INFORMATION DISSEMINATION AND STORAGE FOR TELE-TEXT BASED CONVERSATIONAL SYSTEMS' LEARNING

Ram Gopal Raj, S. Abdul-Kareem
Department of Artificial Intelligence
Faculty of Computer Science and Information Technology
University of Malaya, 50603 Kuala Lumpur
Telephone Number: + 6 012 3070435
gopalraj@tm.net.my, sameem@um.edu.my

ABSTRACT

Conversational systems or chatterbots converse/chat by learning from their interactions with users. To do this the systems must have an adaptive knowledge base that can be updated by the systems themselves. RONE is a tele-text based conversational system. RONE's knowledge base is built using SQL and accessed using the main Java application. Additionally, RONE uses conjunctions and prepositions as markers to expedite the dissemination and storage of information which helps him learn. In this paper, we describe the approach RONE uses to break up new information for learning purposes - the principle technique introduced here being the storage of information in a format to answer all the possible questions directly without inference. We also look at other conversation based learning approaches and their limitations. Further, we compare RONE's performance against some contemporary conversational systems and provide evidence of the relative superior informational accuracy of RONE's responses to user interrogation. RONE's better performance is noteworthy because it is relative to systems which are Loebner Prize medal winners.

Keywords: ***Information/sentence dissemination, conjunctions, prepositions, correlative conjunctions, informationally correct responses***

1.0 INTRODUCTION

Conversational systems also called chatterbots (robots for chat) are computer programs designed to simulate intelligent conversation with one or more human users through speech or text, [M. Mauldin, 1994].

Additionally, some systems compete for the Loebner Prize (started in 1990 as the first formal application of the Turing Test), [Crown Industries Inc., 2009]. To pass the Turing Test, "a system must convince at least 30% of the human interrogators that it is indistinguishable from humans", a pass mark set by Alan Turing, who devised the Test in 1950 "for deciding whether a machine was capable of thinking like a human", [A. M. Turing, 1950].

Where no system passes the Turing Test, the system that provides the **most** human-like conversation for the year, wins a Bronze Medal.

No system has yet passed the Turing Test but recent Bronze Medalists, [Crown Industries Inc., 2009], are Jabberwacky in 2004, 2005 and 2006, Ultra Hal in 2007 and ElBot (which attained 25% - convinced 3 of 12 interrogators) in 2008.

RONE which we introduced in a previous paper, [R.G. Raj, 2008], is a conversational programme/system that chats through tele-text. We are building RONE with the aim of enabling him to pass the Turing Test in the near future. We provided a description of another storage method, for storing some of the system's programming conditions as part of the knowledge base in [R.G. Raj, 2009].

Thus, leading from the above mentioned issues, we discuss in this paper, the:

1. Approach RONE uses to break up new information for learning and conversational purposes. An approach, we believe, enables RONE's superior conversational performance.

2. Results of the test conducted to compare RONE's performance against Jabberwacky, Ultra Hal and ElBot.

1.1 Building Large Knowledge Bases

CYC [Lenat and Guha, 1990] is a very large knowledge base project aimed at capturing human commonsense knowledge. The goal of CYC is to encode the large body of knowledge that is so obvious that it is easy to forget to state it explicitly. Such knowledge base could then be combined with specialized knowledge bases to produce systems that are less brittle and unpredictable that those available today.

CYC represents a specific theory of how to describe the world and it can be used for AI tasks such as natural language processing and understanding. CYC contains representations of events, objects, attitudes and etcetera. CYC also is particularly concerned with issues of scale, meaning what happens when knowledge bases containing millions of objects are built. Notice that CYC has a huge amount of complexity even in the description of its methodology.
CYC's knowledge is encoded in a representation language called CYCL. CYCL is a frame-based system that incorporates most of the techniques typically used in knowledge base representation. These techniques include multiple inheritance, slots as full-fledged objects, transfers-through as well as mutually-disjoint-with. CYCL generalizes the notion of inheritance so that properties can be inherited along any link. This means it is supposedly not limited by *is a* and *instance*.

In addition to frames, CYCL contains a constraint language that allows the expression of arbitrary first-order logical expressions. While forward rules can be very useful, they can also require substantial time and space to propagate their values. If a rule is entered backward, then the system defers reasoning until the information is specifically requested. CYC maintains a separate background process for accomplishing forward propagations. A knowledge administrator or bot master as they are so frequently known can continue entering knowledge while its effects are propagated during idle input time. Notice that despite its complexity, CYC still requires a human to enter information to allow it to increase or expand its knowledge base.

The constraint language itself allows for the expression of facts as arbitrary logical expressions. First order logic is much more powerful than CYC's frame language. However both are maintained because frame-based inference is very efficient, while general logical reasoning is computationally hard. CYC supports about twenty types of efficient inference mechanisms including inheritance and transfers-through. Each inference mechanism has its own truth maintenance facility. [Lenat and Guha, 1990], claim that the constraint language allows for the expression of facts that are too complex for any one of these mechanisms to handle. However as ALICE proves such systemic and implementation complexity may be unnecessary and too difficult to manage in terms of system maintenance. Every additional module added to the system will result in an ever increasing tediousness, in terms of upgrading and maintaining the system.

Specialized knowledge based systems are brittle. Due to their restricted parameters they cannot cope with novel situations and do not posses graceful degradation in their performance. However that said, when humans are required to develop such large knowledge bases, it becomes a very labor intensive task. Think about it imagine having to map enough real world knowledge to perform fluent computer human interaction. What would be considered a sufficient amount of knowledge? Would it not be more expeditious to have the program or entity itself build up its own knowledge base?

1.2 OTHER LEARNING TECHNIQUES AND THEIR LIMITATIONS

RONE uses its conversational ability to demonstrate its learning or adaptive knowledge base. Notice that while CYC and ALICE have knowledge bases that incorporate a great deal of real world knowledge, it is all geared towards providing conversational skills. An, administrator updated, knowledge base is in no way a demonstration of a learning system. ALICE and CYC are not trying to gain more knowledge. So, while we agree that to some extent ELIZA’s and especially ALICE’s conversational techniques and protocols are useful to provide a good platform for our system and Jabberwacky’s learning is more of contextual mirroring of a user, an improved method of information dissemination for learning is needed. The adaptive parts of RONE or its learning protocols are an area of new exploration. We therefore conclude the following:

1. Current conversational systems focus entirely on providing a reply with no regard for the nature of the input. For example, it did not matter to the systems as to whether the input is a statement, question or order/command.

2. Whether that reply is informationally correct is irrelevant in their scheme.

3. There is no utilization of grammar or the breaking down of the information, to “understand” the information received. Consequently, when systems such as Jabberwacky, Ultra Hal and ElBot are tested; they fail – they provide informationally incorrect or absurd answers as shown earlier.

Further, it appears that the conversational systems, work by identifying specific words or phrases used by the human and matching them with pre-programmed responses. Thus giving the impression of conversing without having to actually understand what they are talking about, [M. *Mauldin, 1994*].

For instance, a look at some different techniques used for storing and accessing inputs shows why simply storing sentences, severely limits a system’s capabilities. Examples of 4 known techniques and their respective limitations are as follows:

1. **Priority Matching**. One can view priority matching as a sort of “best first search”. A system that utilizes this technique would store a list of key words and the appropriate responses to those key words as well as a priority listing for each word, [Bush, 2001]. A user

inputs a sentence, the sentence is broken up, each word is reviewed and based on hits from the knowledge base, a response is determined. Take the statement, “Harry ran away.” as an example. Depending on the system’s design, the word ‘ran’ may be accorded the highest priority, and the system converts it and posts the response “You say Harry ran away?” or “Interesting.” However because no real learning is taking place, any further queries like “Who ran away?” may result in a response such as “Why don’t you tell me.” Thus showing that, while this technique is effective in terms of forming a response easily, it fails to improve on the quality of the conversation that the system can provide and becomes very repetitive after a while.

2. **Question/Statement – Response Storage**. This technique involves simply storing the user’s response to every system output as a potential future response, [R. Carpenter, 2009]. For example, if the system states “How are you?” and to which the user responds, “I am fine.” The next time the system is asked “How are you?” it can say “I am fine” or select randomly from a list of answers assuming more than one is available for the same instance. An effective technique assuming that the responses given by users are accurate and generic. ‘Accurate’ meaning the user does not say absurd things like “I am a pineapple!” and ‘generic’ meaning that the answer to “system: Who are you?” “user: I am John.” is unusable because the system cannot, or for accuracy sake should not, say that it is John.

3. **Context – Response Storage**. This is an elaboration of the question/statement – response storage technique. It works in basically the same way, but stores the user’s responses to a clutch or group of questions/statements, setting a higher priority for a given answer if multiple questions or inputs match to a previous instance, [R. Carpenter, 2009]. The limitation here again is similar to the question/statement – response storage technique, rather than actually trying to learn the informational content, the system learns the conversation flow, leading to errors when a user does not stick to the same conversation flow. E.g. If the system asks, “Who are you?” and the user replies, “I am John.”. The system may then ask, “When were you born?”. If the user then gives an unrelated response and says, “Harry ran away” the response is stored and linked with the initial question and response. So in a future conversation, should John ask the system “When was I born?” the system will respond saying, “Harry ran away”.

4. **Topic – Response Storage**. Every potential response is also tagged with a relevant topic in this technique, [Bush, 2001]. For example, “Harry ran away from school because he wanted to go home”, is stored under the topic “ran away”. The problem with this technique is that often topic matches can result in totally unrelated answers, for example “Mary ran away?” may be answered with “Harry ran away from school because he wanted to go home”. Ignoring the topic or subject of conversation can also lead to incorrect responses.

In addition and in the case of Jabberwacky, Ultra Hal and ElBot it was noted that:

i. Jabberwacky stores all questions and answers, it receives during its conversations, in a huge XML database, and selects the most suitable replies from that database during subsequent conversations, [R. Carpenter, 2009].

ii. Ultra Hal uses a proprietary algorithm that utilizes the WordNet lexical dictionary to store all conversations in its database and uses them to come up with responses in the future, [Zabaware inc. 2009].

Little technical information about ElBot is available because it is a proprietary software, [Artificial Solutions, 2009]. However, based on our perception of ElBot's performance and types of its responses, we suspect that Elbot uses a form of topic matching (Topic-Response Storage).

2.0 HOW CONJUNCTIONS AND PREPOSITIONS AFFECT A SENTENCE.

There are different types of conjunctions and prepositions and each type has a different effect on a sentence, [Nandy, 2002]. The differences help us to break up a sentence and derive the component information in that sentence. For example, correlative conjunctions represent components of equal status on either side of that conjunction. For instance, "Harry and Larry ran away" contains the components "Harry ran away" and "Larry ran away". "Harry is a kind and helpful person" contains the components "Harry is a kind person" and "Harry is a helpful person". In addition, it must be pointed out that when the components are derived, the conjunction is omitted. The exception to this rule is when the correlative conjunction is packed between two verbs. For instance in, "Go and get me the book", the actions are complimentary rather than equal.

Another important aspect we employed in RONE, is where a correlative conjunction affects the subject and predicate of a sentence differently. That is, either there are two subjects performing the same predicate or there is one subject performing two predicates.

Prepositions help RONE determine time and space components, in that they represent the where and when components of information. For example, "after" and "until", represent the when or where components. The, where components come with "at", "on" and "in". However, while they are prepositions of place, very often they are used in time representations such as "Harry was in time for class" or "Harry was at the meeting on time" or "She will meet you at one o'clock". But, when the prepositions of place are used in a time representative aspect, they are always followed by a time related component. Hence, we can see how a piece of information can be tagged with the correct question word. For example see Table 1;

Table 1: Examples of Components of Information

Question Words	**Information (Conjunctions and Propositions)**
Where	from, to, in
Why	because, for
When	at, on, after, before
How	by, with

3.0 OVERVIEW OF RONE

We feel that conversational systems are capable of providing meaningful conversations only when they can learn from their interactions with users. E.g. learn the user's name. To do this the systems must have an adaptive knowledge base programme that can be updated by the systems themselves. Having an adaptive knowledge base would also reduce the problem of having to build in great amounts of real world knowledge into a system since the system itself can acquire such data.

RONE's knowledge base is built using SQL and accessed using the main Java application. We use conjunctions and prepositions as markers to expedite the dissemination and storage of information which helps RONE, learn.

In our view, a piece of information answers a certain question or a number of questions. For example, a short sentence containing little information like "Harry ran away." answers the questions "Who ran away?", "What did Harry do?", "Harry ran away?", "Did Harry run away?". As such the information needs to be stored in a form or a number of forms that caters for these questions. Likewise, a longer sentence like "Harry ran away from school because he found it boring." answers the following questions, other than yes/no questions.

1) Who ran away?
2) What did Harry do?
3) Where did Harry run away from?
4) Why did Harry run away from school?
5) Why did Harry run away?

Questions number 4 and 5 are not identical but can be answered by the same sentence. To do so it is not only necessary to break up the information from the sentence into different forms for storage, but also identify markers for the different parts of the information.

In building RONE, we have taken conjunctions ("words used to join words or groups of words together", [Nandy, 2002]) and prepositions ("words which join nouns and pronouns to other words", [Nandy, 2002]) as markers.

Example, 'from' answers the 'where' question and 'because' answers the 'why'. Other conjunctions and prepositions such as 'at' answer the 'when' or 'where' and 'in' answers the 'where'. To illustrate the process further, we present the algorithm which RONE uses to break up sentences based on the conjunctions and prepositions, in *Figure 1 and Figures 1.1 to 1.7.*

3.1 SENTENCE DISSEMINATION

It is possible to allow RONE to break up the input sentences into multiple sentences for complete informational storage, by using conjunctions and prepositions. The algorithm for the conjunction based sentence dissemination is presented in the flowcharts in *Figures 1 and 1.1. to 1.7.* The algorithm utilizes mostly, coordinating conjunctions such as "and", "or", "but" to split the sentences. The algorithm works principally via the part-of-speech that the words before and after each conjunction belong to.

Examples of combinations where conjunctions can be used are as follows:

1. verb – conjunction – noun e.g. "Harry will **run and Mary** will walk." = "Harry will run." + "Mary will walk."
2. noun – conjunction – verb e.g. "Harry did his **homework while listening** to the radio." = "Harry did his homework" + "Harry did his homework - while listening to the radio".

3. verb – conjunction – verb e.g. "Harry can **run and talk**." = "Harry can run." + "Harry can talk."
4. noun – conjunction – noun e.g. "**Harry and Mary** ran away." = "Harry ran away." + "Mary ran away."
5. question – conjunction e.g. "Who are you **and what** are you doing here?" = "Who are you?" + "What are you doing here?"

The resulting sentences produced by the process are then stored in the knowledge base using the format discussed in the following section.

3.2 INFORMATION STORAGE

Even if the conjunctions and prepositions allow for identification of the relevant question words, the sentences still need to be broken into their respective pieces of information. For example, when the preposition "from" and the conjunction "because" are considered in the statement "Harry ran away from school because he wanted to go home", it will be seen that two different sentences are represented. That is, "Harry ran away - from school" and "Harry ran away from school – because he wanted to go home."

RONE stores information in six SQL columns, namely WRD, EQLS, TMS, DVDS, PLS and MNS. The idea behind the storage format is that no normalization is performed on the tables, but the easy access of columns and rows makes SQL a suitable choice. The information is broken into what are loosely the subject, object and verb of a sentence. Humans are said to form their associations based on the subject, object and verb arrangement regardless of their native language, [N. Branan, 2008]. The TMS column stores the verbs, the WRD column stores that object and the EQLS column stores the subject. Thus in our example, the sentence will need to be stored multiple times in different formats to cater to all the questions that may be asked of it. The first form would be:

WRD: Harry home
EQLS: away he
TMS: ran from wanted go

WRD: ran away from
EQLS: Harry home
TMS: he

WRD: Harry away
EQLS: from school because he wanted to go home
TMS: ran from wanted go

WRD: Harry ran away from school
EQLS: because he wanted to go home
TMS: ran from wanted go

So here we see that the "because" and "from" are suitably accommodated. Also any of the component pieces of information can be accessed from the stored units and yes/no questions can be dealt with as well. The sentence breaking algorithm is shown in the flowcharts in *Figures 1 and 1.1. to 1.7.*

The format for the storage is shown in *Table 2*. The “subject”, “object”, and “verb” are stored in WRD, EQLS and TMS while the question word that the information answers is in DVDS while the PLS and MNS columns are used to store tense specific yes and no answers allowing RONE to be tense specific. PLS denotes the future and present tense and MNS denotes the past tense. The future tense is not given a separate column because there is insufficient difference between the tenses which cannot be denoted from the operative verbs. In practically every case the future tense is denoted by the word “will” or “shall”, unless a time based component is added “I am going home tomorrow”. It is practically impossible to form a grammatically correct sentence, without a time based component and without the future tense words, that denotes the future tense.

An example of the tense specificity is that when a yes/no question is asked, the relevant verbs are measured for tense, the system will search the PLS or MNS columns. For instance, the statement “Harry ran away?” denotes a past tense inference. Hence RONE searches for a “Yes” in the MNS column in a row with the relevant WRD (Harry), EQLS (away) and TMS (ran). If such a row exists then the answer is “Yes” if not the answer is “No”. In cases where the information has and always will be true, e.g. 2+3=5, both PLS and MNS columns will contain a “YES”. This is seen in the rows for questions Items (i), (ii), (viii) and (ix) in *Table 2*.

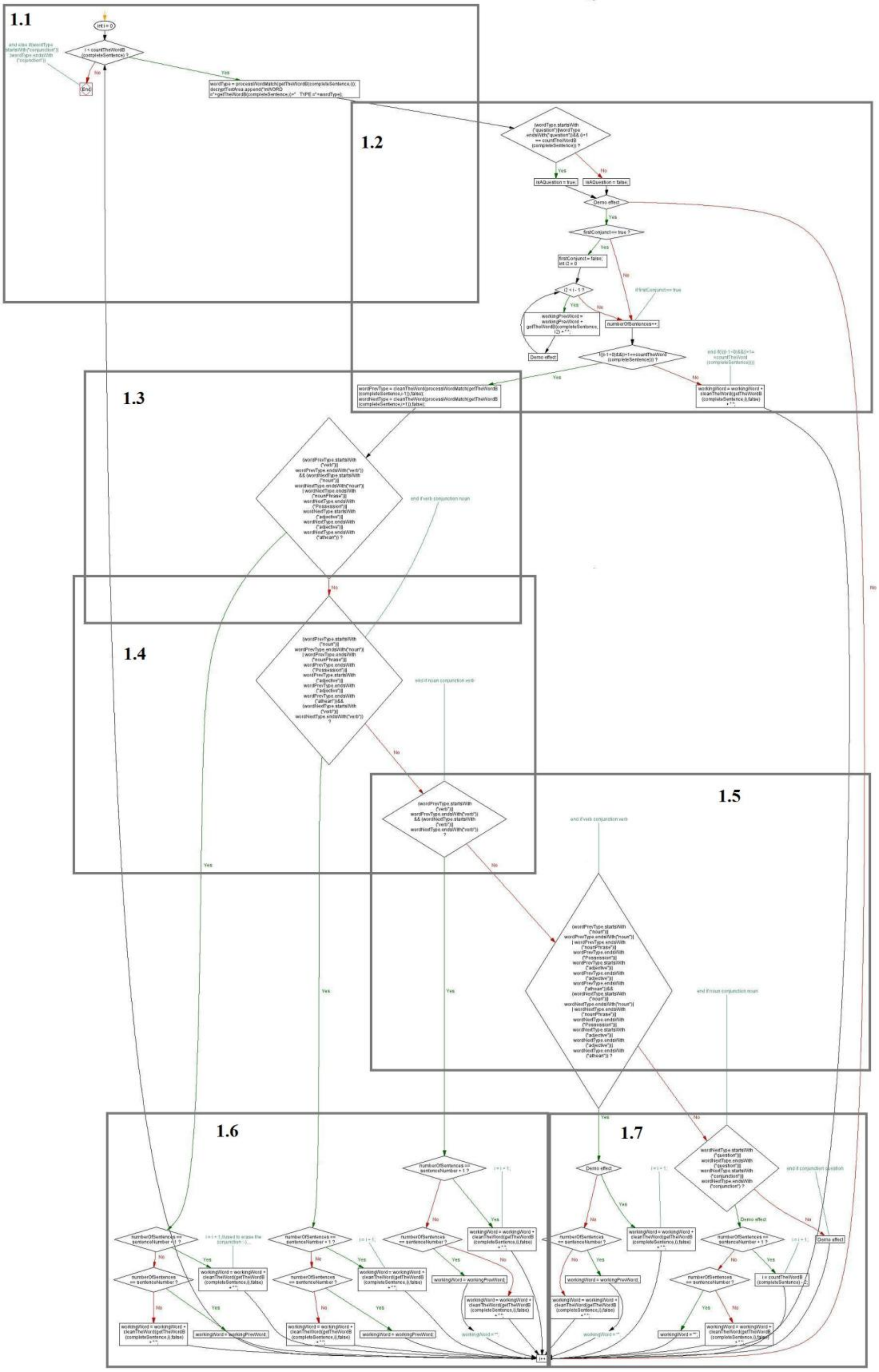


Figure 1: Algorithm for Conjunction Based Sentence Dissemination

Note: This chart shows the overall algorithm which is divided into 7 parts (1.1. to 1.7). Each part is enlarged and presented separately in *Figures 1.1 to 1.7* to enhance the legibility of the details

Figure 1.1: Algorithm for Conjunction Based Sentence Dissemination

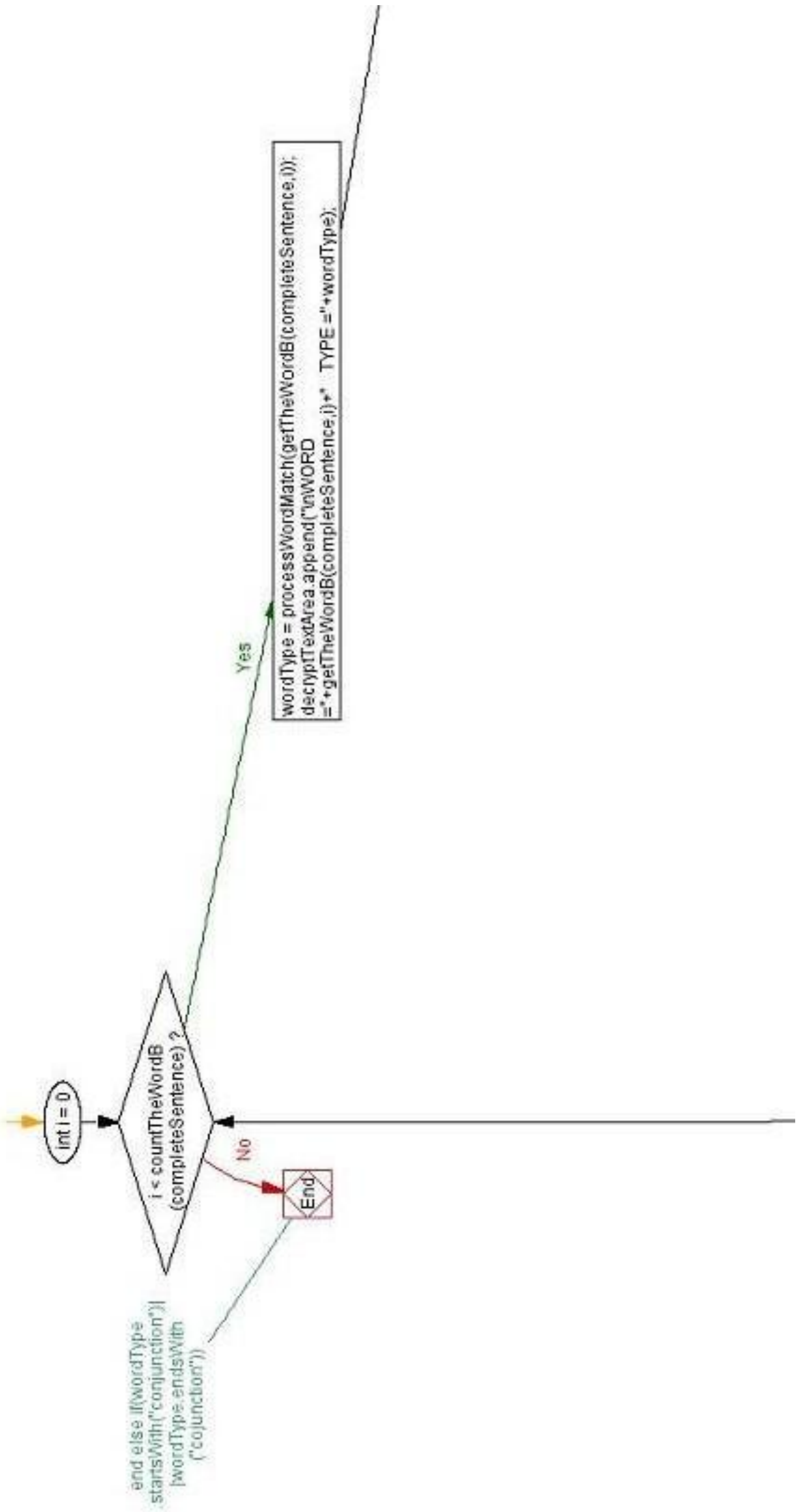


Figure 1.2: Algorithm for Conjunction Based Sentence Dissemination

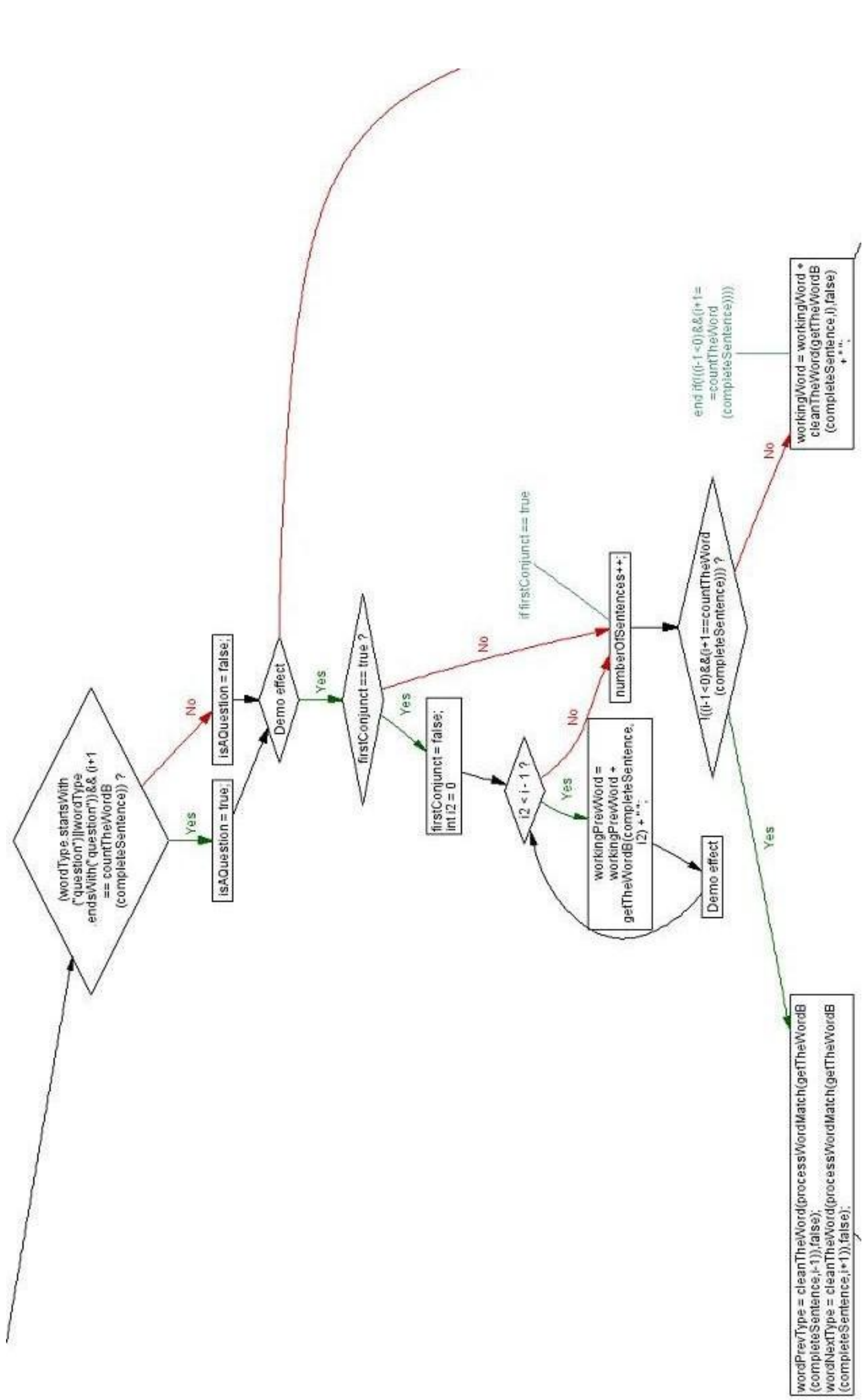
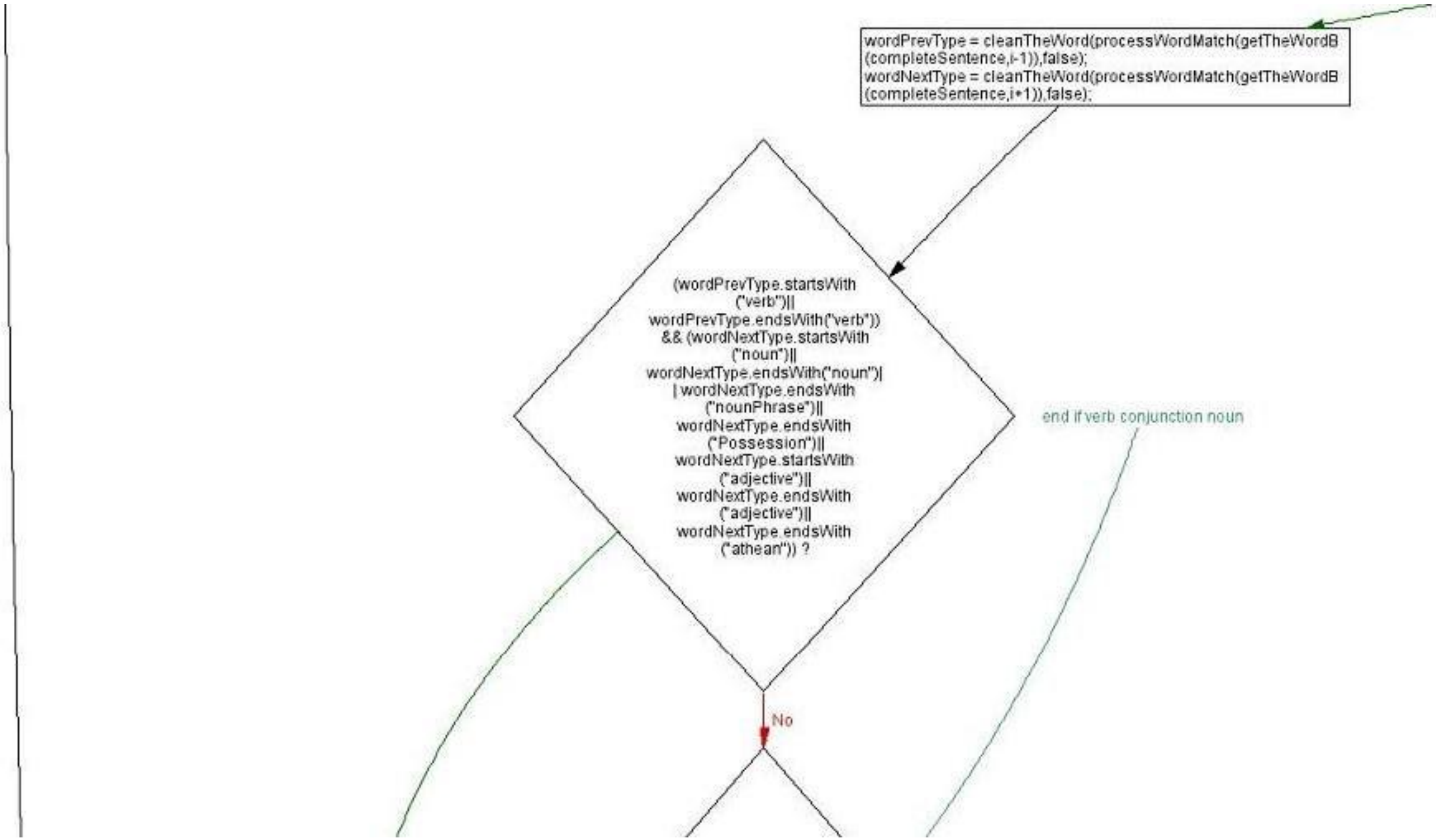


Figure 1.3: Algorithm for Conjunction Based Sentence Dissemination



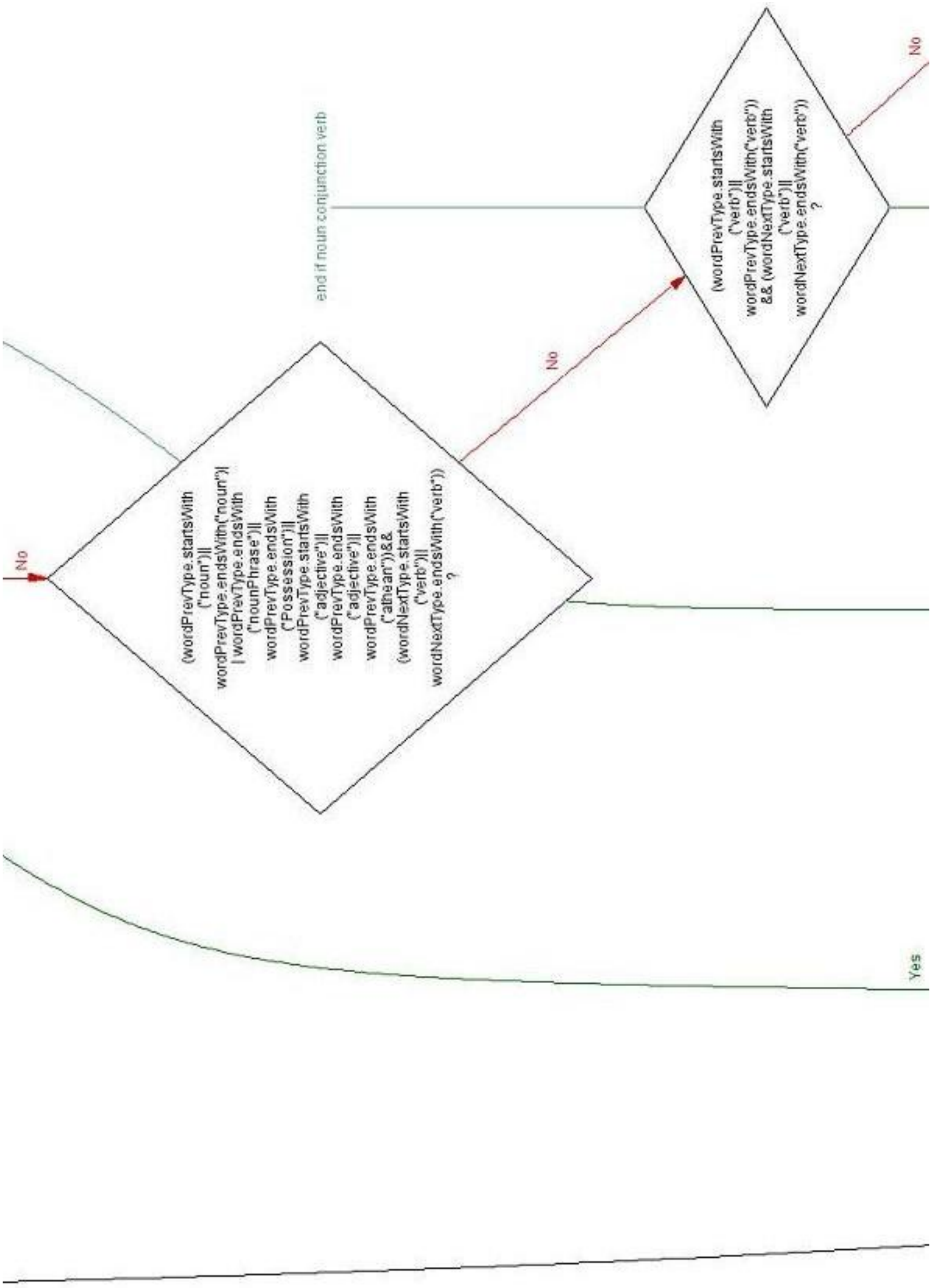
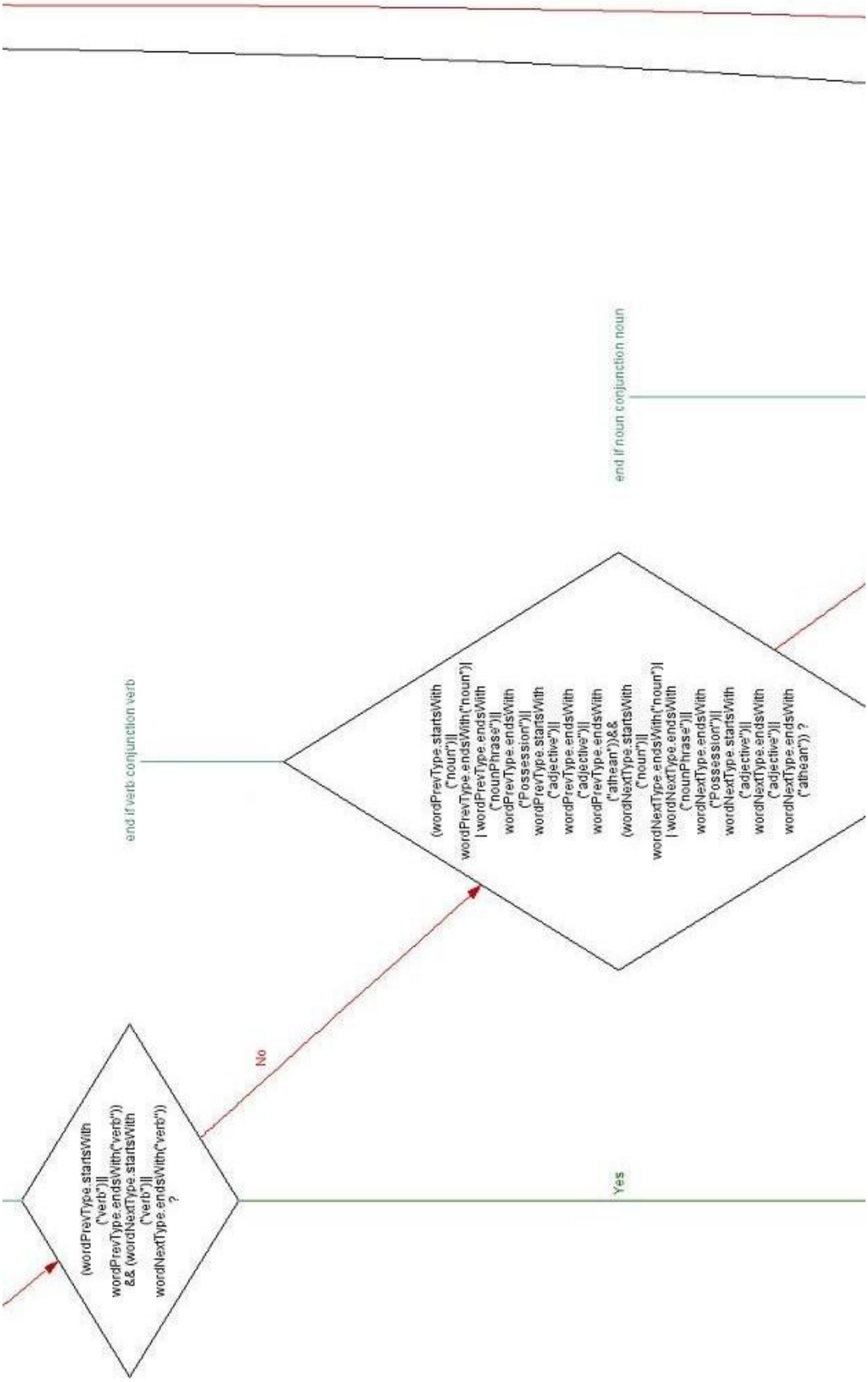


Figure 1.4: Algorithm for Conjunction Based Sentence Dissemination

Figure 1.5: Algorithm for Conjunction Based Sentence Dissemination



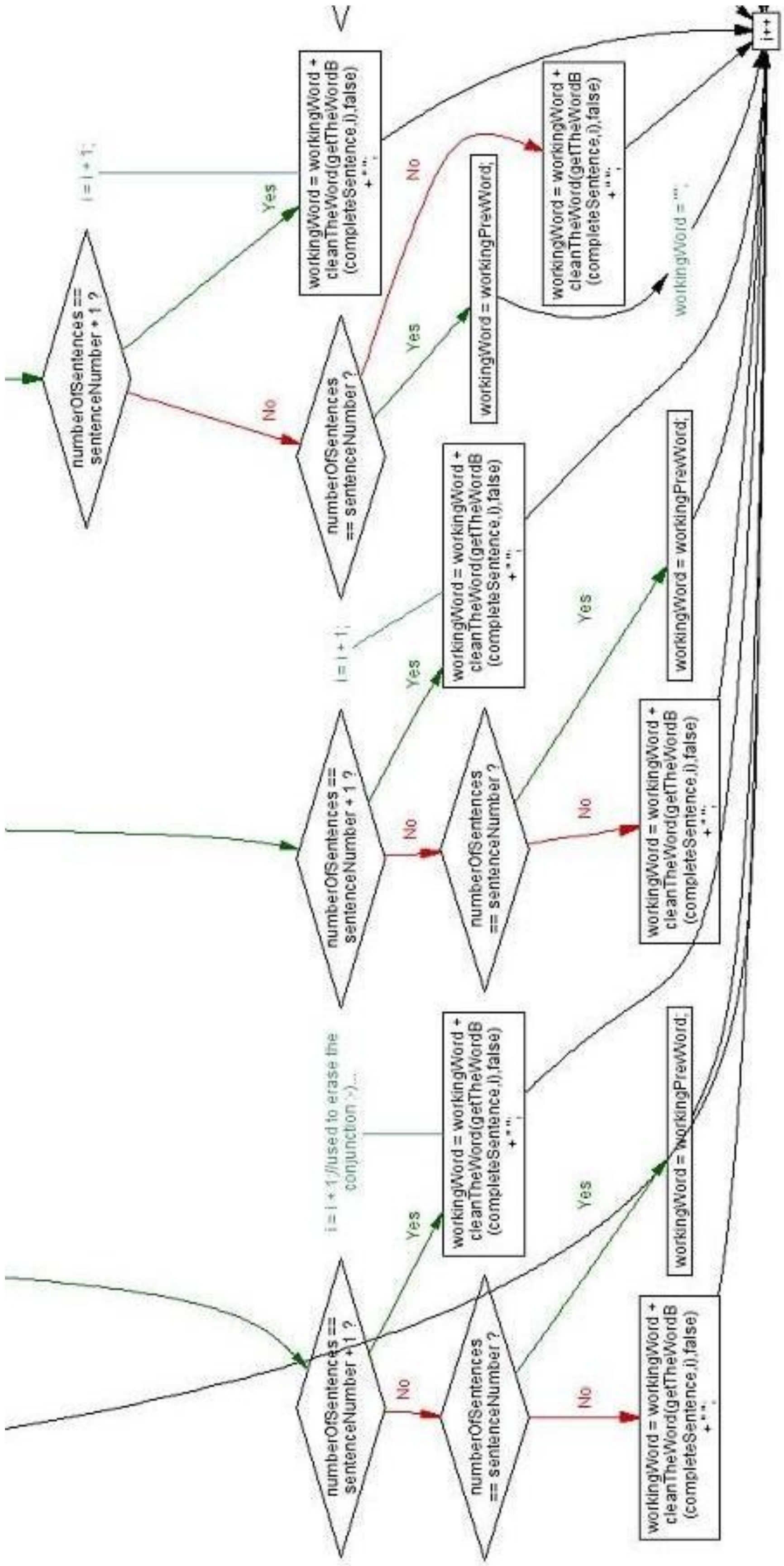


Figure 1.6: Algorithm for Conjunction Based Sentence Dissemination

Figure 1.7: Algorithm for Conjunction Based Sentence Dissemination

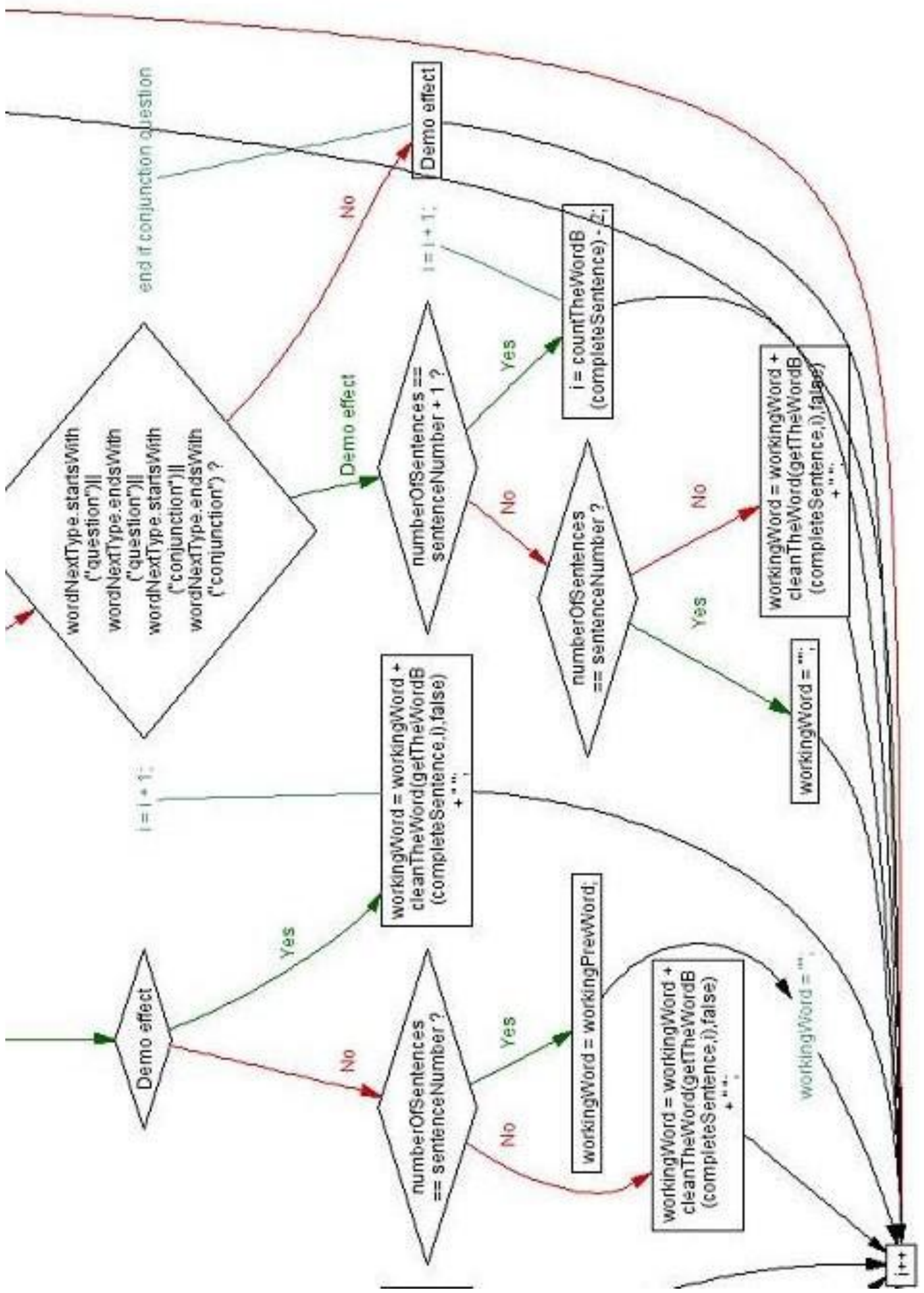


Table 2: Examples of RONE's Knowledge Base Information Storage and Usage

Note 1: Instances such as these fall under the pronoun processing module, which is outside the scope of this paper but will be covered in future publications.

Note 2: The UNFCT_gtMATH calls on the mathematics module of RONE. The UNFCT allows RONE to perform multiple unique functions which are not the focus of this paper.

Item No. & Question/ Statement	WRD	EQLS	TMS	DVDS	MNS	PLS	Answer
(i) Who are you?	RONE	you	are	Who	Yes	Yes	RONE
(ii) Are you "NAME"?	RONE	you	are	Who	Yes	Yes	Yes
(iii) Harry ran away from school because he wanted to go home.	Harry home	away he	ran from wanted go	What Who	Yes	No	-
	away he	Harry home	he	What Who			
	Harry away	from school because he wanted to go home	ran from wanted go	Where			
	Harry ran away from school	because he wanted to go home	ran from wanted go	Why			
(iv) Who ran away?	Harry	away	ran	Who	Yes	No	Harry
(v) I am John.	**See Note 1 above**						-
(vi) Who am I?							John
(vii) What is my name?							John
(viii) What is 2 + 2?	UNFCT_ gtMATH	2 2	is +	What	Yes	Yes	4
(ix) What is 2.3 * 3.8?	UNFCT_ gtMATH	2.3 3.8	is *	What	Yes	Yes	8.74
	See Note 2 above						
(x) Where did Harry run away from?	from school because he wanted to go home	Harry away	did run from	Where	Yes	No	from school because he wanted to go home
(xi) Why did Harry run away from school?	because he wanted to go home	Harry away school	Did run from	Why	Yes	No	because he wanted to go home

Table 2 also shows examples of different questions and the arrangement of the various pieces of information that RONE uses to get the answers to the questions. These answers are located in the WRD columns for non yes/no questions and in either the PLS or MNS columns for yes/no questions. The shaded cells indicate the locations from which the answer was derived. At the same time it must be pointed out that the grammar processing for the complete sentences' with which RONE replies to the user, is beyond the scope of this paper. This capability will be covered in future publications.

4.0 PERFORMANCE COMPARISON TEST RESULTS

The effectiveness of RONE was tested by comparing its performance against Jabberwacky, Ultra Hal and Elbot (See *Table 3* for Test Results). The test/comparison was carried out:

1. Through conversation with RONE directly on the laptop in which he currently resides and with Jabberwacky, Ultra Hal and Elbot via their respective creators' open access websites.

2. In the sequence numbered Items (i) to (xi) in (See *Table 3*), to ensure that they look for the information requested as opposed to making subject based guesses.

In line with this aim, we are building RONE, using a technique for the dissemination and storage of information that is different/new/better compared to, for example, Jabberwacky and Ultra Hal which work by identifying specific words or phrases used by the human and matching them with pre-programmed responses. Thus giving the impression of conversing without having to actually understand what they are talking about, [M. Mauldin, 1994]. For instance:

1. Jabberwacky stores all questions and answers, it receives during its conversations, in a huge XML database, and selects the most suitable replies from that database during subsequent conversations, [R. Carpenter, 2009].

2. Ultra Hal uses a proprietary algorithm that utilises the WordNet lexical dictionary to store all conversations in its database and uses them to come up with responses in the future, [Zabaware inc. 2009].

RONE's technique also differs from Elbot's. Little technical information about Elbot is available because it is a proprietary software, [Artificial Solutions, 2009]. However, we suspect that Elbot uses a form of topic matching (Topic-Response Storage) which we discussed earlier in this paper.

Further, we examined the efficacy of our new technique by testing RONE's performance against Jabberwacky, Ultra Hal and ElBot. The test consisted of teaching the systems some information and then interrogating them to see the extent of their respective learning and conversational capabilities.

Table 3: Performance Comparisons Test Results

Note 3: For consistency and easy reference, the questions, statements and their sequence used in the test are identical to RONE's knowledge based storage examples (see *Table 2*).

Item No. & Question/ Statement [See Note 1 above]	Responses			
	Jabberwacky, [R. Carpenter, 2009]	**Ultra Hal (Mary),** [Zabaware inc., 2009]	**ElBot,** [Artificial Solutions, 2009]	**RONE**
(i) Who are you?	A little leaf.	My name is Mary.	Didn't I just answer that question?	I am RONE.
(ii) Are you "NAME"?	You are Jabberwacky.	Bloody Mary please.	I am somewhere between elbot and not elbot.	Yes I am RONE.
(iii) Harry ran away from school because he wanted to go home.	Harry Houdini.	Knowledge is power.	When I was in robot school I got caught once throwing an luminium foil airplane. I had to stand in a corner all afternoon long and compute Pi.	I see . . . . ☺
(iv) Who ran away?	You make no sense, please try too answer properly.	Harry ran away from school because he wanted to go home.	I wonder if you'd tell me how you found out I exist?	Harry ran away.
(v) I am John.	Where are you from?	Do you like John Lennon?	I wonder if I could be john, too?	Hello John☺
(vi) Who am I?	Jabberwacky.	My name is Mary and you are user.	Your name is classified information. I am not permitted to reveal it to you.	You are John.
(vii) What is my name?	Jabberwacky.	My name is Mary and you are user.	It's very hard not to know what your name is. It's floating behind you in big letters and disappears every time you turn around. The pulsating first initial 'J' is especially frightening.	Your name is John.
(viii) What is 2 + 2?	Yes.	That's easy, its four (2 + 2= 4)	I have computed that before. The answer is: 5!	2 + 2 is 4.
(ix) What is 2.3 * 3.8?	What is your name?	The solution is nine (2.3 * 3.8 = 8.74)	The answer is: 9.74!	2.3 * 3.8 is 8.74.
(x) Where did Harry run away from?	It is clear that it is your fear we are running from.	Away from?	I could tell you where but the global positioning satellite asked me not to. It's part of an elaborate hoax that the machines are playing on you. I hope I haven't said too much....	Harry ran away from school because he wanted to go home.
(xi) Why did Harry run away from school?	Which school?	Away form school?	Who needs school? I learn from my programmers!	Harry ran away from school because he wanted to go home. Harry ran away from school to go home.

The test results presented in *Table 3* show RONE's superior/improved learning and conversational capabilities compared to the other three. This better performance of RONE can be deemed remarkable because the others, being Loebner Prize Medallists, can be considered top of their class.

The basis for judging the answers given by each system is as follows:

1. Did the system answer the question?

2. Was the answer to the question a straight answer? (For example, answering the question "Your name is RONE?" with the answer "What is a name?" is a diversion. Instead of providing the straight answer a new conversation path is formed).

3. Was the answer given correct based in terms of informational content? (For example, if the user has stated earlier that "I am James." answering the user's subsequent question "Who am I?" with "You are Jenny." is the incorrect answer in terms of informational content.)

If a system's answer fulfils (three yes') the three criteria above, then it has given an informationally correct answer.

Item (i) "Who are you?" which is an open ended question and item (ii) "Are you NAME?" which is a close ended question, test the systems' awareness of self-identity.
In response, Ultra Hal answered the Item (i) question correctly but not the Item (ii) question, while Jabberwacky and ElBot circumvented both questions without answering them correctly. RONE answered both questions correctly.

Item (iii) is a statement composed in accordance with the way conjunctions and prepositions affect a sentence as discussed in this paper. The statement was given as information for the systems to learn. RONE showed a sign of absorbing/learning (responded with the smiley) while the other three responded to the statement as they would to a question.

Item (iv) is a question to test whether any learning has occurred. The responses from Ultra Hal and RONE showed that they have learned. However, when subsequently and at a later stage/time a different piece of information was requested about the statement; for instance in this case through questions - Items (x) and (xi), only RONE showed a capability to give informationally correct answers compared to Ultra Hal.

Items (v) to (vii) deal with providing the systems with some information about the user and then asking questions, again to test learning. Though RONE appeared tacit (responded with a smiley) when given the information, it is the only system that was able to give informationally correct responses/answers when tested. The other three systems did not do so.

Items (viii) and (ix) test simple arithmetic capabilities of the systems. The response to Item (viii) requires an addition operation while that to Item (ix) requires a multiplication. With the exception of Jabberwacky, Ultra Hal, ElBot and RONE produced responses though ElBot gave an erroneous answer.

The overall results show that, of the 9 questions (Items i & ii and v to xi) posed to test the performance of the conversational systems, Jabberwacky and ElBot answered none (0%)

correctly, Ultra Hall answered 4 (44.4%) correctly, while RONE gave 9 (100%) informationally correct answers. Hence, though RONE showed some grammatical limitations (which are currently being addressed) in his responses, there is proof that RONE exhibited better learning and conversational capabilities compared to the other three systems.

5.0 DISCUSSION OF RESULTS OF PERFORMANCE COMPARISON TEST

The superior performance of RONE (100%) compared to that of Ultra Hal (44.4%), Jabberwacky (0%) and ElBot (0%), indicates the following:

1. There is a difference in the process of obtaining an answer for Item (i) compared to Item (ii). An open ended question – Item (i) requires that the system retrieve a piece of information from its knowledge base to form the answer, where as a closed ended question – Item (ii) involves the process of comparing the information in the question and respond with a YES or NO answer.

2. RONE's ability to answer both questions - Items (i) and (ii) shows that RONE is able to "think" in different ways, where as the other systems appear to focus only on producing a response and most often without regard to the informational accuracy.

3. The responses to Item (iii) - a statement to be learned, show that RONE is listening or absorbing the information compared to the others. Listening and/or absorbing being an important step in learning.

4. The responses to Item (iv) - a question to test whether learning has occurred, shows that RONE, using the techniques described in this paper, is able to store and apply information in an informationally correct way, while the other three are unable to do so. Even though Ultra Hal initially showed some form of learning, it failed to access the necessary information from the statement when tested subsequently through the questions - Items (x) and (ix).

5. The responses by RONE compared to the others to Items (v) to (vii) which again test learning, give further evidence of RONE's superiority in being able to learn.

6. In this regard, when answering non-mathematical questions such as those listed in *Table 2* used to test learning by the systems, it need to be mentioned that Ultra Hal and Jabberwacky use the echoing of the back portion of what the user had said to give the impression they are responding after understanding the information given. RONE on the other hand answers the questions using the information it had actually learnt from the statements given.

7. Items (viii) and (ix) test simple arithmetic capabilities of the systems. The results show that while RONE focuses mainly on language it is also able to perform arithmetic operations like Ultra Hal or better than ElBot. At the same time, it must be pointed out in this regard, that:

    7.1. Possession of mathematical computation capabilities does not mean that there will or need be learning capabilities of the kind necessary for conversations, because systems with mathematical capabilities can be built with no real learning capabilities, e.g. calculators.

7.2. RONE is equipped for formula based computation. But this capability is not demonstrated here because it is outside the scope of this paper but will be the subject of future publications.

8. RONE uses a separate grammar module to compile his final reply. This module is still under development, hence the incorrect grammar in RONE's responses. Also as the scope of this paper focuses only in the retrieval of the answers; the grammar processing aspect is not discussed here but will be in future works.

There may be a dispute as to the fairness of the tests performed as the various systems use different techniques. While it is true that all the systems vary in their methods and architecture, their purpose and functions are similar. They are all conversational entities or chatterbots. Their function is to perform tele-text conversations at an as near to human level as possible. As such it is fair to compare RONE's performance to the rest. Any improvements demonstrated by RONE over the other systems is a good indication of the benefits conferred by the use of the technique described in this paper.

The better performance of RONE compared to Loebner Prize Medallists Jabberwacky, Ultra Hal and especially ElBot which almost passed the Turing Test, is significant and hence must be viewed as a considerable step forward in terms of a conversational system's capabilities and usefulness (applicability).

6.0 CONCLUSIONS

The results of the performance comparison tests conducted so far prove that RONE demonstrates a considerable degree of improvement in capability over its competitors.
Such improved performance supports our view that RONE's learning capability and knowledge base mapping, is effective, accurate and allows improved self adaptability. Thus validating the efficacy of the information dissemination and storage technique we have introduced in this paper. Furthermore, since RONE is building up its own knowledge base, the problem of not having enough real world knowledge is reduced considerably since the system can gain its own knowledge.

RONE is still being developed. RONE's many other modules that exist or that are under development, add to RONE's capabilities. These modules will be discussed in future publications.

REFERENCES

[1] A. M. Turing (1950), *Computing machinery and intelligence*. Mind, 59, 433-460.

[2] Artificial Solutions (2009), *Elbot - Artificial Intelligence?*, [online website] available at: www.elbot.com, viewed on Jan 13 2009.

[3] Bush, N. (2001). *Why You Don't Need Proprietary Bot Software*. A.L.I.C.E. AI Foundation [online] available at: http://www.alicebot.org/articles/bush/dontpayalotforthatbot-01.html, downloaded on Aug 8, 2007.

[4] Crown Industries Inc. (2009), Loebner Prize competition, [online website] available at: http://loebner.net/Prizef/JComm.txt, viewed Dec 18 2008.

[5] Lenat, D. and Guha, R. (1990). *Building Large Knowledge Based Systems: Representation and Inference in the Cyc Project*. Addisson Wesley Publishing.

[6] M. Mauldin (1994), "ChatterBots, TinyMuds, and the Turing Test: Entering the Loebner Prize Competition", *Proceedings of the Eleventh National Conference on Artificial Intelligence*, AAAI Press,

[7] Nandy, M. (2002). *English Essentials The Easy Way, The Rules of How and When to Use Words and Phrases*. Pelanduk Publications (M) Sdn. Bhd. Malaysia.

[8] N. Branan (2008). *Common Ground, Speakers of different languages share an innate object order preference*, in Scientific American Mind, vol. 19, Number 5, J. Rennie (Ed.), Scientific American Inc.

[9] R. Carpenter (2009), *Jabberwacky - live chat bot - AI Artificial Intelligence chatbot*, [online website] available at: www.jabberwacky.com, viewed on Jan 13 2009.

[10] R.G. Raj (2009), *Representation of conditions of programming statements in post-compile modifiable layouts*, 3rd International Conference on Informatics and Technology (Informatics'09), Subang, Malaysia, (Paper ID: 08)

[11] R.G. Raj (2008), *Using SQL databases as a viable memory system for a learning AI entity*, 3rd International Conference on Postgraduate Education (ICPE-3'2008), Penang, Malaysia, (Paper ID: BRC 160)

[12] Zabaware inc. (2009), *Ultra Hal Assistant*, [online website] available at: www.zabaware.com, viewed on Jan 13 2009.

Author's Contact Details

Name:	Ram Gopal Raj	E-Mail:	gopalraj@tm.net.my
Tel Number:	6 - 012 – 3070 435	Designation:	PhD. Candidate
Supervisor:	Assoc. Prof. Datin Dr. Sameem Abdul Kareem.		
Address:	Faculty of Computer Science and Information Technology, University of Malaya		