

# Recurrent Neural Network Classifier for Three Layer Conceptual Network and Performance Evaluation

Md. Khalilur Rhaman

Department of Computer Science and Engineering, BRAC University, 66 Mohakhali, Dhaka-1212, Bangladesh.

Email: khalilur@bracu.ac.bd

Tsutomu Endo

Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka, 820-8502, Japan.

Email: endo@pluto.ai.kyutech.ac.jp

**Abstract**—Natural language has traditionally been handled using symbolic computation and recursive processes. Classification of natural language by using neural network is a hard problem. Past few years several recurrent neural network (RNN) architectures have emerged which have been used for several smaller natural language problems. In this paper, we adopt Elman RNN classifier for disease classification for a doctor patient-dialog system. We find that the Elman RNN is able to find a representation for natural language. Contextual analysis in dialog is also a major problem. A three layers memory structure was adopted to address the challenge which we referred to as "Three Layer Conceptual Network" (TLCN). This highly efficient network simulates the human brain by discourse information. An extended case structure framework is used to represent the knowledge. We used the same case frame structure to train and examine the RNN classifier. This system prototype is based on doctor-patients dialogs. The over all system performance achieved 84% accuracy. Disease identification accuracy depends on number of disease and number of utterances. The performance evaluation is also discussed in this paper.

**Index Terms**—Three Layer Conceptual Network, Knowledge Representation, Recurrent Neural Network.

## I. INTRODUCTION

In this paper we present a Neural Network Classifier to classify the diseases for our system prototype. ELMAN [1] [2] POLLACK [3] and LAWRENCE [4] are considered to apply in this application. In this initiative we introduced Neural Network in Three Layer Conceptual Network (TLCN). To implement this novelty a number of researches was studied [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] which integrated RNN with NLP. TLCN architecture was developed by RHAMAN [16] from the memory model idea of NOMURA [17]. NOMURA's model was a framework for a memory management but RHAMAN developed architecture to address a real time dialog system. A discourse algorithm was introduced to represent discourse information which was related to [18], [19], [20]. It was a model which could simulate the

human brain memory. Human brain consist three types of memories Long-term, short-term and mid-term which was studied in [21], [22] and [23]. In our system episodic memory module simulates short-term memory, discourse memory module simulates mid-term memory and ground memory module simulates long-term memory. But in case of disease identification the system performance was not satisfactory. So, We use RNN Classifier and get better result. We used the traditional modules for input and response generation but developed an extended case frame model of BRUCE [24] and SHIMAZU & NOMURA [25] for handling the knowledge database. Case frame network is very useful for unstructured and ambiguous languages like Japanese. However, most systems deal with rather simple sentences, in which the analysis is a relatively easy task. Our extension of case frame architecture reduced the complexity using formalism. We use the same structure for RNN classifier.

## II. MOTIVATION

Patient come back to the doctor repeatedly with the same symptoms and expects the doctor to fix it. In another case most of patients come to the doctor with same symptoms. In the decade of 1980 and 1990, a number of medical expert systems were designed using medical knowledge to diagnose the diseases. Medical expert systems have evolved to provide physicians with both structured questions and structured responses within medical domains of specialized knowledge or experience. Most of these systems [26], [27], [15] and [28] were based on doctor. Only medical experts could enter the predefined data set to find the solution. Our proposed system is very different in this point. Our system is designed to communicate with patient directly. Patients can communicate our system by their natural language. Natural language has traditionally been handled using symbolic computation and recursive process. The most successful stochastic models have been based on finite-state descriptions models. However finite-state model can

not represent hierarchical structures found in natural language. In past few decades most of the NLP systems [29], [30], [31] [32], [33], [34], [35]and [36] were developed based on syntactic phrase structure. These systems are good for structured language but not good for encoding linguistic information of unstructured language like Japanese. So, our motivation was to implement a system that can communicate directly with patients and also can handle the process of sequential sentences and complex sentence.

### III. OVERVIEW OF TLCN

Three Layer Conceptual Network model is a framework to use the knowledge that simulates human brain. Discourse information is playing the vital role for this kind of simulation. It consists of three layered memory modules: 1) Ground Memory Module (GMM), 2) Discourse Memory Module (DMM) and 3) Episodic Memory Module (EMM). In our proposed model, the knowledge and their relationships accommodate into the GMM. DMM represents the most basic, useful and recently used discourse knowledge. EMM is a User interface. It contains linguistic and non-linguistic knowledge for understanding input dialogs and generates response dialogs. However, memory modules are not simply an accumulation of the contents of other memories. It is a structured memory coherently organized by merging old and new knowledge. The process relates to what is called knowledge learning. Each memory module consists of a knowledge database and its manager. Fig. 1 depicts the simple process flow of TLCN.

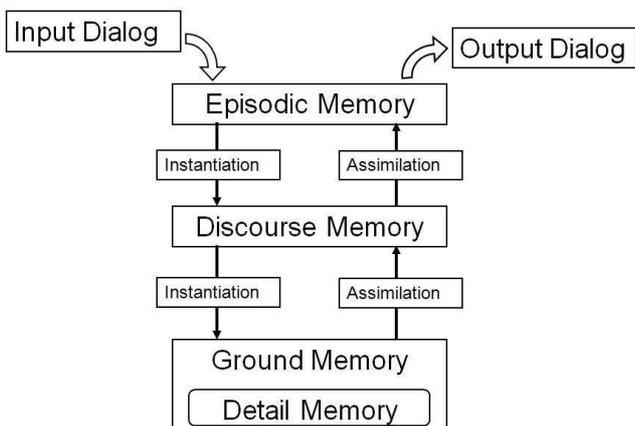


Figure 1. Processing of TLCN

#### A. Architecture

The TLCN-Dialog Processor (DP) architecture is illustrated in Fig. 2. This architecture provides an efficient integration of component modules. Text input utterances are sent to dialog understanding unit of EMM. To understand the input dialog, the entire linguistic knowledge is assimilated to episodic memory from ground memory in advance. After normalizing the input text, it generates case frames that are sent to classifier unit. Here, input

case frames are classified into some predefined classes. DMM manager receive these classified case frames from EMM manager and searches in the discourse memory to match some diseases. DM will be acknowledged whether any disease is identified or not. If the information does not match enough in discourse memory, DM sends the input case frames to GMM manager. GMM manager does the same task as DMM manager have done and assimilates the diseases to discourse memory. If the input case frames do not match with any of the diseases, DM generates a "not identified response". If it could match with some diseases but not enough to satisfy the identification condition, it generates case structures of question response for the dialog generator. Dialog generator then asks a natural language question to the patient. If the matching result satisfies the identification condition, it generates identification response including advice case frames, carefulness case frames and treatment case frames. Then, the dialog generator generates natural language text dialog from the case frames.

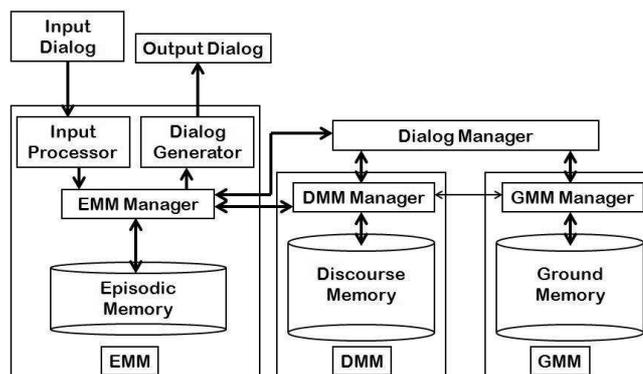


Figure 2. TLCN Dialog Processing System

#### B. Ground Memory Module (GMM)

The ground memory of GMM contains all kinds of knowledge. This is the main data source of the system. It stores information related to linguistic knowledge such as dictionary, grammar, and non-linguistic knowledge which includes disease, symptom, cause, treatment, effect etc. Procedural knowledge which relates inferring a fact from a collection of facts also described here. We considered [37], [38], [39], [40], [41] to implement ground memory. It also includes discourse information like most common diseases, seasonal diseases, previous record of a patient and recently processed disease information. In ground memory, knowledge is represented by a case frame structure form. Fig. 3 is showing the hierarchy of case frame representation where the nodes represent the case structures and the edges represent their relationships. In this figure, X and Y axis represent the case frames and Z axis denotes level of their relationships. More than one case frame of a previous level can be connected with one or more case frames of the next level and vice-versa. In the following sub-section we will discuss the details of knowledge representation in ground memory.

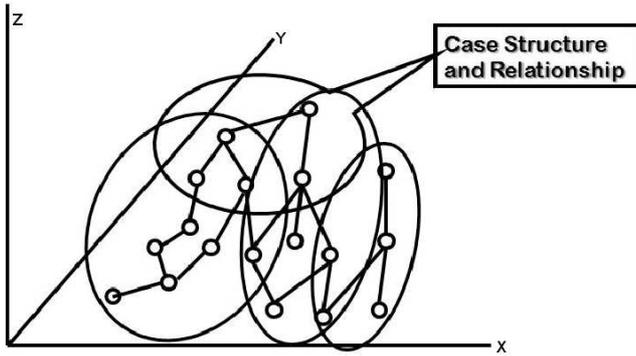


Figure 3. Case structure and their relationship

The knowledge is classified in some predefined classes. For this system prototype the nodes are classified in diseases. A class can be a property of other class. The GMM manager is an algorithm that searches the disease information in ground memory when requested by the Dialog Manager and also finds the discourse knowledge requested by discourse memory module. Ground memory always updates itself after every conversation and never deletes anything.

1) *Case frame Model*: The extended case frame model we used is an understanding model consists of predicates, semantic case relations (roles), modalities, and conjunctive relations that were previously proposed by SHIMAZU [25]. BRUCH [24] model also studied to design our extended model.

*sentence = simple\_sentence|sentence + conjunctive\_relation + sentence*  
*simple\_sentence = predicate + case\_relations + modalities*  
*case\_relations = object|method|direction|timeand space|supplement|modification*  
*modalities = tense|aspect|manner|intention|guess |attitude|negation|ability|necessity|...*  
*conjunctive\_relation = time|cause|reason|result| contrast|goal|assumption|*

- 1) **Case Relations**: one characteristic of this model is the relatively large collection of case relations. The collection includes: 1) basic cases like Fillmore's case system, 2) case relations such as extent, manner, and degree, which appear as adverbial phrases, and 3) additional cases represented by the inflection of a certain verb.
- 2) **Modalities**: semantic structures must include information on modality such as tense, aspect, intention, manner, attitude and assumption. Real sentences convey much of this type of information. Unfortunately, however, most understanding systems and linguistic theories have not dealt sufficiently with modalities.
- 3) **Conjunctive Relations**: A sentence generally consists of several sub-sentences, each of which represents a unit event. In the model, event relations

are expressed by conjunctive relations such as time, cause, reason, result, goal, assumption, contrast and circumstance. These are essential for understanding a sentence, and must be organized into semantic structures.

2) *Extended Case Frame Representation*: An extension of case structure model is proposed as a linguistic model for representing the text meaning structure. Thus the case frames act as a representation scheme for ground memory. The traditional case structure is a structure for a unit sentence which consists mainly in relations between noun and verb. This is not sufficient to represent structures of real sentences which sometimes have complex noun phrase and compound sentences. Also our proposed case structure has to have facilities for representing other structures involving relations between two nouns by verb and preposition. It has been designed to integrate those structures into one linguistic model. Its nature is hierarchical with respect to the way constituents are connected; iterative with respect to conjunction, and recursive with respect to embedding. Using this formalism, the syntactic and semantic structures of sentences can be represented uniformly.

In our proposed structure, every noun has six properties: adjective, delimiter, preposition, auxiliary verb, adverb and verb. Noun can be related by either verb or preposition. This case frame representation makes the matching algorithm simple and efficient because the same structure can represent both case frames and their relationships. The basic structure of case frame is depicted in Fig. 4.

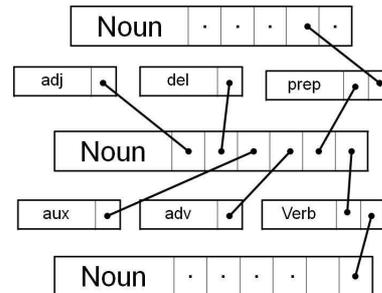


Figure 4. Case frame structure

Fig. 5 shows the representation of case frame. One case structure node can have more than one connection node and one connection node can have more than one case structure node. There is no restriction about consequence case structure and connection node to represent a complex or compound knowledge.

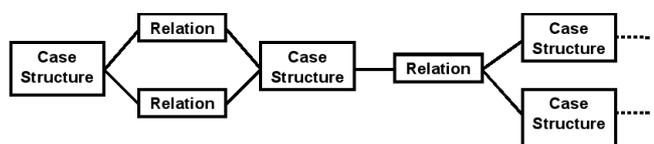


Figure 5. shows the relation of case structures

Fig. 6. shows an example of knowledge representation

in ground memory where the disease is considered as a head class and others are considered as daughter classes. This also explains the representation of knowledge in a real database. Here the example disease is cholera. Symptom, cause and others are sub-nodes of cholera. The sub-nodes of symptom and cause represent the case frames for the natural language texts written in the box located at the right bottom of the figure.

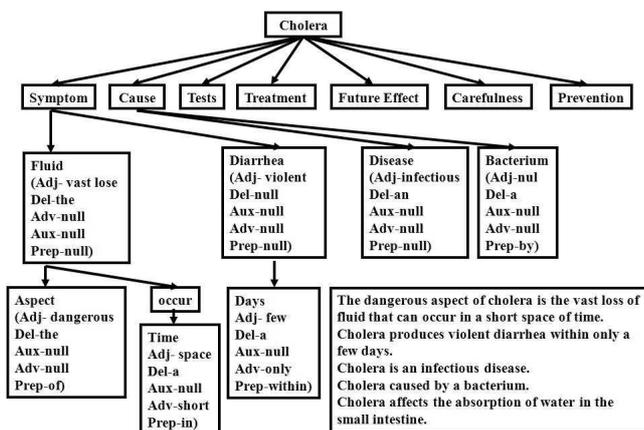


Figure 6. Knowledge representation

3) *Training to the Ground Memory*: The TLCN-DP system is based on doctor-patient dialog conversation. First the system was trained under the supervision of a professional doctor. At the time of training, the trainer carefully considered the discourse information and the level of nouns. The statistics of training corpus are given in Table I. The information about these diseases was collected from medical web links [42], [43] and [44]. The corpus of RHAMAN [45] is also considered to train the system.

TABLE I.  
TRAINING CORPUS STATISTICS

	Number
Diseases	30
Sentences for each disease	95
Normalized Sentences for each disease	220
Vocabulary Size	2500

C. *Discourse Memory Module (DMM)*

Discourse memory contains discourse information that provides situational and contextual information for an utterance environment. It contains nodes with different frequency. The frequency is calculated depending on the uses of knowledge. In our proposed system we used three frequencies: medium, high and very high. We standardize the frequency three after examining with other frequencies. We found better response using the frequency three. The very high frequency is determined by the most used case frames, recently used classes and seasonal disease case frames. The high frequency nodes are determined by the basic information of the patient and his past

medical record. Medium frequency nodes are determined by the medium uses of case frames. The knowledge and their frequencies of discourse memory are different for different patients.

1) *Discourse Information*: Some discourse information is pre-defined and some discourse information depends on real time information. After getting the basic information of patient and environment, the system decides the discourse knowledge considering the following criteria. The discourse frequency is always predefined by the trainer of the system.

- 1) Most common diseases: In every area or age there are some common diseases. This information is pre-defined by the trainer.
- 2) Seasonal disease Information: There are some common diseases for each season. This discourse information is identified by current date and the pre-defined seasonal information.
- 3) Recently treated disease information: The diseases and case frames are mostly handled by the current system in recent time. Its discourse frequency is determined by the frequency of handled disease.
- 4) Previous record of current patient: If the patient's basic information repeats, then case frames of entire previous records are considered as discourse. All previous dialog conversations are uploaded with very high discourse frequency.

If any disease matches with one of above criteria, the frequency will be medium. If it matches with two criteria, frequency will be high. If match with three criteria the discourse frequency will be very high.

2) *Dialog Matching*: First the DMM manager compares the patients symptoms and causes with all the symptoms and causes of discourse memory. If the symptom does not match it searches in ground memory with the help of GMM manager. It selects all of them as a candidate disease and sends all these case frames to dialog manager. DM calculates the highest probability between these diseases with a probability function. This function will be discussed in DM section. Depending on the probabilities, DM requests the responses from DMM.

3) *Response Generator*: In our TLCN-DP architecture, we have incorporated a response module to help patients to find the goal disease. The dialog manager analyzes the result and selects the type of system response. DMM manager provides case frames for responses to the language generator. The system responses are divided in three types: 1) response before disease identification, 2) after disease identification and 3) failed to identify.

- 1) Response before disease identification: in this type of response, first it considers the symptom and the causes of two diseases which have the highest probability between selected diseases. Find the dissimilarities of symptoms and causes between them. Response module then generates case frames for these. After 10 conversations it select case frames for pharmaceutical tests (if any test exists in the selected diseases).

- 2) After Disease Identification: once the disease has been identified, the dialog manager provides case frames for treatment including medicines, effects, and preventions for the specific disease to the language generator. It can also provide more case frames for details of this disease according to the patient request.
- 3) Failed to identify: if the dialog manager does not obtain any information about any diseases within 3 conversations and does not match more than 75% of the important symptoms and causes, it generates a failure notification.

#### D. Episodic Memory Module (EMM)

EMM receives input from the user and provides output to the user. The input processor receives the input text dialog and understands meaning. Episodic memory stores the meaning of the ongoing segment of utterances. As understanding proceeds, the essence of episodic memory assimilated into discourse memory and the essence of discourse memory assimilated into ground memory. The same way the output processor receives case frames from DMM manager and generates the appropriate natural language text for user. To understand the dialog the input processor uses some modules like text normalization unit, converter for case structure and semantic classifier. Episodic memory contains all the linguistic information and the recent conversation with the current user. EMM manager includes the entire previous dialog and responses with the case structure of the new dialog and classifies it in the pre-defined classes. It sends these classified case structures to the discourse memory module. Episodic memory is a temporary memory. It updates only for the ongoing process requirement. After every conversation with a user, it refreshes the episodic memory.

1) *Components*: EMM consists of language understanding module and the language generation module. The understanding module normalizes the input text and converts them to a language-neutral meaning representation – a case frame. Finally it classifies the case frames and send them to DMM. After getting the response from DMM, the language generation module produces the response in text form. In this section we will discuss four components: 1) text normalization, 2) case frame representation, 3) classifier and 4) Language generator.

- 1) Text normalization: this is an essential step for minimizing "noise" variations among words and utterances. The text normalization component is essentially based on using synonyms and other forms of syntactic normalization. The main normalization factors include stemming using a synonyms dictionary, removal of confusions, non-alphanumeric and non-white space characters.
- 2) Case frame representation: it generate the case frames using the same case frame model discussed in GMM section.
- 3) Classifier: in TLCN-DP the nodes are classified in 9 properties: 1) Disease, 2) Symptom, 3) Environment

Cause, 4) Physical Cause, 5) Other Cause, 6) Tests, 7) Treatment & Medicine, 8) Future Effect and 9) Prevention.

Disease is the head node. All the other nodes are related disease. Fig. 7 explains symptom and cause used to find the goal disease. Then pharmaceutical

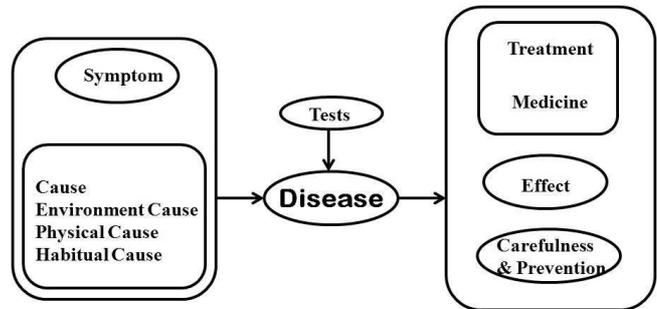


Figure 7. Relationship of knowledge

tests confirm the disease (if any tests exist). After that, the TLCN-DP will be able to provide detailed information to the patient including treatment, medicine, effect, carefulness and prevention.

If any utterance from patient's dialog contains any data related to the body then it will be classified as symptom class. If any utterance contains any data about treatment or medicine then it will be classified as treatment class. If any utterance contains any data about test then it will be classified as test class. Otherwise it will be classified as cause oriented. Classifier sends the classified utterances to the DMM manager.

- 4) Language Generator: It organizes the sentences by using the parts of speech and following grammatical rules from the response case frames, and generates the natural language output text to the user.

#### IV. NEURAL NETWORK

Natural language has traditionally been handled using symbolic computation and recursive processes. The most successful stochastic language models have been based on finite-state descriptions such as n-grams or hidden Markov models. However, finite-state models cannot represent hierarchical structures as found in natural language. In the past few years several RNN architectures have emerged which have been used for grammatical inference. RNNs have been used for several smaller natural language problems. Neural network models have been shown to be able to account for a variety of phenomena in phonology, morphology and role assignment. It has been shown that RNNs have the representational power required for hierarchical solutions, and that they are Turing equivalent. The Elman RNNs investigated in this paper to classify diseases.

##### A. Recurrent Neural Network

Recurrent Neural Network is a class of neural network where connections between units form a directed cycle.

This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. RNN have feedback connections and address the temporal relationship of inputs by maintaining internal states that have memory. RNN are networks with one or more feedback connection. A feedback connection is used to pass output of a neuron in a certain layer to the previous layer(s). The different between NLP and RNN is RNN have feed-forward connection for all neurons (fully connection). Therefore, the connections allow the network show the dynamic behavior.

**B. Elman Recurrent Neural Network**

Elman Recurrent Neural Network: A recurrent network with feedback from each hidden node to all hidden nodes. When training the Elman network back propagation-through-time is used rather than the truncated version used by Elman, i.e. in this paper Elman network refers to the architecture used by Elman but not the training algorithm. Back-propagation-through-time has been used to train the recurrent networks.

**V. CLASSIFIER**

Recurrent Neural Network Classifier is used to identify the probability of each disease. The inputs of this classifier are the indexes of classified case structures. As we mentioned before, each disease is a combination of some specific classes (symptom, cause etc.), each class is a combination of some case frames and each case frame is a combination of some case structures. All of these elements are indexed in advance.

**A. Architecture**

In our proposal we used an Extended Elman RNN for classification. According to the experiment of Pollack [3] and Lawrence-Giles-Fongs [4], Elman RNN classifier provides better performance for Natural Language Processing. Our classifier is using four word inputs, 7 hidden nodes. We did experiment with varying the number of hidden nodes and found optimal performance for 7 hidden nodes considering time and accuracy. The quadratic cost function, the learning rate schedules are shown below. An initial learning rate of 0.2 and we use the random weight initialization strategy. The Recurrent Network we used for this application is depicted in Fig. 8. Where S indicates Symptom, EC indicates Environment Cause, PC indicates Physical Cause, OC indicates Other Cause, T indicates Tests, H indicates Hidden node, D indicates Disease and NI indicates Not Identified.

- 1) Target Output: Target outputs were 0.1 and 0.9 using the logistic activation function. This helps avoid saturating the sigmoid function.
- 2) Weight Initialization: Random weights are initialized with the goal of ensuring that the sigmoid do not start out in saturation but are not very small (corresponding to a flat part of the error surface). In addition, several (20) sets of random weights

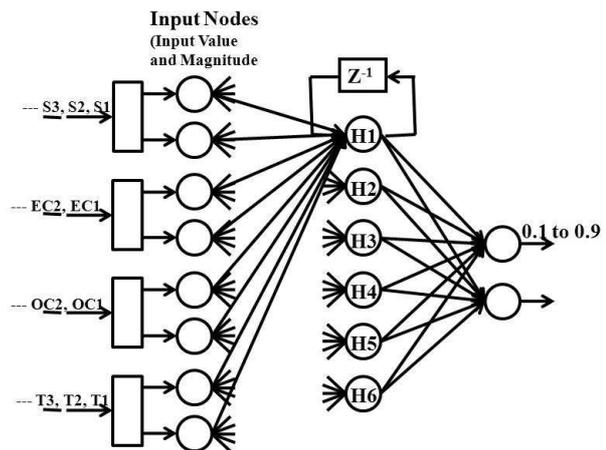


Figure 8. Disease Classifier

are tested and the set which provides the best performance on the training data is chosen. In our experiments on the current problem, it was found that these techniques do not make a significant difference.

- 3) Learning rate schedule: We used the learning rate scheduled by LAWRENCE [4] that was first proposed by Darken and Moody [46].
- 4) Activation Function: Symmetric sigmoid functions often improve convergence over the standard logistic function. For our particular problem we found that the difference was minor and that the logistic function resulted in better performance.
- 5) Cost Function: The relative entropy cost function has received particular attention and has a natural interpretation in terms of learning probabilities. We investigated using both quadratic and relative entropy cost functions.

$$E = \frac{1}{2} \sum_k (y_k - d_k)^2 \tag{1}$$

$$E = \sum_k \left[ \frac{1}{2} (1+y_k) \log \frac{1+y_k}{1+d_k} + \frac{1}{2} (1-y_k) \log \frac{1-y_k}{1-d_k} \right] \tag{2}$$

Where y and d correspond to the actual and desired output values, k ranges over the outputs. We found the quadratic cost function to provide better performance. A possible reason for this is that the use of the entropy cost function leads to an increased variance of weight updates and therefore decreased robustness in parameter updating.

- 6) Sectioning of the Training Data: We investigated dividing the training data into subsets. Initially, only one of these subsets was used for training. After 100% correct classification was obtained or a pre-specified time limit expired, an additional subset was added to the working set. This continued until the working set contained the entire training set. The data was ordered in terms of sentence length with the shortest sentences first. This enabled the

networks to focus on the simpler data first. Elman suggests that the initial training constrains later training in a useful way.

**B. Training and evaluation of RNN**

For each disease, we used different RNN. Each of these networks is trained by positive and same number of random negative set of index. Positive set of index denotes its own disease case frames where negative set of indexes denote all other case frames that are not the member of this disease. These case frames are already discussed in ground memory module section. We were able to train an RNN up to 100% correct classification on the training data. Generalization on disease identification resulted in 89% correct classification on average. This is better than the performance obtained from other researches. A possible reason for this is that there are similarities between input dialog and trained dialog that means the set of index of input dialogs and the set of index of disease information. The output range of this classifier is from 0.9 to 0.1.

**C. Simulation Details**

The network contained three layers including the input layer. The hidden layer contained 7 nodes. Each hidden layer node had a recurrent connection to all other hidden layer nodes. All inputs and outputs were within the range zero to one. Bias inputs were used. The best of 50 random weight sets was chosen based on training set performance. Targets outputs were 0.1 and 0.9 using the logistic output activation function. The quadratic cost function was used. The search then converge learning rate schedule used was Equ.(3)

$$\eta = \frac{\eta_0}{\frac{n}{N/2} + \frac{c_1}{\max(1, (c_1 - \frac{\max(0, c_1(n - c_2 N))}{(1 - c_2))})}} \quad (3)$$

Where

- $\eta = \text{learningrate}$
- $\eta_0 = \text{initiallearningrate} = 0.2$
- $N = \text{totaltrainingepochs}$
- $n = \text{currenttrainingepoch}$
- $C_1 = 50, c_2 = 0.65$

The training set consisted of 220 normalized sentences for each disease. The number of disease were 30.

**D. Dialog Manager**

Dialog manager is a controller for this system. It maintains the data flow between memory modules. Also it calculates the RNN classifier. After classification it selects the disease and system response.

**Pseudo code for Disease Selection:**

1. Select diseases for: Patients symptoms or causes = Disease(n)s symptoms

```

or causes.
if not found then Goto GMM Manager.
Select diseases for: Patients symptoms or causes=Disease(n)s
symptoms or causes.
if found then upload the disease &
goto DMM.
Else Generate "Not found" response.
2.if number of request dialog < 15
then
if (MAX(f(input dialog)/f(n))<80%
of f(n)) then
Generate "Request" response
else Generate "Disease Identified"
response
else Generate "Not found" response
3.upload it to Dialog manager.
4.Goto Dialog Manager
    
```

**VI. SYSTEM**

We developed a prototype system for a real-time medical advice application, called TLCN-DP, to help people who have to go to the hospital frequently. First we collected information about the diseases from some medical books and some medical web links. We built the ground memory with case frame by using this information. We selected some keyword for classification of the case frames. Then we trained the semantic classifier using a corpus under professional supervision. Then we experiment our system with the collection of dialog conversations. The following flow chart showing in the fig. 9 depicts the process flow for dialog conversation.

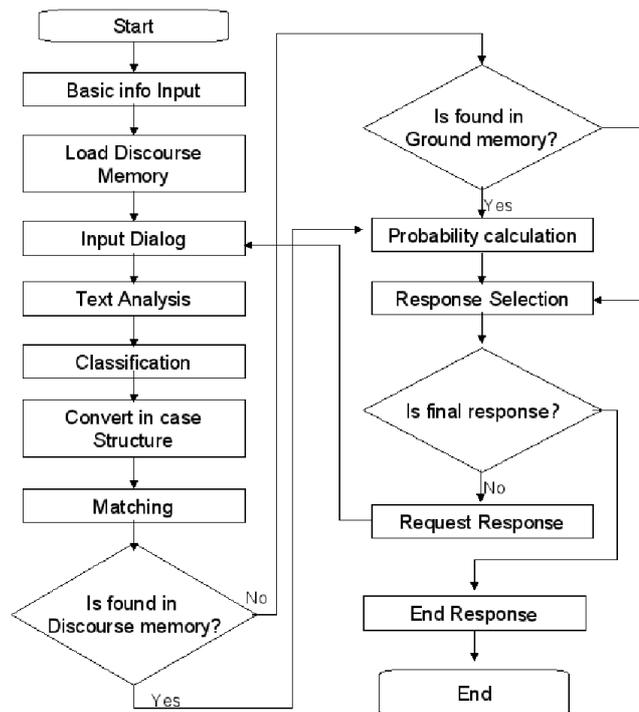


Figure 9. System Flow

A. PERFORMANCE EVALUATION

Evaluation of dialog system performance is a complex task and depends on the purpose of the desired dialog metric. We have trained our system with mostly common 30 diseases from [42], [43], [44] where 11 are mainly related with age, 10 are related with season and the other of 9 diseases are related with food, accident, habitual, health condition and location. The real doctor-patient dialog conversation is collected for this evaluation. The dialogues have been collected from various books on medical interviews and some frequently asked questions-answering medical web links. The entire information and output was verified by a professional doctor. 85 patients dialogs were simulated with this system prototype where 10 patients were tested 3 times and 15 patients 2 times and 25 patients only once. Table II presents the existing disease identification accuracy with the number of dialog conversation. According to the result analysis table; we found that the system achieved a very satisfactory result for 10 to 20 diseases. For 15 diseases we found 79% accuracy of disease identification. For less than 10 diseases the system can identify with very few dialog conversations. 82% accuracy achieved only for 7 utterances. Before utilizing RNN we implemented another system where we used a comparatively simple probability function. More than 10% overall accuracy gained from that system. The result of previous system is depicted in table III. We did not verify more than 20 diseases because the performance was already not satisfactory for 20 diseases.

TABLE II.  
DISEASE IDENTIFICATION EVALUATION

Diseases	Utterances					
	5	7	9	11	13	15
10	57%	82%	83%	83%	84%	84%
15	35%	60%	78%	78%	79%	79%
20	19%	41%	59%	66%	67%	68%
25	14%	37%	51%	53%	55%	56%
30	10%	23%	33%	42%	48%	48%

TABLE III.  
PREVIOUS SYSTEM EVALUATION

Diseases	Utterances				
	7	9	11	13	15
10	56%	59%	62%	64%	66%
15	49%	55%	58%	60%	63%
20	46%	50%	53%	57%	61%

VII. CONCLUSION

The main underlying strategy is to adopt Elman RNN with Three Layer Conceptual Network. Uses of extended case frame, dialog matching algorithm, disease identification algorithm and response generation criteria are also presented. In addition of these fundamental strategies, we discussed the characteristic of Elman RNN with our system. Finally we presented a comparative study with our previous system.

Still, there are lots of scopes to improve in neural network application. We are looking for some scope to implement TLCN using efficient neural network. Adopting some module, it might be possible to train this system by the real time conversation of a doctor and patient. And in future we plan to use voice dialog rather than text dialog.

This is only the beginning of Neural Network with Three Layer Conceptual Network. Our goal is to make human-computer successful interaction. We strongly believe this research is a new breed of handshaking between TLCN and neural network.

ACKNOWLEDGMENT

We thank Hirosato Nomura for numerous discussions concerning this work, Teigo Nakamura and Manuel Medina González for their assistance and the reviewers for their detailed comments.

REFERENCES

- [1] J. L. Elman, "Distributed representations, simple recurrent networks, and grammatical structure," *Machine Learning*, no. 7, pp. 195–226, 1991.
- [2] —, "Finding structure time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.
- [3] J. B. Pollack, "Recursive distributed representations," *Artificial Intelligence*, vol. 46, pp. 77–105, 1990.
- [4] S. Lawrence, C. L. Giles, and S. Fong, "Natural language grammatical inference with recurrent neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, pp. 126–140, 2000.
- [5] M. Towsey, J. Diederich, I. Schellhammer, S. Chalup, and C. Brugman, "Natural language learning by recurrent neural networks: a comparison with probabilistic approaches," pp. 3–10, 1998.
- [6] A. Graves, N. Beringer, and J. Schmidhuber, "A comparison between spiking and differentiable recurrent neural networks on spoken digit recognition."
- [7] D. Chen, G. Z. Sun, H. H. Chen, and Y. C. Lee, "Extracting and learning an unknown grammar with recurrent neural networks," *Advance in neural Information Processing System*, 1992.
- [8] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Computation*, vol. 2, no. 4, pp. 490–501, 1990.
- [9] G. Scheler and T. Munchen, "With raised eyebrows or the eyebrows raised? a neural network approach to grammar checking for definiteness," pp. 160–170, 1996.
- [10] C. Fabrizio, F. Paolo, L. Vincenzo, and S. Giovanni, "Towards incremental parsing of natural language using recursive neural networks," *Towards Incremental Parsing of Natural Language using Recursive Neural Networks*, p. Paginated, 2002.
- [11] Y. Li and J. G. Harris, "A spiking recurrent neural network," *VLSI, IEEE Computer Society Annual Symposium on*, p. 321, 2004.
- [12] S. C. Kwasny, S. Johnson, and B. L. Kalman, "Recurrent natural language parsing 1."
- [13] R. Miikkulainen, "Natural language processing with sub-symbolic neural networks," pp. 120–139, 1997.
- [14] M. H. Christiansen and N. Chater, "Natural language recursion and recurrent neural networks," 1994.

- [15] S. Wermter and V. Weber, "Screen: Learning a flat syntactic and semantic spoken language analysis using artificial neural networks," *Journal of Artificial Intelligence Research*, vol. 6, pp. 35–85, 1997.
- [16] R. M. Khalilur and E. Tsutomu, "Three Layer Conceptual Network Dialog Processor," in *The Twelfth IASTED International Conference on AI and Soft Computing*, 2008, pp. 92–97.
- [17] H. Nomura, "Modeling and representative framework for linguistic and non-linguistic knowledge in natural language understanding," in *Germany-Japan Science Seminar*, 1986, pp. 1–10.
- [18] M. A. Walker, "Evaluating discourse processing algorithms," in *Proceedings of the ACL*, 1989.
- [19] H. Nomura, "Experimental machine translation systems lute," in *Second Joint European-Japanese Workshop on Machine Translation*, 1985, pp. 1–2.
- [20] D. Marcu and A. Echihiabi, "An unsupervised approach to recognizing discourse relations," in *The 40th Annual Meeting of the Association for Computational Linguistics*, July 2002, pp. 368–375.
- [21] E. Tulving and F. I. M. Craik, "The oxford handbook of memory."
- [22] T. K. Landauer, "How much do people remember? some estimates of the quantity of learned information in long-term memory." *Cognitive Science*, vol. 10, pp. 477–493, 1986.
- [23] R. L. Buckner, "Beyond here: Contributions of specific prefrontal brain areas to long-term memory retrieval." *Psychonomic Bulletin and Review*, vol. 3, pp. 149–158, 1996.
- [24] B. Bruce, "Case systems for natural language," *Artificial Intelligence*, vol. 6, no. 4, pp. 327–360, 1975.
- [25] A. Shimazu, S. Naito, and H. Nomura, "Japanese language semantic analyzer based on an extended case frame model," in *International Joint Conference on Artificial Intelligence*, 1983, pp. 717–720.
- [26] S. Tsumoto, "Automated extraction of medical expert system rules from clinical databases based on rough set theory." *Information Sciences*, p. 112, 1998.
- [27] J. H. Frenster, "Expert systems and open systems in medical artificial intelligence." *Am Assoc Medical Systems and Informatics*, vol. 7, pp. 118–120, 1989.
- [28] F. Shahbaz, F. Maqbool, S. Razzaq, K. Irfan, and T. Zia, "The role of medical expert systems in pakistan." *World academy of science engineering and technology*, vol. 27, 2008.
- [29] Y.-S. Lee, D. J. Sinder, and C. J. Weinstein, "Interlingua-based english-korean two-way speech translation of doctor-patient dialogues with cclinc," *Machine Translation*, vol. 17, pp. 213–243, 2002.
- [30] J. F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent, "Conversational human-computer interaction," in *IASTED International Conference on AI and Soft Computing*, vol. 12, Sept. 2001, pp. 628–802.
- [31] M. Kipp, J. Alexandersson, R. Engel, and N. Reithinger, "Dialog processing," *W. Wahlster (ed.) Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 452–465, 2000.
- [32] E. Tsutomu and K. Tsuneo, "Cooperative understanding of utterances and gestures in a dialogue-based problem solving system." *Computational Intelligence*, pp. 152–169, 1999.
- [33] K. Shimada, Y. Uchida, S. Sato, S. Minewaki, and T. Endo, "Speech understanding using confidence measures and dependency relations," *Proc. of PACLING2005*, pp. 278–283, 2005.
- [34] N. Okada and T. Endo, "Story generation based on dynamics of the mind." *Computational Intelligence*, pp. 123–160, 1992.
- [35] K. Shimada, K. Iwashita, and T. Endo, "A case study of comparison of several methods for corpus-based speech intention identification," *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING2007)*, pp. 255–262, 2007.
- [36] S. Minewaki, K. Shimada, and T. Endo, "Interpretation of utterances based on relevance theory: Toward the formalization of implicature with the maximum relevance," *Proc. of PACLING2005*, pp. 214–222, 2005.
- [37] A. Graves, N. Beringer, and J. Schmidhuber, "Grammar learning for spoken language understanding," *Automatic Speech Recognition and Understanding*, pp. 292–295, 2001.
- [38] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, 2000.
- [39] C. Noam, "Three models for the description of language," *IRE Transactions on Information Theory*, p. 113124, 1956.
- [40] ———, "On certain formal properties of grammars," *Information and Control*, p. 1959, 137–167.
- [41] C. Noam, Schtzenberger, and M. P., "The algebraic theory of context free languages," *Computer Programming and Formal Languages. Amsterdam: North Holland*, pp. 118–161, 1963.
- [42] "<http://www.netdoctor.co.uk/>," Weblink, Online Medical Solution.
- [43] "<http://www.doctoronline.nhs.uk/ver02/index.asp>," Weblink, Online Doctor.
- [44] "<http://www.nzdoctor.co.nz/default.aspx>," Weblink, online Medical Solution.
- [45] R. M. Khalilur and E. Tsutomu, "Recurrent neural network classifier for three layer conceptual network and performance evaluation," in *11th International Conference on Computer and Information Technology*, 2008, pp. 747–752.
- [46] D. C. and M. J., "Note on learning rate schedules for stochastic optimization," *In Neural Information Processing Systems*, vol. 3, pp. 832–838, 1991.

**Md. Khalilur Rhaman** received his Dr.Eng. degree from Kyushu Institute of Technology in 2009, Japan and B.Sc. and M.Sc. degrees from Institute of Science and Technology, National University, Bangladesh, in 1997 and 1998, respectively.

Currently he is an Assistant Professor of Department of Computer Science and Engineering, BRAC University, Bangladesh. In 2002, he joined Uttara University, Bangladesh, where he held the position of Lecturer of Department of Computer Science and Engineering. His research interests include Natural Language Processing, Neural Network, Image Processing and Data Mining.

**Tsutomu Endo** received the B.Eng., M.Eng., and Dr.Eng. degrees from Kyushu University in 1972, 1974, and 1979 respectively.

Currently, he is a Professor of the Department of Artificial Intelligence, Kyushu Institute of Technology. His research interests include natural language understanding, computer vision and multimodal interface. He is a member of the IPSJ, JSAI, JSSST, RSJ and ANLP.