JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 365-379 (2021)
DOI: 10.6688/JISE.202103_37(2).0006

# Outpatient Text Classification System Using LSTM

CHE-WEN CHEN[1], SHIH-PANG TSENG[2] AND JHING-FA WANG[3]
[1,3]*Department of Electrical Engineering*
*National Cheng Kung University*
*Tainan, 701401 Taiwan*
[2]*School of Software*
*Changzhou College of Information Technology*
*Changzhou, 213164 P.R. China*
*E-mail:* [1]*kfcmax300@gmail.com;* [2]*tsengshihpang@ccit.js.cn;* [3]*jameswangjf@gmail.com*

Outpatient text classification is an important problem in medical natural language processing. Existing research has conventionally focused on rule-based or knowledge-source-based feature engineering, but only a few studies have utilized the effective feature learning capabilities of deep learning methods. A long short-term memory (LSTM) model for the outpatient text classification system was proposed in this research. The system has the ability to classify outpatient categories according to textual content on website Taiwan E Hospital. The experimental results showed that our system has very well in the task. The success of the LSTM model applications in the outpatient system provide users to inquire about their health status as references.

***Keywords:*** text classification, natural language processing, human-robot interaction, smart healthcare, service robot system

## 1. INTRODUCTION

In recent years, there has been increasing interest in integrating techniques drawn from the fields of artificial intelligence (AI) and robotics, including computer vision [1], privacy [2], traffic [3], emotion recognition [4], face recognition [5], speech recognition [6] and natural language processing (NLP) [7]. Improvements in intelligent control systems and precision sensors have resulted in a wide variety of robot applications in the services field, including in network [8], restaurants [9], tourism [10], markets [11], law [12], health care [13] and at smart home [14]. Nowadays, Service robots have greatly improved people's lives [15]. People still need intelligent, safe, and effective service from service robots. A natural way to interact with service robots during the realization of a task is using speech. Therefore, it is very important for a dialogue system to have outstanding understanding. [16]. On the basis of these considerations, the Zenbo Project [17] was launched with the objective of developing high-level cognitive functions for service robots to make them suitable for human-robot interaction. We previously designed a robot system, which provided basic functions such as product consultation, product searching and FAQ. A schematic diagram of a user talking to a robot is shown in Fig. 1. Users

Received November 14, 2019; revised February 24, 2020; accepted March 30, 2020.
Communicated by Jimson Mathew.

can use natural language to communicate with and command the robot in the pharmacy through the human-machine interface.

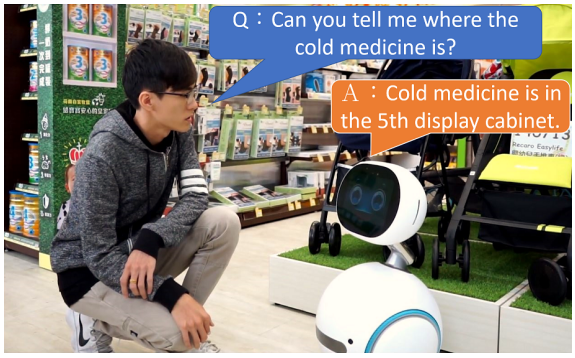


Fig. 1. An illustration of a conversation between a user and a robot.

The difference is that service robots in hospitals need to provide outpatient consultations. With the outpatient consultations, outpatients can talk about their situation to the service robot and the robot can tell them which clinic they should register with. In this paper, we present a model for the behavior of and dialog with service robots based on long short-term memory (LSTM) [18]. As this recurrent neural network (RNN) architecture is very powerful, sequences that are the same as the input sequences tend to be reconstructed. Robots understand human requests by engaging in spoken dialog in a specific domain and then set a goal that satisfies the requests. This may be achieved by using spoken dialog to perform tasks such as guiding people to their requested location, providing hospital information, or providing consulting services. The aim of this study is to create a dialog system for hospitals which requires the following:

- Collecting asked questions and responses texts on Taiwan E Hospital into a database.
- Adapting an LSTM-based model for outpatient classification.
- Integrating the classification module into the service robot system.

The remainder of this paper is organized as follows. Section 2 contains survey related works. Section 3 comprises an overview of the methodology. Section 4 consists of a presentation and analysis of the experimental results via a comparison with other algorithms. Section 5 includes a discussion of NLP, LSTM, and optimization methods for comparisons. Section 6 provides conclusions on the study.

## 2. RELATED WORK

### 2.1 Text Classification

Text classification problems are complex in nature and are always characterized by high dimensionality. Problems such as suggesting medical diagnosis [19], patient record

notes [20] and other text documents [21]. Although research on medical texts is still in its infancy, the research interest in this field has increased rapidly in recent years, especially given the development of new technologies, such as machine learning and deep learning. But there are few studies on outpatient classification and database. Therefore, we collected Q&A from Taiwan E Hospital as dataset, and used it to train the proposed model for text classification system.

### 2.2 Text Classification in Machine Learning

Among the existing text classification methods that are emphasized in the previous works. Zheng *et al*. [22] constructed a Chinese web text classification system model based on the Naïve Bayes. Trstenjak *et al*. [23] presented the possibility of using K-Nearest Neighbors (KNN) algorithm with TF-IDF method for text classification according to parameters, measurement and analysis of results. Krishnala *et al*. [24] proposed a system for online news classification based Support Vector Machine (SVM) to extract the keywords from the online news paper content and classify it according to the categories. Although machine learning-based representation models have achieved comparable performance for these, its shortcomings are obvious. These methods only focus on word frequency features and completely ignore the context structure information of the text, thereby making it difficult to capture the semantics of the text. Unlike these models, we uses LSTM to automatically learn the semantic relationships from the contextual words.

### 2.3 Text Classification in Deep Learning

In recent years, there has clearly been a shift in state-of-the-art approaches from statistical machine learning to deep learning based on text categorization models. These are mainly used to develop an end-to-end deep neural network to extract contextual features from the raw text. Pennington *et al*. [25] devised an approach that learns word embedding with comprehensive training of the global word-word co-occurrence of statistical data based on a corpus which shows an interesting linear substructure in word embedding space models like Word2Vec. Tang *et al*. [26] designed a sentiment-based word embedding model by encoding information from texts together with the contexts of words that can distinguish the opposite polarity of words in similar contexts. On the basis of these improved word embedding modules, Kim [27] adopted a convolutional neural network (CNN) architecture for sentence classification which can capture local features from different positions of words in a sentence. However, too many CNN layers will come with a series of problems such as gradient dispersion, gradient explosion, or degradation. Vaswani *et al*. [28] proposed a simple network architecture based solely on an attention mechanism to dispense with recurrence and convolutions entirely. However, it ignores the context in the text. Among the deep learning-based representation models, RNN has been the mainstream research method for text classification due to its ability to naturally model sequential correlation in the text. The aim of this research is to create an outpatient text classification system to facilitate users to retrieve the information they need.

## 3. METHODOLOGY

In this section we describe our proposed method. The system architecture is shown in Fig. 2 and the input, model and output of the system are described as follows.

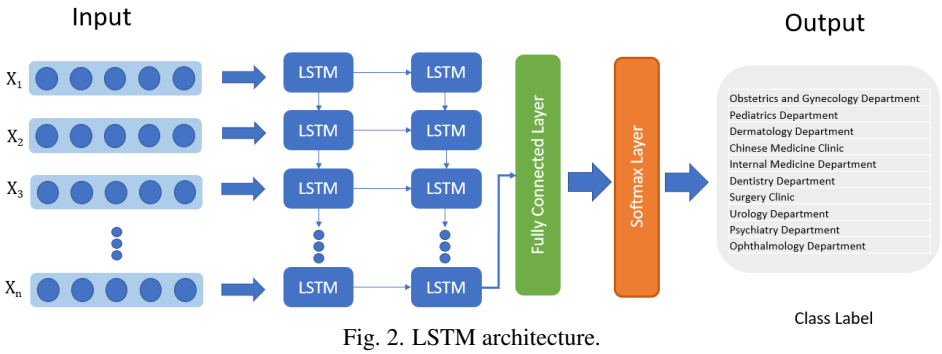


Fig. 2. LSTM architecture.

### 3.1 Input Layer

Before entering the data into the model, we need to do some pre-jobs for segmentation and feature extraction.

The system utillized Jieba [29] as Chinese word segmentation operation. Jieba word segmentation can be exploited to retrieve information in the dataset. Term frequency-inverse document frequency (TF-IDF) is a statistical method used to evaluate the importance of a word to an article or an article in a corpus. The importance of a word increases in proportion to the number of times it appears in the document, but at the same time it decreases in inverse proportion to the frequency of its appearance in the corpus.

$TF$: In a given document, the term frequency (TF) refers to the number of times a given word appears in an article. In practical applications, a long article will give more occurrences. Therefore, we need to normalize the number of times. This equation is as follows:

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

where $n_{i,j}$ is the number of occurrences of a word in a file and $n_{k,j}$ is the sum of the occurrences of all words in the file.

$IDF$: The inverse document frequency is a measure of how much information the word provides. If there are fewer documents containing the keyword, it means that the keyword has good classification ability. This equation is as follows:

$$\mathrm{idf}_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \tag{2}$$

$|D|$ is the total number of files in the corpus. The denominator represents the number of files containing term $t_i$, The product of TF and IDF is then calculated to get the value, as presented in Eq. (3):

$$tf - idf_{i,d} = tf_{i,d} \cdot idf_i. \tag{3}$$

### 3.2 Long Short-Term Memory

Recurrent Neural Networks (RNN) has been handle variable length sequential input. The history record is stored in the cyclic hidden vector, which is the same as the previous hidden vector. LSTM [18] is one of the popular variants of RNN, which reduced the vanishing gradient problem of RNN.

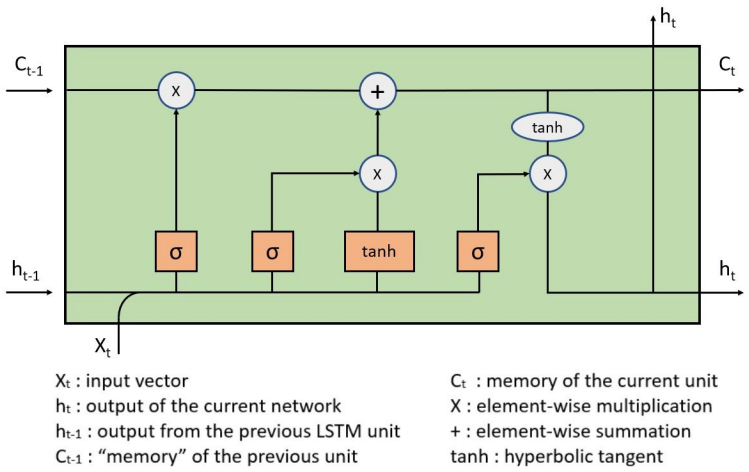


Fig. 3. Structure of long-short term memory (LSTM).

We propose using the LSTM model to develop the outpatient classification, LSTM units has been successfully used to perform sequence learning [30], and used to learn the context and structure in NLP filed. Unlike the traditional recursive unit, the LSTM unit modulates the memory at each step, instead of overwriting the state. This makes it better at exploiting long-range dependencies [31] and finding features in the sequence of sentences. The key component of the LSTM unit is the cell, whose state $C_t$ changes with time, and the LSTM unit decides to modify and add the memory in the cell through sigmoid gates, input gate $i_t$, forget gate $f_t$ and output gate $o_t$. $h_t$ is the signals over the update gate. These updates for the LSTM unit are summarized as follows:

The first step in LSTM is to decide what information we are going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." The LSTM cell is deciding how important is the previous state in the cell $C_{t-1}$ is and we are deciding what will be removed.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{4}$$

Where $\sigma$ is the activation function that ranges from 0 to 1, so that data can be completely removed, partially removed, or completely preserved.

The next step is to record the newly brought in data into the main unit state. This is divided into two steps: deciding what should be recorded, and updating our main unit. A sigmoid layer called the "input gate layer" decides which values we will update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{5}$$

$$\tilde{C}_t = tanh(W_C[h_{t-1}, x_t] + b_C) \tag{6}$$

$\tilde{C}_t$ is a "candidate" hidden state that is computed based on the current input and the previous hidden state. The input gate defines how much of the newly computed state for the current input we wish to let through $h_{t-1}$ is the recurrent connection at the previous hidden layer and current hidden layer, $W$ is the weight matrix connecting the inputs to the current hidden layer, $C$ is the internal memory of the unit. It is a combination of the previous memory. $h_t$ is output hidden state.

The next step to update the previous cell state $C_{t-1}$ into the current cell state $C_t$. We multiply the previous state by $f_t$, forgetting the things we decided to forget earlier. Then we add $i_t * \tilde{C}_t$. This is the new candidate value, scaled according to the size of each state value we decide to update.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{7}$$

The last step is to calculate the output of the LSTM cell. This is performed using the third sigmoid level and additional tanh filter. The output value is based on values in the cell state, but is also filtered by the sigmoid layer. The sigmoid layer essentially determines which parts of the cell state will affect the output value. We pass the cell state value through the tanh filter, and then multiply the output by the third sigmoid level.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t * tanh(C_t) \tag{9}$$

### 3.3 Softmax

The output for the hidden state of the final cell in the LSTM network is the input to a fully connected layer, which uses a basic neural network with one hidden layer to train the output data using the softmax classifier. A simple softmax classifier is used to recognize texts at the last layer. The final result is a probability value, which informs us of the probability that the data will be considered as an outpatient category. The probability is defined by Eq. (10).

$$P = \arg\max_c p(y = c|x) = \arg\max_c \frac{\exp(o_t)}{\sum_{k=1}^{K} O_t} \tag{10}$$

Where $c$ is a class label, $x$ is a sample feature, $y$ is the label variable and $K$ is the number of classes. This decision is made by considering the previous state $h_{t-1}$ and the current input $X_t$.

# 4. EXPERIMENTAL RESULTS

In this section, we introduce our experimental settings, including the hardware, dataset, and baseline algorithm. We then evaluate our design in terms of accuracy. Furthermore, the experiments was compared with other algorithms.

## 4.1 Experimental Environment

We conducted experiments using the webserver on a personal computer with an Intel(R) Core (TM) i9-9700k CPU @ 3.50 GHz and an NVIDIA GEFORCE GTX 1080 Ti graphics card. We set up a deep learning programming environment using Python 3.6 [32], TensorFlow 1.4 [33] and CUDA 9 [34] under the Ubuntu operating system to construct the LSTM. Thus, we realized the deep learning framework directly with Python.

## 4.2 Experimental Datasets

We collected 8 outpatient categories as dataset. The dataset after the capture is reported in Table 2. For the data collection phase, a dataset was obtained from the Taiwan E Hospital and used as training data for the proposed model. The text contains information on user's questions about diseases and the doctor's professional answer. Next, to transform the original test data from the search engine into predefined tested-format data, we applied a series of NLP techniques: specifically, Chinese word segmentation and the elimination of stop words and special symbols. The content of the dialog data from the text reported in Table 1. Each text contains questions asked by the patient, doctor's response, and outpatient category. These collected databases, we open source it on Github.

**Table 1. An example of QA on Taiwan E Hospital website.**

| Outpatient category | Obstetrics and Gynecology |
|---|---|
| Question | Dear physician<br>I want to ask, I have been pregnant for about 14 weeks now.<br>There is a lot of acne on my face.<br>The dermatologist prescribed a cream called "Clindamycin".<br>I wonder if this will affect my pregnancy? |
| Answer | Hello:<br>Clindamycin is a Class B medication.<br>And you are only a small amount externally over a short period,<br>so you can use it with peace of mind during pregnancy.<br><br>Department of Obstetrics and Gynecology, Hsinchu Hospital |

## 4.3 Parameter Setting

We used an optimization algorithm that minimizes the cost function by back-propagating its gradient and updating model parameters. Training was conducted on a GPU-based TensorFlow framework to utilize the parallel computational power of a GPU. The dropout technique was used to avoid overfitting in our model. Although dropout is typically applied to all nodes in a network, we followed the convention of applying

dropout to the connections between layers. The probability of dropping a node during a training iteration is determined by the dropout probability which is a hyper-parameter tuned during training and represents the percentage of units to drop. The parameters used in the experimental setup are summarized in Table 3.

**Table 2. Description of dataset.**

| Outpatient Category | Number of texts |
|---|---|
| Ophthalmology(Oph) | 3672 |
| Urology department(Uro) | 3365 |
| Dentistry(D) | 4312 |
| Medical department(M) | 5153 |
| Surgery(S) | 4523 |
| Orthopedics(Ortho) | 2488 |
| Gynecology(GYN) | 5636 |
| Gastroenterology & Hepatology(G&H) | 6672 |
| Total | 35821 |

**Table 3. Setting of parameters.**

| Parameter | Value |
|---|---|
| Size of input vector | 300 |
| Max features | 150 |
| Number of hidden nodes | 128 |
| Size of batch | 32 |
| Epochs | 50 |
| Learning rate | 0.001 |
| Regularization rate | 0.025 |
| Probability of dropout | 0.2 |
| Activation function | ReLU |
| Optimization | rmsprop |
| Output layer | Softmax |

### 4.4 Comparison with Other Systems

- NB [35]: Naïve Bayes classifier are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle. The parameter is $\alpha$: 0.05.
- SVM [36]: Support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The parameter is kernel: linear.
- KNN [37]: K-nearest neighbor classifier is a supervised learning algorithm that makes predictions without any model training by choosing the number of k nearest neighbors and a distance metric. Finding the k nearest neighbors of the sample that we wished to classify. The parameter is n: 40.
- CNN [27]: Convolutional neural network, the input to NLP tasks are sentences or documents represented as a matrix. Each row of the matrix corresponds to one

token, and each row is a vector that represents a word. The parameters are input dim: 100, filters: 250, activation: ReLU and activation: softmax.

- FastText [38] FastText combines representing sentences with bag of words and bag of n-grams, as well as using subword information, and sharing information across classes through a hidden representation.
- Transformer [28] Transformer is an architecture for transforming one sequence into another one with the help of encoder and decoder and it does not imply any recurrent networks.

### 4.5 Evaluation Settings

To evaluate the system performance, the standard measures of accuracy were used. The corresponding equations are as follows:

- Accuracy: Measures the proportion of correctly predicted labels over all predictions:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

- Precision: Measures the number of true samples out of those classified as positive. The overall precision is the average of the precisions for each class:

$$Precision = \frac{TP}{TP+FP} \tag{12}$$

- Recall: Measures the number of correctly classified samples out of the total samples of a class. The overall recall is the average of the recalls for each class:

$$Recall = \frac{TP}{TP+FN} \tag{13}$$

- F1-score: F1 score is a classifier metric which calculates a mean of precision and recall in a way that emphasizes the lowest value:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

where $TP$ is the overall true positive rate for a classifier on all classes, $TN$ is the overall true negative rate, $FP$ is the overall false positive rate and $FN$ is the overall false negative rate.

### 4.6 Experimental Results

We designed an LSTM-based system to deal with the problem of text classification. The model can be used to learn the weight for each word in a text based on the information of the category where words closely related to the category receive relatively heavy

weighting whereas words that are relatively weak in relation to the category receive lighter weighting. To verify the validity of the model, we compared it with the methods of some baseline systems. Tables 4 and 5 list these models for classification tasks of five-class and eight-class and the results presented in this paper. We implemented machine learning models and deep learning models and compiled their experimental results.

Among the results of machine learning models, the NB and SVM algorithm performed better and reached 94% accuracy and 95% precision in five-class task. Meanwhile, NB and SVM were also performed better in eight-class task. On five-class and eight-class tasks, KNN was particularly bad with 87% and 64% accuracy.

Among the results of deep learning models, LSTM and Transformer have similar accuracy in five-class task. Transformer achieved 95% accuracy in the eight-class task. Compared to the LSTM, LSTM attained an accuracy in 96%. LSTM achieved high accuracy in both five-class and eight-class tasks. This proves that LSTM model is suitable for application in text classification tasks.

**Table 4. Comparison of the different methods on five-class classification.**

| Method | Accuracy | Precise | Recall | F1-Score |
|---|---|---|---|---|
| NB | 94% | 95% | 94% | 94% |
| KNN | 87% | 90% | 87% | 87% |
| SVM | 94% | 95% | 94% | 94% |
| CNN | 93% | 94% | 94% | 94% |
| FastText | 94% | 94% | 94% | 94% |
| Transformer | 95% | 94% | 94% | 94% |
| Proposed | 96% | 96% | 96% | 96% |

**Table 5. Comparison of the different methods on eight-class classification.**

| Method | Accuracy | Precise | Recall | F1-Score |
|---|---|---|---|---|
| NB | 90% | 91% | 90% | 89% |
| KNN | 64% | 78% | 64% | 65% |
| SVM | 90% | 91% | 90% | 90% |
| CNN | 93% | 94% | 94% | 93% |
| FastText | 93% | 94% | 94% | 93% |
| Transformer | 94% | 94% | 94% | 94% |
| Proposed | 95% | 95% | 95% | 94% |

The confusion matrices of each algorithm for five-class and eight-class classification are presented in Figs. 4 and 5. Each column of the confusion matrix represents the prediction category. Each row represents the true attribution category of the data. There are about 100 test texts for each category. The total number of data for each row represents the number of data instances for that category.

It can be seen that the Surgery(S) and Gastroenterology & Hepatology(G&H) departments are often confused in Fig. 5, which may be due to the fact that there are many similar conditions in the two outpatient categories.

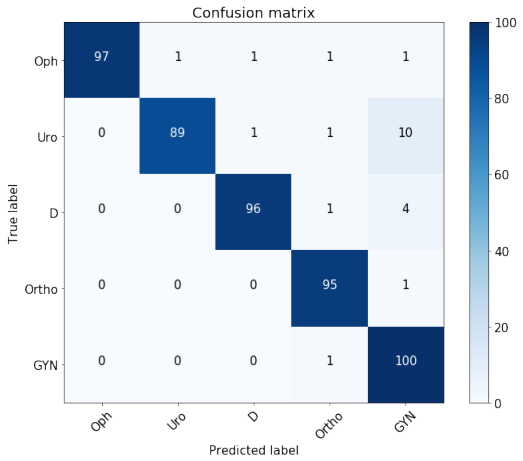


Fig. 4. Confusion matrices of long-short term memory (LSTM) for five-class classification.

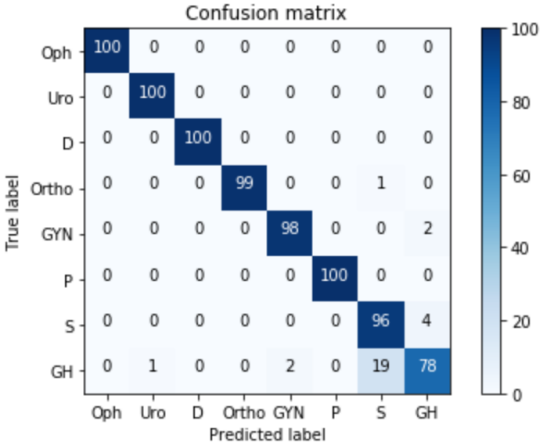


Fig. 5. Confusion matrices of long-short term memory (LSTM) for eight-class classification.

## 5. DISCUSSION

We compared models for five-class and eight-class experimental tasks. As we can see from the results, the performances of the machine learning models were not as good as deep learning models. Although the performances of NB and SVM are excellent, they are still not better than the LSTM model.

The focus of this study was on unstructured data, a discussion of text classification in NLP, and adopting LSTM with TF-IDF to improve semantic cognition and computing. CNN is not completely suitable for learning time series, so it needs various auxiliary processing, and the effect is not necessarily good. Faced with time-sensitive issues and tasks, RNNs are usually more appropriate. LSTM is an excellent variant model of RNN, inheriting the characteristics of most RNN models, and solving the Vanishing Gradient

problem caused by the gradual reduction of the gradient back propagation process. Besides that, Transformer model do not out preform the LSTM in the datasets. We believe this is because the dataset we used were too small. FastText performed similarly in the two experiments. Maybe it performs better in more categories of experiments. Although these models perform very well, LSTM more realistically characterizes or simulates the cognitive processes of human behavior, logical development, and neural organization.

## 6. CONCLUSIONS

We developed a system that focused on outpatient text analysis and differentiation in real-time messaging to give users correct responses to their queries in natural language. The developed system based on LSTM was found to have 96% accuracy. Natural language processing and LSTM model were integrated and used to improve correct outpatient text classification. AI and computational intelligence are key to the success of cognitive computing. We have presented an improved text similarity measurement method which we expect will help to optimize cognitive computing and achieve human-machine interaction via better understanding and analysis of human language. This is meaningful for supporting and improving the development of AI and Health Care.

## REFERENCES

1. Q. Guo and C. Wu, "Fast visual tracking using memory gradient pursuit algorithm," *Journal of Information Science and Engineering*, Vol. 32, 2016, pp. 213-228.
2. S. Inoue and H. Yasuura, "Rfid privacy using user-controllable uniqueness," in *Proceedings of RFID Privacy Workshop*, 2003, pp. 1-9.
3. J. M. Horcas, J. Monteil, M. Bouroche, M. Pinto, L. Fuentes, and S. Clarke, "Context-dependent reconfiguration of autonomous vehicles in mixed traffic," *Journal of Software: Evolution and Process*, Vol. 30, 2018, p. e1926.
4. H.-H. Wu, A. C.-R. Tsai, R. T.-H. Tsai, and J. Y.-J. Hsu, "Building a graded chinese sentiment dictionary based on commonsense knowledge for sentiment analysis of song lyrics," *Journal of Information Science and Engineering*, Vol. 29, 2013, pp. 647-662.
5. M. George, A. Sivan, B. R. Jose, and J. Mathew, "Real-time single-view face detection and face recognition based on aggregate channel feature," *International Journal of Biometrics*, Vol. 11, 2019, pp. 207-221.
6. M. Wang, E. Zhang, and Z. Tang, "Robust speaker verification via rpca under additive noise," *Journal of Information Science and Engineering*, Vol. 35, 2019, pp. 291-305.
7. C.-W. Lee, Y.-L. Wu, and L.-C. Yu, "Combining mutual information and entropy for unknown word extraction from multilingual code-switching sentences," *Journal of Information Science and Engineering*, Vol. 35, 2019, pp. xxx-xxx.
8. Y. Chen, K. Hwang, and W.-S. Ku, "Collaborative detection of ddos attacks over multiple network domains," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, 2007, pp. 1649-1662.

9. S. Pieska, M. Luimula, J. Jauhiainen, and V. Spiz, "Social service robots in wellness and restaurant applications," *Journal of Communication and Computer*, Vol. 10, 2013, pp. 116-123.
10. Q. Li, S. Li, S. Zhang, J. Hu, and J. Hu, "A review of text corpus-based tourism big data mining," *Applied Sciences*, Vol. 9, 2019, p. 3300.
11. C.-H. Cheng, C.-Y. Chen, J.-J. Liang, T.-N. Tsai, C.-Y. Liu, and T.-H. S. Li, "Design and implementation of prototype service robot for shopping in a supermarket," in *Proceedings of IEEE International Conference on Advanced Robotics and Intelligent Systems*, 2017, pp. 46-51.
12. F. Zhao, P. Li, Y. Li, J. Hou, and Y. Li, "Semi-supervised convolutional neural network for law advice online," *Applied Sciences*, Vol. 9, 2019, p. 3617.
13. P. Maia, T. Batista, E. Cavalcante, A. Baffa, F. C. Delicato, P. F. Pires, and A. Zomaya, "A web platform for interconnecting body sensors and improving health care," *Procedia Computer Science*, Vol. 40, 2014, pp. 135-142.
14. S. S. Ghidary, Y. Nakata, H. Saito, M. Hattori, and T. Takamori, "Multi-modal interaction of human and home robot in the context of room map generation," *Autonomous Robots*, Vol. 13, 2002, pp. 169-184.
15. L. Fei, L. Na, and L. Jian, "A new service composition method for service robot based on data-driven mechanism," in *Proceedings of IEEE 9th International Conference on Computer Science and Education*, 2014, pp. 1038-1043.
16. C. Vu, M. Cross, T. Bickmore, A. Gruber, and T. L. Campbell, "Companion robot for personal interaction," 2015, uS Patent 8,935,006.
17. C.-W. Chen, B.-R. Chen, S.-P. Tseng, and J.-F. Wang, "Design and implementation of sentence similarity matching and multimedia feedback for intelligent pharmacy on zenbo robot," in *Proceedings of IEEE International Conference on Orange Technologies*, 2018, pp. 1-4.
18. A. Graves, "Long short-term memory," *Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence*, Vol. 385, Springer, Berlin, 2012, pp. 37-45.
19. A. D. Association, "Midas: an information-extraction approach to medical text classification," *Diabetes Care*, Vol. 42, 2019, pp. S13-S28.
20. J. P. Lalor, H. Wu, L. Chen, K. M. Mazor, and H. Yu, "Comprehenotes, an instrument to assess patient reading comprehension of electronic health record notes: development and validation," *International Journal of Software Engineering and its Applications*, Vol. 20, 2018, p. e139.
21. K. Yi and J. Beheshti, "A hidden markov model-based text classification of medical documents," *Journal of Information Science*, Vol. 35, 2009, pp. 67-81.
22. Z. Gong and T. Yu, "Chinese web text classification system model based on naive bayes," in *Proceedings of International Conference on E-Product E-Service and E-Entertainment*, 2010, pp. 1-4.
23. B. Trstenjak, S. Mikac, and D. Donko, "Knn with tf-idf based framework for text categorization," *Procedia Engineering*, Vol. 69, 2014, pp. 1356-1364.
24. G. Krishnalal, S. B. Rengarajan, and K. Srinivasagan, "A new text mining approach based on hmm-svm for web news classification," *International Journal of Computer Applications*, Vol. 1, 2010, pp. 98-104.

25. J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532-1543.
26. D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, 2015, pp. 496-509.
27. Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.
28. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998-6008.
29. J. Sun, "'jieba' chinese word segmentation tool," 2012.
30. N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *Proceedings of International Conference on Machine Learning*, 2015, pp. 843-852.
31. Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, Vol. 5, 1994, pp. 157-166.
32. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825-2830.
33. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 265-283.
34. S. Ryoo, C. I. Rodrigues, S. S. Baghsorkhi, S. S. Stone, D. B. Kirk, and W.-M. W. Hwu, "Optimization principles and application performance evaluation of a multithreaded gpu using cuda," in *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2008, pp. 73-82.
35. A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, Vol. 752, 1998, pp. 41-48.
36. D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proceedings of the 26th ACM Annual International SIGIR Conference on Research and Development in Informaion Retrieval*, 2003, pp. 26-32.
37. Y. Zhang, S. Peng, and J. Lv, "Improvement and application of tfidf method based on text classification," *Jisuanji Gongcheng / Computer Engineering*, Vol. 32, 2006, pp. 76-78.
38. A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

**Che-Wen Chen** received a B.S. degree from the Industry Engineering Management Department, Yuan Ze University, Taoyuan, Taiwan, in 2008, and an M.S. degree from the Department of Civil Engineering, National Cheng Kung University, Tainan, Taiwan, in 2012. He is currently a Ph.D. candidate in the Department of Electrical Engineering, National Cheng Kung University. His current research interests include deep learning, NLP, robotics, and data science.

**Shih-Pang Tseng** received B.S. and M.S. degrees from the Electrical Engineering Management Department, National Cheng Kung University, Tainan, Taiwan, Taoyuan, Taiwan, and a Ph.D. degree from the Computer Science Engineering Department, National Sun Yatsen University, Kaohsiung, Taiwan. He is a Professor in Changzhou College of Information Technology. His current research interests include deep learning, NLP, robotics, and data science.

**Jhing-Fa Wang** received B.S. and M.S. degrees from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 1973 and 1979, respectively, and a Ph.D. degree from the Department of Computer Science and Electrical Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, in 1983. He is an IEEE Life Fellow, and was the Chair of the IEEE Tainan Section and the Coordinator/Chapter, Region 10, IEEE. He is currently the Chair and a Distinguished Professor in the Department of Electrical Engineering, National Cheng Kung University. He developed a Mandarin speech nition system called Venus-Dictate, which is recognized as a pioneering system in Taiwan. He is currently leading a research group of different disciplines for the development of advanced ubiquitous media for created cyberspace. He has authored nearly 135 journal papers in the IEEE, the Society for Industrial and Applied Mathematics (SIAM), the Institute of Electronics, Information and Communication Engineers, and the Institution of Electrical Engineers, and approximately 220 conference papers.