Semantic Computation in a Chinese Question-Answering System*

Li Sujian[1](李素建), Zhang Jian[2](张健), Huang Xiong[2](黄雄), Bai Shuo[2](白硕), LiuQun[2](刘群)

Institute of Computational Linguistics, Peking University, Beijing 100871, P.R.China
[2]Software Department, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, P.R.China
E-mail: lisujian@pku.edu.cn

Abstract This paper introduces semantic computation into our Chinese Question-Answering system. Based on two kinds of language resources *hownet* and *Cilin*, we present an approach to compute the similarity and relevancy between words. Using these results, we can calculate the relevancy between two sentences and then get the optimal answer for the query in the system. The calculation adopts quantitative methods and can be incorporated into QA systems easily, avoiding some difficulties in conventional NLP problems. We finally present the experiment to show that the results are satisfying.

Keywords similarity, relevancy, hownet, Question Answering, Natural Language Processing

摘要 本文介绍了一种实现语义计算并把它结合到中文问题回答(QA)系统中的方法。基于两种语言资源hownet和《同义词词林》建立一个语义关系的计算模型，从中得到词语的近似度和关联度；然后由词语间的定量计算结果，进一步计算出语句之间的关联度，从而最终确定系统查询的最优结果。由于把定量的计算方法结合到QA系统中，避免了QA系统处理自然语言的难点。实验结果证明该方法是有效的。

关键字 相似度，关联度，知网，问题回答，自然语言处理

1 Introduction

With the explosion of information available on Internet, Question-Answering system can help us to find what closely matches users' needs. Since both questions and answers are mostly expressed in natural languages, Q/A methodologies have to incorporate NLP (Natural Language Processing) techniques, including syntactic and semantic computation. Due to the encouragement of the Text Retrieval Conference (TREC) and the Message Understanding Conferences (MUCs), some QA systems have achieved good performance [1]. However, these systems mainly aim at English. In this paper, based on the characteristics and some language resources, we build a Chinese Question Answering system through the computation of semantic similarity and relevancy.

2 Overview of Language Resources

Hownet is a free Chinese-English bilingual resource which is released recently on Internet [2, 3, 4]. It is a knowledge base describing relations between concepts and relations between the attributes of concepts. In our Chinese QA system we mainly use the knowledge base, which include 66,681 concepts. Every word sense is represented by the combination of several sememes. A sememe is a basic semantic unit that is indivisible in *Hownet*. According to the view of ontology, about 1500 sememes are extracted to compose an elementary set which is the basis of the Chinese glossary, as over 100 kinds of chemical elements constitute all the substances in nature. We describe several definitions in *Hownet* as follows:

*本项目研究工作受到国家重点基础研究计划（973）资助，项目编号是G1998030510和G1998030507-4。

$SS = \{s_1, s_2, \ldots, s_n\}$, n=1541

$WS = \{c_1, c_2, \cdots, c_m\}$, m = 66,681

$REL = \{*, @, ?, !, \sim, \#, \$, \%, \wedge, \&, NULL\}$

$c_i \Rightarrow r_{i1}s_{i1}, r_{i2}s_{i2}, \cdots, r_{ik}s_{ik}$, $r_{it} \in REL$, $s_{it} \in SS (1 < t < k)$

where SS represents the set of the sememes which includes 1,541 elements; WS represents the set of the word senses in *Hownet* whose size is 66,681; REL is the set which describe relations between a concept and a sememe or relations between sememes. For every word sense c_i, that is a concept, its definition is composed by k items, each of which includes a relation symbol in REL and a sememe in SS.

In our system, another language resource available is *Chinese Thesaurus «Cilin»* [5], which conducts semantic classification for Chinese words. It comprises 12 major categories, 94 medium categories, and 1428 minor categories. And the minor categories can be further divided into synsets according to their meanings. Every synset includes several words with the same or similar meanings. This hierarchical classification has embodied synonymous relation and hyponym relation and provided convenience for the expansion and semantic computation of word senses. We formalise several definitions as follows:

$WS' = \{c_1, c_2, \cdots, c_{m'}\}$, $m' = 61{,}125$

$SC = \{sc_1, sc_2, \cdots, sc_p\}$, $p = 11{,}832$

Where WS' represents the set of word senses in *Cilin*, whose size is 61,125, and SC represents the set of synsets whose size is 11,832.

The two language resources introduced above are a great help to our computation in semantic similarity and relevancy of two Chinese words.

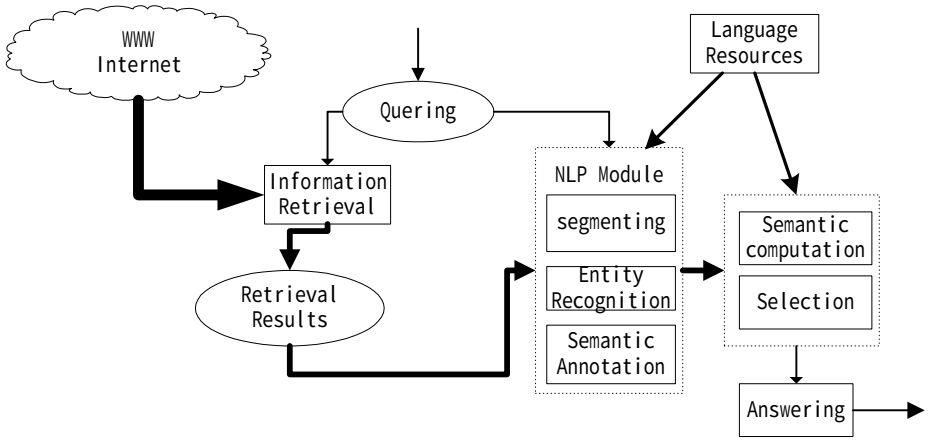

Figure 1.System Structure

3 System Description

At present, the processing mechanism of most QA systems are based on sentences [6], and at the same time, it absorbs the techniques of information retrieval, information extraction and natural language processing [7]. As shown in Figure 1, for the large quantity of information from internet, keywords and mood words such as those extracted from queries are inputted to the process of Information Retrieval to reduce the scope of searching, and at the same time the sentences whose mode or negative/positive mood is not consistent with the query sentence are also filtered out. Then the results obtained and the question needed to query are submitted simultaneously to the modules involved in natural language processing. These modules include segmentation module, entity recognition module, and semantic annotation module. After the processing of these modules, we can get sentences with semantic annotation which can enter the module of semantic computation (SC). SC module gets the relevancies between sentence pairs. Then we select the sentence pairs with the largest value of relevancy.

In Figure 1, the thicker the line, the more information it represents. The language resources include *Hownet* and *Cilin*. According to the characteristics of the Chinese language, we must conduct segmentation for sentences. At the same time or after segmentation, named entity should also be picked out and semantic annotation is conducted for segmented words and named entity. The three natural language

modules don't have explicit boundary. Based on the semantic information collected in the three NLP modules, we conduct semantic computation between query and relevant sentences. The main function of the semantic computation module is to get the relevancy value between sentence pairs and sort them. This paper mainly discusses the techniques concerning how to conduct semantic computation.

4 Semantic Computation

Semantic computation is the kernel of our system, which is conducted in three steps. The first step is to conduct the computation of the similarity and associativity between sememes. Second, similarity and relevancy between words are computed; and in the last step, based on the results of the two steps above, we can calculate the relevancy between sentences and get the sentence pairs with the maximal value of relevancy.

4.1 Similarity and Associativity between Sememes

In *Hownet*, the relations among sememes are built through several feature files. The sememes in one feature file construct a tree structure. As shown in Figure 2, this is a sample structure of nodes that belong to the feature files. Relations between sememes can be obtained from these hierarchical trees and based on these relations we can compute similarity and associativity between sememes within this mechanism. Every node is called a *main sememe*. Every main sememe is followed by some sememes included in the square brackets, which we can see as its explanation called as *explanatory sememes*. Every explanatory sememe is ususally preceded by a symbol which describe its relation with the main sememe. Both main sememes and their explanatory sememes have hyponyms and hypernyms, thus we can get association between sememes in different feature files. It is followed that all the sememes in *hownet* construct a network structure.

In Figure 2, the relation between a main sememe and its hypernym or hyponym is called as *Vertical Relation*, we measure sememes with Vertical relations with similarity; other relations which span different feature structure are called *Horizontal Relation* which can be measured by associativity between sememes.

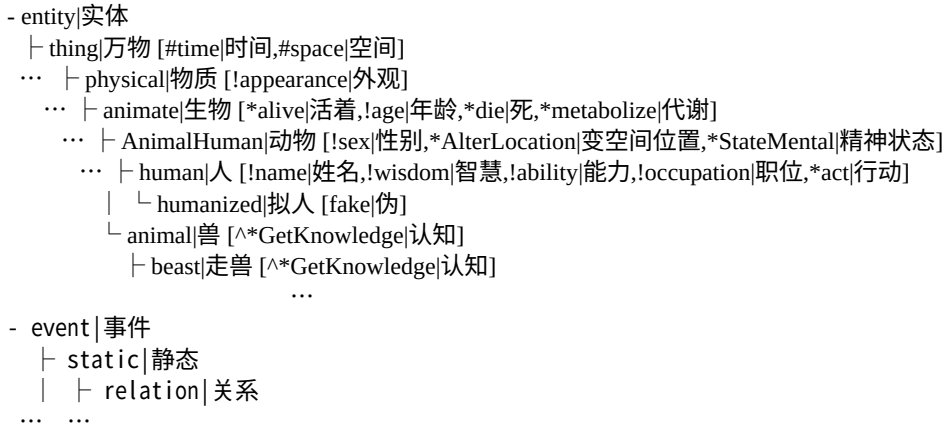

Figure 2. A Sample Tree Structure of Feature Sememes

For two sememes in the tree structure of Figure 2, there exist three possible relations:

1. When the two sememes are in different trees, the similarity will be 0;
2. the two sememes at least have one common ancestral node, but they are in different branches of the ancestral node;
3. one sememe is the ancestral node of the other one;

Then, we compute the similarity between sememes as equations in (1):

$$sim(s_1,s_2)=\begin{cases}\alpha/dist(s_1,s_2) & t(s_1)=t(s_2)\\ 0, & t(s_1)\neq t(s_2)\end{cases} \quad s_1,s_2\in SS \tag{1}$$

where, for any two sememes s_1,s_2 in the sememe set SS, sim(s_1,s_2) represents similarity between s_1 and s_2. $t(s_1)=t(s_2)$ represents that the two semems

are in one tree structure and their similarity is inversely proportional to their distance.

Like the structure of Figure 2, the explanatory sememes build a bridge for two sememes in different trees. For example, there should exist some relation between the sememes 'animate|生物' and 'alive|活着' which don't have any similarity at all. Here we introduce a new measure – associativity – to represent those relations spanning different trees. In doing so, the tree structure becomes a net structure. In order to compute associativities, we need to expand the current sememe in two directions. One is to expand to the hypernyms of explantory sememe which is called Horizontal Associative Expansion (HAE), the other expansion is to the explantory sememes of the hypernyms which is called Verticle Associative Expansion (VAE). We compute associativities according to the equations in (2):

$$\begin{cases} ext(s_j)=\{s_i \mid REL(s_j,s_i)\} \\ Asso(s_1,s_2)=\sum_{s_1\in ext(s_1)} w_i sim(s_i,s_2)+\sum_{s_j\in ext(s_2)} w_j sim(s_1,s_j) \end{cases} \quad (2)$$

Where ext(s_j) is an extension set of the sememe s_i which includes HAE and VAE. We endow a weight to every relation in REL which describes how this kind of relation has an influence on the associativity. In computing the associativity between s_1 and s_2, the first part represents the associativity between s_2 and extensive set of s_1; and the second part is for s_1 and extensive set of s_2.

4.2 Similarity and Relevancy between Words

In section 2 we have introduced two kinds of language resources. For *Hownet* it is easier to construct a net structure for sememes and then to get their similarities and associativities. Because every word sense is composed of sememes, it's difficult for *hownet* to expand the similar or same word senses. Now we utilize the second language resource – *Cilin* – to make expansion of conceptions. As in Figure 3, it is a sample structure of conceptions in *Cilin*. Every node is a semantic class. The nearer to the root node, the more abstract the conception that the node represents. Unlike *Hownet*, not every node in the structure represents a concrete word sense, and only the leaf node is a collection of Chinese word with the same or similar sense.

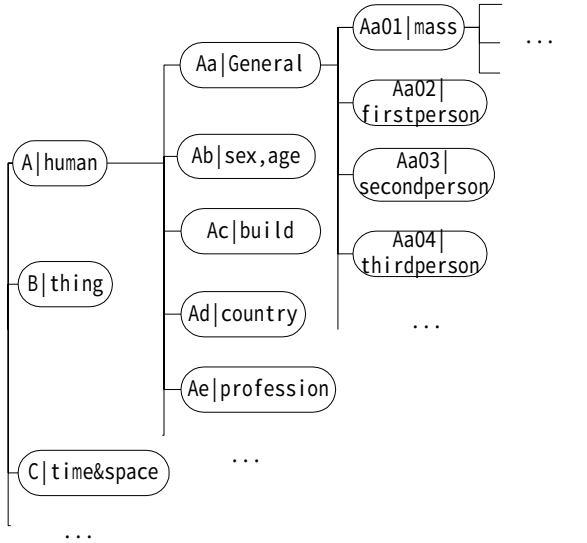

Figure 3.A Sample Struture in *Cilin*

Similar to the computation of sememes, we have the following equation:

$$sim(c_1,c_2)=\begin{cases} \alpha / dist(c_1,c_2) & t'(c_1)=t'(c_2) \\ 0, & t'(c_1)\neq t'(c_2) \end{cases} \quad (3)$$
$$c_1,c_2\in WS'$$

Where c_1 and c_2 are any two word senses in *Cilin*. $t'(c_1)=t'(c_2)$ represents that the two conceptions belong to some same semantic class and their similarity is inversely proportional to their distance.

Here we adopt a measure – relevancy – to represent the associative relation between word senses. The goal of computing the similarity and associativity between sememes is to get the relevancy of word senses according to the equations in (4):

$$\begin{cases} Rele(c_1,c_2)=Rele(def(c_1),def(c_2)) \\ Rele(def(c_1),def(c_2))\approx \sum_{s_i\in def(c_1)} \max_{s_j\in def(c_2)} Rele(s_i,s_j) \\ def(c)=\{s_i \mid REL(c,s_i)\} \\ Rele(s_i,s_j)=w_s sim(s_i,s_j)+w_a asso(s_i,s_j) \end{cases} \quad (4)$$

Where Rele(c_1,c_2) is the relevancy between two word senses c_1 and c_2, and def(c) is a set of explanatory sememes for the word sense c. w_s and w_a are the weights of similarity and associativity between sememes respectively, and we can get a relevancy between sememes Rele(s_i,s_j). To get the relevancy of two sets of sememes, we pick out the possible sememe pairs with maximal value and sum them up.

4.3 Relevancy between Sentences

We assume that the filtered sentences s_1 and s_2 have been segmented, resolved anaphorically and annotated semantically. Then s_1 and s_2 can be

regarded as two sequences of m and n keywords: w_{11} w_{12} …w_{1m} and w_{21} w_{22} …w_{2n}

To compute the relevancy of a sentence pair, we use the similarity and relevancy of word pairs. We select the word pairs that contribute most to the relevancy of the sentence pair. The word pairs are connected with lines as in the figure 4.

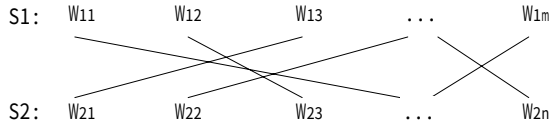

Figure 4. Word Pairs in Two Sentences

We use a dynamic programming algorithm to get the relevancy of a sentence pair as the following equation:

$$\begin{cases} \mathrm{Re}le(S_1, S_2) = M_{m,n} \\ 1/d_{ij} = \alpha \mathrm{Re}le(w_{1i}, w_{2j}) + \beta Sim(w_{1i}, w_{2j}) \\ M_{0,j} = M_{i,0} = 0 \\ M_{1,1} = 1/d_{1,1} \\ M_{i,j} = \max\limits_{1 \le k \le n} \{1/d_{ik} + M_{i-1,j}\} \end{cases} \quad (5)$$

Where α and β are weights that represent the degree that the similarity and relevancy of words contribute to the relevancy of the sentence pair. d_{ij} is the semantic distance between the i^{th} word in the first sentence and the j^{th} word in the second sentence. According to the recursive equation, we can finally get the value of $M_{m,n}$ which represents the relevancy between the two sentences s_1 and s_2.

After we get the relevancy of all sentence pairs, we compare their values. The larger the value of relevancy, the more relevant the two sentences. We get the sentence as the answer of the query that has the largest value of relevancy.

5 Experiments and Discussion

The semantic computation contains three steps and every step makes use of the computation of last step. The three steps conform to the characteristics of the Chinese language: from morphemes to words to phrases.

We did experiments on every step above, and the results are satisfying, reflecting the correlation between elements in every step. Here are some examples: Table 1 illustrates the similarity and associativity of some example sememe pairs, and the examples in Table 2 demonstrate the similarity and relevancy of some word pairs.

Table 1: example of sememe pairs with their similarity and associativity.

Sememe1	Sememe2	Sim	Sememe1	Sememe2	Asso
Discuss| 商讨	Debate| 辩论	0.80	material| 材料	Consume| 摄取	0.35
TalkNonsense| 瞎说	Debate| 辩论	0.32	Human|人	Act|行动	0.80
Spread|撒	Throw| 扔	0.40	Produce| 制造	Software| 软件	0.40
cook|吐出	Throw| 扔	0.533	Compile| 编辑	Software| 软件	0.80
Dream| 做梦	Cool| 制冷	0.114	Planting| 栽植	FlowerGrass| 花草	0.80
Mental| 精神	Machine| 机器	0.267	CauseToLive| 使活	FlowerGrass| 花草	0.267

Table 2: example of word pairs with their similarity and relevancy

Word1	Word2	Sim	Word1	Word2	Rele
摇动(shake)	晃动(rock)	0.90	致意(give one's regards)	恰巧(by chance)	0.0
摇动(shake)	移动(move)	0.64	实行 (implement)	笑(smile)	0.267
病人(patient)	医院(hospital)	0.00	病人(patient)	医院 (hospital)	51.995
医生(doctor)	病人(patient)	0.410	医生 (physician)	生病 (be ill)	50.107
医生(doctor)	护士(nurse)	0.64	勤劳(diligent)	富裕 (wealthy)	51.307
揣测(guess)	了解(know)	0.512	贫穷(poor)	懒惰 (lazy)	51.657
揣测(guess)	推想(suppose)	0.90	勤劳 (diligence)	贫穷 (poor)	27.457
反常(abnormal)	奇怪(strange)	0.64	写(write)	作者 (author)	33.000

In the two tables Table 1 and Table 2, due to the difference of the weights, the quantitative levels of different measure are different and we should compare vertically.

We use the IR module to retrieve 20 relevant documents and extract 50 sentences on average. So for every query sentence there are about 1,000 sentences. We calculate and sort these 1,000 relevancy values between the retrieved sentences and the query sentence, and finally get one or more sentences with the largest value as answers. We illustrate 5 queries to show the effect of our Q-A system. 93 people were selected to evaluate whether these answers are reasonable. This evaluation is simplified with the following standard : if one person thinks the answer reasonable, the score is incremented by 1; otherwise, the score remains unchangeable. Then the maximal score that one answer can get is 93. In Table 3, the first column represents the No. of one query sentence; the second is the sum of the retrieved sentences; the third column represents the largest relevancy which we get by semantic computation; and the last column records the score of one answer.

From Table 3, we can see that the answers are reasonable for most people. The largest values of relevancy for every query are very different, which is because our computation is dependent on the length and words of one sentence.

Table 3. Results of several queries in Q-A

Query No.	Relevant sentences	Largest Relevancy	score
1st	1,029	205.127	89
2nd	986	232.411	93
3rd	997	334.826	92
4th	1003	602.133	93
5th	1002	603.329	91

6 Conclusions

This paper mainly introduces the application of semantic computation in our Question-Answering system. We can compute the similarity and relevancy between words, and get the optimal result by calculating the relevancy between sentences. Our method conforms to the characteristics of the Chinese language, combining semantic information with the computation in three levels and avoiding a lot of complexities in language processing. At the same time, the results of the intermediate process, such as similarity and associativity between sememes, and similarity and relevancy between word senses, are also very helpful in other research fields, e.g. polysemous disambiguation clustering, and bilingual alignment, to name a few.

Acknowledgements

The authors would like to thank Dr. Song Lu, and

Ms. Yan Liang for their help on this work, and also anonymous reviewers for valuable comments on this paper.

References

[1] E. Voorhees, 1999, *The TREC-8 Question Answering Track Report*, National Institute of Standards and Technology, page 77

[2] Dong Zhendong, 1999, *Hownet*, http://www.keenage.com

[3] Zhou Qiang, Feng Songyan, 2000, *Building a relation network representation for how-net*, Proceedings of 2000 International Conference on Multilingual Information Processing, Urumqi, China, pp.139-145.

[4] Gan K. W., Wong P. W, 2000, *Annotating information structures in Chinese texts using HowNet*. Second Chinese Language Processing Workshop, Hong Kong, China, pp. 85-92.

[5] Mei Jiaju, 1983, *Chinese thesaurus «Tongyici Cilin»*, Shanghai thesaurus Press.

[6] B. Katz, 1997, From Sentence Processing to Information Access on the World Wide Web, AAAI Spring Symposium on Natural Language Processing for the World Wide Web, Stanford University, Stanford CA.

[7] Rohini Srihari, Wei Li., 1999. *Information Extraction Supported Question Answering.* (Cymfony Inc.) Proceedings of the 8th Text Retrieval Conference (TREC-8). National Institute of Standards and Technology, Gaithersburg MD.

□

Biographies

Li Sujian received her B.S. degree and M.S. degree in computer science from Shandong University of Technology in 1996 and in 1999 respectively. She is now a candidate doctor and pursues her PH.D degree of computer science at the Institute of Computing Techonolgy, Chinese Academy of Sciences. Her current research interests include machine translation, natural language processing, knowledge discovery, and machine learning.

Zhang Jian received his B.S. degree in physical oceanography from Ocean University of Qingdao, China in 1998, and his M.S degree in computer science from Institute of Computing Technology, Chinese Academy of Sciences in 2001. Now he is a PH.D. student at School of Computer Science, Carnegie Mellon University. His research interests include machine learning, information retrieval and data mining.

Huang Xiong received his B.S. and M.S. degrees from Peking University in 1992 and 1995, respectively. He received his Ph.D. degree from Beijng University of Aeronautics and Astronautics in 1999. From May of 1999 to May of 2001 he conducted research in Institute of Computing Technology as a post-doctor. His major interests lie in analysis and design of combinatorial algorithms, computational complexity, Web information retrieval and Web application development.

Bai Shuo received his M.S. and Ph.D degrees of Computer Science from Peking University respectively in 1987 and 1990. Then he conducted research as a post-doctor in Mathematics Department of Peking University. He has published more than 60 papers in refereed journals and conferences. His research interests are on Computational Linguistics, Natural Language Processing and Network Security.

Liu Qun is a associate professor in Institute of Computing Technology, Chinese Academy of Sciences. He received his B.S. degree in computer science from University of Science and Technology of China in 1989 and his M.S. degree in computer science from Institute of Computing Technology, Chinese Academy of Sciences in 1992. He pursues his on-job doctorate of computer science in Peking University from 1999 till now. His research insterests include Machine Translation, Natural Language Processing and Chinese Information Processing.