information

MDPI

Article

Outpatient Text Classification Using Attention-Based Bidirectional LSTM for Robot-Assisted Servicing in Hospital

Che-Wen Chen [1,*], Shih-Pang Tseng [2], Ta-Wen Kuan [3] and Jhing-Fa Wang [1]

[1] Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan; jameswangjf@gmail.com

[2] Software Department, Changzhou College of Information Technology, Changzhou 213164, China; tsengshihpang@ccit.js.cn

[3] School of AI, Guangdong and Taiwan, Foshan University, Foshan 528000, China; gwam.davin@gmail.com

* Correspondence: kfcmax300@gmail.com

Received: 14 January 2020; Accepted: 11 February 2020; Published: 16 February 2020

Abstract: In general, patients who are unwell do not know with which outpatient department they should register, and can only get advice after they are diagnosed by a family doctor. This may cause a waste of time and medical resources. In this paper, we propose an attention-based bidirectional long short-term memory (Att-BiLSTM) model for service robots, which has the ability to classify outpatient categories according to textual content. With the outpatient text classification system, users can talk about their situation to a service robot and the robot can tell them which clinic they should register with. In the implementation of the proposed method, dialog text of users in the Taiwan E Hospital were collected as the training data set. Through natural language processing (NLP), the information in the dialog text was extracted, sorted, and converted to train the long-short term memory (LSTM) deep learning model. Experimental results verify the ability of the robot to respond to questions autonomously through acquired casual knowledge.

Keywords: text classification; health care; service robot; natural language processing

1. Introduction

There has been increasing interest in integrating and applying techniques drawn from the fields of artificial intelligence (AI) and robotics [1], including in vision [2], navigation [3], manipulation [4], emotion 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 health care [8], tourism [9], markets [10], education [1], and at home [11]. With the rapid development of robotic technologies, service robots have gradually entered into and are improving the quality of people's daily lives [12]. Garcia et al. [13] estimated that the increased use of robots has raised the economic growth rates of various countries by approximately 37% on average and found that robots have increased both wages and productivity, along with evidence that they reduce the hours of both low-skilled and middle-skilled workers. Consequently, the scale of the global service robot market has been growing rapidly. In hospitals, shortage of manpower is an important issue. Automatic consultation can reduce manpower and improve service quality. Hence, people need intelligent, safe, and effective service from service robots. A natural way to interact with service robots during the realization of a task is to use speech, where NLP can help to revolutionize information management and retrieval in healthcare settings.

Based on these considerations, the Zenbo Project [14] was launched with the objective of developing high-level cognitive functions for service robots, in order to make them suitable for human–robot interactions in hospitals and to improve health care in our daily lives.

This study was conducted with the objective of developing a service dialog system for a robot, such that it can provide consulting services and perform tasks. A schematic diagram of a user talking to a robot, which then presents the application field, is shown in Figure 1. Users can use natural language to communicate with and command a robot in the hospital environment through the presented human–machine interface. The interface and functions were designed based on a demand survey and are convenient for use in a hospital environment.

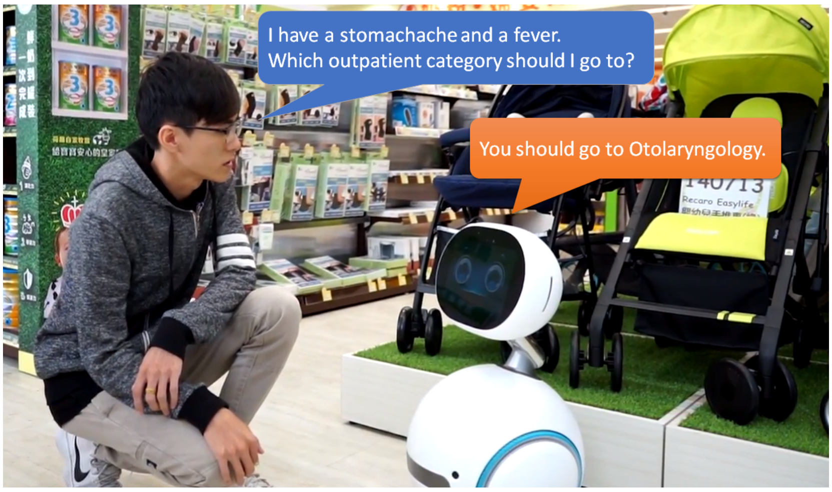

I have a stomachache and a fever. Which outpatient category should I go to?
You should go to Otolaryngology.

Figure 1. Illustration of a conversation between an user and a robot.

In this paper, we present an attention-based bidirectional long short-term memory (Att-BiLSTM) model [15] for the consulting system in the service robot. A flow diagram of the proposed process is shown in Figure 2. With the outpatient text classification system, users can talk about their situation to the service robot and the robot can tell them which clinic they should register with. The aim of this study is to create a outpatient text classification system in a service robot for hospitals, which requires the following:

- Collecting asked questions and response text in hospitals into a database.
- Creating an attention-based bidirectional long-short term memory (LSTM) model for outpatient classification.
- Integrating the classification module into the robot system of a service robot.

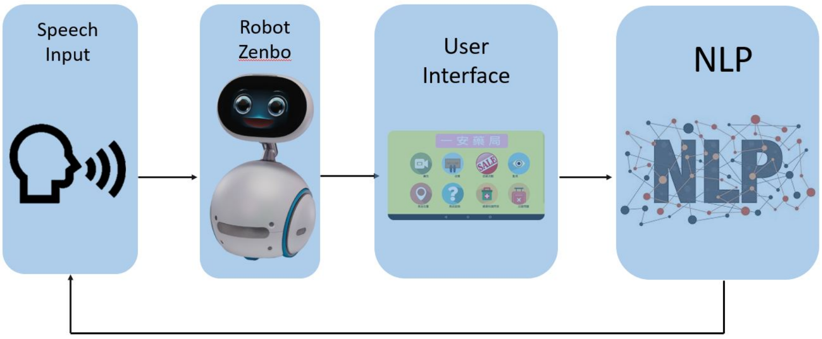


Figure 2. Flow diagram of the proposed process.

The remainder of this paper is organized as follows: Section 2 contains a survey of related works. Section 3 gives an overview of dialog systems. Section 4 provides a description of the architecture of the implemented system based on the service robot. Section 5 consists of a presentation and analysis of the experimental results, including comparison with other algorithms. Section 6 provides the conclusions of the study.

2. Related Work

In this section, we provide an overview of the mainstream representation models for text classification, in terms of knowledge and information. We briefly summarize machine learning-based models in Section 2.1 and deep learning-based models in Section 2.2 .

2.1. Machine Learning-Based Model

Traditional machine learning-based representation models mainly focus on classification algorithms and feature engineering. Specifically, conventional approaches for text analysis use typical features, such as bag-of-words [16], n-grams [17], and term frequency–inverse document frequency (TF–IDF) [18], as the input to machine learning algorithms such as Naïve Bayes classifier (NB) [19], K-nearest Neighbor (KNN) [20], and Support Vector Machine (SVM) [21] for classification. In terms of text classification, text features are mostly designed based on statistical word frequency information of sentiment-related words derived from resources such as lexicons [22]. Zhang et al. [23] presented an improved TF–IDF approach that uses confidence, support, and characteristic words to enhance the recall and precision of text classification. Synonyms defined in the lexicon are also processed in the improved TF–IDF approach. Experiments based on science and technology have given promising results, demonstrating that the new TF–IDF-based approach improves the precision and recall of text classification, compared with the conventional one. Kang et al. [24] proposed the improved performance of an NB classifier for text analysis of restaurant reviews, which can directly affect the text representation capability based on n-grams. Accordingly, it is easy to see how machine learning has become a goldmine for linguistic knowledge, benefiting text classification tasks. Although statistical machine learning-based representation models have achieved comparable performance to these, their shortcomings are obvious. First, these methods only focus on word frequency features and completely ignore the contextual structure information of the text, thereby making it difficult to capture the semantics of the text. Second, the success of these statistical machine learning approaches generally relies heavily on laborious feats of engineering and massive linguistic resources.

2.2. Deep Learning-Based Model

In recent years, there has been a clear shift in state-of-the-art approaches from statistical machine learning to deep learning based on text categorization models [25,26]. These have been mainly used to develop an end-to-end deep neural network to extract contextual features from raw text. Pennington et al. [27] devised an approach that learns a 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 such as Word2Vec. Tang et al. [28] designed a sentiment-based word embedding model by encoding information from text together with the contexts of words, which can distinguish the opposite polarity of words in similar contexts. On the basis of these improved word embedding modules, Kim [29] adopted a convolutional neural network (CNN) architecture for sentence classification, which can capture local features from different positions of words in a sentence. Similarly, Zhang et al. [30] designed a character-level CNN for text classification.

Liu et al. proposed a deep neural network based on a recurrent neural network (RNN) to model text representation for text classification [31]. Among the deep learning-based representation models, RNN has been the mainstream research method for text outpatient classification, due to its ability to naturally model sequential correlation in the text.

Promising results have been achieved by incorporating external knowledge into deep neural networks, but few scholars have classified text describing a person's condition, especially text in

Chinese characters. Our work is in line with these deep learning-based representation models, the major difference being that we incorporate outpatient knowledge as a flexibly integral part of the deep neural network and learn contextual information from different text to generate a powerful outpatient text classification system.

3. Material

In this section, we describe the hardware, robotic system, web server, and experimental environments in this study.

3.1. Robot Hardware

The appearance of the ASUS Zenbo [14] is shown in Figure 3 and Table 1 reports the specifications for the service robot, which is a powerful robot that provides various functionalities, such as time and weather inquiries, storytelling, following the owner to a designated destination, and IoT connectivity. Our dialog system makes the robot more convenient for use in a hospital. The robotic mechanism has good sound reception capability. At an environmental noise level of 70 decibels, the automatic speech recognition (ASR) system can effectively recognize speech for the dialog system to retrieve and generate a response within 1 m of the user.

Table 1. Specifications of the service robot.

Hardware	Specification
Appearance	37 x 37 x 62 cm (L x W x H)
Weight	10 kg
System	Android
Memory	4 GB
Sensors	Ultrasonic ranging sensor Automatic recharge sensor Capacitive touch sensor
Screen	10.1 inch LCD screen
Microphone	Digital microphone
Connection	Wi-Fi 802.11 a/b/g/n/ac 2.4 G/5 GHz, Bluetooth BT4.0

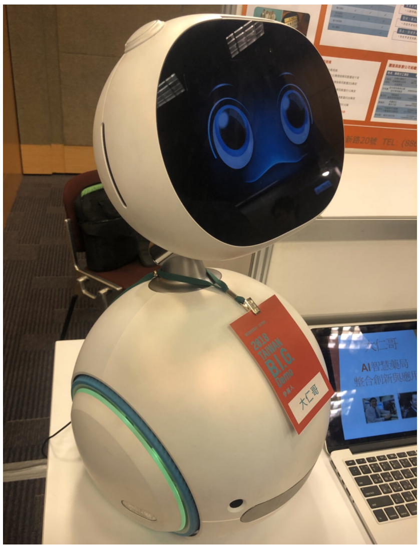


Figure 3. The appearance of the service robot.

3.2. Robot System

We designed an interface for the system that allows users to talk to the robot through natural language. Users can interact with the robot using a touch screen. We also integrated service functions into the system, as depicted in Figure 4. These functions include advertising, drug consultation, an introduction to the hospital, product location search, and disease consultation.

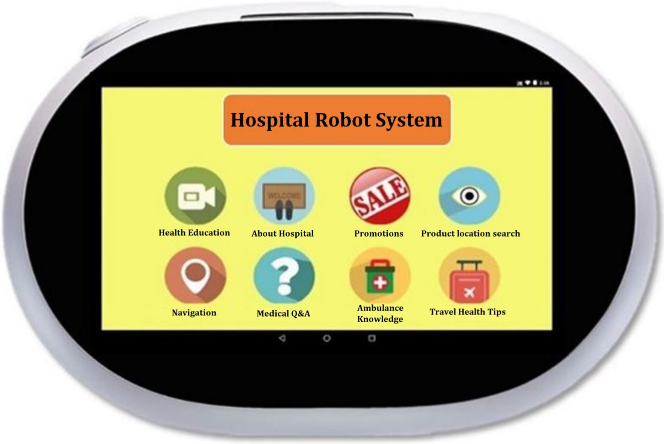


Figure 4. User interface of Zenbo.

We integrated text-to-speech (TTS) and multimedia feedback mechanisms in our robot dialog system, as reported in Table 2. It can also play video and audio as feedback to users and to provide users with more complete information.

Table 2. Feedback actions of the robot system.

Function	Feedback Action
Health Education	Play health care education video
About Hospital	Show information about the hospital
Promotional Activity	Show promotional goods
Product Location Search	Answer questions on the location of drugs and goods
Navigation	Answer questions on map information
Medical QA	Answer medical questions
Ambulance Knowledge	Ambulance knowledge education promotion
Travel Health Tips	Show health information to pay attention to while traveling

3.3. Web Server

We constructed a webserver using Django, which is a free open-source web application framework built on the Python language. The Django framework is shown in Figure 5. When a request is received by HTTP, the event is processed by the corresponding method through the URL. The methods are defined in View files. The output of the method is displayed to the user by HTML and the processed data is read and written to the database. The basic screen design that the user interfaces with is stored in the Template. The Template and its URL are linked, such that the screen displays information according to the URL.

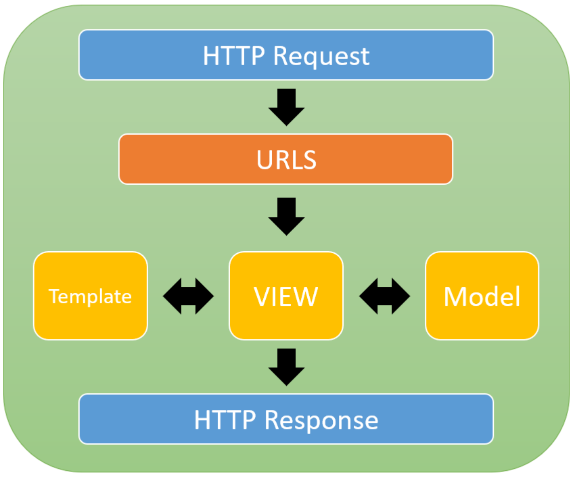


Figure 5. Structure of Django.

3.4. 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 attention-based LSTM. Thus, we realized the deep learning framework directly with Python.

3.5. Dataset

For the data collection phase, a data set was obtained from the Taiwan E Hospital (https://sp1.hso.mohw.gov.tw/doctor/Often_question/) and used as training data for the proposed model. The text contains information on user's questions about diseases and the corresponding professional answers by doctors. Next, to transform the original text 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 is reported in Table 3. In addition, we collected dialog text from the website, the distribution of which is shown in Figure 6.

Table 3. Question and answer example from the Taiwan E Hospital website.

QA	Text content
Question	Hi! Doctor, I have occasionally been dizzy recently, and can't see clearly when I look at things. But I've seen ophthalmology to confirm that the retina is OK. Which department do I need to check for these symptoms? Thank you.
Answer	Hello! According to your description, I suggest you go to the division of Neurology. Changhua Hospital cares about you.

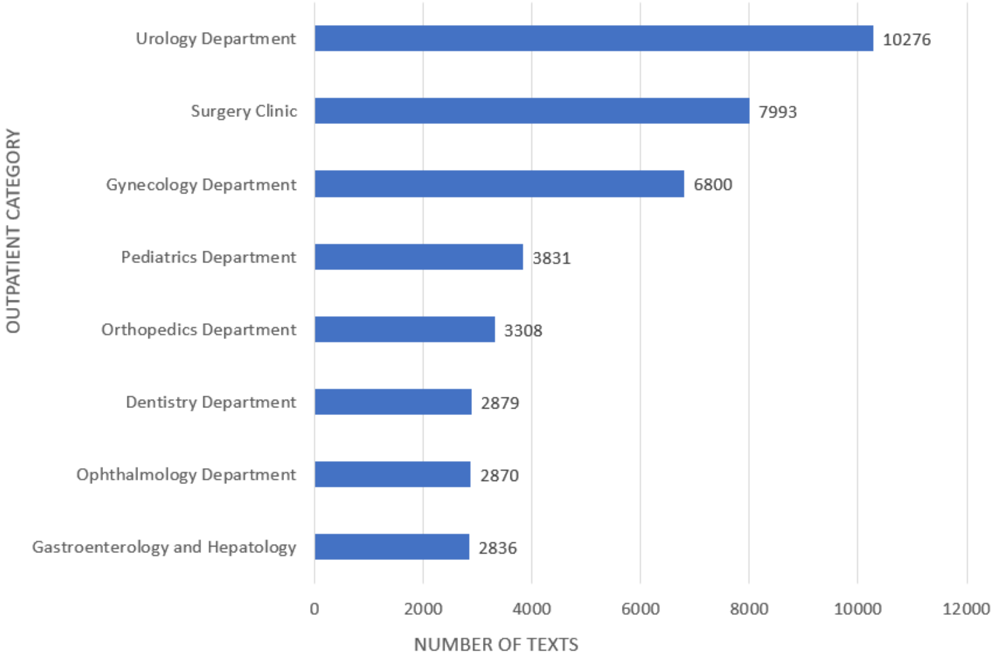


Figure 6. Distribution of the outpatient texts.

4. Methodology

In this section, we describe our proposed method. The system architecture is shown in Figure 7. We collected the existing medical texts from Taiwan E Hospital as our data set. All of these data are real dialog text in Taiwanese. We applied NLP techniques to deal with the data set, in which word segmentation and the elimination of stop words and special symbols were conducted. We performed matrix processing of the processed data and transformed it into a vector space model (VSM) [35]. Next, we entered the question text data set into the proposed attention-based bidirectional LSTM model for training. Finally, we built a text classification system that can classify outpatient categories.

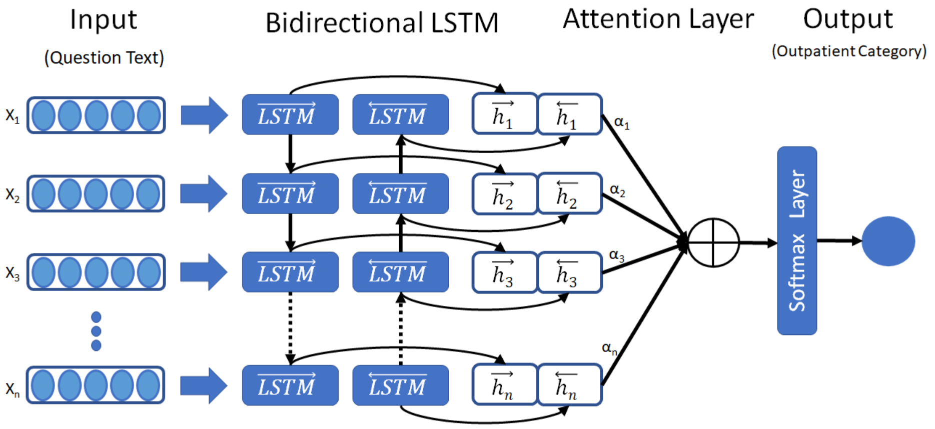


Figure 7. Architecture of the proposed attention-based bidirectional long-short term memory (LSTM) model.

4.1. Pre-Processing

Before entering the data into the model, we needed to conduct some pre-processing for segmentation and feature extraction.

4.1.1. Segmentation

First, the system performs a Chinese word segmentation operation using Jieba [36]. Jieba word segmentation can be exploited to retrieve information and, in particular, to retrieve the critical keywords appearing in a data set. Second, stop words usually refer to the most common words in a language, such as particles, adverbs, and conjunctions. Hence, we adopted the stop word lists provided by Academia Sinica of Taiwan to remove stop words from the data set. Third, as Chinese sentences are always composed of punctuation marks, such as commas, periods, quotation marks, and brackets, we established a list of special symbols, in order to remove punctuation marks and special symbols from the dataset.

4.1.2. TF–IDF

Term frequency–inverse document frequency (TF–IDF) is a statistical method which is used to assess the importance of a word for a file set or one of the files in a corpus. The importance of a word increases proportionally with the number of times it appears in the file, but also decreases inversely with the frequency it appears in the corpus.

TF: In a given document, the term frequency (TF) refers to the number of times a given word appears in the file. This number is usually normalized to prevent it from being biased towards long files. 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. It is the logarithmically scaled inverse fraction of the documents that contain the word. Its equation is as follows:

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

where $|D|$ 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 obtain the TF–IDF value, as presented in Equation (3):

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

4.2. Attention-Based Bidirectional LSTM Model

In this section, the architecture of attention-based bidirectional LSTM neural networks is introduced for the classification of question text.

4.2.1. Long Short-Term Memory

Recurrent neural networks (RNN) have been widely exploited to deal with variable-length sequence input. The long-distance history is stored in a recurrent hidden vector, which is dependent on the immediate previous hidden vector. LSTM [15] is one of the popular variations of RNN, which mitigates the gradient vanishing problem of RNN. Given an input sequence x = $[x_1, x_2, ...x_n]$, where x_t is an E-dimensional word vector in this paper, the hidden vector h_t (with size H) at the time step t is updated as follows.

To learn the outpatient classification, we propose using the LSTM model. First, our model runs through an input sequence to learn one hidden representation, which is the interactive context of a conversation, and then generates the corresponding vectors of the target sequence based on the learned representations. The target sequence is the reverse input sequence, which makes the optimization of our model easier by looking at low-range correlation. The basic building block of our model, the LSTM unit, which has been successfully used to perform sequence learning [37], is used to learn the context and structure in conversations. Unlike traditional recurrent units, the LSTM unit modulates the memory at each step, instead of overwriting the states. This makes it better at exploiting long-range dependencies [38] and discovering long-range features in a sequence of sentences. The key component of the LSTM unit is the cell, which has a state c_t over time, and the LSTM unit decides whether to modify and add the memory in the cell by sigmoid gates: the input gate i_t, forget gate f_t, and output gate o_t. Finally, h_t is the signal over the update gate. These updates for the LSTM unit are summarized as follows: First, the sigmoid layer in an LSTM cell is set at the forget gate level. The LSTM cell decides how important the previous state in the cell C_{t-1} is and, then, decides what new information will be stored in the cell state. This has two parts: First, a sigmoid layer called the "input gate layer" decides which values will be updated. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that may be added to the state. Then, we decide what will be removed. The next step is to update the old cell state C_{t-1} into the new cell state C_t. We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then, we add $i_t * \tilde{C}_t$. This is comprised of the new candidate values, scaled by how much we decided to update each state value. The last step is to calculate the output of the LSTM cell. This is performed using the third sigmoid level and an 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 decides which parts of the cell state will affect the output value. Finally, we put the cell state value through the tanh filter and multiply it by the output of the third sigmoid level. The structure of LSTM is shown in Figure 8 and the formulas can be described as follows:

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

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

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

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

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

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

where σ is the activation function that ranges from 0 to 1 (such that data can be completely removed, partially removed, or completely preserved), $\tilde{C}_t$ is a "candidate" hidden state that is computed based on the current input and the previous hidden state, the input gate i_t 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 (which is a combination of the previous memory), and h_t is the output hidden state.

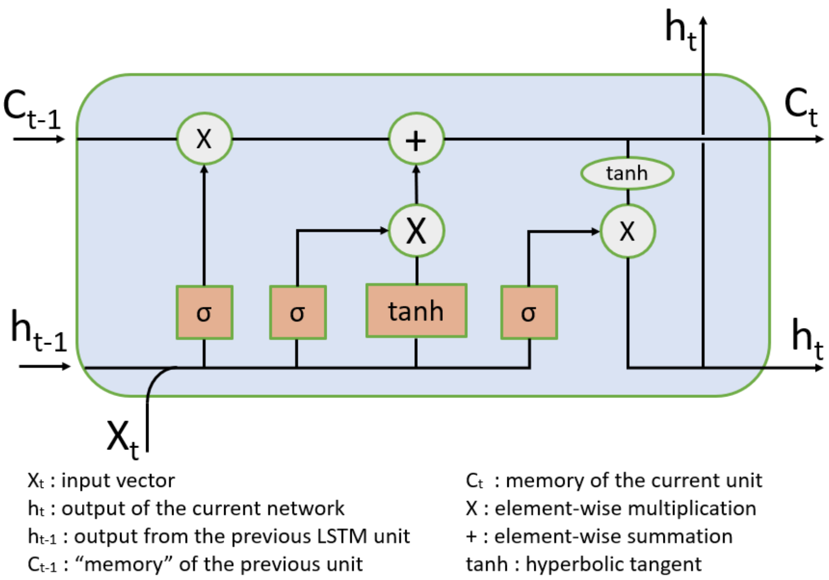

C_{t-1}
h_t
C_t
X
+
tanh
X
X
σ
σ
tanh
σ
h_{t-1}
h_t
X_t
X_t : input vector
h_t : output of the current network
h_{t-1} : output from the previous LSTM unit
C_{t-1} : "memory" of the previous unit
C_t : memory of the current unit
X : element-wise multiplication
+ : element-wise summation
tanh : hyperbolic tangent

Figure 8. The structure of the Long Short-Term Memory (LSTM) neural network.

4.2.2. Bidirectional LSTM

A bidirectional LSTM (BiLSTM) contains two independent LSTMs, which acquire annotations of words by summing up information from the two directions of a sentence and, then, merge the sentimental information in the annotation. Specifically, at each time step t, the forward LSTM calculates the hidden state fh_t based on the previous hidden fh_{t-1} state and the input vector x_t, while the backward LSTM calculates the hidden state bh_t based on the opposite hidden state bh_{t-1} and the input vector x_t. Finally, the vectors of both directions are concatenated as the final hidden state of the BiLSTM model. The two LSTM neural network parameters in BiLSTM networks are independent of each other and share the same word embeddings of the sentence. The final output, h_t, of the BiLSTM model at the step t is as follows: equation:

$$h_t = [fh_t, bh_t]\,. \tag{10}$$

4.2.3. Attention Layer

Recently, attention mechanisms have been developed for word recognition. In this section, we propose an attention mechanism for relation classification tasks. With an attention mechanism, we allow the BiLSTM to decide which part of the text should "attend". The meanings usually relate to different parts of the words; some words in a text can be decisive, while the others are irrelevant. Based on this, an attention mechanism is introduced to attend those informative words and aggregate their representations to form a sentence vector. Based on the above, the LSTM or BiLSTM network will produce a hidden h_t state at each time step. To begin with, the vector h_t is fed into a one-layer Multilayer Perceptron (MLP) to learn a hidden representation u_t. Then, a scalar importance value is computed for h_t, given u_t and a word-level context vector u_w. Finally, the attention-based model computes the weighted mean of the state h_t through a softmax function. The context vector, u_w, can be perceived as a high-level representation for distinguishing the importance of different words. The formulas can be described as follows:

$$u_t = \tanh\left(W_w h_t + b_w\right), \tag{11}$$

$$a_t = \frac{\exp\left(u_t^T u_w\right)}{\sum_t \exp\left(u_t^T u_w\right)}, \tag{12}$$

$$s = \sum_t a_t h_t. \tag{13}$$

4.3. Softmax

The output of the hidden state of the final cell in the LSTM network is used as 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 text 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 Equation (14):

$$P = \arg\max_c p(y = c|x) = \arg\max_c \frac{\exp{(o_t)}}{\sum_{k=1}^{K} O_t}, \tag{14}$$

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 .

5. Experimental Evaluation and Results

In this section, we evaluate the model in the dialog system. We first introduce our experimental settings, including the hardware, data set, and baseline algorithm. Then, we evaluate our design, in terms of accuracy and energy consumption. Furthermore, the experiments compare the recognition results from machine learning and LSTM-based classification methods.

5.1. Experimental Datasets

We collected eight outpatient categories as our data set. The data set after collection is reported in Table 4. The ratio of training set to validation set used was 70%:30%. We randomly selected 100 data from each clinic as the test set.

Table 4. Description of data set.

Outpatient Category	Number of Texts
Ophthalmology (Oph)	2879
Urology department (Uro)	10,276
Dentistry (D)	2870
Pediatrics department (P)	3831
Surgery (S)	7993
Orthopedics (Ortho)	3308
Gynecology (GYN)	6800
Gastroenterology and Hepatology (GandH)	2836
Total	47,093

5.2. Parameter Setting

We conducted an experiment to determine the optimum number of iterations. Among the tested optimizers, the 'Adam' [39] optimizer, which minimizes the cost function by back-propagating its gradient and updating model parameters, performed the best, with an accuracy of 96%; followed by 'RMSprop' with 95.75%, 'Nadam' with 94.5%, and 'Adagrad' with 93.38%. 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 that represents the percentage of units to drop. Adopting the dropout regularization technique led to a significant improvement in performance by preventing overfitting. The gaps between training and testing accuracies, and between training and testing costs, were very small. This indicates that the dropout technique was very effective at forcing the model toward generalization and making it resilient to overfitting. The parameters used in the experimental setup are summarized in Table 5.

Table 5. Experimental parameters.

Parameter	Value
Size of input vector	250
Max features	100
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	Adam
Output layer	Softmax

5.3. Comparison with Other Systems

- NB [19]: Naïve Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm, but a family of algorithms which all share a common principle (i.e., every pair of features being classified is independent of each other). The parameter used was α = 0.05.
- SVM [40]: Support-vector machines are supervised learning models with associated learning algorithms that analyze data, which are used for classification and regression analysis. An SVM model is a representation of the examples as points in space, mapped such that the examples of separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. The parameter used was kernel: linear.
- KNN [41]: The K-nearest neighbor classifier is a supervised learning algorithm which 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, we assigned the class label by majority vote. The parameter used was $n = 40$.
- CNN [29]: In a 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. A CNN is basically a neural-based approach which represents a feature function that is applied to constituting words or n-grams to extract higher-level features. The resulting abstract features have been effectively used in sentiment analysis, machine translation, and question answering, among other tasks. The parameters used were input dim = 100, filters = 250, activation: ReLU, and activation: softmax.

5.4. 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:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{15}$$

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

$$Precision = \frac{TP}{TP + FP}. \tag{16}$$

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

$$Recall = \frac{TP}{TP + FN}. \tag{17}$$

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

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{18}$$

In the above, 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.

5.5. Experimental Results

We designed an attention-based bidirectional LSTM model 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 a lighter weighting. To verify the validity of the model, we compared it with the methods of some baseline systems. Tables 6 and 7 lists these models for five-class and eight-class classification tasks and the results presented in this paper. We implemented machine learning models (NB, KNN, and SVM) and deep learning models (CNN, LSTM, and Att-BiLSTM) and compiled their experimental results.

Table 6. Comparison of the different methods for five-class classification.

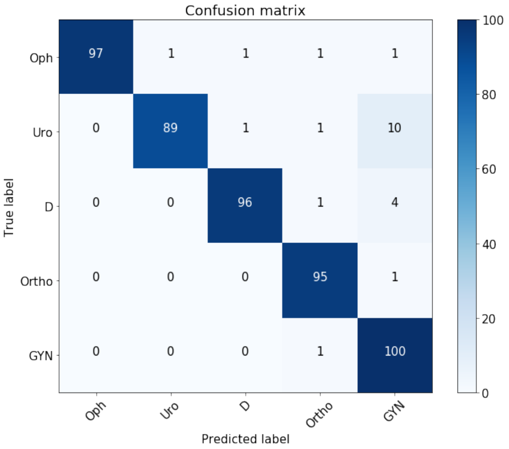
Method	Accuracy	Precise	Recall	F1-Score
NB	94%	95%	94%	94%
KNN	87%	90%	87%	87%
SVM	94%	95%	94%	94%
CNN	93%	94%	94%	94%
LSTM	95%	94%	94%	94%
Att-BiLSTM	96%	96%	96%	96%

Table 7. Comparison of the different methods for 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%
LSTM	95%	94%	94%	94%
Att-BiLSTM	96%	96%	96%	96%

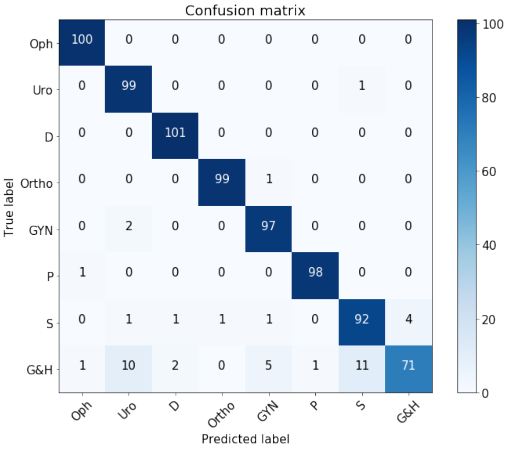
Of the machine learning models, the NB and SVM algorithms performed better, reaching 94% accuracy and 95% precision in the five-class task; they also performed better in the eight-class task. In the five-class and eight-class tasks, KNN was particularly bad, with 87% and 64% accuracy, respectively. Of the deep learning models, LSTM and Att-BiLSTM had similar accuracy in the five-class task. LSTM achieved a 95% accuracy in the eight-class task. Compared to Att-BiLSTM, Att-BiLSTM attained an accuracy of 96%; thus, Att-BiLSTM achieved high accuracy in both five-class and eight-class tasks. This proves that the Att-BiLSTM model is suitable for application in text classification tasks.

The confusion matrices of each algorithm for five-class and eight-class classification are presented in Figures 9, 10 and A1 (Appendix A). Each column of the confusion matrix represents the prediction category, and each row represents the true attribution category of the data. There were 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 was observed that most of the errors were caused by the proximity of the categories Gastroenterology and Hepatology, Urology, and Surgery, which may be due to the fact that there are many similar conditions in these outpatient categories. In many cases, it is hard to differentiate a state of an Gastroenterological/Hepatological illness from a Urological illness. Similarly, patients with urological symptoms could go to Gastroenterology and Hepatology. For example, the category of the text "What medicine should I take for my lower abdominal pain?" was predicted as Urological, whereas its correct label was Gastroenterology and Hepatology. As another misclassification example, the category of the text “I have blood-stained stool, what should I do?" was predicted as Surgery, whereas this text was in the scope of Gastroenterology and Hepatology. If the patient has a bright red blood-stained stool, they should go to Surgery (S). Therefore, it is difficult to classify in the case of insufficient text content information.



	Oph	Uro	D	Ortho	GYN
Oph	97	1	1	1	1
Uro	0	89	1	1	10
D	0	0	96	1	4
Ortho	0	0	0	95	1
GYN	0	0	0	1	100

Figure 9. Confusion matrices of attention-based bidirectional long short-term memory (Att-BiLSTM) for five-class classification.

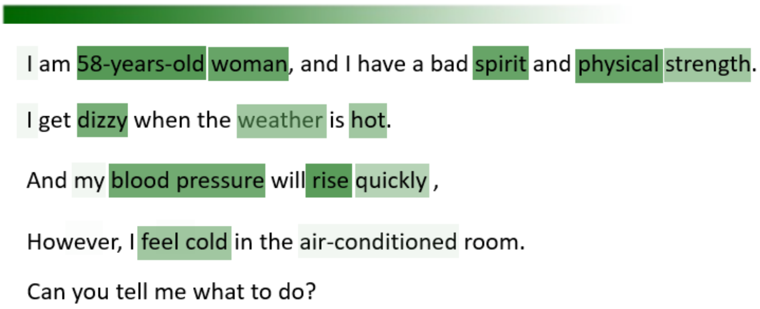


	Oph	Uro	D	Ortho	GYN	P	S	G&H
Oph	100	0	0	0	0	0	0	0
Uro	0	99	0	0	0	0	1	0
D	0	0	101	0	0	0	0	0
Ortho	0	0	0	99	1	0	0	0
GYN	0	2	0	0	97	0	0	0
P	1	0	0	0	0	98	0	0
S	0	1	1	1	1	0	92	4
G&H	1	10	2	0	5	1	11	71

Figure 10. Confusion matrices of Att-BiLSTM for eight-class classification.

5.6. Visualization of Attention

In this study, we utilize an attention mechanism at the word level to distinguish the importance of different words in a text, which improved the classification accuracy. In order to validate the effectiveness of the attention mechanism, we visualized the heatmap of the attention mechanism, at word level, for a document, as shown in Figures 11 and 12, where each line is a sentence in the horizontal direction indicates the word distribution of each sentence. The green bar denotes the weight of word; a darker color indicates higher attention scores, while the lighter part has little importance.



I am 58-years-old woman, and I have a bad spirit and physical strength.

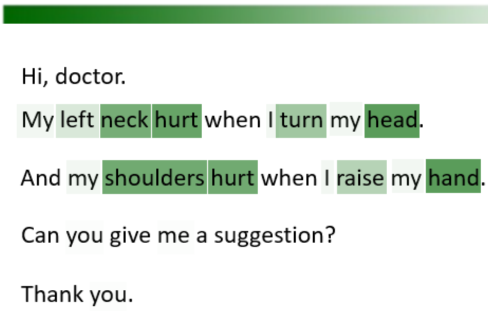
I get dizzy when the weather is hot.

And my blood pressure will rise quickly ,

However, I feel cold in the air-conditioned room.

Can you tell me what to do?

Figure 11. Attention visualization for a document labeled Gynecology.



Hi, doctor.

My left neck hurt when I turn my head.

And my shoulders hurt when I raise my hand.

Can you give me a suggestion?

Thank you.

Figure 12. Attention visualization for a document labeled Orthopedics.

We can observe that our model successfully distinguishes the importance of words. For example, words carrying much sentiment information, such as "Woman", "dizzy", and "shoulders hurt" had higher attention weights than other words. For text with a lot of words containing little sentiment information, such as "cold", "turn", "left", and "rasie", lower attention weights were assigned than other words in the document. The results show that our model can effectively put more focus on important words.

6. Conclusions

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. We developed a system focused on outpatient text analysis and differentiation in messaging to give users correct responses to their queries. Natural language processing and the Att-BiLSTM model were integrated and used to improve correct outpatient text classification. We expect that it will help to optimize cognitive computing and achieve human–machine interactions through better understanding and analysis of human language. We compared our presented model against established models for five-class and eight-class experimental tasks. As we can see from the results, the performance of machine learning models was not as good as that achieved by deep learning models. The developed system based on Att-BiLSTM was found to have 96% accuracy. Although the performances of NB and SVM reached 94%, they were still outperformed by the Att-BiLSTM model. AI and computational intelligence are key to the success of cognitive computing. Finally, we built a dialog interface for a hospital service robot, in order to improve the usability of the proposed system, such that it can provide consulting services and perform tasks. With the outpatient text classification system, users can talk about their situation to the service robot and the robot can tell them which clinic they should register with, which leads to better time efficiency and less manual effort. This is meaningful for supporting and improving the development of AI in health care applications. In future work, we will optimize the model and build a dialog system on multiple platforms, which can take advantage of the effectiveness of the dialog system in a hospital service robot.

Author Contributions: Conceptualization, J.-F.W.; Data curation, S.-P.T.; Formal analysis, S.-P.T.; Investigation, C.-W.C.; Methodology, C.-W.C.; Resources, J.-F.W.; Software, C.-W.C.; Supervision, J.-F.W.; Validation, C.-W.C. and T.-W.K.; Visualization, C.-W.C.; Writing—review & editing, S.-P.T., J.-F.W. and T.-W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: Thanks to ASUS's technical co-operation, the field verification of the Yian Pharmacy Bureau and the database collection consultation of Kaohsiung Veterans General Hospital.

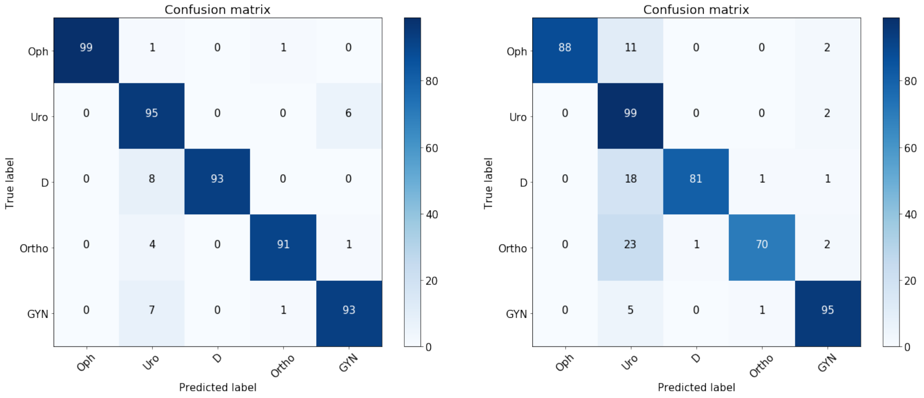
Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ASR	Automatic Speech Recognition
TF–IDF	Term Frequency–Inverse Document Frequency
TTS	Text-To-Speech
NB	Naïve Bayes
SVM	Support-Vector Machine
KNN	K-Nearest Neighbor
CNN	Convolutional Neural Network
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
Oph	Ophthalmology
Uro	Urology department
D	Dentistry
P	Pediatrics department
S	Surgery
Ortho	Orthopedics
GYN	Gynecology
GandH	Gastroenterology and Hepatology

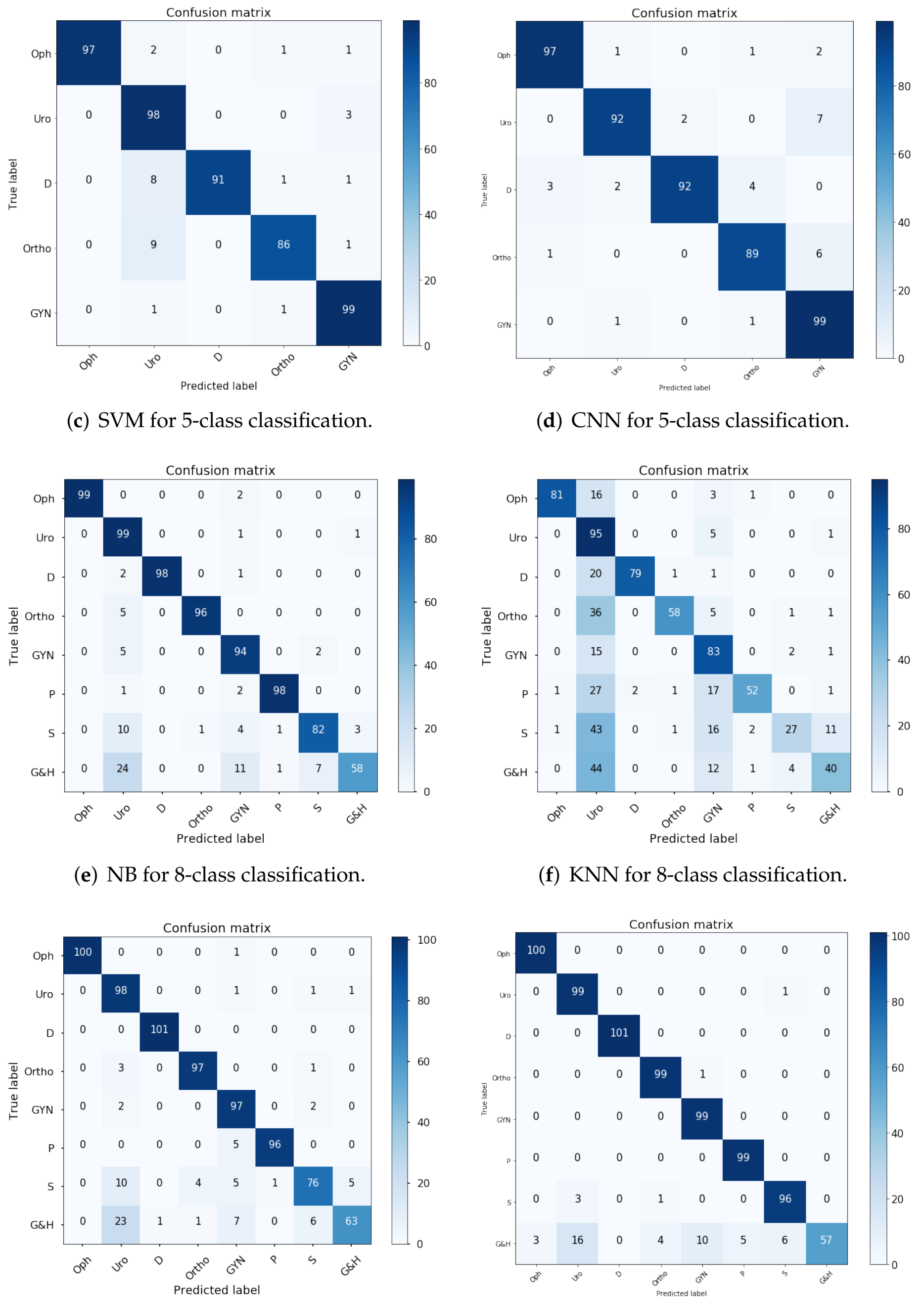
Appendix A



(**a**) NB for 5-class classification.

(**b**) KNN for 8-class classification.

Figure A1. *Cont.*



	Oph	Uro	D	Ortho	GYN
Oph	97	2	0	1	1
Uro	0	98	0	0	3
D	0	8	91	1	1
Ortho	0	9	0	86	1
GYN	0	1	0	1	99

(**c**) SVM for 5-class classification.

	Oph	Uro	D	Ortho	GYN
Oph	97	1	0	1	2
Uro	0	92	2	0	7
D	3	2	92	4	0
Ortho	1	0	0	89	6
GYN	0	1	0	1	99

(**d**) CNN for 5-class classification.

	Oph	Uro	D	Ortho	GYN	P	S	G&H
Oph	99	0	0	0	2	0	0	0
Uro	0	99	0	0	1	0	0	1
D	0	2	98	0	1	0	0	0
Ortho	0	5	0	96	0	0	0	0
GYN	0	5	0	0	94	0	2	0
P	0	1	0	0	2	98	0	0
S	0	10	0	1	4	1	82	3
G&H	0	24	0	0	11	1	7	58

(**e**) NB for 8-class classification.

	Oph	Uro	D	Ortho	GYN	P	S	G&H
Oph	81	16	0	0	3	1	0	0
Uro	0	95	0	0	5	0	0	1
D	0	20	79	1	1	0	0	0
Ortho	0	36	0	58	5	0	1	1
GYN	0	15	0	0	83	0	2	1
P	1	27	2	1	17	52	0	1
S	1	43	0	1	16	2	27	11
G&H	0	44	0	0	12	1	4	40

(**f**) KNN for 8-class classification.

	Oph	Uro	D	Ortho	GYN	P	S	G&H
Oph	100	0	0	0	1	0	0	0
Uro	0	98	0	0	1	0	1	1
D	0	0	101	0	0	0	0	0
Ortho	0	3	0	97	0	0	1	0
GYN	0	2	0	0	97	0	2	0
P	0	0	0	0	5	96	0	0
S	0	10	0	4	5	1	76	5
G&H	0	23	1	1	7	0	6	63

(**g**) SVM for 8-class classification.

	Oph	Uro	D	Ortho	GYN	P	S	G&H
Oph	100	0	0	0	0	0	0	0
Uro	0	99	0	0	0	0	1	0
D	0	0	101	0	0	0	0	0
Ortho	0	0	0	99	1	0	0	0
GYN	0	0	0	0	99	0	0	0
P	0	0	0	0	0	99	0	0
S	0	3	0	1	0	0	96	0
G&H	3	16	0	4	10	5	6	57

(**h**) CNN for 8-class classification.

Figure A1. Confusion matrices for the different text classification methods.

References

1. Tzafestas, S. Roboethics: Fundamental concepts and future prospects. *Information* **2018**, *9*, 148. [CrossRef]
2. Ju, M.; Luo, H.; Wang, Z.; Hui, B.; Chang, Z. The Application of Improved YOLO V3 in Multi-Scale Target Detection. *Appl. Sci.* **2019**, *9*, 3775. [CrossRef]
3. Wang, H.; Zhou, Z. A Heuristic Elastic Particle Swarm Optimization Algorithm for Robot Path Planning. *Information* **2019**, *10*, 99. [CrossRef]
4. Batsuren, K.; Yun, D. Soft robotic gripper with chambered fingers for performing in-hand manipulation. *Appl. Sci.* **2019**, *9*, 2967. [CrossRef]
5. Lee, M.S.; Lee, Y.K.; Pae, D.S.; Lim, M.T.; Kim, D.W.; Kang, T.K. Fast Emotion Recognition Based on Single Pulse PPG Signal with Convolutional Neural Network. *Appl. Sci.* **2019**, *9*, 3355. [CrossRef]

6. Badenhorst, J.; De Wet, F. The usefulness of imperfect speech data for ASR development in low-resource languages. *Information* **2019**, *10*, 268. [CrossRef]
7. Russell, S.J.; Norvig, P. Artificial Intelligence: A Modern Approach. Available online: https://ugeb.pw/30195311.pdf (accessed on 14 February 2020).
8. Santosh, K. Speech Processing in Healthcare: Can We Integrate? Available online: https://www.sciencedirect.com/science/article/pii/B9780128181300000015 (accessed on 12 February 2020).
9. Li, Q.; Li, S.; Zhang, S.; Hu, J.; Hu, J. A Review of Text Corpus-Based Tourism Big Data Mining. *Appl. Sci.* **2019**, *9*, 3300. [CrossRef]
10. Cheng, C.H.; Chen, C.Y.; Liang, J.J.; Tsai, T.N.; Liu, C.Y.; Li, T.H.S. Design and implementation of prototype service robot for shopping in a supermarket. In Proceedings of the 2017 International Conference on Advanced Robotics and Intelligent Systems (ARIS), Taipei, Taiwan, 6–8 September 2017; pp. 46–51.
11. Massaro, A.; Maritati, V.; Savino, N.; Galiano, A.; Convertini, D.; De Fonte, E.; Di Muro, M. A Study of a Health Resources Management Platform Integrating Neural Networks and DSS Telemedicine for Homecare Assistance. *Information* **2018**, *9*, 176. [CrossRef]
12. Fei, L.; Na, L.; Jian, L. A new service composition method for service robot based on data-driven mechanism. In proceedings of the 2014 9th International Conference on Computer Science and Education, Vancouver, Canada, 22–24 August 2014; pp. 1038–1043.
13. Garcia, E.; Jimenez, M.A.; De Santos, P.G.; Armada, M. The evolution of robotics research. *IEEE Robot. Autom. Mag.* **2007**, *14*, 90–103. [CrossRef]
14. ASUS. Zenbo: Your Smart Little Companion. Available online: https://zenbo.asus.com/tw/ (accessed on 12 February 2020).
15. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
16. Wallach, H.M. Topic modeling: Beyond bag-of-words. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25 June 2006; pp. 977–984.
17. Damashek, M. Gauging similarity with n-grams: Language-independent categorization of text. *Science* **1995**, *267*, 843–848. [CrossRef] [PubMed]
18. Joachims, T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. Available online: https://apps.dtic.mil/docs/citations/ADA307731 (accessed on 12 February 2020).
19. McCallum, A.; Nigam, K. A Comparison of Event Models for Naive Bayes Text Classification. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.9324&rep=rep1&type=pdf (accessed on 12 February 2020).
20. Trstenjak, B.; Mikac, S.; Donko, D. KNN with TF-IDF based framework for text categorization. *Procedia Eng.* **2014**, *69*, 1356–1364. [CrossRef]
21. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98*; Nédellec, C., Rouveirol, C.; Springer: Berlin/Heidelberg, Germany, 1998.
22. Melville, P.; Gryc, W.; Lawrence, R.D. Sentiment analysis of blogs by combining lexical knowledge with text classification. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 1275–1284.
23. Zhang, Y.; Gong, L.; Wang, Y. An improved TF-IDF approach for text classification. *J. Zhejiang-Univ.-Sci. A* **2005**, *6*, 49–55. [CrossRef]
24. Kang, H.; Yoo, S.J.; Han, D. Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Syst. Appl.* **2012**, *39*, 6000–6010. [CrossRef]
25. Johnson, R.; Zhang, T. Semi-Supervised Convolutional Neural Networks for Text Categorization via Region Embedding. Available online: https://papers.nips.cc/paper/5849-semi-supervised-convolutional-neural (accessed on 12 February 2020).
26. Johnson, R.; Zhang, T. Supervised and semi-supervised text categorization using LSTM for region embeddings. *arXiv* **2016**, arXiv:1602.02373.
27. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. Available online: https://www.aclweb.org/anthology/D14-1162.pdf (accessed on 12 February 2020).
28. Tang, D.; Wei, F.; Qin, B.; Yang, N.; Liu, T.; Zhou, M. Sentiment embeddings with applications to sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 496–509. [CrossRef]
29. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.

30. Zhou, C.; Sun, C.; Liu, Z.; Lau, F. A C-LSTM neural network for text classification. *arXiv* **2015**, arXiv:1511.08630.
31. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. *arXiv Prepr.* **2016**, arXiv:1605.05101.
32. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
33. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A System for Large-Scale Machine Learning. Available online: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi (accessed on 12 February 2020).
34. Ryoo, S.; Rodrigues, C.I.; Baghsorkhi, S.S.; Stone, S.S.; Kirk, D.B.; Hwu, W.W. 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, Salt Lake City, UT, USA, 20 February 2008; pp. 73–82.
35. Salton, G.; Wong, A.; Yang, C.S. A vector space model for automatic indexing. *Commun. ACM* **1975**, *18*, 613–620. [CrossRef]
36. Sun, J. 'Jieba' Chinese Word Segmentation Tool. Available online: https://github.com/fxsjy/jieba (accessed on 12 February 2020).
37. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised Learning of Video Representations Using LSTMs. Available online: https://proceedings.mlr.press/v37/srivastava15.pdf (accessed on 12 February 2020).
38. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
40. Zhang, D.; Lee, W.S. Question classification using support vector machines. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, ON, Canada, 28 July–1 August 2003; pp. 26–32.
41. Zhang, Y.; Peng, S.; Lv, J. Improvement and application of TFIDF method based on text classification. *Jisuanji Gongcheng/Comput. Eng.* **2006**, *32*, 76–78.