Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.018757
Article

Tech Science Press

Adversarial Training for Multi Domain Dialog 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: 20 March 2021; Accepted: 21 April 2021

Abstract: Natural Language Understanding and Speech Understanding systems are now a global trend, and with the advancement of artificial intelligence and machine learning techniques, have drawn attention from both the academic and business communities. Domain prediction, intent detection and entity extraction or slot fillings are the most important parts for such intelligent systems. Various traditional machine learning algorithms such as Bayesian algorithm, Support Vector Machine, and Artificial Neural Network, along with recent Deep Neural Network techniques, are used to predict domain, intent, and entity. Most language understanding systems process user input in a sequential order: domain is first predicted, then intent and slots are filled according to the semantic frames of the predicted domain. This pipeline approach, however, has many disadvantages including downstream error; i.e., if the system fails to predict the domain, then the system also fails to predict intent and slot. The main purpose of this paper is to mitigate the risk of downstream error propagation in traditional pipelined models and improve the predictive performance of domain, intent, and slot- all of which are critical steps for speech understanding and dialog systems- with a deep learning-based single joint model trained with an adversarial approach and long short-term memory (LSTM) algorithm. The systematic experimental analysis shows significant improvements in predictive performance for domain, intent, and entity with the proposed adversarial joint model, compared to the base joint model.

Keywords: Natural language understanding; speech understanding; dialog systems; joint learning; chatbot

1 Introduction

Speech understanding, natural language understanding (NLU) and natural language processing (NLP) are important parts of Human-computer interaction (HCI). Conversational systems or dialog systems, including virtual assistants, voice control interfaces, chatbots, and robots, are some examples of HCI application deployed to interact with users using human language. ELIZA [1] was the first machine to understand natural language for conversations with humans, based on decomposition rules that respond to humans via by keywords appearing in the user input text. The modelling process of a typical dialog system consists of predicting intent, i.e., what the customer really means, and extracting entities from the user input. In the case of a multi-task-oriented dialog system, the domain predictive model first predicts

the domain; intent and entity are then extracted in a stepwise manner based on the predicted domain, as shown in Fig. 1. Major dialog systems, such as Microsoft Cortana, Apple Siri, Google Dialogflow, Amazon Alexa, Samsung Bixby, and IBM Watson, support multi task-oriented dialog systems for different domains [2]. A multi-domain dialog system mainly has four key parts: domain prediction, intent detection, entity extraction, and dialog generation. Most language understanding systems process user query in a stepwise order; domain, intent, and slots each have their own trained model and are predicted in the respective order. However, this approach may result in prediction error for intent and entity. An error in the first step, i.e., domain prediction, leads to error in downstream tasks, i.e., intent detection and entity extraction, which ultimately reduces prediction accuracy for intent and entity. Typical machine learning algorithms such as Support Vector Machine (SVM), Bayesian algorithm, and Artificial Neural Network (ANN) have only been able to train individual predictive models for domain, intent, and entity. However, the emergence of deep learning technology and increased computing powers enable training of large joint single models for domain, intent, and entity with a single large dataset [3] which shares information between domain, intent, and entity and reduces the number of predictive models [4].

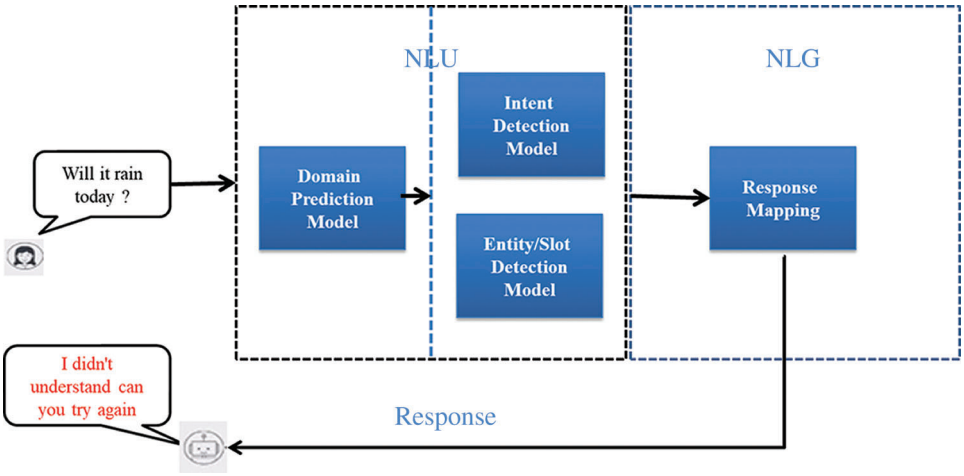

Figure 1: Downstream error for typical traditional dialog system

The main purpose of this study is to jointly train domain prediction, intent detection, and slot filling—all of which are critical steps for dialog systems or chatbots—with a single cell of bidirectional LSTM (Long Short Term Memory) algorithm and adversarial learning, to mitigate downstream error propagation and reduce the number of predictive models used in traditional pipelined approaches, ultimately helping to reduce the amount of human effort necessary to manage predictive models. LSTM is the first memory-based recurrent neural network (RNN) proposed by Hochreiter et al. [5] for time series or sequential data, and adversarial learning [6] helps to increase the loss function of the deep learning model with small inline perturbations to the LSTM cell along with user input. Jointly trained single models with adversarial learning (MDJM-ADV) based on a single cell of LSTM algorithm for multi-domain dialog systems show significant improvements in each task over a base joint multi-domain dataset.

2 Structure

The remainder of this study is structured as follows. Section 3 presents related prior work. Section 4 provides a proposed joint framework based on adversarial LSTM for a multi-domain dialog system. Section 5 presents the systematic analysis and experimental results for the respective predictive

performances, and additional analyses regarding the importance of adversarial learning and joint models in the context of a multi-domain dialog system. Finally, Section 6 concludes with a discussion and interesting areas for further study.

3 Literature Review

Since the first human computer program, ELIZA, was developed at MIT in 1966, there has been a tremendous amount of research in the field of natural language understanding and speech understanding. ELIZA is based on decomposition rules [1] that respond to humans through keywords appearing in the user input text. To mitigate some of the drawbacks of ELIZA, another chatbot called ALICE (Artificial Linguistic Internet Computer Entity) was developed in the 1980s based on AIML (Artificial Intelligent Markup Language) [7]; performance of AIML is further improved in pattern recognition with the help of multiple parameter design [8]. These kinds of applications are mainly enabled with the maturity of Human Language Understanding & Processing by computers and are called conversational agents, also called dialog systems or simply chatbot systems. With the rapid development of technology and advancement of machine learning algorithms, the HCI applications such as ELIZA and ALICE are gaining popularity in academia, as well as in organizations. Chatbot systems help to reduce operating costs by reducing and automating the workload to the customer center and higher management levels, resulting in increased availability for customer services [9].

Although there are various parts and systems involved in implementation of the dialog system, natural language understanding is at the heart of the conversational system. In a dialog or chatbot system, the role of natural language understanding is to receive user input, and to parse, preprocess, and understand what the user really means. Natural language understanding systems further contain three different parts: domain predictor, intent classifier, and entity or slot extractor models [10]. Generally, for task-oriented dialog systems, a natural language understanding system has three supervised or unsupervised learning models for domain prediction, intent classification, and entity extraction. Different machine learning techniques such as TF-IDF, word-to-vector [11], Bayes algorithm, SVM, ANN, Deep Belief Networks, boosting, and maximum entropy [12] are widely used to predict intent and entity in a traditional pipelined NLU system. However, these pipelined models are at increased risk of propagating downstream error to predict intent and entity. Given that all of these individually-trained predictive models (i.e., domain, intent, and entity predictive models) are based on text data—where contextual information has significant importance for any type of traditional machine learning algorithm or recent deep learning approaches, this text data can be treated as sequence or time series datasets, for which LSTM-based deep learning algorithm shows state-of-the-art performance [13].

3.1 Joint Training for Dialog System

Joint training in a dialog system is the process of sharing loss functions between each predictive model i.e., domain, intent and entity extractor. There is extensive prior research on joint modelling for intent and entity extraction. Liu et al. [14] proposed the Attention BiRNN joint model to predict intent and slot with higher accuracy. Ma et al. [15] introduced the sparse attention mechanism for a jointly-trained model for intent classification and slot filling using LSTM. Bekoulis et al. [16] proposed a model for joint entity and relation extraction with adversarial training for diverse data sets, such as news, real estate, and biomedical data that achieved state-of-the-art effectiveness. Goo et al. [17] added related information between intent and slot for joint training. Zhang et al. [18] considered the hierarchical relationship among each layer—words, slots, and intent—in joint models using Capsule Neural Networks. Recently pre-trained model or transfer learning on large unlabeled corpora or utterances such as BERT (Bidirectional Encoder Representations from Transformers), DialogGLUE shows the state-of-art-performance on joint intent prediction and slot filling models [19]. In the case of multi-domain dialog systems, the domain

model is generally separated from the intent and entity models, which could directly cause downstream errors, i.e., if the dialog system fails to predict the domain then it fails to predict intent and entity based on predicted domain frames [3]. Joint domain, intent, entity approaches have been proposed in prior studies as a way to reduce the risk of downstream error.

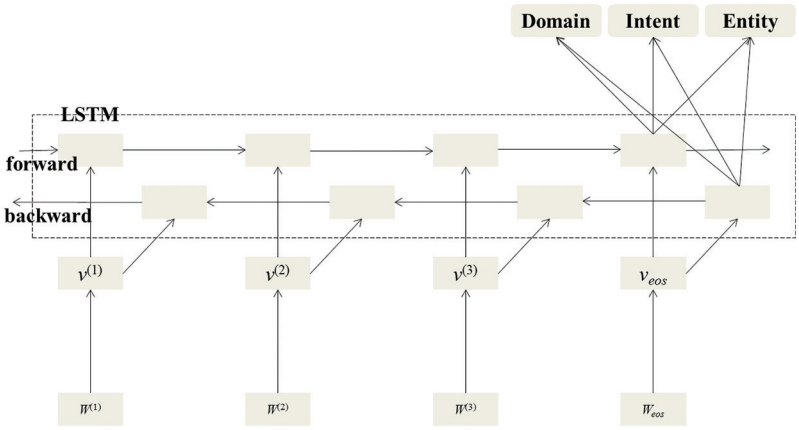

Figure 2: A bidirectional LSTM joint model

There are also several studies on multi domain joint models that leverage existing domains by predicting domain, intent and slot with a single cell of LSTM for multitask/multi-domain dialog systems. Hakkani-Tür et al. [4] proposed RNN-LSTM joint architecture for a multi-domain NLU model. Kim et al. [3] used real user data and jointly trained with bidirectional LSTM to improve predictive performance by reducing downstream error.

3.2 Adversarial Training

Adversarial learning is the process of regularizing the neural networks that help to enhance the predictive performance of neural network or deep learning approaches by combining small perturbations or noises along with the training data which increases the loss of a deep learning model [20]. Many neural network and deep learning approaches have recently been exploited in various natural language understanding and speech understanding systems. However, Miyato et al. [6] observed that adversarial examples, i.e., intentional small scale perturbations to the input of deep learning models may lead to incorrect decisions with high confidence. Further, they proposed an image recognition algorithm using an adversarial regularization approach, where a mixture of clean and small perturbations are applied along with an input dataset to enhance the predictive performance of the learning model [6]. If x and θ are the respective input and various parameters of a predictive model, the adversarial training to the predictive model adds the following term to its cost function:

$$\log p(y \mid x + r_{adv};\ \theta) \text{ where } r_{adv} = \arg\min \log p(y \mid x + r;\ \hat{\theta}\ ) \tag{1}$$

where r is an adversarial example on the input data and $\hat{\theta}$ denotes a set of constant parameters of a predictive model. At each training step, machine learning algorithms identify the worst case perturbations r_{adv} against the current trained model $p(y \mid x;\ \hat{\theta}\ )$ in the above equation with respect to θ.

Adversarial training with semi-supervised learning approaches shows performance improvement for multi-domain dialog systems [21]. Although adversarial training has recently been applied in NLP [6] task, this paper is, to the best of our knowledge, the first to apply adversarial training in a multi-domain joint dialog system. Unlike other regularization methods such as dropout or word dropout that introduce

random noise, adversarial training (as shown in Fig. 3) generates perturbations or random noise that are variations of input examples which can be easily misclassified by the learning model.

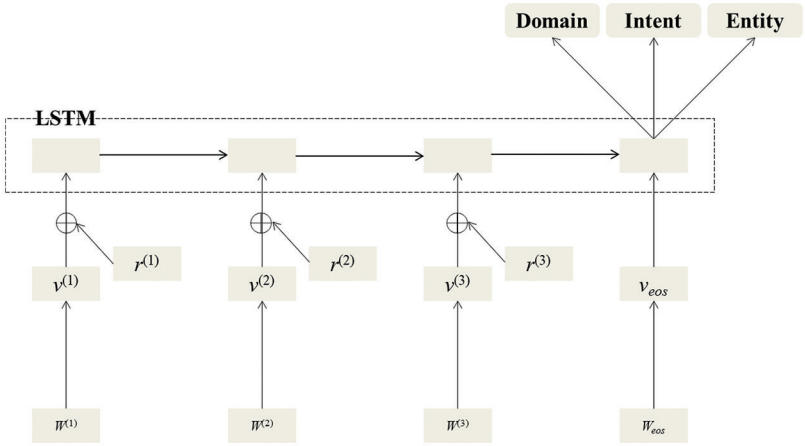

Figure 3: Adversarial-LSTM based joint predictive model

4 Adversarial Multi Domain Joint Model

Our proposed multi domain joint model (shown in Fig. 3) mainly focuses on the embedding layer with inline perturbations to the input, which is the main source of knowledge for sequence modeling for domain, intent, and entity in the LSTM cell. Each layer and data preprocessing step is curated in the following subsections.

4.1 Data Preprocessing

The textual data, which is normally unstructured in nature, should be transformed into a structured dataset that can be employed by a typical machine learning algorithm. Approaches such as frequency-based and vector space [22] methods can be used to extract information from unstructured textual data and transform it into structured data. TF-IDF is one such frequency-based algorithm for unstructured text processing based on the term frequency matrices. Generally, vector space methodology is widely used for text data embedding. It converts the original text documents into a vector feature space. As presented in Fig. 4, the unstructured text documents need to be converted into a structured dataset via various preprocessing techniques, since unstructured datasets cannot be incorporated into a machine learning model directly. The text processing process includes text cleansing, such as tokenizing, and removing special characters. Part-of-speech tagging annotates words with a syntactic category, e.g., nouns, verbs, etc. The stemming process replaces each term with its relative root stem. For example, the term "be" is the stem for words such as "is," "am," and "are." Stemming helps to reduce the number of features derived from examples. The number of features in the *corpus* can further be reduced by filtering irrelevant words. From this preprocessed cleaned *corpus*, a word-embedding vector space model is created.

As shown in Fig. 2, in case of the joint approach, there is a BiLSTM layer with three output layers performing domain, intent, and entity prediction, which are final outputs of the BiLSTM layer. We cumulate the losses of these three outputs, which will be minimized and optimized jointly. Crucially, the optimization of the joint model updates shared loss simultaneously, which is suitable for the domain, intent, and entity prediction process, allowing the joint model to not only avoid any downstream error but also helps to improve the predictive performance of each individual task by sharing loss function.

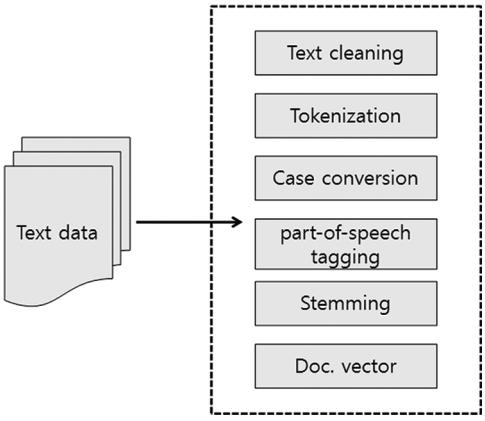

Figure 4: Unstructured text data processing approach

Further, the small worst perturbation in the embedding layer maximizes shred loss. The adversarial loss is then added to the joint loss of domain, intent, and slot to enhance the predictive performance of the multi-domain NLU model shown in Fig. 3. This architecture mainly focuses on the embedding layer, which is the main source of knowledge for sequence modeling in LSTM. Utterances are embedded along with the generated small perturbation i.e., worst noise data. The embedded examples are fed to a bidirectional LSTM to extract sequential and contextual information. The shared single LSTM cell then routes to three different output layers: domain prediction, intent detection, and slot prediction.

4.2 Embedding Layer

The first part of the LSTM cell is the input layer. We construct a word embedding vector layer in order to feed the input to the LSTM. Let $w_1 \ldots w_n \in W$ denote a word sequence, and then word embedding is given as follows:

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

4.3 Adversarial Training Layer

For the adversarial training, small perturbations are fed to neurons along with input x. If θ is the parameter of a predictive model, adversarial cost function is calculated as in the following equation:

$$\log p(y \mid x + r_{adv};\ \theta) \text{ where } r_{adv} = \arg\min log\, p(y \mid x + r;\ \hat{\theta}\ ) \tag{3}$$

where, r is an adversarial example on the input data. And $\hat{\theta}$ denotes a set of constant parameters of a predictive model. At each training step, machine learning algorithms identify the worst case perturbations r_{adv} against the current trained model $p(y \mid x;\ \hat{\theta}\ )$ in above equation with respect to θ.

4.4 BiLSTM Layer

A BiLSTM layer containing forward and backward propagation is shown in Fig. 2, where two LSTM cells- forward and backward- are created. Due to the forward and backward LSTM cells, past and future information can be obtained for each memory cell.

4.5 Joint Predictive Layer

The final training objective of multi domain joint model is to minimize total cumulative loss, i.e., the sum of the domain, intent, entity, and adversarial loss:

$$\mathrm{L}\left(\Theta,\ \Theta^{d},\ \Theta^{i},\ \Theta^{t},\ \Theta^{da},\ \Theta^{ia},\ \Theta^{ta}\right)=\sum_{\alpha\in\{\Theta,\ \Theta^{d},\ \Theta^{i},\ \Theta^{t},\ \Theta^{da},\ \Theta^{ia},\ \Theta^{ta}\}} L^{\alpha}(\Theta,\ \Theta^{\alpha}) \tag{4}$$

The loss for domain, intent, entity, and adversarial learning for domain, intent, and entity $l^{d}, l^{i}, l^{t}, l^{da}, l^{ia}, l^{ta}$ is calculated separately for each annotated example. Then the gradient step is applied and shared loss is calculated as $l^{d}+l^{i}+l^{t}+l^{da}+l^{ia}+l^{ta}$. Using the shared loss parameters Θ, domain, intent, entity, and adversarial training are optimized jointly for all three predictive tasks using Adam optimizer. The detailed steps of a multi-domain dialog system are as follows:

Algorithm 1: Multi domain joint model

Step 1: Prepare input data containing sequence of input, domain intent, and slot in the respective order

Step 2: Construct word embedding, $e_{w}\in R^{64}$ *for each* $w\in W$

Step 3: Create forward and backward LSTM cells

Step 4: Create encoder & decoder for input, domain, intent, and slot

Step 5: Predict and calculate loss function

a. Calculate slot loss function using seq2seq loss

b. Calculate intent, and domain loss using cross entropy

Step 6: Calculate loss function

- $\mathrm{L}\left(\Theta,\ \Theta^{d},\ \Theta^{i},\ \Theta^{t}\right)=\sum_{\alpha\in\{\Theta,\ \Theta^{d},\ \Theta^{i},\ \Theta^{t}\}} L^{\alpha}(\Theta)$, loss for each fields
- Add random perturbation,

$\log p(y\mid x+r_{adv};\ \theta)$ where $r_{adv}=\arg\min log\, p(y\mid x+r;\ \hat{\theta}\ )$

- Calculate loss function by adding adversarial loss

$$\mathrm{L}\left(\Theta,\ \Theta^{d},\ \Theta^{i},\ \Theta^{t},\ \Theta^{da},\ \Theta^{ia},\ \Theta\mathrm{t}^{a}\right)=\sum_{\alpha\in\{\Theta,\ \Theta^{d},\ \Theta^{i},\ \Theta^{t},\ \Theta^{da},\ \Theta^{ia},\ \Theta^{ta}\}} L^{\alpha}(\Theta,\ \Theta^{\alpha})$$

Step 7: Optimize the model based on Adam optimizer

5 Experiment

This study used a publicly available annotated dataset (shown in Appendix A, Fig. A1) with 43k examples from three different domains- Alarm, Reminder, and Weather- of multi-domain personal assistants [23]. The dataset contains 11 unique entities and 12 intent labels as shown in Tab. 1, with some overlapping entities between domains. A sample of the processed dataset is shown in Appendix A, Fig. A2.

The dataset is converted into utterance sets of input, slot-tags, intent label, and domain label in a sequential order, as shown in Fig. A2. Input data is enclosed with BOS (Beginning of Sentence) and EOS (End of Sentence). The examples or utterances are then divided into train, dev, and test datasets at a 70 to 20 to10 ratio respectively, as shown in Tab. 2. Then the annotated datasets are preprocessed and tokenized using a widely-used NLU tokenization python library to create word embedding of size 64 that has an input with length 50. Finally, the word-embedding vector is created based on collected dataset and utilized in a TensorFlow-based LSTM python library for learning and predicting processes with the inline addition of perturbation along with input.

To evaluate the Multi-Domain Joint Model with adversarial learning (MDJM-ADV), we conducted a comparative experiment with a prior Multi-Domain Joint Model (MDJM) using the same dataset. Each

model is trained at the learning rate of 0.01 with 16 batch size and 100 hidden layers, and optimized with the Adam optimizer. The LSTM-based MDJM-ADV shares the loss function between the domain, intent, entities, and adversarial layers. Addition of perturbations along with input increases the loss function of the LSTM cell, and minimizing the loss function using the Adam optimizer equally shares information between domain, intent, and entity and also reduces the number of predictive models to one. We trained the MDJM and MDJM-ADV models for 20 epochs with 16 batch sizes at the learning rate of 0.001 using 100 hidden neurons. Figs. 5 and 6 show the training loss for MDJM and MDJM-ADV, respectively.

Table 1: Number of intent and entity in each domain

Domain	Number of Intent	Number of Entity
Alarm	6	2
Reminder	3	6
Weather	3	5
3 domains	**12**	**11(unique entities)**

Table 2: Train, dev and test dataset

Domain	Train (70%)	Dev (20%)	Test (10%)	Total (100%)
Alarm	9,282	2,621	1,309	**13,212**
Reminder	6,901	1,960	943	**9,804**
Weather	14,338	4,040	1,929	**20,307**
3 domain	**30,521**	**8,621**	**4,181**	**43,323**

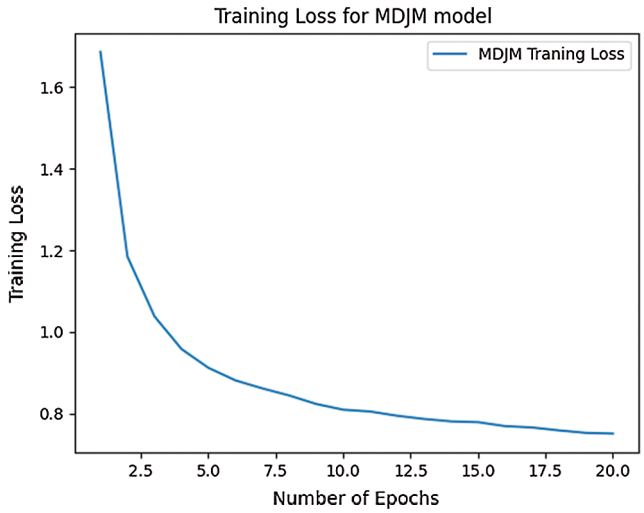

Figure 5: Training loss for MDJM model

Figure 6: Training loss for MDJM-ADV model

Tab. 3 presents the predictive accuracy of the MDJM and MDJM-ADV models for domain, intent, and entity. Our proposed model, MDJM-ADV, shows improvement in predictive accuracy for domain, intent, and entity in comparison to the base joint model, MDJM.

Table 3: Test accuracy for various models

Model Name	MDJM	MDJM-ADV
Domain	67.98%	68.03%
Intent	78.54%	78.68%
Entity	58.12%	58.01%
Average Acc.	68.21%	68.24%

6 Conclusion

In this paper, we proposed an adversarial joint model, MDJM-ADV, for dialog systems to predict domain, intent, and entity using a single LSTM cell to reduce the risk of downstream error propagation that is present in the typical pipelined approach. Experimental results showed a significant improvement in the predictive performance of each model—domain, intent, and entity—based on adversarial learning in comparison to the base joint model. The proposed model reduces the number of predictive models to one, which ultimately reduces the level of human effort necessary to manage multiple predictive models. Further research would benefit from testing our proposed MDJM-ADV model with various languages and datasets of large volume.

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).

Conflict 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, G. De Doncker *et al.,* "Intelligent personal assistants: A systematic literature review," *Expert Systems with Application*, vol. 147, no. 1, 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)*, 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 *Interspeech*, 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., 2016. Available at: https://dokumen.tips/documents/the-elements-of-aiml-style-2-the-elements-of-aimlstyle-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)*, pp. 706–710, 2020.

[10] 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)*, pp. 5754–5758, 2018.

[11] 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.

[12] 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.

[13] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li *et al.,* "Attention-based bidirectional long short-term memory networks for relation classification," *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 2, pp. 207–212, 2016. Available at: https://openreview.net/forum?id=BJW_r2eO-B.

[14] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," arXiv preprint arXiv: 1609.01454, 2016.

[15] 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.

[16] 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.

[17] 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," *Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2 (Short Papers), pp. 753–757, 2018.

[18] 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.

[19] S. Mehri, M. Eric and D. Hakkani-Tur, "Dialoglue: A natural language understanding benchmark for task-oriented dialogue," arXiv preprint arXiv: 2009.13570, 2020.

[20] 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.

[21] 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)*, pp. 6049–6053, 2018.

[22] 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.

[23] S. Schuster, S. Gupta, R. Shah and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog," arXiv preprint arXiv: 1810.13327, 2018.

Appendix A. Sample Dataset

weather/find	12:26:weather/noun,31:44:location	tell me the weather report for half moon bay
weather/find	12:26:weather/noun,31:44:location	tell me the weather report for half moon bay
weather/find	12:26:weather/noun,31:44:location	tell me the weather report for half moon bay
weather/find	12:26:weather/noun,31:44:location	tell me the weather report for half moon bay
weather/find	12:26:weather/noun,31:44:location	tell me the weather report for half moon bay
weather/find	12:18:datetime,19:27:weather/noun,32:45:location	give me the latest forecast for half moon bay
weather/find	12:18:datetime,19:27:weather/noun,32:45:location	give me the latest forecast for half moon bay
weather/find	12:18:datetime,19:27:weather/noun,32:45:location	give me the latest forecast for half moon bay
weather/find	12:18:datetime,19:27:weather/noun,32:45:location	give me the latest forecast for half moon bay
weather/find	12:18:datetime,19:27:weather/noun,32:45:location	give me the latest forecast for half moon bay
weather/find	8:14:datetime,17:25:weather/noun,30:43:location	give me friday's forecast for half moon bay
weather/find	8:14:datetime,17:25:weather/noun,30:43:location	give me friday's forecast for half moon bay
weather/find	8:14:datetime,17:25:weather/noun,30:43:location	give me friday's forecast for half moon bay
weather/find	8:14:datetime,17:25:weather/noun,30:43:location	give me friday's forecast for half moon bay
weather/find	8:14:datetime,17:25:weather/noun,30:43:location	give me friday's forecast for half moon bay
weather/find	19:25:datetime,26:34:weather/noun,39:52:datetime	i need to know the latest forecast for half moon bay
weather/find	19:25:datetime,26:34:weather/noun,39:52:datetime	i need to know the latest forecast for half moon bay
weather/find	19:25:datetime,26:34:weather/noun,39:52:datetime	i need to know the latest forecast for half moon bay
weather/find	12:20:datetime,23:37:weather/noun,42:55:location	Do you have thursday's weather report for half moon bay
weather/find	12:20:datetime,23:37:weather/noun,42:55:location	Do you have thursday's weather report for half moon bay
weather/find	12:20:datetime,23:37:weather/noun,42:55:location	Do you have thursday's weather report for half moon bay
weather/find	12:21:datetime,24:40:weather/noun,45:58:location	Do you have Wednesday's weather forecast for half moon bay
weather/find	12:21:datetime,24:40:weather/noun,45:58:location	Do you have Wednesday's weather forecast for half moon bay
weather/find	12:21:datetime,24:40:weather/noun,45:58:location	Do you have Wednesday's weather forecast for half moon bay
weather/find	12:21:datetime,24:40:weather/noun,45:58:location	Do you have Wednesday's weather forecast for half moon bay

Figure A1: Sample raw dataset

```
BOS tell me the weather report for half moon bay EOS    O O O B-weather-noun O B-location find weather
BOS tell me the weather report for half moon bay EOS    O O O B-weather-noun O B-location find weather
BOS tell me the weather report for half moon bay EOS    O O O B-weather-noun O B-location find weather
BOS tell me the weather report for half moon bay EOS    O O O B-weather-noun O B-location find weather
BOS tell me the weather report for half moon bay EOS    O O O B-weather-noun O B-location find weather
BOS give me the latest forecast for half moon bay EOS   O O O B-datetime B-weather-noun O B-location find weather
BOS give me the latest forecast for half moon bay EOS   O O O B-datetime B-weather-noun O B-location find weather
BOS give me the latest forecast for half moon bay EOS   O O O B-datetime B-weather-noun O B-location find weather
BOS give me the latest forecast for half moon bay EOS   O O O B-datetime B-weather-noun O B-location find weather
BOS give me the latest forecast for half moon bay EOS   O O O B-datetime B-weather-noun O B-location find weather
BOS give me friday's forecast for half moon bay EOS O O B-datetime B-weather-noun O B-location find weather
BOS give me friday's forecast for half moon bay EOS O O B-datetime B-weather-noun O B-location find weather
BOS give me friday's forecast for half moon bay EOS O O B-datetime B-weather-noun O B-location find weather
BOS give me friday's forecast for half moon bay EOS O O B-datetime B-weather-noun O B-location find weather
BOS give me friday's forecast for half moon bay EOS O O B-datetime B-weather-noun O B-location find weather
BOS i need to know the latest forecast for half moon bay EOS    O O O O O B-datetime B-weather-noun O B-datetime find weather
BOS i need to know the latest forecast for half moon bay EOS    O O O O O B-datetime B-weather-noun O B-datetime find weather
BOS i need to know the latest forecast for half moon bay EOS    O O O O O B-datetime B-weather-noun O B-datetime find weather
BOS Do you have thursday's weather report for half moon bay EOS O O O B-datetime B-weather-noun O B-location find weather
BOS Do you have thursday's weather report for half moon bay EOS O O O B-datetime B-weather-noun O B-location find weather
BOS Do you have thursday's weather report for half moon bay EOS O O O B-datetime B-weather-noun O B-location find weather
BOS Do you have Wednesday's weather forecast for half moon bay EOS  O O O B-datetime B-weather-noun O B-location find weather
BOS Do you have Wednesday's weather forecast for half moon bay EOS  O O O B-datetime B-weather-noun O B-location find weather
BOS Do you have Wednesday's weather forecast for half moon bay EOS  O O O B-datetime B-weather-noun O B-location find weather
BOS Do you have Wednesday's weather forecast for half moon bay EOS  O O O B-datetime B-weather-noun O B-location find weather
BOS Do you have Wednesday's weather forecast for half moon bay EOS  O O O B-datetime B-weather-noun O B-location find weather
BOS Give me most recent forecast for half moon bay EOS  O O B-datetime B-weather-noun O B-location find weather
BOS Give me most recent forecast for half moon bay EOS  O O B-datetime B-weather-noun O B-location find weather
BOS Give me most recent forecast for half moon bay EOS  O O B-datetime B-weather-noun O B-location find weather
BOS Give me most recent forecast for half moon bay EOS  O O B-datetime B-weather-noun O B-location find weather
BOS Give me most recent forecast for half moon bay EOS  O O B-datetime B-weather-noun O B-location find weather
BOS give me Thursday's half moon bay weather report EOS O O B-datetime B-location B-weather-noun find weather
BOS give me Thursday's half moon bay weather report EOS O O B-datetime B-location B-weather-noun find weather
```

Figure A2: Sample preprocessed sequential dataset