Chinese Sentences Generating in Network Fault Diagnosis Expert System

Peiqi Liu, Zengzhi Li, Yinliang Zhao

Institute of Architecture & Network, Xi'an Jiaotong University, Xi'an 710049, China

peiqiliu@163.com

Abstract

The expert system is an important field of the artificial intelligence. The traditional form of the interface of the expert system is the command, menu and window at present. These interfaces limit the application of the expert system and embarrass the enthusiasm of using expert system. Combining with the study on the computer network fault diagnosis, the model of expert system of network fault diagnosis and natural language interface of expert system are presented. In this expert system, the interface can understand and generate Chinese sentences. Using this interface, the user and field experts can use expert system to diagnosis the fault of network naturally. In this article, the method of Chinese sentence generation of natural language interface is discussed in detail. According to this method, the algorithm GSCG of Chinese sentence generation from conceptual graphs is designed. Using this model, the expert system of network fault diagnosis has been developed with Prolog**.**

Keyword: Natural Language Generation, Natural Language Interface, Conceptual Graph, Expert System, Fault Diagnosis

I. Introduction

Since the expert system was born, it's theories and applications have been developed rapidly. The expert system deals with the realm of the agriculture, industry, national defense, education and science. Inputted the keywords of the expert system in the search engines of the network, we can get millions results about the expert systems.

The latest theories and achievements of the expert system provided a good tool for network fault management. The fault management includes the fault reporting, analyzing, locating, separating, restoring. The fault diagnosis is composed of fault analyzing and locating. The network fault diagnosis expert system can diagnose the faults of the network with a knowledge base which comes from these aspects: the expert's experience, scientific and technical literature, documents and books, acquisition knowledge from the network based on SNMP, which is short for Simple Network Management Protocol. In according to SNMP protocol, the equipments of the network are divided into SNMP management and SNMP agent. The communication between them is using primitives. The traps in the communication are PDU, be short for Protocol Data Unit. The traps PDU is sent by

SNMP agent when some faults emerged in network. For supervising the status of the network, SNMP management use GET primitive to inquire about all SNMP agents. In the network management, the information about network fault is obtained by using the traps PDU of SNMP. The format of traps PDU is more complicated than others. By analyzing the format of PDU carefully, we can define these traps as

Alarm=(Time, Source, Type, Severity)

Where, Time is time stamp, which is a time from the system starting to fault appearing; The Source denotes an IP address of network equipments; The Type demotes the type of fault, which be divided into the general traps and the particular traps; the Severity denotes a serious degree of network fault, which is divided into Cleared, Indeterminate, Information, Warning, Major and Critical. When all traps are saved into a database, we use KDD technique to mine the knowledge in the transaction database, which is transformed from traps database. The knowledge can be used to diagnose the fault in the network fault diagnosis expert system.

The network fault diagnosis expert system is an expert system with the natural language interface. In the past, the man machine interface of all expert system adopted command lines, menus and windows. Because there are some limitations in these interfaces, it restrains the development and application of expert system seriously. In order to increase the efficiency and convenience of expert system, the natural language interface is adopted in the network fault diagnosis expert system. This interface can comprehend the Chinese sentences inputted by user, and can generate Chinese sentences according to the feedback information of the expert system.

In this article, the model of fault diagnosis expert system of network is presented. It has natural language interface, and the knowledge is represented by the extended production rule [1]. According to this model, the network fault diagnosis system and its interface have been designed in the research of national natural science foundation. The method and algorithm, generating Chinese sentences from conceptual graphs in the expert system, will be discussed in detail in this article.

II. Structure of Network Fault Diagnosis Expert System

The expert system of network fault diagnosis consists of the expert system and its natural language interface. The natural language interface can understand and output Chinese sentences in the field of computer network. The expert system uses the knowledge representation of extended production rules to carry out the inference. The reasoning results are conceptual graphs. This conceptual graph has been transformed into Chinese sentences. The network fault diagnosis expert system consists of 5 parts (see figure 1).

The first part of this system is pretreatment and morphology. When Chinese sentence is inputted, the system cuts the sentence into some words, and then recognizes each word and processes synonym. If the word isn’t included in the vocabulary, the system will report the error information. Moreover, this module can also ignore some unimportant words in progress of the sentence. It makes the system have enough flexibility. The function of this part is lexical analyzer. The output of this part is list of words that system can recognize.

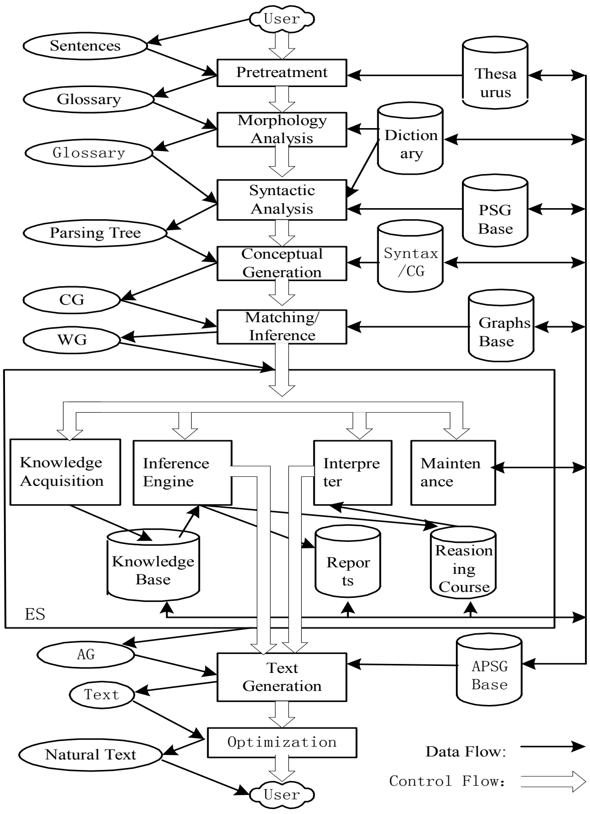

Fig. 1. Structure of the network fault diagnosis expert system

The next part is syntactic analysis. The purpose of this part is using a dictionary and the Phrase Structure Grammar（PSG）rules to carry out the syntactic analysis. If the Chinese sentence is correct, the system outputs a parse tree of the sentence that acts as the base of generating the Conceptual Graphs.

The third part is Semantic Analysis. The semantic analysis consists of the matching / inference and creating conceptual graphs. Firstly, it converts the parsing tree into the conceptual graphs. Next it achieves inference by using the conceptual graphs and graphic base. The result of the semantic analysis is the conceptual graphs that accords with the semantic. This conceptual graph is the Working Graph that is used to drive the expert system to work.

The expert system is an important part in this system. It consists of the inference engine, interpreter, the knowledge acquisition, the maintain knowledge and some knowledge bases. The inference engine reasons the result with WG and the knowledge base, and it watches the faults of network system. The conclusion of expert system inference is also a conceptual graph. We call this conclusion Answering Graph. By tracking the reasoning route, the interpreter of the expert system can explain the process of inference. The maintenance module of knowledge base is responsible for editing, adding and deleting the knowledge in the knowledge bases. When there is not enough knowledge in the knowledge bases, the expert system can use the knowledge acquisition to gain knowledge online [2].

The last part is Text Generation. The model of Text Generation can be divided into the language generation and the language optimization. According to the APSG, which is the Augmented Phrases Structure Grammar, the language generation produces sentences by using answer graph. In order to generate standard natural language, the system can optimize the sentence. Because the text generation is a main part in this article, it will be discussed in detail.

III. Conceptual Graphs

The conceptual graphs was presented by John F. Sowa, is an American scientist, in [3[4]. It is an important method of knowledge representation based on linguistics, psychology and philosophy. It is also a new method of knowledge representation developed from the semantic network. This kind of knowledge representation is based on graphs. It will be defined as

Definition 1: A conceptual graph is a finite, connected and bipartite graph.

In conceptual graphs, we always use conceptual node and relation node. The conceptual node can represent any concrete concept, abstract conception, real object and attribute. It is drawn as a box with conceptual type label and conceptual reference, which can be divided into two parts by semicolon. The conceptual reference can be individual, a set consisted by some individuals and the number begin with symbol #. When conceptual type is a special conception, the reference will be omitted. The relation node can represent deep layer of relation between the concepts. It can express not only general relation, but also case relation. The relation node is drawn as a circle with relational type label. For example, the symbols of CAT and RED are concepts; the symbols of AGNT, OBJ and MANR are relations. An arc with an arrow that links a box and a circle shows the effect relation between nodes. For example, the sentence "A cat eats meat with paw " can be shown in figure 2.

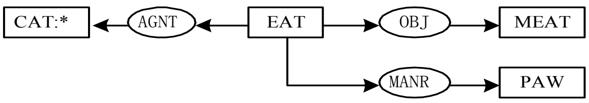

Fig. 2. An example of Conceptual graphs representation

In the figure 2, the relation AGNT is the issuer of EAT; The relation OBJ is the object of EAT. The relation MANR is a method of EAT. The conceptual CAT: * shows that it could be any cat. If we take the relational node for translation of conceptions, the conceptual graph can be defined as

Definition 2: The conceptual graph CG is a 4-tuples,

$$CG=(CS,RS,\delta,R)$$

Where,

$CS=\{C1,C2,\ldots,Cn\}$,the set of conceptions in CG

$RS=\{R1,R2,\ldots,Rm\}$, the set of relations in CG

$R=\{C|C\in P(CS)\wedge|C|\leq 1\}$,the results of mapping

$\delta:CS\times RS\rightarrow R$, the mapping between nodes

The box and circle notation, called the display form of conceptual graphs, can't be typed easily at a computer terminal. In order to process it by computers easily, we replace box with square bracket and circle with bracket. This is the linear form of the conceptual graphs. In the linear form, the conceptual graphs in the figure 2 can be shown as

[EAT]—(AGNT)→[CAT: *]

(OBJ)→[MEAT]

(MANR)→[PAW]

Obviously, the concept graphs can represent not only the case grammar relation AGNT and OBJ, but also can represent the deep relation MANR. And there is a natural relationship between the conceptual graphs and the natural language.

Ⅳ. Natural language Generation

In the figure 1, the output of the expert system is a meaningful conceptual graph. One of the important tasks of natural language interface is generating Chinese sentences according to conceptual graphs. Other problems of natural language interface will be discussed in [5][6].

A. Basic Conceptions

Definition 3: A conceptual graph is called canonical graph if it is a meaningful conceptual graph. For example, we have a conceptual graph

[Sleep]→(AGNT)→[Idea]→(COLOR)→[Green]

This conceptual graph consists of 3 different concepts. So it is not a canonical graph.

Theorem: There is a one-to-many mapping between canonical graph and natural language sentences.

Proof: First, the vocabulary, syntax and semantic relation of any natural language sentence can be created by the sentence analysis. This analysis includes Morphology Analysis, Syntactic Analysis and semantic Analysis. We use the vocabulary, semantic relation and syntax attribute as CS, δ and RS respectively. The canonical graph will be constructed by CS, δ and RS. Whereas, using the depth-first algorithm to traverses all branches of canonical graph, all phrases of natural language will be created. According to different customs of natural language, the different forms of natural language can be created.■

According to the theorem, arbitrary canonical graph can generate the natural language sentence. In general, the APSG can be used to generate natural language sentences. The form of APSG form is

S(condition for applying this rule)→
NP(action for moving the current connection node to the subject and getting the person and number from the current conception)

VP(action for moving the current connection node to the main ACT, for copying person and number from the NP record, and for copying mode and tens from the S record.)

In the rule of APSG, a natural language sentence consists of the Subject part and the predicate part, denoted by NP and VP respectively. In the left part of the rule, the content in the brackets is the condition for applying this rule. The form of condition is

Attribute1 is(not) attribute2

If condition is True, the rule of APSG is fired. Parameters used in the rule are illuminated in Table 1.

Table 1. Parameters in common used in APSG.

Parameters	Functions
□	The current conceptual node
○	The current relational node
type()	Return the type of current node
reference()	Return the reference of concept
Number	The number of Person
Tense	The tense of the Verb
Voice	The voice of the Verb

Moreover, there are 3 kinds of operations in the rule. The operations are

1. Assignment: It causes a new attribute to be created for a record and assign it a value.

2. Move: It causes the symbol to advance to the next node of the utterance.

Move relation node→□

Move relation node←□

3. Mark: It sets the utterance mark on a conception or the traversal mark on a relation.

Mark □→ATTR traversed

Mark □ Uttered

There is different set of ASPG rules corresponding to the canonical graph. For example, following canonical graph is a canonical graph of the network fault diagnosis in Chinese

[发出]－(AGNT)→[服务器]→(CLAS)→[本域内]

(OBJ)→[警告]→(CLAS)→[超时]

This canonical graph uses following rules:

S(type(○) is AGNT)→

NP1(Move AGNT→□; Mark AGNT→□ traversed)

VP(Move AGNT←□; Voice:=Active)

NP1(□←AGNT is not traversed)→

MOD(Move □←CLASS)

NOUN(type:=type(□))

VP(Voice is Active)→

VERB(type:=type(□))

NP2(Move OBJ→□)

NP2(□←OBJ is not traversed)→

MOD(Move □←CLASS)

NOUN(type:=type(□))

MOD(□←CLASS is not traversed)→

MODIFER(type:=type(□))

NOUN(type=服务器)→"服务器"

NOUN(type=警告)→"警告"

VERB(type=发出)→"发出"

MODIFER(type=超时)→"超时的"

MODIFER(type=本域内)→"本域内的"

Using these rules, the system can generate Chinese sentence from the canonical graph:

本域内的服务器发出超时的警告。

The meaning of this Chinese sentence is *the server in this domain issues a time out warning*.

B. *Chinese Sentences Generation*

Chinese sentences generation is a complex procedure. It relates to representation of the conceptual graphs, relation handling and conception handling. These parts will be discussed in this section.

The implement of algorithm of Chinese sentence generation is related to storage form of the conceptual graphs. In the algorithm of generating Chinese sentence, the conceptual graphs is represent in tuples. The graphs(no, Rlist), concept(no, Cname, Creference) and relation(no, Rname, Clist_out, Clist_in) are conceptual graphs, relations of conceptual graphs and concept of conceptual graphs respectively. For example, the conceptual graphs in this section can be represented as

Concept(1,"发出","")

Concept(2,”服务器”,””)

Concept(3,”本域内”,””)

Concept(4,”警告”,””)

Concept(5,”超时”,””)

Relation(1,”Agnt”,”2”,”1”)

Relation(2,”Obj”,”4”,”1”)

Relation(3,”Loc”,”3”,”2”)

Relation(4,”Class”,”5”,”4”)

Graphs(1,”1,2,3,4,5”)

Accord to the representation of the conceptual graphs, the algorithm of Chinese sentence generation can generate Chinese sentence by scan graphs and relation tuples. In conceptual graphs, all relations are dualistic relation, and each relation is a phylum relation. Therefore, we can get a sentence by traversing all relations node in the conceptual Graph. The algorithm of Generating Sentence from Conceptual Graphs, abbreviated to GSCG, consists of seeking the key concept verb, generating pattern from conceptual graphs, replacing reference and type of concept in the pattern. The algorithm GSCG can be described as follows:

```
Algorithm: GSCG
Input: Conceptual graphs CG
Output：Sentence Sent
Begin
RS=Rlist;
CS=Clist; CS1=CS; Sent=Φ;
//Seeking Key verb in Conceptual graphs
While RS<>Φ
{  R0=firstelement(RS); RS=RS-R0;
    Relation(R0,Rname,X,Y);
     If Rname=”AGNT” Then {S=”X,Y”;C={X,Y};RS1=RS;RS=Φ;}}
   //Generating sentence pattern of Conceptual graphs
     If S0<>”” Then
   {  RS=RS1;
        While RS<>Φ
        {  R0=firstelement(RS);
          Relation(R0,Rname,X1,Y1);
          C1={X1,Y1};
          If C1∩C=Φ  Then RS=RS-R0;
          Else
          {  Case
               {
                Rname=”OBJ”:  S0=”Y1,X1”;
```

```
                  Rname="LOC":  S0="在,Y1,的,X1";
                  Rname="Attr": S0="X1,的,Y1";
                  Rname="Time": S0=" 在,Y1,X1";
                  Rname="Spend":S0="用,X1,Y1";
                  Rname="Ride": S0="乘,X1,Y1";
                  Rname="Use":  S0="用,X1,Y1";
                    ⋮            ⋮
                 }
          S=connect_Sentence(S,S0);
          C=C1∪C;RS1=Rs-R0;RS1=RS;}}
    //Replacing type and reference of concepts
    While S<>Φ
    {  N=Firstelement(S);
       If N<>Number Then Sent=Sent+N;
       Else
       While CS<>Φ
       {  C0=Firstelement(CS);
          Concept(C0,Cname,Refence);
          If C0=N Then
             { Sent=Sent+Proc(Refence)+Cname;
               CS=CS1-C0; CS1=CS;}
          CS=CS-C0;}
       S=Last(S);}
    // Output Chinese sentences
    printf(Sent);
End
```

It is easily gained that the complexity of the algorithm is O(n+m) when the conceptual graphs has n relations and m concepts. In the algorithm, Firstelement(), Lastall(), Connect_Sentence() and Proc() are important procedures. Among them, the Firstelement() access first element in the list, the Lastall() generates a new list except for the first element, the Connect_Sentence() merges two languages patterns into a new one, and the Proc() deal with conceptual nodes in the conceptual graphs. In GSCG, the pattern is a list including Chinese characters and number. In order to process these patterns easily, we abstract the pattern for numeric sequence by ignoring Chinese characters. For example, the pattern "在,1,的,2" abstract for "1,2".

Definition 5: let the patterns are :

$S=A_1A_2\ldots A_{i-1}A_iA_{i+1}\ldots A_n$, A_i（$i\in\{1,2,3,\ldots,n\}$）is a number of concept

$S_1=A_iB_1B_2\ldots B_m$, B_j($j\in\{1,2,\ldots,m\}$) is also a number

The merging of S and S1 is:

$S=A_1A_2\ldots A_{i-1}A_iB_1B_2\ldots B_mA_{i+1}\ldots A_n$

In the algorithm, there are many different combinations between concepts and relations. We will discuss these combinations in common used in the algorithm. These are

- [C1]→(Agnt)→[C2]

It means that C2 is an agent of C1. The sentence is C2C1. For example, these is the conceptual graphs

[吃]→(Agnt)→[猫]

It can be translated into 猫吃.

- [C1]→(Obj)→[C2]

It means that C2 is an object of C1.The sentence is C1C2. For example, these is the conceptual graphs

[吃]→(Obj)→[老鼠]

It can be translated into 吃老鼠.

- [C1]→(Loc)→[C2]

It means that C2 is a location of C1.The sentence is 在 C2 的 C1. For example, these is the conceptual graphs

[计算机]→(Loc)→[本域内]

It can be translated into 在本域内的计算机.

- [C1]→(Time)→[C2]

It means that C2 is a time of C1.The sentence is 在 C2C1. For example, these is the conceptual graphs

[吃]→(Time)→[上午]

It can be translated into 在上午吃.

- [C1]→(Spend)→[C2]

It means that C2 is a spend of C1.The sentence is 用 reference(C2)type(C2)C1. For example, these is the conceptual graphs

[买]→(Spend)→[钱:#10]

It can be translated into 用 10 元钱买.

- [C1]→(Ride)→[C2]

It means that C2 is a traffic tool to fulfilling C1.The sentence is 乘 C2C1. For example, these is the conceptual graphs

[去]→(Ride)→[火车]

It can be translated into 乘火车去.

- [C1]→(Use)→[C2]

It means that C2 is a tool to fulfilling C1.The sentence is 用 C2C1. For example, this is the conceptual graphs

[写]→(Use)→[笔]

It can be translated into 用笔写.

- [C1]→(Attr)→[C2]

It means that C2 is a attribution of C1.The sentence is C2 的 C1. For example, these is the conceptual graphs

[信息]→(Attr)→[错误]

It can be translated into 错误的信息.

- [C1]→(Class)→[C2]

It means that C2 is a subclass of C1.The sentence is C2 的 C1. For example, these is the conceptual graphs

[水]→(Class)→[无盐]

It can be translated into 无盐的水.

- [C1]→(Trope)→[C2]

It means that doing C1 look like C2.The phrase of the sentence about this conceptual graph is 像 C2 一样 C1. For example, these is the conceptual graphs

[花费]→(Trope)→[流水]

It can be translated into 像流水一样花费.

Through the processing of the conceptual graphs, we can gain a model of sentence made up of the concepts of the graphs. Then the serial number of the model of sentence is replaced by the concepts in the conceptual graphs. The result is a Chinese sentence corresponding to the graphs. For example, the conceptual graphs in this section can be translated into

"在,3,的,2,1,5,的,4"

This pattern of sentence correspond to the Chinese sentence is

本域内的服务器发出超时的警告。

The algorithm of Chinese sentence generation is very simple. It consists of two parts. First the algorithm generate the model of the sentence correspond to the conceptual graphs deduced from the expert system, then the algorithm translate the model of sentence into Chinese sentences. The complexity of the algorithm is related to the number of relation in the conceptual graphs. When the number of relations in the graphs is n and the number of concepts is m, the complexity of algorithm GSCG is O(n+m).

Ⅴ. Conclusion

In this article, we have presented a model of the network fault diagnosis expert system firstly. Then the natural language interface and algorithm GCCG are discussed chiefly. Based on this model, we have implemented a computer fault diagnosis expert system with natural language interface in Prolog. The expert system can understand and generate Chinese sentences in the field of network faults. As long as renewing the knowledge in the knowledge base, this natural language interface can be used to any software system.

References

[1] Peiqi Liu, Zengzhi Li and Yingliang Zhao, Knowledge Representation of Extended Production Rule, Journal of Xi'an Jiaotong University, Vol. 38, Xi'an, China, 2004, pp.587-590
[2] Liu Kangping, Zengzhi Li, etc., Study and implementation of alarm correlation and fault diagnosis expert system , Computer Engineering, Beijing, China, 2002,pp.11-68
[3] John F. Sowa, Conceptual Structure: Information processing in mind and machine, UK: Addison- Wesley Publishing Co., 1984
[4] John F. Sowa, Conceptual graphs for a database inference, IBM J. Res.& Dev., 1977, pp. 336-357
[5] Elaine Rich, Natural language interface, IEEE Computer Vol.17,1984, pp. 39-47
[6] Paola Velardi, etc., Conceptual graphs for the analysis and generation of sentences, IBM J. RES.&DEV. Vol. 32, 1988, pp. 251-267

Peiqi Liu was born in Xi'an, China. He had studied at the graduate level in Northwest university of China from 1990 to 1993, and had awarded a master's degrees in 1993. He is a Ph.D. candidate at the Institute of Computer Architecture and Networks, Xi'an Jiaotong University. Now, his research interests are data mining and artificial intelligence.

Zengzhi Li was born in 1940. He is a professor and doctoral advisor in Institute of Computer Architecture and Networks. His research interests include Computer Architecture and Networks.

Yinliang Zhao was born in 1954. He is an associate professor in the Institute of Computer Architecture and Networks, Xi'an Jiaotong University. His research interests include optimization algorithm and parallel algorithm