Knowledge-Based Systems 183 (2019) 104846

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A Multi-Task Hierarchical Approach for Intent Detection and Slot Filling☆

Mauajama Firdaus *, Ankit Kumar, Asif Ekbal, Pushpak Bhattacharyya

Department of Computer Science and Engineering, Indian Institute of Technology, Patna, India

ARTICLE INFO

Article history:
Received 28 November 2018
Received in revised form 12 July 2019
Accepted 13 July 2019
Available online 19 July 2019

Keywords:
Multi-task
Hierarchical
Intent detection
Slot filling

ABSTRACT

Spoken language understanding (SLU) plays an integral part in every dialogue system. To understand the intention of the user and extract the necessary information to help the user achieve desired goals is a challenging task. In this work, we propose an end-to-end hierarchical multi-task model that can jointly perform both intent detection and slot filling tasks for the datasets of varying domains. The primary aim is to capture context information in a dialogue to help the SLU module in a dialogue system to correctly understand the user and assist the user in achieving the desired goals. It is vital for the SLU module to capture the past information along with the present utterance said by the user to retrieve correct information. The dependency and correlation between the two tasks, i.e. intent detection and slot filling makes the multi-task learning framework effective in capturing the desired information provided by the user. We use Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to capture contextual information for the utterances. We employ Conditional Random Field (CRF) to model label dependency. Both character and word level embeddings are provided as input to the models. We create a benchmark corpus for the SLU tasks, on TRAINS and FRAMES dataset for capturing more realistic and natural utterances spoken by the speakers in a human/machine dialogue system. Experimental results on multiple datasets of various domains (ATIS, SNIP, TRAINS and FRAMES) show that our proposed approach is effective compared to the individual models and the state-of-the-art methods.

1. Introduction

In every dialogue system, the critical task is to understand the language spoken by the user which is known as the spoken language understanding (SLU). For the proper functioning of the dialogue system, it is essential to perceive the intended meaning from the user's utterance. The primary task of the SLU module in especially goal-oriented dialogue systems is intent detection and slot filling. To understand the intentions of the user as to what the user wants to achieve is termed as intent detection. While the extraction of the necessary information from the user's utterance in the form of slots is referred to as slot filling. To satisfy the user's demand/requests, it is imperative to detect the intent of the user and fill all the possible slots to help the user in achieving their goals. Though natural language understanding is a complex task, still several human/machine dialogue systems have been developed for limited domains. To capture the semantic information from the user's utterance, intent detection and slot filling are the critical tasks of any goal-oriented systems. According to the extracted information from the user's utterance, the system can then decide upon the actions to be taken, to assist the user's in achieving their goals. For better understanding, the goal is to convert the user input, into a semantic representation of the user's intention at every turn. The dialogue manager decides the most appropriate system action after interpreting the user's intent. In our everyday lives, the applications of SLU are increasing significantly. Nowadays, there are numerous devices such as smartphones which have personal assistants that are built on SLU technologies.

1.1. Problem definition

In this paper, we solve two very significant tasks of SLU *viz.* intent detection and slot filling.

1.1.1. Intent detection

The primary objective in every goal-oriented dialogue system is to automatically detect the intention of the user expressed in

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.07.017.

* Corresponding author.
E-mail addresses: mauajama.pcs16@iitp.ac.in (M. Firdaus), ankit.mtcs17@iitp.ac.in (A. Kumar), asif@iitp.ac.in (A. Ekbal), pb@iitp.ac.in (P. Bhattacharyya).

https://doi.org/10.1016/j.knosys.2019.07.017
0950-7051/

Table 1
Examples of different intents from ATIS dataset.

Sentence	Intent
Show me flights from Pittsburgh to Los Angeles	*Flight*
What's the airport at Orlando	*Airport*
What is the arrival time in Dallas for the flight leaving Washington	*Flight_time*
Show me ground transportation in Denver	*Ground_service*
Show me the meals on flights from Atlanta to Seattle	*Meal*

natural language; a task referred to as intent detection. Intent detection primarily is considered a semantic utterance classification task. The main objective is to classify the user utterance x, consisting of a sequence of words $x = (x_1, x_2, \ldots, x_T)$ into one of the N pre-defined set of intent classes, y_i, based upon the utterance of the user such that:

$$\hat{y}_i = \underset{i \in N}{\operatorname{argmax}} P(y_i/x) \tag{1}$$

In Table 1, we show some intents taken from the ATIS dataset [1], the most extensively used dataset in SLU.

1.1.2. Slot filling

The extraction of semantic constituents from an input text is referred to as slot filling. It requires to fill in the values for a predefined set of slots in a semantic frame. The slot filling task is the assignment of 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:

$$\hat{s} = \underset{s}{\operatorname{argmax}} P(s/x) \tag{2}$$

In Table 2, the slot labels are given of the sentences according to the IOB representation.[1] These utterances are taken from the ATIS dataset. Slot filling is treated as a sequence labelling task as the slot of a given word is dependent upon the previous words. Hence, to identify the correct slots, it is essential to capture the information present in the entire sequence.

1.2. Motivation and contribution

Intent detection is a standard utterance classification task and is considered less complex than the other semantic analysis tasks, but the errors made by the intent detector is more visible as it leads to wrong system responses. Hence, a robust intent detection system plays a crucial role in building an effective dialogue system. Though in a chatbot, generally a pipeline structure is followed where the intent of a sentence is detected, and then the slots are extracted, but this approach can lead to errors if the intent is not identified correctly. The errors made by the intent detector propagates to the slot filling module as they are interrelated tasks. For example, if the intent of the sentence is *"flight"* the most probable slots are *"from_loc.city_name, to_loc.city_name"* whereas if the intent of the sentence is *"airfare"* the probable slot are *"class_type"*. Only after the correct detection of intents, the appropriate slots of the utterance can be extracted. Both the tasks are highly inter-related, and the information of one can help in another. This is the underlying motivation of performing both the tasks together. Also, by handling intent and slot together, we can build an end-to-end natural language understanding (NLU) module for any task-oriented chatbot.

[1] Here B, I and O denote the beginning, intermediate and outside entity of a slot.

The "naturalness" of the language is the most challenging part of any NLU module as the system has to deal with different types of realistic, natural sentences spoken by multiple users. In the majority of the existing works intent detection and slot filling have been done in isolation until recently when the interrelatedness of these tasks have been modelled together. In this paper, we propose a multi-task model that can detect the intents and extract the slots from any given utterance simultaneously providing an end-to-end framework for the NLU module. The information shared by both the tasks helps in identifying the intents and slots accurately.

The major contributions of this work are:

- We propose a multi-task model for intent detection and slot filling employing a hierarchical convolutional network and hierarchical convolutional recurrent network to capture utterance dependency.
- We use the conditional random field (CRF) to capture label dependency in our hierarchical multi-task models along with character and word embeddings as input to these models.
- We create a benchmark corpus for the SLU tasks, i.e. intent detection and slot filling on TRAINS and FRAMES dataset for capturing more realistic and natural utterances spoken by the speakers in a human/machine dialogue system.
- We conducted experiments on four datasets of SLU. Our multi-task model obtains significant improvements over the state-of-the-art methods and the previous approaches thereby presenting the effectiveness of our approach in detecting the intents and slot for different datasets belonging to different domains.

Our current work differs from the existing works with respect to the following ways: (i). We employ a hierarchical neural network model for capturing contextual information which has not been investigated in details in the past, especially for joint modelling of intent detection and slot filling. The ability of this model to remember past information helps it in identifying the correct intent and slots present in the current user utterance; (ii). We employ probabilistic CRF classifier to replace the traditional Softmax classifier. The intuition was to exploit the capability of probabilistic CRF classifier that has shown tremendous success in solving sequence labelling tasks. With CRF's ability to capture label dependency majorly helps the slot filling task to obtain improved performance for all the datasets; (iii). The efficiency of our model also stems due to the incorporation of character level embedding (along with word embedding) that helps in dealing with the out-of-vocabulary (OOV) word problem.

2. Related work

SLU is an integral part of every dialogue system. Dialogue systems are mostly like knowledge-based systems. In [2], the author proposed a technique based on a new type of speech recogniser. The recognised speech is further processed for identifying the intents and slots of the utterance. The primary tasks of SLU are intent detection and slot filling which aims to capture semantic concepts from user utterance to satisfy their requests. SLU research has emerged from the call classification systems [3] and the ATIS project [1]. In the past intent detection and slot filling has been done individually as well as together. We provide a brief overview of the works already done concerning these SLU tasks. In [4], a new technique was proposed to increase the performance of dialogue systems that employ an automatic approach to correct frames that were incorrectly generated by the system which helps with spoken language understanding.

Table 2
Slot filling examples from ATIS dataset.

Sentence	Slot labels
Show me flights from Pittsburgh to Los Angeles	*O O O O B-fromloc O B-toloc.city_name I-toloc.city_name*
What's the airport at Orlando	*O O O O B-city_name*
What is the arrival time in Dallas for the flight leaving Washington	*O O O O O O B-toloc.city_name O O O O B-fromloc.city_name*
Show me ground transportation in Denver	*O O O O O B-city_name*
Show me the meals on flights from Atlanta to Seattle	*O O O B-meal O O O B-fromloc.city_name O B-toloc.city_name*

2.1. Intent detection

Formerly, intent detection was implemented by using various traditional machine learning approaches such as support vector machine (SVM) [5] and boosting techniques [6–8]. The authors in [5] proposed a global optimisation technique using SVM for handling a combination of different binary classifiers for complex call routing. In [6], the author used the boosting technique for intent classification in an SLU system. A dependency parsing based approach using boosting for extracting important keywords from the utterances was proposed in [7], for detecting the intents and slots separately of the utterances from the ATIS corpus. In [8], the authors proposed an automated algorithm for mapping intents across applications. Syntactic and semantic graphs (SSG) capturing the heterogeneous features of an utterance was used for intent detection in [9]. The authors in [10], used maximum entropy classifiers for intent detection on the ATIS corpus.

A future direction towards solving the intent detection problem is deep learning, as it combines both feature extraction and classification into the learning process. In [11], enriched word embeddings were given as input to bi-directional LSTM [12] for detecting the intents. A multi-scale RNN structure for efficient learning of low-resource corpora for the SLU task of intent detection was proposed in [13]. Convolutional neural networks (CNN) [14] was employed in [15] for detecting the intents of a user search query. In [16], LSTM along with hashing techniques to handle out of vocabulary words for detecting the intents was employed on the ATIS dataset. Lexical features as input to different neural network architectures to provide a comparative analysis for the task of intent detection was investigated in [17]. An ensemble based deep learning architecture using CNN, LSTM and GRU was employed in [18] for intent detection on ATIS dataset. In [19], the authors proposed various machine learning and deep learning models with different vector representations of words for a code-mixed dataset in Hindi and English language. Recently in [20], capsule neural networks have been used for identifying intents on SNIPs dataset. To identify the intents of a user utterance in the form of questions, domain-type and dialogue act, the authors in [21,22] proposed a multi-task adversarial framework for English and Japanese datasets. The works as mentioned above on intent detection are different from the work proposed in this paper as we model a multi-task framework that performs classification along with sequence labelling. Also, the information shared by the two tasks helps in improving the performance of each task in comparison to the tasks being done individually.

2.2. Slot filling

As already discussed, slot filling is a sequence labelling problem in which every word of the utterance is assigned to a tag. To overcome the label bias problem suffered by locally normalised models, factorised probabilistic models such as maximum entropy markov models (MEMM) [23] and conditional random field (CRF) [24] has been employed for sequence labelling tasks such as slot filling that helps indirectly capturing the global distribution. SVM along with syntactic features captured by syntactic and semantic tree kernels was employed in [25] for slot filling. The exploitation of non-local features to model long-distance dependencies was employed in [26]. In [27], the authors proposed a technique that considered both pieces of knowledge of semantic frames and understanding of words for improving the performance of spoken dialogue systems.

Several deep learning architectures have also been employed for extracting essential information in the form of slots from a given utterance. The authors in [28] investigated deep belief networks (DBN) for slot filling on the ATIS dataset. In [29], the author's investigated Elman and Jordan type RNNs for slot filling. In [30], several hybrid variants of RNN was proposed due to the stronger ability of RNNs to capture dependencies compared to traditional models, such as CRF. In [31] lexical, syntactic and word-class features was used as input to an RNN for the SLU task of slot filling. The authors in [32] used transition features to improve RNNs and the sequence level criteria for optimisation of CRF to capture the dependencies of the output label explicitly. The authors in [33] used deep LSTMs along with regression models to obtain the output-label dependency for slot filling. The usage of kernel deep convex networks (K-DCN) was investigated in [34] for slot filling. In [35], a focus mechanism for an encoder–decoder framework was proposed for slot filling on the ATIS dataset. The authors in [36] introduced a generative network based on the sequence to sequence model along with pointer network for slot filling. In [37], attention-based encoder–decoder framework has been employed for slot filling.

In [38], a pre-trained language model was employed in an RNN framework for the slot filling task. Attention-based RNN framework was proposed in [39] along with pre-trained word embeddings for identifying the slots on ATIS and MEDIA dataset. On the ATIS dataset in [40], an adversarial multi-task model combining bi-directional language model with a slot tagging model was used for identifying the slots in a given user utterance. The adversarial framework was used in [41] for learning common representation across multiple domains for the slot filling tasks. In [42], the authors proposed the concept of transfer learning for the task of slot filling as it is an essential task of language understanding. In contrast to the existing works on slot filling, in this paper, we propose a multi-task model for identifying the intent and slots present in a user utterance simultaneously. To capture the dependencies between the labels we employ CRF that has been used for the various sequence labelling tasks.

2.3. Joint tasks

Due to the correlation between the intent detection and slot filling tasks, these SLU tasks have been modelled together using various deep learning architectures. The authors in [43] employed triangular CRF, that used an additional random variable for detecting the intents on top of the standard CRF along with slot filling. Also, CNN based triangular CRF model for joint intent detection and slot filling was proposed in [44] in which the features were extracted by the CNN layers and was shared by both the tasks. Hierarchical representations within the input text learned using a recursive neural network (RecNN) was proposed for the joint task [45] of intent detection and slot filling. In [46], step n-gram model along with RNN and CNN was used for modelling

both the tasks. In [47], the intent variation was modelled continuously along with the arrival of new words to achieve better performance for the joint task using LSTM. In [48], authors used bidirectional 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 [49]. In [50], the authors investigated alternative architectures for modelling lexical context for SLU and presented a joint approach using single bi-directional RNNs with LSTM cells for the domain, intent and slot filling. In [51], 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 [52].

A bi-model based RNN semantic frame parsing network structures were employed for intent detection and slot filling in [53]. The authors in [54] 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 [55], the authors proposed a multi-task ensemble model using combined word embeddings as input to the neural models. The authors in [56] proposed a zero-shot learning framework for two new languages (Hindi and Turkish) along with a few labelled examples of the new languages. To handle the open-vocabulary slots, a long-term aware attention mechanism along with positional encoding was proposed in [57]. On two real-world datasets, a multi-task model for intent and slot using capsule neural networks was proposed in [58]. The authors in [59] proposed an ELMo-Light model for faster and accurate pre-training of the SLU module.

Our proposed model differs from the previous works as we propose a multi-task model employing hierarchical CNN and hierarchical convolutional recurrent model for modelling intent and slots of a given utterance simultaneously. To the best of our knowledge, this is the first work that incorporates character embeddings, word embeddings and syntactic features as input to hierarchical multi-task models for both the SLU tasks of intent detection and slot filling. The probabilistic CRF was also used for improving the performance of the model as it captures the dependencies among the labels in a given user utterance.

3. Methodology

In this section, we will illustrate our multi-task model (MTM) for modelling the utterances to detect the intents and capture the semantic information in the form of slots. In a broad-way, our model can be said to be composed of the embedding layer having both character level and word level embeddings of the utterance. It is followed by a shared hierarchical representation to capture the contextual information of the utterances followed by two separate output layers which are the CRF layer for performing intent detection and slot filling respectively. The outputs of the hierarchical models are also fed to two different MLPs for intent and slot respectively. To see the effectiveness of CRF in capturing label dependency we have made a fair comparison of CRF with the MLP for both the tasks. The overall architecture of the model is given in Fig. 1.

3.1. Embedding layer

To capture the hidden semantic information of the words in a given utterance we have incorporated both character level and word level embeddings as input to our models.

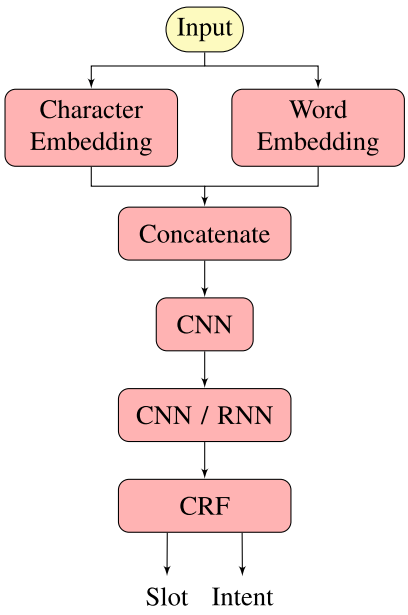


Fig. 1. Block diagram of our proposed approach; The input is represented by their corresponding word and character embeddings, the concatenated representation is fed to CNN followed by CNN/RNN to provide hierarchical information; CRF is used as the final output layer to predict the intent and slots of a given utterance.

3.1.1. Character embeddings

Character embeddings have known to be very useful for classifications tasks [51,60]. Character embeddings are known to perform well for out of vocabulary (OOV) words and infrequent words. In contrast to word embeddings that treat a word atomically, character embeddings construct a vector for a word from the characters that composes it. Word embedding models suffer from a lack of enough training opportunity for uncommon words whereas character embedding models can still learn proper embeddings of words with the help of characters in words. Previously, in [51], the authors used Char LSTMs to capture the character level patterns of words in an utterance. In our work, the character feature vectors are extracted from character embeddings of characters present in a word using LSTM (known as CharLSTM) and CNN (known as CharCNN). We do a comparative study of CharCNN and CharLSTM for capturing character representation of words in the form of character embeddings as they help in obtaining morphological and semantic features of any given the word.

Let, a given sentence ($\mathcal{S}$) which consist of a sequence of words $\mathcal{S} = [w_1, \ldots, w_n]$ where $w_{i|i\in n}$ represents each word. Similarly, each word ($w_{i|i\in n}$) is represented by a set of characters i.e. $w_i = [c_1, c_2, \ldots, c_k]$. Now to get the character feature vector, a composite function $\mathcal{F}$ is used where $\mathcal{F} = f_3 o f_2 o f_1$ where o is a composition operator. Here the functions are defined as follows:

- $f_1 : c_i \rightarrow \vec{y}_i$ where, $\vec{y}_i$ represents the fixed length one hot representation of character c_i
- $f_2 : \{\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m\} \rightarrow \vec{V}$ where, $\vec{V}$ represents a character embedding of a word which is concatenation of fixed length one hot representation of characters
- $f_3 : \vec{V} \rightarrow e_c$ where, e_c is the character feature vector extracted either by CNN or LSTM.

3.1.2. Word embeddings

Word embeddings are very useful in capturing hidden semantic structures. It represents words as real-valued vectors as the

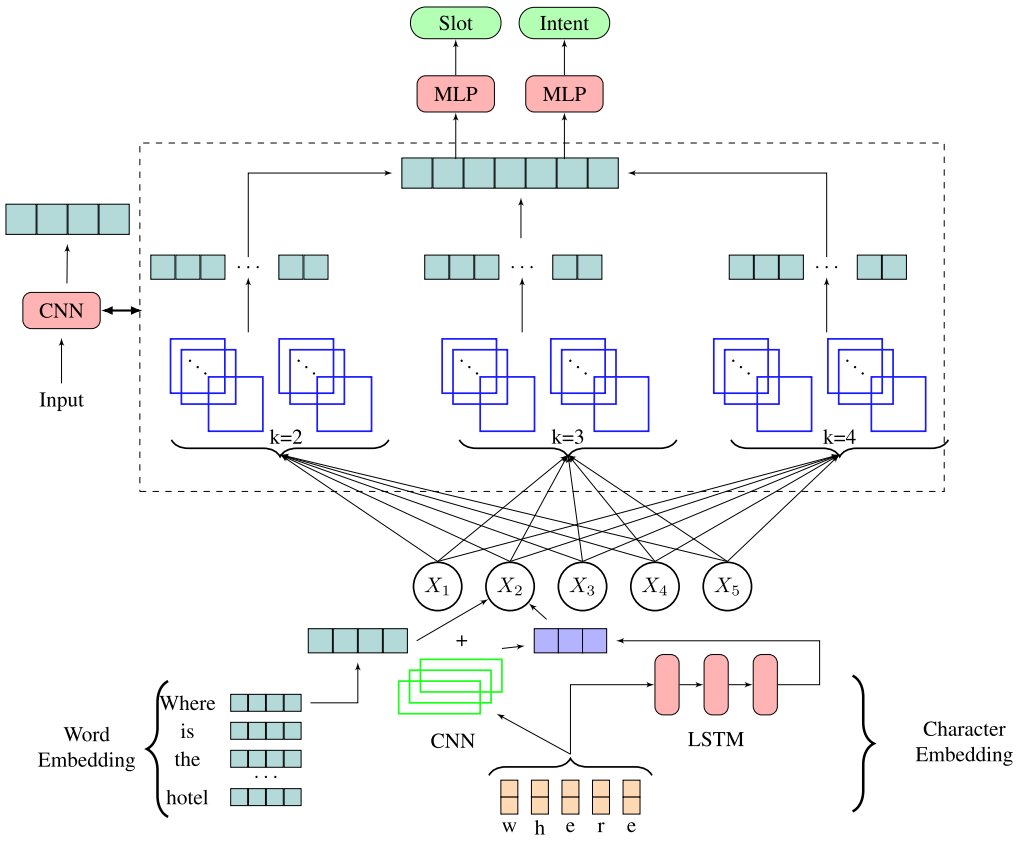


Fig. 2. Baseline CNN model with character and word embeddings; every word is represented by pre-trained word embedding as well as character embedding; character feature vectors is computed using CNN/LSTM, the concated representation is fed as input to the CNN with three filters of sizes 2, 3, 4 respectively, The output of CNN is fed to MLPs for identifying the intents and slots of an utterance.

input to the neural networks based models as opposed to other traditional representations such as one-hot representation. Bag-of-words or one-hot word vectors if fed to the model leads to feature vectors of large dimensions. As an alternative, word embeddings project the large sparse word vectors into a low dimensional, dense vector representation. Word embeddings are usually trained through an unsupervised manner on a huge dataset, and then the embeddings are fine-tuned by the supervised training process. For word embeddings, we use two pre-trained embedding models: GloVe[2][61] and Word2Vec.[3] We use both these embeddings separately as input to the different deep learning models for the SLU tasks of intent detection and slot filling. Also, we combine both the embeddings and provide it as input to the different models. We observe that the combined word embedding performs better as they capture the meanings from both the embeddings, i.e., Glove and Word2Vec. Our analysis shows that this combined representation yields improved performance.

Let, $\mathfrak{D} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_p\}$ is a dataset that contains a set of p sentences. Here a sentence $\mathcal{S}_i$ consist of a set of words i.e. $\mathcal{S}_i = \{w_1^i, w_2^i, \ldots, w_n^i\}$. From the entire dataset $\mathfrak{D}$ we extract a set of unique words $\{\omega_1, \omega_2, \ldots, \omega_m\}$ and store them in a dictionary $\mathbb{D}$. The words are then converted into fixed length vectors with the help of $\mathbb{D}$. Then a mapping function is used to retrieve the word vectors from either Glove or Word2vec for every word in $\mathbb{D}$. In our work, we use a combined representation of both the embeddings as input to the models. A function is used to concatenate the word embeddings from both Glove e_{w_g} and Word2vec e_{w_v} to give final word embeddings e_w, of every word in $\mathbb{D}$ i.e.,

$$e_w = e_{w_g} \oplus e_{w_v} \tag{3}$$

Along with character embeddings, we concatenate word embeddings and provide both the embeddings, i.e., character and word, as input to our models. So, the final input to the model can be defined as:

$$x_i = e_{c_i} \oplus e_{w_i} \tag{4}$$

where, $\oplus$ denotes the concatenation.

3.2. Baseline CNN model

All our models have been developed based on the basic CNN sentence representation, which has been quite useful for many natural language processing (NLP) tasks. Our baseline model is shown in Fig. 2, uses the features extracted by the CNN followed by task-specific multi-layer perceptron (MLP) layers. Let $x_i \in \mathbb{R}^d$ be the d-dimensional word vector of the ith word in the utterance. Now, x_i is fed to a convolutional layer that involves a filter $k \in \mathbb{R}^{hd}$, which is applied to a window of h words in an utterance to produce a new feature. A feature c_i is generated from a window of words $x_{i:i+h-1}$ by

$$c_i = f(k \odot x_{i:i+h-1} + b) \tag{5}$$

where b is the bias term, $\odot$ denotes element-wise multiplication and f is a non-linear function. The feature map of the entire utterance can be written as $c = [c_1, c_2, \ldots, c_{n-h+1}]$ by applying the filter to each possible window of words present in the utterance. A max pooling operation is applied to get a maximum value $\hat{c} = max[c]$ corresponding to a particular filter. In our work, we have used three different filters to capture the features of the utterance. The feature representation of each convolutional layer is concatenated and then fed as input to the task-specific layers for identifying the intents and slots in a given utterance.

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

3.3. Hierarchical models

To capture the contextual information of the utterances we propose hierarchical models for the SLU tasks of intent detection and slot filling.

3.3.1. Hierarchical CNN + CNN model

Our baseline model helps in capturing the information of the sentence. To get contextual information, we use another CNN that takes as input the information of our baseline CNN. The architecture of this hierarchical model can be seen in Fig. 3. The first CNN helps in capturing the word sequence of any given utterance while the second CNN assists in obtaining the sentence sequence. This model takes a hierarchical neural network approach where the high-level CNN is used to capture context information. For the second CNN, the sequence of utterances is represented by fixed length vectors denoting the number of utterances in a conversation. The output of the hierarchical model is fed as input to the task-specific layers for intent detection and slot filling respectively.

3.3.2. Hierarchical CNN + RNN model

Recurrent neural networks (RNN) [62] are known to capture sequential information. RNN is considered as an extension of the feed-forward neural network, that handles the sequences of variable length by using a recurrent hidden state that captures the information of the previous states as well. To capture long-distance dependencies, it is challenging to train RNN because of the vanishing or exploding gradient problem suffered by an RNN. To handle the issues of vanishing or exploding gradients, a few sophisticated activation functions with gating units were designed. Two improvements of RNN are the LSTM and the recently proposed GRU. Here we use both these variations, i.e., LSTM and GRU [63].

LSTM is a special kind of RNN, which can efficiently learn long-term dependencies. LSTM uses three gating units such as input, forget and output to regulate the amount of the current, previous and output representations that should be incorporated in the current timestamp. GRU is also a special kind of recurrent neural network, which overcomes the shortcomings of RNN. Unlike LSTM, GRU uses two gates — reset and update. The update gate takes care of the amount of past information that needs to be passed in the future whereas the reset gate decides the amount of previous information to forget.

In this hierarchical model, the utterance representation is learned by the baseline CNN model, while RNN models the contextual information among utterances. For sequence labelling, it is better to take into account both the past and future information of the sequence at the same time. Therefore, a bidirectional LSTM/GRU is used to learn the contextual representation of the utterance at every time step. The hidden layers of the bi-directional LSTM can be defined as follows:

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(c_t, \overrightarrow{h}_{t-1}) \quad (6)$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(c_t, \overleftarrow{h}_{t+1}) \quad (7)$$

where the forward and backward directions are denoted by $\rightarrow$ and $\leftarrow$ respectively. The forward and backward hidden states are concatenated to give the bidirectional hidden state $\overleftrightarrow{h}_t$ at time t.

$$\overleftrightarrow{h}_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \quad (8)$$

Both the tasks share the hidden states of the bidirectional LSTM (Bi-LSTM). On the one hand, the hidden states of the Bi-LSTM extract the features at every time step, so the outputs of the Bi-LSTM at every time step are used for predicting the slot labels.

Table 3
Statistics of training and test sets and the total intent and slot distribution for all the datasets.

Dataset	#Train	#Test	#Intent	#Slot
ATIS	4 978	893	17	127
TRAINS	5 355	1336	12	32
FRAMES	20 006	6598	24	136
SNIP	13 084	700	7	72

On the other hand, the last output of the Bi-LSTM contains the information of the entire utterance and is hence used for intent detection as seen in Fig. 4. Similarly, we compute the hidden states of the Bi-GRU for capturing the utterance information.

3.4. Task specific layers

3.4.1. Multi-layer perceptron

The last part of our model is the output layer. The softmax function is applied to the representation with the linear transformation to produce the slot labels y_s and the intent labels y_i simultaneously for a given utterance. Formally,

$$y_s = softmax(W_s \overleftrightarrow{h}_t + b_s) \quad (9)$$

$$y_i = softmax(W_i \overleftrightarrow{h}_t + b_i) \quad (10)$$

where W_s and W_i are transformation matrices for slot filling and intent detection, respectively, b_s and b_i are the bias vectors.

3.4.2. Linear chain CRF

In our hierarchical models, the classifier is a linear chain CRF which assists us in modelling label dependencies. The hierarchical models have already captured the dependencies among the utterances. CRFs are undirected graphical models that help in achieving the conditional probability of a label sequence given an observed sequence. Two CRFs are used to capture the label dependencies for intent and slot respectively.

4. Datasets and experiments

4.1. Datasets

We assess our proposed hierarchical model by conducting experiments on four datasets. The first dataset is the extensively used ATIS [1] corpus, and the other datasets are the TRAINS [64], FRAMES [65] and Snips dataset [54]. The TRAINS and FRAMES datasets consist of dialogue conversations, and we have manually annotated this with intents and slots. Three annotators graduate in English linguistics were assigned to annotate these corpora with intent and slot. The inter-annotator score with more than 80% was considered a reliable agreement. The utterance, intent, and slot distribution for all the datasets are given in Table 3.

ATIS dataset: The ATIS (Airline Travel Information System) corpus is an essential by-product of the Defence Advanced Research Program Agency (DARPA) project. For the SLU tasks, one of the most extensively used dataset is the ATIS corpus [1]. ATIS corpus has a few variants, but in this paper, we use the dataset used in [24]. The utterances in ATIS corpus mainly are of flight reservations made by people. There are 17 distinct intent classes and 127 distinct slots in the dataset. There are 4978 utterances in the train set and 893 utterances in the test set.

TRAINS dataset: Although for SLU, there are datasets, e.g., Cortana Data [45] and Bing Query Understanding Dataset [32], they are not publicly available. For building a robust spoken dialogue system, it is essential to capture the intent and slots present in a human conversation. We manually annotate the real and natural utterances of the TRAINS corpus with intents and slots. The

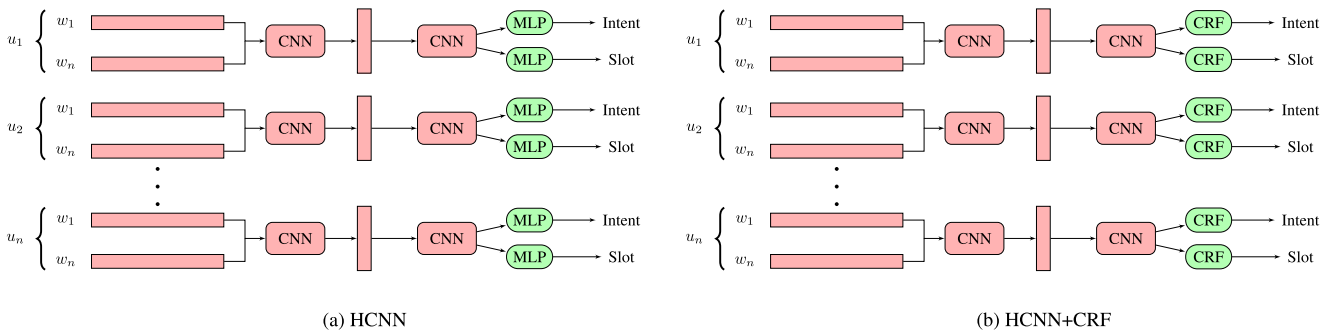


Fig. 3. Block diagram of our proposed hierarchical multi-task models; In (a), we employ hierarchical CNN followed by MLP while in (b), we use CRF as the final output layer.

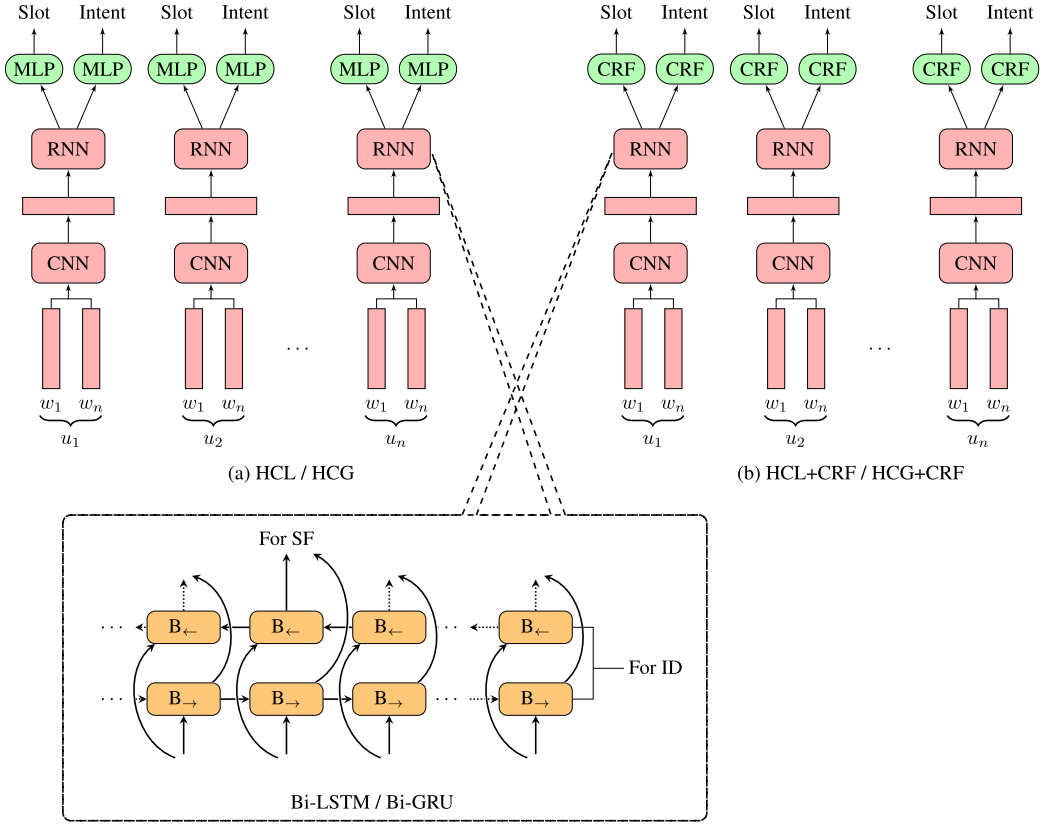


Fig. 4. Block diagram of our proposed multi-task models; we employ Bi-directional LSTM/GRU over the baseline CNN model for capturing contextual information, in (a) we use MLP as the output layer while in (b) we use CRF as the final output layer for predicting the intent and slots of an utterance.

corpus is a collection of problem-solving dialogues. The dataset has been annotated with 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 had an average of 15 turns. The corpus is a collection of multi-domain conversations dealing with hotel bookings. There are 20006 utterances in the training set and 6598 utterances in the test set. The dataset has been manually annotated with 24 intents and 136 slots.

Snips dataset: This dataset is collected from the Snips personal voice assistant, where the number of samples for each intent is approximately the same. The training set contains 13,084 utterances, and the test set includes 700 utterances. There are 72 slot labels and 7 intent types. Snips is more complicated mainly due to the intent of diversity and extensive vocabulary.

4.2. Training details

We use the python dependent package on neural network, Keras[4] for the implementation. In our work, the baseline CNN model has three convolutional layers with different filters followed by a max pooling layer. In the hierarchical CNN-CNN model, the Baseline CNN model is followed by another CNN having one convolutional layer. In the hierarchical CNN-RNN model, we use one layer of Bi-LSTM after the baseline CNN model. The number of neurons on the Bi-LSTM layer is fixed to be 200. Similarly, for the implementation of GRU, we use one layer of Bi-GRU. In each Bi-GRU layer, the number of neurons is fixed to be 200.

The outputs of the hierarchical models are fed either to the MLP or CRF layer. The outputs of MLPs predict intents and slots

[4] www.keras.io.

Table 4
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	Both
Word embedding size	100/200/300	300D each
Pooling	Max/Avg	Max
Dropout	0 – 0.5	0.15
Bidirectional	True/False	True
Learning rate	0.5 – 3	1.0
Number of filters	0 – 5	3
Hidden size	50 – 300	200

for each utterance. To capture label dependency, the outputs of hierarchical models are fed to CRF for intent detection and slot filling. The model uses a 600-dimensional combined word embedding formed using GloVe and Word2Vec. For training character embeddings we use CNN and uni-directional LSTM. To have a fair comparison of both of these models (CNN and LSTM) for character feature representation, we show the results of both in Table 5. For charCNN, we use one filter of size 2, while the number of neurons of the LSTM layer was fixed to be 100. The model uses 100-dimensional character embeddings as input to the model. We use the same 10-fold cross validation setup as in [43,44,49] for all the datasets. Instead of training the model on 9 folds and testing on 1 fold, we kept another 1 fold as the validation set to keep track of model training progress. The averaged intent accuracy and slot F1 score over the 10 folds are reported in Tables 5, 6, 7.

In our model, the intermediate layers use ReLU activation function while the MLP layer uses the softmax activation function. To avoid over-fitting of the neural network a useful regularisation technique known as dropout has been used in our model. At the time of forward propagation, the neurons are randomly tuned-off so that the convergence of weights is restricted to identical positions. For optimisation and regularisation, we use Adam optimiser along with 15% dropout in our model. Categorical cross-entropy is employed to update the model parameters. In our experiment, we have given equal weight to both the losses for the tasks, i.e. intent detection and slot filling. The reason being that we want our multi-task model to handle both the tasks with equal importance. By comprising one task can affect the other task as both the tasks are closely related and significant for the SLU module of a dialogue system hence the losses for the tasks are given the same weight. Table 4 lists the different parameters that we experimented with and the final chosen parameters of the proposed model.

5. Results and discussion

In this section, the experimental results along with necessary error analysis have been reported. We also provide a comparison of our multi-task hierarchical model with the baseline models. The effectiveness of our multi-task model has also been shown in contrast to individual models. Moreover, we provide a comparison of our model against the state-of-the-art approaches. We report accuracy and F1 score as the performance measure for intent detection and slot filling tasks respectively.

5.1. Results

Character embeddings have been very useful in capturing the semantic meaning of infrequent and out-of-vocabulary words. To compute the character feature vector we used both CNN and LSTM. Table 5 shows the results of different multi-task models using only character embeddings. From the table, it is evident that compared to LSTM, CNN helps in capturing the character feature vector better for both the intent detection and slot filling tasks. The probable reason for CNN outperforming LSTM in obtaining the character level features is mainly due to the ability of CNN to extract features more efficiently. In the rest of the experiments, we use the feature vectors obtained by CNN to represent character level features for all the models.

To capture the word level semantic representation of words we have used two pre-trained word embedding models, i.e., Glove and word2vec. Table 6 shows the results of different multi-task models by using different word embeddings. In our work, we have combined both the embeddings to give more semantic information to our models for the intent detection and slot filling tasks. From the table, it is visible that the combined word representation performs better in case of all the multi-task models compared to individual word embedding. In the rest of the experiments, we have used combined word embeddings along with character embeddings as input to the proposed multi-task models. By combining the information of both the pre-trained embedding model, Glove and word2vec provides extensive word level information to our multi-task models. Hence, our combined word embedding multi-task implementations, have higher performance than when we use the pre-trained embeddings individually.

In our proposed multi-task models, we provide both character level and word level features as input to the different implementations. In Table 7, we have given detailed results of all the models for the intent detection and slot filling tasks. It can be seen that the hierarchical multi-task models with CRF have outperformed the baseline CNN model along with hierarchical models without CRF. The hierarchical models help in capturing the contextual information thereby contributing more details for a particular utterance which in terms help in intent detection and slot filling. The hierarchical implementation of the multi-task models, adds additional information of the previous utterances and provides utterance dependency information at the time of classification. The hierarchical multi-task models with CRF as the final layer performs better in comparison to the MLP layer as the final task-specific layer.

The hierarchical encoders capture the dependency among utterances and hence provide information of the past utterances to the final layer for intent and slot prediction. Therefore, the use of CRF helps in capturing the information of the labels of the previous utterances along with the current utterance. In case of intent, the past information along with present utterance makes it a sequence and helps in identifying the intent of the present utterance. The entire information is fed to CRF which can capture these intricate details and help in the task. While for slot filling, an utterance has tags for every word and hence CRF shows better performance. As both the tasks are inter-related therefore, the improvement in the slot filling task helps in enhancing the performance of the intent detection task as well. The inherent property of CRF to capture label dependency majorly helps the slot filling task and shows slight improvement for the intent detection task of SLU. Experimental results in Tables 5–7 validate the fact that there is improvement using CRF as the final output layer. Hence, it can be concluded that bi-LSTM CRF based models show better improvement in comparison to just multi-layer RNN models for sequence labelling tasks. Hence, our proposed models can capture both utterance and label dependency on the assigned tasks. The hierarchical CNN model with CRF performs slightly better than the hierarchical CNN-RNN models with CRF. The possible reason is that CNN can extract the utterance features more efficiently in contrast to the RNN models.

To show the efficacy of the proposed approach we provide visualisation of the features learned by the model at the hidden

Table 5
Results on character embeddings for different proposed multi-task models.

Model	Embeddings	ATIS		TRAINS		FRAMES		SNIPS	
		Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
CNN	CharCNN	**92.67**	**89.16**	**70.89**	**86.33**	**56.43**	**61.74**	**88.27**	**79.34**
	CharLSTM	91.47	88.54	68.95	85.14	55.74	60.25	86.83	78.11
Hierarchical CNN (HCNN)	CharCNN	**93.82**	**91.63**	**74.54**	**88.19**	**59.37**	**65.43**	**90.45**	**84.66**
	CharLSTM	92.59	90.29	72.69	87.26	57.33	63.11	89.08	82.51
Hierarchical CNN - BiLSTM (HCL)	CharCNN	**93.54**	**91.27**	**73.87**	**87.65**	**58.82**	**64.87**	**89.86**	**83.49**
	CharLSTM	92.16	89.66	73.11	86.38	56.95	62.78	88.91	81.68
Hierarchical CNN - BiGRU (HCG)	CharCNN	**93.27**	**91.38**	**74.15**	**87.43**	**58.49**	**64.25**	**89.57**	**83.73**
	CharLSTM	91.89	90.21	73.61	86.11	56.62	63.14	88.95	81.93
Hierarchical CNN + CRF (CNN + CRF)	CharCNN	**94.37**	**92.77**	**76.25**	**90.54**	**61.33**	**69.71**	**93.13**	**87.13**
	CharLSTM	93.11	91.35	75.41	87.63	59.85	65.74	91.67	85.44
Hierarchical CNN - BiLSTM + CRF (HCL + CRF)	CharCNN	**93.85**	**92.08**	**75.88**	**89.82**	**60.85**	**68.23**	**92.85**	**86.52**
	CharLSTM	93.06	91.33	75.03	87.28	58.67	67.46	91.43	85.26
Hierarchical CNN - BiGRU + CRF (HCG + CRF)	CharCNN	**93.46**	**91.95**	**75.84**	**89.53**	**59.91**	**68.59**	**92.87**	**86.54**
	CharLSTM	92.78	91.68	75.23	87.24	58.33	67.82	91.69	85.77

Table 6
Word embedding results of proposed multi-task models.

Model	Embeddings	ATIS		TRAINS		FRAMES		SNIPS	
		Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
CNN	Glove (G)	94.33	91.96	77.63	91.51	68.25	75.63	91.62	85.11
	Word2Vec (W)	93.89	90.12	77.13	90.88	67.99	75.31	90.25	84.58
	Combined (G+W)	**95.02**	**92.42**	**79.88**	**94.73**	**71.51**	**78.36**	**93.86**	**89.77**
Hierarchical CNN (HCNN)	Glove (G)	96.51	94.33	80.17	93.64	71.88	80.54	93.96	89.96
	Word2Vec (W)	95.88	94.10	78.56	92.72	70.63	79.48	93.16	88.32
	Combined (G+W)	**97.68**	**95.42**	**81.69**	**95.21**	**73.64**	**84.81**	**95.46**	**90.32**
Hierarchical CNN - BiLSTM (HCL)	Glove (G)	96.33	94.52	79.63	92.86	70.54	79.51	93.61	88.99
	Word2Vec (W)	96.43	94.12	79.55	92.63	70.88	79.47	93.19	88.82
	Combined (G+W)	**97.14**	**95.21**	**81.21**	**94.56**	**72.89**	**83.88**	**95.11**	**89.73**
Hierarchical CNN - BiGRU (HCG)	Glove (G)	95.88	94.73	79.21	92.14	70.34	79.25	93.47	88.62
	Word2Vec (W)	95.27	94.58	79.03	92.18	70.52	79.69	93.05	88.47
	Combined (G+W)	**96.49**	**95.13**	**81.36**	**94.79**	**73.14**	**83.65**	**95.04**	**89.42**
Hierarchical CNN + CRF (HCNN + CRF)	Glove (G)	96.99	94.78	80.63	94.91	74.23	85.36	95.87	89.16
	Word2Vec (W)	96.54	95.32	80.45	95.10	73.83	85.11	95.42	88.96
	Combined (G+W)	**97.95**	**96.25**	**82.46**	**96.37**	**75.52**	**86.47**	**97.18**	**91.33**
Hierarchical CNN - BiLSTM + CRF (HCL + CRF)	Glove (G)	96.71	95.03	81.66	94.73	73.99	85.32	95.84	88.41
	Word2Vec (W)	96.47	94.93	81.11	94.52	73.25	85.06	95.29	88.23
	Combined (G+W)	**97.56**	**95.81**	**82.14**	**95.47**	**74.88**	**86.11**	**96.54**	**90.82**
Hierarchical CNN - BiGRU + CRF (HCG + CRF)	Glove (G)	96.85	94.76	81.24	95.20	74.83	85.33	95.59	88.23
	Word2Vec (W)	96.55	94.82	81.43	94.77	74.65	84.13	95.17	88.36
	Combined (G+W)	**97.27**	**95.61**	**82.32**	**95.69**	**75.21**	**85.79**	**96.23**	**90.45**

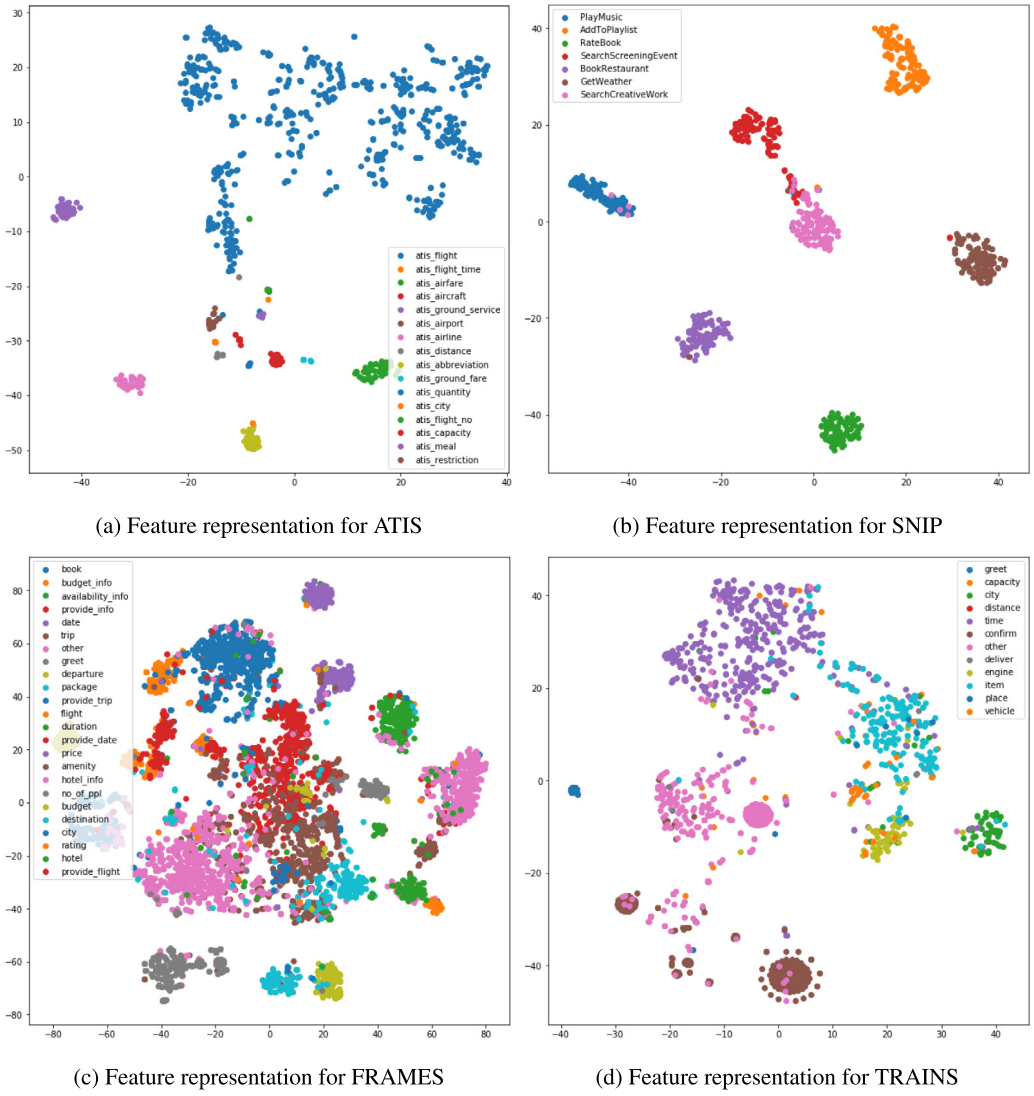
Table 7
Results of multi-task models with character embeddings (using CharCNN) and combined word embeddings.

Models	ATIS		TRAINS		FRAMES		SNIPS	
	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
CNN	95.96	93.89	80.46	96.32	73.48	81.36	95.48	90.88
HCNN	98.34	96.24	83.08	97.81	78.15	87.95	97.10	92.66
HCL	98.14	95.77	82.32	97.34	77.96	86.34	96.68	92.35
HCG	97.99	95.45	82.47	97.57	77.54	86.29	96.45	92.21
HCNN + CRF	**99.09**	**97.32**	**83.99**	**98.93**	**79.64**	**89.94**	**98.24**	**94.38**
HCL + CRF	**98.97**	**96.95**	**83.49**	**98.59**	**79.12**	**89.15**	**97.54**	**93.57**
HCG + CRF	**98.65**	**96.82**	**83.14**	**98.47**	**78.99**	**88.94**	**97.43**	**93.43**

layers for the intent detection task. As the number of slot labels is huge, hence we have not shown the feature representation for the slot filling task. The features produced by the hidden layers were of 128 dimensions which were reduced to 2-dimension using t-SNE [66]. In Fig. 5, it can be seen that the model has been able to learn the features for the respective intent labels in case of all the datasets. Labels belonging to a particular intent are closely grouped and are highly separable from the other labels in case of ATIS and SNIP datasets. The same intent labels in FRAMES and TRAINS dataset are also close to each other, but there is overlapping with other intent labels as shown in Fig. 5(c) and (d).

5.2. Multi-intent detection

In goal-oriented dialogue systems, users tend to ask questions about many things in a given utterance. For example in the given utterance, *"Would you like to increase your budget or adjust*



(a) Feature representation for ATIS

(b) Feature representation for SNIP

(c) Feature representation for FRAMES

(d) Feature representation for TRAINS

Fig. 5. Visualization of features learned by the model at the hidden layers for intent detection task.

Table 8
Results of multiple intents on FRAMES datasets.

Model	F-score	Hamming loss	Subset accuracy
CNN	0.58	0.047	0.42
HCNN	0.66	0.032	0.54
HCL	0.62	0.037	0.48
HCG	0.63	0.035	0.50

the dates or time of travel?", the intents are *"budget, date, time"*. Identifying multiple intents of the users helps in fulfilling the objectives of the user faster and providing better user experience, as it saves the user from repeating their requests several times. To identify multiple intents of an utterance we use the baseline CNN model and the hierarchical CNN and RNN models. FRAMES dataset has been annotated with multiple intents with an utterance having a maximum of three intents for a given sentence. The final output layer of the model is softmax which

Table 9
Multi-task model vs. individual model.

Model	Task	ATIS		TRAINS		FRAMES		SNIPS	
		Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
HCNN + CRF	Only ID	98.10	–	81.72	–	75.36	–	95.11	–
	Only SF	–	96.89	–	95.47	–	84.87	–	90.61
	MTM	**99.09**	**97.32**	**83.99**	**98.93**	**79.64**	**89.94**	**98.24**	**94.38**
HCL + CRF	Only ID	97.68	–	80.63	–	74.85	–	94.36	–
	Only SF	–	96.52	–	95.22	–	84.51	–	90.47
	MTM	**98.97**	**96.95**	**83.49**	**98.59**	**79.12**	**89.15**	**97.54**	**93.57**
HCG + CRF	Only ID	97.54	–	81.21	–	74.96	–	95.09	–
	Only SF	–	96.11	–	95.31	–	84.32	–	90.05
	MTM	**98.65**	**96.82**	**83.14**	**98.47**	**78.99**	**88.94**	**97.43**	**93.43**

Table 10
Comparison of proposed multi-task model with state-of-the-art models.

Model	ATIS		TRAINS		FRAMES		SNIPS	
	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
Attention BiRNN [Liu et. al., 2016 [47]]	98.21	95.98	62.35	82.66	60.20	85.84	96.71	83.90
Attention Encoder-Decoder NN [Liu et. al., 2016 [49]]	98.43	95.87	80.61	94.41	63.30	88.63	97.29	90.14
Bi-GRU [Zhang et. al., 2016 [48]]	98.32	96.89	79.85	94.67	62.88	87.95	97.38	91.48
Bi-Model with Decoder [Wang et. al., [53]]	98.99	96.89	81.41	95.29	64.17	88.36	97.65	92.46
Slot-Gated [Goo et. al., [54]]	94.10	95.20	75.66	81.44	59.42	78.36	97.00	88.80
Proposed HCL + CRF	98.97	96.95	83.49	98.59	79.12	89.15	97.54	93.57
Proposed HCG + CRF	98.65	96.82	83.14	98.47	78.99	88.94	97.43	93.43
Proposed HCNN + CRF	**99.09**	**97.32**	**83.99**	**98.93**	**79.64**	**89.94**	**98.24**	**94.38**

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

gives the probability values of the possible intent labels for a given utterance. In these experiments, in the final softmax layer, we consider the probabilities of the top three intent labels for any given utterance. We report subset accuracy, hamming loss and macro-average F1 score as the performance measure for the multi-intent detection task. Hamming loss [67] identifies the fraction of misclassified labels, hence lower the loss better is the model performance while subset accuracy [67] computes the fraction of the prediction being identical to the true label set. The results for these experiments are provided in Table 8. The evaluation shows that the hierarchical models perform well in comparison to the baseline CNN model. The F-score value for the hierarchical CNN model outperforms the other hierarchical models along with the baseline model. While the decrease in hamming loss denotes better performance of the proposed approach. The subset accuracy of the proposed hierarchical CNN model is better in comparison to the other hierarchical models.

5.3. Hierarchical multi-task models vs. hierarchical individual models

To check the effectiveness of the multi-task model (MTM) we compare it to the individual models. Using similar settings and parameters, we implement the individual models that detect intents and slots individually. In Table 9, we present the results of the multi-task and individual models. From the table, it is quite visible that the multi-task model performs better in comparison to the individual models. The fact that both the SLU tasks, namely intent detection and slot filling are inter-related and provides information to each other. This is evident from the evaluation results presented in Table 9. For all the datasets, it is observed that the multi-task model performs better than the individual models. The analysis of these models is provided in the next section.

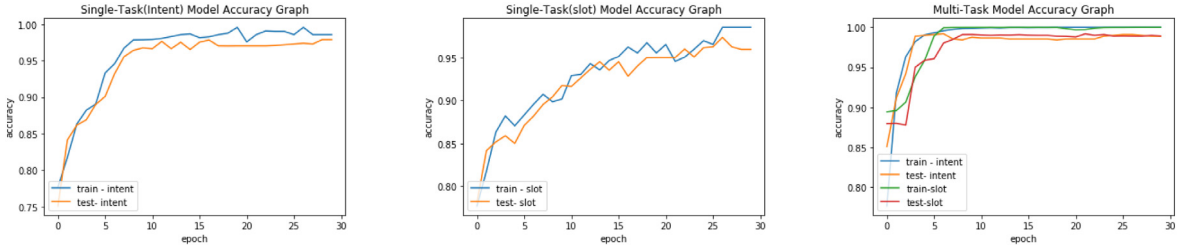
We provide the graphical comparison of the proposed multi-task hierarchical model to the individual task-specific models for all the datasets. The graph represents the accuracy vs. the number of epochs required. As it can been seen from Figs. 6,7,8,9, the multi-task model achieves higher accuracy with less number of epochs in comparison to the individual models. The individual models take more number of epochs to converge, and the performance is less than the proposed approach.

5.4. Proposed approach vs state-of-the-art approaches

In Table 10, we present the results of our proposed hierarchical multi-task model and the existing state-of-the-art models. From the results, we observe that our hierarchical CNN model with CRF outperforms the existing approaches in case of all the datasets. The difference in case of the ATIS dataset is not huge, because the existing approaches have already achieved high performance as this dataset has been studied for more than a decade. In the case of TRAINS and FRAMES datasets, we see that our proposed models perform very well compared to the previous approaches. Also, in the case of SNIPs dataset, we achieve better results for both the tasks of intent detection and slot filling.

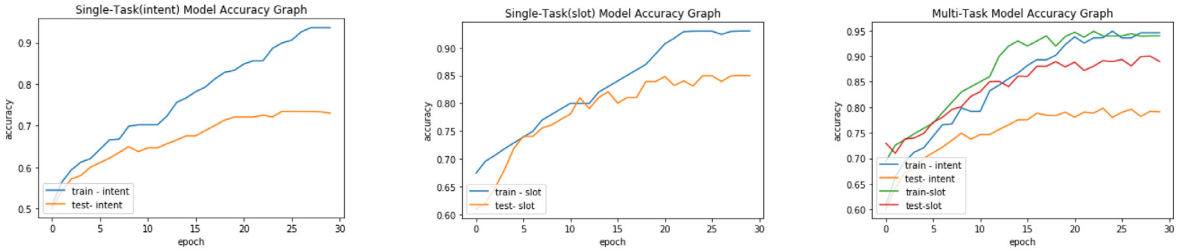
5.5. Error analysis

In this section, we provide a detailed analysis of the results presented in the previous section for both the SLU tasks of intent detection and slot filling. We have done both quantitative and qualitative analysis of the results to get a proper insight into the multi-task models for all the datasets. In case of ATIS dataset, we have presented the confusion matrix for intent detection task in Table 11. Already, the proposed approach has performed fairly well for the task but there have been few errors. This is mainly due to data skewness problem as majority



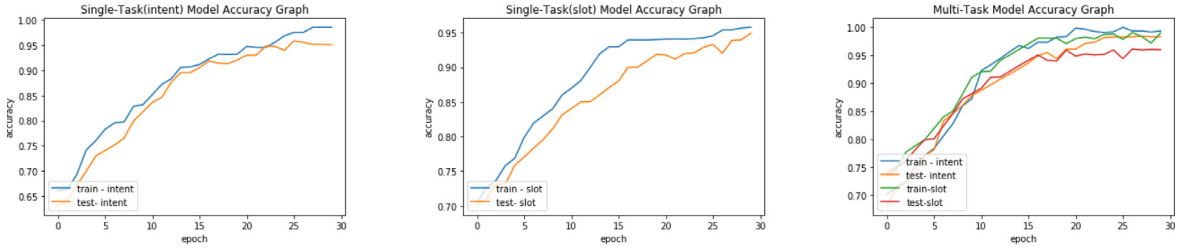
(a) Intent Model (b) Slot Model (c) Multi-task Model

Fig. 6. Graphical representation of the computation efficiency of the proposed hierarchical multi-task approach in comparison to the individual model for ATIS dataset.



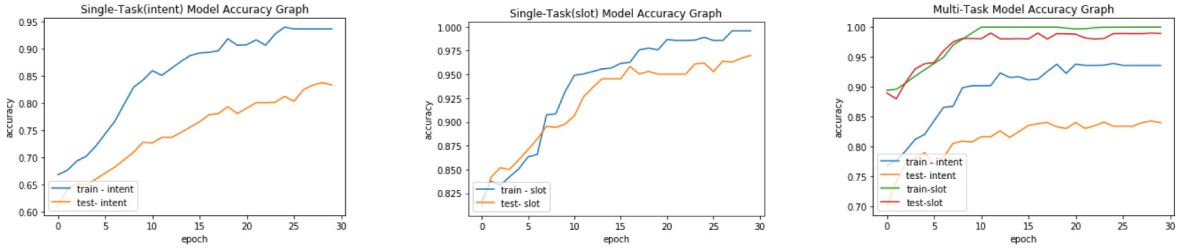
(a) Intent Model (b) Slot Model (c) Multi-task Model

Fig. 7. Graphical representation of the computation efficiency of the proposed hierarchical multi-task approach in comparison to the individual model for FRAMES dataset.



(a) Intent Model (b) Slot Model (c) Multi-task Model

Fig. 8. Graphical representation of the computation efficiency of the proposed hierarchical multi-task approach in comparison to the individual model for SNIP dataset.



(a) Intent Model (b) Slot Model (c) Multi-task Model

Fig. 9. Graphical representation of the computation efficiency of the proposed hierarchical multi-task approach in comparison to the individual model for TRAINS dataset.

of the utterance are labelled as *"flight"*. The intent labels such as *"day_name"*, *"restriction"* have very less representation in the dataset and has been miss-classified as *"flight"*. In case of slot filling task, there have been cases where the slot labels have been labelled as null tags. Mainly due to less number of slot tags in comparison to null tags. Few tags such as *"B-city_name"* has been miss-labelled as *"B-fromloc.city_name"* due to less number of instances of *"B-city_name"* in the training data. For example, in the utterance *"What is the ground transportation from Charlotte airport to downtown"*. Here the word *"Charlotte"* is incorrectly tagged as *"B-fromloc.city_name"* whereas it should have been *"B-city_name"*. In another example, *"Which airport is closest to Ontario California"*, the word *"California"* 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.

For TRAINS dataset, we present the confusion matrix for intent detection task in Table 12. The table shows the misclassification between different intent labels. The intent label *"Confirm"* has been misclassified with other utterances. Also, the labels *"Greet"*, *"Place"* has been confused with other intent labels in most cases.

Table 12
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

Table 13
Confusion matrix for intent detection of SNIPS dataset.

Correct-estimated	a	b	c	d	e	f	g
a. AddToPlaylist	84	0	0	0	1	0	0
b. BookRestaurant	0	124	0	0	0	0	0
c. GetWeather	0	0	80	0	0	0	0
d. PlayMusic	0	0	0	96	1	1	1
e. RateBook	1	0	0	1	87	2	1
f. SearchCreativeWork	0	1	0	1	1	96	1
g. SearchScreeningEvent	1	0	0	1	0	1	100

For example, the utterance *"Is the banana warehouse in Avon"* has been miss-classified as *"Distance"* whereas it should have been classified as *"Place"*. In another example, *"Why do not they take the trucks from Elmira"* has been incorrectly classified as *"Item"* whereas it should have been classified as *"Vehicle"*. Due to similar patterns between the utterances of both the intent labels, hence they have been confused during classification. Since the number of slot labels is less, therefore we get a good F-score. Due to easy patterns and smaller sentences, the slot labels have been identified correctly in the majority of utterances.

In Table 13, we present the confusion matrix for intent detection task of SNIP dataset. We achieve high accuracy mainly because the number of intent classes in this dataset is less. In the case of slot filling task, the complex nature of the dataset causes the miss-classification. Due to the large diversity in the meaning of the utterances, therefore, we get the errors in case of both the tasks. For example, the utterance *"Find a movie called living in America"* has been classified as *"SearchScreeningEvent"* but the actual intent label is *"SearchCreativeWork"*. Also, the words *"living in America"* has been tagged as *"B-movie_name I-movie_name I-movie_name"* whereas the actual tags of the words are *"B-object_name I-object_name I-object_name"*. Since the slot tags are highly dependent on the intent labels, hence the misclassification occurs in both the tasks.

To provide a detailed analysis for the FRAMES dataset, we have presented the confusion matrix for the intent detection task in Table 14. Due to the long length of utterances, many times the intent of the utterance is expressed in the latter part of the utterance causing miss-classification. For example, *"Wow that is very good I am keeping that one in mind, do you have anything in Frankfurt"* is miss-classified as *"Other"* but the actual label should be *"package"*. Also, in this model we handle single intents in an utterance; hence the utterances having multiple intents causes the miss-classification. 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 miss-classification 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 number and some slot tags have very less representation in the dataset. In the case of multi-intent classification, the model makes errors by identifying only a single intent. For example, *"How much is the trip to Fukuoka and what are the dates"*, the predicted label is only *"price"* while the actual labels are *"price, date"*. With the presence of some key intent words, the model has predicted multiple intents while the utterance had single intent. For example, *"could I see some better rated hotels"*, the intent label is *"hotel"* but the model has predicted both *"rating, hotel"* because of the presence of the word "rated" which often occurs in utterances having "rating" as the label.

To check the effectiveness of our multi-task model, we have analysed 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 have 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 in individual model. 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 was miss classified as *"B-city_name"* due to the help of the intent label *"airfare"*.

5.6. Statistical significance test

A statistical hypothesis test named Welch's t-test [68] 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 have calculated the p-values produced by Welch's t-test for comparison of two groups.

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

$$\mathcal{H}_0 : \lambda_\alpha = \lambda_\beta \tag{11}$$

Table 14
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

Table 15
Results of statistical significance test.

Model	ATIS		TRAINS		FRAMES		SNIPS	
	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
CNN	1.90E−060	6.04E−062	2.04E−062	1.77E−057	1.49E−071	5.33E−077	2.16E−058	2.82E−062
HCNN	2.20E−037	3.50E−043	1.90E−040	9.05E−044	2.30E−048	4.66E−053	4.76E−044	1.08E−050
HCL	3.93E−041	5.29E−049	3.27E−050	2.04E−049	2.62E−050	9.74E−063	4.16E−049	2.21E−053
HCG	1.78E−043	4.78E−052	1.09E−048	6.87E−047	6.20E−054	5.79E−063	2.45E−051	1.81E−054
HCL + CRF	3.22E−012	1.47E−026	4.42E−031	2.53E−025	1.11E−031	3.34E−038	2.68E−036	1.34E−038
HCG + CRF	3.85E−029	4.42E−031	2.31E−039	8.21E−030	7.05E−034	5.97E−042	1.34E−038	3.93E−041

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.

$$\mathcal{H}_1 : \lambda_\alpha > \lambda_\beta \tag{12}$$

where λ_k is the average accuracy of kth algorithm. In this study, we consider the alternative hypothesis($\mathcal{H}_1$) is that the proposed method is significantly better than the other baseline approaches. Hence disproving of the null hypothesis($\mathcal{H}_0$) establish the alternative hypothesis.

After computing the means of the two groups, we observed that the mean of the proposed method is always greater than the baselines. Hence, we have conducted the one-sided t-test. As we our conducting one-sided t-test, so there are only two possibilities which are that the proposed model is either better or equal to the baseline model. So, we have considered the alternative hypothesis to be better than the baseline while the null hypothesis to be equal to the baseline. Hence, both the hypothesis are mutually exclusive and complete with respect to the considered scenario.

Now the differences between the average accuracies is calculated by the following t statistic formula:

$$t = \frac{\bar{\chi_1} - \bar{\chi_2}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{13}$$

where χ_i, σ_i^2 and n_i are the mean, variance and size of ith sample, respectively. p-value is the probability, under the assumption of the null hypothesis($\mathcal{H}_0$) and smaller p-value is a strong evidence against null hypothesis($\mathcal{H}_0$). For the statistical test on all the datasets, we execute the experiment 20 times. Table 15 reports p-values produced by Welch's t-test. All the p-values reported in Table 15 are less than 0.05(5% significance level). Hence it shows that our approach is statistically significant.

While in two-sided t-test there can be three possibilities, firstly the mean of the proposed method being equal to the baselines. Secondly, the mean of the proposed method being less than the baselines and thirdly the mean of the proposed method being greater than the baseline. In our case we observe the third condition, which thereby motivates us to perform one-sided t-test for proving the statistical significance of our proposed method. In our case, the second and the third conditions are complementary i.e. if the second condition is true it automatically infers that the third condition is false and vice versa. Hence in this regard, proving of our alternative hypothesis (i.e. proposed method is better than the existing methods) is sufficient to prove the third condition and eliminate the second condition.

6. Conclusion and future work

In this paper, we have proposed a hierarchical multi-task model for the two significant tasks of SLU, i.e., slot filling and intent detection. For the hierarchical multi-task model, we have used the convolutional neural network and recurrent neural network with LSTM and GRU as basic cells. The representations learned from these models are shared by both intent and slot filling task. In our proposed approach, we have captured the contextual information that helps in modelling utterance dependency with the help of the hierarchical nature of our approach. Also, label dependency has been considered by employing CRF as the final layer of our approach. Experiments were conducted

on four datasets. The hierarchical multi-task model exhibits advantages over individual models and outperforms the state-of-the-art methods on slot filling and intent detection tasks on all the datasets irrespective of the domain and nature of these datasets. By using character embedding and combined word embeddings, it further helps our model to identify the intents and slots correctly.

In our future work, we aim to incorporate semantic information as well to model these SLU tasks. Furthermore, we would like to employ different deep learning techniques such as memory networks, pointer networks and autoencoders to capture the contextual information for the dialogue datasets.

Acknowledgments

Asif Ekbal acknowledges the Young Faculty Research Fellowship (YFRF), supported by Visvesvaraya Ph.D. 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).

References

[1] P.J. Price, 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, 1990.
[2] R. López-Cózar, Z. Callejas, Two-level speech recognition to enhance the performance of spoken dialogue systems, Knowl.-Based Syst. 19 (3) (2006) 153–163.
[3] A.L. Gorin, G. Riccardi, J.H. Wright, How may i help you? Speech Commun. 23 (1–2) (1997) 113–127.
[4] R. López-Cózar, Z. Callejas, D. Griol, Using knowledge of misunderstandings to increase the robustness of spoken dialogue systems, Knowl.-Based Syst. 23 (5) (2010) 471–485.
[5] P. Haffner, G. Tur, J.H. Wright, Optimizing svms for complex call classification, in: Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on, Vol. 1, IEEE, 2003, p. I.
[6] G. Tur, Model adaptation for spoken language understanding, in: Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on, Vol. 1, IEEE, 2005, pp. I–41.
[7] G. Tur, D. Hakkani-Tür, L. Heck, S. Parthasarathy, Sentence simplification for spoken language understanding, in: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, IEEE, 2011, pp. 5628–5631.
[8] G. Tur, Multitask learning for spoken language understanding, in: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, Vol. 1, IEEE, 2006, p. I.
[9] D. Hakkani-Tür, G. Tur, A. Chotimongkol, 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.
[10] S. Yaman, L. Deng, D. Yu, Y.-Y. Wang, A. Acero, An integrative and discriminative technique for spoken utterance classification., IEEE Trans. Audio Speech Process. 16 (6) (2008) 1207–1214.
[11] J.-K. Kim, G. Tur, A. Celikyilmaz, B. Cao, Y.-Y. Wang, Intent detection using semantically enriched word embeddings, in: Spoken Language Technology Workshop (SLT), 2016 IEEE, IEEE, 2016, pp. 414–419.
[12] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[13] Y. Luan, S. Watanabe, B. Harsham, Efficient learning for spoken language understanding tasks with word embedding based pre-training, in: Sixteenth Annual Conference of the International Speech Communication Association, 2015.
[14] J. Wu, Introduction to convolutional neural networks, in: National Key Lab for Novel Software Technology, Nanjing University, China, 2017.
[15] H.B. Hashemi, A. Asiaee, R. Kraft, Query intent detection using convolutional neural networks, in: International Conference on Web Search and Data Mining, Workshop on Query Understanding, 2016.
[16] S.V. Ravuri, A. Stolcke, Recurrent neural network and lstm models for lexical utterance classification., in: INTERSPEECH, 2015, pp. 135–139.
[17] S. Ravuri, A. Stoicke, A comparative study of neural network models for lexical intent classification, in: Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on, IEEE, 2015, pp. 368–374.
[18] M. Firdaus, S. Bhatnagar, A. Ekbal, P. Bhattacharyya, Intent detection for spoken language understanding using a deep ensemble model, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2018, pp. 629–642.
[19] P. Jayarao, A. Srivastava, Intent Detection for code-mix utterances in task oriented dialogue systems, arXiv preprint arXiv:1812.02914, 2018.
[20] C. Xia, C. Zhang, X. Yan, Y. Chang, P.S. Yu, Zero-shot user intent detection via capsule neural networks, EMNLP, 2018.
[21] R. Masumura, T. Tanaka, R. Higashinaka, H. Masataki, Y. Aono, Multi-task and multi-lingual joint learning of neural lexical utterance classification based on partially-shared modeling, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 3586–3596.
[22] R. Masumura, Y. Shinohara, R. Higashinaka, Y. Aono, Adversarial training for multi-task and multi-lingual joint modeling of utterance intent classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 633–639.
[23] A. McCallum, D. Freitag, F.C. Pereira, Maximum entropy Markov models for information extraction and segmentation, in: ICML, Vol. 17, (2000) 2000, pp. 591–598.
[24] C. Raymond, G. Riccardi, Generative and discriminative algorithms for spoken language understanding, in: Eighth Annual Conference of the International Speech Communication Association, 2007.
[25] A. Moschitti, G. Riccardi, C. Raymond, Spoken language understanding with kernels for syntactic/semantic structures, in: Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on, IEEE, 2007, pp. 183–188.
[26] M. Jeong, G.G. Lee, Exploiting non-local features for spoken language understanding, in: Proceedings of the COLING/ACL on Main Conference Poster Sessions, Association for Computational Linguistics, 2006, pp. 412–419.
[27] R. López-Cózar, Using knowledge on word-islands to improve the performance of spoken dialogue systems, Knowl.-Based Syst. 88 (2015) 223–243.
[28] A. Deoras, R. Sarikaya, Deep belief network based semantic taggers for spoken language understanding., in: Interspeech, 2013, pp. 2713–2717.
[29] G. Mesnil, X. He, L. Deng, Y. Bengio, Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding., in: Interspeech, 2013, pp. 3771–3775.
[30] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, et al., Using recurrent neural networks for slot filling in spoken language understanding, IEEE/ACM Trans. Audio Speech Lang. Process. 23 (3) (2015) 530–539.
[31] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, D. Yu, Recurrent neural networks for language understanding., in: Interspeech, 2013, pp. 2524–2528.
[32] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, F. Gao, Recurrent conditional random field for language understanding, in: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE, 2014, pp. 4077–4081.
[33] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, Y. Shi, Spoken language understanding using long short-term memory neural networks, in: Spoken Language Technology Workshop (SLT), 2014 IEEE, IEEE, 2014, pp. 189–194.
[34] L. Deng, G. Tur, X. He, D. Hakkani-Tur, Use of kernel deep convex networks and end-to-end learning for spoken language understanding, in: Spoken Language Technology Workshop (SLT), 2012 IEEE, IEEE, 2012, pp. 210–215.
[35] S. Zhu, K. Yu, Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 5675–5679.
[36] L. Zhao, Z. Feng, 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 (Volume 2: Short Papers), Vol. 2, 2018, pp. 426–431.
[37] Y. Shin, K.M. Yoo, S.-g. Lee, Slot filling with delexicalized sentence generation, in: Proc. Interspeech 2018, 2018, pp. 2082–2086.
[38] L. Qiu, Y. Ding, L. He, Recurrent Neural Networks with Pre-trained Language Model Embedding for Slot Filling Task, arXiv preprint arXiv:1812.05199, 2018.
[39] J. Wu, R.E. Banchs, L.F. D'Haro, P. Krishnaswamy, N. Chen, Attention-based semantic priming for slot-filling, in: Proceedings of the Seventh Named Entities Workshop, 2018, pp. 22–26.
[40] O. Lan, S. Zhu, K. Yu, Semi-supervised training using adversarial multi-task learning for spoken language understanding, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 6049–6053.
[41] B. Liu, I. Lane, Multi-domain adversarial learning for slot filling in spoken language understanding, arXiv preprint arXiv:1711.11310, 2017.
[42] S. Zhu, K. Yu, Concept Transfer Learning for Adaptive Language Understanding, SIGDIAL, 2018.
[43] M. Jeong, G.G. Lee, Triangular-chain conditional random fields, IEEE Trans. Audio Speech Lang. Process. 16 (7) (2008) 1287–1302.

[44] P. Xu, R. Sarikaya, Convolutional neural network based triangular crf for joint intent detection and slot filling, in: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on, IEEE, 2013, pp. 78–83.
[45] D. Guo, G. Tur, W.-t. Yih, G. Zweig, Joint semantic utterance classification and slot filling with recursive neural networks, in: Spoken Language Technology Workshop (SLT), 2014 IEEE, IEEE, 2014, pp. 554–559.
[46] Y. Shi, K. Yao, H. Chen, Y.-C. Pan, M.-Y. Hwang, B. Peng, Contextual spoken language understanding using recurrent neural networks, in: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, IEEE, 2015, pp. 5271–5275.
[47] B. Liu, I. Lane, Joint Online Spoken Language Understanding and Language Modeling with Recurrent Neural Networks, SIGDIAL, 2016.
[48] X. Zhang, H. Wang, A joint model of intent determination and slot filling for spoken language understanding., in: IJCAI, 2016, pp. 2993–2999.
[49] B. Liu, I. Lane, Attention-based recurrent neural network models for joint intent detection and slot filling, in: INTERSPEECH, 2016.
[50] D. Hakkani-Tür, G. Tür, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, Y.-Y. Wang, Multi-domain joint semantic frame parsing using bi-directional rnn-lstm., in: INTERSPEECH, 2016, pp. 715–719.
[51] Y.-B. Kim, S. Lee, K. Stratos, Onenet: Joint domain, intent, slot prediction for spoken language understanding, in: Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE, IEEE, 2017, pp. 547–553.
[52] A. Bapna, G. Tur, D. Hakkani-Tur, L. Heck, Sequential dialogue context modeling for spoken language understanding, SIGDIAL, 2017.
[53] Y. Wang, Y. Shen, H. Jin, 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, Volume 2 (Short Papers), Vol. 2, 2018, pp. 309–314.
[54] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, Y.-N. Chen, 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, Volume 2 (Short Papers), Vol. 2, 2018, pp. 753–757.
[55] M. Firdaus, S. Bhatnagar, A. Ekbal, P. Bhattacharyya, A deep learning based multi-task ensemble model for intent detection and slot filling in spoken language understanding, in: International Conference on Neural Information Processing, Springer, 2018, pp. 647–658.
[56] S. Upadhyay, M. Faruqui, G. Tür, H.-T. Dilek, L. Heck, (Almost) zero-shot cross-lingual spoken language understanding, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 6034–6038.
[57] J.-S. Kim, J. Kim, S. Park, K. Lee, Y. Lee, Modeling with recurrent neural networks for open vocabulary slots, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 2778–2790.
[58] C. Zhang, Y. Li, N. Du, W. Fan, P.S. Yu, Joint Slot Filling and Intent Detection via Capsule Neural Networks, arXiv preprint arXiv:1812.09471, 2018.
[59] A. Siddhant, A. Goyal, A. Metallinou, Unsupervised Transfer Learning for Spoken Language Understanding in Intelligent Agents, AAAI, 2019.
[60] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, 2015, pp. 649–657.
[61] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[62] A. Karpathy, J. Johnson, L. Fei-Fei, Visualizing and understanding recurrent networks, arXiv preprint arXiv:1506.02078, 2015.
[63] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, EMNLP, 2014.
[64] P.A. Heeman, J.F. Allen, The TRAINS 93 Dialogues., Technical Report, ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE, 1995.
[65] L.E. Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, K. Suleman, Frames: A corpus for adding memory to goal-oriented dialogue systems, SIGDIAL, 2017.
[66] L.v.d. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.
[67] E. Spyromitros, G. Tsoumakas, I. Vlahavas, An empirical study of lazy multilabel classification algorithms, in: Hellenic Conference on Artificial Intelligence, Springer, 2008, pp. 401–406.
[68] B.L. Welch, The generalization ofstudent's' problem when several different population variances are involved, Biometrika 34 (1/2) (1947) 28–35.