Researches of Sentence Similarity Computation Method Based on the Enhanced Petri Net

Fujin Chen
Computer Teaching and Experiment Center
Xi'an Jiaotong University
Xi'an, Shannxi, China
jin.tian@stu.xjtu.edu.cn

Wenge Chen
Computer Teaching and Experiment Center
Xi'an Jiaotong University
Xi'an, Shannxi, China
wgchen@ctec.xjtu.edu.cn

Abstract—**In the field of Natural Language Processing (NPL), sentence similarity computation has a wide application. The key of sentence similarity computation is to grasp the characteristics and the meaning of the sentence quickly and accurately. In this paper, we present a new and advanced sentence similarity computation model based on the enhanced Petri net, which is mainly for the Chinese sentence. The basic idea of this model is utilizing semantic properties to expand the Petri net and reconstruct Petri net to an opening structure. The deduction method of this model is equation of state, which make the judgment of sentence similarity closer to the process of human thought. Moreover, we have tested the sentence similarity computation methods and algorithms. It proves that this method covers most aspects of sentence similarity computation and has a superior performance during experiments.**

Keywords-Petri net; Semantic attributes; Sentence similarity; NLP;

I. Introduction

Sentence similarity computation is an essential technique and plays important roles in many areas of NPL. For example, in example-based machine translation, the similarity of sentences is mainly used to measure the degree of subsitutability in order to obtain the required translation script. In the information retrieval, the similarity of sentences reflects mainly the corresponding degree between text and query consulted by users; in automatic question-and-answer system, the sentence similarity reflects the match degree between the questions and answers. In Multi-document abstracts system, the sentence similarity reflects the fit degree of the partial theme information, removes redundant information and collects abstracts sentence. In text classification research, it can reflect the relevant degree between text and some class in given classification system by sentence similarity computation. Likewise, in the text similarity computation, it can also compute text similarity by sentence similarity computation. The study process of sentence similarity decides the development of other related fields. Sentence similarity itself is a very complex concept which has been widely discussed in semantics, philosophy and information theory. In different applications, the meaning of similarity is diverse. Therefore, people have been focused on this issue for a long time.

Sentence similarity computation is based on words similarity computation. The current used methods of words similarity computation includes the method based on the same terms, the method based on the semantic interdependence [1], the method of calculating the edit distance [2], the method based on key words, the method based on context framework, the method based on attribute, the method based on pattern [3], the method based on statistics [4] and the method of the integration of multi-level and multi-feature. However, each method has pros and cons.

Petri net is a modeling tool. It uses Place and Transition to simulate the system dynamic behavior and concurrent activities. Petri net can be used to establish state equation as well as other system behavior mathematical models. Petri net not only has a strict form definition and an intuitive graphical representation but also has rich means of the system description and system behavior analysis technology. Therefore, it can be well suited to describe the complicated, dynamic, asynchronous, distributed, parallel, and random components of uncertainty system. These features of Petri net are similar to human thinking process for sentence identification: Human can accurately determine the degree of similarity between the two sentences after one glance. This fact indicates that while reading, human extract attribute information and determine the identification at the same time. It shows a complicated thought process characterized by asynchronous, parallel, uncertainty and randomness. Based on this principle, we improve the basic Petri net, and increase the Petri net source node (generated Token) according to the characteristics of the sentence. More information is also added on each node.

II. Description of the structure and state space of Petri nets

Petri net is a graph model, as shown in Figure 1.

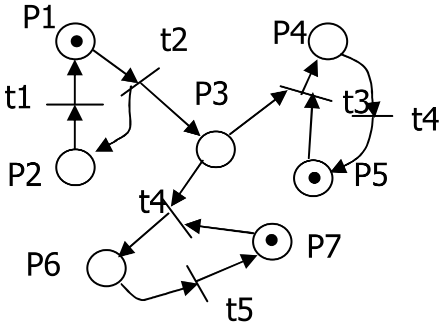

Figure 1. A simple Petri net

There are two elements in Petri net diagram: one is Place which is indicated with the circle "○", the other is Transition which is indicated with short-bar "-".

Circles (Place) and short-bars (Transition) are connected by the directed arc which can be classified as two kinds: one is from Transition to Place and the other is from Place to Transition. If there is an arc from node i to node j (from one Place to one Transition or from one Transition to one Place), then we define that i is the j`s input, while j is i's output. E.g. in Figure 1, the Place p1 is the output of the Transition t1, and the Place p2 and p3 is the output of the Transition t2.

During the running process of Petri net, it employs a token to mark a running process and when the process moves forward, token will move along towards the moving process of Petri nets. And then, the implementation process of Petri net is performed by using token to mark the Place where execute locates, as well as controlling the movement of token.

In the Petri net, the black dot (•) is used to indicate token which is marked in the executable Place (⊙) .

A Petri net is composed of two basic sets: the set of Place (P) and the set of Transition (T). To clarify relationship between the Transition set and the Place set, we define two functions to connect the Place and the Transition – the input function I and the output function O. So a Petri net can be defined as a quadruple: C = (P, T, I, O).

Incidence matrix and state equation are the major tools for Petri net analysis and modeling. The incidence matrix is a linear algebra expression of Petri net structure. Equation of state provides a basis for accessibility issues.

As the Petri net is mainly used to describe dynamic processes and its transformation is constrained by the token in net, the distribution of token in each Place can be characterized by the marks of Petri net. During the implementation processes of Petri net, the number and position of tokens is changing. The State can be described by a vector μ=(μ_1, μ_2,……, μ_n) , where μ_i (i=1, 2, ……, n) denotes the number of the token in the Place P_i. So a marked Petri net can be indicated by a five-tuple: M= (P, T, I, O, μ).

The states of a Petri net are defined by its marks. When a transition is ignited, the states are changed. The state of change ignited by a transition can be defined by the function δ . The result of state function is the next state mark.

During the implementation process of marked Petri net, it will produce two sequences:

One is mark sequence: (μ^0 , μ^1 , μ^2, ……) ;

The other is transition sequence: ($t_j(0)$, $t_j(1)$, $t_j(2)$, ……) ;

The relationship between the two sequences can be expressed by the following status function:

$\delta(\mu^k,\ \ t_j(k)) = \mu^{k-1}$, where, k=0, 1, 2……

III. The concept of sentence similarity computation

At present, there is yet a common definition for similarity, because it relates to language, sentence structure and other factors. Likewise, sentence similarity is also a highly subjective concept. It is difficult to get a uniform definition without a specific context.

From the perspective of information theory, Dekang Lin had given a unified, non-formal definition of similarity which is not related to the field of application [5]. He thinks that, the similarity between A and B, on the one hand, associates with their commonality. The more common they are, the higher their similarity would have. On the other hand, it associates with their difference. The greater difference they have, the lower their similarity would be. When A and B are identical, the similarity is maximum. However, the above definition is hard to apply during actual operation. Therefore, we should find a suitable definition according to the actual situation of the system.

For the reason stated above, we focus mainly on the overall semantics of the sentences and in turn study the sentence semantic similarity. Therefore, we do not concern much about the morphological similarity, the length of sentence similarity and the order of sentence similarity in this paper. It is also consistent with the features of the enhanced Petri net and brings out the advantages of the enhanced Petri net. As a result, we define in this paper the sentence similarity as: In the length similar case, two sentences similarity is the degree of overall semantic proximity.

However, it is necessary to quantify the sentence similarity computation. Literature [6] defines the sentence similarity as a real number in [0, 1]. 0 indicates that the two sentences are not similar at all and 1 indicates that the two sentences are completely similar. The greater value of two sentences similarity it is, the higher their similar degree would be. In this paper, this method is used to compute and evaluate quantificationally.

IV. The Enhanced Petri net and sentence similarity measure strategies

A. The description of the enhanced Petri net

Because the similarity computation of the sentence is not only the word form matching, but rather tries to grasp the meaning of sentences, in this paper, we expand the Place nodes and the structure of Petri net.

The sentence similarity computation can be attributed to the similarity computation of the words, so the Place nodes of Petri net are the words of the sentence.

Definition 1. For the word W in the location of the Place node, a Chinese thesaurus — TongYiCi CiLin (referred to as CiLin) [7] is used as the resources for the similarity computation between the words. Meanwhile, the weights of the words are marked. Finally, a set of words information P_w is established, in which the similarity value of the words are equal to the one of the Place node. This set is called the Place node set.

Definition 2. For the Place node set, the enhanced Petri net will have a token and trigger the next state, if and only if the test word $w \in P_w$ and the weight value of words is in a certain range of the weight values of the set (that will be discussed below). The entire process is called the flow condition judgment.

According to the definitions above, we make the expansion and description for the structure of Petri net as following:

1) The changing start threshold of the place node is the detection threshold of transfer function. The start process is the judgment process of the flow conditions.

2) The prototype of the enhanced Petri net is the Petri net of the order system. However, their difference is that each place node has a source node.

The structure of an enhanced Petri net as shown in Figure 2.

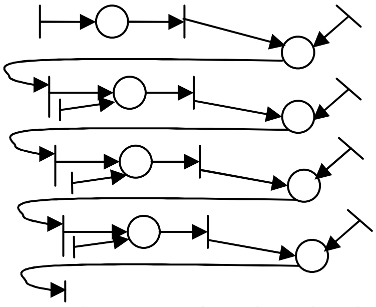

Figure 2. The structure of an enhanced Petri net

3) The weight value of the Place node in Petri net is a bounded real number which is mainly the weight value of the part of speech of the word. Whether to activate the Transition nodes depends on the input characteristics of each input line, connecting strength and whether the value of some transfer function (which we call the flow conditions) is bigger than the start threshold of the Transition node.

Definition 3. The form of the enhanced Petri net is defined as an eight-tuple:

Enhanced_PetriNet = {P, T, I, O, μ , W, S, O},

Where

P is a finite set of Place nodes with a Place node set.

T is a set of transition nodes which can reflect the interdependent of a sentence and be equivalent to a weighted qualitative mapping for a sentence.

I is a set of input function.

O is a set of output function.

μ is a marked vector.

W is the weight of the part of speech of the word.

S is the access function of the node.

O is a function of the flow condition computation.

B. The Sentence similarity measure strategies

In the enhanced Petri net, for the generation of the Place node set, we utilize the basic idea of the "TongYiCi CiLin"in this paper. The principle is that using the semantic encoding of each word provided by CiLin to compute the semantic distance between two words. This literature divides the semantic of the word into 5 level and describes a semantic classification system which is from up to down, from a general concept to the specific word meaning. And the words are collected and categorized in this system. The corresponding to the classification system is a word semantic coding system. It is described as following:

<Semantic encoding>
::=< 1level><2level><3level><4level><5level>
<1level>:: =<uppercase letters>
<2level>:: =<lowercase letters>
<3level>:: =<number><number>
<4level>:: =<uppercase letters>
<5level>:: =<number><number>

This can compute the semantic distance between two words. For any two words A and B, the semantic distance can be compute by the following formula [8]:

$$Dist(A,B) = \min_{a \in P, b \in Q} dist(a,b) \qquad (1)$$

Where

P, Q are the sets of semantic for words A and B respectively. The distance between semantic *a* and *b* is:

$$dist(a,b) = 2 \times (7-n) \qquad (2)$$

Where

n indicates that the semantics code between the two words is different from the beginning of the first n layers. If all are the same, the semantic distance is 0.

Based on the above-mentioned method, we compute the semantic distance for the words in the Place nodes. According to the value of the threshold, the relative information of the words in the test sentence should be added to the Place node set P_w dynamically.

In the enhanced Petri net, all the elements of the place node set can actually be seen as a mapping table, which contains the names of the similar words, the corresponding codes in the extended version of "CiLin", the semantic distance (words similarity), the weight value of the words and so on. Its purpose is to generate token (the flow conditions) easily after the segmentation of the test sentence.

In order to understand the sentence similarity computation formula, we firstly introduce the computation process of the test sentence in the enhanced Petri net which is generated from "standard" sentence dynamically.

1) Given the "standard" sentence and generate a sequence enhanced Petri net dynamically. At this time, each Place node set has only one element which is the word information of the "standard" sentence.

2) Do word segmentation for the test sentence.

3) Read the first word of the test sentence, compute the semantic distance and mark the weight at each source node. According to the flow conditions, whether or not to generate token can be determined.

a) If a token being generated, the token staies in the presence of the node and according to the result of flow condition function to determine whether or not to flow to the next node and compute next word. If matching the flow condition, the node will generate a flow weight and record it in a temporary variable in order to reflect the continuity of the sentence. Otherwise, it also generates a flow weight to involve in the similarity computation.

b) If a token not being generated, the next word will be computed.

4) Compute all the remaining words of the test sentence orderly followed by the same computation method mentioned above until the end of the sentence.

Thus, the formula of the sentence similarity computation takes the form:

$$Sim(S_1,S_2)=\alpha\times\frac{Sum_{Token}}{Sum_p}+\beta\times(\sum W_i^w\times D_i/\sum W_i^w)+\gamma\times(\frac{\sum C_j}{j})$$

(3)

Where

$\mathrm{Sum}_{\mathrm{token}}$ and $\mathrm{Sum}_{\mathrm{p}}$ are the total numbers of the token and the Place node respectively.

W_i^w is the weight of part of speech of word and D_i is the semantic distance between two words.

C_j is the flow weight in order to reflect the continuity of the sentence.

α 、 β 、 γ are dynamic constants which are determined by the number of token and the length of the two sentences, and $\alpha+\beta+\gamma$=1. Obviously, $\mathrm{Sim}(\mathrm{S}_1,\mathrm{S}_2)\in[0,1]$.

V. THE ALGORITHM OF THE ENHANCED PETRI NET MODEL

According to the process of the test sentence computation in the enhanced Petri net, the principle of the sentence similarity computation based on enhanced Petri net can be described as follows: firstly, select a sentence as a standard one, establish its enhanced Petri net. Then, compute the test sentence in the enhanced Petri net through the flow conditions and the state transition equation until all the words in the sentence have finished calculating. Finally, compute the similarity based on the token and the situation of running in the net.

The algorithm is described in the figure 3.

In the running process of the algorithm, we may encounter such a situation: the marks of token and the target marks are not completely consistent. In order to solve this problem, a threshold can be defined to compute in the formula (3).

Algorithm: the sentence similarity computing algorithm based on the enhanced Petri net.

Input: two sentences S_1 and S_2.

Output: the similarity between S_1 and S_2.

Step1: compute the length of S_1 and S_2 then get L_1 and L_2, compare the L_1 and L_2. If the result beyond a certain threshold, return; otherwise, go to next step.

Step2: do the word segmentation for S_1 and generate an enhanced Petri net. Mark the weight of part of speech of word for the elements in the Place node set. Carry out the word segmentation to S_2 and run to step 3 from the first word of S_2.

Step3: visit the start nodes of the enhanced Petri net in turn; compute the semantic distance between the two words. And judge through the threshold in order to generate the marks of the semantic distance. If the word is not in the scope of similarity, goes to the next source node. Otherwise, the word is added to the Place node set. Meanwhile, the node generates a token and computes the flow condition.

Step4: According to the flow conditions, decide whether or not to move in the net. If it can move in the net, go to the next Place node. Otherwise, go to step 3.

Step5: in the next Place node, compute the distance of the next word in the S_2. Repeat steps 3 and 4 until no word in S_2.

Step6: According to the formula (3), compute the similarity and output the result.

Figure 3. The sentence similarity computing algorithm based on enhanced Petri net

VI. THE EXPERIMENTAL RESULTS AND ANALYSIS

Because there is no standard test corpus of sentence similarity, the homework of the students is used as the original corpus in this paper. The question-and-answer parts of the students` homework are sampled and formatted a 400 sentences test set. These sentences are divided into two parts: one is 300 Chinese sentences and the other is 100 English sentences.

The 200 Chinese sentences are randomly selected and constitute a noise set. The remaining 100 ones are hand-selected, divided into 20 groups of 5 sentences which are similar. These 20 groups constitute a standard set. In the noise set, some sentences are related to the standard sentences, but degree of the correlation is low. Therefore, the test set is more comprehensive.

The 100 English sentences are also sampled randomly in order to test the sensitivity of the model for the English sentence. This is an exploratory testing.

Two experiments are performed in this paper. One is the test of Chinese sentences. The method and result are as follows:

In the standard set of 20 groups, one sentence is randomly selected from each group and computed the similarity with the other 299 sentences using the algorithm proposed in this paper. Then, the values of similarity are sorted from big to small. Finally, the results are output. If the selected sentence and the other sentences belonged to the same group with the selected sentence are in the front of the sorted queue, it indicates that the similarity computation is successful. In this paper, this experiment was performed 25 times and compared with the edit-distance algorithm. The experiment makes the former two and the former five sentences in the sorted sequence as criteria for judging sentences. Table 1 gives the results of the comparison.

TABLE I. THE RESULT OF THE CHINESE SENTENCES

The methods	**The criteria**	**Times of test**	**Times of right**	**Correct rate**
The edit-distance algorithm	Top 5	25	16	64%
The enhenced petri net model	Top 5	25	20	80%
The edit-distance algorithm	Top 2	25	18	72%
The enhenced petri net model	Top 2	25	22	88%

As can be seen from the experimental results, the sentence similarity computing algorithm based on enhanced Petri net can improve the correct rate and accuracy.

The other one is the test of English sentences for exploratory testing. The method and result are as follows:

Firstly, select a sentence as "standard" sentence from the 100 English sentences randomly. And then, compute the similarity with the other 99 sentences. Finally, sort the similarity to judge it is correct or not. Table 2 is the result of 30 times test.

TABLE II. THE RESULT OF THE ENGLISH SENTENCES

The methods	**Times of test**	**Times of right**	**Correct rate**
The enhenced petri net model	30	15	50%

As showed in the table 2, the correct rate is low. That is because that for the English sentences, it will not generate the Place node set and be only more sensitive to the morphology. Therefore, this model is no efficiency for the English sentence similarity computation.

VII. ALGORITHM EVALUATION

The algorithm proposed in this paper follows the natural artificial sentences compare and uses the correlation between the Petri net structure and the thinking of the artificial sentences compare. The result is improving the correct rate.

Compared with other models, the enhanced Petri net covers most aspects of the similarity computation. Moreover, the result of experiment is shown that it improves accuracy of the sentence semantic similarity.

1) The Place node set is added to the enhenced petri net, so the judgement of the similar word in the sentence will not be missed.

2) The enhanced Petri net considers the proportion of consecutive words in the sentence and improves the accuracy of similarity computation for the entire sentence.

3) Because the enhanced Petri net extends the structure of net through adding the source nodes, the sentence word order similarity computation is not need to consider. Therefore, the computation is reduced.

However, the similarity computation situation of the English sentence is not very ideal. This is also what the model needs further improvement and research.

VIII. CONCLUSIONS

In this paper, the process of artificial sentences compare is taken as the starting point and the order Petri net which is similar to this process is used as a tool. Moreover, the structure and node information are enhanced on the basis of the order Petri net, which can make it more adapted to the judgment and sentence similarity computation. This model can effectively portray the sentence structure. Meanwhile, it is enhanced through the synonym in the semantic meaning. Thus it increases the computation precision overall.

Through the preliminary trial, this method obtains the satisfactory result for the Chinese sentences. However, this model can not judge the similar word in English sentence, so it lacks the versatility. Moreover, the speed of computation is not very ideal. That is just what we will do in the future.

IX. REFERENCES

[1] Bing LI, Ting Liu, Bing Qin, Sheng LI. Chinese Sentence Similarity Computing Based on Semantic Dependency Relationship Analysis[J]. Application Research of Computers, 2003. Vol.20 No.12.

[2] Jianzhou Liu, Tingting He, DOnghong Ji, Xiaohua Liu. Chinese terminology automatic extraction based on open style corpus [C].20thInternational conference on computer processing of oriental language, Shenyang.

[3] Sichun Yang, Jiehua Cheng, Jiajun Chen, Qixiang Wang. One kind method of Chinese sentence similarity computation based on pattern. Microcomputer & its Applications, 2001.8.

[4] Xiaopeng Tao,Shuigeng Zhou. Chinese word segmentation without auxiliary data[C].20th International conference on computer processing of oriental languages,Shenyang, 2003.

[5] Dekang Lin and Patrick Pantel.2001.DIRT-Discovery of Inference Rules from Text. Journal of Natural Language Engineering. Fall-Winter 2001.

[6] Fang Zhou. Study and Application on Chinese Sentence Similarity Computation [D]. Henan University, 2005, 5.

[7] Jiahen Zheng, Yili Qian, Jing Li。Research of Reasoning Method of Two-character Words Word-sense Combination [J]. Journal of Chinese Information Processing. 2001.Vol.15 No.6

[8] Wanxiang Che , Ting Liu, Bing Qin, Sheng Li. Similar Chinese Sentence Retrieval based on Improved Edit-Distance [J]. Chinese High Technology Letters. 2004.7,15-19.