The Scene Structure of Knowledge

Yi Wang[1, 2], Fei Li[1], XiaoPing Qiu[2]

[1] Department of Network Engineering, Chengdu University of Information Technology, Chengdu, P.R. China
[2]Intelligent Control Development Center, South-west Jiaotong University, Chengdu, P.R. China

Abstract

In this paper we propose a new knowledge representation system that is based on a collective structure called the scene, which is suitable for NLP. A scene is used to represent the internal structure of knowledge. The scene of a knowledge object integrates the referent nodes of other knowledge and can form a hierarchical structure. It is capable of representing both the hierarchical frame knowledge and the relational knowledge such as proposition, process and rule in a relatively uniform manner. Furthermore, a special kind of scene – the scene model, offers us powerful means to represent the meta-level model of knowledge and has the potential benefit of resolving the ambiguity in natural language understanding at semantic level.

Keywords: KRR, NLP, Scene, Scene Model.

1. Introduction

It is well known that the understanding of natural language should be based on a large mount of background knowledge in various forms. Traditional KRR, such as semantic networks, FRAMEs, finite-state machines and Petri Nets, are rigorously formed and not flexible enough to reflect some of the important characteristics of natural language. And they are too dedicated to be used as the fundamental structure of various background knowledge needed by NLP. A new research direction of NLP based on knowledge representation and reasoning systems emerged in recent years. For example, Lwanska developed the UNO Model based on the view that natural language itself is a powerful knowledge representation system[2][3]; Lenhart K. Schubert and Chung Hee Hwang developed Episodic Logic (EL) in[4][5], which is a natural language-like logic system and could be categorized into the non-standard first order logic, to meet both the expressive and inferential requirements of natural language understanding.

One common point of most of these works is that they are based on strong-grammatical languages such as English and Russian, and their major effort is to reflect the characteristics of natural language expression. But in weak-grammatical natural languages such as Chinese, a large amount of sentences are expressed in non-standard forms and the difficulty in grammatical parsing cannot be overlooked in any practical project. One purpose for us to develop a new knowledge representation system is to improve the precision of grammatical parsing with the assistant of semantic analysis. The other purpose is to provide a uniform representation structure for various background knowledge needed for NLP.

The knowledge representation we developed is based on a collective structure called the scene and an “IS-A” relational structure often used in representing taxonomic knowledge in many knowledge bases. The collective scene structure can represent our understanding about the composition of most knowledge, such as a situation, a context or even a domain. Combined with the “IS-A” relational structure, which is used to distinguish different roles of each component in a scene, the scene can then represent the structure of relational knowledge such as propositions, processes and rules in more clarity and detail.

There is a special kind of scene called Scene Model that can represent our general and rich knowledge about a category of concrete knowledge, such as the model of situations, the standard composition of a relational knowledge and the typical attribute set of an object. One advantage of scene model is that it is represented in the same form of scene and so any scene can be treated as a scene model if one or more of its roles are abstract concepts. Besides its ability to represent our meta-level knowledge of the model of the world, it is also beneficial to resolve the ambiguities in natural language and verify phrases in grammatical parsing. Scene model is similar to and more powerful than the lambda expression in CG[6][7].

2. Scene Structure of Knowledge

2.1. Scene

In our knowledge representation system, there are two kinds of knowledge: *primitive knowledge* and *compound knowledge*. Both of them are called *knowledge object* (KO in brief) and represented as *object nodes* in knowledge bases.

Primitive knowledge involves concepts, relations and individuals whose composition can hardly be isolated. *Compound knowledge*, such as propositions, events, rules, consists of sub-knowledge objects, which could be primitive knowledge objects or other compound knowledge objects. Compound knowledge is represented in the form of a scene. Figure 1 shows a simplest scene of a computer.

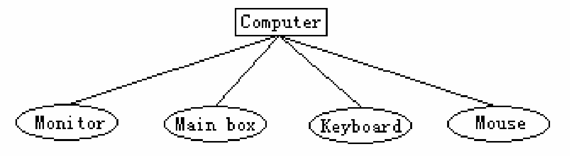

Fig. 1 The scene of a computer

Definition: Scene. A scene is a star network consisting of an *object node* and one or more *referent nodes*. The object node is the representation point of a compound knowledge object and is linked to the referent nodes by *integrative arcs*. Referent nodes are the references to the sub knowledge objects and are called the *roles* of the scene.

Scene is used to describe the composition information of a compound knowledge object. In figure 1, the rectangle node is the object node of computer, oval nodes are the referent nodes of the concepts Monitor, Main box, etc., and the solid lines represent the integrative arcs.

A referent node represents the fact that a knowledge object is used as a component of a compound knowledge. There are two kinds of referent nodes: *universal referent node* and *existential referent node*, corresponding to the universal and existential qualifiers in predicate logic respectively. A universal reference means that the knowledge containing the reference is adaptable to the entire extension of the concept that the referent node refers to. The universal reference of knowledge object K is expressed in symbols as < ∀ K>. An existential reference means the knowledge containing the reference is adaptable to only part of the extension of the concept that the referent node refers to. Expressed in symbols, an existential referent node of a knowledge object K is expressed as < ∃ K>, or simply <K>.

The integrative arcs are used to integrate the referent nodes of sub-knowledge objects to form a compound knowledge object, reflecting the fact that the compound knowledge refers to or integrates those sub-knowledge objects. Integrative arcs are represented by solid lines in the diagram, as shown in figure 1, and a pair of brackets {} in symbol expression. The example in figure 1 could be represented in text as:

Computer{<Monitor>, <Main box>, <Keyboard>, <Mouse>}.

There is a specially represented compound knowledge called *Classifying knowledge* containing the “IS-A” relation that is represented by a special arc called *classifying arc*. It can be used not only to represent the taxonomic knowledge such as *Cat is an animal*, just as many traditional KRR systems do, but also to distinguish the different roles of different sub knowledge objects in a scene, such as *Smith is the agent* in the scene of *Smith hit bill.* An arrowed line is used to represent classifying arc both in text and graphics. Figure 2 shows the classifying knowledge of *Cat is animal* that could be expressed in text as < ∀ Cat>→<Animal>.

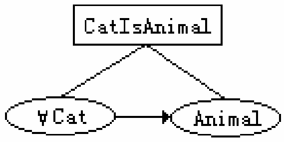

Fig. 2 The Scene of “Cat is animal”

The structure of a scene describes the composition information of a compound knowledge and can be naturally used to represent collective knowledge such as the composition of a car, the extensional set of a concept, the attribute list of a concept, and so on. For example, a simple attribute set of the concept Student can be expressed as:

Student{<Name>, <Birthday>, <Address>, …}

The extensional set of the concept Human can be expressed as:

{<Smith>, <Bill>, … , <Man>, <Adult>, …}

2.2. Relevant Knowledge

Definition: relevant knowledge. A *relevant knowledge* of a knowledge object is such knowledge that refers to or integrates that knowledge object as its role.

The relevant knowledge of a knowledge object describes its category, features, usages and relationships with other knowledge objects, and forms our understanding of the knowledge object, which could be a primitive knowledge such as a concept or an concrete object that may have taxonomic knowledge as its relevant knowledge, or a compound knowledge such as, for example, “Smith likes bill”, which might have the knowledge “It’s not true that Smith likes bill” as its relevant knowledge.

2.3. Scene Model

Definition: scene model. A scene model is such a special scene that its roles are abstract concepts and are called *role concepts*.

Scene model can reflects our common sense of the composition of a group of similar concrete scenes. For example, from our experience we have the composition knowledge of a typical shop, i.e. a shop consists of sellers, customers, goods and counters, etc., which may be represented as a scene model:

SceneModelOfShop {<Seller>, <Customer>, <Counter>, <Goods>, …}

The roles in a scene model are called role concepts because they are the representation of a group of possible objects that can be contained in a particular scene. Role concepts are usually omitted in most concept systems because they usually do not have names or words to represent them in natural language. In our point of view, *any knowledge object is a concept if and only if it represents a group of other knowledge objects*. So, like generic concepts, role concepts may have their own extensional set.

In scene model, the relevant knowledge that describes the extension and classification information of a role concept provides generating power of a scene model, which in turn is constrained by other relevant knowledge of the role concept. For example, the role concept Seller in the scene model of shop is classified as a human being, which means a seller in a shop can be any one in the world. Its relevant knowledge, such as the seller of a shop should live in the same city the shop located, and should be an adult, constrains the scope of persons that could be the seller. Furthermore, sellers should be dressed in uniform stand behind a counter in working, which can then be used to identify sellers from the crowd in a shop.

A concrete scene could be an instance of a scene model. For example, a particular shop might be expressed as following:

Shop A{<Jerry>, <Bill>, <Tom>…}
<<Jerry>> → <Seller>,
<<Bill>> → <Customer>,
<<Tom>> → <Customer>
…

Where each role in the scene of Shop A is classified into the corresponding role concept in the scene model of shop.

Note that <<Jerry>> refers to the role <Jerry> that is a reference of the knowledge object Jerry. The knowledge <<Jerry>>→<Seller> means that Jerry is a seller only when he is in Shop A. Other references of Jerry might be the roles of the scenes of other shops and might be classified into other role concepts of the scene model of shop, such as, for example, a customer or a manager.

2.4. Relational Scene Model and Relational Scene

Relational Scene Model is a scene model with one of its role concepts being a relational concept, or more explicitly, being regarded as the relation among other role concepts. It is used to describe our common sense of the structure of relational knowledge. For example, our common sense of the scene of eating is that there is an action that is recognized as eating and one or more agents who are eating and objects being eaten. The scene model of eating can be expressed as:

SceneModelOfEating
{<Eating>,<AgentOfEating>,<ObjectOfEating>}
<<Eating>>→<Relation>

The scene model of a relational concept is just a relevant knowledge of the relational concept, not its semantic content, which should be, for example, a special movement of mouth. Relational concept is the key clue that makes us associate with a scene model, just as we associate with the scene model of a typical shop by the clue of the concept Shop.

We use AgentOfEating and ObjectOfEating in the scene model expression of eating instead of Agent and Object because the word Agent may represent different concepts in different scene models. For example, the agent of believing can only be a person and so is different from the agent of eating that can be any living creature. AgentOfEating is just a kind of agent, and such knowledge might be just one of the relative pieces of knowledge of AgentOfEating. Following is some of the possibly relevant knowledge of AgentOfEating and ObjectOfEating:

<*AgentOfEating>→<Agent>
<*ObjectOfEating>→<Object>
<*AgentOfEating>→<Person>
<*AgentOfEating>→<Animal>
<*ObjectOfEating>→<Food>

Relational scene is an instance of a relational scene model. In the process of instantiation, the action or relation as well as objects in a real situation is recognized and classified as one of the role concepts in the corresponding scene model. For example, the knowledge "Cat eats Mouse" is an instance of the scene model of eating:

{<AnAction>, <Cat>, <Mouse>}
<<AnAction>>→<Eating>
<<Cat>>→<AgentOfEating>
<<Mouse>>→<ObjectOfEating>

The reason of using AnAction instead of Eating in the above scene is that the action of cat eating mouse is a special eating that is different from the action of snake eating mouse. Thus we assume that the relation in any knowledge is a sub-class concept or an instance of the corresponding relational role concept. AnAction can be briefly represented as [Eating] that implies [Eating]→<Eating>. So the above scene can be simplified as the following:

{[Eating], <Cat>, <Mouse>}
<<Cat>>→<AgentOfEating>
<<Mouse>>→<ObjectOfEating>

2.5. Context And Knowledge Base

Context is essential in any KRR because it provides background and default knowledge in reasoning. In our knowledge representation, context has the same structure of normal compound knowledge. Because the fundamental structure of compound knowledge is actually a collection, a *compound knowledge is the context of the knowledge objects that its roles refer to*.

With this uniform structure, the complicated context relationships in human knowledge can be represented: a hierarchical context is a compound knowledge with one or more of its roles referring to other compound knowledge objects; on the contrary, any knowledge object may have any number of contexts if it is referred by more than one compound objects.

In reasoning, the validation of an object should be based on one of its contexts. *An object is valid in a context if and only if it is consistent with other knowledge in the context*. It is possible that an object is valid in a context but is invalid in another context.

Different from the organization of knowledgebase in CG and many other KRRs, a knowledgebase in our designation is not a context but a physical space storing all learned knowledge objects in the form of object nodes. We believe that a knowledgebase should be able to contain more than one context, for example, the brief context of the owner of the knowledgebase, the real world context, various fictional world contexts, and even other's brief context that the owner knows. These contexts need not to be consistent with each other. Such an organization of knowledgebase and context could provide us flexible and powerful means to express extremely complicated relationships in our knowledge.

3. Experiment

Chinese is a weak-grammatical language and the parsing precision of Chinese parser is too low to provide us a practical platform to acquire knowledge from real text and to reason based on knowledge represented in natural language. One of our original purposes of developing the new knowledge representation is to improve the precision of the grammatical parser with the assistant of semantic analysis. We conducted an experiment to verify the effect the semantically-assisted grammatical parsing.

In our expectation, the parsing precision and efficiency might be improved if a large number of semantically invalid phrases generated in the process of parsing could be pruned by semantic analysis as early as possible. In human reading, a phrase is invalid if we cannot figure out its corresponding possibly valid knowledge object based on our experience, even if it is grammatically correct. Ideally, the ability for a computer to image a possibly valid knowledge object could be implemented with the scene model in our knowledge representation. But in practice, the large number of scene models needed to ensure the reliability of semantic verification makes it practically impossible to be induced and entered into the system manually and the auto-acquisition of scene models still need further studies, we used the *kernel knowledge object* instead of the scene model of knowledge to verify phrases.

A kernel knowledge object is the abstraction of a complicated knowledge object. In the abstraction for kernel object, all attributes of the original object are omitted and all of its roles are replaced by their named super-class concepts, i.e. the concept of word. For example, the kernel object of the knowledge "the clever boy", which should be represented as <x>→<boy> and {<ATTACHTO, <clever>, <x>}, would be the concept of "boy" by pruning its attribute Clever and replacing it with its named super-class concept boy.

Similarly, the kernel object of "The clever boy loves mathematics very much." should be:

{<Love>, <Boy>, <Mathematics>} (1)

Kernel objects should be stored in knowledgebase directly without any context. So discussing their validation will not make any sense. For example, expression (1) does not mean that all boys love mathematics, it just means any knowledge with that kernel is a possibly valid knowledge. So, when parsing new sentences, for example, "Boys loving mathematics should take part in the class", the parser may generate a sub phrase "boys loving mathematics", which would be converted into knowledge structure as:

<y> → <boy>
{<y> + <multiple>}
{<Love>, <y>, <Mathematics>} (2)

Because the kernel object of y is boy, the kernel of knowledge object (2) is the same as (1), so the phrase "Boy loving mathematics" is a possibly valid phrase.

In Chinese parsing, since the category of each word in a sentence is tagged in the stage of word segmentation and is just an estimation, and there is no morphologic evidence to distinguish the different role of a word in different environment, the over-generation of grammar rules is very high and a large mount of ridiculous phrases may be generated in grammatical parsing. In such a case, the semantic verification of phrases based on checking of the kernel objects would be very useful.

In practice, to enhance the efficiency of semantic verification and reduce the memory requirement of the large amount of kernel knowledge objects, we store the hash value of kernel knowledge objects in a sorted array instead of storing their exact representation. In verification, the hash value of the kernel KO of a phrase can be quickly searched in the kernel hash array, and if found, it might be a valid phrase. Otherwise, the phrase would be rejected by the parser.

In the experiment, we used the TCT Chinese tree bank as the learning material, which contains about one million Chinese characters. All phrases in the tree bank were retrieved and converted into KOs, and their kernel hash values are calculated and stored and sorted in the kernel hash array, which then is used to assist the parsing process of the role-reversed chart parser, which uses probabilistic context free grammars PRORD-PCFG[1] that is constructed from TCT Chinese tree bank. Whenever the parser generates a completed edge (a phrase), it is converted into a KO and verified in the kernel hash array. If it is a possibly valid phrase, the probability of the corresponding edge is largely increased to approximate 1 by following formula:

$$P(r) = \mathrm{Power}\ (P(r), 0.001)$$

Otherwise, the edge will be rejected.

The following table shows the result of the compared with that of pure PRORD-PCFG in terms of CBs(Crossing Brackets in a sentence), 0CB(percent of Zero Crossing Brackets), 1CB(percent of 1 Crossing Brackets):

Table 1. Experiment Results

	PRORD-PCGF	Semantic Assistant
CBs	3.46	0.48
0CB	27.61	42.32
1CB	45.24	44.75

The result shows that the accuracy of grammatical parsing is significantly improved with the semantic assistant.

4. Discussion

From the brief description above, we can see some of the advantages of this knowledge representation. First, the scene structure offers us a relatively simple and uniform approach to represent both the hierarchical frame structure and relational structure of knowledge. Second, scene schema can potentially benefit in NLP. The accordance between the hierarchical structure of phrase in natural language and the hierarchical scene structure of compound knowledge object makes it possible to map them to each other directly. Further more, the constraint power offered by the relevant knowledge of the role concept in the scene model could be helpful in both resolving the ambiguity of the words in NLU and choosing proper word in NLG. This type of dual-directional mapping needs to be studied in more detail. Last, its expressive power of representing the most complicate context relationship in knowledge may benefit the context organization of knowledge in expert systems and, combined with the scene model, help us to find omitted components in the sentences of natural language.

5. Acknowledge

This research is supported by the Scientific Research Foundation of CUIT CRF200412 and by NNSF 60474022.

6. References

[1] Hao Zhang, Qun Liu, Kevin Zhang, Gang Zou and Shuo Bai, "Structural Context Conditioned Probabilistic Parsing of Chinese Statistical Chinese Parser ICTPROP, Technology Report", http://mtgroup.ict.ac.cn/docs/download.php?doc_id=203~liuqun/research/publications/techreport.pdf, 2003.

[2] Lucja Iwanska, M. "Natural (Language) Temporal Logic: Reasoning about Absolute and Relative Time," *International journal of Expert Systems* 9(1), pp. 113-149, 1996.

[3] Lucja Iwanska, M. "Logic Reasoning in Natural Language: It Is All about Knowledge," *Minds and Machines* 3(4), pp. 475-510, 1993.

[4] Lenhart Schubert, K. "Framing the Donkey: Toward a Unification of Semantic Representations with Knowledge Representations," *Proc. of AAAI Fall Symposium on Knowledge Representation Systems Based on Natural Language*, pp. 9-11, 1996.

[5] Lenhart Schubert, K. and Chung Hwang, H. "An Episodic Knowledge Representation for Narrative Texts," *Proc. of the First International Conference on the Principles of Knowledge Representation and Reasoning (KR'89)*, pp. 444-458, 1989.

[6] John Sowa, F. "Conceptual Graphs Summary," *Conceptual Structures: Current Research and Practice*, P. Eklund, T. Nagle, J. Nagle, and L. Gerholz, eds., Ellis Horwood, pp. 3-52, 1992.

[7] John Sowa, F. "Conceptual Structures: Information Processing in Mind and Machine," *The System Programming Series*, Addison Wesley, 1984.