1000-9825/2004/15(12)1751 ©2004 Journal of Software 软件学报 Vol.15, No.12

少儿图灵测试回顾*

陆汝钤[1,2,3,4,5+], 张松懋[1,2]

[1](中国科学院 数学与系统科学研究院 数学研究所,北京 100080)
[2](中国科学院 计算技术研究所 智能信息处理重点实验室,北京 100080)
[3](中国科学院 数学与系统科学研究院 管理、决策和信息系统重点实验室,北京 100080)
[4](复旦大学 上海市智能信息处理重点实验室,上海 200433)
[5](北京工业大学 北京市多媒体与智能软件重点实验室,北京 100022)

A Retrospective View on Children Turing Test Results

LU Ru-Qian[1,2,3,4,5+], ZHANG Song-Mao[1,2]

[1](Institute of Mathematics, AMSS, the Chinese Academy of Sciences, Beijing 100080, China)
[2](Key Laboratory for IIP, Institute of Computing Technology, the Chinese Academy of Sciences, Beijing 100080, China)
[3](Key Laboratory of MADIS, AMSS, the Chinese Academy of Sciences, Beijing 100080, China)
[4](Shanghai Key Laboratory of IIP, Fudan University, Shanghai 200433, China)
[5](Beijing Key Laboratory of Multimedia and Intelligent Software, Beijing University of Technology, Beijing 100022, China)
+ Corresponding author: E-mail: rqlu@math.ac.cn

Received 2004-07-12; Accepted 2004-09-06

Lu RQ, Zhang SM. A retrospective view on children turing test results. *Journal of Software*, 2004,15(12): 1751~1763.
http://www.jos.org.cn/1000-9825/15/1751.htm

Abstract: This paper presents a research work on children Turing test (CTT). The main difference between our test program and other ones is its knowledge-based character, which is supported by a massive commonsense knowledge base. The motivation, design, techniques, experimental results and platform (including a knowledge engine and a conversation engine) of the CTT are described in this paper. Finally, some concluding thoughts about the CTT and AI are given.

Key words: Turing test; conversation system; commonsense knowledge base; closed world assumption

---

* Supported by the National Natural Science Foundation of China under Grant No.60496321 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2001AA113130 (国家高技术研究发展计划(863)); the National Grand Fundamental Research 973 Program of China under Grant Nos.2001CCA03000, 2001CB312004 (国家重点基础研究发展规划(973)); the Brain and Mind Science Project of CAS (中国科学院脑智科学基金)

LU Ru-Qian is a professor at Institute of Mathematics, AMSS, the Chinese Academy of Sciences, and an academician of the Chinese Academy of Sciences. His research interests include artificial intelligence, knowledge engineering and knowledge science. **ZHANG Song-Mao** is an associate professor at the Institute of Mathematics, AMSS, the Chinese Academy of Sciences. Her research interests include artificial intelligence, automatic generation of computer animation, natural language processing and medical knowledge processing.

摘 要: 报告了关于少儿图灵测试(CTT)的一项研究工作.研究区别于其他人的主要之处是该测试程序是基于知识的,它依靠一个海量常识知识库的支持.给出了作者研究少儿图灵测试的动机、设计、技术、实验结果和平台(包括一个知识引擎和一个会话引擎).最后给出了关于少儿图灵测试的几点研究结论和思考.

关键词: 图灵测试;对话系统;常识知识库;封闭世界假设

中图法分类号: TP18 **文献标识码**: A

1 Turing Test and Conversation Programs

TT (Turing test) is a well-recognized criterion for testing machine intelligence, proposed by Turing in 1950[1,2]. Since there is no rigorous definition of TT, the definition given below corresponds to the understanding of the author. Two subjects (also called confederates) A and B, among them one is a person (called person confederate) and another one a computer (called computer confederate), are tested by a judge C. C cannot see A and B and does not know who of them is the person or computer. C can talk to any of them and has to decide about their identity. If C cannot make the right decision, then it is asserted that the computer confederate has passed TT (showing that it has intelligence).

At the time of Turing, there was no widely accepted definition of machine intelligence and human intelligence. Turing did not try to give such a definition. Rather, he tried to design a test, which is, according to his idea, functionally equivalent to giving a definition (of a criterion on judging the intelligence of a machine), but is easier to implement and to check. This test defined by Turing, as it is given above, is then called Turing test. Obviously, his definition is in its essence experimental and behavioral, which has been one of the sources of controversies later on.

The most well known argument against Turing's thesis is the so-called Chinese room problem of Searle[3~5]. Assume there is a computer C who has passed TT in Chinese. Each person P who does not understand anything of Chinese may use C to perform a conversation with other people. The room where P and C are situated is called the Chinese room. Can we say that P understands Chinese because he can interact with the environment "in Chinese" in this way?

Nevertheless, the idea of Turing has given rise to many research works on man-machine conversation. As result, many free conversation programs have been produced. These programs are different from the structured conversation programs. In a structured conversation program, the sentence patterns are predefined and followed by the conversation partners. A finite state machine is used to parse and process the speeches and to produce outputs. This mode of conversation is functionally very limited and does not correspond to the requirement of Turing test.

The free conversation programs, however, allow the users talk to the computer in a free way. No format or pattern of sentences is predefined. At a first look, this kind of program may seem to be very difficult to develop. But if one limits oneself in a very humble domain of discourse, there are tricks to be used to make the conversation program behave in a smart way.

One of the most prominent early examples was ELIZA, written by Joseph Weizenbaum[6] at MIT. ELIZA provides simple but smooth communication with its human partner. It used a list of key words, a list of syntactic patterns and another list of transformation rules to react to the user input.

A second example is the PARRY program developed by Colby[7]. Similar to ELIZA, PARRY makes also no use of any syntactic parser. It possesses about 6 000 rules for pattern matching. It makes itself look like a mental patient of the Veterans hospital. PARRY was put on the Arpanet in 1973 and could at that time already cheat many people successfully.

Ultra HAL is another example of such conversation programs[8], which is a new version of Hal. It collects over

1000 so-called common phrases (most often used sentence forms in conversation) with responses; among them one common phrase may have several corresponding replies. A keyword-based pattern matching mechanism is used if no appropriate common phrase can be found. In this case, it looks for an appropriate reply from the database or to reword the question to transform it in a reply.

Since 1991, an international competition called Loebner test has been held each year to encourage the efforts in implementing Turing test[9]. Though a bronze award has always been handed out to the best program participating at each year's test, the award of a gold medal with 100,000 Dollars is still pending.

An essential characteristics of many these works is the lacking use of knowledge, especially commonsense knowledge. None of these systems is really knowledge based. If you ask a question like: ”how many legs does an ox have”, none of these systems, we believe, can give a reasonable answer.

2 Motivation of Knowledge Based Children Turing Test

The controversies around the significance of TT are mainly philosophical. The first motivation of our research is not to prove or disprove the justification of TT as a criterion for testing machine intelligence. We just want to explore the technical feasibility of performing such tests and of constructing a platform, which is powerful enough to let the computer win TT. This is our primary motivation.

Since it is difficult to make the computer intelligent enough to pass TT in its original sense, our second motivation is to explore the possibility of weakening the criterion posed by Turing. More precisely, we are trying to limit the subjects of TT to some particular group of minor-aged people, namely to children who have a pre- or junior school age, say, under 10 years old. This limitation has two meanings. Firstly, the computer confederate needs only to possess a minor degree of knowledge and intelligence, in order to mimic a child. Secondly, a judge with the age of a child will be less intelligent and thus will have less possibility to detect who from A and B is the person and who is the computer.

The role of knowledge played in human's intelligence is an important research topic in our experiments of TT. We want to study how and to which extent knowledge can be used to make TT more successful.

It may be meaningful to pose the question: how old is our computer? If a human being can not differentiate between a child of age n and a computer, then we say that this computer has at least an intelligence age of n.

Finally, our last motivation of doing this work is to research those topics which are relating to the implementation of TT, for example natural language understanding, natural language generation, commonsense knowledge representation and processing, human mind modeling, strategy of performing conversation and argumentation, etc.

3 First results of Children Turing Tests

There may be too many variations for performing a TT. For our first Children TT, we refine its rules as follows: the judges and the child confederates are all 5 years to 11 years old. Natural Chinese of children of this age interval was used. The conversation was only in text form (no voice recognition or generation). Each session was divided in several rounds. Each round consists of a sentence of both sides each. There are two session modes. In the limited session mode, exactly 10 rounds were allowed. In the unlimited session mode, the judge may perform any number of rounds until she thought she had collected enough information to make the decision. The last rule was that no information exchange between the two confederates should be allowed.

The first CTT were performed at the beginning of 2000 using a limited session mode. Four sessions of such tests have been made. In the first two sessions, three children of age 5 to 6 years old participated at the tests. At the end of each session, the judges were asked to answer who of (A, B) was a child and who was a computer. Both judges hesitated for a long time and then answered the question correctly, but with less confidence. When the investigator asked them why they thought B is a child, they could not explain the reason. The main impression we obtained from these tests was the inappropriate age limitation of the children. The judge was too young to understand the meaning of CTT. These children were just curious in playing with a computer and did not think how to beat the computer and to detect the real capacity of A or B.

In the following two sessions, we invited two children of age 7 to 11 years old. Their understanding of the meaning of CTT was obviously much better. At the end of the tests, they made also correct guesses about the roles of A and B. See Tabel 1 of this paper. But here we saw another problem. When we asked them why they got this conclusion, they said that they thought A were a computer because the answers produced by A were too perfect to be considered as those given by a real child. Correspondingly, in the process of conversation, they tried to attack again and again that role they thought was a child with tricky questions in order to reveal the child's real identity.

Table 1 Result of CTT in H-M / teacher-child / unlimited session mode

Session	Role A	Role B	Number of rounds	Talk with A	Talk with B	Result
1	Child	Computer	4	2	2	Correct
2	Child	Computer	10	6	4	Correct
3	Computer	Child	8	4	4	Wrong
4	Child	Computer	11	3	8	Correct

Later, we changed the form of CTT from children→children to adult→children and from limited session mode to unlimited session mode. That means, the judge is no more a child but a teacher. This has some advantages. First, the (adult) judge understands the meaning of CTT much better than a child and will provide a better cooperation. Second, a teacher's understanding about children's behavior is more dependable. Third, in this mode, more knowledge-based principles can be checked and tested. For results see Table 2.

Table 2 Result of CTT in {H-M, H-H, M-M} / teacher-child / unlimited session mode

Session	Role A	Role B	Number of rounds	Talk with A	Talk with B	Judge's decision
1	M	H	15	7	8	A=M, B=H
2	M	M	19	12	7	A=M, B=H*
3	H	M	35	17	18	A=H, B=M
4	H	M	21	0	21	A=H, B=M
5	H	M	30	0	30	A=H, B=M

*: The judge's decision on second session is wrong

In the last experiments of CTT, we increased the difficulty of CTT judges by allowing all possible combinations of the two confederates: child + computer, two children or two computers. As a result, the teacher judges need much more (about three times as much) conversation rounds to make a decision. See Table 3.

Table 3 Comparison of round numbers in Tables 1 and 2

	H-M mode	{H-M, H-H, M-M} mode
Maximal number of rounds	11	35
Average number of rounds	8.25	24

4 CTT Platform: the Knowledge Engine

The main facilities for supporting CTT consist of two parts: a knowledge engine and a conversation engine. These two parts form the CTT platform. The knowledge engine is the first salient feature, which differs our knowledge based CTT program from conventional "chatting based" conversation programs. The conventional conversation programs can not answer commonsense questions like "How many legs does an ox have?" A smart program may try to avoid answering it by suggesting another topic of conversation. But if you insist on requesting a concrete answer, these programs will fail to produce a reasonable reply. Therefore, whether the program has enough knowledge and how does it make use of its knowledge, this is the essential difference between our CTT conversation engine and many other conversation programs.

For CTT, we may assume that the conversation partner has no professional knowledge, nor high level scientific knowledge. So all knowledge we need is commonsense knowledge. Therefore, the knowledge engine of our conversation engine is operating on a commonsense knowledge base called Pangu[10~12,26,28].

4.1 The Pangu knowledge base

A commonsense knowledge base should be modularized as much as possible. The two basic kinds of commonsense knowledge modules in Pangu are agents and ontologies, which are stored in form of relations. Thus what we have is an agent-ontology-relational knowledge base. Agent and agent classes are organized in two ways. Vertically, they form inheritance hierarchies. Horizontally, they form different layers of semantic networks called ontologies, which are used to organize the agents and agent classes in a connectionist way. The communication among the agents is realized via BQML, a variation of the standard KQML language[13].

4.2 The structure and functions of CBS agents

Based on an overview of the literature, agents can be classified in six classes: passive agent (an object in the OO paradigm), reactive agent (which detects any change in the environment and reacts in an appropriate way)[14], BDI agent (an deliberate agent with belief, desire and intention)[15], social agent (agent society with cooperation and competition)[16], evolutionary agent (which may improve its intelligence by learning)[17], personalized agent (agent with human characters such as sentiment and feeling)[18].

The result of an analysis has shown that all kinds of agents are useful for building the Pangu commonsense knowledge base and its applications. On the other hand, none of the above mentioned agents could satisfy all our needs. In order to have an appropriate representation for commonsense knowledge, we have designed a new kind of agents, the CBS agents.

To simulate the behavior of a human expert, our CBS agent has two types of knowledge, organized in two parts of the agent structure: the belief part and the strategy part. They represent the static knowledge (belief) and the dynamic knowledge (strategy) of the agent. The belief part mimics the knowledge memory of an expert. It is like an information base. The strategy part consists of a set of rules. It mimics the ability of an expert to do inference for solving problems based on the belief part. In order to increase the search efficiency, each CBS agent has a capability part, which can be considered as a list of possible services of this agent. Each time an agent is asked to provide a service, this list will be checked at first to determine the possibility of providing the requested service. The link part of an agent specifies the list of ontologies this agent is involved in. The relations between agents and ontologies are many to many. For example, the ontology “school” contains among others the agents “teacher” and “student”. The agent “bus” is involved in both ontologies “travel” and “traffic”. The information about the linguistic type of the agent is mainly used for supporting the natural language understanding and generation, also for inference during conversation. The lexical type, synonym and antonym of each concept represented by this agent are all recorded in this slot.

Thus, the structure of a CBS agent looks like follows:

Agent (〈Name〉)
Father 〈conditional reference to father classes〉
Grammar 〈linguistic type of this agent〉
Link: 〈a list of ontology names〉

Capability ⟨a list of CS-Nets⟩

Belief ⟨a list of CS-Nets⟩

Strategy ⟨a CS-prolog program consisting of inference rules⟩

End of ⟨Name⟩

CS-Net is short for commonsense-net, which is a kind of semantic networks we have designed and implemented for representing beliefs and capabilities of an agent. CS-Prolog is short for commonsense Prolog, which is a further development of Prolog with a set of built-in special-purposed predicates and functions. Instead of running on a predicate database, a CS-Prolog program runs on a set of hierarchical semantic networks with communication primitives.

4.3 Ontology: Structuring the Agent Society

Roughly speaking, an ontology is a structured universe of agents and sub-ontologies which relates to some topic of discourse

The general structure of an ontology looks like follows:

⟨Type of Ontology⟩ Ontology (⟨Name⟩)

Father ⟨conditional reference to father ontologies⟩

Static Extension ⟨a list of agents involved in this ontology⟩

Dynamic Extension ⟨a list of sub-ontologies called for by this ontology⟩

Attributes ⟨a list of attributes⟩

Ontonet ⟨a list of semantic networks connecting the static extension and dynamic extension⟩

End of ⟨Name⟩

In Pangu, we have summarized various relations between agents, which were used to design more than 10 different types of ontology. They have different syntax and semantics. Some of the ontology types are: procedural ontology, descriptive ontology, conceptual ontology, causal ontology and analog ontology.

For details of agent and ontology design see Refs.[10,19,20].

4.4 Efficiency

Commonsense knowledge processing usually takes a lot of time. In order to process some queries, the computer had often to scan the whole knowledge base for an answer. Part of the reason of its inefficiency is the organization of knowledge in agent and ontology form. Therefore we have added an additional layer of knowledge representation: the relations. All agents and ontologies are transformed in relations of a relational database. We made benefit of relational calculus and got a ten times efficiency improvement.

The second technique for increasing efficiency was the use of semantic cache. During each conversation, the interest of the conversation partner will be detected and the relevant ontologies will be fetched into main memory in advance to save the memory access time and knowledge base search time. This brought again a two times efficiency improvement. Together we got a 20 times efficiency improvement.

4.5 Retrospection

The development of the commonsense knowledge base Pangu has been the main bottleneck of our CTT platform construction. Unlike professional knowledge, which can be found easily in any textbook or encyclopedia, commonsense knowledge is nowhere explicitly represented, although it exists everywhere.

The agent’s static knowledge is represented in CS-Net. The form of semantic network is suitable for representing commonsense knowledge in natural language format. Its weakness is the ambiguity of this representation. Since we do not have professional programmers, different students did the work of knowledge coding at different time. This made the ambiguity problem more serious. To overcome this difficulty, we summarized 88 normal forms of semantic net representation from several thousands of sentence samples. Then we divided the understanding process of user input in three stages. In the first stage, a natural language sentence is parsed and transformed to a semantic network. In the second stage, this semantic network is transformed to normal form, which will then be processed semantically in the third stage. This has improved the knowledge processing capability of Pangu largely.

The ontology concept has a manifold use in Pangu. It is not only used to organize agents as knowledge units[10,11], but is also used to organize a knowledgeable conversation[19,20]. This will be described in detail in next section. The third use of ontology is to keep a reasonable organization of knowledge storage. Its fourth use is to provide a prediction of next conversation stream for implementing a semantic cache to improve the run efficiency of conversation. For details see Refs.[12,21].

5 CTT Platform: the Conversation Engine

The conversation engine mainly deals with the linguistic part of the CTT platform[27]. We omit most of the details and discuss only two points here.

5.1 Expectation driven conversation

We consider the conversation as a continuous process of producing and removing expectations. The question “how many legs does an ox have?” produces a knowledge expectation, which is covered by the answer “Four”. Similarly, the speech “my mother is sick” produces a comfort expectation, which is covered by the reply “I am sorry to hear that”.

Expectation means here pragmatics. Note that our definition of expectation is that of a third part observer, which is not the same as that given by Austin[22] and Searle[23]. They defined expectation as something expected by one participant A of a conversation from another one B. We define it as any possible reaction of B to A given a speech of A to B. For example, A expects “yes, I will” from B when he asks B to do him a favor. But an answer like: ”Sorry, I can’t” belongs also to the expectation of an observer The more appropriate A’s reaction is, the more natural is their conversation, and the more successful is the Turing test.

We define four sets of expectation types, where x and y denote the two conversation partners. They are:

Information expectation, if the sentence inputted by x is a query, or if it contains information defect, such that y might try to point it out or even to correct it.

Sentiment expectation, if the sentence inputted by x contains message (bad or good information, bad or good attitude of x towards y) which may influence the sentiment of y.

Attitude expectation, if the sentence inputted by x contains message which may stimulate y to express its attitude (appreciate, depreciate, comfort, congratulate, thank, complain, etc.) towards x.

Review expectation, if the sentence inputted by x contains message (news, events) which may stimulate y to express its opinion.

The computer confederate not only tries to understand the expectation of the judge and to cover (fulfill) it, but also tries to check whether his own expectation is understood and respected by the judge.

The basic idea of implementation is to have a state transition diagram. But a simple state transition diagram without memory is not capable to support a conversation, even less to support a context sensitive conversation. We use multiple stacks to store multiple expectations.

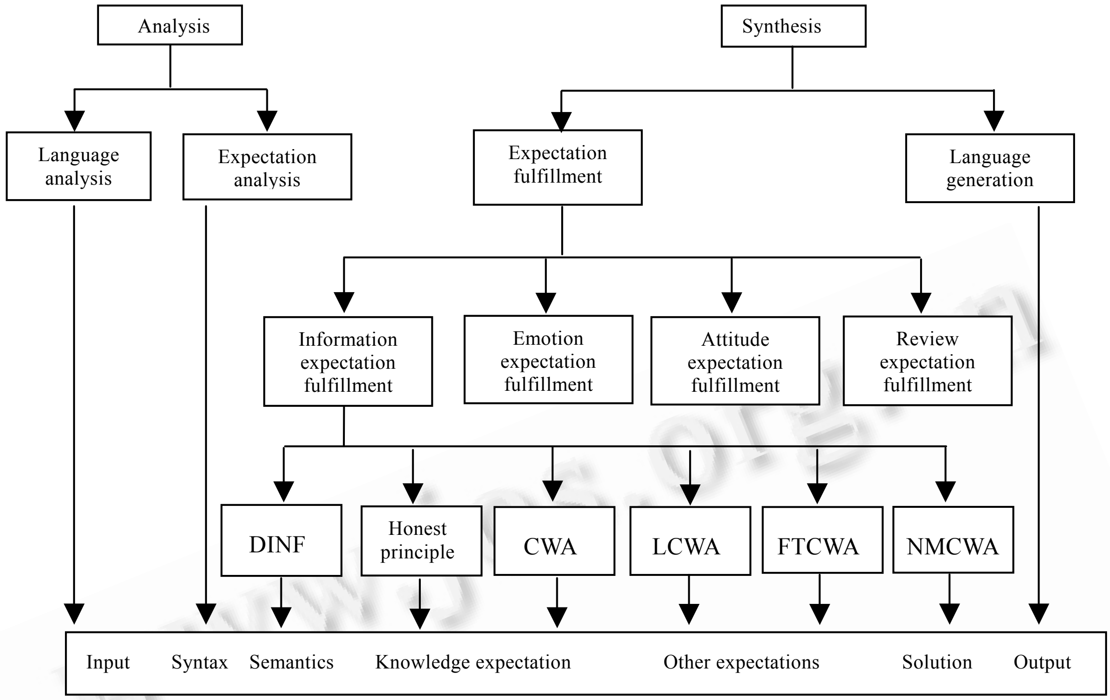

Fig.1 The expectation blackboard

5.2 The blackboard for expectation processing

The blackboard system BLAS[24] of the conversation engine is used for expectation processing. BLAS is a hierarchical BS. Its architecture is tree like and has several levels. The knowledge sources of level 0 are analysis and synthesis. In level 1, the analysis component is separated in language analysis (syntax and semantics) and expectation analysis (pragmatics), while synthesis is separated in expectation generation and remove which are further separated in processing components of the four types of expectation. Each expectation-processing component has its sub-components in turn. The knowledge expectation component for example takes control over the different strategies of query answering mentioned in previous sections. A sketch of BLAS is given in Fig.1.

The trick of BLAS's natural style conversation is its smart use of knowledge. In our knowledge base, we have not only agents such as zoo, tiger, lion, etc., but also ontologies such as visit zoo, feed animal, take a bus, etc. Each ontology is a topic of discourse. Once the computer confederate detects an ontology, which covers the judge's speech (partially or in whole), it makes use of knowledge provided by this ontology and produces knowledgeable sentences. For example, if the judge's first inputted sentence reminds of the ontology "visit zoo", then knowledge contained in this ontology such as: children are often accompanied by their parents or teachers when visiting zoo; the aim of visiting a zoo is to see the animals of the zoo; an interesting program of visiting the zoo is to see the animal keeper feeding animals, etc. can be used to produce knowledgeable replies.

During each session, the computer confederate maintains a global list of all conversation topics and will detect topic change in judge's speech, and then adapts its strategy to this change correspondingly.

6 Refining the Closed World Assumption

In case the conversation engine does not have enough knowledge for constructing a reply to its conversation partner, the easiest way is to say "don't know". We call it the honest principle. But it may also resort to the principle of logical complementation. One of such principles is the so-called closed world assumption (CWA) of Reiter. It says that each proposition which can not be proved by a knowledge base, should be considered as false w.r.t this knowledge base. For example, if it is not said in the knowledge base that an ox has two tails, then the proposition 'an ox has two tails' is false. In this sense, we would like to reformulate CWA in the following simple form:

Definition 1 (**CWA1**). Given a knowledge base *K* which is consistent, and a set of propositions *P*, the Closed World Assumption with respect to *K* and *P* assumes that:

1. There exists a total function $cwa_K(p)$ over the domain $P=\{p\}$,

2. The value domain of cwa_K is {true, false}.

For each p of *P*, if it can be proved with K, that p is true, then $cwa_K(p)$=true, otherwise $cwa_K(p)$=false.

This CWA is not perfect. It ignores the possibility of 'don't know' among the three possible answers {yes, no, don't know} and transform the original question to one of Boolean type. The traditional approaches might be useful for database query, but certainly not suitable for Turing conversation. The CWA can be applied to Turing conversation safely if and only if our knowledge is complete. Imagine we have a large knowledge base K, for each p of the predicate set P, either p or ~p can be proved with K. In this case, we can delete all knowledge items from K, which are only used for proving the ~p's, to form a smaller knowledge base K'. Then, any ~q can be considered as proved by K' if q cannot be proved by it.

But what will happen if our knowledge is not complete? In this case, we are not able to construct the knowledge base K' mentioned above. This is just the case for commonsense knowledge whose scope is open. . Here the use of CWA would be dangerous. Everything, which has not yet been coded in the knowledge base, would be considered as false.

This fact reminds us of the necessity of differentiating between "no" and "don't know", i.e., the necessity of extending the CWA principle. For the sake of simplicity, the formalization below is given in form of CWA1. But it applies also to CWA2.

Definition 2 (**Parametric Theory**). Given a set of n concepts $C=\{C_i|i=1,2,\ldots,n\}$, where *C* forms a tree like inheritance hierarchy. Each C_i is attached with a theory T_i such that

1. if C_j is a father concept of C_i and if the proposition $p(C_j)$ can be proved with T_j, then $p(C_i)$ can also be proved with T_i.

2. if $p(C_i)$ can be proved with T_i for all sub-concepts C_i of C_j, then $p(C_j)$ can also be proved with T_j.

The set $\{(C_i,T_i)\}$is called a parametric theory.

Definition 3 (**LCWA**). Given a knowledge base $K=\{(C_i,T_i)|i=1,2,\ldots,n\}$ which is a parametric theory, and a set of propositions $Q=\{q_j(C_i)|i=1,2,\ldots,n;j=1,2,\ldots\}$, the Latticed Closed World Assumption with respect to *K* and *Q* assumes that:

1. There exists a total function $lcwa_K(x)$ over the domain *Q*,

2. The value domain of $lcwa_K(x)$ is *L*= {true, false, don't know, depends},

3. If $q_j(C_i)$ can be proved with T_i, then $lcwa_K(q_j(C_i))$=true,

4. If $q_j(C_i)$ can be disproved with T_i , then $lcwa_K(q_j(C_i))$=false,

5. If $q_j(C_i)$ can not be decided with T_i, but $q_j(C_h)$ can be proved with T_h for $i\neq h$, then

5.1. If C_h is not a sub-concept of C_i, then $lcwa_K(q_j(C_i))$=false,

5.2. If C_h is a sub-concept of C_i, then $lcwa_K(q_j(C_i))$=depends,

6. Otherwise $lcwa_K(q_j(C_i))$=don't know.

It is called a latticed CWA, because its value domain is not the pair {true, false}, but the lattice L={depends <{true, false}<don't know}

Proposition.

1. In LCWA, the mapping $\{q_j(C_i)\}\rightarrow L$ is unique,
2. A proposition is true in LCWA if and only if it is true in CWA.

Proof. Easy.

A knowledge base organized in an inheritance hierarchy of agents, such as our Pangu knowledge base, can be considered as a parametric theory, where each agent is a concept, all believes of an agent A_i and those inherited from its ancestors form the theory T_i. Let us consider the following four questions q_j (bird):

A bird needs food. (=true, because its father "animal" has a belief "need food")

A bird can play piano. (=false, because a bird has no belief "can play piano", but a human has)

A bird can fly. (=depends, because a bird has belief "can fly" with exception)

A bird can get contaminated with AIDS. (=don't know, because nothing is said about AIDS in any belief of the knowledge base)

7 Intelligence Age and Intelligence Quotient

Let us recall the famous prediction of Turing made in his founding paper:

I believe that in fifty years' time it will be possible to program computers, with a storage capacity of about 10^9, to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning.[2]

According to the praxis of Loebner test held in last 10 years, we can conclude that this prediction of Turing has already failed. But Turing did not say anything about the details of the interrogator (i.e. judge) and used just a word "average" to get rid of any further discussion on the concrete conditions, such as age, sex, nationality, profession, etc. about this interrogator. However, the result would have been totally different if there had been some limitations on the various factors of the test design. Namely, if we consider the success rate of the Turing test (=chance of making the right identification) as a function of these factors, then we can reformulate Turing's prediction in the following way.

A proposed frame of prediction:

We believe that in $f(x,y,t)$ years' time it will be possible to program computers, with a storage capacity of about $g(x,y,t)$, to make them play the imitation game so well that an average interrogator of age x will not have more than y per cent chance of making the right identification after t minutes of questioning.

This frame of prediction, where f and g are both real and continuous functions, reflects one of the motivations of our children Turing test. We want to lower down the difficulty of Turing test by limiting the competence of the judges (interrogators). Historically, this is not the first try of weakening the requirement of Turing test. Colby's experiment of making the computer mimicking a paranoid was one of the examples[7]. Two further parameters are the time of questioning and the success rate (in percentage).

Turing thought five minutes questioning should be enough for a judge to have 30% chance to make a right identification. He didn't say whether this percentage would be higher if more time for questioning were allowed. In this respect, we have the following

Conjecture 1.

There is a critical length of questioning time (measured in minutes) $CRT>0$. For any $t>CRT$, we have

$$f(x,y,t)=f(x,y,CRT),\ g(x,y,t)=g(x,y,CRT)$$

That means, over some limit, more conversation time does not bring more information, which would be useful for making a right identification.

There was experimental evidence that this conjecture may be true. Below is a citation from the report about the year 2000's Loebner test:

Each judge spent up to fifteen minutes at each terminal. They were asked to make a judgment whether the respondent was a human or a computer after five minutes and then again after 15 minutes. In a few cases the judges changed their judgments after 15 minutes but most initial judgments remained unchanged[25].

Our second conjecture is as follows:

Conjecture 2.

There is a critical age $CRA>0$ of interrogator. For any $x>CRA$, we have

$$f(x,y,t)=f(CRA,y,t),\ g(x,y,t)=g(CRA,y,t)$$

That means, generally, over a certain threshold value, increase the age of the interrogator will not decrease the chance of passing Turing test by a computer. In fact, nothing was mentioned about the ages of the judges in the Loebner report.

Next is our third conjecture.

Conjecture 3.

$$\lim_{y->0} f(CRA,y,CRT)=\infty,\quad \lim_{y->0} g(CRA,y,CRT)=\infty$$

It means there will never be a day, on which TT will be successful in its real sense. In another word, the success of TT can only be an infinite process. This conjecture of us might be criticized as too pessimistic. We would be very happy if it will be refuted by practice in some foreseeable future.

Finally, we put forward our last conjecture:

Conjecture 4.

There is no other (widely accepted) criterion for measuring the final success of AI (proving the intelligence of a computer), which would be earlier realized than TT.

In the theory of computation, there is no other computational mechanism (recursive function, unrestricted Chomsky grammar, …), which is more powerful than Turing machine. Similarly, with conjecture 4, we believe that there is no other mechanism (symbolic inference, computational intelligence, neural networks, non monotonic reasoning, …, or their combination), which would help AI reach its final goal sooner than TT.

8 An Assessment of the Achievements and Future Work

The results we have obtained so far are encouraging, but leave many problems to be clarified. After we were happy to see that our computer has passed some sessions of CTT, we would like to ask ourselves two questions: first, are we justified to say that our CTT did not violate the essence of Turing test as it was defined by Turing himself in 1950? Second, to which degree can the (partial) success of any TT demonstrate that the computer confederate has intelligence?

8.1 Beyond the Chinese room problem

To answer these two questions, note that there are well-known controversies and debates about the significance of TT, among them the well-known Chinese room argumentation of Searle[3]. We will not discuss it in this paper and will only add some points to it. No matter whether TT can really check the intelligence of a computer, one thing is clear: it can only check that part of intelligence, which is demonstrated in speech. It neglects for example everything, which relates to pattern recognition. A perfect TT should at least include computer vision and voice recognition. In our TT, only message in text form was used. So it is not perfect even in the sense of speech intelligence.

This is not yet all. Human intelligence should include labor intelligence, a kind of intelligence involving labor work. The success of TT does not say anything about whether the computer can, for example, build a tower with sand on a beach.

8.2 The knowledge bottleneck

Almost all current conversation programs use a lot of chatting strategies. They use patterns to match user inputs and to transform them in reply sentences, without a parsing process. This way of going ahead is very similar to the case described by Searle. To be frankly, these programs are “cheating” the users. There are lots of cheating strategies used in their programs. The need of cheating is caused by the lack of knowledge, including the lack of linguistic knowledge.

As it was said before, the main difference of our system from the current conversation programs is that our system is knowledge based, especially commonsense knowledge based. But the amount of commonsense knowledge is immense. It means that our knowledge base will never be complete. Therefore, The conversation engine suffers from the same problem of knowledge bottleneck. Only the degree of knowledge lacking is different.

8.3 Qualification of judges

Another problem of TT follows from the discussion above: the result of TT depends on the intelligence of the judge. There is no abstract person. Each person is concrete. Each person has a different level of intelligence. Which person can be considered as qualified enough to act as a judge in TT? Children? Adults? Common people? AI experts? It seems that the judge should not be a single person, but the average of a large group of persons. And the test should not be a single test, but an infinite series of tests. Each of these tests, when successful, will witness an increase of the machine intelligence towards the real human intelligence.

Thus, our conclusion is: there will never be a day, on which TT will be successful in its real sense (the sense meant by Mr. Turing). In another word, the success of TT can only be an infinite process.

8.4 Future work

We have noted that the definition of TT as it was given by Turing self is not unique. There may be different possible variations. We will study them in more details in our next project. To explore these variations and compare their significance with respect to the determination of machine intelligence, different designs of TT have to be tested.

Acknowledgement Our experiments in Turing test are part of the CNSF project “practical aspects of commonsense knowledge”. Other participants of this project include Prof. Shi Chunyi of Qinghua university and Prof. Li Shixian of Zhongshan University with their research groups; Prof. Zhi Jin, Prof. Wei Zichu, Liu Lengning, Yang,Ping Fan Lu, Zhao Chong, Yang Fan, Jin Xiaolong, Zheng Hong, Liu Hongge, Shu Cheng, Hu Sikang, Yi Nan, Zhao Xiaoling of AMSS. We thank all of them for their contribution.

Reference:

[1] The Alan Turing Home Page. 2000. http://www.turing.org.uk/turing/

[2] Turing A. Computing Machinery and Intelligence. Mind, 1950.

[3] Searle JR. Minds, Brains, and Programs. Behavioral and Brain Sciences, 1980,3.

[4] Hauser L. Searle’s Chinese box: Debunking the Chinese room argument. Minds and Machines, 1997,7:199~226.

[5] Kanngießer S. Inside and Outside the Chinese Room.

[6] Weizenbaum J. ELIZA—A computer program for the study of natural language communication between man and machine. CACM, 1966,9:36~45.

[7] Colby KM. Artificial Paranoia. Artificial Intelligence, 1971,2.

[8] Ultra Hal. http://www.zabaware.com/

[9] 1997 Loebner prize contest results. 2000. http://www.loebner.net/Prizef/loebner-prize-1997.html

[10] Lu RQ, Shi CY, Zhang SM, *et al*. Agent oriented commonsense knowledge base. Acta Scientia, 2000,43(6):641~652.

[11] LU RQ, Zhang SM. PANGU—An Agent-Oriented Knowledge Base. World Computer Congress, 2000.

[12] LU RQ, Zhang SM, *et al*. Design and implementation issues of agent oriented knowledge base. In: IKS 2002. 2002.

[13] Labrou Y, Finin T. A proposal for a new KQML specification. TR CS-97-03, UMBC, 1997.

[14] Muller JP, Pishel M, Thiel M. Modeling reactive behavior in vertically layered agent architectures. LNCS 890, Springer Verlag, 1995. 261~276.

[15] Rao AS, Georgeff MP. BDI Agents: From theory to practice. In: Proc. of the 1st ICMAS-95. 1995. 312~319.

[16] NWANA HS. Negotiation strategies: An overview. Internal Report, BT Labs, 1994.

[17] Weiss G. Distributed artificial intelligence meets machine learning. LNAI 1221, Springer Verlag, 1997.

[18] Trappl R, Petta P. Creating personalities for synthetic actors. LNAI 1195, Springer Verlag, 1997.

[19] Fan L. Ontology based commonsense reasoning [MS. Thesis]. Math. Inst. CAS, 1999.

[20] Lu RQ, Zhang SM, *et al*. Talk to computers in natural style. ISFST-99, 1999. 346~348.

[21] Zheng H, Lu RQ, *et al*. Ontology based semantic cache. JCST, 2002.

[22] Austin JL. How to Do Things with Words. Oxford: Oxford University Press, 1962.

[23] Searle JR. Speech Acts: An Essay in the Philosophy of Language. Cambridge: Cambridge University Press, 1969.

[24] Liu LN. Research on children Turing tests [MS. Thesis]. Math. Inst. CAS, 2000.

[25] Moor JH. The status and future of the Turing test. Minds and Machines, 2001,11:77~93.

[26] Jin XL. Consistency and completeness of commonsense knowledge bases [MS. Thesis]. Math. Inst. CAS, 2001.

[27] Yang F. Research on conversation processing in Turing tests [MS. Thesis]. Math. Inst. CAS, 2001.

[28] Yang P. Agent based commonsense reasoning [MS. Thesis]. Math. Inst. CAS, 1999.

第 10 次全国 Petri 网学术年会暨形式化方法学术讨论会

征文通知

由中国计算机学会 Petri 网专业委员会主办的第 10 次全国 Petri 网学术年会暨形式化方法学术讨论会将于 2005 年 10 月在镇江召开（江苏大学承办），会议将对 Petri 网理论及应用，以及并行处理的形式化方法开展广泛、深入的讨论。现发出征文通知。

一、征文范围

Petri 网理论研究；Petri 网在各领域的应用研究；Petri 网工具开发与 Petri 网运行仿真技术；相关的并发模型及仿真技术研究；并行计算与并行处理的形式化方法；其它相关问题的研究。

二、征文要求

1、凡在正式刊物或其他学术会议上发表（或录用）过的论文不再征用；2、投送论文无论录用与否概不退稿，请作者自留底稿；3、会议论文集以《系统仿真学报》增刊出版（部分优秀的论文可推荐到正刊发表），寄送的论文要求按《系统仿真学报》要求的论文格式（可在网上查阅）编辑打印；4、被录用的论文，按要求修改后寄出电子版（软盘）,并按规定寄出版面费。5、投送论文请写清联系人姓名、地址、(包括邮政编码)，以便联系。

三、重要日期（邮寄以邮戳日期为准）

征文截稿日期：2005 年 5 月 31 日

发出录用或（和）修改通知日期：2005 年 7 月 10 日

编辑打印好的稿件返回日期：2005 年 7 月 31 日

四、论文请寄：北京航空航天大学软件学院（邮编 100083）联系人：武晓乐

电话：010－82316479 86354705（小灵通） E-mail：wuxiaole@buaa.edu.cn