Information Fusion 91 (2023) 299–315

Contents lists available at ScienceDirect

Information Fusion

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

Full length article

Multitask learning for multilingual intent detection and slot filling in dialogue systems

Mauajama Firdaus [a], Asif Ekbal [a], Erik Cambria [b,*]

[a] Department of Computer Science and Engineering, Indian Institute of Technology Patna, India
[b] School of Computer Science Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:
Multitask learning
Multilingual analysis
Information fusion
Intent detection
Slot filling
Deep learning

ABSTRACT

Dialogue systems are becoming an ubiquitous presence in our everyday lives having a huge impact on business and society. Spoken language understanding (SLU) is the critical component of every goal-oriented dialogue system or any conversational system. The understanding of the user utterance is crucial for assisting the user in achieving their desired objectives. Future-generation systems need to be able to handle the multilinguality issue. Hence, the development of conversational agents becomes challenging as it needs to understand the different languages along with the semantic meaning of the given utterance. In this work, we propose a multilingual multitask approach to fuse the two primary SLU tasks, namely, intent detection and slot filling for three different languages. While intent detection deals with identifying user's goal or purpose, slot filling captures the appropriate user utterance information in the form of slots. As both of these tasks are highly correlated, we propose a multitask strategy to tackle these two tasks concurrently. We employ a transformer as a shared sentence encoder for the three languages, i.e., English, Hindi, and Bengali. Experimental results show that the proposed model achieves an improvement for all the languages for both the tasks of SLU. The multi-lingual multi-task (MLMT) framework shows an improvement of more than 2% in case of intent accuracy and 3% for slot F1 score in comparison to the single task models. Also, there is an increase of more than 1 point intent accuracy and 2 points slot F1 score in the MLMT model as opposed to the language specific frameworks.

1. Introduction

Advancements in Artificial Intelligence (AI) have led to the development of intelligent agents that can converse with humans and assist them in their daily tasks. Thus, language understanding and generation will be important for making the lives of people easier in the future. An increase in availability of spoken language understanding (SLU) technologies in smartphones and personal assistants like Apple's Siri, Amazon's Alexa, Microsoft's Cortana, etc., has inspired an in-depth investigation on understanding the language of the user. With the progress in technology, the forthcoming generations will be highly dependent on virtual assistants hence it is imperative to make the agent capable of understanding the user to assist them to achieve their specified objectives [1,2]. For every dialogue system, the primary target is to provide user satisfaction by helping users reach their desired goals [3–6]. In this process, knowing the user's purpose and supplying them with insightful responses is essential [7]. The dialogue system is an example of human–computer communication, and it includes various modules focused on user comprehension and generating responses to help the user achieve their intended goal.

The primary modules of every conversational agent are identifying the intents and slot filling. Understanding the utterance is itself a difficult task, as the system needs to understand the intended meaning and extract the necessary information from the utterances of the user. The task of identifying the meaning or objective (either implicit or explicit) of the user utterance is defined as intent detection (ID). In contrast, the task of extracting information in the form of slots is called slot filling (SF). With the enhanced usage of personal assistants, the research and development of human–machine interaction have increased rapidly over the decade [8–10]. For the widespread application of chatbots, personal assistant should be able to understand different languages spoken by the user. It reduces the language barrier and assists the users to achieve their cherished goals. Developing multilingual dialogue systems is challenging, as the system needs to understand the syntax, semantics, and patterns for each language. In order to deal with the low-resource languages, the sharing of information across the languages is essential as it captures intricate details among the related languages.

* Corresponding author.
E-mail addresses: mauajama.pcs16@iitp.ac.in (M. Firdaus), asif@iitp.ac.in (A. Ekbal), cambria@ntu.edu.sg (E. Cambria).

https://doi.org/10.1016/j.inffus.2022.09.029
Received 29 October 2021; Received in revised form 27 September 2022; Accepted 30 September 2022
Available online 21 October 2022

Table 1
Intent examples in different languages.

Utterance	Intent	Language
Show me the flights arriving on Baltimore on June fourteenth.	Flight	English
अन्य शहरों के माध्यम से बोस्टन से वाशिंगटन डीसी के लिए कौन सी एयरलाइनें उड़ान भरती हैं?? (Anya shehro ke madhyam se Boston se Washington D.C. ke liye kaun si airlines udaan bharti hai?) (Which airlines fly from Boston to Washington DC via other cities?)	Airlines	Hindi
আমি বোস্টন থেকে ফিলাডেলফিয়ার কম খরচে ভাড়া পেতে চাই?? (Aami Boston theke Philadelphia kom kharcha bhada pete chai?) (I'd like to find the least expensive fare from Boston to Philadelphia?)	Airfare	Bengali
I want to listen to seventies music	PlayMusic	English
मुझे पड़ोस में एनिमेटेड फिल्मों के लिए समय दिखाएं। (Mujhe pados mein animated filmo ke liye samay dikhaye) (Find me showtimes for animated movies in the neighborhood.)	SearchScreeningEvent	Hindi
ভেনিজুয়েলা দেশে ফুরেনের মতো আবহাওয়া কেমন?? (Venezuela desera Frewen śaharē ābahāōyā kēmana?) (What is the weather like in the city of Frewen in the country of Venezuela?)	GetWeather	Bengali

Table 2
Examples of multiple slots for different languages.

Utterance	Slots	Language
Play a chant by MJ Cole	O O B-music_item O B-artist I-artist	English
इस उपन्यास को 5 स्टार दें। (Iss upanyaas ko 5 star den.) (Give this novel 5 stars.)	O B-object_type O B-rating_value B-rating_unit O	Hindi
সিনেমা সময়সূচী প্রদর্শন করুন। (Cinema samaychuchi pradarshan karun.) (Show movie schedules.)	B-object_type I-object_type O O	Bengali
Chicago to Milwaukee	B-fromloc.city_name O B-toloc.city_name	English
मुझे डलास से डेल्टा उड़ानें दिखाएं। (Mujhe Dallas se Delta udaane dikhaye.) (Show me Delta flights from Dallas.)	O B-fromloc.city_name O B-airline_name O O	Hindi
কি প্লেন ইউনাইটেড ব্যবহার করে?? (Ki plena unaiteda byabahara kare?) (What planes does United use?)	O O B-airline_name O O	Bengali

Recent works on multilingual SLU for the task of utterance classification [11,12] have opened new frontiers for more in-depth investigation in multilinguality for the dialogue systems. In this work, we focus on developing an end-to-end SLU module that can jointly identify the intent and necessary slots for different languages. We investigate developing a multitask model for intent detection and slot filling for English, Hindi, and Bengali languages. We show examples of different intents for the different languages in Table 1. The examples of multiple slots in an utterance for different languages are presented in Table 2. As both the tasks are related, hence in this work, we model intents and slots together in a unified framework. Multitask learning has shown improved performance for the various tasks [13–15]. Inspired by these recent works, we propose a multitask SLU framework for simultaneously modeling intent and slots present in an utterance for different languages.

1.1. Problem definition

This paper addresses two vital and crucial tasks of any conversational agent, namely: identifying intents and slot filling.

1.1.1. Intent detection

In any goal-oriented dialogue system, the main objective is to automatically identify the user's intention represented in natural language. This a difficult natural language processing (NLP) task called intent detection or intention awareness [16], which goes beyond the explicit content of the dialogue message.

The main goal is to label the user utterance x, comprising of a sequence of words $x = (x_1, x_2, \ldots, x_T)$ into one of the N predetermined set of intent classes, y_i, depending upon the given utterance such that:

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

For better understanding, Table 1 displays a few instances of different intents associated with diverse domains and varied languages.

1.1.2. Slot filling

Semantic constituent extraction from an input utterance is termed as slot filling. It includes filling in the values in a semantic frame for a predetermined set of slots. The purpose of slot filling is to allocate semantic tags to each word in the utterance. Having a sentence x consisting of a sequence of words $x = (x_1, x_2, \ldots, x_T)$, the purpose of a slot filling task is to find a set of semantic classes $s = (s_1, s_2, \ldots, s_T)$ for each word in the sentence, such that:

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

The slot labels for different utterances in varied languages are presented in Table 2, according to the IOB representation.[1] Slot filling is viewed as a sequence labeling task because a word's slot relies on the previous words. Therefore, it is necessary to capture the information present in the entire sequence to identify the appropriate slots.

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

1.2. Motivation and contribution

The motivation for taking up this task is to build an interactive goal-oriented dialogue system that can handle user inputs in different languages, thereby providing its application suitable for multilingual information access. In our current study, we mainly focus on English, Bengali, and Hindi. In Tables 1 and 2, we show the multilingual examples of user utterances from the different datasets used in our task.

The motive for employing this setup was to provide the flexibility of sharing the information between the languages and improve the performance of the overall model by capturing the different information simultaneously. Intent detection corresponds to classifying each utterance in a dialogue. This is regarded as comparatively a less complicated task than the other semantic analysis tasks. Still, the errors made by the intent detector are more apparent because they lead to incorrect system responses. Therefore, a reliable intent detection system plays a vital role in developing an efficient dialogue system. Although a chatbot usually follows a pipelined structure where a sentence's intention is detected, then the slots are obtained, but this method can introduce errors if the intention is not correctly identified. These tasks are strongly interrelated with each other, and one's knowledge can aid in solving another. This is the primary motivation for building a consolidated architecture that can handle both tasks together. Hence, we create an end-to-end SLU module for any task-oriented chatbot by performing intent detection and slot filling together.

This paper develops a multilingual enabled multitask model that can simultaneously identify the intents and capture the slots from any input utterance belonging to any language. The capability of a machine to handle multiple languages is essential for its widespread applications. Users tend to communicate more in their preferred language, enabling a system to understand different languages is essential and challenging. One of the desired goals of AI is to build dialogue systems to establish effective communication facilitating the agent to understand different languages that humans choose to communicate.

The key contributions of this work are as follows:

- We propose a multilingual multitask framework[2] for intent detection and slot filling using an attentive BERT architecture.
- We construct a benchmark corpus[3] for the SLU tasks, i.e., for intent detection and slot filling on ATIS, TRAINS, SNIPs, and FRAMES datasets for Hindi and Bengali to capture more meaningful and realistic statements spoken by the speakers in a human-machine dialogue system.

The rest of this article is structured as follows: Section 2 presents a brief study on the related works; Section 3 describes our proposed approach; Section 4 briefly introduces the different datasets and discusses the data preparation for Hindi and Bengali languages; Section 5 presents the implementation details, while Section 6 provides a detailed analysis of the experimental results; finally, Section 7 presents the concluding comments and directions for future studies.

2. Related work

Understanding the user utterance is a significant and essential component of every dialogue system [17]. Intent detection and slot filling are the critical tasks of SLU that aim to extract semantic meanings from the user utterances to assist users achieve their goals. The SLU research has originated from the ATIS project [18], and the call classification systems [19]. Detection of intent and slot filling was performed both in isolation and together in the past. We provide a summary of the works that have already been conducted on these SLU tasks.

[2] https://github.com/senticnet/MLMT.

[3] https://sentic.net/SLU.zip.

2.1. Intent detection

Previously different traditional machine learning approaches such as support vector machine (SVM) [20] and boosting techniques [21] were employed for identifying intents in the user utterance. In [22], syntactic and semantic graphs (SSG) representing the various properties of an utterance were used for intent detection. The authors in [23] used the ATIS corpus to detect intent with maximum entropy classifiers. With advancements in artificial intelligence, deep learning techniques have shown a promising direction in solving various NLP tasks. In [16], computational analysis and tracking of semantic and affective information associated with human actors' intentions were applied to minimize miscommunication and uncertainty in time-sensitive and information-saturated situations. In [24], Convolutional Neural Networks (CNN) [25] was used to detect the intentions of a user search query. Recurrent neural networks (RNN) [26] with long short term memory [27] have also been employed for identifying the intents of an utterance in [28,29]. An ensemble framework using CNN, Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) was investigated for the task of intent detection on the ATIS dataset in [30]. In [31], the authors suggested varying machine learning and deep learning models for a code-mixed dataset in Hindi and English with different vector representations of words. Capsule neural networks have recently been used in [32] to identify intents on the SNIPs dataset. The authors in [11,12] suggested a multitask adversarial structure for English and Japanese datasets to detect the intents of a user utterance in the form of domain-type, questions, and dialogue act. The authors in [33] employed bidirectional LSTM with margin loss to identify the unknown user intent. The authors in [34] devised a Gaussian mixture model for handling unknown intents. For handling multiple intents in a user utterance in a multilingual scenario, the authors [35] employed a multilingual attention framework.

In [36], the authors proposed a novel acoustics-based intent recognition system that uses discovered phonetic units for intent classification. The authors in [37] leverage BERT-style pairwise encoding to train a binary classifier that estimates the best-matched training example for user input. Also, they propose to boost the discriminative ability by transferring a natural language inference (NLI) model for identifying the intents. For identifying multiple intents, [38] utilize a universal thresholding experience on data-rich domains and then adapt the thresholds to certain few-shot domains with a calibration based on non-parametric learning. A novel semantic matching and aggregation network where semantic components are distilled from the utterances via multihead self-attention with additional dynamic regularization constraints was proposed in [39] for identifying the intents. A multiview clustering mechanism was employed for dialogue intent induction in [40]. The authors in [41] proposed a SofterMax and deep novelty detection (SMDN), a simple yet effective post-processing method for detecting unknown intent in dialogue systems based on pre-trained deep neural network classifiers. For faster intent classification BranchyNet scheme [42] has been designed.

2.2. Slot filling

As mentioned above, slot filling is a sequence labeling problem in which a tag is allocated to each word of the utterance. Factorized probabilistic models such as Maximum Entropy Markov Model (MEMM) [43] and Conditional Random Field (CRF) [44] were used for slot filling to solve the label bias problem with locally normalized models. In [45], SVM was used for slot filling together with syntactic features captured by syntactic and semantic tree kernels. The authors suggested a method in [46] that considered both parts of semantic frame information and word understanding to improve the efficiency of spoken dialogue systems.

Several deep learning architectures such as deep belief networks (DBN) [47], deep convex networks (K-DCN) [48] and RNN using LSTM [49–51] have also been employed for extracting essential information in the form of slots from a given utterance. An attention mechanism in the RNN framework was employed for slot filling on the ATIS dataset in [52–54]. The authors in [55] presented the sequence-to-sequence model-based generative network, including the pointer network for slot filling. A pre-trained language model for identifying the slots was used in [56]. The adversarial learning framework has also been investigated for identifying the slots in [57,58]. The authors suggested the idea of transfer learning for the task of slot filling in [59]. The authors in [60] explored the usage of lexicons for slot tagging.

In [61], the authors introduced MultiATIS++ that extends the ATIS dataset for 9 more languages for the task of slot filling. The authors in [62] designed an Attention-Informed Mixed-Language Training (MLT), a novel zero-shot adaptation method for cross-lingual task-oriented dialogue systems. In [63], a Cluster-to-Cluster generation framework for Data Augmentation was proposed for identifying the correct slots on ATIS and SNIPS dataset.

2.3. Joint models for intent detection and slot filling

Intent detection and slot filling tasks are highly correlated, and previously some deep learning models have been constructed for modeling both the SLU tasks together. Formerly, CNN-based triangular CRF [64] and recursive neural network (RecNN) [65] have been proposed for jointly modeling intent and slots. Several RNN frameworks employing LSTM [66,67], and GRU [68] have been devised for both the tasks. Step n-gram model, along with RNN and CNN, was used to formulate both the tasks in [69]. Bi-directional attention-based RNNs have also been implemented to collectively address the task of slot filling and intent detection in [70]. In [71], word and character embeddings were taken as input to the neural framework for jointly identifying the domain, intent, and slots in a given utterance. In [72], sequential context modeling using RNN for the SLU task has been explored.

For intent detection and slot filling in [73], the authors proposed a bi-model network employing RNN. The authors in [74] implemented an attention mechanism that focused on studying the connection between the intents and slot vectors to model the tasks jointly. A multitask ensemble model using combined word embeddings as input to the neural models was presented in [75]. A zero-shot learning framework for two new languages (Hindi and Turkish) was introduced in [76]. An attention mechanism with position information was considered in [77]. Capsule neural networks for jointly modeling intent and slot were investigated in [78]. For faster and efficient pre-training of the SLU module, the ELMo-Light model was designed in [79]. The authors employed a stack-propagation framework for both the tasks in [80]. The self-attention framework has been used in [81], and data augmentation with data noising for both the tasks has been explored in [82]. The identification of multiple intents along with slots has been proposed in [83]. In recent times, the BERT framework has been investigated for jointly identifying the intent and slots of an utterance [84,85]. In [86], direct connections between the two tasks were established for mutually promoting the performance of both the SLU tasks. A hierarchical framework for context modeling was employed in [35] for the multitask learning of both tasks. A multitask framework for both the SLU tasks and dialogue logistic inference was considered in [87].

The authors in [88] designed a novel two-pass iteration mechanism to handle the problem of the uncoordinated slots caused by conditional independence of non-autoregressive model based upon the Transformer network on the widely used ATIS and Snip dataset. Lately, in [89], an Adaptive Graph-Interactive Framework (AGIF) for joint multiple intent detection and slot filling was proposed to introduce an intent-slot graph interaction layer to model the strong correlation between the slot and intents. For jointly performing both the tasks co-interactive transformer network has been investigated in [90].

In [91], the authors simultaneously identified the intents and slots along with language identification and translation using m-BART. For low-resource NLU, lightweight data augmentation was used in [92]. Recently, in [93], the authors proposed an intent pooling attention mechanism and reinforced the slot filling task by fusing intent distributions, word features, and token representations. In [94], the authors present a new multilingual dataset named MTOP for both the NLU tasks of intent detection and slot filling. In [95], an Attention-Informed MLT framework is employed, i.e., a novel zero-shot adaptation method for cross-lingual task-oriented dialogue systems.

In [96], the authors have proposed a novel Transformer encoder-based architecture with syntactical knowledge encoded for intent detection and slot filling task. For zero-shot learning scenario [97], a novel method was employed for intent detection and slot filling task to augment the monolingual source data using multilingual code-switching via random translations to enhance a transformerâ s language neutrality when fine-tuning it for a downstream task such as dialogue generation. In [91], a multilingual BART framework was investigated for jointly identifying the intents and slots. Continual Learning Interrelated Model (CLIM) was proposed to consider semantic information with different characteristics and balance the accuracy between intent detection and slot filling in [98]. A transfer learning approach employing a Context Encoding Language Transformer (CELT) model to facilitate exploiting various context information for SLU was explored in [99]. The authors in [100] proposed a collaborative memory network to capture slot-specific and intent-specific features from memories for simultaneously identifying intent and slots in a given utterance. In [101], a dual learning approach was investigated for the SLU task of intent detection and slot filling on the ATIS and SNIPs dataset. Recently, in [102] graph convolutional network was employed for multidomain SLU. An adaptive graph interactive framework was proposed in [89], for jointly identifying the intents and slots in a given utterance. In [103], the authors have proposed a novel Parallel Interactive Network (PIN) to model the mutual guidance between intent detection and slot filling tasks. Lately, wheel graph attention network was investigated in [104] for simultaneously identifying the intents and slots in a given user utterance.

Our proposed model differs from the previous works in the sense that we propose a multitask framework that can perform both intent detection and slot filling together for three languages, *viz.* Bengali, Hindi, and English. To the best of our knowledge, this is the very first attempt to investigate the SLU tasks under a resource-scarce scenario involving Indian languages like Hindi and Bengali. Hindi is the most widely spoken language in India, and in terms of native speakers, it ranks 5th all over the world. Bengali is a popularly spoken language in India and also a national language in Bangladesh.

3. Methodology

This section describes the proposed MultiLingual MultiTask (MLMT) Model to detect the intents and capture the semantic information as slots for various languages. Our model comprises the embedding layer having word embeddings of the utterance as input to the model. A shared encoder for utterance representation that captures the contextual information of the utterances, followed by two separate output layers for performing intent detection and slot filling simultaneously. The outputs of the proposed multilingual multitask models are fed to two different multilayer perceptron (MLP) models for intent and slot detection, respectively. The problem can be mathematically expressed as follows: For a given utterance S in the jth language, having a sequence of words $w_1^j, w_2^j, \ldots, w_n^j$ with n being the length of the utterance, the task is to predict the slot label sequence y_s and intent label y_i simultaneously.

3.1. Embeddings

Continuous distributional word embedding has become de-facto input to the neural networks for solving various NLP tasks [30,35]. The word embedding is used to convert words into their continuous vector representation. It has the interpretable property that words with similar meanings in vector space are close together. Predominantly, word embedding has been language-specific, where for every language, word embeddings are trained separately, and they exist in entirely different vector spaces. Recent research focuses on developing cross-lingual word embeddings that use shared embedding space across two (bilingual word embedding) or more languages (multilingual word embedding). Using this method, embedding occurs in the same vector space for each language and retains the property that terms with similar meanings (regardless of the language) are close together in the vector space. For example, in the embedding space, the terms “paani” in Hindi, “jol” in Bengali, and “water” in English will appear very near because they correspond to the same thing in different languages. We use the word-level embedding to capture the latent semantic knowledge of the words in a given utterance as input to our models.

Multilingual Embeddings: Multilingual Word Embeddings (MWEs) represent the words in a single distributional vector space belonging to different languages. Shared word representation across multiple languages offers exciting opportunities. For example, in machine translation, translating a word in a given language not present in the training data can be overcome by seeking its neighbors in the vector space. Also, multilingual embeddings are beneficial for transfer learning, in which models trained in a given language can be deployed for other languages for a particular NLP task.

In this work, we learn a single multilingual embedding for all three languages, i.e., English, Hindi, and Bengali. For learning multilingual embeddings, we follow the work of [105,106]. For monolingual embeddings of all the three languages, we use pre-trained FastText embeddings [107]. The word embedding matrices of English, Hindi, and Bengali are denoted by X, Y, Z such that the ith row of X_i, Y_i and Z_i denote the embedding of the ith word in their respective vocabularies. We learn the linear transformation matrices W_X, W_Y, W_Z so that the mapped embeddings XW_X, YW_Y, and ZW_Z are in the same vector space.

3.2. Baseline MLMT models

The output of the embedding layer is passed to a sentence encoder to obtain the sentence-level representation. We use CNN, LSTM, and GRU as the sentence encoder for the baseline models, while in our proposed approach, we employ BERT as the sentence encoder.

3.2.1. Recurrent neural network

In our task, we use Bi-directional LSTM (Bi-LSTM) [27] to model features from both directions to provide additional context, as shown in Fig. 1. At every time step, it looks at the current input and previous hidden memory to generate its next output and hidden memory. LSTM mainly uses its forget gate, input gate, and output gate to control its output and next hidden memory. In our case, Bi-LSTM is used to obtain the hidden representation of every word by processing multilingual embeddings for the different languages at every time-step.

GRU [108] is also a special variant of RNN that deals with the vanishing gradient problem to learn long-range dependencies. GRU uses only two gates, reset and update, to control its output and hidden memory. Thus GRU has fewer parameters to learn, which facilitates efficient training. Similar to LSTM, GRU is used to get sentence representation. The Bi-LSTM/GRU (both are used interchangeably in the equations) hidden layers are represented as follows:

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(w_t^j, \overrightarrow{h}_{t-1}) \tag{3}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(w_t^j, \overleftarrow{h}_{t-1}) \tag{4}$$

where $\rightarrow$ and $\leftarrow$ represent the forward and backward directions, respectively. The concatenation of forward and backward hidden states yields the final bidirectional hidden state $\vec{h}_t$ at time t.

$$\vec{h}_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{5}$$

The final hidden representation of a given utterance in a particular language is fed as input to the softmax layer for identifying the intents y^i and slots y^s simultaneously.

$$y^i = softmax(W_i \vec{h}_t + b_i) \tag{6}$$

$$y^s = softmax(W_s \vec{h}_t + b_s) \tag{7}$$

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

3.2.2. Convolutional neural network

To perform text classification using CNN [25], a two-dimensional matrix is created by stacking word embedding of the given sentence. L dimensional convolution filters are applied to obtain a new representation of the given word. To compute the final hidden representation, a max-pooling operation is performed across different filters as shown in Fig. 2.

For the ith word in a given utterance, we consider $x_i \in \mathbb{R}^d$ to be the d-dimensional multilingual word vector. In the convolutional layer, we feed x_i having a filter $k \in \mathbb{R}^{hd}$, which is applied to a window of h words to produce a new feature for an utterance in a given language. From a window of words $x_{i:i+h-1}$ we obtain a feature p_i denoted by:

$$p_i = f(k \odot x_{i:i+h-1} + b) \tag{8}$$

where the bias term is represented by b. Similarly, the element-wise multiplication and non-linear function are denoted by $\odot$ and f, respectively. For the complete utterance, the feature map is represented as $p = [p_1, p_2, \ldots, p_{n-h+1}]$ by the application of the filter to every possible window of words present in the utterance. To obtain the maximum value for a particular filter, we apply a max-pooling operation $\hat{p} = max[p]$ for every utterance. In this study, to capture the features of a given utterance, we utilize three different filters. Every convolutional layer's feature representation is concatenated and then fed as input to the output layer with a softmax activation function to classify the intents and slots in a given utterance in a similar manner as in the case of Bi-LSTM/GRU.

3.3. Proposed MLMT model

In this section, we present our proposed methodology. In our current work, we design the proposed network for three languages i.e., English, Hindi and Bengali. The proposed methodology can be extended for multiple languages as well thereby making the model language-invariant. As discussed earlier, the multilingual embeddings of different languages can be computed as explained. In addition, the different language information can be fed as input to the encoder followed by task-specific layers to predict the corresponding intent and slots.

3.3.1. Bidirectional encoder representations from transformer

BERT [109] is an attention-based architecture to learn language representations. It builds from the recent work in pre-training contextual representations â including semi-supervised sequence learning, generative pre-training, ELMo, and ULMFit. Compared to these previous models, BERT is the first bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. Learned language representations can be context-free or contextual and contextual representation can be *unidirectional* or *bidirectional*.

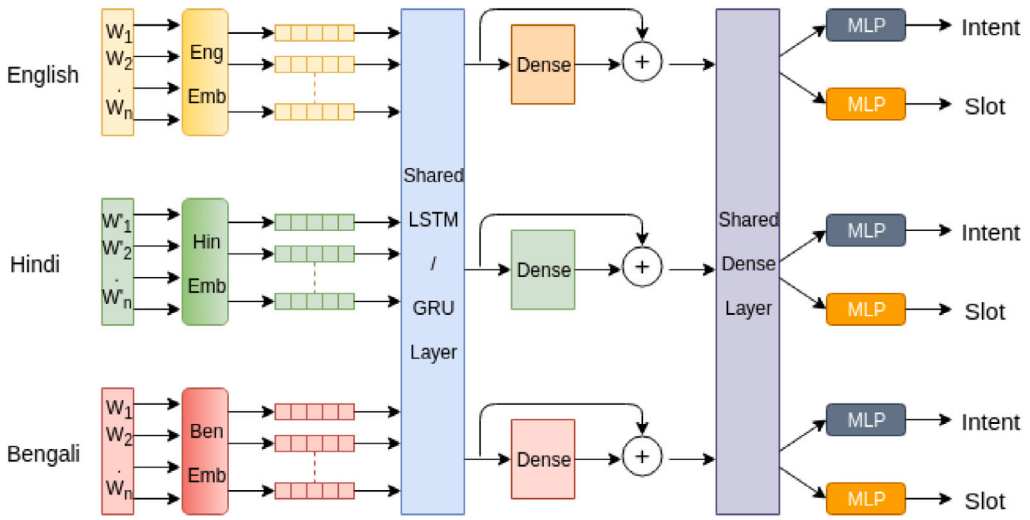

Fig. 1. Baseline multilingual RNN model with either LSTM or GRU as basic RNN units; here word embeddings of different language utterances are fed as input to the shared RNN layer followed by the dense layer; finally, the task specific layer uses the dense representations for intent and slot predictions using MLP and CRF layer, respectively.

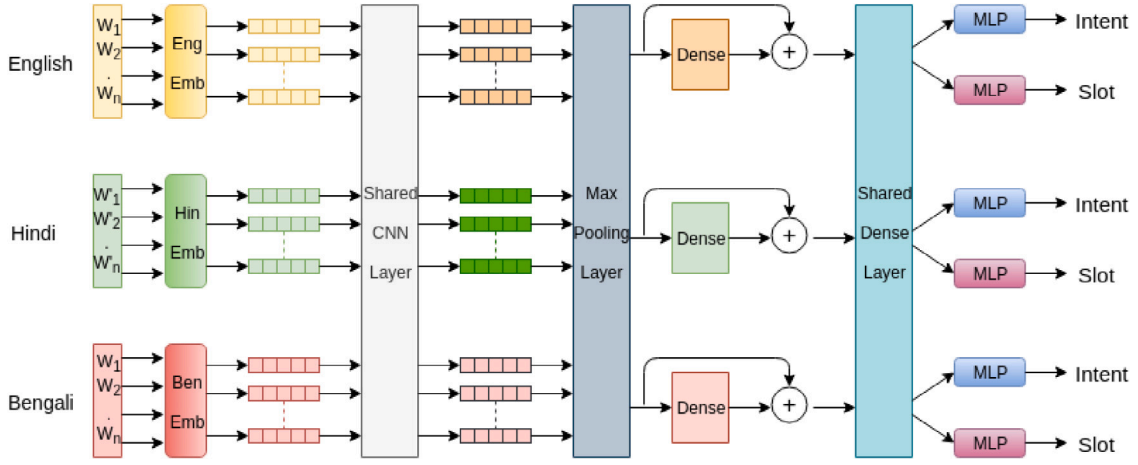

Fig. 2. Baseline multilingual CNN Model, here the word embeddings are fed as input to the shared CNN layer followed by max pooling layer, the output of pooling layer is fed as input to the dense layer and finally there are task-specific layers for intent and slot prediction.

In our baseline model, we use Fasttext context-free language representation with a word having a single vector. For example, in context-free representation the word "book" will have the same representation in "book a cab" and "buy a book". However, the contextual model will generate a representation-based context formed by other words in the given sentence. Hence, we propose our final model based on contextual representation. BERT learns its contextual representations using masked language modeling (MLM) and the next sentence prediction (NSP) task.

The architecture of BERT is a bidirectional multilayer transformer network, which is shown in Fig. 3. It takes wordpiece embeddings [110], positional embeddings, and segment embedding as input. It is composed of N identical transformer blocks. Every transformer can be divided into two parts. The first part is multihead self-attention, and the second is a position-wise feed-forward network. A residual connection is used around each of the two sub-layers, followed by layer normalization. After pre-training over a large corpus such as BooksCorpus [111] and English Wikipedia, it is fine-tuned for different target tasks such as intent detection and slot filling. Due to pre-training, the BERT model achieves the capability of learning a powerful context-dependent representation of sentences that is useful for many downstream utterance level NLP tasks. Given a sequence of tokens in a given language $x_1, \ldots, x_n$, it computes a sequence of representations $h = (h_1, \ldots, h_n)$ to capture the contextual information for each token.

A special classification embedding ([CLS]) and a special token ([SEP]) is inserted as the first and final token, respectively. In [109], the authors suggested using the [CLS] token's final hidden state h_0 for the classification task, which should represent a fixed dimensional pooled representation of the sequence. While in the sequence labeling task, for every token x_i in a given utterance sequence, its corresponding hidden representation h_i is used to classify into the target categories.

3.3.2. Multilingual BERT

To identify the intents and slots for multiple languages (in our case, English, Hindi, Bengali), we employ Multilingual BERT (mBERT)[4] as presented in Fig. 4 that has the similar architecture and training mechanism as the BERT [109], with an exception that it has been trained from Wikipedia in 104 languages. In mBERT, the training does not require explicit cross-lingual or multilingual information such as the pair of words, sentences, or documents linked across the languages (parallel data). The modeling approach for WordPiece allows the system to share the embedding across languages in mBERT. In distantly related languages such as English and Bengali, the same word has a similar meaning for both languages.

[4] https://huggingface.co/bert-base-multilingual-cased.

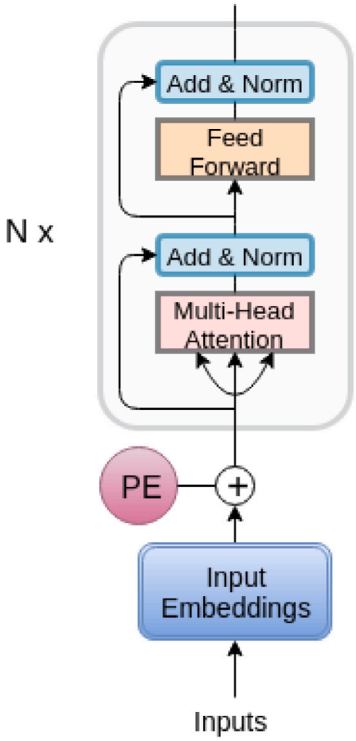

Fig. 3. The transformer block: takes wordpiece, positional and segment embeddings as input; the transformer block consists of two parts: multi-head self-attention and the feed-forward network.

The training uses a heuristic to sub-sample or over-sample words when running WordPiece and sampling a training batch, random words for cloze, and random sentences for next sentence classification to account for varying sizes of Wikipedia training data in different languages. The mBERT described above has been extended to identify intents and slots together for multiple languages simultaneously. Based on the first special token ([CLS]) having hidden representation h_1, the intent of a given utterance U_i for a particular language j can be predicted as:

$$y^i = softmax(W^i h_1 + b^i), \tag{9}$$

In the case of slot filling, the hidden representations of the other tokens $(h_2, \ldots, h_T)$ are fed as input to a softmax layer to identify the slot labels for every word in a given utterance. To make soft filling suitable with the WordPiece tokenization, every tokenized input word is fed to the WordPiece tokenizer, and the corresponding hidden representation of the first sub token is given as input to the softmax layer.

$$y_n^s = softmax(W^s h_n + b^s), n \in 1 \ldots N \tag{10}$$

where, the hidden representation corresponding to the first sub-token word x_n is denoted by h_n.

The objective function for simultaneously modeling the intent and slots for a given utterance in a particular language is defined by:

$$p(y^i, y^s/x) = p(y^i/x)\Pi_{n=1}^{N} p(y_n^s/x) \tag{11}$$

In our case, the objective is to maximize the conditional probability $p(y^i, y^s/x)$. The MLMT model is fine-tuned in an end-to-end fashion via minimizing the cross-entropy loss.

3.3.3. Conditional random field

Slot filling is a sequence labeling task where the slot label of a particular word is dependent on the surrounding words. In the past [35,84], structured prediction models such as CRF have been used for improving the performance of the slot filling task.

Conditional Random Fields are undirected graphical models which, given an observed sequence, help to achieve the conditional probability of a label sequence. In our proposed model, we examine the effectiveness of incorporating CRF for modeling the dependency among the slot labels. This is done to increase the efficiency of prediction.

3.3.4. Attention

Generally, in mBERT, a linear layer is applied to the contextual embedding to jointly classify intents and slots in a given utterance for different languages. The different layers of BERT capture the syntactic and semantic information of an utterance gradually. In the initial layer, the BERT captures the syntactic information, while in the latter layers, it acquires more semantic information [112]. For identifying the intents and slots, we require a different amount of syntactic and semantic information as it depends on the given utterance. Basically, BERT generates L layers of hidden states for all the Byte Pair Encoding (BPE) tokens for a given utterance.

Inspired by the work done in conversational question answering [113] in which the weighted-sum approach between the hidden states is employed to obtain contextualized embedding of the words assists in improving the performance of the overall task, we also apply a similar approach. Similarly, we employ a weighted sum of these hidden states to obtain contextualized representation to incorporate the ability to focus on the different features (syntactic or semantic) of a given utterance. Suppose a word w is tokenized to s BPE tokens $w = b_1, b_2, \ldots, b_s$, and BERT generates L hidden states for each BPE token, h_t^l, $1 \leq l \leq L$, $1 \leq t \leq s$. The contextual embedding $BERT_w$ for word w is then a per-layer weighted sum of average BERT embedding, with weights $\alpha_1, \ldots, \alpha_L$ as shown in Fig. 5.

3.4. Baseline models

We compare our proposed *MLMT* model with the following baselines and existing approaches, respectively:

3.4.1. Model variants

To demonstrate the effectiveness of each of the components in the proposed model, we experiment with model variants having these components.

1. CNN: This baseline utilizes the CNN framework to encode the utterances described in Section 3.2.2 of the manuscript.
2. LSTM: This baseline employs a bidirectional LSTM network for encoding the utterances, as illustrated in Section 3.2.1 of the manuscript.
3. GRU: This baseline is similar to the previous baseline with the difference being in the RNN cell, i.e., we use bidirectional GRU for capturing the utterance representation as explained in Section 3.2.1 of the paper.
4. RoBERTa: In this baseline, instead of deep learning frameworks such as CNN, LSTM, GRU, we utilize recent Transformer architecture RoBERTa [114] to encode the utterance for identifying the intents and slots simultaneously.
5. XLM: For capturing the utterance representation, in this baseline, we employ the recently proposed XLM model [115] for the detection of intents and slots.
6. mBERT: In this baseline, we utilize multilingual BERT(mBERT) architecture to encode the utterance for capturing the correct intents and slots from different languages as explained in Section 3.3.2 of the paper without having attention upon BERT layers and CRF for slot filling.
7. mBERT + Attn: In this baseline, we apply attention among the different layers of the BERT architecture as discussed in Section 3.3.4 of the manuscript without having the CRF layer.

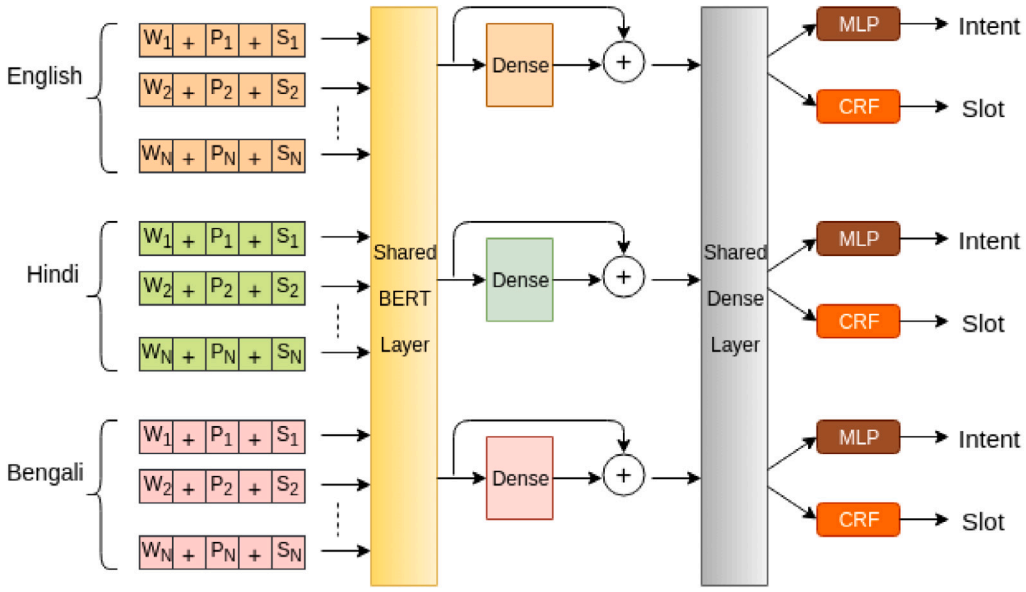

Fig. 4. Proposed multilingual multi task model, every utterance is fed as input to the shared BERT layer followed by dense layer; the output of the dense layer is fed as input to the task specific layer (MLP, CRF) for intent and slot predictions.

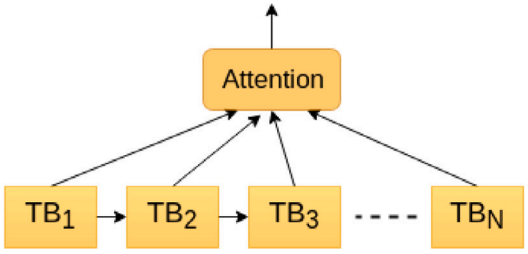

Fig. 5. BERT layer; here, weighted sum of the hidden states are calculated to obtain contextualized representation in order to focus on the different features of an utterance.

3.4.2. Existing approaches

To demonstrate the effectiveness of our proposed model for the English language (results in Table 7, we compare with the previous State-of-the-Art (SoTA) models:

1. RNN-LSTM [67]: This existing framework utilizes a bi-direction LSTM[5] network for jointly identifying the intents and slots from a given user utterance.
2. Attention BiRNN [70]: In this framework, the authors employ an attention-based bi-directional RNN model[6] for jointly detecting intents and slots.
3. Bi-Model with Decoder [73]: This framework uses a Bi-model based RNN[7] semantic frame parsing network structures designed to perform the intent detection and slot filling tasks jointly by considering their cross-impact to each other using two correlated bidirectional LSTMs.
4. Slot-Gated [74]: In this approach, a slot gate[8] focused on learning the relationship between intent and slot attention vectors to obtain better semantic frame results by the global optimization of both the tasks simultaneously is employed.
5. Capsule-NLU [78]: In this framework, capsule network[9] via a dynamic routing-by-agreement schema was employed for capturing the intents and slots jointly from the given user utterance.
6. BERT-Joint [85]: In this approach, the BERT framework was utilized for joint intent detection and slot filling tasks.
7. Hierarchical NLU [35]: This approach uses a hierarchical CNN-RNN framework for concurrently detecting the intents and capturing the slots from a given utterance.

To demonstrate the effectiveness of our proposed model for different languages (results in Table 9, we compare with the previous State-of-the-Art models:

1. AIMT [62]: Attention-Informed MLT,[10] utilizing bi-directional LSTM network was used as one of the baselines.
2. CoSDA-ML [95]: This existing approach utilizes mBERT[11] along with a data augmentation network to fine-tune the BERT architecture is also considered as one of the baselines.

4. Dataset

We conduct experiments on four datasets, namely ATIS [18], SNIPS [74], FRAMES [116] and TRAINS [117].

4.1. ATIS dataset

An important by-product of the Defense Advanced Research Project Agency (DARPA) program is the ATIS (Airline Travel Information System) corpus. The ATIS corpus [18] is one of the most widely used datasets for SLU tasks. ATIS corpus has a few variants, and in our current work, we use the one as presented in [44]. The utterances in the ATIS corpus are primarily about the people making flight reservations. The dataset comprises of 17 distinct intent classes and 127 distinct slot labels. The train set contains 4978 utterances, and the test set contains 893 utterances.

[5] https://github.com/yvchen/JointSLU.
[6] https://github.com/DSKSD/RNN-for-Joint-NLU.
[7] https://github.com/ray075hl/Bi-Model-Intent-And-Slot.
[8] https://github.com/MiuLab/SlotGated-SLU.
[9] https://github.com/czhang99/Capsule-NLU.
[10] https://github.com/zliucr/mixed-language-training.
[11] https://github.com/kodenii/CoSDA-ML.

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

Language	Dataset	# Train	# Test	# Intent	# Slot
Engish	ATIS	4978	893	17	127
	TRAINS	5355	1336	12	32
	SNIPS	13 084	700	7	72
	FRAMES	20 006	6598	24	136
Hindi	ATIS	4221	780	16	108
	TRAINS	3762	959	12	22
	SNIPS	12 152	700	7	72
	FRAMES	9877	3841	24	119
Bengali	ATIS	3956	741	16	100
	TRAINS	3829	971	12	82
	SNIPS	12 762	700	7	72
	FRAMES	6797	3852	24	104

4.2. TRAINS dataset

It is essential to capture the intent and slots present in a human conversation to create a robust spoken dialogue framework. The TRAINS corpus is a set of problem-solving dialogues. The dataset has been labeled with 32 slots and 12 intent classes. The training set contains 5355 utterances, and the test set contains 1336 utterances.

4.3. FRAMES dataset

The FRAMES corpus comprises of 1369 human–human dialogues. The average length of dialogue is 15. The corpus is a series of conversations concerning multidomain hotel and trip reservations. The training set contains 20 006 utterances, and the test set contains 6598 utterances. The corpus is labeled with 24 intents and 136 slots.

4.4. SNIPs dataset

This dataset is collected from Snips personal voice assistant, where for each expressed intent, the number of samples is about the same. The training set includes 13,084 utterances, and the test set includes 700 utterances. There are 7 labels of intent, while there are 72 labels of slots in the dataset. Due to the diversity of intent labels and comprehensive vocabulary, the SNIPs dataset is more complex.

4.5. Dataset creation

Due to the unavailability of Hindi and Bengali datasets for the primary SLU tasks focusing on intent detection and slot filling, we manually create the language-specific datasets. The existing English datasets, such as ATIS, TRAINS, SNIPs, and FRAMES, have been manually translated using our in-house English to Indian Language Machine Translation System.[12] Human experts manually verified the translated sentences for correctness. We assigned three Bengali and Hindi native speakers with post-graduate experience for this particular task. Also, for the translated utterances in Hindi and Bengali, the annotators were assigned to annotate the utterances with their corresponding slot values. As the word order changes in Hindi and Bengali as opposed to English, we need to mark the slots for the translated utterances in both languages to obtain the correct slot values. The inter-annotator score of more than 90% was considered a valid agreement for the translation in Hindi and Bengali. While annotating the translated utterances with corresponding slot values, we observed the inter-annotator score with more than 85%, which can be considered a reliable agreement. Table 3 shows the utterance, intent, and slot distribution for all the datasets for various languages.

5. Experiments

For implementation, we use the Python-based neural network package, Keras.[13] Here, we define a baseline model based on CNN with two convolutional layers having different filters (the filters used are 4,5,6) followed by the max-pooling layer. We also use multilingual embedding [118] as input to our CNN model, and we take it as a baseline for our multilingual multitask model. The intermediate layers use the ReLU activation function in our CNN method, whereas the last layer uses the softmax activation function for intent detection and CRF for the slot filling. We also use batch normalization after the activation function to increase the stability of the neural network. In our LSTM/GRU model, we use two layers of bidirectional LSTM/GRU. The first layer of LSTM/GRU has 256 hidden units and the second layer of 128 hidden units. For our BERT-based model, we use cased-multilingual BERT[14] for sentence-level as well as word-level representations. After getting sentence representation (768 Dimension) from BERT, we fed it to the next dense layer with 256 hidden neurons and finally to the softmax for intent classification. We fed word representation (1024 dimension) for slot prediction, which we took from BERT, into a fully connected layer with 256 neurons and finally to the CRF layer for slot prediction.

6. Results and discussion

The results of various experiments, together with the analysis of errors made by these approaches, are discussed in this section. The detailed results of the baseline models, along with the proposed multilingual multitask models, are analyzed. In comparison to the individual task-specific methods, the usefulness of our proposed multitask system is also demonstrated. Furthermore, the efficacy of our proposed approach is exhibited in comparison to the language-specific multitask models. Besides, we provide a brief comparison of our proposed framework with the state-of-the-art approaches for the English dataset. As for the evaluation metrics in the case of intent detection and slot filling tasks, we report accuracy and F1 score, respectively. We also report template/overall accuracy for the sentence-level semantic frame parsing. This metric ensures that the semantic information captured is correct in a given utterance. This metric is evaluated similarly as [90].

6.1. Results

In Table 4, we provide the detailed results for both the SLU tasks for all the languages, i.e., English, Hindi, and Bengali, for the different baseline models followed by the results of the proposed model. From the table, it can be seen that the CNN baseline models for all the languages do not perform well in contrast to the other approaches for either of the two tasks.

[12] Currently, the MT systems show the BLEU scores in the range of 45–56 for English–Hindi and English–Bengali language pairs.

[13] https://keras.io.

[14] https://huggingface.co/bert-base-multilingual-cased.

Table 4
Results of multilingual multitask models. Here, MLMT: mBERT + Attn + CRF.

Dataset	Models	English			Hindi			Bengali		
		Intent (Accuracy)	Slot (F1 score)	Template (Accuracy)	Intent (Accuracy)	Slot (F1 score)	Template (Accuracy)	Intent (Accuracy)	Slot (F1 score)	Template (Accuracy)
ATIS	CNN	94.84	90.56	82.11	91.79	91.58	80.27	91.14	91.94	79.64
	LSTM	95.18	91.78	82.89	92.89	92.47	76.89	92.33	92.62	75.89
	GRU	95.98	92.94	82.19	93.97	93.38	77.43	92.96	93.01	76.27
	RoBERTa	98.95	96.99	89.21	97.18	96.68	90.88	96.22	95.83	88.97
	XLM	98.67	96.13	88.25	96.85	96.05	89.74	95.89	95.42	88.67
	mBERT	98.15	96.85	88.78	96.18	95.12	89.95	95.37	95.14	88.93
	mBERT + Attn	98.63	97.07	89.65	96.91	96.08	91.07	96.68	95.75	89.56
	Proposed MLMT	99.18	97.93	90.05	97.59	97.24	91.33	96.91	96.31	90.14
TRAINS	CNN	82.01	94.12	81.05	81.88	93.76	81.78	81.76	94.78	81.95
	LSTM	82.78	95.41	81.89	82.78	94.84	82.75	82.59	95.15	82.56
	GRU	83.12	95.45	81.93	83.76	94.96	83.41	82.78	95.87	82.61
	RoBERTa	85.93	97.85	88.98	85.63	97.33	90.89	86.23	98.03	90.07
	XLM	85.22	96.98	88.12	85.32	96.80	89.95	85.67	97.43	89.65
	mBERT	85.16	97.95	88.78	85.52	97.27	90.34	85.51	97.08	89.85
	mBERT + Attn	86.27	98.64	89.18	86.33	97.95	90.41	86.77	98.12	90.01
	Proposed MLMT	86.45	99.01	89.56	86.76	98.99	91.34	86.94	99.05	90.56
FRAMES	CNN	73.87	83.05	71.56	70.19	79.23	67.42	69.24	80.96	69.87
	LSTM	75.13	86.17	72.88	71.72	82.71	69.06	70.51	83.95	70.76
	GRU	76.29	85.98	72.97	73.85	83.15	71.67	72.81	83.92	71.99
	RoBERTa	80.53	90.71	83.09	76.23	87.05	79.21	76.11	87.45	79.43
	XLM	79.66	89.91	82.65	75.34	86.25	78.44	75.49	87.23	79.10
	mBERT	79.15	89.82	83.15	75.37	86.02	79.21	75.51	86.95	79.35
	mBERT + Attn	80.34	90.65	83.22	76.16	86.54	79.52	76.28	87.57	79.63
	Proposed MLMT	80.91	91.67	83.56	76.32	87.39	79.67	76.43	88.29	79.98
SNIPS	CNN	93.37	88.95	83.22	90.19	82.67	78.90	89.94	81.92	76.54
	LSTM	94.51	91.17	84.65	92.67	84.15	80.17	91.51	83.25	79.59
	GRU	94.29	91.19	84.54	92.05	84.44	80.19	91.81	83.97	79.63
	RoBERTa	98.57	96.68	90.15	96.15	88.72	84.71	95.07	87.51	86.34
	XLM	98.18	95.44	89.87	96.02	87.95	84.21	94.86	86.79	85.45
	mBERT	98.25	95.85	90.43	95.59	88.12	84.33	94.51	86.85	86.72
	mBERT + Attn	98.93	96.54	90.75	96.13	88.87	85.14	95.03	87.42	86.89
	Proposed MLMT	99.11	97.08	91.20	96.42	89.79	85.60	95.19	88.23	87.21

This could be due to the inability of CNN to capture sequential information. Though LSTM and GRU are similar, we still find GRU to outperform LSTM by 1 point or more for both intent detection and slot filling tasks, especially for Hindi in the case of all four datasets. The GRU-based approach outperforms LSTM by 2% for the task of intent detection in the case of the FRAMES dataset for both Bengali and Hindi languages. From the experimental results, it is evident that the baseline models employing CNN, LSTM, and GRU demonstrate satisfactory performance for both the tasks for all the three languages, still are not robust enough, and hence we employ the BERT framework. The evaluation shows that the multilingual BERT approach outperforms the previous baseline models with an improvement of at least 3% or more for both the SLU tasks of intent detection and slot filling. The performance of both the tasks increases when they are modeled together; thereby, the shared representation of the sentence encoders helps to collect intricate information about the language and task by adding more knowledge to a particular utterance that helps in respect of intent detection and slot filling.

By using attention among the different mBERT layers, we achieve an improvement of almost 1% in the case of all the datasets for all three languages. Hence, it can be concluded that the attention between the mBERT layers enhances the accuracy and F1 score of the intent detection and slot filling tasks, respectively. As already discussed in the methodology section, we employ CRF to capture the label dependency for the task of slot filling. Hence, our proposed model employs an attentive mBERT with CRF for identifying the intents and slots in a given utterance. For the slot filling task, it is noticeable that CRF gives an increase of almost 1% hence enriching the performance of the model. While we do not employ CRF for intents in the proposed framework, we still see performance improvement for intent detection in our final proposed model. This is because CRF implicitly improves the performance of the intent detection task by enhancing the performance of the slot filling task.

Since both the tasks are interrelated, improvement in the task of slot filling also helps to improve the performance of intent detection and vice versa. Experimental results in Table 4 verify the fact that improvements are made using an attentive mBERT as the sentence encoder along with CRF with a gain of at least 5% from the baseline models (CNN) for both the tasks. As opposed to deep learning frameworks such as CNN, LSTM, GRU, recent Transformer architectures (such as RoBERTa, XLM) have performed remarkably better, showcasing the effectiveness of these networks. With attentive BERT and CRF, our proposed architecture performs better than RoBERTa and XLM models validating the significance of attention and CRF for both tasks. As it is evident from the Table, the overall accuracy of our proposed MLMT framework is higher than all the baselines indicating that the proposed architecture is capable of identifying the correct semantic information in a given utterance. In comparison to the CNN, LSTM, GRU models, we see an improvement of more than 4 points in the case of all the languages for different datasets. In comparison to RoBERTa and XLM frameworks, our proposed MLMT network performs better for both tasks. This is mainly because we have applied attention to BERT's different layers to focus on different semantic information captured by the different layers. Also, using CRF at the final layer boosts the performance of our proposed framework in comparison to RoBERTa and XLM models. It can be concluded that the ability to identify the correct intents and slot information from different languages belonging to different domains is performed well by our proposed framework.

6.2. Multilingual multitask models vs. multilingual single-task models

For a complete analysis of our work, we compare the multilingual individual frameworks to examine the efficacy of the proposed multilingual multitask model. We implement the task-specific models using identical settings and parameters which detect intents and slots for all the languages individually.

Table 5
Proposed MLMT model vs. Multilingual individual model (For all the languages). Here, MLMT: mBERT + Attn + CRF.

Dataset	Task	English		Hindi		Bengali	
		Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
ATIS	mBERT (Only ID)	97.70	–	95.97	–	93.85	–
	mBERT (Only SF)	–	94.35	–	93.17	–	93.23
	Proposed MLMT	99.18	97.93	97.59	97.24	96.91	96.31
TRAINS	mBERT (Only ID)	83.19	–	83.50	–	84.14	–
	mBERT (Only SF)	–	97.76	–	96.22	–	95.36
	Proposed MLMT	86.45	99.01	86.76	98.99	86.94	99.05
FRAMES	mBERT (Only ID)	76.89	–	74.56	–	73.84	–
	mBERT (Only SF)	–	88.89	–	83.56	–	87.91
	Proposed MLMT	80.91	91.67	76.32	87.39	76.43	88.29
SNIPS	mBERT (Only ID)	96.83	–	93.15	–	93.85	–
	mBERT (Only SF)	–	93.39	–	84.66	–	84.98
	Proposed MLMT	99.11	97.08	96.42	89.79	95.19	88.23

Table 6
Multilingual multitask BERT model vs. Language-specific multitask BERT model.

Dataset	English		Hindi		Bengali	
	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
ATIS	98.18	95.91	–	–	–	–
	–	–	94.79	94.19	–	–
	–	–	–	–	93.87	94.88
	99.18	97.93	97.59	97.24	96.91	96.31
TRAINS	84.41	96.89	–	–	–	–
	–	–	83.39	95.80	–	–
	–	–	–	–	83.14	96.66
	86.45	99.01	86.76	98.99	86.94	99.05
FRAMES	78.73	89.49	–	–	–	–
	–	–	73.19	85.87	–	–
	–	–	–	–	74.89	86.85
	80.91	91.67	76.32	87.39	76.43	88.29
SNIPS	96.89	95.73	–	–	–	–
	–	–	94.02	87.22	–	–
	–	–	–	–	92.98	85.11
	99.11	97.08	96.42	89.79	95.19	88.23

Evaluation results of the multitask model and individual models for all the languages are presented in Table 5. Contrary to the single-task models, it is quite clear from the table that the multitask system performs better. For intent detection tasks on the ATIS dataset, there is an increase of approximately 2% for English, more than 2% for Hindi, and more than 3% for Bengali, respectively, compared to the individual intent detection model. There is more than 3 points improvement for the slot filling task for English, Bengali, and Hindi languages in our proposed MLMT approach that performs both the task simultaneously compared to the slot filling model(single task).

Similarly, for the TRAINS dataset, there is an increase of at least 2% for both the tasks in all the languages compared to the individual intent and slot models. Also, for the FRAMES dataset and SNIPS dataset, the multitask model performs better for all the languages for both tasks. It is visible from the evaluation results presented in Table 5 that information sharing between the SLU tasks helps enhance the performance of both the tasks simultaneously in the MLMT model. Hence, it can be concluded that the MLMT model outperforms the individual slot filling and intent detection models for all the languages.

6.3. Multilingual multitask models vs. language specific multitask models:

Furthermore, to validate the efficacy of our proposed multilingual multitask framework, we compare it to the language-specific multi-task models. Using the same parameter settings, we implement the language-specific multitask models that detect intent and extract the slots simultaneously for every language.

In Table 6, we display the results of the proposed multilingual multitask model and language-specific multitask models. From the table, it is evident that the MLMT model shows better performance as opposed to the language-specific multitask models. The fact that the information between the languages is shared can be seen from the results as there is marked improvement in the MLMT models. Also, information sharing among the related languages is visible in the case of Hindi and Bengali, as performance improvements for these languages in the MLMT models are more significant than the language-specific models in comparison to English. For all the datasets and for both the tasks of SLU, the MLMT models outperform the language-specific models.

6.4. Comparison of the proposed MLMT model with the existing approaches

We compare the proposed MLMT model with the existing approaches for both intent detection and slot filling tasks in addition to the task-specific and language-specific models. In Table 7, we present the results for all the datasets in the case of the English language. MLMT outperforms existing approaches for both tasks on all the datasets. There is a marked improvement of more than 1 point for both the tasks in the case of the SNIPS and FRAMES dataset in comparison to [35]. There is an increase in accuracy and F1 score for ATIS and TRAINS dataset compared to the previous approaches. Hence, it can be established that the proposed MLMT model is better than all the existing models. The improvement of MLMT compared to the existing baselines is primarily because jointly training for different language sharing of information between languages facilitates boosting the performance for every language as opposed to language-specific models.

Table 7
Comparative results of the existing approaches with the proposed MLMT model (For English only);RNN-LSTM: employs a Bi-directional LSTM network, Attention BiRNN: uses attention based bi-directional LSTM network, Bi-Model with Decoder: employs two correlated BiLSTMs, Slot-Gated: slot-gated attention using BiLSTM, Capsule-NLU: employs capsule networks, BERT-Joint: Uses BERT architecture for both the task, Hierarchical NLU: Uses Hierarchical CNN + CRF.

Models	ATIS		SNIPS		TRAINS		FRAMES	
	Intent	Slot	Intent	Slot	Intent	Slot	Intent	Slot
RNN-LSTM [67] [Hakkani-Tur et al. 2016]	94.2	92.6	96.9	87.3	62.35	82.66	60.20	85.84
Attention BiRNN [70] [Liu et al. 2016]	98.43	95.87	97.29	90.14	80.61	94.41	63.30	88.63
Bi-Model with Decoder [73] [Wang et al. 2018]	98.99	96.89	97.65	92.46	81.41	95.29	64.17	88.36
Slot-Gated [74] [Goo et al. 2018]	94.10	95.20	97.00	88.80	75.66	81.44	59.42	78.36
Capsule-NLU [78] [Zhang et al. 2018]	95.0	95.2	97.7	91.8	76.28	80.56	60.74	79.12
Bert-Joint [85] [Castellucci et al. 2019]	97.8	95.7	99.0	96.2	80.58	93.84	65.29	87.34
Hierachical NLU [35] [Firdaus et al. 2019]	99.09	97.32	98.24	94.38	83.99	98.93	79.64	89.94
Proposed MLMT	99.18	97.93	99.11	97.08	86.45	99.01	80.91	91.67

Table 8
Analysis of multi-lingual multi-task model vs. Individual model (For all the languages).

Utterance	Proposed Multilingual Multi-task Model		Multilingual Intent Model	Multilingual Slot Model
	Intent	Slot		
What is MCO?	abbreviation	B-airport_code	airport	B-airport_code
ग्रिबल इंस्ट्रूमेंट्स प्लेलिस्ट में सबरीना सालेर्नो जोड़ें। (Gribal instrooments plelist mein sabareena saalerno joden.) (Add Sabrina Salerno to the grime instrumentals playlist.)	AddToPlaylist	B-artist I-artist B-playlist I-playlist	AddToPlaylist	B-artist I-artist O O
আপনি বোস্টন থেকে অন্য কাছাকাছি প্রস্থান শহর সুপারিশ করতে পারে। (Apani bostana theke anya kachakachi prasthana sahara suparisa korte pare.) (Could you suggest another nearby departure city from Boston.)	City	B-city_name	Flight	B-from.loc_city.name
Music from Clark Kent in the year 1987	PlayMusic	B-artist I-artist O	AddToPlaylist	B-artist O O
बिजनेस क्लास में ट्वाय की कितनी उड़ानें होती हैं?? (Bijanes klass mein TWA ki kitanee udaanen hotee hain?) (How many flights does TWA have in business class?)	Quantity	B-airline_code B-class_type O	Flight	O B-class_type O
আমরা সেন্ট লুই মধ্যে মহিমান্বিত হোটেল রিজার্ভ করা উচিত। (Amara senta lui madhye mahimanbita hotela rijarbha kara ucita.) (Should we reserve Glorious hotel in St Louis.)	book_hotel	B-hotel_name B-city_name I-city_name	book_hotel	O O B-city_name

The attention among different layers of BERT helps capture relevant semantic and syntactic information for correctly identifying the intents and slots simultaneously from a given user utterance.

6.5. Comparison of our proposed MLMT model with multilingual approaches

To better understand the effectiveness of our proposed Multilingual Multitask framework, we compare our proposed method with the recent multilingual approaches reported in [62,95]. Both of these existing frameworks are built upon the BERT architecture. In Table 9, we present results of the different existing approaches for different languages compared to our proposed MLMT network. From the table, it is evident that our proposed framework performs better than the existing multilingual networks. This is mainly due to the fact that our proposed network is capable of focusing on different layers of the BERT with the help of attention. The CRF layer also further improves the performance of the slot and implicitly of the intent tasks due to the inherent capacity of CRF that captures the dependency between the slot labels. Also, existing approaches utilize a zero-shot learning paradigm. In contrast, our approach uses a joint training mechanism for different languages (having annotated data for every language) simultaneously, enhancing the performance of our proposed work.

6.6. Result analysis

We perform a detailed analysis of the results obtained from the proposed MLMT model to gain better insights. In comparison to the individual models, multitask models show improved performance as both the tasks are highly correlated. Table 8 shows a few examples of the identified intents and slots from both the multitask model and the individual models. As it is evident from the table, the multitask model has been able to identify the correct intents and slots due to information sharing between the tasks. In contrast, the individual models (for intent and slot) encountered a few errors in identifying the correct slots and intents in an utterance for all three languages. We see that the predicted intents and slots are correctly identified in the multitask model compared to the individual models for all the datasets and languages.

Table 11 presents the predicted intent examples from the multilingual multitask models and the language-specific multitask models. It is evident from the table that the proposed model can predict correct intents instead of the language-specific models. This is mainly because the information across different languages is shared in the proposed model, and this provides more evidence in comparison to the language-specific models. Hence, the proposed model can learn the different

Table 9
Comparative results of the existing multilingual approaches with the proposed MLMT model.

Model description		English		Hindi		Bengali	
		Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)	Intent (Accuracy)	Slot (F1 score)
ATIS	AIMT [62]	99.05	97.65	96.65	97.20	95.86	96.21
	CoSDA-ML [95]	99.13	97.78	97.17	97.22	96.64	96.25
	Proposed MLMT	99.18	97.93	97.59	97.24	96.91	96.31
TRAINS	AIMT [62]	85.46	98.73	86.27	98.14	86.73	98.43
	CoSDA-ML [95]	86.12	98.95	86.58	98.64	86.81	98.77
	Proposed MLMT	86.45	99.01	86.76	98.99	86.94	99.05
FRAMES	AIMT [62]	79.14	90.98	75.87	86.56	75.89	87.64
	CoSDA-ML [95]	80.55	91.23	76.09	87.12	76.21	88.05
	Proposed MLMT	80.91	91.67	76.32	87.39	76.43	88.29
SNIPS	AIMT [62]	98.67	96.54	95.78	89.32	94.37	87.86
	CoSDA-ML [95]	99.02	96.83	96.32	89.54	94.80	88.11
	Proposed MLMT	99.11	97.08	96.42	89.79	95.19	88.23

Table 10
Results of the existing multilingual approaches with the proposed MLMT model on Spanish–Thai dataset [62].

Model description	Spanish		Thai	
	Intent (Accuracy)	Slot (F1-Score)	Intent (Accuracy)	Slot (F1-Score)
AIMT [62]	87.88	73.89	73.46	27.12
Multi-CoVe [119]	85.39	72.87	70.70	35.62
CoSDA-ML [95]	94.80	80.40	76.80	37.30
Proposed MLMT	**95.11**	**82.23**	**77.50**	**39.04**

patterns and information in all three languages, showcasing improved performance compared to the other language-specific multitask models.

In Table 10, we present the results of our proposed framework along with the existing baselines on the Spanish–Thai dataset [62,119]. From the results, it is evident that the proposed framework in comparison to the existing baselines shows improvement on the recently proposed Spanish–Thai [62,119] dataset having 12 intents and 11 slots in total. The existing baselines AIMT [62], Multi-CoVe [119] and CoSDA-ML [95] achieve the intent accuracy of 87.88, 85.39 and 94.80, respectively, for the Spanish while our proposed MLMT framework yields an intent accuracy of 95.11 showcasing the efficacy of our proposed network. Similarly for the Thai dataset, the MLMT network outperforms the existing baselines for both the tasks. The attentive BERT framework along with CRF improves the performance of both the tasks, thereby giving better scores for both the languages.

From Table 11, it is evident that intents are correctly identified in a multilingual setting, proving the efficacy of our proposed MLMT framework. As intents and slots are closely related, therefore identifying the correct intents is crucial for predicting the necessary slots in a particular utterance. By jointly training for all the three languages, we see that for the Hindi language, the proposed framework has correctly identified the intent "AddtoPlaylist" compared to the monolingual model that wrongly predicts the intent of the given utterance. Since the intent is wrongly identified the corresponding slot information is also incorrect as "kraibafish" is labeled as "restaurant-name" instead of "B-playlist". Similarly, in the case of the Bengali language, we see that identifying correct intents through multitask training for both the tasks has improved the performance of the model.

6.7. Error analysis

In this section, we highlight the major sources of errors, and discuss them. The key errors committed by the proposed model are listed below:

- Long utterances: Utterances, which are longer in length and having their intents present at the end are not correctly recognized. For example, "put playa fly onto my 2010 decade playlist" is wrongly labeled as "PlayMusic", while the correct label is "AddToPlaylist". In case of Bengali, "হ্যালো আমি 3 সুন্দর ছোট স্বর্গদূতের খুশি বাবা এবং আমি তাদের সমস্তকে আজীবন ছুটিতে আনতে চাই।" (Hyālō āmi 3 sundara chōṭa sbargadūtēra khuśi bābā ēbaṁ āmi tādēra samastakē ājĪbana chuṭitē ānatē cāi.) ("Hello I am the happy father of 3 beautiful little angels and I would like to bring them all on the vacation of a lifetime.") is wrongly labeled as "Other" while the correct intent is "Provide_info".
- Ambiguity in utterances: These types of errors occur when the utterances themselves are unclear. For the example, "Excellent! Can you see what is available in Athens?", the predicted intent is "City", but according to the context the correct intent is "Trip". Also in case of Hindi, the utterance "तीन किस्से एल्बम दिखाएं।"(Teen kisse elbam dikhaen.)("Show the three tales album.") is wrongly identified as "SearchScreeningEvent" while the correct intent is "SearchCreativeEvent".
- Unseen phrases: This type of error occurs when the words and phrases are not present during training. Example: In case of slots, the utterance "Maybe milan" is predicted with the slot label "O" while the correct label is "$B-city\_name$". For Bengali, the words ("ওয়াইল্ড কান্ট্রি") (Ōẏāilḍa kāṇṭri) ("Wild Country") in the utterance "এই গান- টি আমার প্লেলিস্টে ওয়াইল্ড কান্ট্রি যুক্ত করুন।" (Ēi gānaṭi āmāra plēliṣṭēōẏāilḍa kāṇṭri yukta karuna.)("Add this song to my playlist named Wild Country.") is wrongly labeled as ("O O") while the correct tags are ($B-playlist\ I-playlist$).
- Insufficient instances: These kind of errors occur when a particular class is under-represented. For example, the utterance "How many Canadian airlines international flights use J31?" is predicted as "Airlines" while the correct intent is "Quantity". Similarly for Hindi, "अमेरिकन एयरला- इंस पर मुझे उड़ानें दिखाएं जो सेंट लुइस के रास्ते सेंट पीटर्सबर्ग से ओंटारियो कैलिफोर्निया तक जाती हैं।" (Amerikan eyaralains par mujhe udaanen dikhaen jo sent luis ke raaste sent peetarsabarg se ontaariyo kailiphorniya tak jaatee hain.) ("Show me the flights on American Airlines which go from St. Petersburg to Ontario California by way of St. Louis.") the word कैलि-फोर्निया *(California)* is wrongly labeled as "$I-city\_name$" while the correct slot label is "$B-state\_name$". Also the word "सेंट लुइस" (*St. Louis*) is incorrectly tagged as "$B-city\_name\ I-city\_name$" while the correct tags are "$B-stoploc.city\_name\ I-stoploc.city\_name$".

Table 11
Analysis of multi-lingual multi-task model vs. Monolingual multi-task models (For Intent).

Utterance	Proposed multilingual multi-task Model	Monolingual English model	Monolingual Hindi model	Monolingual Bengali model
Find now and forever	SearchScreeningEvent	PlayMusic	–	–
मेरी प्लेलिस्ट में क्रैबफ़िश जोड़ें। (Meree plelist mein kraibafish joden.) (add the crabfish to my playlist.)	AddtoPlaylist	–	BookRestaurant	–
আপনি আমাকে ডালাস থেকে অর্থনীতি ভাড়া ফ্লাইট প্রদর্শন করতে পারেন। (Apani amake dalasa theke arthaniti bhara phlaita pradarsana karate parena?) (Can you show me the economy fares flights from dallas.)	Flight	–	–	Airfare
would like to know the aircraft on a flight from cleveland to dallas	Aircraft	Flight	–	–
इस पैकेज में क्या गतिविधियाँ शामिल हैं?? (Is paikej mein kya gatividhiyaan shaamil hain?) (What activities are included in this package?)	Amenity	–	Package_info	–
আপনি ক্যাল্যাগারি আগ্রহী হবে। (Apani kyaligari agrahi habe.) (Would you be interested in Calgary.)	City	–	–	Trip

Table 12
Statistical significance results.

Datasets	Models	English		Hindi		Bengali	
		Intent	Slot	Intent	Slot	Intent	Slot
ATIS	CNN	1.09E−065	2.55E−074	1.46E−070	1.83E−062	6.48E−066	1.52E−054
	LSTM	2.43E−064	2.61E−071	4.14E−067	9.99E−058	1.04E−060	1.79E−048
	GRU	1.17E−060	7.94E−068	7.91E−063	7.07E−051	4.24E−057	1.27E−043
	mBERT	2.01E−042	3.51E−043	1.79E−047	4.35E−054	6.73E−049	9.73E−045
	mBERT + Attn	1.52E−032	1.51E−039	7.65E−036	2.50E−044	7.17E−020	8.03E−033
TRAINS	CNN	1.32E−060	3.21E−052	8.66E−065	2.86E−045	2.83E−063	1.19E−046
	LSTM	4.84E−056	2.86E−040	1.04E−060	3.44E−044	3.83E−059	1.83E−041
	GRU	1.53E−053	8.03E−033	1.08E−054	7.16E−020	1.68E−050	5.42E−021
	mBERT	4.89E−046	6.98E−043	2.12E−045	1.09E−050	1.06E−047	6.80E−053
	mBERT + Attn	9.71E−017	1.47E−026	8.55E−029	1.41E−042	4.76E−016	8.60E−041
FRAMES	CNN	9.56E−074	5.33E−077	6.31E−071	4.47E−078	6.58E−073	1.03E−077
	LSTM	1.66E−070	1.43E−069	5.01E−066	3.22E−070	1.89E−069	4.90E−071
	GRU	7.93E−067	3.93E−070	3.34E−055	5.17E−061	2.14E−060	9.39E−060
	mBERT	4.61E−051	7.15E−052	3.94E−041	5.24E−047	1.28E−040	1.19E−046
	mBERT + Attn	4.28E−033	2.88E−042	2.47E−015	2.32E−039	1.36E−014	9.70E−037
SNIPS	CNN	2.16E−070	2.29E−074	9.76E−072	5.56E−073	6.32E−069	2.18E−069
	LSTM	2.04E−057	2.03E−068	2.48E−058	2.02E−065	6.54E−058	9.51E−065
	GRU	9.64E−059	9.51E−065	1.17E−061	1.76E−059	1.05E−061	1.47E−061
	mBERT	1.51E−039	2.86E−045	5.52E−039	3.28E−050	7.66E−036	3.99E−047
	mBERT + Attn	9.71E−017	2.92E−032	4.77E−023	1.28E−040	2.47E−015	1.34E−038

6.8. Statistical significance test

A statistical hypothesis test called Welch's t-test [120] is performed at a significance rate of 5% (0.05) to confirm whether the performance improvement in our proposed model is statistically significant. This is intended to prove that the best accuracy of our proposed method is statistically significant and has not happened by chance. The performance metric (accuracy) is obtained by 20 consecutive runs of each algorithm for the statistical test on all the datasets. We measure the p-values provided by Welch's t-test to compare two groups to assess the statistical significance of our approach as shown in Table 12. Hence, from the table, it can be concluded that the results are statistically significant.

7. Conclusion and future work

Conversational systems are an important application that is supposed to assist people in leading their lives comfortably in the upcoming years. With the advancement in technology, understanding the user for providing the correct answers to their queries is essential for building efficient dialogue agents. In this paper, we have proposed a multilingual multitask model for the two primary SLU tasks, i.e., slot filling and intent detection.

We have used a CNN and recurrent neural network with LSTM and GRU as basic cells for sentence representation as to the baseline models. We have used BERT in our final proposed model to encode the utterances that all languages share in order to learn the intricate details among the languages. Both the intent and the slot filling tasks share the representations learned from these models. We have captured the language and task information in our proposed methodology, which helps to model the utterance of different languages and share the information among the tasks through a common sentence encoder. Experiments on four datasets are performed to evaluate our proposed MLMT model. Empirical results show that the proposed multilingual multitask model exhibits superiority over the individual models for a particular task and language. It also outperforms state-of-the-art slot filling and intent detection tasks on all the datasets regardless of the domain or nature of the English language datasets. It also allows our model to correctly identify the intents and slots for different languages using multilingual word embedding. With the help of CRF, we can learn the dependency among labels for the slot filling task.

As future work, we plan to integrate semantic knowledge in order to model these SLU tasks as well. In addition, we would like to use different deep learning techniques such as memory networks and auto-encoders to capture contextual information for dialogue datasets.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research is supported by the Imprint 2C sponsored project titled "Sevak-An Intelligent Indian Language Chatbot". This research is also supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

Informed consent

Informed consent was not required as no human or animals were involved.

Human and animal rights

This article does not contain any studies with human or animal subjects performed by any of the authors.

References

[1] T. Young, F. Xing, V. Pandelea, J. Ni, E. Cambria, Fusing task-oriented and open-domain dialogues in conversational agents, in: AAAI, 2022, pp. 11622–11629.
[2] J. Ni, V. Pandelea, T. Young, H. Zhou, E. Cambria, HiTKG: Towards goal-oriented conversations via multi-hierarchy learning, in: Proceedings of AAAI, 2022, pp. 11112–11120.
[3] Y. Ma, K.L. Nguyen, F. Xing, E. Cambria, A survey on empathetic dialogue systems, Inf. Fusion 64 (2020) 50–70.
[4] W. Li, W. Shao, S. Ji, E. Cambria, BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis, Neurocomputing 467 (2022) 73–82.
[5] J. Wen, D. Jiang, G. Tu, C. Liu, E. Cambria, Dynamic interactive multiview memory network for emotion recognition in conversation, Information Fusion 91 (2023) 123–133.
[6] G. Tu, C. Liu, D. Jiang, J. Wen, E. Cambria, Context- and sentiment-aware networks for emotion recognition in conversation, IEEE Transactions on Artificial Intelligence 3 (5) (2022) 699–708.
[7] J. Schuurmans, F. Frasincar, Intent classification for dialogue utterances, IEEE Intell. Syst. 35 (1) (2020) 82–88.
[8] H. Xu, H. Peng, H. Xie, E. Cambria, L. Zhou, W. Zheng, End-to-End latent-variable task-oriented dialogue system with exact log-likelihood optimization, World Wide Web 23 (2020) 1989–2002.
[9] T. Young, V. Pandelea, S. Poria, E. Cambria, Dialogue systems with audio context, Neurocomputing 388 (2020) 102–109.
[10] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, M. Huang, Augmenting end-to-end dialogue systems with commonsense knowledge, in: AAAI, 2018, pp. 4970–4977.
[11] 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, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, 2018, pp. 3586–3596.
[12] 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, Brussels, Belgium, October 31 - November 4, 2018, 2018, pp. 633–639.
[13] X. Liu, P. He, W. Chen, J. Gao, Improving multi-task deep neural networks via knowledge distillation for natural language understanding, 2019, arXiv preprint arXiv:1904.09482.
[14] V. Sanh, T. Wolf, S. Ruder, A hierarchical multi-task approach for learning embeddings from semantic tasks, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, 2019, pp. 6949–6956.
[15] Y. Xu, X. Liu, Y. Shen, J. Liu, J. Gao, Multi-task learning with sample re-weighting for machine reading comprehension, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), 2019, pp. 2644–2655.
[16] N. Howard, E. Cambria, Intention awareness: Improving upon situation awareness in human-centric environments, Human-Centric Comput. Inf. Sci. 3 (9) (2013).
[17] J. Ni, T. Young, V. Pandelea, F. Xue, V. Adiga, E. Cambria, Recent advances in deep learning based dialogue systems: A systematic survey, Artif. Intell. Rev. (2022).
[18] 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.
[19] A.L. Gorin, G. Riccardi, J.H. Wright, How may I help you? Speech Commun. 23 (1–2) (1997) 113–127.
[20] P. Haffner, G. Tur, J.H. Wright, Optimizing SVMs for complex call classification, in: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '03, Hong Kong, April 6-10, 2003, Vol. 1, IEEE, 2003, pp. 632–635.
[21] G. Tur, D. Hakkani-Tür, L. Heck, S. Parthasarathy, Sentence simplification for spoken language understanding, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic, IEEE, 2011, pp. 5628–5631.
[22] 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.
[23] S. Yaman, L. Deng, D. Yu, Y.-Y. Wang, A. Acero, An integrative and discriminative technique for spoken utterance classification, IEEE Trans. Audio, Speech Lang. Process. 16 (6) (2008) 1207–1214.
[24] 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.
[25] J. Wu, Introduction to convolutional neural networks, National Key Lab for Novel Software Technology, Nanjing University, China, 2017.
[26] A. Karpathy, J. Johnson, L. Fei-Fei, Visualizing and understanding recurrent networks, 2015, arXiv preprint arXiv:1506.02078.
[27] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[28] S.V. Ravuri, A. Stolcke, Recurrent neural network and LSTM models for lexical utterance classification, in: INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015, 2015, pp. 135–139.
[29] J.-K. Kim, G. Tur, A. Celikyilmaz, B. Cao, Y.-Y. Wang, Intent detection using semantically enriched word embeddings, in: 2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, December 13-16, 2016, IEEE, 2016, pp. 414–419.
[30] M. Firdaus, S. Bhatnagar, A. Ekbal, P. Bhattacharyya, Intent detection for spoken language understanding using a deep ensemble model, in: PRICAI 2018: Trends in Artificial Intelligence - 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28-31, 2018, Proceedings, Part I, Springer, 2018, pp. 629–642.
[31] P. Jayarao, A. Srivastava, Intent Detection for code-mix utterances in task oriented dialogue systems, 2018, arXiv preprint arXiv:1812.02914.
[32] C. Xia, C. Zhang, X. Yan, Y. Chang, P.S. Yu, Zero-shot user intent detection via capsule neural networks, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, 2018, pp. 3090–3099.
[33] T.-E. Lin, H. Xu, Deep unknown intent detection with margin loss, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, 2019, pp. 5491–5496.
[34] G. Yan, L. Fan, Q. Li, H. Liu, X. Zhang, X.-M. Wu, A.Y. Lam, Unknown intent detection using Gaussian mixture model with an application to zero-shot intent classification, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, 2020, pp. 1050–1060.
[35] M. Firdaus, A. Kumar, A. Ekbal, P. Bhattacharyya, A multi-task hierarchical approach for intent detection and slot filling, Knowl.-Based Syst. (2019).
[36] A. Gupta, X. Li, S.K. Rallabandi, A.W. Black, Acoustics based intent recognition using discovered phonetic units for low resource languages, 2020, arXiv preprint arXiv:2011.03646.
[37] J.-G. Zhang, K. Hashimoto, W. Liu, C.-S. Wu, Y. Wan, P.S. Yu, R. Socher, C. Xiong, Discriminative nearest neighbor few-shot intent detection by transferring natural language inference, 2020, arXiv preprint arXiv:2010.13009.
[38] Y. Hou, Y. Lai, Y. Wu, W. Che, T. Liu, Few-shot learning for multi-label intent detection, 2020, arXiv preprint arXiv:2010.05256.

[39] H. Nguyen, C. Zhang, C. Xia, P.S. Yu, Dynamic semantic matching and aggregation network for few-shot intent detection, 2020, arXiv preprint arXiv:2010.02481.

[40] H. Perkins, Y. Yang, Dialog intent induction with deep multi-view clustering, 2019, arXiv preprint arXiv:1908.11487.

[41] T.-E. Lin, H. Xu, A post-processing method for detecting unknown intent of dialogue system via pre-trained deep neural network classifier, Knowl.-Based Syst. 186 (2019) 104979.

[42] A. Tyagi, V. Sharma, R. Gupta, L. Samson, N. Zhuang, Z. Wang, B. Campbell, Fast intent classification for spoken language understanding, 2019, arXiv preprint arXiv:1912.01728.

[43] A. McCallum, D. Freitag, F.C. Pereira, Maximum entropy Markov models for information extraction and segmentation, in: Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 -July 2, 2000, Vol. 17, (2000) 2000, pp. 591–598.

[44] C. Raymond, G. Riccardi, Generative and discriminative algorithms for spoken language understanding, in: INTERSPEECH 2007, 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, August 27-31, 2007, 2007, pp. 1605–1608.

[45] A. Moschitti, G. Riccardi, C. Raymond, Spoken language understanding with kernels for syntactic/semantic structures, in: IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2007, Kyoto, Japan, December 9-13, 2007, IEEE, 2007, pp. 183–188.

[46] R. López-Cózar, Using knowledge on word-islands to improve the performance of spoken dialogue systems, Knowl.-Based Syst. 88 (2015) 223–243.

[47] 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, 2013, pp. 2713–2717.

[48] 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: 2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, December 2-5, 2012, IEEE, 2012, pp. 210–215.

[49] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, F. Gao, Recurrent conditional random field for language understanding, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014, IEEE, 2014, pp. 4077–4081.

[50] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, Y. Shi, Spoken language understanding using long short-term memory neural networks, in: 2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014, IEEE, 2014, pp. 189–194.

[51] 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.

[52] 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 2017, New Orleans, la, USA, March 5-9, 2017, IEEE, 2017, pp. 5675–5679.

[53] Y. Shin, K.M. Yoo, S.-g. Lee, Slot filling with delexicalized sentence generation, in: Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018, 2018, pp. 2082–2086.

[54] 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, NEWS@ACL 2018, Melbourne, Australia, July 20, 2018, 2018, pp. 22–26.

[55] 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers, 2018, pp. 426–431.

[56] L. Qiu, Y. Ding, L. He, Recurrent neural networks with pre-trained language model embedding for slot filling task, 2018, arXiv preprint arXiv:1812.05199.

[57] B. Liu, I. Lane, Multi-domain adversarial learning for slot filling in spoken language understanding, 2017, arXiv preprint arXiv:1711.11310.

[58] 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 2018, Calgary, AB, Canada, April 15-20, 2018, IEEE, 2018, pp. 6049–6053.

[59] S. Zhu, K. Yu, Concept transfer learning for adaptive language understanding, in: Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018, 2018, pp. 391–399.

[60] K. Williams, Neural lexicons for slot tagging in spoken language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 2 (Industry Papers), 2019, pp. 83–89.

[61] W. Xu, B. Haider, S. Mansour, End-to-end slot alignment and recognition for cross-lingual NLU, 2020, arXiv preprint arXiv:2004.14353.

[62] Z. Liu, G.I. Winata, Z. Lin, P. Xu, P. Fung, Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems, in: Proceedings of the AAAI Conference on Artificial Intelligence (2020), Vol. 34, (05) 2020, pp. 8433–8440.

[63] Y. Hou, S. Chen, W. Che, C. Chen, T. Liu, C2C-GenDA: Cluster-to-cluster generation for data augmentation of slot filling, 2020, arXiv preprint arXiv:2012.07004.

[64] P. Xu, R. Sarikaya, Convolutional neural network based triangular crf for joint intent detection and slot filling, in: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013, IEEE, 2013, pp. 78–83.

[65] D. Guo, G. Tur, W.-t. Yih, G. Zweig, Joint semantic utterance classification and slot filling with recursive neural networks, in: 2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014, IEEE, 2014, pp. 554–559.

[66] B. Liu, I. Lane, 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, 2016, pp. 22–30.

[67] 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, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016, 2016, pp. 715–719.

[68] X. Zhang, H. Wang, 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 2016, New York, NY, USA, 9-15 July 2016, 2016, pp. 2993–2999.

[69] Y. Shi, K. Yao, H. Chen, Y.-C. Pan, M.-Y. Hwang, B. Peng, Contextual spoken language understanding using recurrent neural networks, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015, IEEE, 2015, pp. 5271–5275.

[70] B. Liu, I. Lane, 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, 2016, pp. 685–689.

[71] Y.-B. Kim, S. Lee, K. Stratos, Onenet: Joint domain, intent, slot prediction for spoken language understanding, in: 2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017, IEEE, 2017, pp. 547–553.

[72] A. Bapna, G. Tur, D. Hakkani-Tur, L. Heck, Sequential dialogue context modeling for spoken language understanding, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, SaarbrÜCken, Germany, August 15-17, 2017, 2017, pp. 103–114.

[73] 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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), 2, 2018, pp. 309–314.

[74] 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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), Vol. 2, 2018, pp. 753–757.

[75] 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: Neural Information Processing - 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part IV, Springer, 2018, pp. 647–658.

[76] 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 2018, Calgary, AB, Canada, April 15-20, 2018, IEEE, 2018, pp. 6034–6038.

[77] 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, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, 2018, pp. 2778–2790.

[78] C. Zhang, Y. Li, N. Du, W. Fan, P.S. Yu, Joint slot filling and intent detection via capsule neural networks, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers, 2019, pp. 5259–5267.

[79] A. Siddhant, A. Goyal, A. Metallinou, Unsupervised transfer learning for spoken language understanding in intelligent agents, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, 2019, pp. 4959–4966.

[80] L. Qin, W. Che, Y. Li, H. Wen, T. Liu, A stack-propagation framework with token-level intent detection for spoken language understanding, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, 2019, pp. 2078–2087.

[81] M. Chen, J. Zeng, J. Lou, A self-attention joint model for spoken language understanding in situational dialog applications, 2019, arXiv preprint arXiv:1905.11393.

[82] H.-Y. Kim, Y.-H. Roh, Y.-G. Kim, Data augmentation by data noising for open-vocabulary slots in spoken language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 3-5, 2019, Student Research Workshop, 2019, pp. 97–102.

[83] R. Gangadharaiah, B. Narayanaswamy, Joint multiple intent detection and slot labeling for goal-oriented dialog, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), 2019, pp. 564–569.

[84] Q. Chen, Z. Zhuo, W. Wang, BERT for joint intent classification and slot filling, 2019, arXiv preprint arXiv:1902.10909.

[85] G. Castellucci, V. Bellomaria, A. Favalli, R. Romagnoli, Multi-lingual intent detection and slot filling in a joint BERT-based model, 2019, arXiv preprint arXiv:1907.02884.

[86] E. Haihong, P. Niu, Z. Chen, M. Song, A novel Bi-directional interrelated model for joint intent detection and slot filling, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, 2019, pp. 5467–5471.

[87] H. Bai, Y. Zhou, J. Zhang, C. Zong, Memory consolidation for contextual spoken language understanding with dialogue logistic inference, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, 2019, pp. 5448–5453.

[88] D. Wu, L. Ding, F. Lu, J. Xie, SlotRefine: A fast non-autoregressive model for joint intent detection and slot filling, 2020, arXiv preprint arXiv:2010.02693.

[89] L. Qin, X. Xu, W. Che, T. Liu, Towards fine-grained transfer: An adaptive graph-interactive framework for joint multiple intent detection and slot filling, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, 2020, pp. 1807–1816.

[90] L. Qin, T. Liu, W. Che, B. Kang, S. Zhao, T. Liu, A Co-interactive transformer for joint slot filling and intent detection, 2020, arXiv preprint arXiv:2010.03880.

[91] J.G. FitzGerald, STIL–simultaneous slot filling, translation, intent classification, and language identification: Initial results using mBART on MultiATIS++, 2020, arXiv preprint arXiv:2010.00760.

[92] S. Louvan, B. Magnini, Simple is better! lightweight data augmentation for low resource slot filling and intent classification, 2020, arXiv preprint arXiv:2009.03695.

[93] M. Hardalov, I. Koychev, P. Nakov, Enriched pre-trained transformers for joint slot filling and intent detection, 2020, arXiv preprint arXiv:2004.14848.

[94] H. Li, A. Arora, S. Chen, A. Gupta, S. Gupta, Y. Mehdad, MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark, 2020, arXiv preprint arXiv:2008.09335.

[95] L. Qin, M. Ni, Y. Zhang, W. Che, CoSDA-ML: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP, 2020, arXiv preprint arXiv:2006.06402.

[96] J. Wang, K. Wei, M. Radfar, W. Zhang, C. Chung, Encoding syntactic knowledge in transformer encoder for intent detection and slot filling, 2020, arXiv preprint arXiv:2012.11689.

[97] J. Krishnan, A. Anastasopoulos, H. Purohit, H. Rangwala, Multilingual code-switching for zero-shot cross-lingual intent prediction and slot filling, 2021, arXiv preprint arXiv:2103.07792.

[98] Y. Hui, J. Wang, N. Cheng, F. Yu, T. Wu, J. Xiao, Joint intent detection and slot filling based on continual learning model, 2021, arXiv preprint arXiv:2102.10905.

[99] Q. Chen, Z. Zhuo, W. Wang, Q. Xu, Transfer learning for context-aware spoken language understanding, in: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2019, pp. 779–786.

[100] Y. Liu, F. Meng, J. Zhang, J. Zhou, Y. Chen, J. Xu, Cm-net: A novel collaborative memory network for spoken language understanding, 2019, arXiv preprint arXiv:1909.06937.

[101] S. Zhu, R. Cao, K. Yu, Dual learning for semi-supervised natural language understanding, IEEE/ACM Trans. Audio, Speech, Lang. Process. 28 (2020) 1936–1947.

[102] L. Qin, M. Ni, Y. Zhang, W. Che, Y. Li, T. Liu, Multi-domain spoken language understanding using domain-and task-aware parameterization, 2020, arXiv preprint arXiv:2004.14871.

[103] P. Zhou, Z. Huang, F. Liu, Y. Zou, PIN: A novel parallel interactive network for spoken language understanding, 2020, arXiv preprint arXiv:2009.13431.

[104] P. Wei, B. Zeng, W. Liao, Joint intent detection and slot filling with wheel-graph attention networks, 2021, arXiv preprint arXiv:2102.04610.

[105] M. Artetxe, G. Labaka, E. Agirre, A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, 2018, pp. 789–798.

[106] X. Chen, C. Cardie, Unsupervised multilingual word embeddings, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, 2018, pp. 261–270.

[107] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2017) 135–146.

[108] 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, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A Meeting of SIGDAT, A Special Interest Group of the ACL, 2014, pp. 1724–1734.

[109] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), 2018, pp. 4171–4186.

[110] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, 2016, arXiv preprint arXiv:1609.08144.

[111] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, 2015, pp. 19–27.

[112] I. Tenney, D. Das, E. Pavlick, Bert rediscovers the classical nlp pipeline, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, 2019, pp. 4593–4601.

[113] C. Zhu, M. Zeng, X. Huang, Sdnet: Contextualized attention-based deep network for conversational question answering, 2018, arXiv preprint arXiv:1812.03593.

[114] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019, arXiv preprint arXiv:1907.11692.

[115] G. Lample, A. Conneau, Cross-lingual language model pretraining, 2019, arXiv preprint arXiv:1901.07291.

[116] 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, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, SaarbrÜCken, Germany, August 15-17, 2017, 2017, pp. 207–219.

[117] P.A. Heeman, J.F. Allen, The TRAINS 93 Dialogues, Technical Report, Rochester univ ny dept of computer science, 1995.

[118] S.L. Smith, D.H. Turban, S. Hamblin, N.Y. Hammerla, Offline bilingual word vectors, orthogonal transformations and the inverted softmax, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

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

[120] B.L. Welch, The generalization ofstudent's' problem when several different population variances are involved, Biometrika 34 (1/2) (1947) 28–35.