Computers, Materials & Continua
DOI:10.32604/cmc.2022.023127
Article

Tech Science Press

The Impact of Semi-Supervised Learning on the Performance of Intelligent Chatbot System

Sudan Prasad Uprety and Seung Ryul Jeong*

The Graduate School of Business Information Technology, Kookmin University, Seoul, 02707, Korea
*Corresponding Author: Seung Ryul Jeong. Email: srjeong@kookmin.ac.kr
Received: 29 August 2021; Accepted: 29 September 2021

Abstract: Artificial intelligent based dialog systems are getting attention from both business and academic communities. The key parts for such intelligent chatbot systems are domain classification, intent detection, and named entity recognition. Various supervised, unsupervised, and hybrid approaches are used to detect each field. Such intelligent systems, also called natural language understanding systems analyze user requests in sequential order: domain classification, intent, and entity recognition based on the semantic rules of the classified domain. This sequential approach propagates the downstream error; i.e., if the domain classification model fails to classify the domain, intent and entity recognition fail. Furthermore, training such intelligent system necessitates a large number of user-annotated datasets for each domain. This study proposes a single joint predictive deep neural network framework based on long short-term memory using only a small user-annotated dataset to address these issues. It investigates value added by incorporating unlabeled data from user chatting logs into multi-domain spoken language understanding systems. Systematic experimental analysis of the proposed joint frameworks, along with the semi-supervised multi-domain model, using open-source annotated and unannotated utterances shows robust improvement in the predictive performance of the proposed multi-domain intelligent chatbot over a base joint model and joint model based on adversarial learning.

Keywords: Chatbot; dialog system; joint learning; LSTM; natural language understanding; semi-supervised learning

1 Introduction

Natural language understanding (NLU) and Speech understanding (SU) play a significantly important role in human-computer interaction (HCI) applications. Intelligent NLU systems, including chatbots, robots, voice control interfaces, and virtual assistants, are well-known HCI applications developed to communicate with humans via natural language. HCI is now a global trend and has drawn attention from different communities with the advancement and rapid development of machine learning (ML) and deep neural network (DNN) and reinforcement learning. ELIZA [1] was the first

machine with ability to exhibit human behavior to understand human language and communicate with humans using pattern matching to respond to user. The modeling process of a single domain conversational system or intelligent chatbot consists of detecting intent and recognizing entities from the user query. Virtual customer assistants, or chatbots, reduce information overload and call center efforts, enabling better customer experience (CX) on HCI applications or company websites. Some institutions also deploy role-based assistants that can significantly help improve interactions with their customers, business partners, and employees. By reducing the complexity of data and rules, organizations can focus on repetitive and simple interactions where customer needs are well-satisfied and understood. Organizations are struggling to manage the growth of such user query data. They have been implementing intelligent chatbot to provide service to customers 24/7 with or without call center help to address these issues. Such intelligent systems have three most important parts: domain classification, intent detection, and entity recognition. For a multi-tasking chatbot, the domain classification model first classifies the domain and then intent and entity are recognized based on the frames of the classified domain, as shown in Fig. 1. A large amount of user-annotated data is needed to train a multi-domain dialog system. Major intelligent chatbot systems, such as Amazon Alexa, Apple Siri, Google Dialogflow, IBM Watson, Microsoft Cortana, and Samsung Bixby support multi domain conversation [2]. A typical multi-tasking or multi-domain chatbot system (as shown in Fig. 1) mainly has domain classification, intent prediction, entity recognition, and response generation or dialog management parts. Most intelligent chatbot process user queries in a sequential order: domain classification, intent prediction, slot prediction. Each has its separate machine learning (ML) model and is predicted in the sequential order. A large number of user-annotated examples of utterances in each domain is essential before training the model. In addition, separate models are generated for the domain, intent, and entity, making it difficult to manage large sets of models. Furthermore, with this approach, an error in the domain prediction step may lead to errors in intent prediction and entity recognition, ultimately reducing predictive performance of the chatbot. Typical supervised ML algorithms such as Bayesian algorithm, Support Vector Machine (SVM), Logistic Regression, and Neural Networks (NNs) could extract domain and intent from user queries with separate model. However, the advanced deep learning (DL) approaches, increased computing powers, generating large number of open-source dataset enable training a single joint model for domain classification, intent prediction, and entity recognition using a single set of utterances [3] containing multiple domain, intent, and slot or entity information, reducing the number of trained ML models [4].

This study reduces human efforts for manual annotation of utterances by incorporating unannotated datasets from various data sources, such as user query logs into a DNN algorithm, i.e., a single jointly trained long short-term memory (LSTM) based NLU model of a multi-domain intelligent chatbot. The single jointly trained LSTM based NLU model reduces the number of classification and recognition models used in sequential approaches and attempts to mitigate downstream error propagation. LSTM was proposed in 1997 by Hochreiter and Schmidhuber for sequential modeling [5], which is a concept of adding an extra memory cell to a recurrent neural network (RNN), achieving better performance in representing and storing historical information. In the standard LSTM network, information transmission is one-way, and each memory cell can use historical information but cannot use the future one. Bidirectional LSTM (Bi-LSTM, shown in Fig. 2) was introduced to transmit and store past and future information in each memory cell.

The principle of Bi-LSTM is to connect the same output of each input cell with two opposite timings. Forward LSTM can forward historical information to the next step, and LSTM networks directed backward can obtain future contextual information. Furthermore, extra unlabeled data [6] contributes to an increase in the information gain for DL model trained with the LSTM algorithm.

A single semi-supervised multi-domain joint model (SEMI-MDJM) based on LSTM outperforms a joint base model and an adversarial multi-domain joint model in each task i.e., domain classification, intent prediction, and entity recognition.

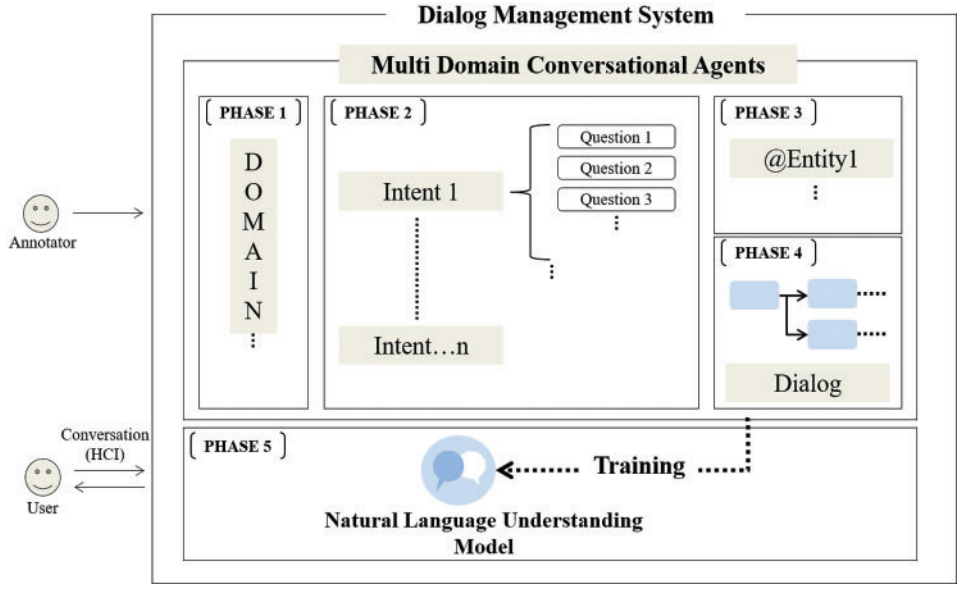

Figure 1: General architecture of multi-domain chatbot

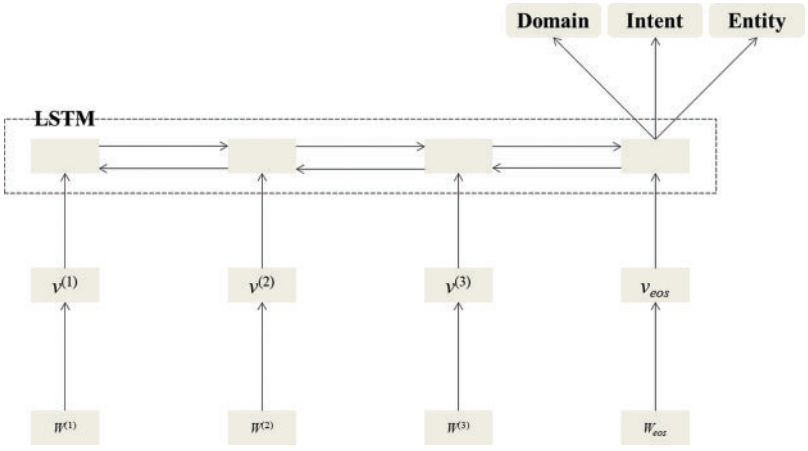

Figure 2: Bi-LSTM joint model

The remainder of this work is structured as follows. Next section presents related prior work on intelligent dialog system. Section 3 provides a proposed LSTM based semi-supervised joint framework. Section 4 presents the experimental results and detail analysis of predictive accuracies, and additional analyses on the importance of unannotated data in the context of a general chatbot having multiple domains. Finally, Section 5 concludes with a discussion and interesting areas for future study.

2 Literature Review

The first idea of an HCI application comes from the Turing test or "imitation" game created by Alan Turing in 1950. ELIZA was the first conversational system developed in 1966 based on pattern matching that respond to a user using keywords from the user query [1]. In 1980s, another HCI application called ALICE was developed using artificial intelligence markup language (AIML) [7] to mitigate the drawbacks of ELIZA. The performance of AIML was further improved [8] by applying multiple parameter design pattern to the decomposition rules. With the rapid development and advancement of ML algorithms, emergence of DL techniques, and natural language processing (NLP), these intelligent chatbot systems are gaining popularity in various fields. Conversational systems help reduce various costs by automating the workflow of a customer or call center, resulting in rapid response to customer queries [9]. Almansor and Hussain classified conversational systems into non-task-oriented and task-oriented categories [10]. Non-task-oriented systems are generally retrieval-based chatbots, which provide a similar or highly ranked list of information related to user input. In contrast, task-oriented conversational systems are supervised or unsupervised models performing users' specific tasks based on ML algorithm rather than decomposition rules or keyword filtering. Recently, commercial chatbot systems such as Microsoft Cortana, IBM Watson, Amazon Alexa, Google Dialogflow, Apple Siri, and Bixby are gaining interest from organizations [11]. These systems are mainly implemented in medical education, health assistance, educational system, e-learning, e-commerce [12], sports, games, privacy, infrastructure, and other fields [2]. Recently, public administrators have begun implementing chatbot systems for real-time customer services [13]. Autonomous vehicles and smart home systems also embed natural language interactions applications [14]. The implementation of these dialog systems requires technical knowledge about NLP and NLU [15–17]. Recent new studies report various new NLP and NLU, such as bag-of-concepts and bag-of-narratives [18].

Although there are several technical and logical parts involved in implementing intelligent chatbot systems, NLU is at the core part of a chatbot. In an intelligent chatbot, the role of NLU is to parse the user query and learn what the user means. NLU systems contain three main subsystems: domain classifier, intent detector, and entity recognition models [19]. Generally, a multi domain chatbot has three unsupervised or supervised ML models for recognizing each field. Different supervised and unsupervised learning algorithms include term frequency and inverse document frequency (TF-IDF), bag of words, word2vec, SVM, Bayes algorithm, NNs, boosting, maximum entropy, and deep belief networks [20] are widely applied to extract intent and slots in sequential NLU models. These separate pipelined ML models are created using a large number of utterances or examples [3]. Creating and annotating these utterances demands huge human efforts. Recently, much open research shares previously annotated large datasets from diverse domains in multiple languages. In addition, unannotated user-query data can be used and analyzed in the future. Vedula et al. [21] curated and released an annotated dataset of 25 k utterances for developing an intent model. Schuster et al. [22] curated 57 k annotated examples for English, Thai, and Spanish languages for three different domains – weather, alarm, and reminder – to develop cross-lingual dialog system. Larson et al. [23] evaluated for out of scope with a dataset containing 150 intent classes from 10 different domains. Furthermore, these sequential frameworks are at a high risk of introducing downstream errors to the intent detection and entity recognition phase. Since each predictive model is trained with sequence of text corpus, contextual information of the previous step has significant importance for traditional ML algorithms and recent DL approaches. These text data i.e., utterances or examples are time-series in nature, for which an LSTM-based DL framework demonstrates state-of-the-art performance [24].

2.1 Domain Prediction

Domain prediction is the process of filtering user input to a specific category in a multi-tasking dialog system. Many previous works on domain prediction exist. Hakkani-Tur et al. [25] proposed a semi-supervised domain prediction model using AdaBoost with user click logs on Bing web search engine. Zheng et al. [26] proposed an out-of-domain detection mechanism to avoid unnecessary responses to user input. Xu et al. [27] proposed a contextual domain classification mechanism to reduce consecutive queries by a user to different domains. Gupta et al. [28] proposed an RNN-based context encoding method to improve the predictive accuracy and computational efficiency of an NLU model using two different domains.

2.2 Intent Detection

Intent prediction is the main part of NLU system. Intent means what a user means or wants to obtain from the system. Although traditional intent predictor models are based on SVM and ANN, with the advancement in DL and sequence modeling, RNN and LSTM algorithms have demonstrated state-of-the-art performance in text classification tasks. Liu et al. [29] proposed attention-based RNN to predict intent and slot. In addition, a hybrid approach that combines LSTM and a convolutional neural network (CNN) shows performance improvement in intent prediction using the ATIS dataset [30]. Goo et al. [31] proposed Slot-Gated Bi-LSTM model with an attention mechanism to predict intent. Systems can make errors for similar words that appear in different contexts. Confusion2vec [32] can reduce confusing errors and predict the intent of user input. For multi-task and multi-turn dialog systems, previous domain information can be used as contextual information for new turns to improve the performance of dialog systems [33]. In addition, incorporating previous contextual session information [34] into intent and slot prediction models can improve predictive performance.

2.3 Entity Extraction or Slot Filling

Entity extraction, also called entity recognition (NER), extracts attributes such as location, place, date, and time from user query text. Entity extraction aims to extract entities of interest from user input text. As important information of user input can appear at any position, entity extraction becomes a more challenging process [24], making it difficult to extract entities from text. Early NER prediction systems relied on rules or dictionaries created by humans. After that, supervised learning based on SVM, decision trees, hidden Markov chain, conditional random fields, and dynamic vector representations [35] have been used to extract entities from text. Recently, ANNs and DL techniques such as LSTM, CNNs [36] have been used to extract entities from user text. Liu and Lane introduced slot filling based on RNN algorithms [29]. Dernoncourt et al. [37] proposed NeuroNER tools based on ANN for non-expert users of ANNs. Generally, models trained over previously build NER algorithms such as a distantly supervised slot-filling system [38] proposed at Stanford and a tweeter-based NER system [39] can improve the performance of entity extraction systems. The main challenges and misconceptions for NER system development were investigated in detail by Ratinov et al. [40] to improve prediction accuracy on the CoNLL dataset. An entity extraction model based on sequence modeling [41] can further improve its predictive performance.

Although these individual training approaches improve the performance of an individual model, there will be a lack of contextual sharing between each model, and the total number of models increases with the total number of domains. The total number of models for a typical traditional dialog system is calculated as Eq. (1).

$$\text{Total Predictive Models} = (2 \times \text{N}) + 1 \quad (1)$$

where N represents the total number of domains. The total number of predictive ML models in a typical traditional multi-domain chatbot system is the sum of domain predictive model, N number of intent, and N number of slot models. If the number of domain increases, the number of predictive models also increases. Thus, various joint training approaches that incorporates higher correlation information between intent and entity show better performance with a single joint predictive model.

2.4 Joint Training for Multi-Domain Intelligent Chatbot System

Joint training based on LSTM in a conversational system involves sharing cost or loss functions among domain, intent, and entity predictors. There are some prior works on joint modeling for intent detection and entity recognition. Liu et al. [29] proposed a joint model based on Attention Bi-RNN to recognize intent and entity with higher predictive performance. Ma et al. [30] introduced a sparse attention patterns to a jointly trained model based on LSTM for intent detection and slot extraction. Bekoulis et al. [42] applied adversarial learning to a joint model for various datasets, such as biomedical data, real estate data, and news data, achieving state-of-the-art performance for entity and relation extraction. Goo et al. [31] added related information for joint training between intent detector and slot extractor model. Zhang et al. [43] applied the hierarchical relationship between slots and intent to the joint model based on capsule neural networks. Recently, transfer learning i.e., pre-trained models, such as DialogGLUE (BERT show state-of-the-art-performance for joint model [44]. For multi-task-oriented conversational systems, a predictive domain model is trained separately, which could bring downstream error propagation, i.e., if an intelligent chatbot system fails to classify the domain then intent predictor and entity extractor does not work anymore [3].

There are some prior works on multi-task-oriented joint models based on LSTM with a single cell. Hakkani-Tur et al. [4] introduced the RNN-LSTM framework for a multi-task oriented chatbot. Kim and Lee used real user chatting logs from Microsoft Cortana and jointly trained the model with Bi-LSTM algorithm to inhance the classification accuracy by mitigating downstream error propagation [3]. We refer readers to Abdul-Kader et al. [12] and Ahmad et al. [15] studies, which provide comprehensive literature reviews of various ML and rule-based techniques used in chatbot systems or NLU studies.

2.5 Adversarial Learning

Adversarial learning regularizes neural networks and improves the classification accuracy of DNN algorithms by combining small noise or perturbations with annotated data, thereby increasing the loss function of a DL model [45]. Many DNN algorithms have recently been used in NLU and SU systems. Miyato et al. [6] observed the incorrect decision for DNNs with intentional random noise to the DNNs along with input examples. Furthermore, they proposed an object detection algorithm based on DNN using an adversarial learning to improve the classification accuracy of a ML model [6].

Semi-supervised learning with adversarial perturbations shows classification improvement for intelligent chatbot system having multi-domains [46]. Adversarial learning to DNNs (as shown in Fig. 3) generates small perturbations to the embedding layer along with input examples that gives variations to input, which the learning model can easily misclassify.

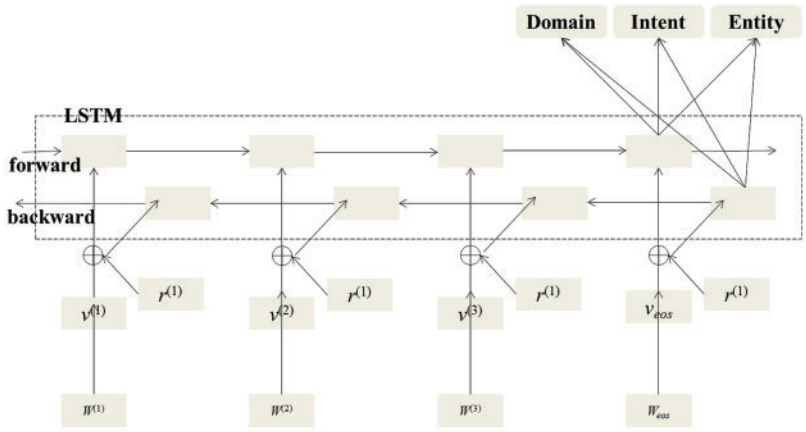

Figure 3: Adversarial joint model based on Bi-LSTM

2.6 Semi-Supervised Learning for NLU

Semi-supervised learning is the process of training ML model with both annotated and unannotated utterances. First, the supervised model developed with annotated or labeled dataset and then predicts and labels unannotated samples. Afterward, retraining the originally annotated datasets, along with machine-annotated datasets, creates new predictive supervised models. This entire training, predicting, and retraining process using predicted datasets, along with originally labeled utterances, presents the concept of semi-supervised learning shown in Fig. 4. The semi-supervised technique helps reduce human efforts in the manual annotation of utterances and helps create a self-learning model with robust information gain, ultimately improving a predictive performance or accuracy. A semi-supervised learning approach can help annotators annotate new user inputs with a small user-annotated dataset.

There are extensive prior studies on semi-supervised learning approaches for developing SLU and NLU models for intent prediction and entity extraction. Diverse techniques have been used to predict intent for a single domain dialog system using a semi-supervised learning approach [47]. A semi-supervised joint model for intent prediction and slot fillings [45,48] reduces human efforts in annotating examples, improving the model's performance with robust information gain. For further investigation, we refer readers to Singh et al. [11], which provide comprehensive literature reviews of data extraction, data processing, various data sources, and reinforcement and ensemble learning methods used in NLU studies.

Although semi-supervised learning has recently been used in multi-domain dialog systems, this study is, to the best of our knowledge, the first to apply semi supervised-learning to a single joint model based on LSTM. Compared with a prior joint model and adversarial joint model, our approach trains a single LSTM-based model using small user-annotated examples and unannotated samples from user chatting logs resulting in higher predictive performance and reduced human efforts in create annotated examples of AI dialog systems.

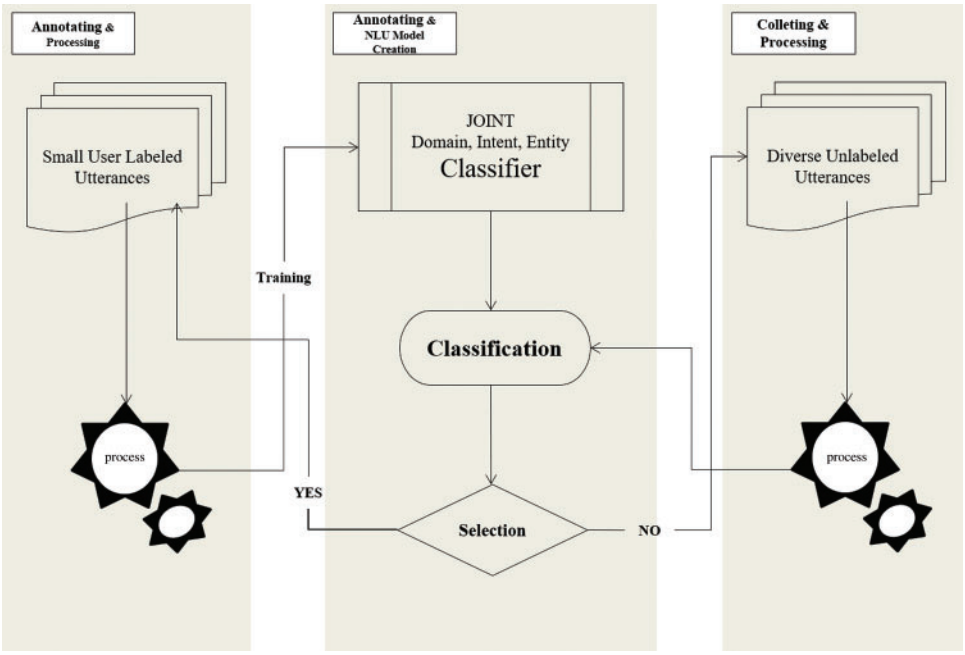

Figure 4: Semi-supervised joint model

3 Semi-Supervised Intelligent Chatbot

Our proposed SEMI-MDJM (shown in Fig. 4) focuses on self-automating the annotating process with user chatting logs, which could be the important data source for intelligent chatbot. Each component of SEMI-MDJM is discussed in the following subsection.

3.1 Data Preprocessing

User chatting logs are unstructured text data and should be converted into a structured example that a DNN algorithm can use it to train the model. Bag of words, term-frequency-matrices, and vector space [49] methods are widely applied to transform unstructured data into structured dataset. TF-IDF uses term frequency matrices to extract information from text data. Creating these matrices involves various data cleansing and wrangling approaches including tokenization, stemming, POS tagging as shown in Fig. 5. Then a word embedding set is created from the preprocessed cleaned corpus.

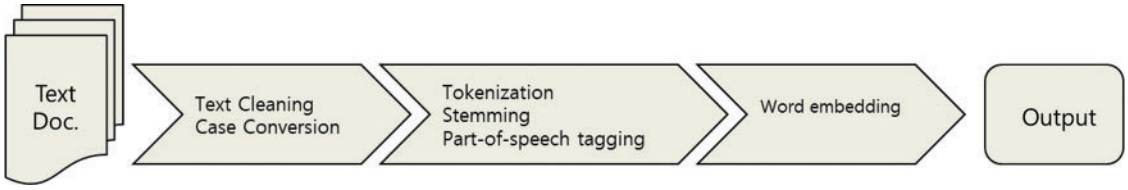

Figure 5: Text preprocessing

Furthermore, the previously developed joint model is used to predict unlabeled user chatting logs, and annotated utterances are added to the previous training dataset, retraining the model to increase the information gain for the LSTM cell. Then, utterances are preprocessed and fed into a Bi-LSTM

cell to extract previous and future information. Then the single LSTM model predicts domain, intent, and extract the entity.

3.2 Embedding and Bi-LSTM Layer

The embedding layer feeds the sequential data to a LSTM cell by creating embedding vector of words. Word embedding for word sequence $w_1 \ldots w_n \in W$ is given as Eq. (2):

$$\textbf{Word embedding}: \; e_w \in R^{64} \; for \; each \; w \in W \tag{2}$$

Fig. 2 shows a Bi-LSTM model with forward and backward propagation of information. Due to the bidirectional information propagation, previous and future contextual information can be memorized for each LSTM cell.

The final training objective of MDJM is to minimize shared loss among domain, intent, and entity. Total cumulative loss is calculated using Eq. (3):

$$L(\theta, \theta^d, \theta^i, \theta^t) = \sum_{\alpha \in \{\theta, \theta^d, \theta^i, \theta^t\}} L^{\alpha}(\theta) \tag{3}$$

The losses l^d, l^i, l^t of each output layers are calculated for each annotated utterance. Then, the shared loss among domain, intent, and entity is calculated as $l^d + l^i + l^t$ in each gradient step. Finally, the model is optimized using the shared loss θ. The algorithm of the proposed semi-supervised multi-domain chatbot system is designed as follows:

Algorithm 1: Semi-Supervised Multi Domain Intelligent Chatbot System

1: **Input:** Prepare and preprocess annotated and unannotated dataset
2: Create word embedding layer, $e_w \in R^{64} \; for \; each \; w \in W$
3: Create Bi-LSTM cells
4: Create encoder and decoder for each utterances
5: Train the model and calculate the loss function
   – Use seq2seq for slot loss
   – Use cross entropy for intent loss and domain loss
6: Calculate shared loss or cost function
   $L(\theta, \theta^d, \theta^i, \theta^t) = \sum_{\alpha \in \{\theta, \theta^d, \theta^i, \theta^t\}} L^{\alpha}(\theta)$
7: Optimize the model using Adam optimizer
8: Predict unlabeled data using the model created from **Step 1** to **Step 7**
9: Add predicted dataset to the original training dataset
10: Retrain the model by following **Step 1** to **Step 7**

3.3 Evaluation and Optimization

3.3.1 Evaluation Criteria

There are many standard performance matrices and criteria for comparing predictive performance between various classifiers [50]. The widely used measures in text classifications are predictive accuracy (ACC) and F-score. A detail description of these criteria can be clarified using the confusion matrix described in Tab. 1. The classification or predictive accuracy of a predictive model is defined as in Eq. (4):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{4}$$

In the above equation, TP denotes the false positive rate for a predictive model on all classes, whereas TN denotes the true negative rate. FP denotes the false positive, and FN is the false-negative rate of the model.

Table 1: Classification confusion matrix

		Actual classification	
		Positive	Negative
Predicted classification	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

The precision or positive predictive value of a given classification model is calculated as in Eq. (5):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

Recall, which is also called sensitivity or true positive rate of a classifier, is calculated using Eq. (6):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

Specificity, also called selectivity or true negative rate, is calculated as in Eq. (7):

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{7}$$

Another criterion is F1-score, which is the harmonic mean of precision and recall of a ML model, is calculated using Eq. (8):

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{8}$$

Area under curve (AUC) shown in Eq. (9) is another famous criterion to measure the accuracy ML algorithm:

$$\text{AUC} = \frac{\text{sensitivity} + \text{specificity}}{2} \tag{9}$$

In the above equation, sensitivity is the interaction between sensitivity and 1-specificity; specificity is the percentage of false ratings predicted as false. In this study, classification accuracy is used for performance comparison.

3.3.2 Optimization

Adam, stochastic gradient descent (SDG), and RMSProp are the three most widely used optimizers for ANNs and DL models. This study used the Adam optimizer for training our proposed model, adversarial model, and joint base model. The Adam optimizer helps control the sparse gradient problems of a model. It is a widely used optimization mechanism for DL applications such as NLU and SU models by expanding stochastic gradient descent.

4 Experiment

This study used 43 k of user annotated dataset containing weather, alarm, and reminder domains (shown in Fig. 6) of multi-domain intelligent chatbot system [22,51]. The dataset contains three different domains with12 intent labels and 11 unique entities.

weather/find	8:19:datetime,20:28:weather/noun,33:46:location	Give me most recent forecast for half moon bay
weather/find	8:19:datetime,20:28:weather/noun,33:46:location	Give me most recent forecast for half moon bay
weather/find	8:19:datetime,20:28:weather/noun,33:46:location	Give me most recent forecast for half moon bay
weather/find	8:19:datetime,20:28:weather/noun,33:46:location	Give me most recent forecast for half moon bay
weather/find	8:16:datetime,19:32:location,33:47:weather/noun	give me Thursday's half moon bay weather report
weather/find	8:16:datetime,19:32:location,33:47:weather/noun	give me Thursday's half moon bay weather report
weather/find	8:16:datetime,19:32:location,33:47:weather/noun	give me Thursday's half moon bay weather report
weather/find	8:16:datetime,19:32:location,33:47:weather/noun	give me Thursday's half moon bay weather report
weather/find	8:16:datetime,19:32:location,33:47:weather/noun	give me Thursday's half moon bay weather report
weather/find	12:23:datetime,24:37:location,38:52:weather/noun	Give me the most recent half moon bay weather report
weather/find	12:23:datetime,24:37:location,38:52:weather/noun	Give me the most recent half moon bay weather report
weather/find	12:23:datetime,24:37:location,38:52:weather/noun	Give me the most recent half moon bay weather report
weather/find	12:23:datetime,24:37:location,38:52:weather/noun	Give me the most recent half moon bay weather report
weather/find	12:23:datetime,24:37:location,38:52:weather/noun	Give me the most recent half moon bay weather report
weather/find	7:20:location,23:37:weather/noun,42:48:datetime	i need half moon bay's weather report for sunday
weather/find	7:20:location,23:37:weather/noun,42:48:datetime	i need half moon bay's weather report for sunday
weather/find	7:20:location,23:37:weather/noun,42:48:datetime	i need half moon bay's weather report for sunday
weather/find	7:20:location,23:37:weather/noun,42:48:datetime	i need half moon bay's weather report for sunday
weather/find	7:20:location,23:37:weather/noun,42:48:datetime	i need half moon bay's weather report for sunday
weather/find	7:20:location,23:37:weather/noun,42:48:datetime	i need half moon bay's weather report for Monday

Figure 6: Raw dataset-sample

A sample of the preprocessed user utterances is shown in Fig. 7. Furthermore, the publicly available large unannotated user chatting log dataset [52] of 25 k user utterances from 21 domains are collected and only 2 k, i.e., 2,510 of the unannotated user queries (alarm, reminder, and weather) dataset is used for semi-supervised learning.

The utterances are restructured (as shown in Fig. 7) into annotated sets of user queries, entities, intent labels in respective order. User queries are enclosed with the BOS and EOS symbols. Dataset are then divided into training, evaluation, and testing dataset in 70:20:10 ratios, as shown in Tab. 2. Annotated and unannotated utterances are then preprocessed using Python NLU tokenization library. Each input example size is fixed to 50 characters and created word embedding of size 64. Then LSTM model from TensorFlow library is used to train and predict user queries.

```
BOS I want to be reminded to file tax EOS   O O O O O O B-reminder-todo set_reminder reminder
BOS I don't want to forget to file taxes EOS    O O O O O O B-reminder-todo set_reminder reminder
BOS My alarms EOS   O O show_alarms alarm
BOS show my alarms EOS  O O O show_alarms alarm
BOS snooze EOS  O snooze_alarm alarm
BOS what's the weather in Seattle EOS   O O B-weather-noun O B-location find weather
BOS Weather EOS B-weather-noun find weather
BOS What's the weather today EOS    O O B-weather-noun B-datetime find weather
BOS What's the weather tomorrow EOS O O B-weather-noun B-datetime find weather
BOS What's the weather this weekend EOS O O B-weather-noun B-datetime find weather
BOS Temperature EOS B-weather-noun find weather
BOS What's the temperature today EOS    O O B-weather-noun B-datetime find weather
BOS What's the temperature next week EOS    O O B-weather-noun B-datetime find weather
BOS What's the forecast this week EOS   O O B-weather-noun B-datetime find weather
BOS Change my 3 PM alarm to the next day EOS    O O B-datetime O B-alarm-alarm_modifier B-datetime modify_alarm alarm
BOS Change my 3 PM alarm to the next day EOS    O O B-datetime O B-alarm-alarm_modifier B-datetime modify_alarm alarm
BOS Change my 3 PM alarm to the next day EOS    O O B-datetime O B-alarm-alarm_modifier B-datetime modify_alarm alarm
BOS what's the weather morgan hill EOS  O O B-weather-noun B-location find weather
BOS weather in the tri-cities today EOS B-weather-noun O O B-location B-datetime find weather
BOS remind me to grab paperwork  before leaving the office EOS  O O O B-reminder-todo set_reminder reminder
BOS Sound an alarm in 45 seconds EOS    O O O B-datetime set_alarm alarm
BOS Alarm please EOS    O O set_alarm alarm
```

Figure 7: Preprocessed sample dataset

Table 2: Train, Eval test dataset

Train (70%)	Eval (20%)	Test (10%)	Total (100%)
30,521	8,621	4,181	43,323

The experiments were conducted by using tensorflow 1.10.0 library on python 3.6. The experimental platform runs Windows 10 with an Intel Core CPU at a clock speed of 1.60 GHz with 8 GB RAM.

To evaluate SEMI-MDJM, we conducted experimental analysis and compared with a prior MDJM and "multi-domain joint model with adversarial learning" (MDJM-ADV) [51]. SEMI-MDJM is created by annotating publicly available user chatting logs using MDJM and retraining the proposed model by adding this predicted dataset to the original training sets. LSTM cell of each model is created with 100 hidden neurons. Then the model is trained for 20 epochs and optimized with Adam optimizer. The learning rate is set to 0.01 and batch size of training dataset is set to 16. The MDJM shares the loss function among domain, intent, entities predictors, whereas MDJM-ADV further adds the adversarial loss to the original MDJM model. Incorporating user chatting logs into the base MDJM provides information gain for each output layer. Fig. 8 shows the training and test loss for MDJM, MDJM-ADV and SEMI-MDJM.

Tab. 3 presents the classification accuracy of previous joint model along with our proposed SEMI-MDJM. SEMI-MDJM outperforms the joint base model, MDJM, and the adversarial joint model, MDJM-ADV, in terms of classification accuracy for the domain, intent, and entity.

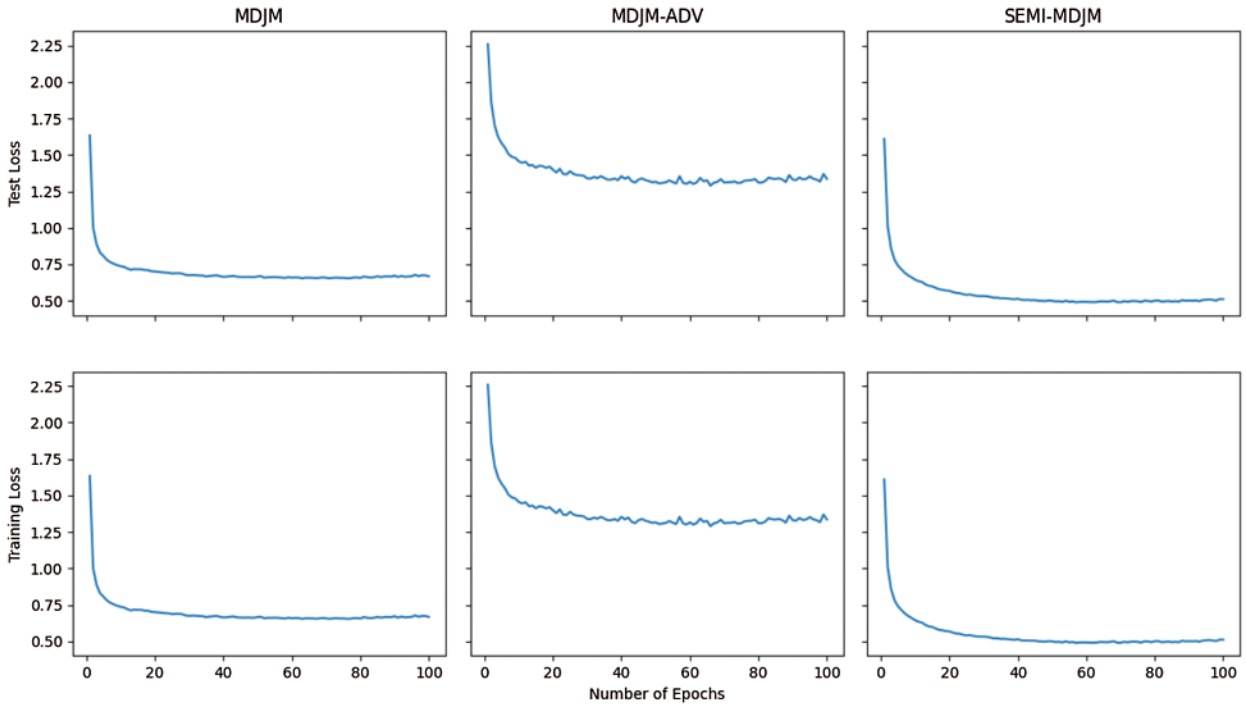

Figure 8: Test and training loss for each model

Table 3: Accuracy of each model

Model	MDJM (%)	MDJM-ADV (%)	SEMI-MDJM (%)
Domain	67.98	68.03	76.65
Intent	78.54	78.68	84.84
Entity	58.12	58.01	59.37
Avg. accuracy	68.21	68.24	73.62

5 Conclusion

In this study, we proposed a semi-supervised joint model, SEMI-MDJM, for intelligent chatbot system to extract the domain, intent, and entity of user queries using a single model ML model based on LSTM to mitigate the propagation of downstream error. This is a limitation of the typical sequential approach and reduces the effort required to manage a large number of NLU predictive models and manual data annotation. Experimental results showed a significant improvement in the predictive performance of each model, i.e., - domain, intent, and entity-predictions, based on semi-supervised learning compared to the joint base model and joint model with adversarial learning. The proposed SEMI-MDJM reduces the number of trained models to one along with the self-annotation process, which reduces human effort necessary to annotate and manage multiple intent detector and entity extractor. In addition, it provides a self-learning approach to the conversational dialog system by continuously incorporating domain-related utterances from user chatting logs into the initially

developed MDJM. Furthermore, it reduces the human effort required to annotate a large number of the domain, intent, and entity examples. We encourage testing our proposed SEMI-MDJM model with domain related to education, health for various languages with large datasets for future study. In addition, incremental prediction and annotation of all unannotated dataset can also improve and reduce the proposed model's overfitting problem.

Funding Statement: This research was supported by the BK21 FOUR (Fostering Outstanding Universities for Research) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NFR).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[2] A. de Barcelos Silva, M. M. Gomes, C. A. da Rosa Righi, J. L. V. Barbosa and G. De Doncker *et al.*,"Intelligent personal assistants: A systematic literature review," *Expert Systems with Application*, vol. 147, pp. 113–193, 2020.

[3] Y. B. Kim, S. Lee and K. Stratos, "Onenet: Joint domain, intent, slot prediction for spoken language understanding," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Okinawa, Japan, pp. 547–553, 2017.

[4] D. Hakkani-Tür, G. Tür, A. Celikyilmaz, Y. N. Chen, J. Gao *et al.*, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Proc. of Interspeech*, San Francisco, USA, pp. 715–719, 2016.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] T. Miyato, A. M. Dai and I. Goodfellow, "Adversarial training methods for semi-supervised text classification", *arXiv preprint arXiv*: 1605.07725, 2016.

[7] R. Wallace, "*The elements of AIML style*", ALICE AI Foundation, Inc., United States, 2016. [Online]. Available: https://dokumen.tips/documents/the-elements-of-aiml-style-2-the-elements-of-aiml-style-is-a-no-nonsense-technical.html.

[8] I. M. Sukarsa, P. W. Buana and U. Yogantara, "Multi parameter design in AIML framework for balinese calendar knowledge access," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 1, pp. 114–130, 2020.

[9] T. P. Nagarhalli, V. Vaze and N. K. Rana, "A review of current trends in the development of chatbot systems," in *2020 6th Int. Conf. on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, pp. 706–710, 2020.

[10] E. H. Almansor and F. K. Hussain, "Survey on intelligent chatbots: State-of-the-art and future research directions," in *Conf. on Complex, Intelligent, and Software Intensive Systems*, Springer, Cham, pp. 534–543, 2019.

[11] S. Singh and H. K. Thakur, "Survey of various AI chatbots based on technology used," in *2020 8th Int. Conf. on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, pp. 1074–1079, 2020.

[12] S. A. Abdul-Kader and J. C. Woods, "Survey on chatbot design techniques in speech conversation systems," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 7, pp. 72–80, 2015.

[13] D. A. Park, "A study on conversational public administration service of the chatbot based on artificial intelligence," *Journal of Korea Multimedia Society*, vol. 20, no. 8, pp. 1347–1356, 2017.

[14] E. Okur, S. H. Kumar, S. Sahay, A. A. Esme and L. Nachman, "Natural language interactions in autonomous vehicles: Intent detection and slot filling from passenger utterances," *ArXiv Preprint ArXiv*: 1904.10500, 2019.
[15] N. A. Ahmad, M. H. Che, A. Zainal, M. F. Abd Rauf and Z. Adnan, "Review of chatbots design techniques," *International Journal of Computer Applications*, vol. 181, no. 8, pp. 7–10, 2018.
[16] J. Zhang, J. Zhang, S. Ma, J. Yang and G. Gui, "Chatbot design method using hybrid word vector expression model based on real telemarketing data," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 4, pp. 1400–1418, 2020.
[17] J. A. Nasir and Z. U. Din, "Syntactic structured framework for resolving reflexive anaphora in Urdu discourse using multilingual NLP," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 4, pp. 1409–1425, 2021.
[18] E. Cambria and B. White, "Jumping NLP curves: A review of natural language processing research," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014.
[19] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu *et al.*, "Towards end-to-end spoken language understanding," in *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, pp. 5754–5758, 2018.
[20] R. Sarikaya, G. E. Hinton and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 778–784, 2014.
[21] N. Vedula, N. lipka, P. Maneriker and S. Parthasarathy, "Open intent extraction from natural language interactions," in *Proc. of the Web Conf. 2020*, Association for Computing Machinary, New York, NY, USA, pp. 2009–2020, 2020.
[22] S. Schuster, S. Gupta, R. Shah and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog," *arXiv preprint arXiv*: 1810.13327, 2018.
[23] S. Larson, A. Mahendran, J. J. Peper, C. Clarke, A. Lee *et al.*, "An evaluation dataset for intent classification and out-of-scope prediction," *arXiv preprint arXiv*: 1909.02027, 2019.
[24] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, vol. 2, pp. 207–212, 2016.
[25] D. Hakkani-Tür, L. Heck and G. Tur, "Exploiting query click logs for utterance domain detection in spoken language understanding," in *2011 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, pp. 5636–5639, 2011.
[26] Y. Zheng, G. Chen and M. Huang, "Out-of-domain detection for natural language understanding in dialog systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1198–1209, 2020.
[27] P. Xu and R. Sarikaya, "Contextual domain classification in spoken language understanding systems using recurrent neural network," in *2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, pp. 136–140, 2014.
[28] R. Gupta, A. Rastogi and D. Hakkani-Tur, "An efficient approach to encoding context for spoken language understanding," *ArXiv Preprint ArXiv*: 1807.00267, 2018.
[29] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *ArXiv Preprint ArXiv*: 1609.01454, 2016.
[30] M. Ma, K. Zhao, L. Huang, B. Xiang and B. Zhou, "Jointly trained sequential labeling and classification by sparse attention neural networks," *ArXiv Preprint ArXiv*: 1709.10191, 2017.
[31] C. W. Goo, G. Gao, Y. K. Hsu, C. L. Huo, T. C. Chen *et al.*, "Slot-gated modeling for joint slot filling and intent prediction," in *Proc. of the, 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Lousisiana, vol. 2, pp. 753–757, 2018.
[32] P. G. Shivakumar, M. Yang and P. Georgiou, "Spoken language intent detection using confusion2vec," *ArXiv Preprint ArXiv*: 1904.03576, 2019.

[33] M. Mensio, G. Rizzo and M. Morisio, "Multi-turn QA: A RNN contextual approach to intent classification for goal-oriented systems," in *Companion Proc. of the Web Conf. 2018*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, pp. 1075–1080, 2018.
[34] A. Bhargava, A. Celikyilmaz, D. Hakkani-Tür and R. Sarikaya, "Easy contextual intent prediction and slot detection," in *2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, pp. 8337–8341, 2013.
[35] Y. Ji, C. Tan, S. Martschat, Y. Choi and N. A. Smith, "Dynamic entity representations in neural language models," *arXiv preprint arXiv*: 1708.00781, 2017.
[36] N. T. Vu, "Sequential convolutional neural networks for slot filling in spoken language understanding," *ArXiv Preprint ArXiv*: 1606.07783, 2016.
[37] F. Dernoncourt, J. Y. Lee and P. Szolovits, "NeuroNER: An easy-to-use program for named-entity recognition based on neural networks," *ArXiv Preprint ArXiv*: 1705.05487, 2017.
[38] M. Surdeanu, S. Gupta, J. Bauer, D. McClosky, A. X. Chang *et al.*, "Standford's distantly-supervised slot-filling system," 2011. [Online] Available: https://www.semanticscholar.org/paper/Stanford's-Distantly-Supervised-Slot-Filling-System-Surdeanu-Gupta/677455e832f1f07d060188238c4164e2450c3cd1.
[39] A. Ritter, S. Clark and O. Etzioni, "Named entity recognition in tweets: An experimental study," in *Proc. of the 2011 Conf. on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK, pp. 1524–1534, 2011.
[40] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proc. of the Thirteenth Conf. on Computational Natural Language Learning (CoNLL-2009)*, Boulder, Colorado, pp. 147–155, 2009.
[41] L. Zhao and Z. Feng, "Improving slot filling in spoken language understanding with joint pointer and attention," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, vol. 2, pp. 426–431, 2018.
[42] G. Bekoulis, J. Deleu, T. Demeester and C. Develder, "Adversarial training for multi-context joint entity and relation extraction," *ArXiv Preprint ArXiv*: 1808.06876, 2018.
[43] C. Zhang, Y. Li, N. Du, W. Fan and P. S. Yu, "Joint slot filling and intent detection via capsule neural networks," *ArXiv Preprint ArXiv*: 1812.09471, 2018.
[44] S. Mehri, M. Eric and D. Hakkani-Tur, "Dialogue: A natural language understanding benchmark for task-oriented dialogue," *ArXiv Preprint ArXiv*: 2009.13570, 2020.
[45] M. Koziński, L. Simon and F. Jurie, "An adversarial regularisation for semi-supervised training of structured output neural networks," *ArXiv Preprint ArXiv*: 1702.02382, 2017.
[46] O. Lan, S. Zhu and K. Yu, "Semi-supervised training using adversarial multi-task learning for spoken language understanding," in *2018 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, pp. 6049–6053, 2018.
[47] T. Gasanova, E. Zhukov, R. Sergienko, E. Semenkin and W. Minker, "A semi-supervised approach for natural language call routing," in *Proc. of the SIGDIAL 2013 Conf.*, Metz, France, pp. 344–348, 2013.
[48] S. Zhu, R. Cao and K. Yu, "Dual learning for semi-supervised natural language understanding," *ArXiv Preprint ArXiv*: 2004.12299, 2020.
[49] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.
[50] W. K. Sari, D. P. Rini and R. F. Malik, "Text classification using long short-term memory," in *2019 Int. Conf. on Electrical Engineering and Computer Science (ICECOS)*, Batam, Indonesia, pp. 150–155, 2019.
[51] S. P. Uprety and S. R. Jeong, "Adversarial training for multi domain dialog system," *Intelligent Automation & Soft Computing*, vol. 31, no. 1, pp. 1–11, 2022.
[52] X. Liu, A. Eshghi, P. Swietojanski and V. Rieser, "Benchmarking natural language understanding services for building conversational agents," *ArXiv Preprint ArXiv*, 1903.05566, 2019.