applied sciences

Article

AidIR: An Interactive Dialog System to Aid Disease Information Retrieval

Da-Jinn Wang [1], Tsong-Yi Chen [2] and Chia-Yi Su [2,]*

[1] Department of Business Computing, National Kaohsiung University of Science and Technology, Kaohsiung 81157, Taiwan; wangdaj@nkust.edu.tw
[2] Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 80778, Taiwan; chentso@nkust.edu.tw
* Correspondence: f109152134@nkust.edu.tw

Abstract: This paper proposes an interactive dialog system, called AidIR, to aid information retrieval. AidIR allows users to retrieve information on diseases resulting from coronaviruses and diseases transmitted by vector mosquitoes with natural language interaction and Line chat media. In a subjective evaluation, we asked 20 users to rate the intuitiveness, usability, and user experience of AidIR with a range between −2 and 2. Moreover, we also asked these users to answer yes–no questions to evaluate AidIR and provide feedback. The average scores of intuitiveness, usability, and user experience are 0.8, 0.8, and 1.05, respectively. The yes–no questions demonstrated that AidIR is better than systems using the graphical user interface in mobile phones and single-turn dialog systems. According to user feedback, AidIR is more convenient for information retrieval. Moreover, we designed a new loss function to jointly train a BERT model for domain classification and sequence label tasks. The accuracy of both tasks is 92%. Finally, we trained the dialog policy network with supervised learning tasks and deployed the reinforcement learning algorithm to allow AidIR to continue learning the dialog policy.

Keywords: interactive dialog system; human–computer interaction; natural language processing; transfer learning; reinforcement learning

check for updates

Citation: Wang, D.-J.; Chen, T.-Y.; Su, C.-Y. AidIR: An Interactive Dialog System to Aid Disease Information Retrieval. *Appl. Sci.* **2022**, *12*, 1875. https://doi.org/10.3390/app12041875

Academic Editors: Yuen-Hsien Tseng, Lung-Hao Lee and Po-Lei Lee

Received: 21 December 2021
Accepted: 7 February 2022
Published: 11 February 2022

1. Introduction

According to the statistics of We Are Social [1], nearly 66.6% of the population are unique mobile users. However, only 59.5% of the population are unique Internet users. This phenomenon indicates that some people may be unique mobile users but they are not Internet users. Considering the large number of unique mobile users, automating tasks through mobile phones has become a crucial topic.

The COVID-19 pandemic has expedited the diffusion of misinformation across social media [2]. The spread of misinformation results in negative consequences in society [3]. For example, misinformation concerning the COVID-19 vaccine diminishes people's willingness to receive the vaccine [4]. Furthermore, the dispersion of misinformation increases the risk of mental health problems such as depression [5]. To reduce the impact of misinformation, an intuitive system that enables users to easily retrieve medical information from reliable resources is crucial to research.

Current systems using graphical user interface (GUI) [6] have multiple disadvantages. First, current systems using GUI cannot guide users to specify keywords. This system will return as many articles as possible. Moreover, this system cannot utilize the information that the user has provided. That is, users are usually required to provide the information that they have provided again. This step is redundant and not intuitive for end-users. Moreover, systems with conventional GUI do not exploit the context to retrieve the information. Therefore, the information will sometimes not relate to the information that users want to retrieve. Finally, users cannot use the Chinese language to retrieve information with this system. Hence, a more intuitive system that allows end-users to use the Chinese language to retrieve information is crucial to research.

Current dialog systems such as Google Assistant and Microsoft Cortana cannot entirely address the problems of current systems using GUI. These systems usually focus on automating tasks in mobile phones and computers. For example, users can ask Google Assistant to make a phone call or open Apps. However, when users request specific information, Google Assistant usually only provides users with the website. Moreover, users cannot rectify errors when Google Assistant makes mistakes. Users must restart the dialog when Google Assistant makes mistakes. Hence, a system that precisely returns the information that users request and allows users to rectify the errors will enhance their intuitiveness, usability, and user experiences.

In summary, according to the world phenomenon, the number of unique mobile users is greater than that of unique Internet users. Furthermore, the spread of misinformation gives rise to harmful effects in society such as mental health problems and a reduction in the willingness to be vaccinated. To enable unique mobile users to more easily retrieve information and reduce the damage that misinformation inflicts upon society, an intuitive system that enables users to more easily retrieve information must be developed and researched. However, the usability, intuitiveness, and user experiences of current dialog systems and systems using GUI still have much to improve upon. To address their issues, an intuitive information retrieval system in mobile phones is studied in this paper. Since diseases related to coronaviruses and diseases transmitted by mosquitoes have been the most relevant in recent years, we mainly aim to empower users to retrieve information on those two types of disease in AidIR. To make AidIR more intuitive, we also deploy AidIR onto Line chat medium because Line chat medium is the most dominant chat medium in Aisa [7]. Therefore, the main contributions of this paper are as follows:

- Enabling users to retrieve information on diseases corresponding to coronaviruses and diseases transmitted by vector mosquitoes with natural language interaction;
- Enabling users to retrieve information with Line, a well-known chat medium in Asia;
- Fine-tuning the pre-trained BERT model [8] with the mask LM task and disease corpora;
- Proposing a weighting method to jointly train a BERT model [9] for domain classification and sequence labeling;
- Inventing regular expressions to match some types of numbers in the sentence and normalizing those numbers with special tokens;
- Allowing the system to continue learning the dialog policy online.

The organization of this paper is as follows. Section 2 will introduce task-oriented dialog systems. Section 3 will explain the architecture of AidIR. Then, the experimental results are mentioned and dialog samples are demonstrated in Sections 4 and 5. Finally, we conclude this paper and discuss future works in Sections 6 and 7.

2. Task-Oriented Dialog System

Task-oriented dialog systems are the backbone of AidIR. This architecture enables AidIR to grasp the meaning of user dialog and allows users to rectify AidIR's misunderstandings. A task-oriented dialog system [10–13] encompasses natural language understanding (NLU), dialog management (DM), and natural language generation (NLG). The operation of task-oriented dialog systems is demonstrated in Figure 1. NLU allows the system to understand the user dialog and extract the vital information in the user dialog. Because NLU is not robust, DM is employed to manage the state of the dialog systems and track the status of the dialog so that the system can address the misunderstanding of NLU models. Finally, NLG will generate the response based on the information in the DM. In AidIR, we generate the response by using the template-based method. We will introduce the response templates in Section 3.4.

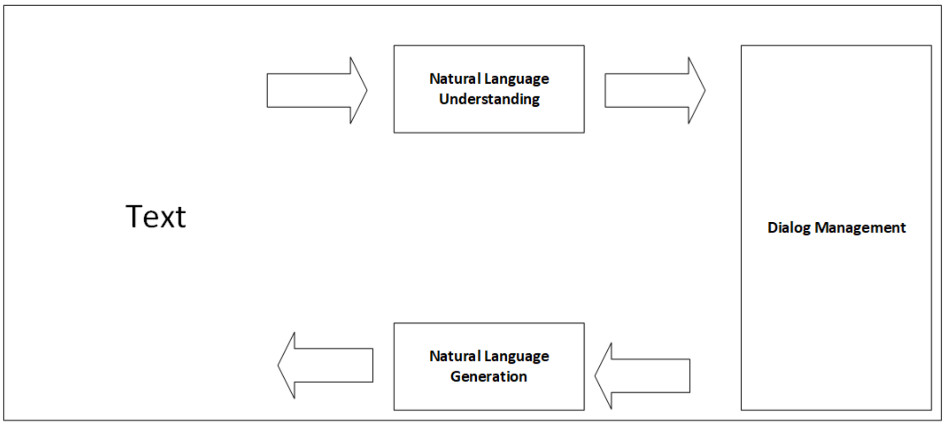


Figure 1. Operation of the task-oriented dialog system.

2.1. Natural Language Understanding (NLU)

NLU allows the system to understand the user dialog and extract the vital information in the user dialog. In task-oriented dialog systems, NLU is the first component. Furthermore, NLU can be further divided into three components: domain classification, intent detection, and sequence labeling. Domain classification enables the system to understand the types of information that the user wants to retrieve so that dialog systems can find the right DataTable for information retrieval. Intent detection detects the intent of the dialog. This enables dialog systems to perform the corresponding task. Sequence labeling extracts the information in the dialog. Dialog information extraction allows dialog systems to know the information that users provide. In AidIR, we jointly trained a BERT model [9] to simultaneously perform the domain classification and sequence label tasks to reduce the number of models. We employed the naive Bayes algorithm [14] to detect the user intent. We listed all of the intents in AidIR in Table 1.

Table 1. Intents in AIDIR.

Types of Intents	Description
Disease	Enable users to retrieve types of diseases
Pathogen	Enable users to retrieve pathogen of diseases
Symptom	Enable users to retrieve symptom of diseases
Transmission	Enable users to retrieve how the disease is transmitted

2.2. Dialog Management (DM)

Since the NLU is prone to misunderstand user dialog, DM enables dialog systems to address the mistakes that the NLU makes. DM includes dialog state tracking (DST) and dialog policy making systems. DST records the history and the current state of the dialog. This information allows the dialog systems to track the current dialog state so that the dialog system can make the right policy. The structure of the DST is usually a dictionary structure, which includes a key and its related value. Furthermore, the information in the DST allows dialog policy-making systems to determine the right dialog policy. We will elucidate the implementation of DM in Section 3.3.

3. Architecture of AidIR

The architecture of AidIR is demonstrated in Figure 2. First, we pre-process the corpora. In the pre-processing stage, we operationalize the taxonomy to decide the domains of dialog, replace some numbers with the special tokens to train a general model instead of a specific model, and use WordPiece tokenization to tokenize the words into sub-word level tokens. This stage enables the ensuing models to more smoothly process the sentences. The pre-processing of the corpora is discussed in Section 3.1. After pre-processing, we invent a new method to train a BERT model to perform domain classification and sequence label simultaneously. In addition, we utilized the Naive Bayes algorithm to classify the intent of the dialog. These models enable AidIR to grasp the meaning of the user dialog and extract the crucial information in the user dialog. Section 3.2 mentions the model for domain classification and sequence label tasks. To enable AidIR to deal with the mistakes of NLU models and track the status of the dialog, we first classify the dialog act of the user to enable AidIR to know the current dialog act of the user. This step allows the system to know whether users rectify the error of the system. Then, we construct the state vector with the output of NLU models and the dialog act classification model for the dialog policy making system. The dialog policy-making system will determine the action of the dialog for AidIR based on the current dialog state. Moreover, we combine a simple neural network and some rules to make the dialog policy based on the information in the DST. Section 3.3 will elucidate the DM in AidIR. To generate the response based on the dialog policy, we define some response templates for each dialog policy. Section 3.4 introduces the response templates of AidIR. Finally, we discuss how AidIR learns the dialog policy online in Section 3.5 to enable AidIR to continue learning the dialog policy. The detailed implementation of AidIR is discussed in Section 3.6.

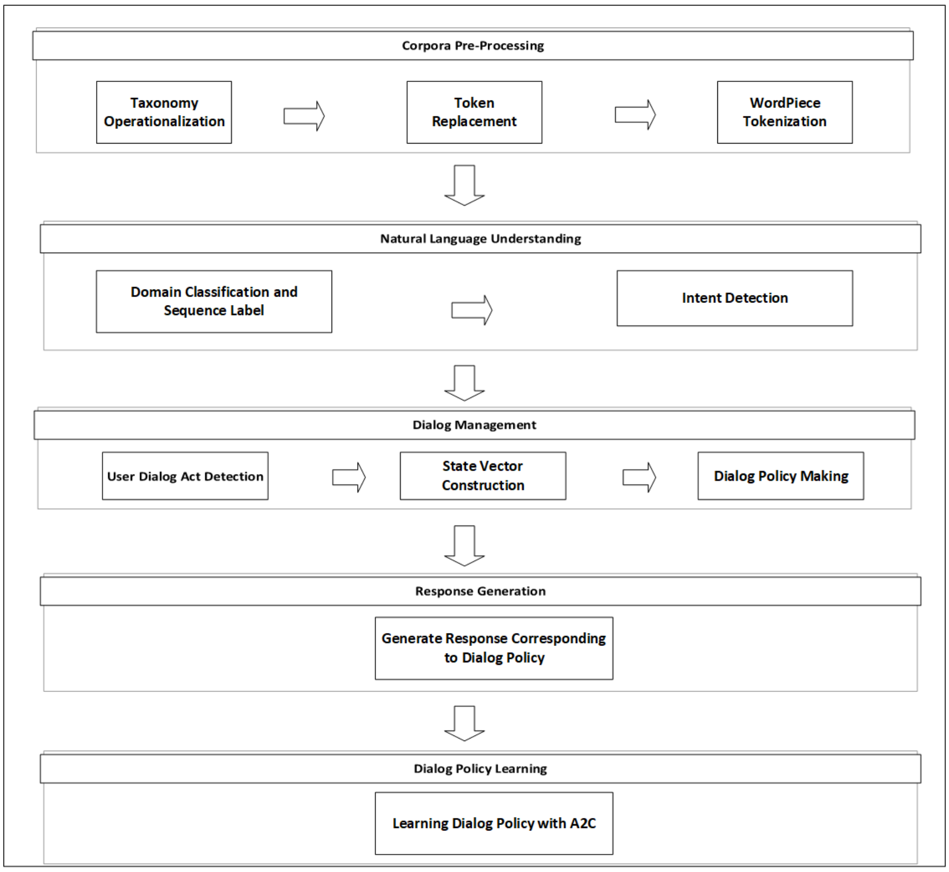

Corpora Pre-Processing
Taxonomy Operationalization
Token Replacement
WordPiece Tokenization
Natural Language Understanding
Domain Classification and Sequence Label
Intent Detection
Dialog Management
User Dialog Act Detection
State Vector Construction
Dialog Policy Making
Response Generation
Generate Response Corresponding to Dialog Policy
Dialog Policy Learning
Learning Dialog Policy with A2C

Figure 2. Architecture of AidIR.

3.1. Corpus Pre-Processing

Corpora pre-processing pre-processes corpora for the ensuing models. This stage encompasses three steps: taxonomy operationalization, token replacement, and WordPiece tokenization [15]. We will elucidate these steps in the following paragraphs.

3.1.1. Operationalization of Taxonomy

First, we operationalize the taxonomy. This stage enables us to decide the domains of AidIR. Initially, we decide to treat each disease as the domain of AidIR. However, we find that some diseases have similar characteristics. As a result, we study the information of each disease on the CDC's website [16] and group the diseases that have similar features into the same category. After our operationalization, the domains in AidIR are diseases associated with coronaviruses and diseases transmitted by vector mosquitoes. According to our study, some individual diseases may have similar features. For example, both SARS and COVID-19 cause cough, fever, and pneumonia. Moreover, diseases, transmitted by vector mosquitoes, such as dengue fever and Zika virus, also have similar features. These features cause the obstacle to classify the domain of the dialog. Thus, we treat the diseases corresponding to coronaviruses as a single category and the diseases transmitted by vector mosquitoes as an individual category. The diseases related to coronaviruses include SARS, MERS, and COVID-19. The diseases transmitted by vector mosquitoes encompass dengue fever, Zika virus, and chikungunya. Furthermore, taxonomy operationalization also reduces the number of domains in AidIR. In other words, we will require six domains if we do not operationalize the taxonomy but we only need two domains after taxonomy operationalization.

3.1.2. Token Replacement

After operationalization, we start to replace some types of numbers with special tokens to normalize sentences. This stage enables us to train a general model instead of a specific model. For example, we normalize the sentence, body temperature ? °C, to the sentence, body temperature [temp]°C. We will demonstrate the effectiveness of this method by using the mask LM task and the downstream task in Sections 4.1 and 4.2. The special tokens that we use are as follows:

- [temp]: a special token for ? degrees Celsius;
- [start_incubation]: a special token for the start of the incubation period;
- [end_incubation]: a special token for the end of the incubation period.

Moreover, we write the regular expression [17] to match the raw text because of the difficulty of training a model to perform this task. The regular expressions that we write are as follows:

- 潛伏期.?[09]+\.?[0-9]?至[0-9]+.?[0-9]天: this matches sentences such as 潛伏期 5 至 10 天 (incubation period is from 5 to 10 days);
- [體溫 | 發燒].*[0-9]+\.?[0-9]: this matches sentences such as 體溫 39 °C (body temperature is 39 degrees Celsius).

3.1.3. WordPiece Tokenization

The final stage of the pre-processing is to tokenize words into small pieces of words so that the ensuing models can more smoothly process the sentences. In AidIR, we use WordPiece tokenization to tokenize sentences into subword-level tokenization. WordPiece tokenization [15] is the combination of word-level tokenization and token-level tokenization. This method is originally proposed to bootstrap the BLEU score [18] in machine translation. The principle of this method is to train a language model to tokenize the sentences into the words that frequently appear. WordPiece tokenization reduces the numbers of unknown words because WordPiece tokenization tokenizes words into more general word forms. The reduction in unknown words makes the model more robust. In AidIR, we use the PyTorch version of the pre-trained WordPiece tokenizer [8] to tokenize words. We also remove some stop words such as punctuation and some stop words in general NLP such as 的 ('s) to trim the length of the sentences. In addition, we add some words that are not included in the pre-trained tokenizer [8] and normalize the English words to lowercase letters. Those words are vital in this research since those words are related to the characteristics of diseases. The words that we add are as follows:

- 痢: one of the Chinese characters for diarrhea;
- 泌: one of the Chinese characters for secretion;
- 沫: one of the Chines characters for droplets;
- [temp]: our special token for body temperature;

- [start_incubation] and [end_incubation]: our special token for the start and end of the incubation period.

3.2. Domain Classification and Sequence Label in AidIR

After the pre-processing stage, we fit the pre-processed corpora into our NLU models. In terms of domain classification and sequence label, we first fine-tune the pre-trained Chinese BERT model [8] with the mask LM task and the medical corpora that we pre-process to make the downstream task more accurate. The results will be discussed in Section 4.1. Then, we transfer this model to simultaneously perform the domain classification task and the sequence label task. The architecture of our BERT model, which we call ABERT, is similar to the BERT model proposed by Delvin et al. [9]. However, we jointly train ABERT to perform domain classification and sequence label tasks with the loss function that we design.

3.2.1. Pre-Training with Mask LM Task and Medical Corpora

In the pre-training stage, we use the mask LM task, proposed by Delvin et al. [9], and medical corpora to fine-tune the pre-trained Chinese BERT model [8]. This task randomly replaces 15% of the words with the special token, [MASK]. The objective of this model was to predict the correct words that are replaced by this special token. This method empowers the model to learn the general linguistic rules and the relationship between each word. Although the model that we use was trained with the large corpora, we further fine-tune this model with the corpora that we collect and the mask LM task. This stage enables our model to learn the relationship of the medical corpora. We will show the effectiveness of our model in Sections 4.1 and 4.2. Unlike traditional language models such as RNNLM, this method enables us to train a bidirectional language model. Since bidirectionality enables the model to understand the language not only from left to right but from right to left, bidirectionality reduces the ambiguity of the polysemous words. We use the cross-entropy loss as our loss function. The formulae of the cross-entropy loss are defined in Equation (1):

$$\mathcal{L}(y, \hat{y}) = \sum_j y_j \ln(\hat{y}_j) \tag{1}$$

$\hat{y}_j$ is defined as the predicted probability of the word j; y_j is defined as the ground-truth probability of the word j; $\hat{y}_j$ is calculated using the softmax function which is defined in Equation (2):

$$\hat{y}_j = \frac{\exp(w_j)}{\sum_{i=0}^{V} \exp(w_i)} \tag{2}$$

w_j is defined as the score of the word j. This stage enables us to improve the accuracy of the downstream tasks. We will discuss the results in Section 4.2.

3.2.2. BERT for Sequence Label and Domain Classification

In terms of the downstream task, we use a loss function with the weighting method that we propose to jointly train a BERT model for sequence label and domain classification tasks. The parameters that we use are α and β. α enables us to bootstrap the accuracy of the domain classification task and β enables us to enhance the accuracy of the sequence label task. Both α and β are the hyperparameters that require fine-tuning. That is, the optimal value of α and β can be determined through the experiment. We will discuss the experiment in Section 4.2. In this model, we classify the domain by using the vector of the special token, [CLS]. This token is a special token for the sentence-level classification task in the original model [9]. In addition, we use the token related to each input to label the sequence. The labels that we use are as shown in Table 2. The encoding method that we utilize is BIO encoding. Then, we use the loss function that we design to fine-tune the model that we pre-train in Section 3.2.1. Formally, we define the loss function for the ABERT model in Equation (3):

$$\mathcal{L}_{ABERT}(\mathcal{L}_{DC}, \mathcal{L}_{SL}) = \alpha\mathcal{L}_{DC} + \beta\mathcal{L}_{SL}, \text{where} \quad \alpha + \beta = 2 \tag{3}$$

We define $\mathcal{L}_{SL}$ as the loss for the sequence label. $\mathcal{L}_{DC}$ is defined as the loss for domain classification. α and β are hyperparameters to fine-tune. We will discuss the experimental results in Section 4.2. Mathematically, we define $\mathcal{L}_{SL}$ and $\mathcal{L}_{DC}$ in Equations (4) and (5), respectively:

$$\mathcal{L}_{SL}(y, \hat{y}) = \sum_i \sum_i y_{i,j} \ln(\hat{y}_{i,j}) \tag{4}$$

$$\mathcal{L}_{DC}(y, \hat{y}) = \sum_j y_j \ln(\hat{y}_j) \tag{5}$$

In $\mathcal{L}_{SL}(y, \hat{y})$, $y_{i,j}$ is defined as the ground-truth probability of the label j of the word i. We define $\hat{y}_{i,j}$ as the predicted probability of the label j of word i. In terms of $\mathcal{L}_{DC}(y, \hat{y})$, y_j is defined as the ground-truth probability of class j. $\hat{y}_j$ is defined as the predicted probability of class j. In addition, we add a special token, [PAD], to equalize the length of the sentences. In the sequence label task, we ignore the loss for this special token to prevent the model from predicting the sequence label of this special token.

Table 2. Sequence labels and their descriptions.

Type of Label	Description
TOD	Label type of disease
Neural_System	Label neural system
Symptom	Label symptom
Organ	Label organ
Cell	Label cell
Pathogen	Label pathogen
Transmission	Label transmission
Incubation_Period	Label incubation period
Precaution	Label precaution
O	None of the above

3.3. Dialog Management in AidIR

After the NLU models, we start to track the state of the dialog and make the decision based on the current dialog state. To obtain the user dialog act, AidIR first classifies the user dialog act. We will discuss the user dialog act classification in Section 3.3.1. Then, AidIR starts to construct the state vector with the information of NLU models and the dialog act classification model. The state vector in AidIR includes the probability distribution of the intent, the probability distribution of the domain classification, and the one-hot vector of the current dialog act of the user. The probability distributions of the intent and the domain classification are the direct output of the model. The one-hot vector of the current dialog act of the user is generated by selecting the maximum probability of the probability distribution of the dialog act classification. Finally, AidIR determines the appropriate dialog policy based on the state vector. We will discuss how AidIR decides the dialog policy in Section 3.3.2.

3.3.1. User Dialog Act Detection

This section explains our method for detecting the dialog act of the user. First, we find whether the disease features exist in the sentences with the ABERT model. Specifically, we label each token with the ABERT model and extract the tokens whose label is not O. Then, we construct those tokens into the words and add them to the self-defined dictionary for Jieba [19] to tokenize the original sentence into meaningful words. In this task, we use the word-level tokenization method to tokenize original sentences. Second, we replace the words with special tokens. For example, we replace the SARS with TOD, a special token that represents the type of disease. This method allows us to train a dialog act model that can be used across all types of domains. More specifically, we do not need to train the dialog act model for each type of domain in which we replace some words with special tokens. Because we can use the sequence label model to label each word, we do not need to write regular expressions to match those words. Hence, we utilize the sequence label model to label each word at this time. Finally, we use the skip-gram model [20,21], which predicts the context based on the center word, to transform each word into a dense vector and fit those vectors into the LSTM model [22]. According to Lee and Dernoncourt [23], we only need to utilize a top few words to classify the dialog act. As a result, we use the short-text classification method [23] to classify the dialog act of the user. We will discuss the training results of this work in Section 4.4. The dialog acts of users are presented in Table 3.

Table 3. Dialog act in AidIR.

Act of Dialog	AidIR	Users
Greeting	✓	✓
Farewell	✓	✓
Request	✓	✓
Inform	✓	✓
Reject	×	✓
Confirm	×	✓
Check	✓	×

3.3.2. Dialog Policy Making

To choose the right policy based on the dialog state, AidIR uses a simple neural network and some rules to select the dialog policy. The dialog policies of AidIR are demonstrated in Table 3. The dialog policy network in AidIR outputs two categories of information. The first one is the dialog policy. The other is the token for the dialog policy. The dialog policy network decides four types of dialog policies: request, inform, check, and farewell. Furthermore, we use the rule-based methods to determine two kinds of dialog policy. The first one is greeting. If the dialog act of the user is a greeting, AidIR will directly output a greeting message to the user. The other is the termination of the conversation. If the number of interactions exceeds 10, AidIR will directly terminate the conversation. In terms of token prediction, the tokens encompass intent, feature, and both. These tokens only matter if the dialog policy is either request or check. The intent token is for AidIR to check or request the intent of the dialog. The feature token is for AidIR to check or request the disease features. The both token is for AidIR to check or request both disease features and the intent of the dialog.

3.4. Response Templates Corresponding to Each Dialog Policy

This section will introduce the response templates associated with each dialog policy. Each dialog policy and its corresponding response template are described in Table 4. To make the response more precise, we also write the templates related to each token for some dialog policies. When the dialog policy is inform, we will directly return the corresponding information. We obtain the disease information from the websites of the CDC, Society of Laboratory Medicine, and National Yang Ming Chiao Tung University Hospital in Taiwan, R.O.C.

Table 4. Dialog policy and its corresponding response.

Types of Response	Response Template
Greeting	你好,請問有甚麼問題? (Hello, what questions do you want to ask?)
Farewell	掰掰啦! (Goodbye!)
Request (Intent)	請提供所要搜尋的資訊種類 (Please provide the type of information that you'd like to retrieve.)
Request (Feature)	請提供更多疾病資訊 (Please provide more disease information.)
Request (Both)	請提供所要搜尋的資訊種類及疾病資訊 (Please provide the disease information and type of information that you'd like to retrieve.)
Check (Intent)	你所要查詢的資料種類為__嗎? (Is the type of information that you'd like to retrieve __?)
Check (Feature)	你所提供的疾病資訊為<slot_type>:<slot_value>嗎? (Is the disease information that you provide <slot_type>:<slot_value>?)
Check (Both)	你所要查詢的資料種類為__嗎? (Is the type of information that you'd like to retrieve __?) 你所提供的疾病資訊為<slot_type>:<slot_value>嗎 ? (Is the disease information that you provide <slot_type>:<slot_value>?)
Termination	不好意思, 無法回答您的問題 (Sorry! We cannot answer your questions.)

3.5. Dialog Policy Learning

We apply Advantage Actor Critic (A2C) [24–28] to enable the model to learn the dialog policy online when the interactions are finished. The dialog policy learning stage will enable AidIR to continue learning the dialog policy from the interactive experiences. The dialog policy learning can be divided into two stages. First, we warm up our policy network with supervised learning (SL) because the reinforcement learning (RL) algorithm always requires many interactions to yield reasonable results. According to Su et al. [29,30], SL prevents the model from yielding bad results in the beginning. Second, we use the A2C algorithm to improve the dialog policy network online based on the reward that we define. We will introduce the reward corresponding to each state in Section 3.5.3. Moreover, we parallelize the learning process using Python's built-in package, threading [31]. This way not only enables the system to update the policy online but prevents this process from affecting the user's interaction with the system.

3.5.1. Warm-Up Stage

We trained the model with the SL task to prepare the policy network for the A2C algorithm during the warm-up stage. Because dialog policy making is a classification task, we employed cross-entropy loss to calculate the loss during the warm-up stage. The training objective of the warm-up task is to minimize the cross-entropy loss between the ground-truth label and the predicted label. Formally, we defined two losses for our dialog policy network in Equations (6) and (7):

$$\mathcal{L}_{dp}(y,\hat{y}) = \sum_{i=0}^{N_{dp}} y_i \ln(\hat{y}_i) \tag{6}$$

$$\mathcal{L}_{token}(y,\hat{y}) = \sum_{i=0}^{N_{dp=request \cup check}} \sum_{j=0}^{N_{token}} y_{i,j} \ln(\hat{y}_{i,j}) \tag{7}$$

$\mathcal{L}_{dp}$ is defined as the loss for the dialog policy prediction. $\mathcal{L}_{token}$ is defined as the loss for the token prediction. We will only calculate the loss for token prediction if the dialog policy is either request or check. y_i is the ground-truth probability of the dialog policy, which is represented using a one-hot vector. $y_{i,j}$ is the ground-truth probability of the token. $\hat{y}$ represents the predicted probability. This probability is generated using the softmax function. N_{dp} is the number of dialog policies. N_{token} is the number of tokens. $N_{dp=request \cup check}$ is the number of dialog policies that are either request or check. We define the total loss at the warm-up stage in Equation (8):

$$\mathcal{L}_{total}\Big(\mathcal{L}_{dp}(y,\hat{y}),\mathcal{L}_{token}(y,\hat{y})\Big) = \mathcal{L}_{dp}(y,\hat{y}) + \mathcal{L}_{token}(y,\hat{y}) \tag{8}$$

3.5.2. Online Learning Stage

In the online learning stage, we use the A2C algorithm to update the weights that we train in the warm-up stage. The A2C algorithm can be divided into two parts. The first part is the actor network, which predicts the action that the agent will perform. The second part is the critic network, which will predict the Q value related to the action that the agent performs. A2C is a TD learning method, which means that A2C will bootstrap the reward forward. In AidIR, we use the TD(0) learning method, which will only bootstrap the reward one step forward. The actor network in AidIR is similar to the network in the warm-up stage. However, we did not use the cross-entropy loss to update the actor network. Instead, we employed the policy gradients to update the model. Then, we used the critic network to generate the Q value of the state s and the action a at turn t and the Q value of action a and state s at turn $t+1$. The training objective of the critic network is to minimize to mean square error of the advantage function. We mathematically defined policy gradients for dialog policy prediction and token prediction in the actor network in Equations (9) and (10):

$$\bigtriangledown_\theta J_{dp}(\theta) = \mathbb{E}_{a\sim\pi_\theta}\Big[\bigtriangledown_\theta \ln\Big(\pi_\theta^{dp}(s_t,a_t)\Big)\hat{A}_{dp}(s_t,a_t)\Big] \tag{9}$$

$$\bigtriangledown_\theta J_{token}(\theta) = \mathbb{E}_{a\sim\pi_\theta}\Big[\bigtriangledown_\theta \ln\Big(\pi_\theta^{token}(s_t,a_t)\Big)\hat{A}_{token}(s_t,a_t)\Big] \tag{10}$$

where $\bigtriangledown_\theta J_{dp}(\theta)$ is the policy gradient for dialog policy prediction. $\bigtriangledown_\theta J_{token}(\theta)$ is the policy gradient for token prediction. We will only calculate the policy gradient for token prediction if the dialog policy is either request or check. $\pi_\theta(s_t,a_t)$ is the probability of performing the action a under state s at turn t. This is calculated using the softmax function. $\hat{A}_{dp}(s_t,a_t)$ and $\hat{A}_{token}(s_t,a_t)$ are the advantage functions for dialog policy prediction and token prediction at turn t which are defined in Equations (11) and (12), respectively. The total loss of the actor network is defined in Equation (13):

$$\hat{A}_{dp}(s_t,a_t) = r(s_t,a_t) + \gamma Q_{dp}(s_{t+1},a_{t+1}) - Q_{dp}(s_t,a_t) \tag{11}$$

$$\hat{A}_{token}(s_t,a_t) = r(s_t,a_t) + \gamma Q_{token}(s_{t+1},a_{t+1}) - Q_{token}(s_t,a_t) \tag{12}$$

$$\mathcal{L}_{actor}\Big(\bigtriangledown_\theta J_{dp}(\theta),\bigtriangledown_\theta J_{token}(\theta)\Big) = \bigtriangledown_\theta J_{dp}(\theta) + \bigtriangledown_\theta J_{token}(\theta) \tag{13}$$

where $\mathcal{L}_{actor}\Big(\bigtriangledown_\theta J_{dp}(\theta),\bigtriangledown_\theta J_{token}(\theta)\Big)$ is the total loss of the actor network. $r(s_t,a_t)$ is the immediate reward while performing a under state s at turn t. γ is a discount factor. $Q_{dp}(s_t,a_t)$ is the Q value of the dialog policy prediction generated by the critic network for state s and action a at turn t. $Q_{token}(s_t,a_t)$ is the Q value of token prediction generated by the critic network for state s and action a at turn t. This is the TD(0) learning method which bootstraps the reward one step forward. In addition, we define the mean square errors of dialog policy prediction and token prediction in the critic network in Equations (14) and (15).

$$MSE_{dp}(s_t,a_t) = \frac{1}{N}\sum_{t=1}^{N}\Big[\hat{A}_{dp}(s_t,a_t)\Big]^2 \tag{14}$$

$$MSE_{token}(s_t,a_t) = \frac{1}{N}\sum_{t=1}^{N_{dp=request\cup check}}\Big[\hat{A}_{token}(s_t,a_t)\Big]^2 \tag{15}$$

where N is the number of turns in one episode. $N_{dp=request\cup check}$ is the number of turns in one episode whose dialog policy is either request or check. $Q_{dp}(s_t,a_t)$ is the state-action value of dialog policy prediction at turn t. $Q_{token}(s_t,a_t)$ is the state-action value of token prediction at the turn t. We define the total loss of the critic network in Equation (16):

$$MSE_{total}\Big(MSE_{dp}(s_t,a_t), MSE_{token}(s_t,a_t)\Big) = MSE_{dp}(s_t,a_t) + MSE_{token}(s_t,a_t) \quad (16)$$

3.5.3. Rewards Corresponding to Each State

We summarize the rewards associated with each state in Table 5. The model will earn the most reward and the least reward when the dialog act of the user's response is farewell and rejection, respectively. When the user provides the information, we penalize the model because this action may have resulted from the system's request. We do not give any penalties and rewards when users request information since the request may result from the user's eagerness to know other types of information.

Table 5. Rewards corresponding to each state.

Dialog Act of User	Reward Values
Bye	10
Request	0
Inform	−5
Rejection	−10

3.6. Implementation of AidIR

In this research, we mainly used PyTorch [32] to implement the dialog act model and the A2C algorithm. Furthermore, we revised the PyTorch version of Hugging Face's API [8] for the implementation of the ABERT model and used the PyTorch version of Hugging Face's WordPiece tokenization API to tokenize words into subword-level tokens. During the pre-training stage, we used the hugging face's mask LM API to fine-tune the original Chinese model with the mask LM task and medical corpora. We implemented the naive Bayes algorithm with scikit-learn [33]. Moreover, we used Line's official API [34] to enable users to retrieve information with Line chat media. We used Python's built-in library, re [35] , to perform the regular expressions that we designed and enable AidIR to match some special numbers.

4. Experimental Results

We discuss the experimental results of our models in this section. We discuss the results of the BERT model fine-tuned with the mask LM task and medical corpora in Section 4.1. We mention the training results of the ABERT model in Section 4.2. We use BertViz [36] to visualize the ABERT model in Section 4.3. We discuss the results of dialog act models in Section 4.4. We discuss the results of the subjective evaluation in Section 4.5. Finally, we will compare AidIR with the popular system with traditional GUI [6] in Section 4.6.

4.1. Fine-Tuning Results of BERT Model with Mask LM Task

In this section, we discuss the results of the BERT model fine-tuned with the mask LM task and medical corpora. The model that we fine-tune is the pre-trained Chinese BERT model [8]. The optimization algorithm is Adam [37]. The training epochs are 10. The learning rate is 0.0001. The number of corpora is 340 in total. We used 70% of these corpora to fine-tune the model and used the keywords related to diseases such as 新冠肺炎 (COVID-19), SARS, and 登革熱 (dengue fever) to retrieve the related articles in Google's search engine. Most of our corpora came from news articles because of the difficulty of finding Chinese research articles, although we also retrieved some articles from more reliable resources such as [16]. In addition, corpora are pre-processed by using the methods that we discussed in Section 3.1. The processed corpora are published to our GitHub repository. The link is mentioned in the Data Availability Statement. Table 6 demonstrates the fine-tuned results. Compared with the model that does not require further fine-tuning with the mask LM task and the medical corpora, the output probability distribution of the fine-tuned model is more peak. Moreover, the model without fine-tuning sometimes cannot predict the right result, as shown in Table 7.

To demonstrate the fact that the model with special tokens yields better results, we fine-tune the model with the same optimization algorithm, task, epoch, and learning rate. However, we do not replace some types of numbers with special tokens at this time. The training results are demonstrated in Table 7.

The experiments indicate that the model with special tokens yields better results. This model not only accurately predicts the token but outputs peak probability distribution. On the other hand, the model without special tokens sometimes incorrectly predicts the token. Furthermore, its output probability is not peak. Moreover, we investigated whether this approach influences other sentences that do not have to replace some types of numbers with special tokens. According to our experiments, this approach does not affect other sentences that do not have to replace numbers with special tokens. We will further investigate whether this approach benefits the downstream task in Section 4.2.

Table 6. Results of the BERT model fine-tuned with mask LM task.

Original Sentence	Sentence with [MASK]	Token/Probability (%)	Sentence in English
新冠肺炎潛伏期通常為[end_incubation]天	新冠肺炎潛伏期通常為[MASK]天	[end_incubation]/99	The incubation period of COVID-19 is usually [end_incubation] days.
茲卡病毒感染症潛伏期通常為 [start_incubation] 至 [end_incubation] 天最長可達[end_incubation]天	茲卡病毒感染症潛伏期通常為 [start_incubation] 至 [end_incubation] 天最長可達[MASK]天	[end_incubation]/99	The incubation period of Zika virus is usually from [start_incubation] to [end_incubation]. The longest incubation period is [end_incubation] days.
SARS 和新冠肺炎都可能造成高燒>[temp]°C	SARS 和新冠肺炎都可能造成高燒>[MASK]°C	[temp]/98	SARS and COVID-19 may cause high fever which means the temperature is greater than [temp]°C.
[CLS]登革熱會造成高燒[temp]°C	登革熱會造成高燒[MASK]°C	[temp]/56	Dengue fever may cause high fever [temp]°C.
[CLS]登革熱可能會發燒至[temp]°C	登革熱可能會發燒至[MASK]°C	[temp]/87	Dengue fever may cause fever up to 39 °C.
[CLS]茲卡病毒感染症潛伏期通最長可達[end_incubation]天	茲卡病毒感染症潛伏期通最長可達[MASK]天	[end_incubation]/99	The longest incubation period of Zika virus is [end_incubation] days.
登革熱可以經由病媒蚊傳播傳播	登革熱可以經由[MASK]媒蚊傳播傳播	病/95	Dengue fever is transmitted by vector mosquitoes.
SARS 及新冠肺炎主要是由飛沫傳播但也有可能從氣溶膠傳播	SARS 及新冠肺炎主要是由飛沫傳播但也有可能從氣[MASK]膠傳播	溶/99	SARS and COVID-19 are mainly transmitted by droplets but they are possibly transmitted by aerosol.
屈公病會讓患者感覺關節疼痛	屈[MASK]病會讓患者感覺關節疼痛	公/7	Chikungunya causes arthralgia.
登革熱又稱斷骨熱是由黃病毒所造成	登革熱又稱斷骨[MASK]是由黃病毒所造成	熱/55	Dengue fever is also known as 斷骨熱(another name for dengue fever in Chinese) is caused by the flavivirus.
冠狀病毒是造成新冠肺炎及SARS的主要病毒	冠[MASK]病毒是造成新冠肺炎及SARS的主要病毒	狀/99	Coronavirus is the virus that causes COVID-19 and SARS.
SARS可能造成乾咳四肢無力也可能會從胸部x光看出肺炎的現象	SARS可能造成乾咳四肢無力也可能會從胸部x光看出肺[MASK]的現象	[-0.8em]炎/99	SARS may cause dry cough and limb weakness. We may also find pneumonia by chest X-ray.
茲卡病毒只要在病媒蚊如埃及斑蚊體內增殖幾天後就具有傳播力	茲卡病毒只要在病媒蚊如埃及[MASK]蚊體內增殖幾天後就具有傳播力	斑/99	Zika virus will be transmissible after incubation in the body of vector mosquitoes such as yellow fever mosquito for a few days.
新冠肺炎以及SARS病毒藉由飛沫及氣溶膠傳播	[MASK]冠肺炎以及SARS病毒藉由飛沫及氣溶膠傳播	新/34	The virus of COVID-19 and SARS is transmitted by droplets and aerosol.
SARS與新冠肺炎病毒為具有套膜正性單股的rna病毒	SARS與新冠肺炎病毒為具有套膜正性單股的[MASK]病毒	rna/82	The virus of SARS and COVID-19 is an enveloped and positive-strand RNA virus.

Table 7. Results of the BERT Model without special tokens.

Original Sentence	Sentence with [MASK]	Token/Probability (%)	Sentence in English
新冠肺炎潛伏期通常為14天	新冠肺炎潛伏期通常為[MASK]天	10/36	The incubation period of COVID-19 is usually 14 days.
茲卡病毒感染症潛伏期通常為3至7天最長可達12天	茲卡病毒感染症潛伏期通常為3至7天最長可達[MASK]天	10/75	The incubation period of Zika virus is usually from 3 to 7. The longest incubation period is 12 days.
SARS 和新冠肺炎都可能造成高燒 > 38 °C	SARS 和新冠肺炎都可能造成高燒 > [MASK]°C	38/42	SARS and COVID-19 may cause high fever which means the temperature is greater than 38 °C.
登革熱會造成高燒40 °C	登革熱會造成高燒[MASK]°C	40/11	Dengue fever may cause high fever 40 °C.
登革熱可能會發燒至40 °C	登革熱可能會發燒至[MASK]°C	40/11	Dengue fever may cause fever up to 40 °C.
茲卡病毒感染症潛伏期通最長可達12天	茲卡病毒感染症潛伏期通最長可達[MASK]天	30/7	The longest incubation period of Zika virus is 30 days.
登革熱可以經由病媒蚊傳播傳播	登革熱可以經由[MASK]媒蚊傳播傳播	病/95	Dengue fever is transmitted by vector mosquitoes.
SARS 及新冠肺炎主要是由飛沫傳播但也有可能從氣溶膠傳播	SARS 及新冠肺炎主要是由飛沫傳播但也有可能從氣[MASK]膠傳播	溶/99	SARS and COVID-19 are mainly transmitted by droplets but they are possibly transmitted by aerosol.
屆公病會讓患者感覺關節疼痛	屆[MASK]病會讓患者感覺關節疼痛	×	Chikungunya causes arthralgia.
登革熱又稱斷骨熱是由黃病毒所造成	登革熱又稱斷骨[MASK]是由黃病毒所造成	熱/0	Dengue fever is also known as 斷骨熱 (another name for dengue fever in Chinese) is caused by the flavivirus.
冠狀病毒是造成新冠肺炎及SARS的主要病毒	冠[MASK]病毒是造成新冠肺炎及SARS的主要病毒	狀/42	Coronavirus is the virus that causes COVID-19 and SARS.
SARS可能造成乾咳四肢無力也可能會從胸部x光看出肺炎的現象	SARS可能造成乾咳四肢無力也可能會從胸部x光看出肺[MASK]的現象	炎/52	SARS may cause dry cough and limb weakness. We may also find pneumonia by chest X-ray.
茲卡病毒只要在病媒蚊如埃及斑蚊體內增殖幾天後就具有傳播力	茲卡病毒只要在病媒蚊如埃及[MASK]蚊體內增殖幾天後就具有傳播力	斑/14	Zika virus will be transmissible after incubation in the body of vector mosquitoes such as yellow fever mosquito for a few days.
新冠肺炎以及SARS病毒藉由飛沫及氣溶膠傳播	[MASK]冠肺炎以及SARS病毒藉由飛沫及氣溶膠傳播	×	The virus of COVID-19 and SARS is transmitted by droplets and aerosol.
SARS與新冠肺炎病毒為具有套膜正性單股的rna病毒	SARS與新冠肺炎病毒為具有套膜正性單股的[MASK]病毒	rna/8	The virus of SARS and COVID-19 is an enveloped and positive-strand RNA virus.

4.2. Training Results of ABERT

In this section, we examine the performance of the ABERT model. We transfer the model that we fine-tuned with the mask LM task and medical corpora to train the ABERT model. Furthermore, we chose Adam [37] as our optimization algorithm. The learning rate is 0.0001. We trained 10 epochs. We used the same corpora in the pre-training stage to train the ABERT model and used the remaining corpora to test the model. The corpora that we used in this stage are discussed in Section 4.1. During the training stage, we used the same parameters and the same data to train the model with different α and β. The experimental results are presented in Table 8.

Table 8. Training results of the ABERT model.

Parameters		**Accuracy (%)**		**Pre-Trained**	**ST** [c]
α	β	**DC** [a]	**SL** [b]		
1	1	95	87	✓	✓
0.5	1.5	94	89	✓	✓
1.5	0.5	97	87	✓	✓
0.05	1.95	63	92	✓	✓
0.2	1.8	92	92	✓	✓
0.2	1.8	68	89	×	✓
0.2	1.8	80	87	✓	×

[a] DC: domain classification; [b] SL: sequence label; [c] ST: special token.

We first trained the model with $\alpha = 1$ and $\beta = 1$. To pinpoint whether the increment in β can boost the accuracy of the sequence label task, we increase the value of β and reduced the value of α. The experiment demonstrates that the enlargement of β can increase the accuracy of the sequence label task (refer to Row 2 and Row 5 in Table 8). In addition, we investigate whether the enlargement of α increases the accuracy of the domain classification task. We find that the increment in α also increases the accuracy of the domain classification task (refer to Row 3 in Table 8). Moreover, we find that this also diminishes the accuracy of the sequence label task (refer to Row 3 in Table 8). In addition, to investigate whether larger β can further improve the sequence label task, we use extreme values for α and β. According to our experiment, we need to find the trade-off. We cannot overvalue one task and devalue another task. Overvaluing one task cannot yield better results. Furthermore, it worsens another task dramatically (refer to Row 4 in Table 8). As a result, we conclude that the α and β can be treated as weights to value the task whose performance we want to boost.

To examine whether fine-tuning the model with the mask LM task and medical corpora is required in our task, we used the same corpus, the same parameters, and $\alpha = 0.2$ and $\beta = 1.8$ to train the ABERT model without fine-tuning with the mask LM task and medical corpora. According to our experiment, fine-tuning the model is necessary since this yields a better result than the model without fine-tuning (refer to Row 5 and Row 6 in Table 8). Interestingly, we also found that the accuracy of the model with higher weights diminishes by only 3%. However, the accuracy of the model with lower weights dramatically decreases. This further proves the effectiveness of our weighting method.

To verify the effectiveness of the replacement of the numbers, we used the same corpus, the same parameters, and $\alpha = 0.2$ and $\beta = 1.8$ to train the model. However, we did not replace some types of numbers with special tokens at this time. The training result demonstrates that the model with special tokens produces a better result (refer to Row 5 and Row 7 in Table 8). We believe that replacing some categories of numbers with special tokens is a way to normalize the sentences. As a result, the model can more easily learn general linguistic patterns.

In addition, the examination of the requirement of the fine-tuning and replacement of the numbers also suggests the effectiveness of our weighting method. We discovered that the accuracy of the task whose weight is greater only slightly decays. However, the accuracy of the task whose weight is lighter dramatically decays. Therefore, this result further proves that our weighting method is effective.

4.3. *Visualization of ABERT Model*

In addition to accuracy, we used BertViz [36] to visualize the training results of the ABERT model with $\alpha = 0.2$ and $\beta = 1.8$ in the final layer. In the attention visualization graph, we find that the ABERT model can develop relationships between words that are in the same category. For instance, Figure 3 shows one of the Chinese characters of dengue fever, 登, attends to both chikungunya and dengue fever. This phenomenon is reasonable since the ABERT model is the combination of the domain classification task and the sequence label task. In the domain classification task, those words are in the category of the disease transmitted by vector mosquitoes. In terms of the sequence label task, the label of those words is TOD. In Figure 4, we discovered that the ABERT model constructs the relationship between the disease and the virus. For example, we found the relationship between SARS and coronaviruses in Figure 4a. The relationship between those words is appropriate since the pathogen of SARS is the coronavirus. Moreover, we also demonstrated that the ABERT model can capture the semantics of words. Take Figure 5a, for example; SARS has not only a relationship with SARS but also a relationship with its Chinese name. As a result, we find that the ABERT model constructs the relationship between the words and accurately captures the semantics of words.

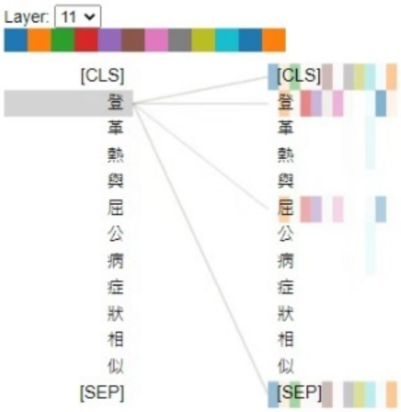
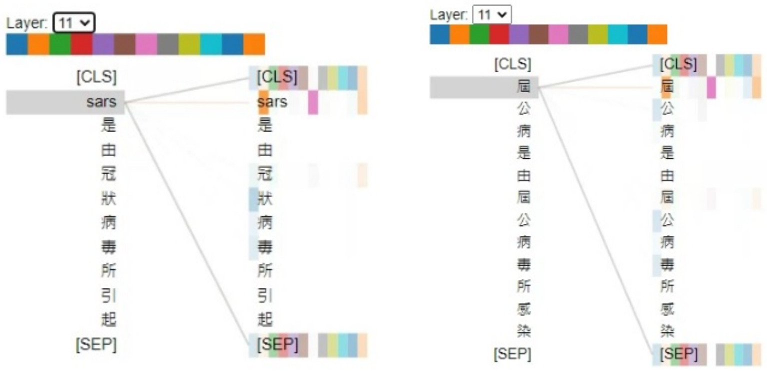
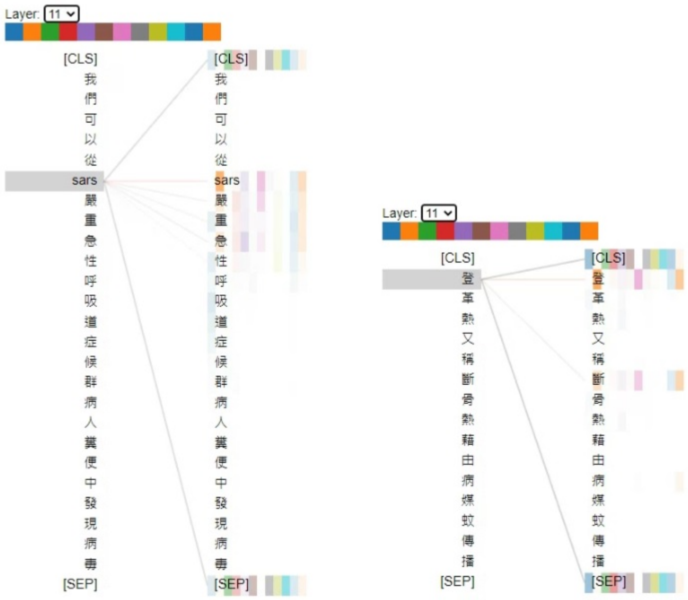


Figure 3. Visualization of attention between words in same category; the sentence is "The symptoms of dengue fever and chikungunya are similar."



(**a**) (**b**)

Figure 4. Visualization of attention between disease name and virus: (**a**) SARS is caused by the coronavirus; and (**b**) chikungunya is caused by chikungunya virus.



(**a**) (**b**)

Figure 5. Visualization of semantics capturing: (**a**) we can find the virus in the feces of SARS 嚴重急性呼吸道症候群 (SARS in the Chinese language) patients; and (**b**) dengue fever is also known as 斷骨熱 (another Chinese name for dengue fever) which is transmitted by vector mosquitoes.

4.4. Results of Dialog Act Models

This section demonstrates the results of our dialog act models. First, we visualize dialog act vectors with TSNE visualization (TSNE vector is calculated by using scikit-learn and the graph is plotted by using Matplotlib). Figure 6 is the TSNE visualization of dialog act vectors. Furthermore, we used the similarity task to discuss the results. Finally, we presented dialog act classification details in Table 9.

According to TSNE visualization, most words with similar semantics are in the nearby vector space. Take the words in the greeting category for example (refer to the blue box in Figure 6 and Table 10); most of the words in Table 10 are all related to the words in the greeting category. This shows that our model can transform the words with similar semantics into similar vectors. Although some words in Table 10 do not correspond to the greeting category, the results are acceptable overall.

Table 9. Dialog act classification details.

Parameters	**Value**
Accuracy (%)	98
Optimizer	Adam [37]
Epoch	300
Learning rate	0.001

Table 10. Training results of Word2Vec for dialog acts classification.

安安 **(Morning)**	哈囉 **(Hello)**	**Hello**
呀 (AW [a])	久違 (LTNS [c])	歐嗨唷 (morning)
初次 (FT [b])	初次 (FT [b])	嘿 (hey)
見面 (meet)	安安 (hi)	空你基哇 (morning)
早 (morning)	再會 (SY [d])	哩賀 (hi)
哩賀 (Hi)	見面 (meet)	hey
哈囉 (Hello)	哩賀 (Hi)	嗨 (hi)
歐嗨唷 (morning)	嘿 (hey)	黑 (hey)
黑 (hey)	空你基哇 (morning)	敖早 (morning)
hey	歐嗨唷 (morning)	hi

Note: abbreviation [a] AW: auxiliary word; [b] FT: first time; [c] LTNS: long time no see; [d] SY: see you.

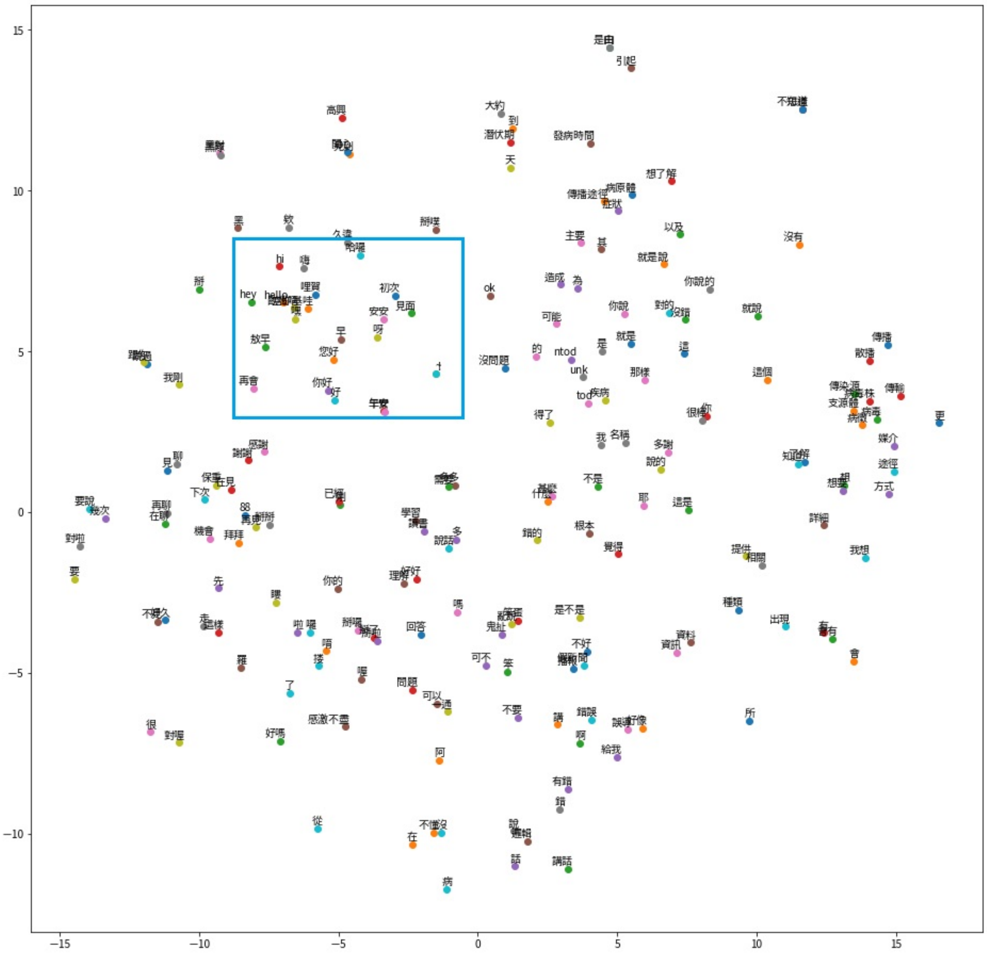


Figure 6. TSNE visualization of dialog acts vector.

4.5. Results of Subjective Evaluation

In this section, we discuss the results of subjective evaluation. In the subjective evaluation, we ask 20 users to use AidIR to retrieve information and fill out the questionnaire. In this questionnaire, we ask those users to rate the intuitiveness, usability, and user experience of AidIR with a score between −2 and 2. The worst is −2 and the best is 2. The average scores and the variance are presented in Table 11. Moreover, we ask those users to evaluate AidIR by answeing yes–no questions. The questions and results are presented in Table 12. Finally, we ask those users to list the advantages and disadvantages of AidIR. The feedback is presented in Tables 13 and 14.

Table 11 shows the average scores and the variance of the intuitiveness, usability, and user experience. Average scores of intuitiveness, usability, and user experiences are 0.8, 0.8, and 1.05, respectively. This indicates those users value the user experience of AidIR the most, although some people think that AidIR still has space to improve. We will conduct a survey to ask the users who think AidIR still has space to improve to provide their opinions in the future.

Table 11. Average scores of user rating.

Task	Average Scores
Intuitiveness	0.8 ± 0.48
Usability	0.8 ± 0.2
User experience	1.05 ± 0.68

In Table 12, we ask those users to answer yes–no questions to evaluate AidIR and compare AidIR with systems using GUI and single-turn dialog systems. In the first two questions, we ask those users to compare AidIR with systems using GUI. The results show that 75% of those users think AidIR is more intuitive than systems using GUI. Furthermore, 70% of those users think that AidIR has a better user experience than systems with GUI. These results indicate the pre-eminence of AidIR. Moreover, we also ask those users to answer whether the deployment of AidIR to Line chat media is better. 75% of those users think that the deployment of AidIR to Line chat media is better. Surprisingly, 90% of our graduate students think AidIR is better than single-turn dialog systems. This shows that the importance of multi-turn dialog systems. Finally, 80% of users will use systems such as AidIR to retrieve information. This demonstrates the value of AidIR and the intuitive way to retrieve information.

Table 12. Results of yes–no questions.

Questions	Yes (%)	No (%)
Is it more intuitive to retrieve information with AidIR than GUI?	75	25
Is the user experience of AidIR better than GUI?	70	30
Is the user experience of integrating AidIR to Line chat media better than without integrating AidIR to Line chat media?	75	25
Would you like to use the systems such as AidIR to retrieve information?	80	20
Is the user experience of AidIR better than the single-turn system?	90	10

In Table 13, we show the advantages of AidIR. Most users think AidIR is more convenient to use. Some users also think that the integration of AidIR in Line chat media is easier to use in daily life. Furthermore, some users think information retrieval with AidIR is very intuitive and AidIR can more quickly provide the right information.

Table 13. Advantages of AidIR.

Original Feedback	Feedback in English
在獲得醫療資訊方面可以更方便、快速。	It is very convenient to obtain medical information with this system.
整合到Line 更貼近生活	Integrating this system to Line is easier to use in daily life.
使用習慣後相當方便。	It is very convenient to use after getting used to it.
在問答方面可以直覺性的回答出所需要的資訊	It is very intuitive to retrieve the information.
可迅速了解相關疾病的正確資訊	This system can help to understand the correct medical information quickly.
相當方便 對於我自己有很好的使用者體驗	It is very convenient. I have a good user experience.
可以更為精準地傳達想要檢索的內容，讓搜尋引擎能找到更貼合所想要的資訊內容	AidIR delivers the information more precisely and this information can be treated as the keywords for search engines to enable search engines to find more precise information.

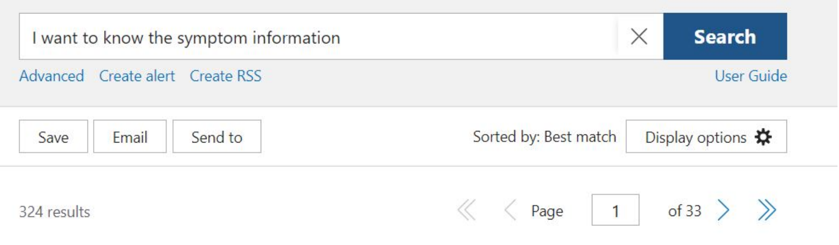
Although the results overall are positive, we also ask those users to provide the disadvantages of AidIR. The feedback is presented in Table 14. Most users think that we can use pictures to aid the explanation of the information. Still, a user thinks that we can add more types of content for users to retrieve. We will consider these opinions and improve these problems in future research.

Table 14. Disadvantages of AidIR.

Original Feedback	Feedback in English
對於對話內容種類內容還有可以增加的空間。	More types of content can be added to the system.
在回答問題時可提供相關圖片，以便更了解資訊	It is better to provide some pictures in the answer to help the understanding of information.
建議可加入圖片輔助說明	It is better to add pictures to aid the explanation of the information.
如果能結合圖片搭配檢索內容會更好	It would be better if we can use pictures to retrieve information.

4.6. Comparison between AidIR and Systems with GUI

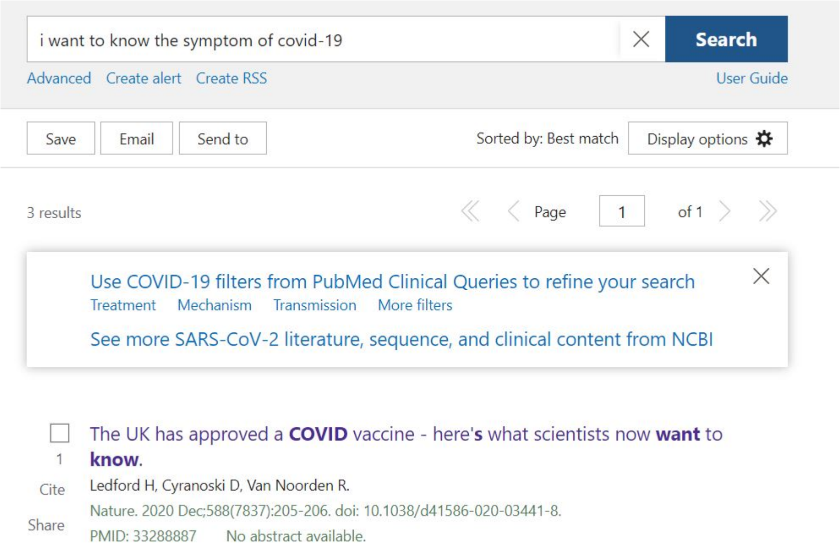
In this section, we will compare AidIR with popular systems using traditional GUI [6]. Because we cannot find the system for medical information retrieval that accepts the Chinese language, we compare AidIR with systems with GUI that only accepts English. In terms of keywords search, systems with conventional GUI cannot guide the user to specify the information. Systems with conventional GUI will only return as many articles as possible if the information that the user provides is too general. The result is as shown in Figure 7. On the other hand, AidIR will guide the user to provide the information in a more specific manner. The result will be discussed in Section 5.6. Furthermore, the user needs to provide the same information as in systems with traditional GUI since systems with traditional GUI cannot track the information that the user provides. On the other hand, AidIR can utilize the information that the user has provided to the system. AidIR does not require the user to provide the same information again. The result will be mentioned in Section 5.4. Nonetheless, this tracking system enables users to rectify the system's error. The results will be explained in Sections 5.3 and 5.5. Moreover, systems with conventional GUI do not use the context to retrieve the articles. In other words, sometimes the articles that systems with conventional GUI retrieve do not correspond to the topics that the user wants to retrieve. The result is as shown in Figure 8. However, AidIR can exploit the context in the sentences, which is as shown in Figure 9. Although the search engine in systems with conventional GUI can return many articles to users, end-users may feel frustrated with these systems. The search result is as shown in Figure 7. When the user uses more specific keywords to retrieve information, this system still returns three articles. The result is as shown in Figure 8. According to our survey, end-users usually want specific results since some end-users do not even have the fundamental knowledge on information that they want to retrieve. The result of our survey is as shown in Table 13. The result of AidIR is as shown in Figure 9. Although the system with GUI [6] allows users to retrieve articles with natural language, users can only use English to retrieve information. Furthermore, this system uses space to tokenize each word only. This system does not exploit the context to tokenize the words. Therefore, the difficulty of using the phrase to retrieve information in this system is beyond doubt. Even though the system with GUI uses logic operators to connect each separate word, the system with GUI also retrieves datum of information related to each word. This process results in general instead of specific results. The process is as shown in Figure 10. In addition, this mechanism is not suitable for the Chinese language since the Chinese language does not have space to separate each word.



I want to know the symptom information

324 results

Figure 7. Results of general keywords in systems with GUI.



i want to know the symptom of covid-19

3 results

Use COVID-19 filters from PubMed Clinical Queries to refine your search

See more SARS-CoV-2 literature, sequence, and clinical content from NCBI

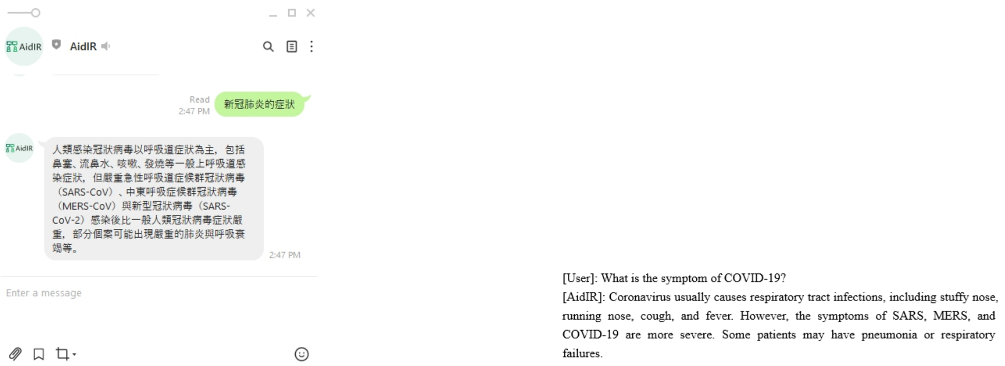
□ 1 The UK has approved a **COVID** vaccine - here'**s** what scientists now **want** to **know**.

Ledford H, Cyranoski D, Van Noorden R.

Nature. 2020 Dec;588(7837):205-206. doi: 10.1038/d41586-020-03441-8.

PMID: 33288887 No abstract available.

Figure 8. Results of keywords search in systems with GUI.



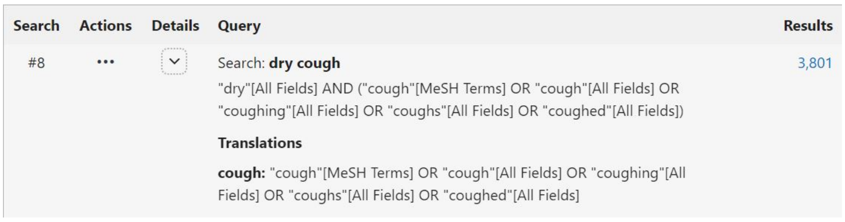
Read 2:47 PM 新冠肺炎的症狀

人類感染冠狀病毒以呼吸道症狀為主，包括鼻塞、流鼻水、咳嗽、發燒等一般上呼吸道感染症狀，但嚴重急性呼吸道症候群冠狀病毒（SARS-CoV）、中東呼吸症候群冠狀病毒（MERS-CoV）與新型冠狀病毒（SARS-CoV-2）感染後比一般人類冠狀病毒症狀嚴重，部分個案可能出現嚴重的肺炎與呼吸衰竭等。 2:47 PM

[User]: What is the symptom of COVID-19?
[AidIR]: Coronavirus usually causes respiratory tract infections, including stuffy nose, running nose, cough, and fever. However, the symptoms of SARS, MERS, and COVID-19 are more severe. Some patients may have pneumonia or respiratory failures.

(**a**) (**b**)

Figure 9. Results of information retrieval in AidIR: (**a**) original dialog; and (**b**) dialog in English.



Search	Actions	Details	Query	Results
#8	•••	⌄	Search: **dry cough** "dry"[All Fields] AND ("cough"[MeSH Terms] OR "cough"[All Fields] OR "coughing"[All Fields] OR "coughs"[All Fields] OR "coughed"[All Fields]) **Translations** **cough:** "cough"[MeSH Terms] OR "cough"[All Fields] OR "coughing"[All Fields] OR "coughs"[All Fields] OR "coughed"[All Fields]	3,801

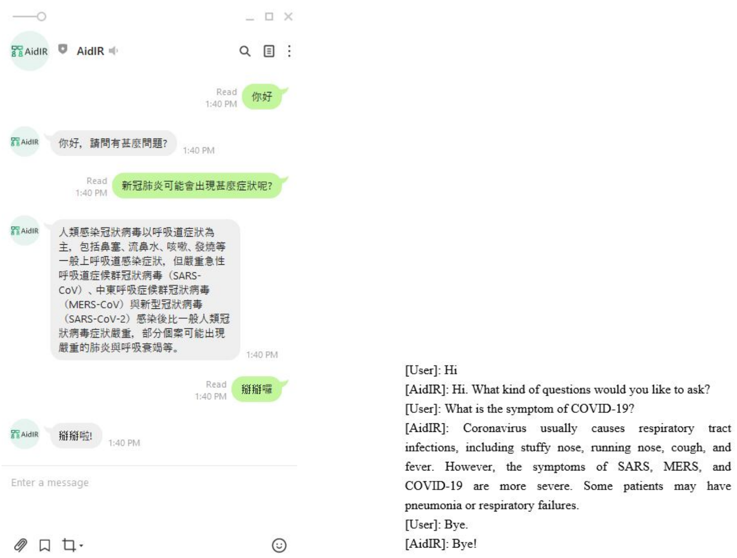
Figure 10. Results of keywords process in systems with GUI.

5. Dialog Samples of AidIR

This section will demonstrate some dialog samples.

5.1. User Provides Enough Information at First

As shown in Figure 11, the user provides enough information in the beginning. AidIR correctly identifies the information that the user provides. Therefore, AidIR provides the information that the user requests.



Read 1:40 PM 你好

你好，請問有甚麼問題? 1:40 PM

Read 1:40 PM 新冠肺炎可能會出現甚麼症狀呢?

人類感染冠狀病毒以呼吸道症狀為主，包括鼻塞、流鼻水、咳嗽、發燒等一般上呼吸道感染症狀，但嚴重急性呼吸道症候群冠狀病毒（SARS-CoV）、中東呼吸症候群冠狀病毒（MERS-CoV）與新型冠狀病毒（SARS-CoV-2）感染後比一般人類冠狀病毒症狀嚴重，部分個案可能出現嚴重的肺炎與呼吸衰竭等。 1:40 PM

Read 1:40 PM 掰掰囉

掰掰啦! 1:40 PM

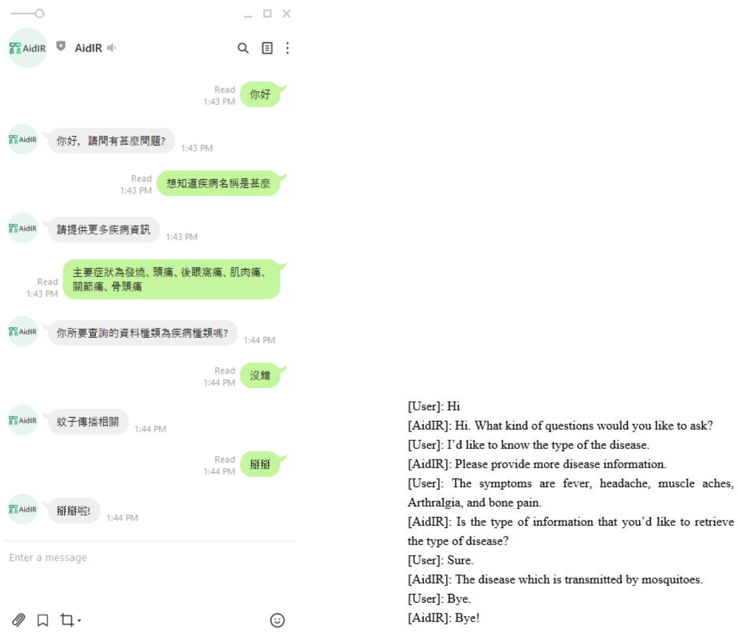
[User]: Hi
[AidIR]: Hi. What kind of questions would you like to ask?
[User]: What is the symptom of COVID-19?
[AidIR]: Coronavirus usually causes respiratory tract infections, including stuffy nose, running nose, cough, and fever. However, the symptoms of SARS, MERS, and COVID-19 are more severe. Some patients may have pneumonia or respiratory failures.
[User]: Bye.
[AidIR]: Bye!

(**a**) (**b**)

Figure 11. User provides enough information at first: (**a**) original dialog; and (**b**) dialog in English.

5.2. AidIR Requests and Checks Information

As shown in Figure 12, the user does not provide enough information at first. AidIR starts requesting more information. The user provides symptoms of disease to AidIR. Subsequently, AidIR is still not quite sure about the type of information that the user is requesting. It checks with the user. After confirmation from the user, AidIR replies the information to the user.



Read 1:43 PM 你好

你好，請問有甚麼問題? 1:43 PM

Read 1:43 PM 想知道疾病名稱是甚麼

請提供更多疾病資訊 1:43 PM

Read 1:43 PM 主要症狀為發燒、頭痛、後眼窩痛、肌肉痛、關節痛、骨頭痛

你所要查詢的資料種類為疾病種類嗎? 1:44 PM

Read 1:44 PM 沒錯

蚊子傳播相關 1:44 PM

Read 1:44 PM 掰掰

掰掰啦! 1:44 PM

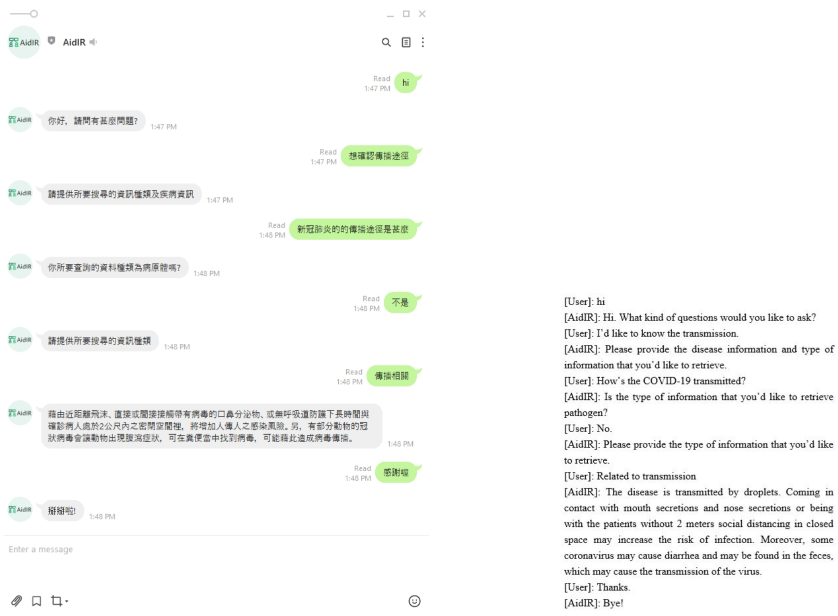
[User]: Hi
[AidIR]: Hi. What kind of questions would you like to ask?
[User]: I'd like to know the type of the disease.
[AidIR]: Please provide more disease information.
[User]: The symptoms are fever, headache, muscle aches, Arthralgia, and bone pain.
[AidIR]: Is the type of information that you'd like to retrieve the type of disease?
[User]: Sure.
[AidIR]: The disease which is transmitted by mosquitoes.
[User]: Bye.
[AidIR]: Bye!

(**a**) (**b**)

Figure 12. AidIR requests and checks information: (**a**) original dialog; and (**b**) dialog in English.

5.3. AidIR Checks with User but Got Rejected

As shown in Figure 13, AidIR checks whether the user wants to know the pathogen of the disease but AidIR is rejected by the user. AidIR asks the user to provide the type of information that the user wants to retrieve again. Then, the user says that the information corresponds to the transmission. After that, AidIR returns the related information to the user.



AidIR

[User]: hi

[AidIR]: 你好，請問有甚麼問題?

[User]: 想確認傳播途徑

[AidIR]: 請提供所要搜尋的資訊種類及疾病資訊

[User]: 新冠肺炎的的傳播途徑是甚麼

[AidIR]: 你所要查詢的資料種類為病原體嗎?

[User]: 不是

[AidIR]: 請提供所要搜尋的資訊種類

[User]: 傳播相關

[AidIR]: 藉由近距離飛沫、直接或間接接觸帶有病毒的口鼻分泌物、或無呼吸道防護下長時間與確診病人處於2公尺內之密閉空間裡，將增加人傳人之感染風險。另，有部分動物的冠狀病毒會讓動物出現腹瀉症狀，可在糞便當中找到病毒，可能藉此造成病毒傳播。

[User]: 感謝喔

[AidIR]: 掰掰啦!

(**a**)

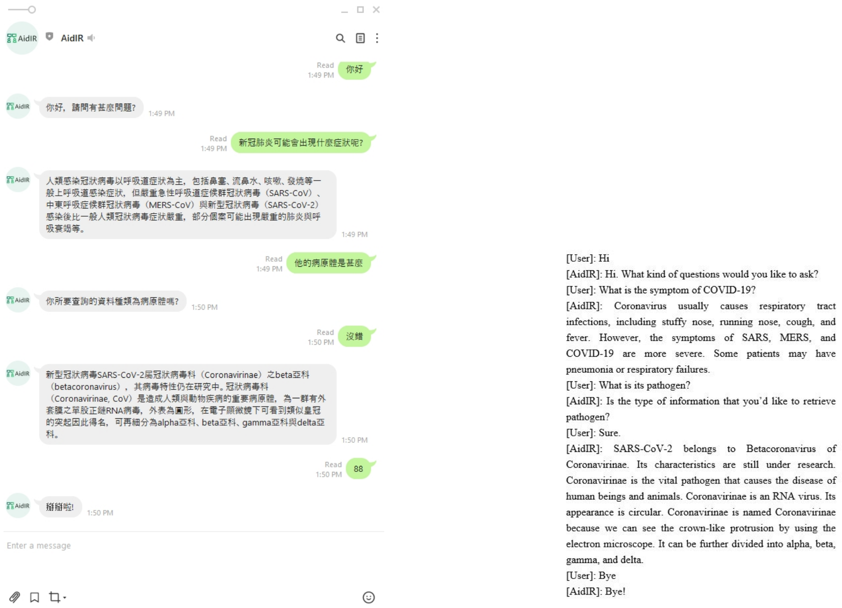
[User]: hi
[AidIR]: Hi. What kind of questions would you like to ask?
[User]: I'd like to know the transmission.
[AidIR]: Please provide the disease information and type of information that you'd like to retrieve.
[User]: How's the COVID-19 transmitted?
[AidIR]: Is the type of information that you'd like to retrieve pathogen?
[User]: No.
[AidIR]: Please provide the type of information that you'd like to retrieve.
[User]: Related to transmission
[AidIR]: The disease is transmitted by droplets. Coming in contact with mouth secretions and nose secretions or being with the patients without 2 meters social distancing in closed space may increase the risk of infection. Moreover, some coronavirus may cause diarrhea and may be found in the feces, which may cause the transmission of the virus.
[User]: Thanks.
[AidIR]: Bye!

(**b**)

Figure 13. AidIR checks with the user but was rejected: (**a**) original dialog; and (**b**) dialog in English.

5.4. User Asks Many Questions Related to One Disease

As demonstrated in Figure 14, the user first asks for information on the symptoms of COVID-19. AidIR returns the related information to the user. Then, the user requests information on its pathogen. AidIR checks with the user. Finally, AidIR returns the related information to the user. AidIR does not require the user to provide the name of the disease again.



AidIR

[User]: 你好

[AidIR]: 你好，請問有甚麼問題?

[User]: 新冠肺炎可能會出現什麼症狀呢?

[AidIR]: 人類感染冠狀病毒以呼吸道症狀為主，包括鼻塞、流鼻水、咳嗽、發燒等一般上呼吸道感染症狀，但嚴重急性呼吸道症候群冠狀病毒（SARS-CoV）、中東呼吸症候群冠狀病毒（MERS-CoV）與新型冠狀病毒（SARS-CoV-2）感染後比一般人類冠狀病毒症狀嚴重，部分個案可能出現嚴重的肺炎與呼吸衰竭等。

[User]: 他的病原體是甚麼

[AidIR]: 你所要查詢的資料種類為病原體嗎?

[User]: 沒錯

[AidIR]: 新型冠狀病毒SARS-CoV-2屬冠狀病毒科（Coronavirinae）之beta亞科（betacoronavirus），其病毒特性仍在研究中。冠狀病毒科（Coronavirinae, CoV）是造成人類與動物疾病的重要病原體，為一群有外套膜之單股正鏈RNA病毒，外表為圓形，在電子顯微鏡下可看到類似皇冠的突起因此得名，可再細分為alpha亞科、beta亞科、gamma亞科與delta亞科。

[User]: 88

[AidIR]: 掰掰啦!

(**a**)

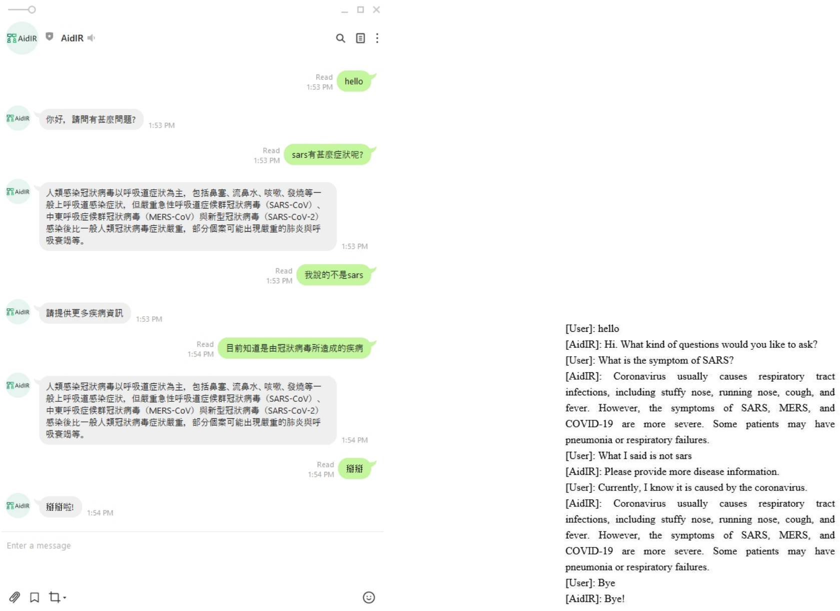
[User]: Hi
[AidIR]: Hi. What kind of questions would you like to ask?
[User]: What is the symptom of COVID-19?
[AidIR]: Coronavirus usually causes respiratory tract infections, including stuffy nose, running nose, cough, and fever. However, the symptoms of SARS, MERS, and COVID-19 are more severe. Some patients may have pneumonia or respiratory failures.
[User]: What is its pathogen?
[AidIR]: Is the type of information that you'd like to retrieve pathogen?
[User]: Sure.
[AidIR]: SARS-CoV-2 belongs to Betacoronavirus of Coronavirinae. Its characteristics are still under research. Coronavirinae is the vital pathogen that causes the disease of human beings and animals. Coronavirinae is an RNA virus. Its appearance is circular. Coronavirinae is named Coronavirinae because we can see the crown-like protrusion by using the electron microscope. It can be further divided into alpha, beta, gamma, and delta.
[User]: Bye
[AidIR]: Bye!

(**b**)

Figure 14. User asks many questions related to one disease: (**a**) original dialog; and (**b**) dialog in English.

5.5. Error Information Included in Rejection Dialog

As demonstrated in Figure 15, error information is included in the rejection dialog. AidIR asks the user to provide the information again. Then, AidIR correctly provides the information. Please note that Figure 15 is only for demonstration. AidIR does not make any mistakes.



hello
Read 1:53 PM

你好，請問有甚麼問題?
1:53 PM

sars有甚麼症狀呢?
Read 1:53 PM

人類感染冠狀病毒以呼吸道症狀為主，包括鼻塞、流鼻水、咳嗽、發燒等一般上呼吸道感染症狀，但嚴重急性呼吸道症候群冠狀病毒（SARS-CoV）、中東呼吸症候群冠狀病毒（MERS-CoV）與新型冠狀病毒（SARS-CoV-2）感染後比一般人類冠狀病毒症狀嚴重，部分個案可能出現嚴重的肺炎與呼吸衰竭等。
1:53 PM

我說的不是sars
Read 1:53 PM

請提供更多疾病資訊
1:53 PM

目前知道是由冠狀病毒所造成的疾病
Read 1:54 PM

人類感染冠狀病毒以呼吸道症狀為主，包括鼻塞、流鼻水、咳嗽、發燒等一般上呼吸道感染症狀，但嚴重急性呼吸道症候群冠狀病毒（SARS-CoV）、中東呼吸症候群冠狀病毒（MERS-CoV）與新型冠狀病毒（SARS-CoV-2）感染後比一般人類冠狀病毒症狀嚴重，部分個案可能出現嚴重的肺炎與呼吸衰竭等。
1:54 PM

掰掰
Read 1:54 PM

掰掰啦!
1:54 PM

(**a**)

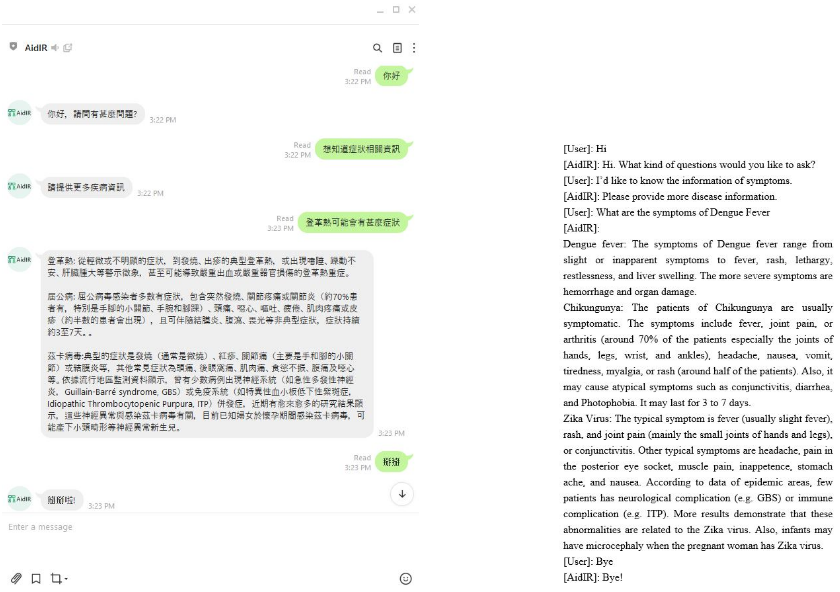
[User]: hello
[AidIR]: Hi. What kind of questions would you like to ask?
[User]: What is the symptom of SARS?
[AidIR]: Coronavirus usually causes respiratory tract infections, including stuffy nose, running nose, cough, and fever. However, the symptoms of SARS, MERS, and COVID-19 are more severe. Some patients may have pneumonia or respiratory failures.
[User]: What I said is not sars
[AidIR]: Please provide more disease information.
[User]: Currently, I know it is caused by the coronavirus.
[AidIR]: Coronavirus usually causes respiratory tract infections, including stuffy nose, running nose, cough, and fever. However, the symptoms of SARS, MERS, and COVID-19 are more severe. Some patients may have pneumonia or respiratory failures.
[User]: Bye
[AidIR]: Bye!

(**b**)

Figure 15. Error information included in rejection dialog: (**a**) original dialog; and (**b**) dialog in English.

5.6. AidIR Requests More Information

As shown in Figure 16, the user does not provide enough information at first. AidIR requests more information. After the user provides enough information, AidIR returns the related information to the user. In AidIR, dengue fever is included in the category of diseases transmitted by vector mosquitoes. Therefore, when the user requests information related to this category, AidIR will simultaneously return all of the information related to this category.



你好
Read 3:22 PM

你好，請問有甚麼問題?
3:22 PM

想知道症狀相關資訊
Read 3:22 PM

請提供更多疾病資訊
3:22 PM

登革熱可能會有甚麼症狀
Read 3:23 PM

登革熱: 從輕微或不明顯的症狀，到發燒、出疹的典型登革熱，或出現嗜睡、躁動不安、肝臟腫大等警示徵象，甚至可能導致嚴重出血或嚴重器官損傷的登革熱重症。

屈公病: 屈公病毒感染者多數有症狀，包含突然發燒、關節疼痛或關節炎（約70%患者有，特別是手腳的小關節、手腕和腳踝）、頭痛、噁心、嘔吐、疲倦、肌肉疼痛或皮疹（約半數的患者會出現），且可伴隨結膜炎、腹瀉、畏光等非典型症狀，症狀持續約3至7天。。

茲卡病毒:典型的症狀是發燒（通常是微燒）、紅疹、關節痛（主要是手和腳的小關節）或結膜炎等，其他常見症狀為頭痛、後眼窩痛、肌肉痛、食慾不振、腹痛及噁心等。依據流行地區監測資料顯示，曾有少數病例出現神經系統（如急性多發性神經炎，Guillain-Barré syndrome, GBS）或免疫系統（如特異性血小板低下性紫斑症，Idiopathic Thrombocytopenic Purpura, ITP）併發症，近期有愈來愈多的研究結果顯示，這些神經異常與感染茲卡病毒有關，目前已知婦女於懷孕期間感染茲卡病毒，可能產下小頭畸形等神經異常新生兒。
3:23 PM

掰掰
Read 3:23 PM

掰掰啦!
3:23 PM

(**a**)

[User]: Hi
[AidIR]: Hi. What kind of questions would you like to ask?
[User]: I'd like to know the information of symptoms.
[AidIR]: Please provide more disease information.
[User]: What are the symptoms of Dengue Fever
[AidIR]:
Dengue fever: The symptoms of Dengue fever range from slight or inapparent symptoms to fever, rash, lethargy, restlessness, and liver swelling. The more severe symptoms are hemorrhage and organ damage.
Chikungunya: The patients of Chikungunya are usually symptomatic. The symptoms include fever, joint pain, or arthritis (around 70% of the patients especially the joints of hands, legs, wrist, and ankles), headache, nausea, vomit, tiredness, myalgia, or rash (around half of the patients). Also, it may cause atypical symptoms such as conjunctivitis, diarrhea, and Photophobia. It may last for 3 to 7 days.
Zika Virus: The typical symptom is fever (usually slight fever), rash, and joint pain (mainly the small joints of hands and legs), or conjunctivitis. Other typical symptoms are headache, pain in the posterior eye socket, muscle pain, inappetence, stomach ache, and nausea. According to data of epidemic areas, few patients has neurological complication (e.g. GBS) or immune complication (e.g. ITP). More results demonstrate that these abnormalities are related to the Zika virus. Also, infants may have microcephaly when the pregnant woman has Zika virus.
[User]: Bye
[AidIR]: Bye!

(**b**)

Figure 16. AidIR requests more information: (**a**) original dialog; and (**b**) dialog in English.

6. Conclusions

AidIR enables users to retrieve information on diseases related to coronaviruses and diseases transmitted by vector mosquitoes by using natural language interaction and Line chat medium. In subjective evaluation, we asked users to rate AidIR in the range between −2 and 2. The average scores of intuitiveness, usability, and user experiences were 0.8, 0.8, and 1.05, respectively. Moreover, we asked those users to evaluate AidIR by answering yes–no questions. The results show that AidIR is better than systems using GUI and single-turn dialog systems. Furthermore, most of the users will use systems such as AidIR to retrieve information. Finally, we asked those users to provide the upsides and downsides of AidIR. Most users thought that AidIR was more convenient for information retrieval. In addition to subjective evaluation, we also compared AidIR with systems using GUI. Systems using GUI cannot guide users to provide specific information. This system will only return as much information as possible. Since systems with GUI do not have a tracking system, users need to provide the same information and they cannot rectify the error of the system. Furthermore, systems with GUI do not exploit the context to retrieve the information and tokenize the words. This mechanism results in general information. As a result, end-users may feel frustrated since end-users usually prefer specific information.

In the pre-processing stage, we operationalized the taxonomy and replace some types of numbers with special tokens. In the taxonomy operationalization stage, we found that some diseases have similar features. These characteristics may hinder the classification of the diseases. Therefore, we operationalized the diseases with similar characteristics into the same category. In the token replacement stage, we wrote regular expressions to match some categories of numbers and replaced them with special tokens to normalize the sentence. According to our experiments, this method yields better results in both the mask LM task and the downstream task. Specifically, the output probability in the mask LM task is more peak when we replace some types of numbers with special tokens. In terms of the downstream task, the accuracy of the ABERT model trained with these special tokens is 12% higher in the domain classification task and 5% higher in the sequence label task. The reason for which the accuracy of the ABERT model in the sequence label task only slightly decays is because the weight of the sequence label task is greater than that of the domain classification task. Therefore, these experimental results demonstrate the effectiveness of our methods.

In terms of NLU models, we trained a BERT model to perform domain classification and sequence labels simultaneously. To this end, we use a loss function with the weighting method that we design to train the model for two tasks. The accuracy of both tasks reaches 92%. Furthermore, our weighting method enables us to give the task that we want to enhance its accuracy with a higher weight. According to our experiments, we can enlarge the accuracy of the task by increasing the weight of that task. However, we need to find the trade-off. That is, we cannot overvalue one task and undervalue another task. Moreover, we examine whether fine-tuning the model with disease corpus and the mask LM task produces a better result. The experiment demonstrates that the accuracy of the model, fine-tuned with disease corpora and the mask LM task, is 24% higher in the domain classification task and 3% higher in the sequence label task.

To make the system further learn the dialog policy, we deployed the RL algorithm to AidIR. To prevent bad interaction in the beginning, we warmed up the model with the SL task. Then, we employed the A2C algorithm to enable AidIR to continue learning the dialog policy online. Moreover, we parallelized this process to prevent this process from influencing the interaction between users and AidIR.

7. Future Works

Future trajectories are divided into two aspects: models and user experiences. In terms of models, we will research how to enable the system to be optimized as an end-to-end architecture. This way will enable us to treat the whole dialog system as one system instead of the combination of several systems. Furthermore, we will investigate how to apply reading comprehension approaches to DST and invent more complex functions to give the crucial task with the higher weight for the ABERT model. In terms of user experiences, we will extend our work to multimodality and multilingualism with the few-shot learning method. To make the response more human, we will expand our response generation to the language model generation method. Furthermore, we will conduct interviews with users to pinpoint the user requirement and improve the usability, intuitiveness, and user experiences of the system.

Author Contributions: Conceptualization, D.-J.W., T.-Y.C. and C.-Y.S.; data curation, C.-Y.S.; formal analysis, C.-Y.S.; investigation, C.-Y.S.; methodology, C.-Y.S.; project administration, T.-Y.C.; resources, C.-Y.S.; software, C.-Y.S.; supervision, D.-J.W. and T.-Y.C.; validation, C.-Y.S.; visualization, C.-Y.S.; writing—original draft, C.-Y.S.; writing—review and editing, D.-J.W., T.-Y.C. and C.-Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in experiments are available on request from the corresponding author. The data for the ABERT model are published to https://github.com/chiayisu/ABERT_Corpus, accessed on 10 October 2021.

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

References

1. Digital 2021: Global Overview Report. Available online: https://datareportal.com/reports/digital-2021-global-overview-report (accessed on 30 June 2021).
2. Tagliabue, F.; Galassi, L.; Mariani, P. The "Pandemic" of Disinformation in COVID-19. *SN Compr. Clin. Med.* **2020**, *2*, 1287–1289. [CrossRef] [PubMed]
3. Swire-Thompson, B.; Lazer, D. Public Health and Online Misinformation: Challenges and Recommendations. *Annu. Rev. Public Health* **2020**, *41*, 433–451. [CrossRef] [PubMed]
4. Loomba, S.; de Figueiredo, A.; Piatek, S.J.; de Graaf, K.; Larson, H.J. Measuring the impact of COVID-19 vaccine misinformation on vaccination intent in the UK and USA. *Nat. Hum. Behav.* **2021**, *5*, 337–348. [CrossRef] [PubMed]
5. Leung, J.; Schoultz, M.; Chiu, V.; Bonsaksen, T.; Ruffolo, M.; Thygesen, H.; Price, D.; Geirdal, A. Concerns over the spread of misinformation and fake news on social media—Challenges amid the coronavirus pandemic. In Proceedings of the 3rd International Electronic Conference on Environmental Research and Public Health, Online, 11–25 January 2021; pp. 1–6.
6. PubMed. Available online: https://pubmed.ncbi.nlm.nih.gov/ (acessed on 30 June 2021).
7. LINE—Statistics and Facts. Available online: https://www.statista.com/topics/1999/line/#dossierKeyfigures (accessed on 30 June 2021).
8. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; pp. 38–45. [CrossRef]
9. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [CrossRef]
10. Chen, H.; Liu, X.; Yin, D.; Tang, J. A Survey on Dialogue Systems: Recent Advances and New Frontiers. *SIGKDD Explor. Newsl.* **2017**, *19*, 25–35. [CrossRef]
11. Williams, J.D.; Raux, A.; Henderson, M. The Dialog State Tracking Challenge Series: A Review. *Dialogue Discourse* **2016**, *7*, 4–33. [CrossRef]
12. Jurafsky, D.; Martin, J.H. Speech and Language Processing. 2020. Available online: https://web.stanford.edu/~jurafsky/slp3/ (accessed on 30 June 2021).
13. Chen, Y.N.; Celikyilmaz, A.; Hakkani-Tür, D. Deep Learning for Dialogue Systems. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 8–14.
14. Raschka, S. Naive Bayes and Text Classification I-Introduction and Theory. *arXiv* **2017**, arXiv:1410.5329.
15. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv* **2016**, arXiv:1609.08144.
16. Website of Center for Disease Control in Taiwan, ROC. Available online: https://www.cdc.gov.tw/ (accessed on 30 June 2021).

17. Regular Expression Language-Quick Reference. 2021. Available online: https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference (accessed on 30 June 2021).
18. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318. [CrossRef]
19. Jieba Tokenization Tool. Available online: https://github.com/fxsjy/jieba (accessed on 30 June 2021).
20. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 2, pp. 3111–3119.
21. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
22. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. 9.8.1735. [CrossRef] [PubMed]
23. Lee, J.Y.; Dernoncourt, F. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 515–520. [CrossRef]
24. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Harley, T.; Lillicrap, T.P.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1928–1937.
25. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256. [CrossRef]
26. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In Proceedings of the 12th International Conference on Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December 1999; pp. 1057–1063.
27. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An introduction*; MIT Press: Cambridge, MA, USA, 2018.
28. Vitay, J. Deep Reinforcement Learning. Available online: https://julien-vitay.net/deeprl/ (accessed on 30 June 2021).
29. Su, P.H.; Gasic, M.; Mrksic, N.; Rojas-Barahona, L.; Ultes, S.; Vandyke, D.; Wen, T.H.; Young, S. Continuously Learning Neural Dialogue Management. *arXiv* **2016**, arXiv:1606.02689.
30. Su, P.H.; Budzianowski, P.; Ultes, S.; Gašić, M.; Young, S. Sample-efficient Actor-Critic Reinforcement Learning with Supervised Data for Dialogue Management. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, 7–9 September 2017; pp. 147–157. [CrossRef]
31. Threading—Thread-Based Parallelism. Available online: https://docs.python.org/3/library/threading.html#module-threading (acessed on 10 January 2022).
32. PyTorch. Available online: https://pytorch.org/ (acessed on 30 June 2021).
33. Scikit-Learn. Available online: https://scikit-learn.org/stable/index.html (acessed on 30 June 2021).
34. LINE Developers. Available online: https://developers.line.biz/en/ (acessed on 30 June 2021).
35. re—Regular Expression Operations. Available online: https://docs.python.org/3/library/re.html (accessed on 10 January 2022).
36. Vig, J. A Multiscale Visualization of Attention in the Transformer Model. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Florence, Italy, 28 July–2 August 2019; pp. 37–42. [CrossRef]
37. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.