A Question Answer Approach to Building Semantic Memory

Basawaraj[1], Janusz A. Starzyk[1,2] and Marek Jaszuk[2]

[1]School of Electrical Engineering and Computer Science, Ohio University, Athens OH, USA
{Basawaraj.Basawaraj.1, starzykj}@ohio.edu
[2]University of Information Technology and Management, Rzeszow, Poland.
mjaszuk@wsiz.rzeszow.pl

Abstract. Semantic memory is an integral part of intelligent systems dealing with natural language processing (NLP). Building these memories is a challenging task. Different approaches have been proposed and tested, using a variety of corpuses. The corpora used to build the semantic memories vary from well structured to highly unstructured. The more structured a corpus, the easier it is to build a semantic memory using it. This is because a structured corpus delivers the NLP system more knowledge about the language and its grammar. In this paper we show how a question answering based approach can be used in learning of concepts and building the semantic memory.

Keywords: semantic memory; concepts; question answering.

1 Introduction

Interaction between humans and computers has been researched since the early days of modern computers, with the goal being to use natural language for such interactions. An intelligent agent should be able to both understand and generate meaningful statements in natural language to successfully interact with humans. REQUEST and PLANES were some of the earlier successful attempts. REQUEST, a question answering (QA) system developed in the mid 1970's, employed grammar based analysis and could answer questions posed in plain, but restricted, English using a formatted database [1]. Similarly, PLANES was another QA system that could answer questions posed in English using a relational database consisting of aircraft maintenance and flight information data [2]. QA systems have come a long way since then. Modern QA systems, using information retrieval (IR) and information extraction (IE) techniques such as those shown in [3] and [4] have been very successful at answering questions posed in natural language. QA systems such as [5], [6] have been successful in answering factoid questions but have not been shown to succeed at answering questions requiring contextual analysis. Some researchers, want to improve the performance of search techniques by helping users better formulate their queries by providing a list of possible questions related to the topic being searched and thus guiding them towards fulfilling their need [7]. For a review of the state of the art in QA refer to [8]. [9] provides a very good analysis of the problems with the present direction of QA research. For a QA system to be able to succeed in answering questions on complex context, something that humans are capable of, its working should be grounded in cognitive principles.

Memory is central to human cognition and semantic memory plays a major role in it. Semantic memory stores the knowledge about the world and learned relationships [10]. Concepts are created, managed, and related to one another through activation of the semantic memory. Relations between them are context dependent. Concepts and their relations play a major role in human cognition and their absence would severely handicap us. In this work we propose a simple approach that uses question answering to build a semantic memory.

2 Question Answering Approach to Knowledge Acquisition

The approach in most NLP systems has been to use statistical processing to solve the problems faced. This is not the approach that humans take. Humans solve problems based on the world knowledge that they have acquired during their lifetime. There is a very good, though not complete, understanding about how humans acquire language skills. This understanding of human language skills should be used by NLP systems in hope to achieve context understanding of the human speech.

NLP systems use large databases and use them to build their semantic memories. Humans do not do so; they start from forming simple relations and then build on them as they acquire more knowledge. A child does not start building its semantic memory by reading an encyclopedia; it does it by observing others and by asking or answering simple questions. It uses such interactions to learn relationships and form concepts, for example it learns that the pronouns *she* and *he* refer to the names of a female and male respectively. The proposed approach uses a similar approach to building semantic memory. Our system has access to a database that it can use to answer the questions posed to it, but building the semantic memory and learning the grammar of the language is done while answering questions. This approach was first suggested in [11].

Though most NLP systems tend to use a semantic memory that is build by statistically processing a large corpus of text, this need not be so. [12] shows an approach in which knowledge is acquired and semantic memory grows through dialogues or QA. In this case the system can not only ask questions but also verifies information before writing it to the memory.

3 System Organization

The organization of the QA system is shown in figure 1. The system consists of: a) Input Unit, b) Output Unit, c) Dataset, d) Working Memory, e) Semantic Memory, and f) Episodic Memory.

The **input** and **output units** are used by the user to interact with the system. The **dataset** consists of the documents that are to be used to find the answers. The

working memory is sequential in nature. It reads the user inputs and documents in the dataset. It converts the documents into paragraphs, sentences and finally into words or phrases. These words or phrases then activate nodes in the semantic memory that represent them. It can also read in the user feedback for a generated answer. The user feedback is a reward signal that is used to train/improve the system. The working of this is explained later.

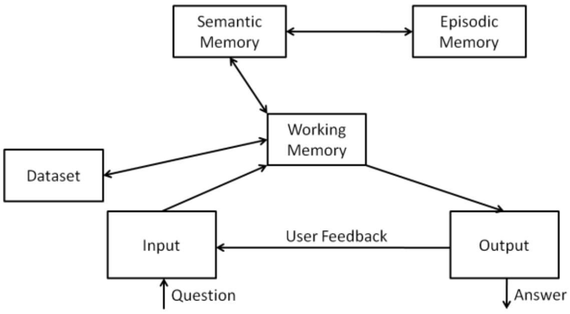

Fig. 1. The System Organization

When the documents are read by the working memory, the system extracts the words from the file and activates the corresponding nodes in the semantic memory. Fig. 2 shows the different stages that the working memory goes through when the system is dividing the document read from the dataset into words. F**ig. 2** shows the example of a single paragraph in a document, created for testing the algorithm, but it can be extended to documents of any size.

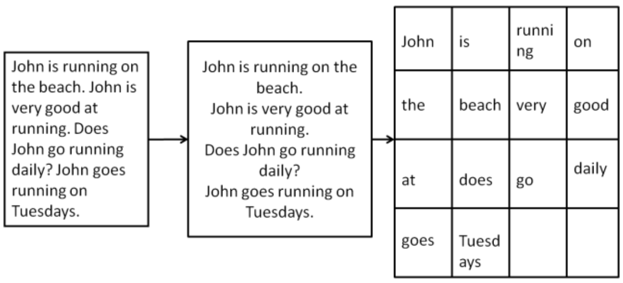

Fig. 2. Document being broken into words in the working memory

The **semantic memory** and the **episodic memory** consist of Long-Term Memory (LTM) cells [13] capable of remembering sequences of vectors.

There are four types of nodes in the semantic memory, shown in Fig. 3. The first kind are the sensory nodes representing the basic sensory activation, i.e., characters in the alphabet that are activated only when the particular character is received on the input. The second kind are the sequence nodes that are activated only if all their inputs are activated in the correct order. They are illustrated by word or sentence nodes. The third kind are those nodes that have no requirement about the order of

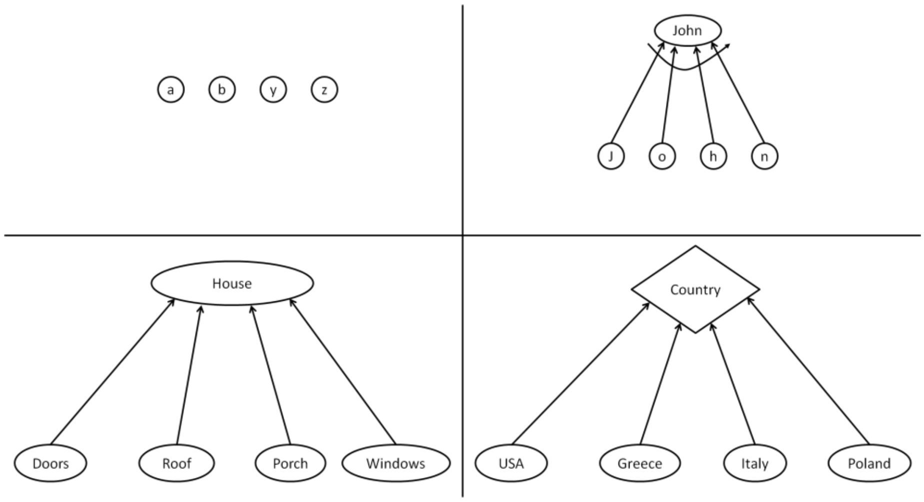

Fig. 3. Four types of nodes. Clockwise from top-left: basic characters, sequence, set and concept nodes.

their inputs and are activated if all or most of their inputs are activated. They are called set nodes and are winner-take-all (WTA) nodes. The nodes of the last kind are activated if any one of their input is activated. They are called concept nodes. The sentence nodes do not store the actual sentence; they only store the locations of the words (in the order in which they occur) that constitute the sentence. Thus all nodes in the semantic network can be activated depending on activities of their inputs and rules for activation specified by the type of node. The level of activation of the nodes is calculated in accordance with the LTM recognition mechanism discussed in [13], with modifications to account for the three types of nodes discussed here.

4 Building Semantic Memory

The proposed approach is based on understanding of learning in humans. That is, we start with learning simple sentences and move to complex sentences whose meaning depends on context. Consider the scenario where the system has no prior knowledge, except for the paragraph read in Fig. 2, and the following question is asked: "*Where is John?*". The system generates an answer based on the activations that the question leads to in the semantic and episodic memories. The working memory breaks down the question into the words {*where*, *is*, *John*} and activates *is* and *John* nodes in the semantic memory. The nodes are activated considering the order of inputs (in case of word and sentence nodes) and the number of inputs according to the LTM recognition mechanism discussed in [13]. The organization of the paragraph from Fig. 2 in the semantic memory with the activations[1] (in brackets) due to this is shown in Fig. 4.

[1] Activations shown in the figure are simplified for representational purpose.

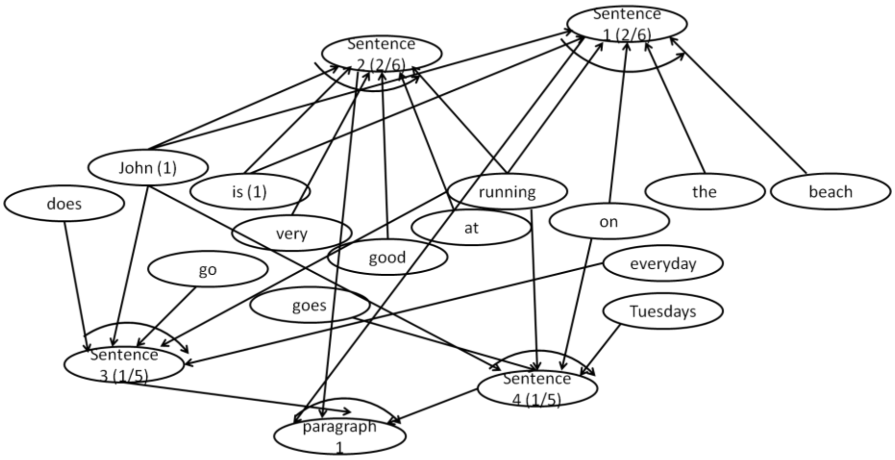

Fig. 4. Nodes and their activations

As, at this point, the system has no prior knowledge. It looks for the sentence that has the highest activation and returns this sentence as the answer[2]. If more than one sentence has the same activation, as in this case, the system gives one of them as the answer. Assume it gives sentence 1, "*John is running on the beach*", as the answer. Now the user has the option to accept the answer, ask for another, accept the answer but with modifications (the corrected answer has to be given to the system) or give the answer. The users can provide a reward signal if they desire. If the users feel that the first answer is false or not appropriate they can ask for the second best answer. If they feel that even the second best answer is incorrect or not appropriate then the system expects them to provide the answer.

In the present case assume that the user accepts the answer, but with modifications: "*John is on the beach*". The system, using this question-answer pair, builds its knowledge and generalizes it as follows:

1. Analyze this question answer pair: (shown in Fig. 5)

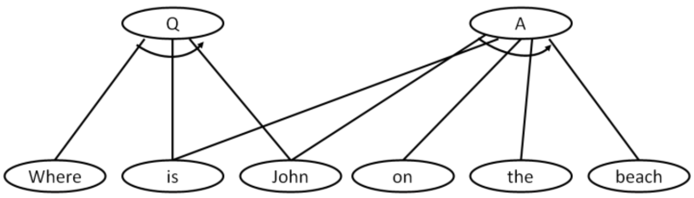

Fig. 5. Question Answer (QA) pair

   a. Find parts / words that are common to both question and answer: {John, is}, and link them to both question and answer.

[2] For simplification have shown sentence activation equal to number of activated inputs.

b. Find parts that are different and form a relation ('R' in Fig. 6):{where | "on the beach"}.

In Fig. 6, 'S' is a sequence node and is activated only if all its inputs: {on, the, beach}, are activated. The 'R' nodes are used to show a relationship between a question and its related answer (e.g., *where - place*).

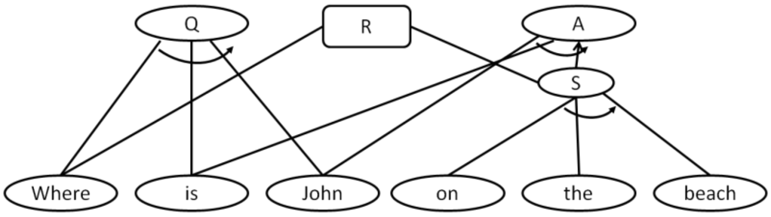

Fig. 6. Relations and categories from the QA pair

2. Form a general relation about the format of the questions and answers for future use {"where *is* ***something***" -> "***something*** *is* ***somewhere***"}. This means that whenever the system is asked a question starting with "where is" followed by a noun, it knows that the answer starts with the noun (***something***) followed by *is* and a place phrase (***somewhere***).
3. To make use of general relation obtained in 2 in even broader sense we first find the sentence $\mathbf{S_A}$ with the highest activation when a correct answer **A** is played. Next we align $\mathbf{S_A}$ and **A**, determine their matching parts and find differences. The categories of differences show acceptable deviation from previously learned answer format. For instance if,

   A = "John is in the park", and

   S_A = "My brother John is playing in the park with his friends",

then the system learns that for every question of the type "where *is* ***something***" the sentence containing the answer can have additional words in it, i.e., the sentence can be of the type: "*some phrase* ***something*** *is some phrase* ***somewhere*** *some phrase*", and the acceptable format of the answer contains: ***something*** followed by *is* and there is a place phrase (***somewhere***) at the end. Only the simplest case of the answer within a single sentence has been shown here. The analysis, however, can be extended to answer spreading over multiple sentences, paragraphs etc.

Now if a new question of the same type, say "Where is Jim?" is asked, the system searches for sentence containing the words {Jim, is} in the required order to find the best answer. If more than one sentence has them, then it looks for the best match based on its knowledge. Assume that there are two sentences: a) "Jim is sleeping", and b) "Jim is in the park", that matches this criterion. In such cases the system uses its knowledge (about the relation: {"where" | "on the beach"}) and finds that the previously seen relation is more similar to the second sentence and thus gives it as the answer.

The system now finds relations between the new question answer pair and appropriately modifies the sequence node 'S' (Fig. 7). Hence the system learns that its inputs can be concepts, node ‘C’ in Fig. 7. The concept nodes are activated if any one of their inputs is activated.

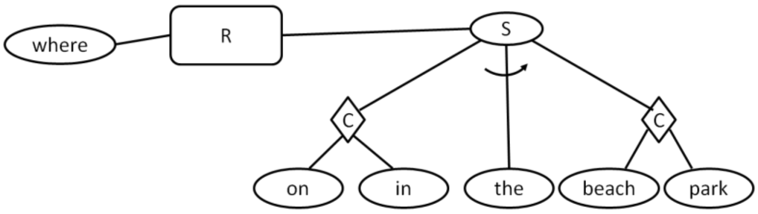

Fig. 7. Relations and categories from additional QA pairs

Fig. 7 shows that by answering two simple questions of the type: *where is* ***somebody****?*, the system is able to form the concept of *place*, learn the relation: {*where* | *place*}, and also the grammatically correct form of referring to ***somebody is*** in a *place*. Similarly to the above relation the system learns a variety of concepts, correct grammatical structures and relations in addition to the different types of questions that it can answer.

The ‘R’ nodes introduced in Fig. 6 are used to specify a variety of relations and their interpretation changes accordingly. The connections between ‘R’ nodes and the concepts that they are connected to are bidirectional. Thus activation on any one of the node connected to an ‘R’ node will affect all other nodes connected to the ‘R’ node. Thus in Fig. 6 if the concept *where* is activated it activates the concept *place* (*on the beach* in the above example). These ‘R’ nodes are also used to connect ‘synonyms’ to show equivalence relationship, thus if one of the concepts is activated (or inhibited) all its synonyms are activated (or inhibited). Similarly, they are also used to connect ‘antonyms’ (using inhibition links) so if one concept is activated its antonyms are inhibited.

5 Discussion and Future Work

A well structured semantic memory is necessary for systems to succeed in NLP tasks, but building associative semantic memory that can consistently provide unambiguous answers to user’s questions is difficult. The proposed semantic memory consists only of concept nodes and the links between them. Using these simple elements, the system is able to represent the knowledge it has gained through question answering. Since the method is very general, it can be applied to any language. Consider the example: *dog chases cat*, a simple and grammatically correct sentence in English. But this form: *noun verb noun*, would be considered grammatically incorrect in many other languages. As the proposed method uses examples to learn the grammar of a language, it is very flexible, and can be applied to any language in open learning environment without specific rule based grammar.

The proposed method is capable of creating the semantic memory structure and algorithm to retrieve associated information from the memory given a query input, so one can use everyday examples to build the memory. The proposed approach can be used to train the system to learn both formal and colloquial form of a language.

We will use this work to test the datasets [14] from now discontinued QA Track from the annual TREC conference and compare the accuracy of our method to those of others. We are also working on extensions that could allow the proposed method to solve the four challenges presented by Jackendoff and discussed in [15]. Our method is able to create and modify relations between different concepts based on the questions, and their answers. Thus, if suitably trained, this approach can be used to retrieve documents and extract information from multi-lingual sources in addition to translating them.

6 References

1. Plath, W.J.: Request – Natural Language Question-Answering System. IBM Journal of Research and Development. 20, 326 – 335 (1976)
2. Waltz, D.L.: An English language question answering system for a large relational database. Commun. ACM. 21, 526 – 539 (1978)
3. Robertson, S.E., et al.: Okapi at TREC-3. In: Third Text REtrieval Conference (TREC-3), pp. 109 – 126, http://trec.nist.gov/pubs/trec3/t3_proceedings.html
4. Buckley, C., et al.: Using clustering and super concepts within SMART: TREC 6. In: Sixth Text Retrieval Conference (TREC-6), pp. 107 – 124, http://trec.nist.gov/pubs/trec6/t6_proceedings.html
5. Katz, B., Lin, J.: REXTOR: A System for Generating Relations from Natural Language. In: ACL 2000 Workshop of Natural Language Processing and Information Retrieval (NLP&IR), http://groups.csail.mit.edu/infolab/publications/Katz-Lin-ACL00.pdf
6. Bill, E., Dumais, S., Banko, M.: An Analysis of the AskMSR Question-Answering System. In: Conf. Empirical Methods in Natural Language Processing (EMNLP ‘02), pp. 257 – 264, http://research.microsoft.com/en-us/um/people/sdumais/EMNLP_Final.pdf
7. Kotov, A., Zhai, C.: Towards natural question guided search. In World Wide Conference (WWW’2010), pp. 541 – 550, http://sifaka.cs.uiuc.edu/czhai/pub/www10-quse.pdf
8. Mittal, S., Mittal, A.: Versatile Question Answering Systems: Seeing in Synthesis. Int. J. Intelligent Information and Database Systems. 5, 119 – 142 (2011)
9. Ferrucci, D., et. al.: Towards the Open Advancement of Question Answering Systems. IBM Research Report RC24789 (2009)
10. Tulving, E.: Organization of Memory: Quo Vadis?. In: Gazzaniga, M.S. (Ed.) The Cognitive Neurosciences, pp. 753 – 847. The MIT Press, Cambridge, MA (1995)
11. Starzyk, J.A., Basawaraj.: Self-Organizing Neural Network for Question Answering. In: 13th Intl Conf on Cognitive and Neural Systems (ICCNS), Boston University, Boston (2009)
12. Szymanski, J., Duch, W.: Semantic Memory Knowledge Acquisition Through Active Dialogues. In: 20th Intl Joint Conf Neural Networks (IJCNN), pp. 536 – 541. IEEE Press, New York (2007)
13. Starzyk, J.A., He, H.: Spatio-Temporal Memories for Machine Learning: A Long-Term Memory Organization. IEEE Transactions on Neural Networks. 20, pp. 768 – 780 (2009)
14. Text REtrieval Conference (TREC), http://trec.nist.gov/
15. van der Velde, F., de Kamps, M.: Neural blackboard architectures of combinatorial structures in cognition. Behavioral and Brain Sciences. 29, pp. 37 – 108 (2006)