Cognitive Computation
https://doi.org/10.1007/s12559-020-09718-4

A Deep Multi-task Model for Dialogue Act Classification, Intent Detection and Slot Filling

Mauajama Firdaus[1] · Hitesh Golchha[1] · Asif Ekbal[1] · Pushpak Bhattacharyya[1]

Received: 20 November 2018 / Accepted: 13 February 2020

Abstract

An essential component of any dialogue system is understanding the language which is known as spoken language understanding (SLU). Dialogue act classification (DAC), intent detection (ID) and slot filling (SF) are significant aspects of every dialogue system. In this paper, we propose a deep learning-based multi-task model that can perform DAC, ID and SF tasks together. We use a deep bi-directional recurrent neural network (RNN) with long short-term memory (LSTM) and gated recurrent unit (GRU) as the frameworks in our multi-task model. We use attention on the LSTM/GRU output for DAC and ID. The attention outputs are fed to individual task-specific dense layers for DAC and ID. The output of LSTM/GRU is fed to softmax layer for slot filling as well. Experiments on three datasets, i.e. ATIS, TRAINS and FRAMES, show that our proposed multi-task model performs better than the individual models as well as all the pipeline models. The experimental results prove that our attention-based multi-task model outperforms the state-of-the-art approaches for the SLU tasks. For DAC, in relation to the individual model, we achieve an improvement of more than 2% for all the datasets. Similarly, for ID, we get an improvement of 1% on the ATIS dataset, while for TRAINS and FRAMES dataset, there is a significant improvement of more than 3% compared to individual models. We also get a 0.8% enhancement for ATIS and a 4% enhancement for TRAINS and FRAMES dataset for SF with respect to individual models. Results obtained clearly show that our approach is better than existing methods. The validation of the obtained results is also demonstrated using statistical significance *t* tests.

Keywords Multi-tasking · Dialogue act classification · Intent detection · Slot filling

Introduction

In the area of dialogue systems, spoken language understanding (SLU) is a critical step towards understanding the utterance of the user. To create robust human/machine dialogue systems or chatbots, it is essential to understand the user and respond according to the user's request. To satisfy the user, it is vital to have an SLU module in every human/machine dialogue systems that help in understanding the intentions and extracting necessary information from the user utterance. Spoken language understanding mainly deals with assigning a functional tag to the user input. The functional tag expresses the communicative intentions behind every user utterance also known as utterance's dialogue act. The first step in dialogue processing is to identify the dialogue acts of the user utterance, and is known as dialogue act classification (DAC). The classification of the dialogue acts in a user utterance can assist an automated system in producing an appropriate response to the user. Dialogue acts (DA) can be said to understand the intention of the user. An example of DAC is given in Table 1. The correct classification of dialogue acts will help the system in resolving the queries of the user. For every dialogue system, it is essential to understand the intentions of the user. Many works have been done to understand the different aspects of the user and their feelings as in [23, 36, 64] to create a system that can help in increasing the interaction between the human and machine. Also, several works are being done for properly

✉ Mauajama Firdaus
mauajama.pcs16@iitp.ac.in

Hitesh Golchha
hitesh@iitp.ac.in

Asif Ekbal
asif@iitp.ac.in

Pushpak Bhattacharyya
pb@iitp.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology, Patna, Bihar, India

Published online: 23 March 2020

Table 1 An example of DAC, ID and SF

Sentence	*When*	*is*	*the*	*flight*	*from*	*Chicago*	*to*	*Dallas*
Slots	O	O	O	O	O	B-fromcity_ name	O	B-tocity_ name
Intent	Flight_time							
Dialogue act	Question							

replying to the user queries as in [58, 66, 74] to complete the different modules of a dialogue system that can understand the user and appropriately respond to them.

For dialogue systems, especially for goal-oriented systems, the second step in dialogue processing is to identify the intent of the user, i.e. the primary goal of the user. The global properties of an utterance are the intent that signifies the primary goal of the user. Intent detection (ID) is a critical processing step of semantic analysis in dialogue systems. While it is a standard utterance classification task and distinctly less complex than the other tasks of semantic analysis, the errors made by a classifier for intent detection are more visible—as they often lead to wrong system responses. Therefore, a robust intent detection system plays a crucial role in building an effective dialogue system. An example of intent detection is given in Table 1. Intents are mainly domain-dependent. Hence, for different goal-oriented dialogue systems, we have a unique set of intents.

The final step in spoken language understanding is to extract the necessary information in the form of slots automatically. The task is to fill in a set of arguments or 'slots' embedded in a semantic frame to accomplish a goal in human-machine dialogue systems. We show the example of slots in Table 1. This task of finding a suitable label for every word in the utterance is referred to as slot filling.

As already discussed, the primary tasks of goal-oriented dialogue systems are the dialogue act classification (DAC), intent detection (ID) and slot filling (SF) that capture the semantic information of the user utterances. According to the information extracted, the system can then decide on the appropriate actions to be taken, to help the users achieve their demands. SLU applications are becoming increasingly significant in our everyday lives. Numerous devices, such as smartphones, have personal assistants that are built with SLU technologies.

Problem Definition

In this paper, we solve three very important problems of SLU, viz. dialogue act classification, intent detection and slot filling. Dialogue act classification has been treated as an utterance classification problem. It aims to classify a given user utterance x, consisting of words in a sequence $x = (x_1, x_2, \ldots, x_T)$ into one of the D pre-defined set of dialogue acts, y_d, based upon the contents of the sentence such that:

$$y_d = \underset{d \in D}{\operatorname{argmax}} P(y_d/x) \tag{1}$$

Intent detection is basically treated as a semantic utterance classification problem. It aims to classify a given user utterance x, consisting of words in a sequence $x = (x_1, x_2, \ldots, x_T)$ into one of the N pre-defined set of intent classes, y_i, based upon the meaning of the sentence such that:

$$y_i = \underset{i \in N}{\operatorname{argmax}} P(y_i/x) \tag{2}$$

Slot filling refers to the extraction of semantic constituents from an input text, and to fill in the values for a pre-defined set of slots in a semantic frame. The slot filling task is considered as assigning semantic labels to every word in the utterance. Given a sentence x comprising of a sequence of words $x = (x_1, x_2, \ldots, x_T)$, the objective of a slot filling task is to find a sequence of semantic labels $s = (s_1, s_2, \ldots, s_T)$, for every word in the sentence, such that:

$$\hat{s} = \underset{s}{\operatorname{argmax}} P(s/x) \tag{3}$$

Motivation and Contributions

In the literature, there exists a significant number of works related to dialogue act classification, intent detection and slot filling, but there still is room for progress, especially with regard to making these models as task- and domain-invariant as possible. The problem is more challenging when the system has to deal with more realistic, natural utterances expressed in natural language, by several speakers. Irrespective of the approach being adopted, the biggest problem is the 'naturalness' of the spoken language input. In most of the existing works, dialogue act classification, intent detection and slot filling have been carried out in isolation.

In this paper, we propose a multi-task model for dialogue act classification (DAC), intent detection (ID) and slot filling (SF). Information of one task can provide useful evidence for the other, and sharing of this information might be helpful to improve the quality of the task. Our multi-task model makes use of this shared representation, and solve all the three problems concurrently. Another motivation for employing a multi-task model is that the essential elements of SLU, i.e. DAC, ID and SF can be predicted at once providing an end-to-end

neural network system. Experiments on the benchmark datasets show that our proposed model performs superior compared to the individual models when these three tasks (DAC, ID and SF) are handled in isolation, i.e. in a single-task framework.

The major contributions of this work are:

- We propose a multi-task model for dialogue act classification, intent detection and slot filling by employing different RNN architectures such as LSTM and GRU.
- We create a benchmark corpus for the SLU tasks, i.e. DAC, intent detection and slot filling on TRAINS and FRAMES datasets for capturing more realistic and natural utterances spoken by the speakers in a human/machine dialogue system.

The remainder of this paper is organized as follows: In the "Related Work" section, we present a very brief survey about the related works. We describe our proposed approach in the "Proposed Approach" section. Experimental setup and the datasets are reported in the "Dataset and Experiment" section. Results and its analysis are discussed in the "Results and Error Analysis" section. Finally, the concluding remarks and directions for future research are presented in the "Conclusion and Future Work" section.

Related Work

As a significant component in spoken dialogue systems, spoken language understanding system captures the semantic meanings transmitted by speech signals. The primary units in SLU systems mainly deal with DAC, intent detection and slot filling. In the past, these tasks have mostly been performed in isolation.

Dialogue Act Classification

In the past, identification of dialogue acts (DAs) has been carried out by framing the problem either as classification or as a sequence labelling task. Different machine learning-based approaches such as support vector machines (SVMs) [40, 53], hidden Markov models (HMM) [54, 57, 61], maximum entropy models (MEMM) [1], Bayesian networks [12, 21, 25, 26], naive Bayes [4, 55] and conditional random fields (CRF) [29, 33] have been used for the recognition of dialogue acts. In [6], the authors used prosodic cues for automatically classifying dialogue acts with the help of SVM on a Spanish CallHome database. Multi-class dialogue act classification with several binary classifiers combined through error correction output codes using SVM on ICSI meeting corpus was explored in [40]. The influence of contextual information on dialogue act classification with the help of SVM was explored in [53] on the Switchboard corpus. In [61], HMM-based dialogue act taggers were investigated which were trained on unlabelled data that helped in reducing the tagging errors on the SPINE dialogue corpus. The authors in [54] explored HMM and neural network-based methods for speech act detection on the Spanish CallHome dataset. Automatic segmentation and classification of dialogue act from the ICSI meeting corpus with the help of decision trees and maximum entropy classifier was explored in [1]. A complete analysis of conditional and generative dynamic Bayesian networks on the ICSI meeting corpus was explored in [21] for dialogue act detection. In [33], syntactic features were used for classifying Czech dialogue acts using CRF. The authors in [29] used CRF that helped in learning sequential dependencies for dialogue act classification. Prosodic features and gestures also help in understanding the communicative intentions of the user as in [6, 62].

Due to the effectiveness of deep learning, it has been adopted for many language processing tasks, including dialogue act classification. Recurrent neural network (RNN) has been extensively employed for the classification of DAs [22, 27, 39, 47]. The authors in [27] used stacked LSTMs for dialogue act classification on the Switchboard and MRDA corpus. Contextual language model-based RNN to tract the interactions between different speakers in a dialogue was designed in [39] for the Switchboard corpus. A latent variable RNN for modelling the words and sentences together was proposed in [22]. RNNs, along with convolutional neural networks (CNN), has also been employed in the past [24, 41]. For recognizing the DAs, deep neural networks with CRF have also been used [34, 75]. These approaches have utilized various lexical, syntactic and prosodic cues as features for modelling the DAs. The authors in [34] used hierarchical RNN along with CRF for classifying the utterances into its corresponding dialogue acts.

Intent Detection

Historically, SLU research has come into view from the call classification systems [11] and the ATIS project [49]. For intent detection, machine learning-based traditional approaches such as support vector machine (SVM) [14] and Adaboost [59, 60] have been employed for detecting the intents of a user utterance. Authors in [15] presented an approach for intent classification by considering the heterogeneous features comprising of user utterances. For detecting the intents, the authors in [28] enriched the word embeddings to make the performance of the model better. A promising direction towards solving these problems is deep learning, which combines both classification and feature design into the learning process. For efficient learning under low-resource SLU tasks, the authors in [42] have proposed a multi-scale RNN structure. Several deep learning techniques have been successively utilized for

intent detection such as [17]. The method proposed here makes use of CNN. Recurrent neural networks (RNNs) and long short-term memory (LSTM) [19] have also been previously explored for intent detection [50, 51]. The authors in [51] used RNN along with word hashing to take care of the out-of-vocabulary (OOV) words present in the corpus. A comparative study of different neural network architectures considering only lexical information of the utterance as a feature has been investigated in [50]. An ensemble-based deep learning architecture was employed in [7] for intent detection on the ATIS dataset.

Slot Filling

For sequence labelling, factorized probabilistic models such as maximum entropy Markov model (MEMM) [43] and conditional random field (CRF) [52] have been used that directly capture the global distribution. Syntactic features via syntactic tree kernels with SVM were employed in [46] for slot filling. For sentence simplification, a dependency parsing-based approach was proposed in [60] for completing the SLU tasks. For slot filling, various deep learning-based methods such as deep belief network (DBN) [5] and RNNs [44, 45, 69] have been proposed due to their keen abilities to capture dependencies, and it has proved to outperform the traditional models, such as CRF. The authors in [71] used transition features to improve RNNs and the sequence level criteria for optimization of CRF to capture the dependencies of the output label explicitly. The authors in [70] used deep LSTMs along with regression models to obtain the output label dependency for slot filling. In [76], a focus mechanism for an encoder-decoder framework was proposed for slot filling on the ATIS dataset. The authors in [73] introduced a generative network based on the sequence to sequence model along with pointer network for slot filling.

Joint Tasks

Lately, intent detection has been jointly done with slot filling using deep learning techniques. Various RNN models using LSTM or GRU as its basic cell have been employed [16, 37, 38, 72] for detecting the intents and slots together. Different deep learning architectures have been employed for intent detection and slot filling together using CNN [67] and recursive neural networks [13]. The authors in [20] employed a triangular CRF that used an additional random variable for detecting the intents on top of the standard CRF. Also, CNN-based triangular CRF model for joint intent detection and slot filling was proposed in [67] where the features were extracted by the CNN layers and were shared by both the tasks. Hierarchical representations within the input text learned using a recursive neural network (RecNN) were proposed for the joint task [13] of intent detection and slot filling. In [38], the intent variation was modelled continuously along with the arrival of new words to achieve better performance for the joint task using LSTM. [72] used bi-directional GRUs to learn the representations of the sequence shared by the intent and slot filling tasks. Recently, attention-based bi-directional RNNs were also proposed for jointly addressing the task of intent detection and slot filling [37]. A bi-model-based RNN semantic frame parsing network structure was employed for intent detection and slot filling in [63]. The authors in [10] used a slotted gate that focused on learning the relationship between intent and slot vectors for joint modelling of the tasks on the ATIS and SNIPs dataset. In [16], the authors investigated the alternative architectures for modelling lexical context for SLU and presented a joint approach using single bi-directional RNNs with LSTM cells for a domain, intent and slot filling. In [31], the authors used character embeddings and word embeddings as input to LSTM for domain, intent and slot filling. Sequential dialogue context modelling using RNN for SLU was investigated in [2]. The authors in [3] employed a deep learning architecture for jointly performing dialogue act classification and slot filling in DSTC2 corpus. In our previous work [8], we have proposed an ensemble method for jointly identifying the intents and slots in a given utterance. In another work reported in [9], a hierarchical approach was employed to capture the contextual information for identifying the intents and slots simultaneously in a given utterance.

In our present work, we propose a multi-task approach for performing dialogue act classification, intent detection and slot filling tasks using attention-based deep learning architecture. To the best of our knowledge, this is the very first attempt employing an in-depth learning approach using combined word embedding representation for solving these three tasks concurrently.

Methodology

The overall block diagram of our proposed architecture is depicted in Fig. 1. Our model is a multi-task deep learning-based architecture that performs three tasks, namely intent identification, slot filling and dialogue act classification. These three tasks share the underlying representations through common layers but have their task-specific classifying layers.

Proposed Approach

The three tasks share the underlying representations through common layers but have their task-specific classifying layers. Each word representation is a concatenation of two components: a vector representation from word embeddings and another one from a single layer CharCNN over the character embeddings of the word followed by a highway layer. For sequentially encoding information, the obtained word

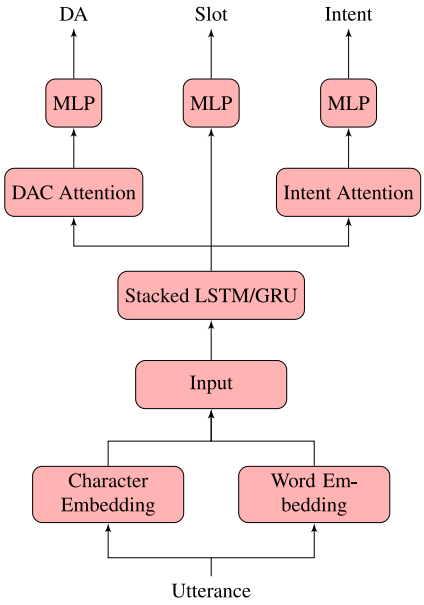


Fig. 1 Block diagram of our proposed approach

representations are operated with multiple stacked LSTM layers having residual connections between the consecutive layers.

Slot filling applies dense layers and softmax over the hidden representation of each time step to obtain the predictions. Intent detection and dialogue act classification, however, first apply attention to get representation and then apply dense and softmax layers over them to obtain predictions. The detailed architecture of our proposed multi-task model is given in Fig. 2.

Word Representation

The NLU component receives each utterance as a sequence of words $w = (w_1, w_2, \ldots, w_T)$. The representation of the i^{th} word, x_i, is obtained as the concatenation of two vectors: the word embedding (x_i^w) and the output from a single layer CharCNN network over the character embeddings (x_i^c).

Word Embedding Several algorithms exist for learning distributed representations for words in a given corpus. The word vectors pre-trained with these learning objectives often display semantic and distributional informativeness. Words which occur in similar contexts and are similar in meaning are closer to each other in these embedding spaces. Such embeddings are useful as basic representations in diverse applications of natural language processing (NLP). For word embeddings, we use three pre-trained embedding models: GloVe[1] [48], Word2Vec[2] and Fasttext.[3] We use these pre-trained word vectors to obtain (x_i^w)—one of the components of our word representations, and the choice of the algorithm for pre-training is a hyperparameter. Slot filling is a sequence labelling problem where each word provides the context for the next word. Hence, proper representation of words for this task is essential. Previously, there have been quite a few works for proper representation of words for sequence labelling tasks as in [35].

CharCNN and Single Layer Highway Network over Character Embeddings We derive word representations similar to [30] to utilize the semantic and morphological features that can be extracted from character-level representations.

Let the k^{th} word w_k be represented as a padded sequence of characters $[c_1, c_2, \ldots, c_l]$, with l being the maximum character length, the vocabulary of characters be C, embedding dimension of the characters be td, and $Q \in R^{d \times |C|}$ be the embedding matrix for the characters. Using Q, we obtain the character matrix $C_k \in R^{l \times d}$, where the j^{th} row corresponds to the character embedding for c_j.

We then apply the convolution operation over C_k using multiple filters of varying sizes. For j^{th} filter F_j, the output of the convolution operation is obtained by applying the filter F repeatedly with unit strides on sub-matrices of C_k:

$$out_k[i,j] = tanh\left(F_j \cdot C_k[i : i+m-1] + b_j\right) \tag{4}$$

Here, we represent the sub-matrix of C_k from i^{th} row to $(i+m-1)^{th}$ row as $C_k[i : i+m-1]$ where $i = 1, 2, \ldots, n-m+1$. m is the size of the filter and b_j is the bias term. Finally, we take max over time:

$$y_k[j] = \max_i out_k[i,j] \tag{5}$$

The weight sharing in CNN helps filters to search for n-gram features over space, and each of the filters learns to search for its feature. The global max pooling helps in identifying the presence of the n-gram feature invariant to the position.

The max-pooled convolutional layer is followed by a single layer highway network, represented by the eq:

$$\boldsymbol{x_k^c} = \boldsymbol{t} \odot g(\boldsymbol{W}_H \boldsymbol{y}_k + \boldsymbol{b}_H) + (\boldsymbol{1} - \boldsymbol{t}) \odot \boldsymbol{y}_k \tag{6}$$

$$\boldsymbol{t} = \sigma(\boldsymbol{W}_T \boldsymbol{y}_k + \boldsymbol{b}_T) \tag{7}$$

Here, the activation function is g; t and (1-t) are the transform gate and the carry gate, respectively. W_h, W_T, b_H and b_T are the parameters of the highway layer. Highway layers essentially help develop deep layers by separately controlling the expression of inputs and transformations to the output for each dimension.

Final Word Representation Thus, the i^{th} word w_i is represented as x_i which is a concatenation of x_i^w and x_i^c.

[1] http://nlp.stanford.edu/projects/glove/
[2] https://code.google.com/archive/p/word2vec/
[3] https://fasttext.cc/

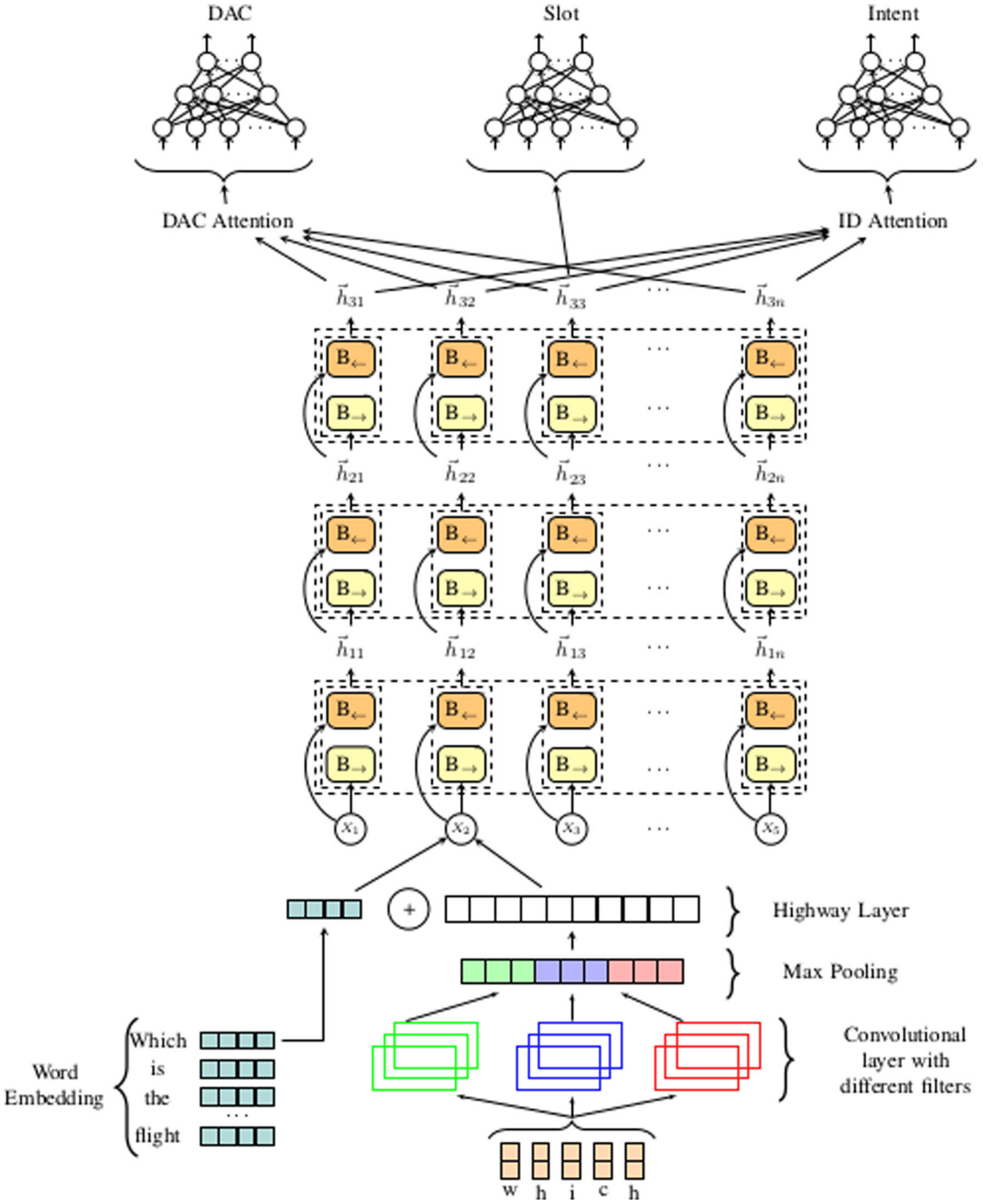
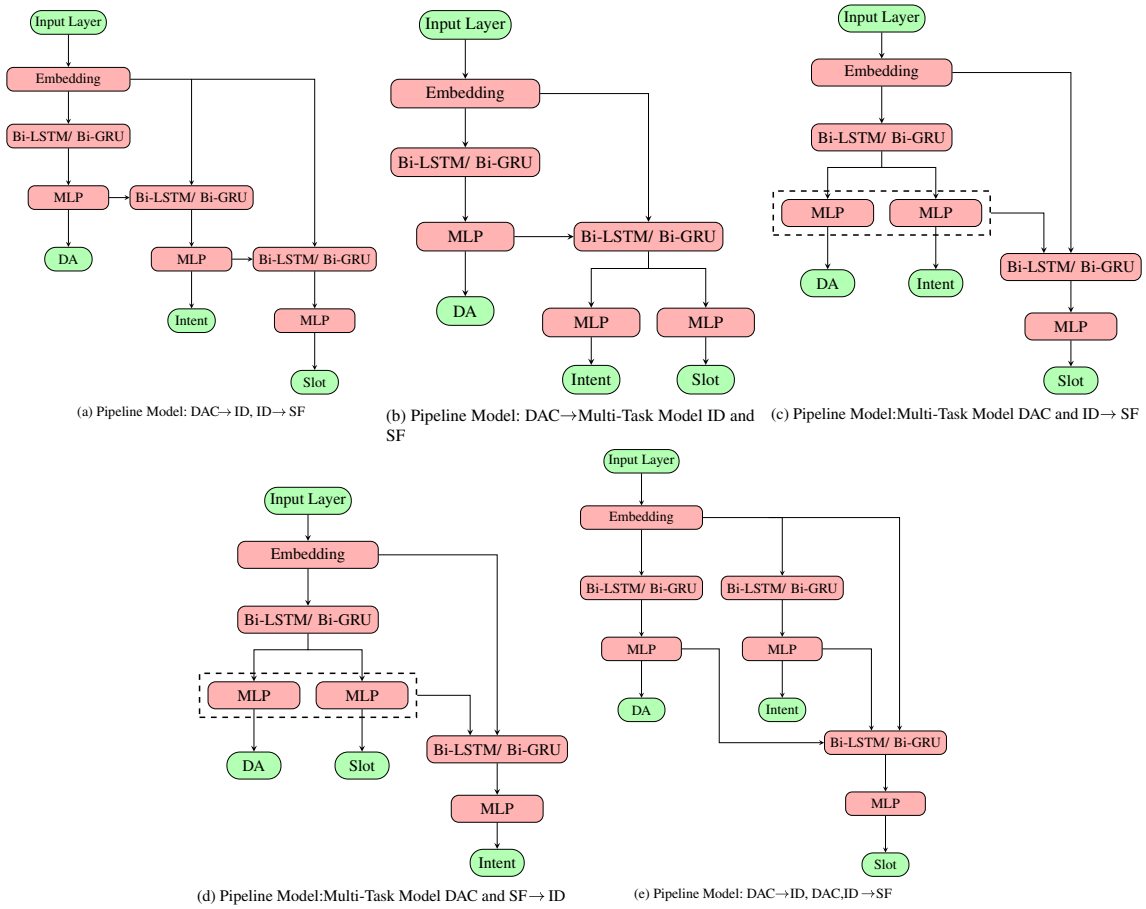


Fig. 2 Overall architecture of the proposed multi-task model

Sentence Representation

We use stacked bi-directional RNN layers with LSTM and GRU as its basic cell unit to process the sequence $x = (x_1, x_2, \ldots, x_T)$, as they are designed to process the sequential input. The number of layers L is a hyperparameter. This layer grounds each token representation with contextual information from both the directions in the input sequence, thus making it easier for downstream classification layers which make use of such information.

Given any inputs $u_1, u_2, \ldots, u_T$, a bi-directional LSTM/GRU layer computes a set of T vectors $h_1, h_2, \ldots, h_T$. The h_t is the concatenation of a forward LSTM/GRU hidden state $\overrightarrow{h_t}$ which reads the sentence in the forward direction, and a backward LSTM/GRU hidden state $h_t^{\leftarrow}$ that reads the sentences in the reverse direction.



(a) Pipeline Model: DAC→ ID, ID→ SF

(b) Pipeline Model: DAC→Multi-Task Model ID and SF

(c) Pipeline Model:Multi-Task Model DAC and ID→ SF

(d) Pipeline Model:Multi-Task Model DAC and SF→ ID

(e) Pipeline Model: DAC→ID, DAC,ID →SF

Fig. 3 Block diagram of the various models for DAC, ID and SF

$$\overrightarrow{h_t} = \overrightarrow{RNN_t}(u_1, u_2, \ldots, u_T) \tag{8}$$

$$h_t^{\leftarrow} = RNN_t^{\leftarrow}(u_1, u_2, \ldots, u_T) \tag{9}$$

$$h_t = \left[\overrightarrow{h_t}, h_t^{\leftarrow}\right] \tag{10}$$

Now, the output of the Bi-LSTM/Bi-GRU layer is added to the input from previous LSTM/GRU layer employing residual connections (if it is not the first Bi-LSTM/Bi-GRU layer), to enhance the flow of gradients during backpropagation, i.e. the input to layer $l+1$,

$$[u_1, u_2, \ldots, u_T]^{l+1} = \begin{cases} [u_1, u_2, \ldots, u_T]^l + \\ [h_1, h_2, .., h_T]^l \\ [h_1, h_2, .., h_T]^l \end{cases} \begin{array}{l} \text{if } l! = 1 \\ \text{if } l == 1 \end{array} \tag{11}$$

Intent and Dialogue Act Classification

For both intent and dialogue act classification, we use identical architectures: an attention layer over the task-shared sentence representation followed by dense layers and softmax. However, the parameters are task-specific and are not shared.

Firstly, a self-attention layer [68] is applied over the output of the stacked LSTMs/GRUs to obtain an importance weighted mean of the hidden states of all the time steps u. The salient contexts required to identify the class type are aggregated by the attention mechanism to build the output vector. Often the expressions indicative of the class appear in the short spans of text within the sentence, and the attention expertly attends to those specific LSTM/GRU encoded contexts.

$$\begin{aligned} \widetilde{h}_t &= \tanh(W_a h_t + b_a) \\ \alpha_t &= \frac{e^{\widetilde{h}_t^T * u_w}}{\sum_{t=1}^{T} e^{\widetilde{h}_i^T * u_w}} \\ u &= \sum_{t=1}^{T} \alpha_t * \widetilde{h}_t \end{aligned} \tag{12}$$

Where h_t denotes the LSTM/GRU's hidden state for time t, u_w is the randomly initialized query trained through backpropagation, α_t models the saliency of the t^{th} state normalized by softmax.

Table 2 Datasets with their representation of dialogue acts, intents and slots used in the experiments

Data set	# train	# test	# dialogue act	# intents	# slots
ATIS	4978	893	3	17	127
TRAINS	5355	1336	5	12	32
FRAMES	20,006	6598	10	24	136

The dense layers help to non-linearly combine the obtained features from the attention layer. The softmax layer applies an affine transformation to reduce the dimension to the number of output classes, and normalize the scores to obtain a probability distribution over the possible classes.

$$P\left(\hat{y} = i|x,\theta\right) = softmax_i(pTw_i + z_i) = \frac{e^{p^T w_i + z_i}}{\sum_{s=1}^{S} e^{p^T w_s + z_s}} \quad (13)$$

where z_s and w_s are the bias and weight vector of the s^{th} labels, p is the output from the dense layers, and S is the number of total classes. The system predicts the most probable class.

Slot Filling

For slot filling, the hidden unit for each time step coming from the output of the shared sentence representation is transformed by a dense layer and then by a softmax layer. The dense layer helps to combine the hidden features for that time step non-linearly. The softmax layer first projects the output from the dense layer to the number of possible slot classes and then transforms the scores for each class into a probability distribution.

Table 3 Hyperparameter tuning: 1st column lists the different parameters, 2nd column lists the values tried, 3rd column lists the final value chosen for each parameter

Parameter	Range	Final
Word embedding	Glove/Word2vec/Fasttext	Fasttext
Word embedding size	100/200/300	250D
Dropout	0–0.5	0.15
Bi-directional	True/false	True
Learning rate	0.5–3	1.0
Residual	Yes/no	Yes
Stacked LSTM layer	1–5	3
Hidden size	50–300	200

Objective Function

The objective is to minimize the sum of the cross-entropy losses of the three tasks for the entire training dataset.

Pipeline Models

There are other ways in which we can use the knowledge of DA or intent for the slot filling tasks and vice versa. To compare with our multi-task model, we implement various models which operate in a pipelined way. We develop five pipelined models, as shown in Fig. 3.

Model-1

In Fig. 3a, we use the information of dialogue acts for the detection of intents, whereas we use the knowledge of intents for slot filling. In this model, we have three different stacked Bi-LSTMs/Bi-GRUs for each task. The first stacked Bi-LSTM/Bi-GRU is used for DAC, and the output firstly applies attention, which gives a representation which is then fed to a multi-layer perceptron (MLP) for classifying the DAs. The second stacked Bi-LSTM/Bi-GRU is employed for intent detection, which is implemented similarly as dialogue act classification. The inputs are embeddings, and the output of the MLP corresponds to DAC. The output is fed to an MLP for the detection of intents. Finally, the third stacked Bi-LSTM/Bi-GRU is utilized for slot filling with embeddings and output of intent as inputs to MLP. The output is again fed to an MLP classifier for extracting the slots.

Model-2

The second model is shown in Fig. 3b. Here, we implement a different type of pipelined structure where the information of dialogue act is used in a multi-task model (MTM) that performs intent detection and slot filling simultaneously. Here, we employ two stacked Bi-LSTM/Bi-GRU. The first Bi-LSTM/Bi-GRU model performs DAC. While the second Bi-LSTM/Bi-GRU model is used for identifying the intents as well as for extracting the slots. The outputs of second Bi-LSTM/Bi-GRU model are fed to two different MLPs for intent detection and slot filling. For intent detection, before the output is supplied to MLP, attention is applied, and the representation obtained is used as input for the MLP for identifying the intents.

Model-3

The third pipelined model is constructed as shown in Fig. 3c. In this model, we implement a multi-task model (MTM) for DAC and ID using a stacked Bi-LSTM/Bi-GRU. On the output, attention is applied separately for both DAC and intent. The obtained representation from both attention layers is fed

Table 4 Results of different proposed multi-task models with character embeddings

Model	ATIS			TRAINS			FRAMES		
	DA (accuracy)	Intent (accuracy)	Slot (F1 score)	DA (accuracy)	Intent (accuracy)	Slot (F1 score)	DA (accuracy)	Intent (accuracy)	Slot (F1 score)
Bi-GRU	92.23	92.67	89.16	72.85	70.89	85.14	55.69	45.66	61.74
Bi-LSTM	93.82	93.56	90.63	72.54	68.95	86.33	55.82	45.02	60.25
Bi-GRU with attention	93.54	93.11	91.27	74.25	73.11	88.19	57.36	47.96	64.87
Bi-LSTM with attention	94.37	93.52	92.77	74.66	73.05	87.26	57.88	48.65	69.71

to two MLPs for DAC and ID, respectively. The information of this model, along with the embeddings, is supplied to a Bi-LSTM/Bi-GRU whose output is given to an MLP for slot filling.

Model-4

The fourth pipelined model is constructed as shown in Fig. 3d. In this model, we implement a multi-task model (MTM) for DAC and SF using a stacked Bi-LSTM/Bi-GRU, and the output is fed to two MLPs for DAC and SF, respectively. For DAC, before the output is fed to MLP, attention is applied, and the representation obtained is used as input for the MLP for identifying the dialogue acts. The information of this model, along with the embeddings, is fed to a Bi-LSTM/Bi-GRU whose output is given to an MLP for intent detection.

Model-5

The last model is constructed as shown in Fig. 3e. This pipelined model is implemented in a similar way to the first model with a slight difference. Here, the information of DA along with intent is subjected as input to the slot filling model.

Dataset and Experiment

Datasets

We evaluate our proposed multi-task model on two benchmark datasets. The first dataset is the well-known ATIS corpus which has been manually annotated for DAC. The other dataset is TRAINS, consisting of dialogue conversations, and we manually annotated this corpus with dialogue acts, intents and slots. The utterance, dialogue act, intent and slot distribution for both the datasets are given in Table 2.

ATIS Dataset A significant by-product of DARPA (Defence Advanced Research Program Agency) project was the ATIS (Airline Travel Information System) corpus. The ATIS corpus [49] is one of the most extensively used

Table 5 Results of different word embeddings

Model	Embeddings	ATIS			TRAINS			FRAMES		
		DA (accuracy)	Intent (accuracy)	Slot (F1 score)	DA (accuracy)	Intent (accuracy)	Slot (F1 score)	DA (accuracy)	Intent (accuracy)	Slot (F1 score)
Bi-GRU	Word2Vec	94.23	93.81	91.77	78.23	75.36	90.81	62.54	51.47	81.66
	Glove	95.72	94.68	92.03	78.99	76.81	91.23	63.81	52.69	82.84
	Fasttext	*96.89*	*96.11*	*93.67*	*80.55*	*79.82*	*94.75*	*65.41*	*54.05*	*83.15*
Bi-LSTM	Word2Vec	94.55	94.17	92.09	78.04	74.82	90.23	62.99	51.78	81.93
	Glove	95.88	94.93	93.01	79.11	77.35	92.56	64.02	52.98	83.26
	Fasttext	*96.98*	*96.83*	*93.91*	*80.14*	*79.37*	*94.20*	*66.06*	*55.21*	*84.78*
Bi-GRU with attention	Word2Vec	95.62	94.68	92.45	79.36	76.91	92.33	65.71	55.96	85.69
	Glove	96.10	95.36	93.71	80.47	80.23	93.54	66.23	56.71	86.52
	Fasttext	*97.63*	*97.21*	*94.32*	*82.69*	*83.05*	*96.32*	*68.33*	*58.06*	*89.37*
Bi-LSTM with attention	Word2Vec	95.91	95.06	92.98	78.86	76.26	91.87	66.32	56.23	86.15
	Glove	96.51	95.78	94.21	79.52	75.66	91.45	66.82	56.78	86.93
	Fasttext	*97.81*	*97.54*	*94.85*	*82.24*	*82.57*	*95.69*	*68.73*	*59.24*	*89.83*

The results in italics indicate the highest values

Table 6 Results of multi-task models with different deep learning layers

Model	# layers	ATIS			TRAINS			FRAMES		
		DA (accuracy)	Intent (accuracy)	Slot (F1 score)	DA (accuracy)	Intent (accuracy)	Slot (F1 score)	DA (accuracy)	Intent (accuracy)	Slot (F1 score)
Bi-GRU	1	90.16	92.86	90.78	75.76	76.84	92.50	63.10	51.32	81.43
	2	92.71	93.09	91.89	76.38	79.44	94.44	64.41	52.96	82.21
	3	*97.15*	*96.89*	*94.11*	*81.67*	*81.75*	*96.18*	*66.49*	*55.82*	*85.33*
Bi-LSTM	1	93.94	92.61	92.63	75.93	77.08	92.96	63.91	54.97	83.48
	2	94.13	94.33	93.01	77.58	80.35	94.86	64.74	55.79	84.12
	3	*97.41*	*97.53*	*94.48*	*82.33*	*82.11*	*96.45*	*67.47*	*56.37*	*85.50*
Bi-GRU with attention	1	94.13	94.91	94.01	77.31	80.69	90.11	66.34	57.19	87.30
	2	95.35	95.37	94.88	80.63	81.03	94.53	67.95	58.73	88.52
	3	*98.45*	*98.86*	*97.83*	*84.05*	*84.92*	*98.65*	*70.15*	*60.33*	*91.95*
Bi-LSTM with attention	1	95.16	94.87	95.19	76.99	78.13	89.28	68.58	58.14	88.39
	2	95.92	95.41	96.04	79.54	79.71	93.50	69.26	60.93	89.01
	3	*98.63*	*99.06*	*98.11*	*83.83*	*84.88*	*98.78*	*71.31*	*62.43*	*92.72*

The results in italics indicate the highest values

datasets for the SLU task. There are a few variants of the ATIS corpus, but in this paper, we follow the ATIS corpus used in [18, 52]. The ATIS corpus comprises of utterances of people making flight reservations. There are 4978 utterances in the training set of the corpus. The test set comprises 893 utterances. There are 17 distinct intent classes in the corpus. Flight represents about 70% of the dataset hence making the corpus highly skewed. There are three dialogue acts in the corpus, such as Question, Command and Statement. There are 127 distinct slots in the dataset.

TRAINS Dataset Although for SLU, there are many datasets, e.g. Cortana Data [13] and Bing Query Understanding Dataset [71], they are non-public. For building a robust spoken dialogue system, it is essential to capture the dialogue act (DA), intent and slots present in a human conversation. To be able to find DA, intent and slots of a real and natural utterance in a conversation, we manually annotate the TRAINS corpus. TRAINS corpus is a part of the TRAINS project. The corpus is a collection of problem-solving dialogues. The dialogues involve two speakers: one speaker plays the role of a user and has a specific goal to achieve, and another speaker plays the role of the system by acting as a planning assistant. Three annotators with post-graduate exposure were assigned to annotate this corpus with dialogue acts, intent and slot. We obtain an inter-annotator score of more than 80%, which may be considered a strong agreement. For dialogue systems, the tag-set used in our annotation comprises of the basic tags present in any dialogue annotated corpus. The labels for intent and slot were designed by going through the corpus in detail and by capturing the different intentions present in every utterance. The dataset comprises of 12 intents and 32 slots. There are 5355 utterances in the training set and 1336 utterances in the test set.

FRAMES Dataset The corpus consists of 1369 human-human dialogues. Each dialogue has an average of 15 turns. The corpus is a collection of multi-domain dialogues dealing with hotel bookings. There are 20,006 utterances in the training set and 6598 utterances in the test set. The dataset has been manually annotated with 24 intents and 136 slots.

Training Details

We use the python-based neural network package, Keras[4] for the implementation. In our work, we use one layer of Bi-LSTM/Bi-GRU, followed by an MLP. We fix the number of neurons on the Bi-LSTM/Bi-GRU layer to be 200.

The model uses a 250-dimensional word embedding. We use ReLU activations for the intermediate layers of our model and softmax activation for the output layer. Dropout [56] is a very efficient regularization technique to avoid over-fitting of the network. We use 15% dropout and 'Adam' optimizer [32] for regularization and optimization. Model parameters are updated using the categorical cross-entropy. Table 3 lists the different parameters that we experimented with and the final chosen parameters of the proposed model.

In this section, we present the details of evaluation results on three datasets. We also provide a comparison of our multi-task attention model with the baseline

[4] www.keras.io

Table 7 Results of multi-task model vs individual models

Model	Task	ATIS			TRAINS			FRAMES		
		Dialogue act (accuracy)	Intent (accuracy)	Slot (F1 score)	Dialogue act (accuracy)	Intent (accuracy)	Slot (F1 score)	Dialogue act (accuracy)	Intent (accuracy)	Slot (F1 score)
Bi-LSTM with attention	Only DAC	96.54	–	–	81.13	–	–	67.45	–	–
	Only ID	–	97.12	–	–	80.74	–	–	58.91	–
	Only SF	–	–	97.23	–	–	94.34	–	–	86.47
	MTM	*98.63*	*99.06*	*98.11*	*83.83*	*84.88*	*98.78*	*71.31*	*62.43*	*92.72*
Bi-GRU with attention	Only DAC	95.57	–	–	81.99	–	–	64.34	–	–
	Only ID	–	96.83	–	–	81.03	–	–	57.25	–
	Only SF	–	–	96.62	–	–	95.13	–	–	85.33
	MTM	*98.45*	*98.86*	*97.83*	*84.05*	*84.92*	*98.65*	*70.15*	*60.33*	*91.95*

The results in italics indicate the highest values

models. The effectiveness of our multi-task model has also been shown in contrast to individual models and pipeline models. Moreover, we provide a comparison of our model against the state-of-the-art approaches. In the

Table 8 Results of multi-task model with various pipeline models

Approach	Task	ATIS			TRAINS			FRAMES		
		Dialogue act (accuracy)	Intent (accuracy)	Slot (F1 score)	Dialogue act (accuracy)	Intent (accuracy)	Slot (F1 score)	Dialogue act (accuracy)	Intent (accuracy)	Slot (F1 score)
Bi-LSTM with attention	Multi-task model: DAC, ID, SF	*98.63*	*99.06*	*98.11*	*83.83*	*84.88*	*98.78*	*71.31*	*62.43*	*92.72*
	Pipeline model: DAC → ID, ID → SF	96.54	97.99	97.21	81.13	82.11	96.40	67.45	59.04	86.54
	Pipeline model: DAC → MTM (ID and SF)	96.54	98.32	97.65	81.13	82.63	96.88	67.45	59.63	87.09
	Pipeline model: MTM (DAC and ID) → SF	97.11	98.41	97.26	81.64	82.55	96.73	68.21	59.42	86.71
	Pipeline model: MTM (DAC and SF) → ID	97.23	98.04	97.51	81.55	82.36	96.54	68.03	59.17	86.99
	Pipeline model: DAC → ID, DAC, ID → SF	96.54	97.99	97.33	81.13	82.11	96.66	67.45	59.04	86.60
Bi-GRU with attention	Multi-task model: DAC, ID, SF	*98.45*	*98.86*	*97.83*	*84.05*	*84.92*	*98.65*	*70.15*	*60.33*	*91.95*
	Pipeline model: DAC → ID, ID → SF	95.57	96.42	96.17	81.99	82.85	96.97	64.34	57.48	85.63
	Pipeline model: DAC → MTM (ID and SF)	95.57	96.88	96.61	81.99	83.38	97.25	64.34	57.93	86.32
	Pipeline model: MTM (DAC and ID) → SF	96.45	96.63	96.45	82.26	83.40	97.13	65.84	57.81	85.73
	Pipeline model: MTM (DAC and SF) → ID	96.72	96.79	96.55	82.31	83.27	97.33	65.47	57.66	85.91
	Pipeline model: DAC → ID, DAC, ID → SF	95.57	96.42	96.21	81.99	82.85	97.11	64.34	57.48	85.69

The results in italics indicate the highest values

Table 9 Comparison of proposed multi-task model with state-of-the-art models

Model	ATIS		TRAINS		FRAMES	
	Intent (accuracy)	Slot (F1 score)	Intent (accuracy)	Slot (F1 score)	Intent (accuracy)	Slot (F1 score)
Attention BiRNN [37]	98.21	95.98	62.35	82.66	60.20	85.84
Attention encoder-decoder NN [38]	98.43	95.87	80.61	94.41	61.30	88.63
Bi-GRU [72]	98.32	96.89	79.85	94.67	59.88	87.95
Bi-model with decoder [63]	98.99	96.89	81.41	95.29	60.17	88.36
Slot-gated [10]	94.10	95.20	75.66	81.44	59.42	78.36
Proposed Bi-GRU with attention	98.86	97.83	84.92	98.65	60.33	91.95
Proposed Bi-LSTM with attention	*99.06*	*98.11*	*84.88*	*98.78*	*62.43*	*92.72*

The results in italics indicate the highest values

literature [8–10, 34, 37, 63, 72], the authors have used accuracy as an evaluation metric to model the performance of intent detection and dialogue act classification tasks while F1 score is used to evaluate the performance of the slot filling task. Hence, we report accuracy as the performance measure for dialogue act classification and intent detection tasks while F1 score is reported as the performance measure for the slot filling task.

Results

Character embeddings are known to capture the semantic information of infrequent and out-of-vocabulary words. To capture character-level features, we used a convolutional neural network to obtain the character feature representation. The results of the multi-task models with character embeddings as input are given in Table 4. Though the use of character embeddings does not help in achieving better performance, it helps in capturing the semantic representation of the unknown words.

To capture the word-level semantic information, we use three pre-trained word embedding models, i.e. Glove, Fasttext and word2vec. Experimental results by employing these embeddings only as input to the multi-task model are shown in Table 5. From the table, it can be seen that the model using Fasttext embeddings as input outperforms the other models using Glove and word2vec embeddings as input. Hence, in further experiments, we only use Fasttext as the word embedding input in all the models.

To provide both character-level and word-level features, we combine both character embeddings and word embeddings and feed the combined representation as input to our deep learning models. In Table 6, we show the results of using different deep learning layers using both character and word embeddings as input. From the table, it is evident that the model with 3 RNN layers outperforms the other models. Hence, we use stacked RNN layers to learn the utterance representation for all the tasks of SLU.

Individual Tasks vs Multi-task To analyse the performance of our proposed multi-task model, we implemented individual models for all the three tasks, i.e. dialogue act classification (DAC), intent detection (ID) and slot filling (SF). Table 7 shows the performance of individual models with respect to the multi-task model. The individual models have been implemented similarly as the multi-task model, with the only difference being that each model performs only one task. From the table, we can easily infer that the multi-task model performs better than the individual models as the representations learned by one task help in another, thereby improving the performance of all the tasks simultaneously.

Pipeline vs Multi-task The multi-task model has the flexibility of performing all the tasks together and therefore saves time and complexity as there would not be any individual model for each task. But to perform these tasks, one can take the pipelined approach as discussed above. In Table 8, we present the results of different pipelined approaches for performing the SLU tasks of

Table 10 Confusion matrix for dialogue act classification (DAC) on ATIS dataset

	Statement	Question	Command
Statement	217	0	1
Question	0	277	0
Command	7	1	390

Table 11 Confusion matrix for intent detection of ATIS dataset

Correct-estimated	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
a. Flight	624	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b. Flight_time	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c. Airfare	1	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d. Aircraft	1	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
e. Ground_service	0	0	0	0	36	0	0	0	0	0	0	0	0	0	0	0	0
f. Airport	1	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0
g. Airline	1	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0
h. Distance	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
i. Abbreviation	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0
j. Ground_fare	0	0	1	0	1	0	0	0	0	6	0	0	0	0	0	0	0
k. Quantity	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0
l. City	1	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0
m. Flight_no	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0
n. Capacity	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0
o. Meal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
p. Restriction	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q. Day_name	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

dialogue act classification, intent detection and slot filling. From the evaluation results, we can see that multi-task setting helps in improving the performance of each task. The main reason for the multi-task approach to outperform the different pipelined approaches is that in a multi-task approach, information is shared between the tasks, unlike the pipelined approach where information sharing is one-way. The error propagation is also handled well in the proposed multi-task model than the pipelined models.

Comparison with Previous Approaches SLU tasks are vital for every dialogue systems. In most of the existing models, intent detection and slot filling have been performed together. To analyse the effectiveness of our proposed approach, we compare it with the existing approaches. In Table 9, we present the results of the existing state-of-the-art approaches for intent detection and slot filling along with our proposed approach. It can be seen that our model outperforms the previous approaches for both the tasks of intent detection and slot filling for all the three datasets. Though there is a slight improvement for the intent detection task compared to the previous approaches for slot filling, we see more than 1% increase in accuracy in comparison to the previous approaches. We do not show have not the comparison with respect to dialogue act classification task because there has not been any prior work to the best of my knowledge on these datasets for DAC.

Error Analysis

To get an idea where our system fails, we perform a detailed error analysis of our best-performing multi-task model.

For the ATIS dataset, our proposed model performs quite well for the DAC task. However, there have been some errors where the statements have been misclassified as commands and vice versa. The confusion matrix for dialogue act classification on the ATIS dataset is given in Table 10. For example, '*Please find a flight from Las Vegas to Michigan*' was incorrectly classified as '*Statement*' whereas it should be labelled as '*Command*'.

Table 12 Confusion matrix for dialogue act classification (DAC) on TRAINS dataset

	Greeting	Statement	Question	Acknowledge	Command
Greeting	21	1	0	1	0
Statement	0	511	16	31	8
Question	0	46	132	7	2
Acknowledge	2	55	3	434	2
Command	0	26	3	1	34

Table 13 Confusion matrix for intent detection of TRAINS dataset

Correct-estimated	a	b	c	d	e	f	g	h	i	j	k	l
a. Capacity	20	0	0	0	0	0	1	0	0	0	0	0
b. City	0	23	0	0	0	0	1	0	0	1	0	3
c. Confirm	0	0	58	0	9	0	0	0	4	10	0	3
d. Deliver	0	0	0	1	0	0	0	0	0	0	0	0
e. Distance	0	0	5	2	366	0	13	0	2	12	0	0
f. Engine	0	0	0	0	1	224	4	0	0	0	0	0
g. Greet	0	1	1	0	19	20	198	15	1	7	0	0
h. Item	0	0	1	0	0	0	1	7	0	3	0	0
i. Other	0	0	0	0	1	0	1	0	57	6	0	0
j. Place	0	0	2	0	11	0	0	1	7	156	0	4
k. Time	0	0	0	0	0	0	0	0	1	5	0	0
l. Vehicle	0	0	1	0	2	0	1	0	9	0	0	11

Our detailed analysis further reveals that the intent errors are due to the embedding of prepositional phrases inside the noun phrase. The confusion matrix for the intent detection task on the ATIS dataset is given in Table 11.

In ATIS dataset, for example, the phrase '*Airfare of the flight from Texas to Chicago*', where the prepositional phrase suggests the utterance to be '*flight*' whereas the intent class is misclassified by the headword of the noun phrase (*airfare* in this case). Some errors were also encountered due to incorrect annotation. Certain utterances in the ATIS dataset are ambiguous and also ill-formulated, such as '*What's the airfare for a taxi to the Chicago airport?*'. In this case, the word '*airfare*' implies the intent class to be '*Airfare*', whereas the actual intent class should be '*Ground Service*'. In case of slot filling task, there have been cases where the slot tags have been incorrectly labelled as null tags. These happened mainly due to the less number of slot tags in comparison to the null tags. Few tags such as '*B-city_name*' have been miss-labelled as '*B-fromloc.city_name*' due to less number of instances of '*B-city_name*' in the training data. Let us consider the following utterance '*What is the ground transportation from Denver airport to downtown*'. Here, the word '*Denver*' is incorrectly tagged as '*B-fromloc.city_name*' whereas it should have been '*B-city_name*'. In another example, '*Which airport is closest to Montreal Quebec*', the word '*Quebec*' is miss-labelled as '*I-city_name*' whereas the actual tag is '*B-state_name*'. This is because the tag '*B-state_name*' has less representation in the data.

The relatively low accuracies for DAC and intent classification in the TRAINS dataset are mainly due to the ill-formulated sentences. The intents of many sentences are towards the end of the sentence and are not clearly stated. For example, '*So from Corning to Bath how far is that*' has been misclassified as '*Statement*' whereas it should have been a '*Question*'. Also, the intent for this sentence has been misclassified to be '*City*' while it should have been '*Distance*'. We perform quantitative analysis for both DAC and intent detection in the form of confusion matrices in Tables 12 and 13, respectively. In another example, '*Why do not they take the trucks from Avon*' has been incorrectly classified as '*Item*' whereas it should have been classified as '*Vehicle*'. In the utterance, '*before Avon we could we actually pick up those two boxcars which are at Bath*' was misclassified as '*Question*' but the correct DAC label is '*Statement*'. The slot result of the TRAINS dataset is high as the number of slots is less and also because the slots have easy patterns, which are learned by the model very well.

In case of FRAMES dataset, the confusion matrices for dialogue act classification and intent detection are demonstrated in Tables 14 and 15, respectively. From Table 14, we can analyse that the major confusion takes place between the DAC labels '*offer, suggest*', '*request, suggest*' and '*inform, request and suggest*'. For example, '*Would any packages to Mos Eisley be available if I increase my budget to 2500*' has been misclassified as '*Inform*' but the correct label should be '*Request*'. Similarly, in the utterance '*Would you be interested in Calgary*' has been misclassified as '*Offer*', but the correct label is '*Suggest*'. Due to the long length of utterances, many

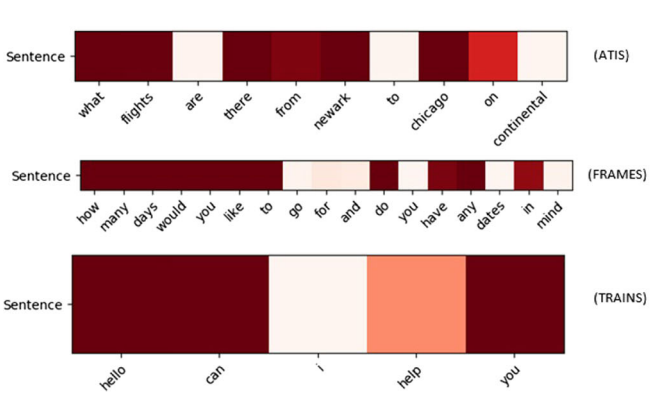
Table 14 Confusion matrix for dialogue act classification (DAC) on FRAMES dataset

Correct-estimated	a	b	c	d	e	f	g	h	i	j
a. Affirm	127	22	0	7	4	0	2	0	0	8
b. Confirm	21	170	18	0	0	15	12	2	5	3
c. Greeting	14	12	297	63	7	37	3	0	0	45
d. Inform	12	25	52	2057	7	18	49	305	221	95
e. Negate	0	0	10	0	26	36	0	0	0	29
f. No-result	7	20	0	0	102	156	0	0	0	31
g. Offer	0	0	18	0	0	0	453	20	127	0
h. Request	0	8	0	18	0	0	67	796	123	0
i. Suggest	0	0	0	9	0	0	14	52	133	0
j. Switch-frame	0	26	0	13	0	57	19	0	3	390

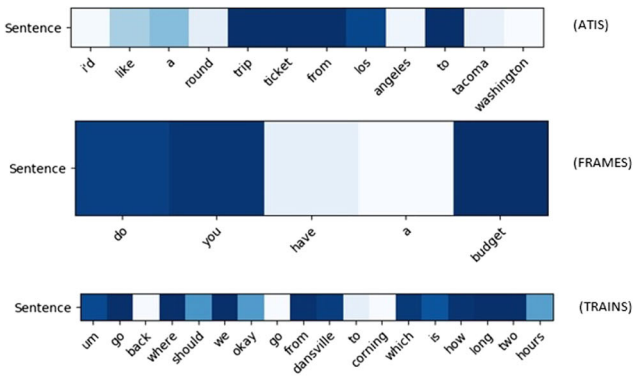
Table 15 Confusion matrix for intent detection of FRAMES dataset

Correct-estimated	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
a. Amenity	243	0	1	6	1	5	10	0	0	1	0	1	0	0	0	0	0	2	0	0	0	0	1	3
b. Availability_info	1	141	2	5	0	1	3	0	0	1	2	0	0	0	1	0	0	0	0	0	0	0	0	8
c. Book	0	1	241	6	0	1	2	1	0	0	1	0	0	1	0	0	6	0	5	0	0	0	1	0
d. Budget	1	1	3	447	5	3	9	4	2	8	19	1	4	28	2	6	25	21	0	1	3	0	6	2
e. Provide_budget	0	0	2	4	138	2	7	0	3	1	1	0	1	1	0	0	0	0	1	0	1	0	1	0
f. City	2	4	9	7	15	1258	2	9	11	9	7	2	5	7	12	11	6	25	1	7	8	3	4	2
g. Date	14	4	64	60	6	29	521	1	3	23	11	2	8	0	3	2	18	0	7	11	3	7	13	5
h. Departure	1	0	1	6	0	0	15	252	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
i. Destination	0	0	0	34	1	2	2	0	240	1	1	1	0	0	0	0	0	2	0	7	0	0	0	0
j. Duration	3	2	11	17	2	9	21	0	1	170	39	0	2	5	2	0	4	0	1	0	1	0	2	2
k. Flight	0	10	0	12	0	1	2	0	0	7	366	0	0	4	0	0	35	0	0	0	0	0	0	6
l. Greet	1	2	4	19	0	0	8	1	0	1	2	28	0	0	0	0	0	0	0	0	0	0	0	8
m. Hotel	0	0	1	4	4	0	1	0	0	2	2	1	43	0	1	0	1	2	0	0	0	0	2	0
n. Provide_hotel	0	0	1	4	0	2	1	0	0	0	1	0	0	10	0	0	0	0	0	0	0	0	0	0
o. No_of_ppl	1	3	0	7	2	2	8	0	0	3	1	1	0	0	98	1	0	0	1	0	0	0	1	0
p. Other	0	0	0	0	1	0	2	0	0	2	1	0	0	0	0	20	6	0	0	0	0	0	4	0
q. Package	0	0	7	8	0	1	3	0	0	2	7	0	0	0	0	3	448	0	0	0	0	3	2	0
r. Price	1	0	1	18	1	4	1	0	1	1	2	0	0	0	0	0	0	85	1	0	0	0	0	1
s. Provide_date	0	0	3	1	0	0	1	0	0	0	0	0	0	0	5	0	0	5	122	0	0	0	0	0
t. Provide_flight	0	2	2	12	0	6	7	0	6	0	0	0	0	0	0	0	0	1	0	123	1	0	1	0
u. Provide_info	0	0	1	7	0	7	8	0	2	3	0	0	0	0	0	0	0	0	0	1	5	0	2	0
v. Provide_trip	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	7	0	0	0	0	32	5	0
w. Rating	0	0	5	5	0	0	4	0	1	6	9	0	0	1	1	4	34	0	0	0	0	6	80	0
x. Trip	0	4	1	3	0	0	3	0	0	0	5	15	0	0	0	0	0	0	0	0	0	0	0	97

times the intent of the utterance is expressed in the latter part of the utterance causing misclassification. For example, '*Wow that is very good I am definitely keeping that one in mind, do you have anything in Frankfurt*' is misclassified as '*Other*', but the actual label should be '*package*'. In this work, we also handle single intents in an utterance; hence, the utterances having multiple intents cause the misclassification. For example, the utterance '*When would you like to travel and how many people will you be*' is incorrectly classified as '*no_of_ppl*' whereas in the dataset it has been labelled as '*date*'. Some misclassification also occurs due to less representation of some intent labels in the dataset. For example, '*Do you prefer a 3.5 star hotel or a 4 star hotel*' has been incorrectly labelled as '*rating*' but the original intent label for this utterance is '*hotel*'. The slot filling results for this dataset are not very high mainly because the number of tags is in huge

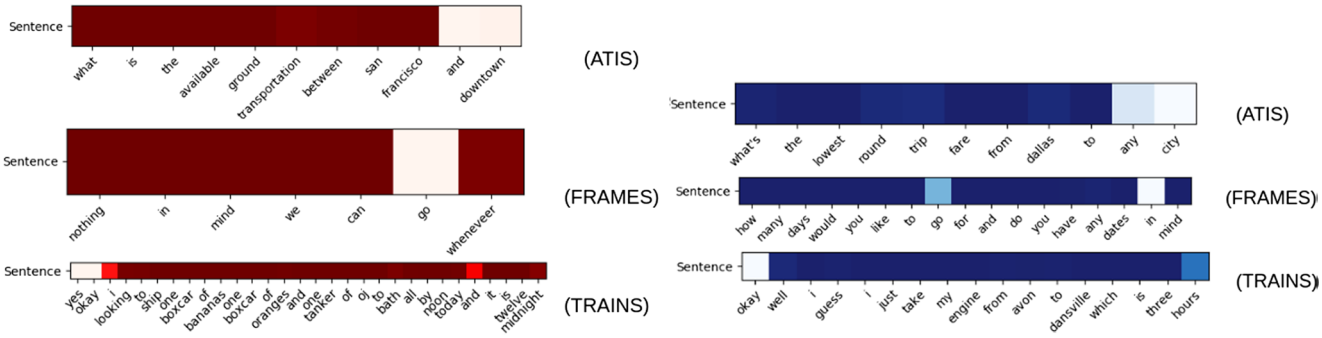


(a) DAC Attention visualization



(b) Intent Attention visualization

Fig. 4 Attention visualization for the multi-task model



(a) DAC Attention visualization

(b) Intent Attention visualization

Fig. 5 Attention visualization for the individual models

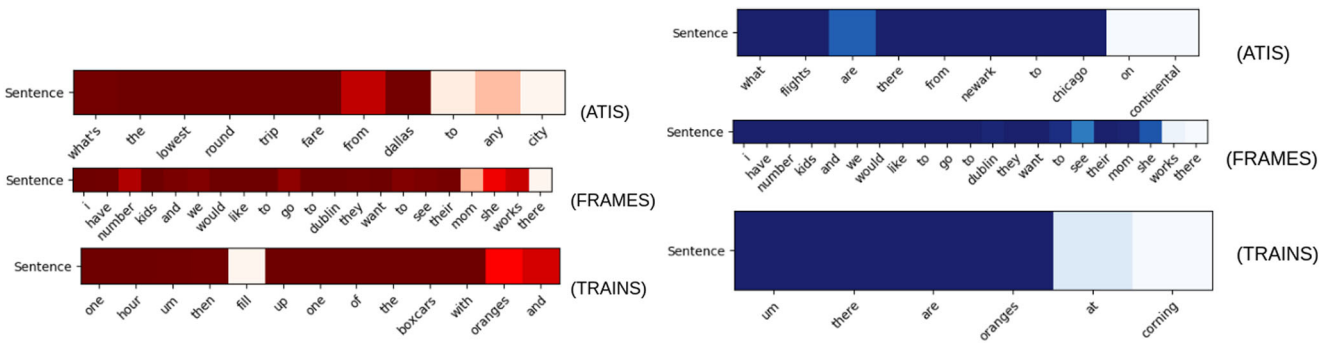
number and some slot tags have very less representation in the dataset.

To check the effectiveness of our multi-task model, we analyse the errors of the individual models as well. For example, the utterance '*How much is the Porto Alegre package for economy*' was incorrectly classified as '*package*' by the individual intent model but in the multi-task model, this utterance has been correctly classified as '*price*' with the help of slot labels. In another example, in the utterance '*Business or economy*', the slot labels '*B-flight_class*' help in correct intent detection which is '*flight*'. This utterance was incorrectly classified by the individual models. Similarly, intent detection has helped in slot filling task when both are modelled together in the multi-task setup. For example, the slot label for the word '*Denver*' in the utterance '*What is the airfare for economy class from Denver*' was correctly tagged as '*B-fromloc.city_name*' whereas in the individual slot model, it is misclassified as '*B-city_name*' due to the help of the intent label '*airfare*'.

In Fig. 4, we show the attention visualizations for both the task of dialogue act classification and intent detection. From Fig. 4a, we can see that by using attention, the model has shown improvement by focusing on the inputs that help in identifying the correct dialogue act of the sentence. Similarly, from Fig. 4b, we see that incorporating attention to our proposed model has helped by focusing on the input to detect the true intents of the utterances. In Figs. 5 and 6, we present the attention visualizations of the individual models and the pipeline model (where, DAC->*ID* − >SF), respectively. It is evident from the visualizations that the proposed multi-task model shown in Fig. 4 can attend the information correctly. Hence, it increases the performance of the model. In contrast, the pipeline and individual models are unable to focus on the correct information and give equal importance to all the words in an utterance which confuses the model and lowers the performance of the individual and pipeline models.

Statistical Significance Test

A statistical hypothesis test named Welch's *t* test [65] is conducted at the 5% (0.05) significance level to verify whether the improvement in our model is significant or not. This is done to show that the best accuracy obtained by our proposed method is statistically significant and has not occurred by chance. For the statistical test on all the dataset, the performance metric (accuracy) is produced by 20 consecutive runs of each algorithm. To establish the statistical significance of our method, we calculated the *p* values produced by Welch's *t* test for comparison of two groups.



(a) DAC Attention visualization

(b) Intent Attention visualization

Fig. 6 Attention visualization for the pipeline models

Table 16 Results of statistical significance test

Model	ATIS			TRAINS			FRAMES		
	DA (accuracy)	Intent (accuracy)	Slot (F1 score)	DA (accuracy)	Intent (accuracy)	Slot (F1 score)	DA (accuracy)	Intent (accuracy)	Slot (F1 score)
Bi-GRU	9.58E-26	8.49E-38	5.41E-24	6.59E-46	7.65E-36	1.27E-40	9.58E-26	8.49E-38	1.48E-22
Bi-LSTM	2.20E-37	3.50E-43	1.90E-40	9.05E-44	2.30E-48	4.66E-53	4.16E-49	2.21E-53	1.08E-50
Bi-GRU with attention	4.89E-13	6.85E-38	8.29E-07	5.41E-24	4.89E-13	3.71E-26	3.85E-29	4.42E-31	2.31E-39

Among these two groups, the first one corresponds to the accuracies produced by our best proposed multi-task model, and then another group corresponds to the accuracies produced by the other baseline and hierarchical models. The *t* test is a null hypothesis test that determines whether two sets of data are significantly different or not.

$$\mathcal{H}_0 : \lambda_1 = \lambda_2 = \lambda_3 \tag{14}$$

On the contrary, the alternative hypothesis ($\mathcal{H}_1$) is that there are significant differences between the average accuracies obtained by any of the two groups.

$$\mathcal{H}_0 : \lambda_1 = \lambda_2 = \lambda_3 \tag{15}$$

On the contrary, the alternative hypothesis ($\mathcal{H}_1$) is that there are significant differences between the average accuracies obtained by any of the two groups.

$$\mathcal{H}_1 : \exists \alpha, \beta : \alpha \neq \beta \Rightarrow \lambda_\alpha \neq \lambda_\beta \tag{16}$$

Where λ_k is the average accuracy of k^{th} algorithm. Now the differences between the average accuracies are calculated by the following *t* statistic formula:

$$t = \frac{\chi_1 - \chi_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{17}$$

where χ_i, σ_i^2 and n_i are the mean, variance and size of the i^{th} sample, respectively. The *p* value is the probability, under the assumption of the null hypothesis (H0) and the smaller *p* value is strong evidence against the null hypothesis($\mathcal{H}_0$).

For the statistical test on all the datasets, we execute the experiment 20 times. Table 16 reports *p* values produced by Welch's *t* test. All the *p* values reported in Table 16 are less than 0.05 (5% significance level). Hence, the better performance brought about with our approach is statistically significant.

Conclusion and Future Work

In this paper, we have proposed a multi-task approach for dialogue act classification, intent detection and slot filling, which are the primary tasks in SLU. For the multi-task model, we use Bi-LSTM and Bi-GRU to learn the representations of the sequence shared by all the tasks. The multi-task model exhibits advantages over individual models. On the ATIS dataset, our model outperforms the state-of-the-art approaches on intent detection and slot filling tasks, while it performs considerably well for DAC as well. On the TRAINS dataset, our model has shown good performance on the slot filling task while for DAC and intent detection, it has performed relatively well. By using combined word embeddings, it further helps our model to identify the dialogue acts, intents and slots correctly.

In our future work, we plan to incorporate syntactic and semantic information into our model. We want to expand the scale of our dataset and use more dialogue datasets, which can be useful for building robust dialogue systems and help in SLU research.

Acknowledgements Authors duly acknowledge the support from the Project titled "Sevak-An Intelligent Indian Language Chatbot", Sponsored by SERB, Govt. of India (IMP/2018/002072). Asif Ekbal gratefully acknowledges Young Faculty Research Fellowship (YFRF), supported by Visvesvaraya PhD scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Ang J, Liu Y, Shriberg E. Automatic dialog act segmentation and classification in multiparty meetings, In: IEEE International Conference on Acoustics, Speech, and Signal Processing, {ICASSP} '05, Philadelphia, Pennsylvania, USA, March 18-23, 2005, Vol 1, pp 1061–1064.
2. Bapna A, Tur G, Hakkani-Tur D, Heck L. Sequential dialogue context modeling for spoken language understanding, In: Proceedings of the 18th Annual SIGdial Meeting on Discourse

and Dialogue, Saarbrucken, Germany, August 15–17, 2017; pp 103–114.
3. Barahona LMR, Gasic M, Mrkšić N, Su PH, Ultes S, Wen TH, Young S. Exploiting sentence and context representations in deep neural models for spoken language understanding, In: 26th International Conference on Computational Linguistics, (COLING), Proceedings of the Conference: Technical Papers, December 11–16, 2016, Osaka, Japan; pp 258–267.
4. Chen L, Di Eugenio B. Multimodality and dialogue act classification in the RoboHelper Project; In: Proceedings of the SIGDIAL 2013 Conference, The 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 22–24 August 2013, SUPELEC, Metz, France; pp 183–192.
5. A. Deoras, R. Sarikaya, Deep belief network based semantic taggers for spoken language understanding., In: INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013, pp. 2713–2717.
6. Fernandez R, Picard RW. Dialog act classification from prosodic features using support vector machines, In: Speech Prosody 2002, International Conference; 2002.
7. Firdaus M, Bhatnagar S, Ekbal A, Bhattacharyya P. Intent detection for spoken language understanding using a deep ensemble model, In: 15th Pacific Rim International Conference on Artificial Intelligence (PRICAI), Nanjing, China, August 28-31, 2018, Proceedings, Part {I}, Springer, pp 629–642.
8. Firdaus M, Bhatnagar S, Ekbal A, Bhattacharyya P. A deep learning based multi-task ensemble model for intent detection and slot filling in spoken language understanding, In: Neural Information Processing - 25th International Conference, (ICONIP) 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part {IV}, Springer, pp 647–658.
9. Firdaus M, Kumar A, Ekbal A, Bhattacharyya P. A Multi-task hierarchical approach for intent detection and slot filling, In: Knowledge-Based Systems, Elsevier; vol-183; 2019.
10. Goo CW, Gao G, Hsu YK, Huo CL, Chen TC, Hsu KW, Chen YN. Slot-gated modeling for joint slot filling and intent prediction, In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pp 753–757.
11. Gorin AL, Riccardi G, Wright JH. How may I help you? Speech Comm. 1997; vol-23, pp 113–27.
12. Grau S, Sanchis E, Castro MJ, Vilar D. Dialogue act classification using a Bayesian approach, In: 9th Conference Speech and Computer; 2004.
13. Guo D, Tur G, Yih Wt, Zweig G. Joint semantic utterance classification and slot filling with recursive neural networks, In: Spoken Language Technology Workshop (SLT), IEEE, South Lake Tahoe, NV, USA, December 7-10, 2014; pp 554–559.
14. Haffner P, Tur G, Wright JH. Optimizing SVMs for complex call classification. In: Acoustics, Speech, and Signal Processing, IEEE International Conference, Hong Kong, April 6-10, 2003, vol 1, pp 632–635.
15. Hakkani-Tür D, Tur G, Chotimongkol A. Using syntactic and semantic graphs for call classification, In: Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing; 2005.
16. Hakkani-Tür D, Tür G, Celikyilmaz A, Chen YN, Gao J, Deng L, Wang YY Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM, In: 17th Annual Conference of the International Speech Communication Association, Interspeech, San Francisco, CA, USA, September 8-12, 2016; pp 715–719.
17. Hashemi HB, Asiaee A, Kraft R. Query intent detection using convolutional neural networks, In: International Conference on Web Search and Data Mining, Workshop on Query Understanding; 2016.
18. He Y, Young S. A data-driven spoken language understanding system, In: IEEE Workshop on Automatic Speech Recognition and Understanding, pp 583–588; 2003.
19. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80.
20. Jeong M, Lee GG. Triangular-chain conditional random fields. IEEE Trans. Audio Speech Lang Process. 2008; vol-16(7); pp 1287–302.
21. Ji G, Bilmes J. Dialog act tagging using graphical models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP) '05, Philadelphia, Pennsylvania, USA, March 18-23, 2005; vol 1, pp 33–36.
22. Ji Y, Haffari G, Eisenstein J. A Latent variable recurrent neural network for discourse relation language models, arXiv preprint arXiv:1603.01913; 2016.
23. Justo R, Alcaide JM, Torres MI, Walker M. Detection of sarcasm and nastiness: new resources for Spanish language. In: Cognitive Computation; 2018; vol-10; pp 1135–1151.
24. Kalchbrenner N, Blunsom P. Recurrent convolutional neural networks for discourse compositionality, In: Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, CVSM@ACL 2013, Sofia, Bulgaria, August 9, 2013, pp 119–126.
25. Keizer S. A Bayesian approach to dialogue act classification, In: BI-DIALOG 2001: Proceedings of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue, pp 210–218; 2001.
26. Keizer S, Nijholt A, et al. Dialogue act recognition with Bayesian networks for Dutch dialogues, In: Proceedings of the SIGDIAL 2002 Workshop, The 3rd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Thursday, July 11, 2002 to Friday, July 12, 2002, Philadelphia, PA, USA; Association for Computational Linguistics, pp 88–94.
27. Khanpour H, Guntakandla N, Nielsen R. Dialogue act classification in domain-independent conversations using a deep recurrent neural network, In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, December 11-16, 2016, Osaka, Japan, pp. 2012–2021.
28. Kim JK, Tur G, Celikyilmaz A, Cao B, Wang YY. Intent detection using semantically enriched word embeddings, In: Spoken Language Technology Workshop (SLT), IEEE, San Diego, CA, USA, December 13-16, 2016; pp 414–419.
29. Kim SN, Cavedon L, Baldwin T. Classifying Dialogue acts in one-on-one live chats, In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 9-11 October 2010, {MIT} Stata Center, Massachusetts, USA; pp 862–871.
30. Kim Y, Jernite Y, Sontag D, Rush AM. Character-Aware Neural Language Models, In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, pp 2741–2749.
31. Kim YB, Lee S, Stratos K. ONENET: Joint domain, intent, slot prediction for spoken language understanding, In: Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, Okinawa, Japan, December 16-20, 2017 pp 547–553.
32. Kingma D, Ba J. Adam: a method for stochastic optimization, In: 3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.
33. Kral P, Cerisara C. Automatic dialogue act recognition with syntactic features. Lang Resour Eval. 2014;48(3):419–41.
34. Kumar H, Agarwal A, Dasgupta R, Joshi S, Kumar A. Dialogue act sequence labeling using hierarchical encoder with CRF, In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of

Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pp 3440–3447.
35. Lauren P, Qu G, Yang J, Watta P, Huang GB, Lendasse A. Generating word embeddings from an extreme learning machine for sentiment analysis and sequence labeling tasks. In: Cognitive Computation, 2018; Springer; vol- 10; pp 625–638.
36. Li Y, Yang L, Xu B, Wang J, Lin H. Improving user attribute classification with text and social network attention. In: Cognitive Computation, 2019; Springer; vol- 11; pp 459–468.
37. Liu B, Lane I. Attention-based recurrent neural network models for joint intent detection and slot filling, In: Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016, pp 685–689.
38. Liu B, Lane I. Joint online spoken language understanding and language modeling with recurrent neural networks. In: Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA, pp 22-30.
39. Liu B, Lane I. Dialog context language modeling with recurrent neural networks, In: IEEE International Conference on Acoustics, Speech and Signal Processing; ICASSP, New Orleans, LA, USA, March 5-9, 2017; pp. 5715–5719.
40. Liu Y. Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus, In: Ninth International Conference on Spoken Language Processing, Interspeech, Pittsburgh, PA, USA, September 17-21, 2006.
41. Liu Y, Han K, Tan Z, Lei Y. Using context information for dialog act classification in DNN framework, In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, September 9-11, 2017; pp. 2170–2178.
42. Luan Y, Watanabe S, Harsham B. Efficient learning for spoken language understanding tasks with word embedding based pre-training, In: Sixteenth Annual Conference of the International Speech Communication Association, Interspeech, Dresden, Germany, September 6-10, 2015; pp 1398–1402.
43. McCallum A, Freitag D, Pereira FC. Maximum entropy Markov models for information extraction and segmentation. ICML. 2000;17:591–8.
44. Mesnil G, He X, Deng L, Bengio Y. Investigation of recurrent neural network architectures and learning methods for spoken language understanding, In: 14th Annual Conference of the International Speech Communication Association, Interspeech, Lyon, France, August 25-29, 2013; pp 3771–3775.
45. Mesnil G, Dauphin Y, Yao K, Bengio Y, Deng L, Hakkani-Tur D, et al. Using recurrent neural networks for slot filling in spoken language understanding. IEEE-ACM T Audio Spe. 2015;23(3): 530–9.
46. Moschitti A, Riccardi G, Raymond C. Spoken language understanding with kernels for syntactic/semantic structures. In: IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU, Kyoto, Japan, December 9-13, 2007; pp 183–188.
47. Papalampidi P, Iosif E, Potamianos A. Dialogue act semantic representation and classification using recurrent neural networks, In: Proc. SEMDIAL 2017 (SaarDial) Workshop on the Semantics and Pragmatics of Dialogue, pp. 77–86; 2017.
48. Pennington J, Socher R, Manning C. Glove: global vectors for word representation, In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), October 25-29, 2014, Doha, Qatar, pp 1532–1543.
49. Price PJ. Evaluation of spoken language systems: the ATIS domain, In: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27; 1990.
50. Ravuri S, Stoicke A. A comparative study of neural network models for lexical intent classification, In: IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, AZ, USA, December 13-17, 2015, pp 368–374.
51. Ravuri SV, Stolcke A. Recurrent neural network and LSTM models for lexical utterance classification, In: 16th Annual Conference of the International Speech Communication Association, Interspeech, Dresden, Germany, September 6-10, 2015, pp 135–139.
52. Raymond C, Riccardi G. Generative and discriminative algorithms for spoken language understanding, In: Eighth Annual Conference of the International Speech Communication Association, Interspeech; Antwerp, Belgium, August 27-31, 2007, pp 1605–1608.
53. Ribeiro E, Ribeiro R, de Matos DM. The influence of context on dialogue act recognition, arXiv preprint arXiv:150600839; 2015.
54. Ries K. Hmm and neural network based speech act detection, In: IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP), Phoenix, Arizona, USA, March 15-19, 1999; vol 1, pp 497–500.
55. Samei B, Li H, Keshtkar F, Rus V, Graesser AC. Context-based speech act classification in intelligent tutoring systems, In: International Conference on Intelligent Tutoring Systems, Springer, pp 236–241; 2014.
56. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.
57. Stolcke A, Ries K, Coccaro N, Shriberg E, Bates R, Jurafsky D, et al. Dialogue act modeling for automatic tagging and recognition of conversational speech. Comput Linguist. 2000;26(3):339–73.
58. Sun X, Peng X, Ding S. Emotional human machine conversation generation based on long short-term memory. In: Cognitive Computation, 2018; Springer; vol-10(3); pp 389–397.
59. Tur G. Model adaptation for spoken language understanding. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, Pennsylvania, USA, March 18-23, 2005; vol 1, pp 41–44.
60. Tur G, Hakkani-Tür D, Heck L, Parthasarathy S. Sentence simplification for spoken language understanding, In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 22-27, 2011, Prague Congress Center, Prague, Czech Republic; pp 5628–5631.
61. Venkataraman A, Ferrer L, Stolcke A, Shriberg E. Training a prosody-based dialog act tagger from unlabeled data, In: Acoustics, Speech, and Signal Processing, Proceedings (ICASSP'03), IEEE International Conference on, IEEE, Hong Kong, April 6-10, 2003; vol 1, pp 272–275.
62. Wang P, Song Q, Han H, Cheng J. Sequentially supervised long short-term memory for gesture recognition. In: Cognitive Computation, 2016; Springer; vol-8(5); pp 982–91.
63. Wang Y, Shen Y, Jin H. A bi-model based RNN semantic frame parsing model for intent detection and slot filling, In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), vol 2, pp 309–314.
64. Wang Z, Lin Z. Optimal feature selection for learning-based algorithms for sentiment classification. In: Cognitive Computation, 2019; Springer; vol-12, pp 238–248.
65. Welch BL. The generalization of student's problem when several different population variances are involved. Biometrika. 1947;34(1/2):28–35.
66. Xing C, Wu W, Wu Y, Liu J, Huang Y, Zhou M, et al. Topic aware neural response generation. In: Proceedings of the Thirty-First (AAAI) Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA; pp 3351–3357.
67. Xu P, Sarikaya R. Convolutional neural network based triangular CRF for joint intent detection and slot filling, In: IEEE Workshop

on Automatic Speech Recognition and Understanding (ASRU), Olomouc, Czech Republic, December 8-12, 2013, pp 78–83.

68. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification, In: 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pp 1480–1489.
69. Yao K, Zweig G, Hwang MY, Shi Y, Yu D. Recurrent neural networks for language understanding, In: 14th Annual Conference of the International Speech Communication Association (Interspeech), Lyon, France, August 25-29, 2013; pp 2524–2528.
70. Yao K, Peng B, Zhang Y, Yu D, Zweig G, Shi Y. Spoken language understanding using long short-term memory neural networks, In: IEEE Spoken Language Technology Workshop, {SLT} 2014, South Lake Tahoe, NV, USA, December 7-10, 2014; pp 189–194.
71. Yao K, Peng B, Zweig G, Yu D, Li X, Gao F. Recurrent conditional random field for language understanding, In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, May 4-9, 2014; pp 4077–4081.
72. Zhang X, Wang H. A joint model of intent determination and slot filling for spoken language understanding, In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, (IJCAI), New York, NY, USA, 9-15 July 2016, pp 2993-2999.
73. Zhao L, Feng Z. Improving slot filling in spoken language understanding with joint pointer and attention, In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, {ACL} 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers}, pp 426–431.
74. Zhou H, Huang M, Zhang T, Zhu X, Liu B. Emotional chatting machine: emotional conversation generation with internal and external memory, In: Proceedings of the Thirty-Second {AAAI} Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th {AAAI} Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018; pp 730–739.
75. Zhou Y, Hu Q, Liu J, Jia Y. Combining heterogeneous deep neural networks with conditional random fields for Chinese dialogue act recognition. In: Neurocomputing, 2015; Vol - 168; pp 408–17.
76. Zhu S, Yu K. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding, In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP), New Orleans, LA, USA, March 5-9, 2017, pp 5675–5679.