n
7	Make Decision	ADM	Guru	1

Where m- is count of object subclasses, n- is current class count.

First of all starts working agents which are computing properties and separating objects. Properties computation agents starts working in case when any object exists or is separated.

In this case also starting working property certainty agents, when property values are already calculated. After this activating argument generation agents. If for current class is already computed properties and exists counterarguments for this property about current class, then starting working relation certainty agents and next those relational arguments generator agents. If still exists for current class counterarguments then starting working proposed structure and arguments generator agents.

After every argument generator agent generates own argument set, they are passing their arguments, when for every agent knows other agents arguments then they are generating counterarguments set. Attack appears when object compatibility value is better than other agent's value and h- value is confidential, for example argument on $(m_1(o,t_1), P, certainity, +)$ appears attack from argument $(m_2(o,t_2), P, certainity, -)$, if m_1 and m_2 are same objects to compatible different patterns, or $(m_2(o,t_1), R, certainity, -)$ – case we have same pattern compatibility.

When arguments and certainty sets are created, agents should compute preferred extension. Argumentation set is globally acceptable if each element appear in every agent preferred extension set. If argument is not acceptable then we can say that it is indefensible.

14. Formal Algorithm

```
For All Agents
    For All Object
        Generation Qualitative Arguments ()
        Generating Relative Arguments ()
        Generation Structural Arguments ()
For All Agents
    For All Objects
        If exists relation (obj, patt, arg)
        Then we have certainty
        Else we have uncertainty

For All Certainty
    For All Agents
        If certainty attacked by agent and exist counterargument
        Then Certainty Confirmed
        Else
            If certainty attacked by agent and not exist counterargument
            Then Certainty Unconfirmed
```

15. Conclusion

In this paper we have outlined a framework that provides how agents can communicate, who use different ontologies. This is achieved using an argumentation process in which candidate correspondences are accepted or rejected, based on the ontological knowledge and the agent's preferences. We briefly described how agents are generating different type of arguments and how they can interact using their ontology and knowledge. We were talking what are multi agent systems and what is argumentation. In this article everyone can see clearly preferences of multi agent system with argumentation.

References:

1. Grimm S. (2011). A Unifying Formal Ontology Model. A Simple Formal Model for Unifying the Presentation of Ontologies in Semantic Web Research. http://www.joint-research.org/wp-content/uploads/2011/07/GrimmKEOD2009FormalOntologyModel.pdf.

2. Wang P. (2013). Non-Axiomatic Logic A Model of Intelligent Reasoning. Copyright *© 2013 by World Scientific Publishing Co. Pte. Ltd.*

3. Smith R. (2012). Aristotle's logic. in *The Stanford Encyclopedia of Philosophy*,ed. Zalta, E. N. (Spring 2012 edn.).

4. Nilsson N. J. (1991). Logic and artificial intelligence. *Artificial Intelligence* 47, 31–56.

5. Zadeh L. A. (1983). The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and System* 11, 199–227.

6. Gruber T. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 6(2): 199–221.

7. Merritt D. (2004). Best Practices for Rule-Based Application Development. http://msdn.microsoft.com/en-us/library/aa480020.aspx

8. Russell S., Norvig P. (2010). Artificial Intelligence: A Modern Approach. 3rdedn. (Prentice Hall, Upper Saddle River, New Jersey).

მათემატიკური მოდელი და არგუმენტაციის თეორია სახეთა გამოცნობისთვის მულტიაგენტებზე დაფუძნებული არქიტექტურით

ზურაბ ბოსიკაშვილი, გიორგი არჩვაძე

საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

აგენტების ურთიერთობებისას ერთმანეთთან, ისინი არ იყენებენ ერთიდაიმავე ლექსიკონს და ონთოლოგიას. იმისათვის, რომ მათი ურთიერთობა იყოს წარმატებული, საჭიროა იპოვნონ კორესპონდენცია ტერმინებს შორის, რომლებსაც ისინი იყენებენ საკუთარ ონთოლოგიებში. სტატია აღწერს ჩვენ სამუშაოს თუ როგორ ავაწყოთ ფორმალური გარემო, რათა მოხდეს შეთანხმების მიღწევა აგენტებს შორის იმ ტერმინოლოგიით, რომლებსაც ისინი იყენებენ ურთიერთობისთვის. თითოეულ აგენტს შეუძლია გადაწყვიტოს, თავისი ინტერესიდან გამომდინარე, დაეთანხმოს თუ უარყოს სხვა კანდიდატის კორესპონდენცია. ჩვენ ვცდილობთ ავაწყოთ ობიეტების გამომცნობი მოდელი და ასევე გამოვიყენოთ სხვადასხვა გზა, რათა ამოვიცნოთ ობიექტები, ამ სტატიაში ჩვენ ვაწყობთ აგენტების მოდელს, რომელიც შედგება განსხვავებული ქვემოდელებისგან.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ И ТЕОРИЯ АРГУМЕНТАЦИИ ДЛЯ РАСПОЗНАВАНИЯ ОБРАЗОВ С АРХИТЕКТУРОЙ НА МУЛЬТИАГЕНТНОЙ ОСНОВЕ

Босикашвили З., Арчвадзе Г.

Грузинский Технический Университет

Резюме

Когда агенты общаются друг с другом, они не используют одну и туже лексику и антологию. Для того, чтобы их общение было успешным, необходимо найти корреспонденцию между терминами, которые агенты используют в своих антологиях. Статья описывает вопросы построения формальной области, чтобы между агентами можно было заключить соглашение о той терминологии, которую они испльзуют для общения. Каждый агент может решить, исходя из своих интересов, принять или отвергнуть корреспонденцию другого кандидата. В работе рассматривается построение модели распознавания обьекта и испльзование различных путей для данного процесса. Модель агентов строится с использованием различных подмоделей.