A Chatbot Using LSTM-based Multi-Layer Embedding for Elderly Care

Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan
{huntfox.su, chunghsienwu, iamkyh77}@gmail.com

Qian-Bei Hong, Hsin-Min Wang
PhD Program for Multimedia Systems and Intelligent Computing
National Cheng Kung University and Academia Sinica
Taiwan
qbhong75@gmail.com, whm@iis.sinica.edu.tw

***Abstract*—According to demographic changes, the services designed for the elderly are becoming more needed than before and increasingly important. In previous work, social media or community-based question-answer data were generally used to build the chatbot. In this study, we collected the MHMC chitchat dataset from daily conversations with the elderly. Since people are free to say anything to the system, the collected sentences are converted into patterns in the preprocessing part to cover the variability of conversational sentences. Then, an LSTM-based multi-layer embedding model is used to extract the semantic information between words and sentences in a single turn with multiple sentences when chatting with the elderly. Finally, the Euclidean distance is employed to select a proper question pattern, which is further used to select the corresponding answer to respond to the elderly. For performance evaluation, five-fold cross-validation scheme was employed for training and evaluation. Experimental results show that the proposed method achieved an accuracy of 79.96% for top-1 response selection, which outperformed the traditional Okapi model.**

Keywords- Chatbot system; LSTM; Elderly care; Pattern matching

I. INTRODUCTION *(HEADING 1)*

Population aging has become a significant issue all over the world in recent years. Services designed for the elderly have become more needed than ever before. In Taiwan, statistics in 2016 show that about 13.20% of the total population was aged 65 or over. To respond to the demographic changes, Taiwan government has put forward a number of long-term policies and encouraged industries and academia to invest technology funds in the areas of elderly care.

Artificial intelligence is the key direction of science and technology development in recent years, including Natural Language Processing (NLP), Spoken Dialogue System (SDS), image understanding and recognition, chess playing and so on. Among these applications, spoken dialogue system has continuously been a hot topic over the past few decades and is commonly used in our lives today.

Many scholars put their effort in how to build a dialogue system which can interact with human beings by natural language. A chatbot or dialog system is no doubt a good choice and could be applied in this area. A dialog system could be designed specifically for elderly care to accompany the elderly and encourage them to communicate with them by serving as a listener. Giving an appropriate response could help the elderly to increase their affective interactions, and reduce their sense of loneliness and isolation. This could slow the rate of disability of the elderly, and reduce social costs and the burden of care workers.

Since the elderly people usually like to share their interesting stories and memories with others, an LSTM-based multi-layer embedding mechanism is used to extract the semantic information of the long sentences by the elderly people. In addition to daily conversation, we want to provide recommendations for activities, restaurants and health information to the elderly. Based on the idea of elderly care chatbot system mentioned above, this study mainly focuses on developing a chatbot system, which provides a daily conversation with placation response on mobile devices.

The rest of this paper is organized as follows: Section II introduces the state-of-the-art approaches. Section III presents the database collection. Next, the system framework is described in Section IV. Section V presents experimental setup and results. Finally, conclusions and future works are described in Section VI.

II. STATE-OF-THE-ART APPROACHES

A. Question Answering and Dialogue System

Question Answering (QA) is in the fields of information retrieval (IR) and natural language processing (NLP). From the perspective of knowledge, the QA system can be divided into "closed domain" and "open domain". Systems in closed domain [1] only focuses on answering questions in the specific field, such as medicine or law, and can have better performance. An open-domain QA system [2-3] is able to answer all the questions, not limited to a specific field. Therefore, it is more difficult than a closed domain system.

There are many large structured knowledge databases on the Internet, such as DBpedia [4], Freebase [5], etc. Based on the structured knowledge databases, many Knowledge-based QA (KBQA) [6-7] systems were proposed, but such systems can only answer specific words and lack contextual information. Most important of all, the predefined database may not cover all the user questions. Frequently Asked Questions (FAQs) is a common application customer service, in which questionnaires and corresponding answers are pre-defined in the database. User's query is compared with questions in existing FAQs to get a related answer to respond to the query. In contrast, Information Retrieval-based QA (IRQA) systems retrieve the answer from the result page of the search engine, which might be more in line with natural situation. Such a system must have a strong ability to retrieve information in order to

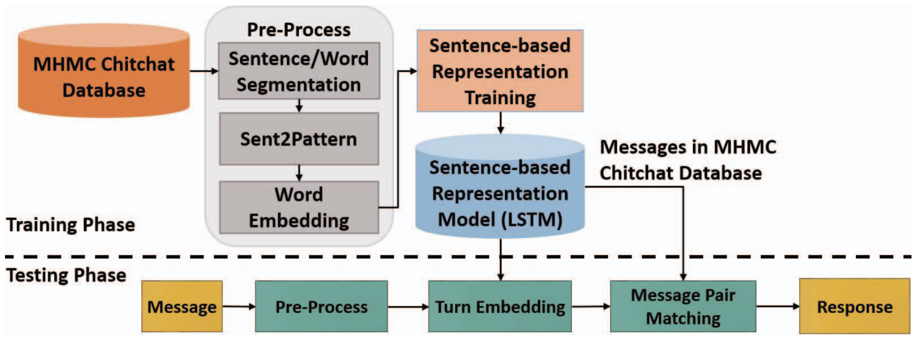

Figure 1. The proposed chatbot framework.

filter out the correct answer in a large number of web pages. Among them, Community Question and Answering Systems (CQA), such as Yahoo! Answers or Baidu Zhidao, are the recent use of many researchers.

Dialogue system is one of the areas of QA systems, in which task-oriented dialogue system has been developed for more than a decade. Recently, experts started to do research in non-task-oriented dialogue system. General speaking, task-oriented dialogue systems provide a framework for users to have simple conversations with a machine to complete a specific task like reserving a flight or querying tourist information. One of the limitation of task-oriented service is that the conversation cannot exceed the system topic scope. Non-task-oriented systems do not intend to help user to complete any specific task but for chitchat or entertainment.

B. Chatbot System

The non-task-oriented dialog system, also known as the Chatbot or the Conversational system is unable to predefine the values in semantic slots for spoken language understanding, making it much more difficult than task-oriented systems.

The early chatbot system usually used a rule-based or template-based approach, which required a manually collected database as the input. However, the custom templates and rules cannot cover most of the user' utterances in an open domain system. The reason is that it is impossible to predict the usage of user's words in an open domain. Therefore, scholars began to focus on automatic or semi-automatic response generation system [8-9]. There are two types of automatic response generation systems. The first is the generation-based method [10], which uses the sequence-to-sequence model (Seq2Seq) to generate a response sentence. This method does not need to manually process the database and can handle open domain problems. Nevertheless, sequence-to-sequence model must be trained by a very large database, and the resulting sentence is difficult to maintain consistency and fluency. The other is retrieval-based method [11-12], which compares the queries with the message-response pairs in predefined databases. The retrieval-based method can retrieve the sentence with high naturality and fluency but is usually used in a closed domain. In order to provide fluent response sentences for the elderly, we believe that the retrieval-based method is more appropriate for our system.

III. Database Design and Collection

There are many community-based chatting data on the Internet used by scholars in the field of chatbot system at present. Different from other chatbots, we aim to build a daily conversation chatbot system. In order to meet the actual situation, we collected daily conversations with the elderly people to construct the MHMC chitchat database. The MHMC chitchat database is composed of 2239 Message-Response pairs (MR pairs), which were collected from daily life. Responses in MR pairs only consider the message in the current turn, such as "Don't you feel bored (你不覺得無聊嗎)?" paired with "It's not boring to have you chatting with me (有你陪我聊天就不會無聊)".

As it is impossible to collect all types of chitchat utterances, we first convert all messages and responses in the database into sentence patterns based on 61 pre-defined tags. Then, we categorize the sentence patterns to reduce the complexity of the utterance and operation time, and increase the coverage of the collected database. Finally, we obtain 23 categories, 317 message patterns and 164 response patterns.

IV. System Framework

Figure 1 shows the training and testing phase of the proposed system framework. In the training phase, the proposed system is divided into three parts, which are pre-processing, sentence-based representation, and response selection. The MHMC chitchat database was collected from Chinese daily dialogues, which contain multiple MR pairs. In pre-processing, each message turn in a dialogue will be split into sentences and words. Then, message patterns are generated by replacing the words in the message with manually pre-defined tags. Message patterns can reduce the variability of the messages in the database. Therefore, our system can handle complex dialogues. Finally, each message pattern is converted into a vector representation based on the method of word/sentence embedding. Then, we use the long short-term memory (LSTM) model to convert the content of sentences into sentence-based representation vectors. Finally, Euclidean distance is employed to select a proper question pattern and its corresponding response. In the test phase, the input message is first segmented into words, converted into patterns and represented using word embedding. Next, sentence representation embedding is used to extract the sentence representation vector of the input message. In the end, we calculate the Euclidean distance to select the appropriate message pattern and choose the corresponding response as the final response sentence.

A. Chinese Dialog Sentence Pre-process

The MHMC chitchat database contains fluent spoken sentences in Chinese. To convert the sentence into a comprehensible embedding vector, we first use Jieba [26], an open source Chinese word segmentation tool, to separate words for Chines sentences. Although there are many statements expressed with similar meaning in semantics, a keyword made the statement different, such as "What do you want to eat for dinner (晚餐要吃什麼)" and "What do you want to eat for lunch (午餐要吃什麼)". Then, we convert the sentences into patterns, and hope to reduce the complexity of our dataset. Each word in the database will be compared with the 61 pre-defined tags. If the word belongs to a tag, it will be replaced by the tag and stored in a dictionary, as shown in Figure 2. From now on, all the messages are represented by patterns.

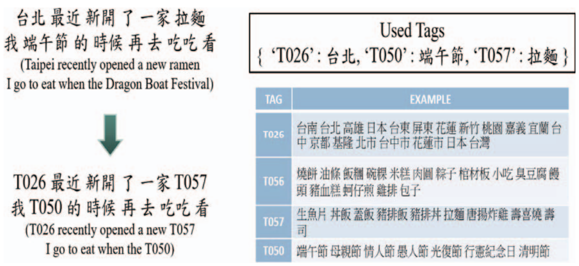

Figure 2. Example of sentence to pattern conversion.

Next, we use GloVe [27] method to train word vector model. We use Chinese Gigaword Corpus containing about 17.9 million sentences and MHMC chatting database to train the GloVe model. After training the word vector model, the words in the MHMC chatting database are converted to 300-dimensional vectors, as the input of sentence modeling, as shown in Figure 3.

B. Sentence-based Representation Model

LSTM is a special kind of Recurrent Neural Network (RNN) architecture. The architecture of LSTM is shown in Figure 4. There are three gates including input gate, forget gate and output gate, which are used to add or remove the information to the cell state. The output of each gates will go through a sigmoid function, and output the values between zero and one. A value of zero means "nothing through" and a value of one means "everything through." For an input sequence X, X_t is the input at time *t*. Given the previous hidden layer output h_{t-1} and X_t, we can receive the hidden layer output h_t by controlling the gates and the cell state *C*. The forget gate is used to decide what information will be thrown away from the cell state, and the decision is made by a sigmoid layer. The input is used for the output of the previous hidden layer h_{t-1} and input data X_t.

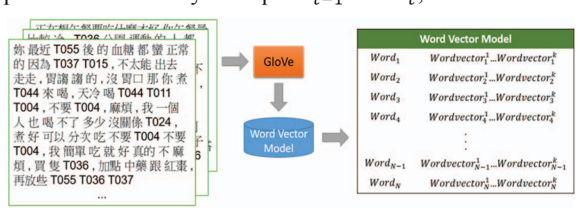

Figure 3. Training method for the word vector model.

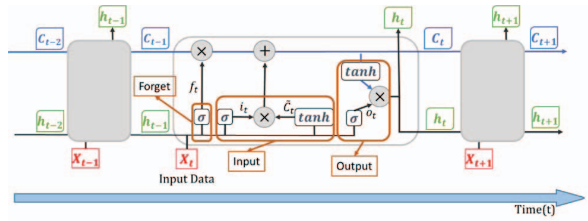

Figure 4. The architecture of an LSTM.

$$f_t = \sigma(W_f h_{t-1} + W_f X_t + b_f) \quad (1)$$

where X denotes the input data, W the weight and b the bias value. The input gate is used to decide what information is stored in the cell. In Equation (2), a sigmoid layer decides which values will be updated. In Equation (3), a $tanh$ layer creates a vector of new candidate values C_t which could be added to the state. In the next step, these two parts will be combined to update the state.

$$i_t = \sigma(W_i h_{t-1} + W_i X_t + b_i) \quad (2)$$

$$C_t = tanh(W_C h_{t-1} + W_C X_t + b_C) \quad (3)$$

After calculating the previous steps, we update the old cell state C_{t-1} which can be updated into the new cell state C_t. As shown in Equation (4), sum up the output of forget gate $f_t * C_{t-1}$ and the output of input gate $i_t * C_t$, which actually drops the old information and adds the new information.

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (4)$$

The output gate decides the output values. Firstly, a sigmoid layer decides which part of the cell state is the output, as shown in Equation (5). Then, project the cell state to a value between -1 and 1 through a $tanh$ layer. Finally, multiply the output of the $tanh$ layer and the sigmoid layer to decide the hidden layer output h_t, as shown in Equation (6). Output data will be delivered to time $t + 1$, and impact the next input data.

$$o_t = \sigma(W_o h_{t-1} + W_o X_t + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

This study uses a two-layer LSTM model to solve the ignorance of the semantic transition between sentences in a message. The first layer of the LSTM model accepts the word-based sequence to extract the semantic information between words in a sentence. Words in the message patterns are converted into word vectors, and the word vectors are fed to the LSTM model sequentially. Hidden vector of the last word in the message is treated as the output vector V_s. The second layer of the LSTM model accepts the sentence-based sequence, which can extract the relation between sentences. As shown in Figure 5, each message in the dataset is divided into sentences, the same as the training data of the first-layer LSTM. After the process in the first layer LSTM, sentences are converted into a sentence-based semantic vector, which is the input vector of the second layer LSTM. Similar to the first layer, hidden vector of the last sentence is treated as the semantic feature $V_{message}$.

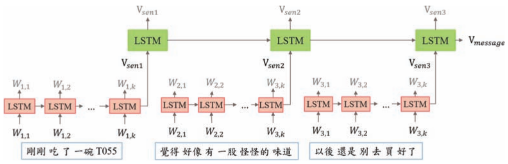

Figure 5. Training method for the word vector model.

C. *Response Selection*

For a chatbot system, real-time response is important. The response delay caused by comparing all sentences might reduce user's willingness to chat. Thus, we compare the input message with the representative sentences for each category by using Euclidean distance method, and get the corresponding response of the candidate message from top *n* categories.

In mathematics, the Euclidean distance is the "ordinary" straight-line distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space. In Cartesian coordinates, if $\boldsymbol{p} = (p_1, p_2, \dots, p_n)$ and $\boldsymbol{q} = (q_1, q_2, \dots, q_n)$ are two points in Euclidean *n*-space, then the distance between p and q is given by the Pythagorean formula as

$$Distance(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{7}$$

In this study, we calculate the Euclidean distance to select the appropriate message pattern and choose the corresponding response as the final response sentence.

V. Experimental Results

The number of training epochs influences the system performance. If the number of epochs is not high enough, the system cannot learn well. If the epoch number is too many, it might lead to overfitting and increase the training time. Thus, we conducted an experiment to decide the epoch number. As shown in Figure 6, we set 500 epochs and recorded the loss and accuracy values of each epoch. The accuracy at 150 epochs tended to be gentle and the loss did not change at 200 epochs. In the end, we set the number of epochs as 200.

The hidden size of the two-layer LSTM is changeable. Thus, we cannot do all the experiments with all combinations of the variables. We evaluated the turn embedding model in different hidden sizes by a loss function first. When the turn embedding model obtained the lowest loss value, it achieved the best effect. This means the hidden vector can indeed represent the input vector. The loss function is described in Equation (8).

$$loss = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{8}$$

The result is shown in Figure 7. The loss value of the first layer LSTM was nearly flat when the hidden layer dimension was 512. Then we fixed the hidden size of the first layer to 512, and repeated the experiment for the second layer of the LSTM. Since we can see that the 512 hidden nodes still obtained a substantially consistent result, the node number in the first- and the second-layer LSTM were set to 512.

In this study, we used Okapi BM25 as the baseline model. Okapi BM25 is a ranking function to rank documents according to the relevance to a given query. The comparison between the proposed system and the baseline is shown in TABLE I. The experimental result shows that the proposed method achieved an accuracy of 79.96% for top 1, 93.14% for top 5, and 94.85% for top 10 message pair matching. The result shows that the proposed system performance in the message pair matching task is much better than the baseline system. Although Okapi considered

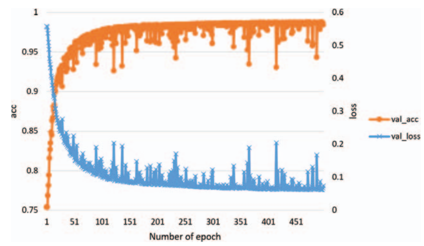

Figure 6. Experiment result of number of epoch.

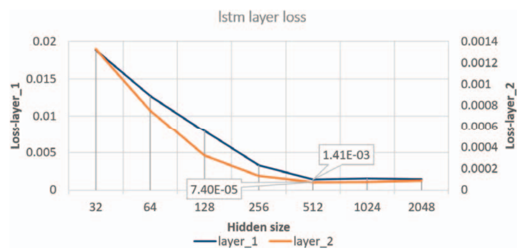

Figure 7. The loss value in different hidden size.

the saturation and reduction, it mainly dealt with the word frequency in the sentences and in the corpus. The words in the representative sentences and the messages are rarely in common. Therefore, Okapi model is unable to select the correct category.

TABLE I. Overall result for Message Pair matching

	P@1	P@5	P@10
Proposed System	16.01%	37.72%	56.19%
Okapi System	**79.96%**	**93.14%**	**94.85%**

VI. CONCLUSIONS AND FUTURE WORK

This study proposed a chatbot for elderly care, which can accompany and chat with the elderly people to reduce their loneliness. In order to achieve daily conversation, we collected the MHMC chitchat database from 40 subjects as the training data. All messages in the MHMC chitchat database were categorized and patternized. Then, we used a two-layer LSTM model to deal with the semantic relation between words and sentences. Finally, we selected the corresponding response by comparing message pairs through Euclidean distance. The result shows that the proposed system can output a better response, which achieved 79.96% at top 1 message pair matching.

There are several issues needed to be considered in our future work. First, more topics in the dialog need to be provided. Secondly, emotional and personality factors can be considered. We believe this will increase the wellness of the elderly by chatting with the system.

References

[1] M. Vargas-Vera and M. D. Lytras, "AQUA: A Closed-Domain Question Answering System," Information Systems Management, vol. 27, no. 3, pp. 217-225, 2010.

[2] S. Quarteroni and S. Manandhar, "Designing an interactive open-domain question answering system," Natural Language Engineering, vol. 15, no. 01, pp. 73-95, 2009.

[3] E. Cabrio, J. Cojan, A. P. Aprosio, B. Magnini, A. Lavelli, and F. Gandon, "QAKiS: an open domain QA system based on relational patterns," Proc. the 2012th International Conference on Posters & Demonstrations Track-Volume 914, CEUR-WS. Org, 2012, pp. 9-12.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," Proc. the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, Springer Press, pp. 722-735, 2007.

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," Proc. the 2008 ACM SIGMOD international conference on Management of data, ACM, 2008, pp. 1247-1250.

[6] S. He, K. Liu, Y. Zhang, L. Xu, and J. Zhao, "Question Answering over Linked Data Using First-order Logic," Proc. the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP), SIGDAT, 2014, pp. 1092-1103.

[7] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic Parsing on Freebase from Question-Answer Pairs,", Proc. the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP), SIGDAT, 2013, vol. 2, no. 5, pp. 6.

[8] R. Higashinaka, K. Imamura, T. Meguro, C. Miyazaki, N. Kobayashi, H. Sugiyama, T. Hirano, T. Makino, and Y. Matsuo, "Towards an open-domain conversational system fully based on natural language processing," Proc. COLING 2014: the 25th International Conference on Computational Linguistics, ACL, 2014, pp. 928-939.

[9] H. Sugiyama, T. Meguro, R. Higashinaka, and Y. Minami, "Open-domain utterance generation for conversational dialogue systems using web-scale dependency structures," Proc. The 14th annual SIGdial Meeting on Discourse and Dialogue, SIGDIAL, 2013, pp. 334-338.

[10] O. Vinyals and Q. Le, "A neural conversational model," arXiv preprint, 2015, Available: https://arxiv.org/abs/1506.05869.

[11] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," Proc. Advances in neural information processing systems 27, NIPS, 2014, pp. 2042-2050.

[12] Y. Wu, W. Wu, Z. Li, and M. Zhou, "Response Selection with Topic Clues for Retrieval-based Chatbots," arXiv preprint arXiv:1605.00090, 2016.