Chapter 4

Aspect Term Extraction for Hindi

In the ABSA problem, the sentiment expression's attention transitions from a full sentence or text to either an entity or a particular aspect of that entity. The process involves segmenting the text data and determining its sentiment based on its many features. Opinion terms linked to aspect terms enhance comprehension of sentiment. The aspect term refers to the specific target of opinion that is clearly mentioned in the provided language. In the context of e-commerce, a specific product might serve as the target, with its qualities such as price and size being considered as relevant aspects. A deep learning based Aspect Term Extraction(ATE) model is presented. A comparison with existing techniques for Hindi data is performed.

4.1 Framework

The deep BiLSTM model for ATE consists of two BiLSTM layers and one attention layer. The embedding layer converts individual words or tokens from a text into a consistent vector of predetermined dimensions. The last output layer employs the softmax function and is basically accountable for aspect word extraction by means of BIO tag categorization. The utilization of two BiLSTM layers enables the model to effectively capture complex interdependencies and contextual

information (Section 1.5) included in the input data. BiLSTMs analyze sequences in both the forward and backward directions and capture context from both sides of each word, allowing for a more thorough comprehension of the context.

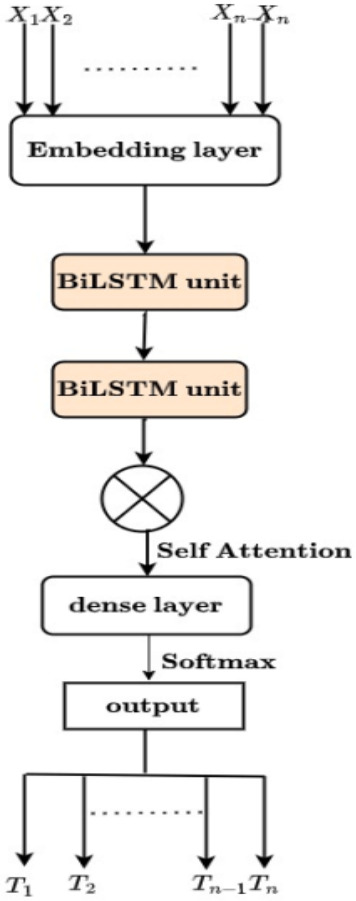

Figure 4-1: Block diagram for ATE model

The proposed BiLSTM-based model captures the probabilistic linkages present in the input text. It optimizes the probability of accurately predicting the aspect term. This is especially advantageous for tasks such as ATE, where the significance of words and their connections within sentences is vital. Let X_1, X_2,..., X_{n-1}, X_n represent a sequence of words, and let T_1, T_2……, T_{n-1}, T_n represent the corresponding output BIO tags. The arrangement of layers in the model, together with the utilization of the Softmax activation function, is illustrated in figure 4-1.

4.1.1 Data pre-processing

The Hindi sentences are sourced from the IITP-I dataset, which is freely accessible. Data pre-processing is necessary. The IITP-I dataset underwent BIO-labelling as a pre-processing step [58]. Subsequently, distinct labels, namely B (Begin), I (Inside), and O (Outside) are allocated. The sentence including BIO-labels is shown in table 4.1.

Table 4.1: Review and predicted labels-example I

Review	इसकी	ऑडियो	क्वालिटी	शानदार	है।
BIO labels	O	B	I	O	O

Table 4.2, shows the B(Begin), I(Inside) and O(Outside) tag percentage in IITP-I.

Table 4.2: Some basic statistics

Tag	**Percentage**
B	4.899 %
I	3.496 %
O	91.603 %

IITP-I dataset consists of Hindi reviews of twelve domains. The Hindi reviews distribution between these domains is presented in figure 4-2.

4.2 Experimental Design

The suggested model is implemented using the Python programming language, with the aid of the Keras, Scikit-learn and TensorFlow libraries. The Google Colab virtual platform functions as the development environment. The hardware

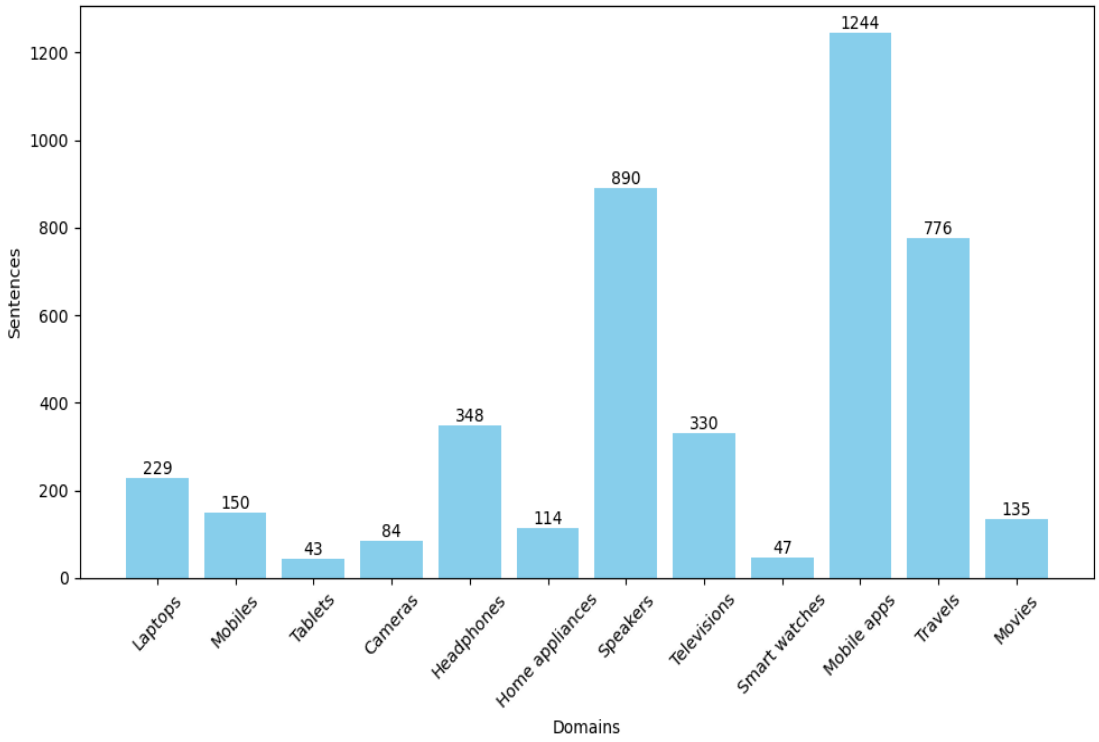

Figure 4-2: Hindi sentence distribution between domains

criteria consists of a CPU that is readily accessible, equipped with two virtual CPUs, and accompanied by 12 GB of RAM. The Matplotlib software is utilized to view and interpret the results. The Deep BiLSTM model in figure 4-1 utilizes the following neural network parts.

1. **Deep BiLSTM Neural Network**

   There are two LSTM units in BiLSTM. One LSTM unit is forward LSTM and another LSTM unit is backward LSTM. The input sequence is processed in a forward direction by the forward LSTM and a reverse direction by the backward LSTM. The architectural configuration of the LSTM unit is depicted in figure 4-3. A LSTM unit computes the hidden state H_t at time t.

   Forget gate:

$$F_t = \sigma(w_f \cdot [H_{t-1}, X_t] + b_f) \tag{4.1}$$

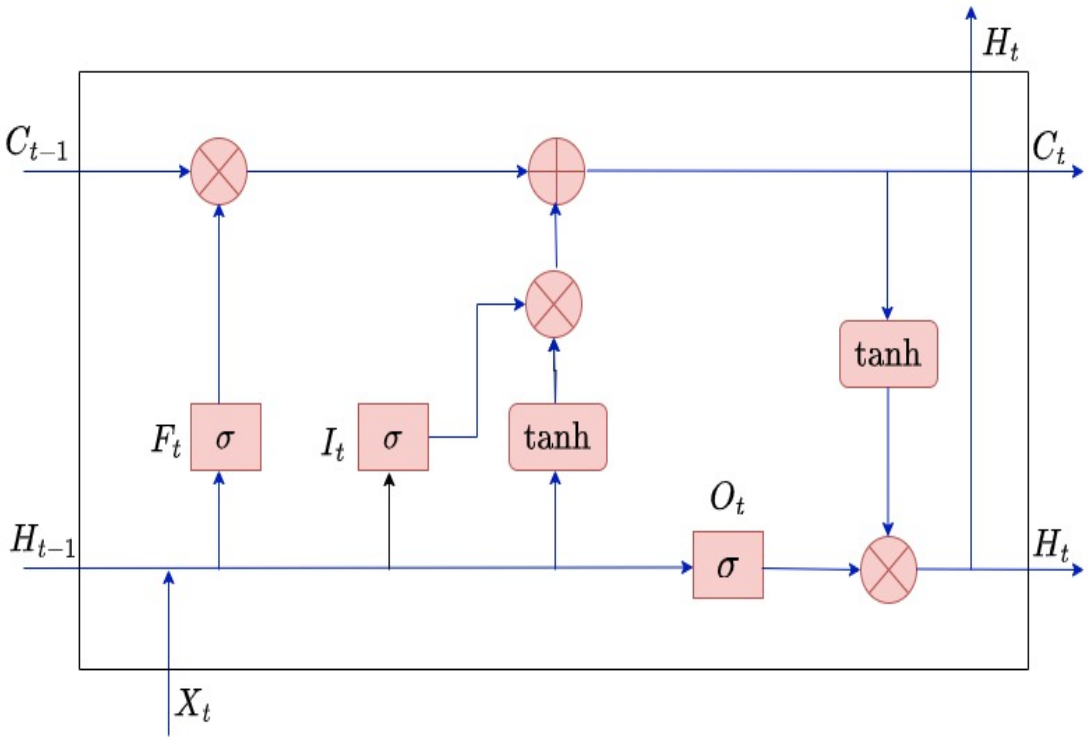

Figure 4-3: LSTM unit architecture

Update (Input) gate:

$$I_t = \sigma(w_i \cdot [H_{t-1}, X_t] + b_i) \tag{4.2}$$

Output gate:

$$O_t = \sigma(w_o \cdot [H_{t-1}, X_t] + b_o) \tag{4.3}$$

The equation 4.1 presents the *forget gate*. The forget gate is the first block in an LSTM network. Across time steps, the forget gate decides which data is retained or removed from the cell state. The purpose of the *forget gate* is to assist the LSTM network in remembering only the information that is essential and discarding the rest. For the *input gate*, equation 4.2 indicates what new data will be stored in the cell state. The *output gate* is represented by equation 4.3 and is utilized to activate the LSTM unit block's final output at the time stamp. Equation 4.4 represents the cell state, equation 4.5 shows

candidate cell state, and final result is represented by equation 4.6:

$$C'_t = \tan(w_c \cdot [H_{t-1}, X_t] + b_c) \tag{4.4}$$

$$C_t = F_t \cdot C_{t-1} + I_t \cdot C'_t \tag{4.5}$$

$$H_t = O_t \cdot \tan^{-1}(\tanh) \tag{4.6}$$

The function σ is sigmoidal. A word vector at time stamp t is denoted by X_t. The cell's gate vectors are F_t, I_t and O_t. The cell's weights and bias parameters are denoted by w and b.

Figure 4-4 represents the BiLSTM unit architecture.

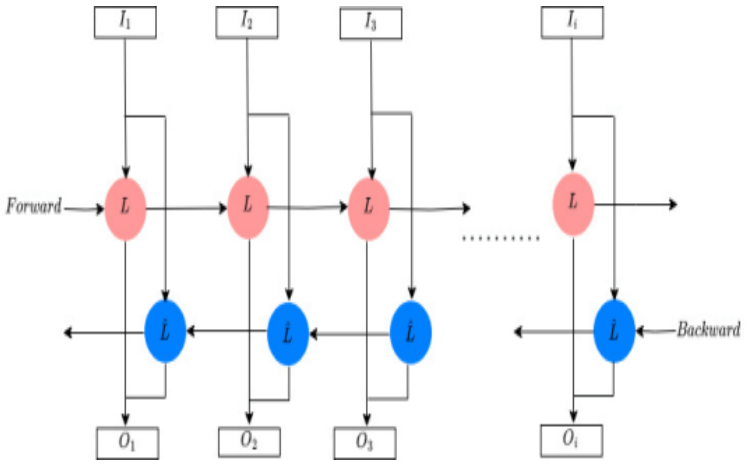

Figure 4-4: BiLSTM unit architecture

The sentence is processed from X_1 to X_n by the forward LSTM unit. The sentence is processed using backward LSTM units from X_n to X_1. The forward LSTM extracts a word feature $\overrightarrow{H}_l$ for word X_t. Backward LSTM yields a word feature $\overleftarrow{H}_l$ for the same word X_t. Here's how the feature H is computed:

H is equal to

$$\overrightarrow{H}_l \oplus \overleftarrow{H}_l$$

Here, the concatenation operation is shown by $\oplus$.

2. **Attention layer**

   The attention layer is employed to ascertain the distinct influence of every word on the entirety of the text. The attention mechanism adds a weight w_i to each word feature H_i. A weighted sum function is used to calculate the hidden states in order to build a feature vector r for the hidden phrase. The mathematical representation of these stages is given by equation 4.7, 4.8 and 4.9.

$$E_i = \tanh(W_h H_i + b_h), \quad E_i \in [-1, 1] \tag{4.7}$$

$$w_i = \frac{e^{E_i}}{\sum_{t=1}^{N} e^{E_t}}, \quad \sum_{i=1}^{N} w_i = 1 \tag{4.8}$$

$$r = \sum_{i=1}^{N} w_i h_i \tag{4.9}$$

   The weights and bias w_h and b_h are assigned by the attention layer.

3. **Time-distributed Dense Layer**

   A crucial component of recurrent neural networks(RNNs), including LSTMs and BiLSTMs, is the time-distributed dense layer. It is essential for processing sequential data, particularly in applications involving time series analysis, NLP and sequence-to-sequence modeling. It serves a vital role in connecting the recurrent layers' output to completely connected output layers that have the appropriate number of outputs. The Time-distributed layer applies the same dense layer independently to each time step in a sequence. Equation 4.10 and equation 4.11 describe the mathematical operation that occurs at a single step within a sequence:

$$Y_t = X_t \cdot W^T + b \tag{4.10}$$

$$O = f(Y_t) \tag{4.11}$$

   At time t, X_t is sent in, Y_t is sent out. There is a weight matrix for thick layers called W^T. Layers' bias vector is b. The fully connected layer's output

is shown by O.

After being pre-processed, the IITP-I dataset was used to train the deep BiLSTM model in this work. As explained in section 4.1.1, data pre-processing is done. A test set, training set and validation set were generated from the IITP-I dataset. For these groups, 70% are set aside for training, 20% for validation, and 10% for testing. There are 3791 sentences in Hindi in the training set, the validation set was made up of 1073 sentences, and the test set is made up of 553 sentences.

Eight samples are processed in each training to fine-tune. An early stopping technique was incorporated to enhance the training process and prevent over-fitting. In this method, the model would stop training if the validation loss did not go down by less than 1.0×10^{-5} for three epochs in a row. It prevents the model from continuing training unnecessarily when further improvements were unlikely.

The loss estimation for the model is performed using the categorical cross-entropy method. To avoid the model from over-fitting, dropout [38] is implemented. During each stage of training, dropout randomly selects and deactivates a small number of neurons. For gradient optimization, the root mean square propagation algorithm [52] is used. The learning rate employed is 0.001. Random search strategies are employed to identify the most suitable hyper-parameters for optimization. Table 4.3 provides a concise overview of the individual optimized hyper-parameters that are used throughout the development of the model.

Table 4.3: Hyper-parameters for the ATE model

Parameter	Description
The input's longest length	80
Size of mini-batch	08
Learning rate	1.0×10^{-3}
Sentences in the training set	3791
Sentences in the validation set	1073
Sentences in the test set	553
Validation steps	135
Activation function	Softmax
Dropout rates (D_1, D_2)	0.8, 0.4

D_1: Dropout for BiLSTM-1

D_2: Dropout for BiLSTM-2

4.3 Evaluation Results and Analysis

The performance of the suggested model is tested in the experiments, with a particular emphasis on its accuracy. The model exhibits a remarkable accuracy of 91.27% without any decrease in F1-score when compared to other models. The model's high accuracy demonstrates its ability to accurately classify and assign labels to the data. The behavior of the loss function is depicted in figure 4-5. Loss function measures prediction error, guiding model optimization by minimizing the difference from ground truth. It is seen that the model is acquiring knowledge and enhancing its ability to do tasks. A sharp decline in the loss function implies that the model is quickly approaching a solution that accurately fits the data. Nevertheless, it is crucial to check the model's performance on a validation set to ensure that it remains consistent with the training data. Over-fitting refers to a model that has been excessively specialized to the training data and lacks the ability to generalize well to unfamiliar data. As previously stated, early pausing can alleviate over-fitting by monitoring the validation loss. The model's accuracy, along with the initial behavior of the loss function, indicates that it is successful

and suitable for ATE. It demonstrates both high accuracy and efficient learning throughout training.

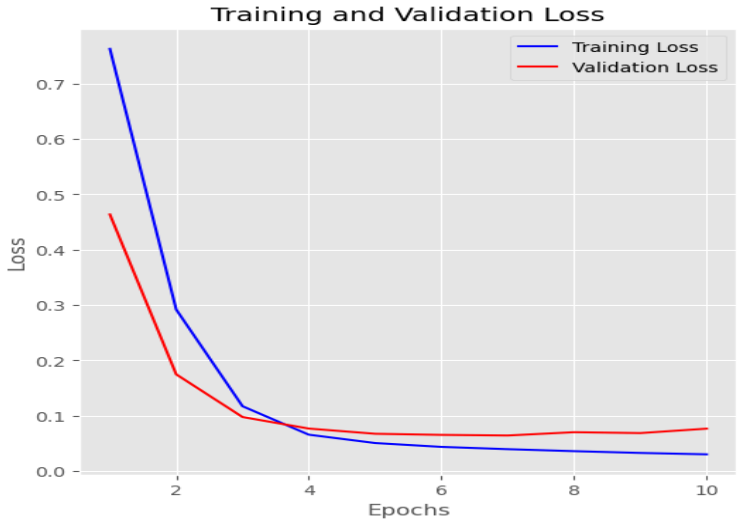

Figure 4-5: Loss function vs epochs

Once the model has gained knowledge about the distinctive features of the labeled classes, the validation loss lowers and the validation accuracy increases. The classifier, which has been trained, makes predictions of BIO tags for the Hindi reviews. The following are the actual and expected BIO tags for a small number of Hindi reviews. It accurately predicts single-word aspect phrases, as seen in the example provided in table 4.4. The classifier faces difficulties when attempting to forecast multi-word aspect words. These multi-word aspect terms denote particular phrases or word combinations. It accurately recognizes the multi-word in example-III in table 4.5. The inclusion of conjunctions and prepositions within the aspect word poses a difficulty. The system should accurately detect and recognize entire multi-word aspect words. It recognizes the aspect phrase **"वॉयस"** alone. The aspect word **"और वीडियो कॉल्स"** is still unmarked. The right prediction of the B(Begin) tag may be observed in example-IV shown in table 4.6.

Table 4.4: Review and predicted labels-example II

Review	व्हाट्सप्प	जैसा	फीचर	है।
Actual	O	O	B	I
Predicted	O	O	B	I

Table 4.5: Review and predicted labels-example III

Review	इसकी	ऑडियो	क्वालिटी	शानदार	है।
Actual	O	B	I	O	O
Predicted	O	B	I	O	O

Table 4.6: Review and predicted labels-example IV

Review	प्रतिस्पर्धियों	पर	भारी	पड़ने	के	लिए	वॉयस	और	वीडियो	कॉल्स	की
	जरूरत	है।									
Actual	O	O	O	O	O	O	B	I	I	I	O
	O	O									
Predicted	O	O	O	O	O	O	B	O	O	O	O
	O	O									

High accuracy indicates improved interpretability and accurate prediction of the majority groups. A low F1-score indicates the presence of class imbalance and the difficulty in accurately classifying the minority classes. Figure 4-2 illustrates the presence of class imbalance in the IITP-I dataset. Training with a larger dataset can enhance its performance. Future advancements of this work would require expansion of the dataset. G-mean for class B(Begin) is 0.216. G-mean for I(Inside) is 0.1835 and 0.2256 for O(outside) tags. Less G-mean value implies poor model performance on minority (B/I) tags. A comparative analysis based on accuracy and F1-score criteria is performed to compare the suggested strategy with existing approaches shown in table 4.7. Only studies that have a comparable experimental design for the Hindi language are taken into account for this comparison. The comparison of these works is presented in table 4.7. Reported studies indicate a relatively low F1-score for ATE in Hindi. BiLSTM units in our

approach enable the detection of intricate linkages present in textual data. An attention mechanism enhances the acquisition of a more advanced analysis of Hindi texts. These strategies enhance the accuracy of ATE for Hindi in comparison to existing methods. Our research demonstrates the a high precision in extracting aspect terms, achieving a remarkable F1-score of 43.16 in comparison to existing methods.

4.4 Summary

The ATE model, which includes an attention layer [111], is designed to extract aspect phrases from Hindi review sentences. The study included trials using the IITP-I, yielding encouraging results that surpassed those achieved by prior documented approaches. The model exhibited high precision with an accuracy of 91.27% and an F1-score of 43.16, highlighting its efficacy. The BiLSTM units were crucial in capturing the complex connections between aspect terms and the words around them in the sentences. The attention layer prioritizes crucial input items and enhances the accuracy of predictions and computational efficiency in ATE. Domains with a small number of reviews exhibit statistically negligible results when analyzed on a domain-by-domain basis. To improve the performance of the model, particularly the F1-score, we aim to analyze domain-specific features for ATE and address the challenge of processing multi-word expressions.

Table 4.7: Comparison of aspect term extraction methods for Hindi

References	Method	Dataset	Accuracy	F1-score
Akhtar et al [2]	CRF based	IITP-I	54.05 %	41.07
Gandhi and Attar [36]	BiLSTM based	IIP-I	—	44.49
Bhattacharya et al. [16]	Seq2Seq4ATE based	IITP-I	—	35.04
Bhattacharya et al. [16]	Seq2Seq4ATE based	Own dataset	—	68.61
Shrivastav and Kumar [101]	GRU based	IITP-I	88.02 %	—
Yadav et al [120]	LSTM based	Hindi product review	87.0 %	—
Rani and Kumar [90]	Dependency parser based	Hindi movie review	83.2 %	—
Our model	Attention-based BiLSTM	IITP-I	**91.27 %**	**43.16**