The Design and Implementation of an Artificially Intelligent Personal Assistant

Chun-Hsiang Hsieh, and Daniel J. Buehrer

***Abstract*—**We describe the design of an Artificially Intelligent Personal Assistant (AIPA) system that uses concepts from the artificial neural networks, a triple-based knowledge representation that is used for both an ontology and its binary relationships, fast access via hash functions, and Class Algebra as the theoretical basis. Instead of providing the computers with the ability to create intelligence, the purpose of the research and implementation herein is to apply a QA system for actual use by users who may converse via the Chinese language with the computer, which then will understand the semantics and respond to the users with corresponding data.

***Keywords*—**Machine Learning, Semantic Network, Ontology, QA system.

I. Introduction

THIS research and implementation of an Artificially Intelligent Personal Assistant (AIPA) uses the following methods. First, the lexicon provided by the Chinese word segmentation system CKIP (Chinese Knowledge and Information Processing) [1] of Academia Sinica, Taiwan, was used as the basic lexicon for AIPA. There are more than 70,000 common vocabulary words and phrases in the CKIP lexicon, of which there is a corresponding POS tagging for each word and phrase. Next, we did word segmentation by using the MM (Maximum Matching) method [2], and constructed an AIPA semantic network by the PDFT (phrase driven formal translation) method suggested herein, and built a Neural Token List and processed the knowledge recalls by using the HNN (Hopfield Neural Network) relational module [3].

A. Word Segmentation

When doing research and implementation for a Chinese QA system [4], the first challenge is the word segmentation for Chinese sentences. Unlike English sentences, with spaces between words as an assistance to distinguish phrases, without the assistance of a word segmentation system, there is no way to correctly analyze the relationships within a Chinese sentence. Imprecise processing of word segmentation may mislead the subsequent grammar analysis and sentence processing; for example: the correct segmentation for "阿里山在台灣 (Mt. Ali is in Taiwan)" should be "阿里山 (Mt. Ali)／在 (in)／台灣 (Taiwan), while the incorrect one is "阿 (Ah)／里山在台 (Mt. Li in Tai)／灣 (wan). Apparently, the correctness of the segmentation may affect the subsequent sentence analysis and semantic interpretation.

Chun-Hsiang Hsieh is with National Chung Cheng University, R.O.C.(Taiwan)(corresponding phone:+886922-710219 ; e-mail: hsiehch@cs.ccu.edu.tw).

Daniel J. Buehrer, was with National Chung Cheng University, R.O.C.(Taiwan) (e-mail: dan@cs.ccu.edu.tw).

B. Part-of-Speech Tagging

Besides segmentation processing, POS (Part-of-Speech) Tagging [5] is also important. The segmented vocabulary phrases play crucial roles in the subsequent grammar parsing, because each vocabulary has its own POS Tagging (e.g. U.S. (noun)). Different POS Tagging represents different parsing conversion, and determines the correctness of the sentence parsing.

C. Unknown Words

The learning of unknown words is another important subject. Although a system may understand almost all common vocabulary phrases with the assistance of an existing lexicon, in a sentence there are often words not included in the lexicon, such as new phrases or names, which are the so-called "unknown words". Since the system cannot recognize these unknown words in time, they may affect the correctness of sentence and semantic identification.

D. Semantic Network

The building of a semantic network may associate the objects or categories between related knowledge. For example, “A pigeon is a bird,” entails an “is a” association between pigeon and bird, and a “kind of” association between bird and animal. With associated reasoning by a semantic network, we can tell that a pigeon is "a kind of" animal. In addition, the description of other attributes of knowledge are also necessary, such as the appearance or shape of an object of a category (has shape), or that the property belongs to a certain object or a certain category (has property). For example, birds can fly (i.e. has a "has property" association to “can fly”).

II. AIPA architecture

There are two major parts in the AIPA program. One is the Knowledge Base, which contains words, the QSSDB/QASDB (Question Sample/Answer Sentence Databases) and SRADB (Sentence Relation/Action Database), the ontology, and the NTL (Neural Token List). The other part concerns association and learning. First, a sentence input into the AIPA is analyzed by the Chinese NLP preprocessing, and the analyzed data are sent to the "Sentence Parsing Analysis Sentence Pattern

Module", where the analysis of the sentence patterns is passed into the next PDFT (Phrase Driven Formal Translation) module, which analyzes the semantics. Finally, the HNN associates and recalls the ontology and NTL knowledge, and responds to the users with the inquired data, as shown in Fig. 1.

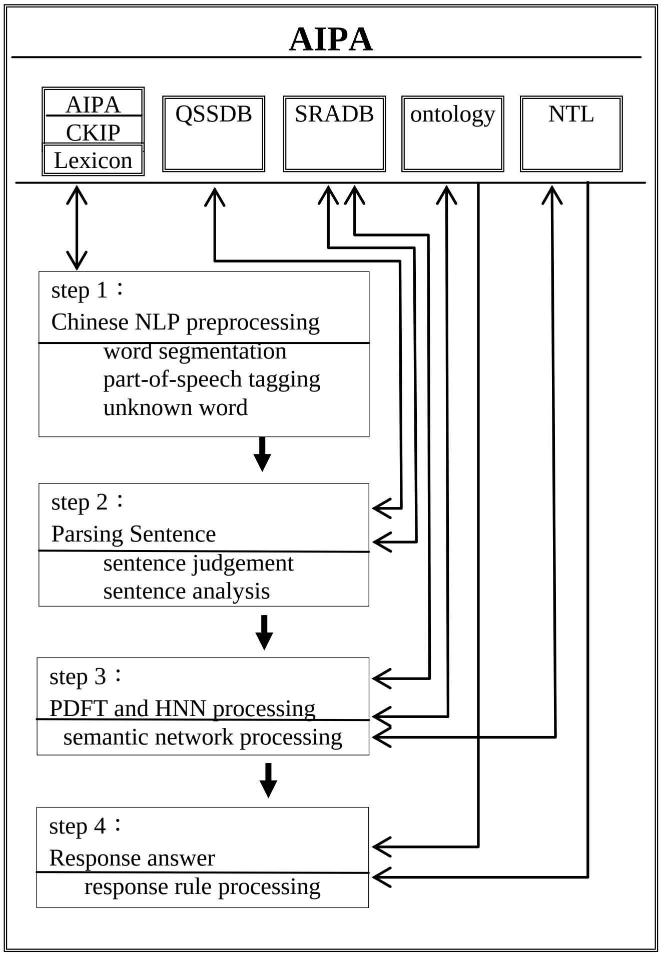

Fig. 1 The architecture of the AIPA

A. Chinese NLP Preprocessing

1. Word Segmentation

This study adopted the MM algorithm for word segmentation, identified the permutations and combinations of each vocabulary phrase, and compared from left to right or right to left with the Lexicon, repeating these actions until it met the longest word permutation and combination. We consider this permutation and combination to be the best segmentation result.

Example:
Vocabulary: 中正大學 (Chung Cheng University)
Step 1: 中／正大學; inquire for "中" in the Lexicon, answer: NA.
Step 2: 中正／大學; inquire for "中正" in the Lexicon, answer: Yes, reserved.
Step 3: 中正大／學; inquire for "中正大" in the Lexicon, answer: NA.
Step 4: 中正大學; inquire for "中正大學", answer: Yes, and confirm this is the best result for word segmentation.

2. Unknown Words

Using the human-machine interface, the sentence input by a user is processed by segmentation, and the segmented phrases are compared with the terminals in the ontology. If the comparison result is the same, then such phrase is considered a new word.

3. Sentence Parsing

AIPA does two parsing actions to a sentence: identifying the sentence pattern to determine whether it is a interrogative or declarative sentence, and analyzing the sentence pattern with triple-mode parsing module.

i. Sentence Judgment

This research classifies Chinese sentences into either declarative or interrogative sentences. Since this is a mutually disjoint judgment, so AIPA only judges against the interrogative sentences. Those sentences that are not interrogative are directly categorized as declarative. The way AIPA judges whether a sentence is interrogative is to analyze and conclude common sentence patterns by the characteristics of Chinese interrogative sentences, and then build a QSSDB (Question Sentence Sample Database) where each sentence pattern is assigned into a slot of a triple. The slots are constructed as left-script, mid-script or right-script and each script may contain one or a pair of interrogatively described vocabulary. When a script contains NULL it means that such a description is dispensable. Finally, the system judges whether a sentence is interrogative by the way of script matching.

```
FOR all sentences UNTIL No Match for slot on QSSDB
    IF MATCH ( sentence ) is script THEN
        RETURN QASS_TRUE_FLAG;
    END IF
END FOR
RETURN QASS_FALSE_FLAG;
```

QASSDB record Examples:
Slot: (請問/may I ask, NULL, NULL)
Slot: (請教/may I ask, NULL, NULL)
Slot: (什麼是/What is, NULL, NULL)
Slot: (怎麼/How, NULL, NULL)
Slot: (何時/When, NULL, NULL)
Slot: (何地/Where, NULL, NULL)
Slot: (哪/Which, NULL, NULL)
Slot: (NULL, 有多少/how many, NULL)
Slot: (能夠/Can, 回應/respond, NULL)
Slot: (可以/Can, 回應/respond, NULL)
Slot: (NULL, 嗎/right?)
Slot: (NULL, ？)

ii. Sentence Analysis

The Sentence Relation-Action Database (SRADB) is used to divide a sentence or phrase into three parts: Active-Object, Relation-Action and Passive-Object. By putting all Relation-Actions in the database, these Relation-Actions include common Chinese relation-stop-words, relation-verb-words and relation-words. The relation-stop-words and relation-verb-words are directly associated with the stop-words and verb-words recorded in the lexicon, while the relation-words are built by a rule base.

The way to match a Relation-Action is to scan the sentence from left to right; if there are vocabulary phrases consistent with Relation-Actions stored in the SRDB, the system separates out the Active-Object and Passive-Object.

```
FOR all sentences UNTIL No Matching word in SRADB
    IF MATCH ( word ) is Relation-Action THEN
        Split Relation-Action left phrase To Active-Object;
        Split Relation-Action right phrase To Passive-Object;
        EXIT;
    END IF
END FOR
```

SRADB record Examples:
Relation-word: 就是 (indeed), 喜歡 (like), 屬於 (belong to), 希望 (hope), 堅持 (insist), 出現 (appear), 隨著 (follow), 似乎 (seem), 不過 (but), 跟著 (with), 除了 (beside), 不管 (no matter), 雖然 (although), 落幕 (end), etc.

Verb-word：吃 (eat), 走 (walk), 遮住 (cover), 聞 (smell), 設計 (design), 擦 (scrub), 擊出 (hit), 分析 (analyze), 增加 (increase), 預估 (estimate), 等 (wait), 吸引 (attract), 推出 (launch), etc.

Stop-word：的 (‘s), 在 (at), 有 (have), 之 (of), 是 (be), 於 (at), etc.

B. PDFT(Phrase Driven Formal Translation)

For a sentence, the system first analyzes the sentence pattern, and determines whether it's declarative or interrogative. Next, convert the sentence into triple mode [6] and convert the corresponding vocabulary words into non-terminals or terminals. Then store them into slots of (object, event, object) or (object, attribute, object) triples. Each slot generated from the sentence has its own OID (Object ID)[7]. All slots may be associated with each other and form a semantic network, including sub-class relationships and IS-A hierarchy associations.

PDFT adopts the SRADB algorithm to disassemble the sentence into each slot (left-object, mid-status, right-object), until the left-object and right-object slots are all terminals, of which the left-object and right-object may store terminal or non-terminal objects. A non-terminal stands for a collection of objects that has not yet been determined, while a terminal stands for identified objects.

Mid-status may store descriptions of two kinds of objects: attributes and events. Attributes stand for the description (left-object) of other attributes (right-object) of knowledge, while events stand for the events (right-object) of the (left-object) knowledge. However, the terminal or attribute or event in each slot may independently or repeatedly use triples to define a semantic sub-network, as shown in Fig. 2 and Fig. 3.

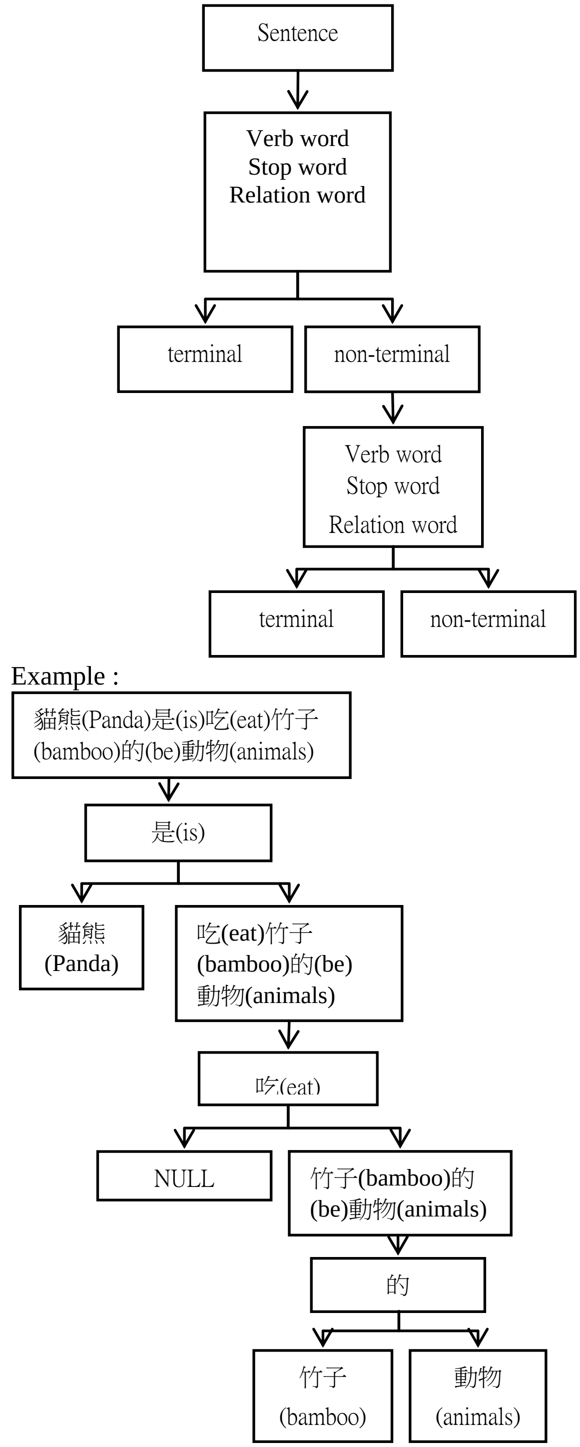

Fig. 2 Triple mode representations

1. PDFT Algorithm

```
SEGMENT( sentence );
FOR each sentence UNTIL MATCH( SRADB )
    FOR each sub sentence UNTIL MATCH( SRADB )
        Put left sub sentence into left.object;
        Put right sub sentence into right.object;
        IF CHECK ( found-word ) is attribute THEN
            IF CHECK ( slot.attribute.value ) is Known THEN
                Set Same slot.attribute.value;
        ELSE
            Build New slot.attribute.value;
        END IF
```

```
        IF CHECK( left.object ) is terminal THEN
            Put sub sentence to left.object AND Set terminal;
        ELSE
            Put sub sentence to left.object AND Set non-terminal
        END IF
        IF CHECK( right.object ) is terminal THEN
            Put sub sentence to right.object AND Set terminal;
        ELSE
            Put sub sentence to right.object AND Set non-terminal
        END IF
        ELSE IF CHECK ( found-word ) is event THEN
        IF CHECK ( slot.event.value ) is Known THEN
            Set Same slot. event.value;
        ELSE
            Build New slot. event.value;
        END IF
        IF CHECK( left.object ) is terminal THEN
            Put sub sentence to left.object AND Set terminal;
        ELSE
            Put sub sentence to left.object AND Set non-terminal
        END IF
        IF CHECK( right.object ) is terminal THEN
            Put sub sentence to right.object AND Set terminal;
        ELSE
            Put sub sentence to right.object AND Set non-terminal
        END IF
        ELSE IF
            Set the slot to its Unique ID (OID);
    END FOR
END FOR
```

2. PDFT Infinite Context-Free Language

A Context-Free Language for Triples:

S: Sentence, L: Left-sentence, M: Mid-status, R: Right-sentence

A: Attribute, E: Event, N: Non-terminal, t: terminal

$S \rightarrow LMR$

$L \rightarrow t \mid N$

$M \rightarrow A \mid E$

$R \rightarrow t \mid N$

$N \rightarrow S$

$A \rightarrow t$

$E \rightarrow t$

Dependency graph：

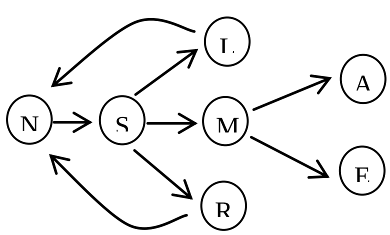

Fig. 3 Dependency graph of PDFT

C. NTL(Neural Token List)

Lastly, AIPA looks up all terminals in the CKIP Lexicon. If the answer is correct, then AIPA identifies the terminal as "knowledge" (Knowledge Token), and stores that Knowledge Token into a slot. All slots storing Knowledge Tokens form an NTL (Neural Token List), as shown in Fig. 4.

NTL provides AIPA's "longitudinal relation " and "lateral relation ". For example, suppose that NTL stores three Knowledge Tokens, respectively, U.S., president and Obama.

When we inquire AIPA with "president", we can find the related knowledge "Obama". As for the "lateral relation", besides the interactive learning on the human-machine interface, Knowledge Tokens of an OID in the same layer of a slot in the NTL are compared, and two knowledge Tokens are considered laterally associated if they share an 80% similarity.

A HNN (Hopfield Neural Network) relational module is used to do the association and recalling from NTL and the ontology. In order to make sure that the recalling converges, when the Knowledge Token recalls the same OID, the recalling stops.

```
FOR all terminals UNTIL No Matches on CKIP Lexicon
    IF MATCH (terminal) THEN
        Set terminal to Knowledge Token
        Put Knowledge Token into slot
        Build relation on NTL
        Set Unique ID (OID) to slot;
    END IF
END FOR
```

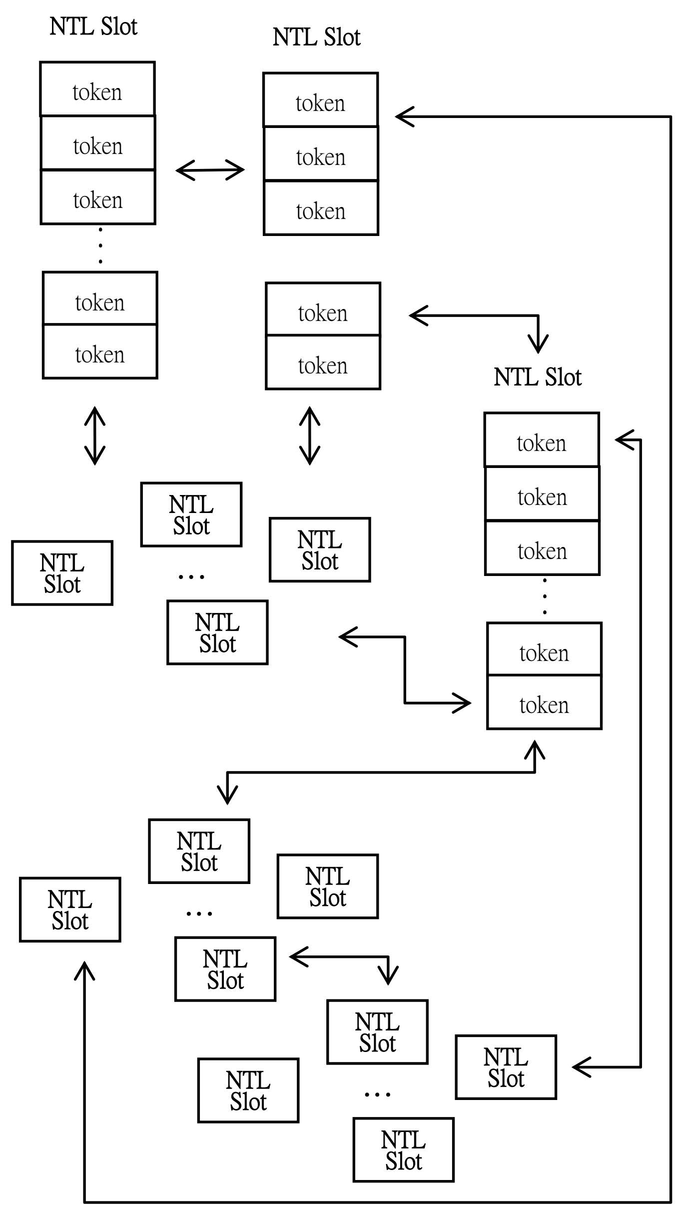

Fig. 4 The architecture of NTL

III. Examples Of Actual Operations

Example 1：

a. Untreated sentence: 韓國人是喜歡吃很辣的泡菜 (Koreans indeed like to eat spicy kimchi.)

b. Segmented sentence: 韓國人(Koreans)／是(indeed)／喜歡(like)／吃(to eat)／很辣 (spicy)／的(of)／泡菜 (kimchi).

c. Pattern Analysis: 『韓國人(Koreans) 』／『是(indeed)』／『喜歡(like)／吃(to eat)／很辣(spicy)／的(of)／泡菜(kimchi)』

d. PDFT processing：

Slot: 1.1 [oid] (韓國人 [terminal], 是 [attribute], 喜歡吃很辣的泡菜[non-terminal])

Slot: 1.2[oid](NULL[terminal], 喜歡[event], 吃很辣的泡菜[non-terminal])

Slot: 1.3[oid](NULL [terminal], 吃[event], 很辣的泡菜[non-terminal])

Slot: 1.4[oid](很辣[non-terminal], 的[attribute], 泡菜[terminal])

e. NTL processing：

Slot: 1.1 [oid] (韓國人 [terminal] in the Lexicon), is [Knowledge Token]

Slot: 1.2 [oid] (泡菜[terminal] in the Lexicon), is [Knowledge Token]

Example 2:

a. Untreated sentence: 韓國人是喜歡吃很辣的泡菜配飯 (Koreans indeed like to eat spicy kimchi with rice)

b. Segmented sentence: 韓國人(Koreans)／是(indeed)／喜歡(like)／吃(to eat)／很辣(spicy)／的 (of)／泡菜(kimchi)／配(with)／飯(rice)

c. Pattern Analysis: 『韓國人(Koreans)／『是(indeed) 』／『喜歡(like)／吃(to eat)／很辣(spicy)／的(of)／泡菜(kimchi)／配(with)／飯(rice)』

d. PDFT processing：

PDFT process in the Example 1 is omitted.

Slot: 2.1 [oid] (很辣[non-terminal], 的[attribute], 泡菜配飯[non-terminal])

Slot: 2.2 [oid] (泡菜[terminal], 配[event], 飯[terminal])

e. NTL processing：

Slot: 2.1 [oid] (飯[terminal] in the Lexicon), is [Knowledge Token]

Example 3：

a. Untreated sentence: 台灣人喜歡清明節吃潤餅 (Taiwanese like to eat Chinese burritos on Tomb Sweeping Day)

b. Segmented sentence: 台灣人(Taiwanese)／喜歡(like)／清明節(Tomb Sweeping Day)／吃 (to eat)／潤餅(Chinese burritos)

c. Pattern Analysis: 『台灣人(Taiwanese)』／『喜歡(like to)』／『清明節(Tomb Sweeping Day)／吃(to eat)／潤餅(Chinese burritos) 』

d. PDFT processing:

Slot: 3.1 [oid] (台灣人[terminal], 喜歡[event], 清明節吃潤餅[non-terminal])

Slot: 3.2 [oid] (清明節[terminal], 吃[attribute], 潤餅[terminal])

e. NTL processing：

Slot: 3.1 [oid] (台灣人[terminal] in the Lexicon), is [Knowledge Token]

Slot: 3.2 [oid] (清明節[terminal] in the Lexicon), is [Knowledge Token]

Slot: 3.3 [oid] (潤餅[terminal] in the Lexicon), is [Knowledge Token]

Example 4:

a. Untreated sentence: 潤餅真是好吃 (Chinese burritos are really delicious)

b. Segmented sentence: 潤餅(Chinese burritos)／真是(be really)／好吃(delicious)

c. Pattern Analysis:『潤餅(Chinese burritos)』／『真是(be really)』／『好吃(delicious)』

d. PDFT processing:

Slot: 4.1 [oid] (潤餅[terminal], 是[attribute], 好吃[terminal])

e. NTL processing：

Slot: 4.1 [oid] (潤餅[terminal] in the Lexicon), is [Knowledge Token]

Any terminal or attribute or event in a slot may independently or repeatedly use the triple mode to define the semantic network; for example:

Slot: 5.1[oid]（韓國人[terminal], 是[attribute], 人民[terminal]）
Slot: 5.2[oid]（韓國人[terminal], 是[attribute], 國人[terminal]）
Slot: 5.3[oid]（韓國人[terminal], 是[attribute], 國民[terminal]）
Slot: 6.1[oid]（台灣人[terminal], 是[attribute], 人民[terminal]）
Slot: 6.2[oid]（台灣人[terminal], 是[attribute], 國人[terminal]）
Slot: 6.3[oid]（台灣人[terminal], 是[attribute], 國民[terminal]）
Slot: 7.1[oid]（清明節[terminal], 是[attribute], 節日[terminal]）
Slot: 8.1[oid]（人民[terminal], 屬於[attribute], 人[terminal]）
Slot: 9.1[oid]（國人[terminal], 屬於[attribute], 人[terminal]）
Slot: 10.1[oid]（國民[terminal], 屬於[attribute], 人[terminal]）
Slot: 11.1[oid]（喜歡[terminal], 等於[attribute], 愛[terminal]）
Slot: 12.1[oid]（吃[terminal], 動作[attribute], 咬[terminal]）
Slot: 13.1[oid]（屬於[terminal], 相似[attribute], 等於[terminal]
）
Slot: 14.1[oid]（就是[terminal], 等於[attribute], 是[terminal]）

Respond to the users' questions, as shown is Fig. 5:

Example 1:

Question: 哪一國人愛吃泡菜？ (Direct word-by-word translation: In which country do people love to eat kimchi?)

1) Dete Determine the sentence pattern

Consistent with the slot: (哪/where, NULL, NULL)

Result: interrogative

2) Analyze the model of Active-Object, Relation-Action, Passive-Object

"愛吃 (love to eat)" is consistent with the verb-word "吃 (eat)" in the Relation-Action

Result: 國人 (country people)／愛吃 (love to eat)／泡菜 (kimchi)

3) Inquire the semantic network

Slot: 11.1[oid]（喜歡[terminal], 等於[attribute], 愛[terminal]
Slot: 9.1[oid]（國人[terminal], 屬於[attribute], 人[terminal]）
Slot: 5.2[oid]（韓國人[terminal], 是[attribute], 國人[terminal]）
Slot: 6.2[oid]（台灣人[terminal], 是[attribute], 國人[terminal]）
Slot: 2.2 [oid] (泡菜[terminal], 配[event], 飯[terminal])
Slot: 2.1 [oid] (很辣[non-terminal], 的[attribute], 泡菜配飯[non-terminal])
Slot: 1.4[oid](很辣[non-terminal], 的[attribute], 泡菜[terminal])

Slot: 1.3[oid](NULL [terminal], 吃 [event], 很辣的泡菜 [non-terminal])
Slot: 1.2[oid](NULL[terminal], 喜歡 [event], 吃很辣的泡菜 [non-terminal])
Slot: 1.1 [oid] (韓國人 [terminal], 是 [attribute], 喜歡吃很辣的泡菜[non-terminal])
4) Respond to the Question
韓國人(Korean)[terminal] 喜歡(like)[terminal] 吃(to eat)[event] 泡菜(kimchi)[terminal]
韓國人喜歡吃泡菜 (Koreans like to eat kimchi)

Example 2:
Question: 台灣人喜歡吃泡菜的感覺嗎？ (Taiwanese people like the feeling of biting kimchi?)
1) Determine the sentence pattern
Consistent with the slot: (NULL, NULL, ？)
Result: interrogative
2) Analyze the model of Active-Object, Relation-Action, Passive-Object
「喜歡」符合Relation-Action中Relation-word「喜歡」
台灣人／喜歡／吃／泡菜[的／感覺]
3) Inquire the semantic network
Slot: 11.1[oid]（喜歡[terminal], 等於[attribute], 愛[terminal]）
Slot: 3.2 [oid] (清明節[terminal], 吃[attribute], 潤餅[terminal])
Slot: 3.1 [oid] (台灣人[terminal], 喜歡[event], 清明節吃潤餅 [non-terminal])
4) Respond to the Question
After inquiry, the only knowledge about 台灣人 (Taiwanese) is 愛吃潤餅 (love to eat Chinese burritos) [terminal] in the ontology, NTL, so the answer is negative.
台灣人(Taiwanese)[terminal] [不(not)] [喜歡(like)] [吃(eating)[event]] 泡菜(kimchi)[terminal]，[喜歡(like)] [吃(to eat)[event]] 潤餅(Chinese burritos)[terminal] 清明節(Tomb Sweeping Day) [terminal]
不喜歡吃泡菜 台灣人喜歡清明節吃潤餅(Taiwanese not like to eat kimchi, but like to eat Chinese burritos on Tomb Sweeping Day).

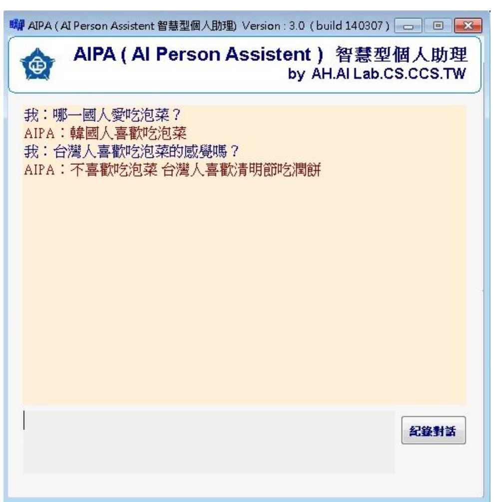

Fig. 5 The AIPA interactive talk

IV. Test Results and Discussion

We actually input 100 online news articles for AIPA to learn and had a user ask 50 questions to AIPA, respectively with 30 longitudinal relations and 20 lateral relations. The results showed that AIPA failed to answer 3 unrecognizable questions and succeeded in answering 27 longitudinally and 10 laterally associated questions. The accuracy for longitudinally associated questions was higher than 90%, while the accuracy for laterally associated questions was only about 50%. Therefore, we input another 50 relevant articles into AIPA and asked the same 50 questions. The second result showed that the number of correct answers for longitudinally associated questions remained 27, while the number for laterally associated relations became 15. It thus increased by 5 and showed significant progress in accuracy. Later on, we input another 50 unrelated articles into AIPA and asked it again the same questions. This time, the result showed that the numbers of correct answers for both longitudinally and laterally associated questions remained the same, 27 and 15, as shown in TABLE I. Therefore, the correctness of AIPA knowledge recalling has a positive correlation with the amount of learned related data. Constant and relevant learning will further increase the accuracy of answers.

TABLE I
Information about 50 questions in the AIPA system

relation / articles	correct answers	
	longitudinal	lateral
100 news	27	10
100 news + 50 relevant articles	27	15
100 news + 50 relevant articles + 50 unrelated articles	27	15

Building QA systems is interesting research. By implementing an AIPA system, we have improved the communication between humans and computers. With the AIPA human-machine interface and the design of learning models, the computers may step-by-step increase their learning efficiency like children learning a language under the instructions of users.

Although the AIPA in this study cannot have the intelligence of humans, the research of the study is expected to bring people and computers closer and make computers understand our verbal conversation, correctly interpret questions, learn articles rapidly, accurately analyze and summarize knowledge, and respond to the users with its learned knowledge.

References

[1] THE CKIP CHINESE TREEBANK: GUIDELINES FOR ANNOTATION, Chen, Keh-Jiann, Chi-Ching Luo, Zhao-Ming Gao, Ming-Chung Chang, Feng-Yi Chen, Chao-Jan Chen, and Chu-Ren Huang (2002) “The CKIP Chinese Treebank Guidelines for Annotation,” technical report,URL:http://godel.iis.sinica.edu.tw/CKIP/treebank/CHINESE TREEBANK.

[2] Chooi-Ling Goh, Masayuki Asahara, and Yuji Matsumoto. “Chinese Word Segmentation by Classification of Characters”. In Proc. of Third SIGHAN Workshop, 2004a, pp. 57–64.

[3] Hopfield, J, and Tank, D. "Neural computations of decisions in optimization problems". Biological Cybernetics, 51, 1985, pp. 141-152.

[4] Yang, Hui, Tat-Seng Chua, Shuguang Wang, and Chun-Keat Koh, “Structured Use of External Knowledge for Event-Based Open Domain

Question Answering" Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 2003, pp. 33-40.

[5] Kevin Zhang (Hua-Ping Zhang), Qun Liu, Hao Zhang, and Xue-Qi Cheng, "Automatic Regcognition of Chinese Unknown Words Based on Roles Tagging", In Proceedings of 1st SIGHAN Workshop on Chinese Language Processing, 2002. http://dx.doi.org/10.3115/1118824.1118841

[6] D.J. Buehrer, "Organizing Multimedia Objects Using a Class Algebra Database", Distributed Multimedia Databases: Techniques and Applications, Timothy K. Shih (ed.), Idea Group Publishing, Hershey, Pennsylvania, Chapter 20, pp.318-326, 2001.

[7] D.J. Buehrer, "An Object-Oriented Class Algebra", in Proceedings of ICCI '95: 7th International Conference on Computing and Information, Peterborough, Ontario, Canada, July 5-8, 1995, pp.669-685.