Multi-Lingual Attention based Multi-Intent Detection in Dialogue System

Mauajama Firdaus*, Ankit Kumar, Asif Ekbal, and Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Patna
Patna, India
(mauajama.pcs16,ankit.mtcs17,asif,pb)@iitp.ac.in

Abstract Natural language understanding (NLU) is an essential and critical module of every dialogue system. One of the key tasks of this module is to identify the intent and help the user in achieving the desired goals. In this work, we propose a multi-lingual, multi-intent detection model that can handle user utterances having multiple intents belonging to different languages. We employ an attention-based Recurrent neural network (RNN) for detecting multiple intents from a given user utterance. To model the dependency across the different languages, we use both cross-lingual and mono-lingual attention mechanisms. We evaluate our proposed approach on English and Hindi datasets having multiple intents. As there is no existing Hindi dataset, we create the data for Hindi by manually translating the English dataset. Experimental results show the effectiveness of our proposed approach with an absolute improvement of 20 F-score points over the baselines.

Keywords: Multi-lingual, Multi-intent, Natural Language Understanding, Deep Learning

1 Introduction

The dialogue system is an example of human-computer interaction. It contains different modules that focus on understanding the user and generate responses to assist the user in achieving their desired goals. Spoken language understanding (SLU) is a vital component of the dialogue systems which deal with the critical task of understanding the language spoken by the user. For the proper functioning of the dialogue system, it is essential to perceive the intended meaning from the user’s utterance. One of the primary tasks of the SLU module is intent detection. Intent detection is a standard problem of utterance classification, as the errors made by the intent detection module often lead to wrong system responses. Hence, it is crucial that designing of the intent module is done carefully so that the users can attain their desired objectives. In this work, we focus on identifying multiple intents in a given user utterance which is a typical case of multi-label classification.

For every dialogue system, it is essential to not only identify the different intents in an utterance but also to understand the different languages a user can choose to converse with the system. In our current work, we propose a deep learning model that identifies the intents from both English and Hindi user utterances. As opposed to a single intent detection module, a multi-intent detection module can be used to identify multiple intents belonging to different domains. For example, “When is the flight and how far is the restaurant from the airport?” Here, the intents of the utterance belong to two different domains “Flight, Restaurant”. Also, for quick and fast processing it is essential for a system to understand the various intentions present in a given utterance, thereby solving the objectives of the user at the earliest by simultaneously providing correct responses to the user.

<table>
<thead>
<tr>
<th>Example</th>
<th>Intents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Would you like to increase your budget or adjust the dates</td>
<td>budget, date</td>
</tr>
<tr>
<td>कितने लोग एक साथ यात्रा करेंगे और कब</td>
<td>लोगों की संख्या, (time)</td>
</tr>
<tr>
<td>(How many people will travel together and when)</td>
<td>(no_of_ppl, date)</td>
</tr>
<tr>
<td>Can you tell me where the nearest city is and what are the available flights</td>
<td>city, flight</td>
</tr>
<tr>
<td>आप कहां से हैं और आप कहां जाने के बारे में सोच रहे हैं</td>
<td>प्रवास, गतिविधि</td>
</tr>
<tr>
<td>(Where are you from and where are you thinking of going)</td>
<td>(departure, destination)</td>
</tr>
<tr>
<td>Can you give me the prices and hotel ratings</td>
<td>price, rating</td>
</tr>
</tbody>
</table>

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 two languages, namely English and Hindi. In Table 1, we show
the multi-intent multi-lingual examples of user utterances from the dataset used for the task. The motive for employing this setup was to share the information between the languages and improve the performance of the overall model by capturing different intents simultaneously. The key contributions of our current research can be summarized as follows:

- We create a resource for the task of multi-intent detection in a multi-lingual scenario (for both English and Hindi). We create Hindi dataset by manually translating English into Hindi.
- We propose a multi-lingual attention based multi-intent model that can handle different languages as well as detect multiple intents from a user utterance simultaneously.

The rest of the paper is structured as follows: In section 2, we present a brief review of the existing literature. In Section 3, we explain the proposed methodology followed by the dataset description in Section 4. The experimental results are presented in Section 5 along with detailed analysis including error analysis. In Section 6, we show the concluding remarks along with future directions of research.

2 Related Work

Initial works on intent detection focused on the use of various traditional machine learning approaches such as support vector machine (SVM) [1], maximum entropy classifier [2] and boosting techniques [3, 4]. With the ability to combine both feature extraction and classification in the learning process, deep learning framework has helped in improving the performance of this task. Variants of recurrent neural networks such as long short term memory (LSTM) [5] have been employed in [6, 7] for identifying the intents. In [8], hashing techniques along with LSTM was employed to take care of the out-of-vocabulary (OOV) words for classifying the intents.

Convolutional neural networks (CNN) was employed in [9] for detecting the intents of a user search query. An ensemble-based deep learning architecture using Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) was employed in [10] for intent detection on ATIS dataset. Recently, multi-lingual intent detection has been done for English and Japanese languages using a deep LSTM network with adversarial training [11, 12]. Intent detection in a code-mixed environment has also been proposed in [13]. Our work differentiates from these previous works in a way that we capture multiple intents present in an utterance as opposed to almost all the systems which focus on single intent.

Due to the correlation between the intent detection and slot filling tasks, these SLU tasks have been modeled together using various deep learning architectures. Multi-task learning has been performed by employing various RNN structures in [14–17]. Previously, a few works have been carried out for multi-intent detection [18, 19]. Our work is different from these in the sense that it can encompass different languages (English and Hindi, in this case) as is the demand of the current day situation.

3 Methodology

Classification of multiple intents in a given utterance is a standard case of multi-label classification. Here, given a multi-intent dataset \((D)\), having user utterance \((S) = [w_1, \ldots, w_n]\), in which \(n\) is the total number of words in \(S\). The set of intent labels is defined by \(Y \in \{y_1, \ldots, y_l\}\), where \(l\) is the total number of intents present in \(D\). For a given user utterance \(S\), the goal is to identify the possible number of intent labels \(y_{i_1}, \ldots, y_{i_k}\) present in the utterance, where \(k \leq l\). The overall architecture of the model is given in Figure 1.

In this section, we illustrate our multi-lingual multi-intent model (MLMI) to detect the different intents present in a user utterance which could belong to either of the languages.

3.1 Embedding Layer

In the embedding layer, we convert every word token to its high dimensional vector representation by concatenating its character and word-level embedding.

**Character embedding:** Character embedding layer is responsible for mapping each word to a high-dimensional vector space. Following [20], we get character level embedding of each word token. The outputs of CNN are max-pooled over entire width to obtain a fixed size vector for each word.

**Word embedding:** Word embedding layer maps each word to a high dimensional vector. We use pre-trained FastText [21] model to get the fixed word embedding of each word in both languages i.e., English and Hindi.

So, the final input to the model can be defined as:

\[
x_i = c_{e_{c_i}} \oplus e_{w_i}
\]

where, \(\oplus\) denotes the concatenation, \(e_{c_i}\) and \(e_{w_i}\) denotes the character level and word level embeddings respectively.
3.2 Utterance Encoder

After representing the words using the embedding layer, we apply utterance encoder (CNN or RNN) to model interaction between words.

**RNN based Utterance Encoder:** Recurrent Neural Network (RNN) [22] is considered as an extension of feed-forward neural network, that handles the sequences of variable length by using a recurrent hidden state that captures the information of the previous states as well. Long Short Term Memory (LSTM) [5] is a special kind of RNN, which can efficiently learn long-term dependencies by solving vanishing gradient problem. The hidden layers of the bi-directional LSTM can be defined as follows:

\[
\begin{align*}
    \overrightarrow{h}_t & = \text{LSTM}(x_t, \overrightarrow{h}_{t-1}) \\
    \overleftarrow{h}_t & = \text{LSTM}(x_t, \overleftarrow{h}_{t+1})
\end{align*}
\]

where \( x_t \) is the input and \( h_t \) is the hidden state of LSTM at time \( t \). The forward and backward directions are denoted by \( \rightarrow \) and \( \leftarrow \), respectively. The forward and backward hidden states are concatenated to give the bidirectional hidden state \( h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t] \) at time \( t \).

**CNN based Utterance Encoder:** The primary CNN sentence encoder, useful for many natural language processing (NLP) tasks, is used to encode the utterances. Let \( x_i \in \mathbb{R}^d \) be the \( d \)-dimensional word vector of the \( i^{th} \) word in the utterance. Now, \( x_i \) is fed to a convolutional layer that involves a filter \( k \in \mathbb{R}^{pd} \), which is applied to a window of \( p \) words in an utterance to produce a new feature. A hidden state \( h_i \) is generated from a window of words \( x_{i:i+p-1} \) by

\[
    h_i = f(k \ast x_{i:i+p-1} + b)
\]

where \( b \) is the bias term, and \( f \) is a non-linear function, which, in our case is Relu [23]. We use \( n \) different filters to perform convolution operation and then denote the resulting feature as \( h_i \). The hidden state of the entire utterance can be written as \( H = [h_1, h_2, \ldots, h_n] \) by applying the filter to each possible window of the words present in the utterance. In multi-lingual scenario we get hidden state representation \( H_j = [h_{j1}, \ldots, h_{jn}] \) for given \( j^{th} \) language.
3.3 Multi-Lingual Attention Layer:

Multilingual Attention is computed to link and fuse information across languages. Our model adopts two kind of attention mechanisms to exploit inter and intra language patterns and dependencies. Firstly, we employ cross-lingual attention which detects counter information in the other language. Secondly, we use mono-lingual attention which selects a useful feature for a given language. Suppose \( j \) indicates a language and \( k \) is another language (such that \( k \neq j \)).

**Cross-Lingual Attention:** To model dependencies across the languages, we employ cross-lingual attention after obtaining the utterance representation from either CNN or LSTM. To calculate attention distribution for \( i \)-th word of the utterance in \( j \)-th language, \( h_{ji} \) will be used as a query vector over the hidden states \( H_k \) of \( k \)-th language. Formally, \( \hat{h}_{ji} \) is defined as the weighted sum of hidden state representation of the \( k \)-th language.

\[
\hat{h}_{ji} = \alpha_a H_k^T
\]

where \( \alpha_a \) is a cross-lingual attention score of each hidden state which is defined as :

\[
\alpha_a = \text{softmax}(h_{ji}^T W_a H_k)
\]

where \( W_a \) is a trainable parameter. Finally, we concatenate \( \hat{h}_{ji} \) and \( h_{ji} \) to form the new hidden state representation \( c_{ji} \).

**Mono-Lingual Attention:** We employ mono-lingual attention mechanism to model language specific characteristics in user utterance in each language. After obtaining enhanced hidden state representation \( c_{ji} \) from cross-lingual attention we aim to aggregate all the information into single vector \( M_j \). Mono-lingual attention works similar to cross-lingual attention. \( M_j \) is calculated as weighted sum of \( C_j = [c_{j1}, \ldots, c_{jn}] \).

\[
\beta_a = \text{softmax}(W_a^T C_k), M_j = \beta_a C_{k}^T
\]

3.4 Training and Inference:

After obtaining the utterance representation \( M_j \) from multi-lingual attention, Sigmoid layer is used to detect multiple intents in an utterance. The equation gives the final probability of the intent \( \hat{y}_j \):

\[
\hat{y}_j = \frac{1}{1 + e^{-W^T M_j + b}}
\]

\( W \) and \( b \) are trainable parameters. The parameters are optimised by minimising the cross-entropy between the predicted \( (\hat{y}_j) \) and reference probabilities \( (y_j) \):

\[
L_{CE} = -\sum_{j=1}^{|\mathcal{Y}|} (y_j \log \hat{y}_j + (1 - y_j) \log (1 - \hat{y}_j))
\]

4 Dataset and Experiment

**Dataset:** The FRAMES corpus [24] consists of dialogue conversations, and we have manually annotated every utterance with all the possible intents. Three annotators proficient in the English language were assigned to annotate the corpus with intent classes. The corpus consists of 1369 human-human dialogues. Each dialogue has an average of 15 turns. There are 18623 utterances in the training set, 2660 utterances in the validation set and 5321 utterances in the test set. We use 24 different intent classes. We observe the multi-rater Kappa agreement ratio of approximately 80%, which may be considered reliable. As there was no dataset for intent detection in Hindi, we create it by manually translating our English dataset with the help of three native Hindi annotators. The number of utterances in the train, validation and test data are the same as that of the English dataset. Also, the number of intents are the same for both languages\(^1\).

**Training Details:** All the implementations are done using the Keras framework. We also use 300-dimensional cross-lingual embeddings [25] as input to our model. In our model, the intermediate layers use ReLU activation function while the last layer uses the Sigmoid activation function. We use the dropout [26] with probability 0.45. We initialize the model parameters randomly using a Gaussian distribution with Xavier scheme [27]. The hidden size for LSTM layer is 128. We employ AMSGrad [28] as the optimizer for model training to mitigate the slow convergence issues. We use uniform label smoothing with \( \epsilon = 0.1 \) and perform gradient clipping when gradient norm is over 5. Categorical cross-entropy is employed to update the model parameters.

\(^1\) The dataset will be made available after acceptance.
5 Results and Discussion

In this section, we report the results along with the necessary analysis. We report subset accuracy, hamming loss, and F1 score as the performance measure for the multi-intent detection task. Hamming loss \[29\] identifies the fraction of misclassified labels. Hence, lower the loss better is the model performance. While subset accuracy \[29\] computes the fraction of the prediction being identical to the true label set.

Table 2: Experimental results of different models; CNN: is the baseline model having both character and word embeddings as input to the model; LSTM: is the second baseline with both character and word embedding; LSTM + CE: is the model having cross-lingual embeddings (CE) as input; LSTM + MLA: is the model using only Mono Lingual Attention; LSTM + CLA: is the model using only Cross Lingual Attention, MLMI model: is the proposed Multi-lingual multi-intent model having both the attention mechanisms; Individual model: is the multi-intent model of a particular language

<table>
<thead>
<tr>
<th>Model</th>
<th>English</th>
<th></th>
<th>Hindi</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F-score</td>
<td>Hamming Loss</td>
<td>Subset Accuracy</td>
<td>F-score</td>
</tr>
<tr>
<td>CNN</td>
<td>0.38</td>
<td>0.047</td>
<td>0.30</td>
<td>0.41</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.41</td>
<td>0.045</td>
<td>0.31</td>
<td>0.42</td>
</tr>
<tr>
<td>LSTM + CE</td>
<td>0.48</td>
<td>0.040</td>
<td>0.35</td>
<td>0.47</td>
</tr>
<tr>
<td>LSTM + MLA</td>
<td>0.51</td>
<td>0.035</td>
<td>0.40</td>
<td>0.50</td>
</tr>
<tr>
<td>LSTM + CLA</td>
<td>0.57</td>
<td>0.031</td>
<td>0.44</td>
<td>0.53</td>
</tr>
<tr>
<td>MLMI model</td>
<td>0.61</td>
<td>0.029</td>
<td>0.49</td>
<td>0.58</td>
</tr>
<tr>
<td>Individual model</td>
<td>0.58</td>
<td>0.032</td>
<td>0.46</td>
<td>0.55</td>
</tr>
</tbody>
</table>

Results: Evaluation results of the MLMI model along with with CNN and LSTM models, and the language-specific individual models (where we train for one language at a time) are shown in Table 2. In comparison to CNN, LSTM shows better performance. Hence, further experiments are performed using LSTM as the sentence encoder. From the table, it is evident that the proposed MLMI model outperforms the other models as well as the language-specific models. These improvements were found to be statistically significant \[2\]. We observe a remarkable performance improvement with the sharing of information for both the languages using multi-lingual attention mechanism. For English, our proposed MLMI model shows an improvement of 20% (0.20) in F1-score compared to the baseline LSTM model. Whereas for Hindi, our model shows an increase of 16% (0.16) compared to the baseline.

To show the effectiveness of our proposed multi-lingual multi-intent model, we also compare its performance with the individual language-specific (non-shared) models. The individual models were implemented such that at a time only one language (either English or Hindi) utterances were given as input to the model. As presented in Table 2, our MLMI model has better performance for all the metrics concerning the individual models.

Error Analysis: To provide a detailed analysis of the results, we perform both quantitative and qualitative error analysis. In Table 3, we show the confusion matrix for English. From the table, we can infer that most of the labels have got confused with the intent label “other”. The intent label “city” has not been predicted correctly for any of single utterance. In Table 4, we present the confusion matrix for the Hindi dataset. The intent label “trip” is mainly confused with “provide_info”. The intent label “trip” is confused with other labels such as “greet, other”.

Due to the extended length of utterances, many times the intent of the utterance is expressed in the latter part of the utterance causing misclassification. Some misclassification also occurs due to less representation of some intent labels in the dataset. For example, “Do you prefer a 3.5 star hotel or a 4 star hotel” is incorrectly labeled as “rating” but the original intent label for this utterance is “hotel”. In the case of multi-intent classification, the model makes errors by identifying only a single intent. For example, “How much is the trip to Fukuoka and what are the dates?”, the predicted label is only “price” while the actual labels are “price, date”. With the presence of some key intent words, the model has predicted multiple intents while the utterance had single intent. For example, “could I see some better rated hotels”, the intent label is “hotel” but the model has predicted both “rating, hotel” because of the presence of the word “rated” which often occurs in utterances having “rating” as the label.

For Hindi, the system makes errors for cases such as in the following example, “कौन सा होटल वही है लेकिन लागत उच्च है” (what are the different hotels and from which cities are they), the predicted intent is “flight” while the original intents were “city, flight”. The reason behind such errors is due to the reason that the model fails to capture the intent appearing in the latter half of the utterance. For the given example, “होटल वही है लेकिन लागत उच्च है” (For the same hotel what are the available dates), the predicted intent label

\[^2\] we perform statistical significance tests \[30\], and it is conducted at 5% (0.05) significance level
"hotel" while the correct intent label is "date". This type of errors occurs because of the head-word of the utterance, which is same as one of the intent labels. In this case, the headword "hotel" is labeled as the final intent of the utterance.

In some cases, multiple intents are identified while the utterance had single intent. For example, "क्या आप 6 दिसंबर को पेȝरस šĥथान करना चाहते हȈ" (Can you travel to Paris on 6th December?), the model predicted the intents to be "city, date, departure" while the original intent of the utterance is "date". To visualize the effect of attention, we present the heatmap for both English and Hindi utterances in Figure 2 and Figure 3, respectively. From the figure, it is evident that the use of multi-lingual attention helps in detecting the true intents for both the languages. For example, in the utterance "can you tell me more about your travel dates and budget", the focus is on the words "travel dates" and "budget".
is given to the keywords “budget, date”, thereby helping in correct classification. Similarly, for Hindi, we can see that attention helps in improving the classification performance of the model. Overall, including multi-lingual attention in the proposed model ensures better identification of intents.

6 Conclusion and Future Work

NLU module has been an integral part of every dialogue systems with intent detection being one of its primary tasks. In this paper, we have proposed a multi-lingual attention based multi-intent model for identifying the intents of the user utterance for both English and Hindi languages. We have employed two types of attention mechanism for capturing language-specific information as well as shared information of both the languages using multi-lingual attention mechanism. Experimental results show that the proposed model performs better than the baselines and the individual models for all the evaluation metrics.

In future, along with the opportunity of extending the architectural design and training methodologies to enhance the performance of our system, we would also like to continue this work by incorporating the other tasks of NLU such as slot filling, dialogue act classification in this setup. Also, we would like to model a code-mixed version of our dataset since it is closer to the real-world scenario.

7 Acknowledgment

Asif Ekbal acknowledges Young Faculty Research Fellowship (YFRF), funded by Visvesvaraya PhD scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being executed by Digital India Corporation (formerly Media Lab Asia).

References


30. Welch, B.L.: The generalization of student’s problem when several different population variances are involved. Biometrika 34(1/2), 28–35 (1947)