INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Check for updates

OPEN ACCESS

Received: 01-11-2023

Accepted: 09-01-2024

Published: 14-02-2024

Citation: Gupta A, Sharma U (2024) Deep Learning-Based Aspect Term Extraction for Sentiment Analysis in Hindi. Indian Journal of Science and Technology 17(7): 625-634. https://doi.org/10.17485/IJST/v17i7.2766

* **Corresponding author**.

ashwani@tezu.ernet.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Deep Learning-Based Aspect Term Extraction for Sentiment Analysis in Hindi

Ashwani Gupta[1]*, Utpal Sharma[2]

1 Research Scholar, Department of Computer Science and Engineering, Tezpur University, Tezpur, Assam, India
2 Professor, Department of Computer Science and Engineering, Tezpur University, Tezpur, Assam, India

Abstract

Objectives: Aspect terms play a vital role in finalizing the sentiment of a given review. This experimental study aims to improve the aspect term extraction mechanism for Hindi language reviews. **Methods:** We trained and evaluated a deep learning-based supervised model for aspect term extraction. All experiments are performed on a well-accepted Hindi dataset. A BiLSTM-based attention technique is employed to improve the extraction results. **Findings:** Our results show better F-score results than many existing supervised methods for aspect term extraction. Accuracy results are outstanding compared to other reported results. Results showed an outstanding 91.27% accuracy and an F–score of 43.16. **Novelty:** This proposed architecture and the achieved results are a foundational resource for future studies and endeavours in the field.

Keywords: Sentiment analysis; Aspect based sentiment analysis; Aspect term extraction; Deep Learning; Bi LSTM; Indian language; Hindi

1 Introduction

The World Wide Web allows individuals to express their opinions through text-based mediums - primarily online reviews and social media channels. The volume of these online reviews is witnessing a daily surge, making it a substantial source of information. While English remains a global language, there has been a notable uptick in the availability of online reviews and opinions in Indian languages, attributable to the growing number of Indian internet users. Recognizing that the ultimate sentiment cannot be accurately gauged solely through English content, there is a compelling need to devote attention to sentiment analysis in Indian languages.

Hindi stands as a prominent and extensively spoken language across India. Notably, web pages in Hindi are experiencing rapid growth, with numerous websites now delivering content in this language. Furthermore, users are actively sharing their opinions and providing reviews in Hindi. Consequently, there arises a critical need to assess sentiment in Hindi-language reviews.

Sentiment analysis primarily focuses on sentiment analysis at the document, sentence, and aspect levels. Aspect-based sentiment Analysis (ABSA), also known as aspect-level sentiment analysis, is fine-grained.

Within the Aspect-Based Sentiment Analysis (ABSA) problem, the focus of sentiment expression shifts from a complete sentence or document to either an entity or a specific aspect of an entity. It divides the text data and defines its sentiment based on its aspects. Opinion terms associated with aspect terms help to understand the sentiment better. The aspect term is the opinion target explicitly appearing in the given text. For example, in the E-Commerce domain, the target can be a particular product, and its attributes, like price and size, serve as related aspects.

There has not been much work done on Hindi aspect term extraction. We present an aspect term extraction approach for Hindi based on deep learning. This approach involves probabilities at different stages. Text data is, by nature, sequential, with significant information in word order. RNN[(1)] and BiLSTM[(2)] networks are examples of deep learning-based neural components that are ideally suited for aspect-term extraction tasks since they are built to capture dependencies in sequential data.

1.1 Related Work

The significant works done for aspect term extraction and aspect-based sentiment analysis in general and in the Hindi language are discussed in the literature survey[(3–8)]. Md Shad Akhtar et al.[(9)] reported the first supervised approach to the aspect term extraction task; they created an annotated dataset of high quality and built a machine learning model for sentiment analysis to show practical usage of the dataset. They used conditional random field (CRF) based probabilistic classification for aspect term extraction. They reported an F-score of 41.07 and 54.05% accuracy. Accuracy was too low for the said system. Hetal Gandhi and Vahida Attar[(10)] presented a detailed study of aspect term extraction sub-tasks for Hindi with CRF and Bi-LSTM classifiers. The CRF-based model attained an F-measure of 44.54, and the Bi-LSTM classifier-based model obtained an F-measure of 44.49 on Md Shad Akhtar et al.[(9)] dataset.Accuracy was not mentioned in the reported results.

Bhattacharya et al.[(11)] proposed a novel Seq2Seq4ATE architecture for aspect-term extraction tasks. They performed experiments on Md Shad Akhtar et al.[(9)] and their datasets. They reported a 35.04 F-score on Md Shad Akhtar et al.[(9)] dataset and 68.61 on their developed dataset. Accuracy was not reported in the results, and the F-score was lower in the results of the Md Shad Akhtar et al.[(9)] dataset. Kush Shrivastava and Shishir Kumar[(12)] enabled a gated recurrent unit (GRU) network to capture the semantic and syntactic relations between Hindi words and classify them into sentiment classes. The observed accuracies were 65.96% for the CNN-SVM approach, 72.01% for the RNN approach, and 88.02% for the proposed approach. Vandana Yadav et al.[(13)] reported their sentiment analysis on product reviews in Hindi. The experiment showed 87% accuracy. Sujata Rani and Prateek Kumar[(14)] presented a dependency parser-based sentiment analysis method. They experimented on the Hindi movie dataset and reported 83.2% accuracy for sentiment analysis.

Hindi is a free-order language[(15)]. The subject, object and verb can be in any order. A BiLSTM[(2)] unit is capable of understanding order dependence. Sequential data is processed both forward and backward. Aspect terms can consist of one or more words. By giving each word in a sentence sequence a distinct attention weight, the attention layer[(16)] handles sequences of varying length. Our paper proposes a novel deep learning-based supervised model with two BiLSTM units and an attention layer for aspect term extraction. It provides an outstanding 91.27% and 43.16 F-score against the accepted Md Shad Akhtar et al. dataset[(9)]. The paper's organization is as follows: Section 2 presents crucial details regarding the Md Shad Akhtar et al.[(9)] dataset and our methodologies in this study. This section not only elucidates the theoretical underpinnings of our approach but also offers an in-depth description of the proposed method. Section 3 focuses on our experimental setup, performance metrics, and discussions related to our findings. Lastly, Section 4 concludes the paper and references.

2 Methodology

The proposed deep BiLSTM model for aspect-term extraction includes two Bidirectional Long Short-Term Memory (BiLSTM) layers and one attention layer. The embedding layer transforms words or tokens of a sentence into a continuous vector of fixed size. The final output layer utilizes the softmax function and is primarily responsible for aspect term extraction through BIO tag classification.

Using two BiLSTM layers allows the model to capture intricate dependencies and contextual information within the input data. Bidirectional LSTMs process sequences in both forward and backward directions, enabling a more comprehensive understanding of the context. The proposed BiLSTM-based model captures probabilistic relationships within the input text. It maximizes the likelihood of correct aspect term prediction. This is particularly beneficial for tasks like aspect term extraction, where the context of words and their relationships within sentences are crucial.

Suppose X_1,X_2 ….. ,X_{n-1}, X_n are word sequence and T_1, T_2…….,T_{n-1}, T_n are output BIO tags for respective words. The sequence of layers in the model combined with the Softmax activation function employed, are depicted in Figure 1. Hindi dataset neural elements are explained in detail in the remaining section.

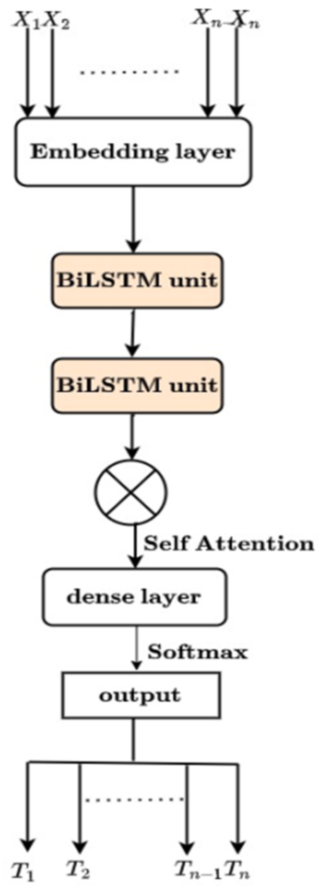

Fig 1. Block Diagram for aspect term extraction model

2.1 Dataset and pre-processing

Hindi sentences are taken from a freely available Review Sentiments Dataset[9] – Hindi made openly available by IIT-Patna. The dataset is in XML (Extensible Markup Language) format. These Hindi review sentences belong to 12 different domains. These domains are (i) Laptops, (ii) Mobiles, (iii) Tablets, (iv) Cameras, (v) Headphones, (vi) Home appliances, (vii) Speakers, (viii) Televisions, (ix) Smartwatches, (x) Mobile apps, (xi) Travels and (xii) Movies. The distribution of Hindi sentences between these domains is depicted in Figure 2.

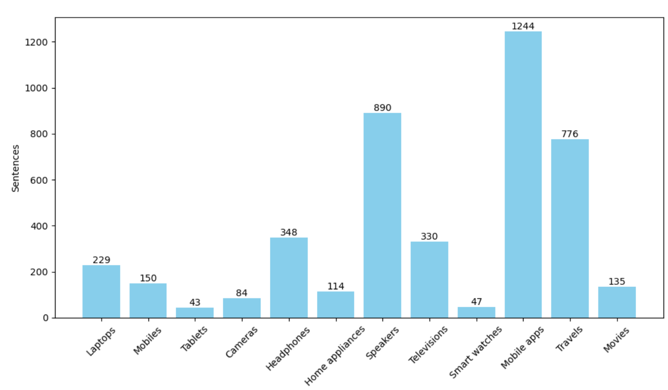

Fig 2. Hindi sentence distribution between domains

Two instances of the dataset with their structure are depicted in Figure 3. The <sentences> node represents the root node of the XML file. It contains sentences of the review as its children, i.e. <sentence>. An id is associated for uniquely identifying each <sentence>. It indicates domain information. Each sentence has two children, namely <text> and <aspectTerms>. The <text> node consists of one review sentence, whereas <aspectTerms> contains zero or more <aspectTerm> nodes. These nodes handle aspect term information. Each <aspectTerm> node holds four attributes: 'term', 'from', 'to' and 'polarity'. Attribute 'term' defines the aspect term represented by the current node, attribute 'from' indicates the starting character position of the aspect term in the sentence, and attribute 'to' indicates the end character position of the aspect term in the sentence. Attribute 'polarity' stores the sentiment towards the 'term', which is either 'pos'(positive), 'neg'(negative) or 'neu'(neutral).

Labelled Structure

```
<sentences>
<sentence id="mob_1396">
   <text>बॅक रबरयुक्त प्लास्टिक से बनी है, जिससे डिवाइस सिर्फ प्लास्टिक का ही ना दिखकर
प्रिमियम स्मार्टफोन का लूक रखता है।</text>
  </sentence>
  <sentence id="mob_1397">
   <text>साथ ही इसकी बनावट इर्गोनोमीकली है और यह हाथ और जेब में अच्छी तरह से फीट
बैठता है, और एक हाथ से इस्तेमाल करने के लिए आसान है।</text>
   <aspectTerms>
    <aspectTerm from="12" polarity="pos" term="बनावट" to="18"/>
   </aspectTerms>
  </sentence>
--------------
--------------
</sentences>
```

Fig 3. Hindi dataset labelled structure

The labelled sentences presented here belong to the 'mobile' domain. Sentence 1 with sentence id "mob_1396" has no aspect term. Sentence 2 with sentence id "mob_1397" has one aspect term. A data pre-processing is required. BIO–labelling[17] was performed on the IIT Patna Dataset as a pre-processing part. After that, different labels, B(Begin), I(Inside) and O(Outside) were assigned. A review sentence with BIO- labels is as follows.

	इसकी	ऑडियो	क्वालिटी	शानदार	है
BIO encoding	O	B	I	O	O

Table 1 displays some of the IIT-Patna dataset's basic metrics.

Table 1. Basic statistics

Metrics	IIT Patna Dataset
Sentences	5417
Aspect terms	4509
Tokens	96140
Average of aspect terms per sentence	0.81
% of Bs	4.899
% of Is	3.496
% of Os	91.603
Number of domains	12
Maximum number of words in any sentence	76

2.2 The following neural network elements are used to develop the proposed Deep BiLSTM model

2.2.1 Deep BiLSTM Neural Network

BiLSTM[2] consists of two LSTM[18] units: forward LSTM and backward LSTM. The architectural structure of the LSTM unit is illustrated in Figure 4.

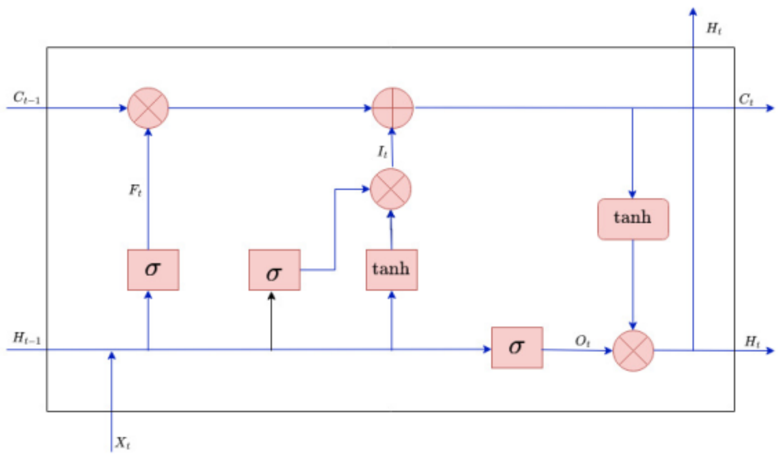

Fig 4. LSTM unit architecture

An LSTMunit calculates the hidden state H_t at time t. The following equations are used:
Forget gate:

$$F_t = \sigma\left(w_f.[H_{t-1}, X_t] + b_f\right)$$

Update (Input) gate:

$$I_t = \sigma\left(w_i.[H_{t-1}, X_t] + b_i\right)$$

Output gate:

$$O_t = \sigma\left(w_o.[H_{t-1}, X_t] + b_o\right)$$

The first equation is for the forget gate. The forget gate is the first block in an LSTM (Long Short-Term Memory) network. It determines which information is kept or discarded from the cell state across time steps. The forget gate's job is to help the LSTM network keep only the crucial information while forgetting the irrelevant ones. The second equation is for the input gate, which shows what new information will be stored in the cell state. The third one is for the output gate, which is used to activate the final output of the LSTM unit block at the time stamp. The following equations represent the cell state, candidate cell state, and final result:

$$C_t^{'} = tan\left(w_c\left[H_{t-1}, X_t\right] + b_c\right)$$

$$C_t = F_t.C_{t-1} + I_t.C_t^{'}$$

$$H_t = O_t.tan^{-1}tanh$$

σ is a sigmoidal function. X_t is a word vector at the time stamp t. F_t, I_t, O_t and O_t are gate vectors of the cell. $W$and b are the weights and bias parameters of the cell.

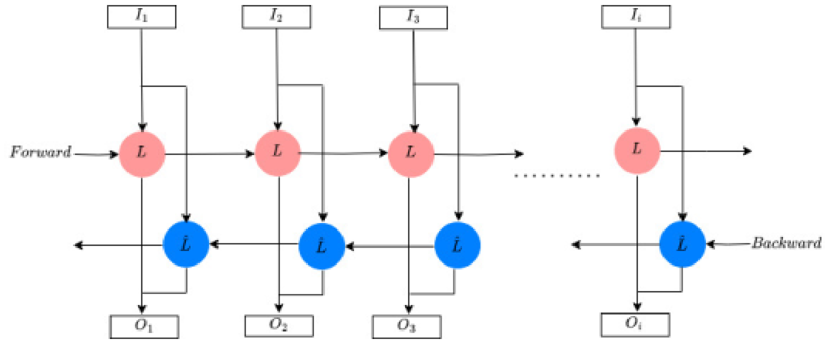

Fig 5. BiLSTM unit architecture

The BiLSTM unit obtains word features $H = (H_1, H_2, H_3, \ldots\ldots .H_n)$ concatenated from both directions. Figure 5 represents the BiLSTM unit architecture.

Forward LSTM unit processes the sentence from X_n to X_1. Backward LSTM units process the sentence from X_n to X_1. For word X_t, forward LSTM obtains a word feature $\overrightarrow{H_l}$. For the same word X_t, backward LSTM obtains a word feature $\overleftarrow{H_l}$. The feature H is calculated as follows:

$H = \overrightarrow{H_l} \oplus \overleftarrow{H_l}$

Here $\oplus$ denotes the concatenation operation.

2.2.2 Attention layer

The attention layer [16] is employed to find the contribution of each word to the whole sentence. The attention mechanism assigns a weight w_i to each word feature H_i. A weighted sum function computes the hidden states to generate a hidden sentence feature vector r. These steps are expressed mathematically as follows:

$$E_i = tanh\,(W_h H_i + b_h)\ ,\ E_i \in [-1, 1]$$

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

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

W_h and b_h are weights and bias assigned by the attention layer.

2.2.3 Time-Distributed dense layer

The Time-Distributed dense [19] layer is a crucial component in recurrent neural networks, such as LSTMs or BiLSTMs, for processing sequential data, particularly in tasks like sequence-to-sequence modelling, time series analysis, or natural language processing. It serves a vital role in connecting the output of recurrent layers to fully connected output layers with the desired number of outputs. The Time-Distributed layer applies the same dense layer to each time step independently. Here is the mathematical representation of the operation for a one-time step within a sequence:

$$Y_t = X_t.W^T + b$$

$$O = f\,(Y_t)$$

X_t is input at timestamp t, Y_t is output at timestamp t. W^T is a weight matrix for dense layers. b is the bias vector for the dense layer. O represents the output of the fully connected layer.

3 Results and discussion

The deep BiLSTM model in this study was trained using a dataset described in section 2. Three subsets were created from the IIT dataset: a test set, a validation set, and a training set. The distribution of these subsets was set at 70% for training, 20% for validation, and 10% for testing. Consequently, the training set was constructed with 3,791 Hindi sentences, the validation set comprised 1,073 sentences, and the test set included 553 sentences.

3.1 Experiment setup

The proposed model was implemented in Python, using the Keras, Scikit-learn, and TensorFlow libraries, and the development environment was the Google Colab virtual platform. The hardware specifications included an Intel Xeon CPU with two virtual CPUs and 12GB of RAM. The Matplotlib library was employed to visualize and interpret the results.

The training process was fine-tuned, with 8 samples processed in each training iteration. An early-stopping approach was integrated to optimize the training process and prevent overfitting. This approach enabled the model to halt training if the validation loss did not improve by a value less than $1.0X10^{-5}$ for three consecutive epochs. It safeguarded the model from training for an extended period when further performance improvements were improbable.

The model's loss estimation was conducted using the categorical cross-entropy approach. An efficient regularization technique, Dropout[20], is used to prevent the model from overfitting. At each layer of training, dropout skews a few neurons randomly.

The root mean square propagation algorithm[21] is used for gradient optimization. The learning rate used is $1.0X10^{-3}$. To determine the best appropriate hyperparameters, random search techniques are used for the hyperparameter tuning. Table 2 summarizes the specifics of the optimized hyperparameters that were utilized in developing the model.

Table 2. Hyperparameters for the proposed model

Parameter	Description
Maximum length of the input	80
Mini-batch size	08
Learning rate	$1.0X10^{-3}$
Size of the training dataset	3791
Size of the validation dataset	1073
Size of the test dataset	553
Validation steps	135
Activation function	Softmax
Dropout rates (D_1, D_2)	0.8,0.4

D_1: Dropout for Bi LSTM -1
D_2: Dropout for Bi LSTM-2

3.2 Performance metrics used for model evaluation

The deep BiLSTM model's performance is assessed using the measures[22] listed below:

1. **Accuracy:** It is ratio of correctly classified aspect terms to the total number of aspect terms as determined by a classifier. It measures how well the classifier correctly classifies aspect terms within the total set of aspect terms.
2. **Precision:** Precision is the ratio of the number of correctly classified aspect terms to the total number of aspect terms classified as positive (correctly or incorrectly) by the classifier. It measures the accuracy of the classifier specifically in terms of identifying aspect terms, indicating how many of the positively classified terms were indeed relevant.
3. **Recall:** This is the ratio of correctly identified aspect terms to the overall number of aspect terms present. Recall is also called sensitivity.
4. **F-Score:** It displays the harmonic average of precision and recall. It attempts to balance precision and recall.

3.3 Result Analysis

The suggested model's performance is evaluated in the experiments, with a primary focus on assessing its accuracy. The model demonstrates an impressive accuracy of 91.27% without a drop in F-score compared to other models. This high accuracy indicates that the model correctly classifies and labels the data. The loss function's behaviour is visualized in Figure 6. This suggests that the model is learning and improving its performance, which is typical behaviour during the early stages of training. A steep decrease in the loss function indicates that the model is rapidly converging to a solution that fits the data well. However, monitoring the model's performance on a validation set is essential to ensure it stays within the training data. A model that needs to generalize better to unknown data but has grown overly specialized to the training set is said to be overfitting. As mentioned earlier, early stopping can help mitigate overfitting by monitoring the validation loss. The accuracy achieved by the model, combined with the initial behaviour of the loss function, suggests that the model is effective and well-suited for aspect term extraction, demonstrating both high accuracy and efficient learning during the training process.

Fig 6. Loss function vs epochs

After the model has acquired knowledge about labelled class characteristics, the validation loss decreases while the validation accuracy improves.

The trained classifier predicts BIO tags for Hindi reviews. Actual and predicted BIO tags for a few Hindi reviews are shown below. It correctly predicts one-word aspect terms. The classifier encounters problems when it tries to predict multi-word aspect terms. These multi-word aspect terms refer to specific phrases or combinations of words. The presence of conjunctions and propositions within the aspect term makes it challenging. In the last Hindi review, the system needs to identify complete multi-word aspect terms. It only identifies the aspect term "वॉयस" The remaining part of the aspect term "और वीडियो कॉल्स" remains unmarked. It predicts the B(Begin) tag correctly.

Review	व्हाट्सप्प जैसा फीचर है।
Actual	O O B I
Predicted	O O B I
Translated	There is a feature like WhatsApp.
Review	इसकी ऑडियो क्वालिटी शानदार है।
Actual	O B I O O
Predicted	O B I O O
Translated	Its audio quality is fantastic.
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 shows better interpretability and correct prediction of the majority classes. A low F-score shows class imbalance and a challenge to classify the minority classes correctly. Class imbalance in the IIT-Patna dataset is shown in Figure 2. It can improve with training with a larger dataset. The future direction of this work includes larger dataset development. A comparative analysis is also conducted, comparing the proposed method with existing approaches. Only studies with a similar experimental setup for the Hindi language are considered for this comparison. The comparison of such forms is given in Table 3 based on accuracy and F-score parameters. Reported works show a low F-score for aspect term extraction for Hindi. BiLSTM units make it possible to identify complex relationships found in text. An attention mechanism helps to develop a more proficient comprehension of Hindi. Compared to current techniques, these improve aspect term extraction accuracy for Hindi. Our work presents the highest accuracy for aspect term extraction with a 43.16 F-score.

Table 3. Comparison with previous Hindi language research methods used

References	Method	Dataset	Accuracy	F-score
(9)	CRF based	IIT Patna	54.05%	41.07
(10)	Bi-LSTM based	IIT Patna	——–	44.49
(11)	Seq2Seq4ATE based	IIT Patna	——–	35.04
(11)	Seq2Seq4ATE based	Own Dataset	——–	68.61
(12)	GRU based	IIT Patna	88.02%	——
(13)	LSTM based	Hindi product review	87.0%	——
(14)	Dependency parser based	Hindi movie review	83.2%	——
This work	**Attention-based BiLSTM**	**IIT Patna**	**91.27%**	**43.16**

4 Conclusion

The proposed BiLSTM model, with an attention layer, extracts aspect terms from Hindi review sentences. The research involved experiments on the IIT Patna dataset, and the results were remarkable and better than those of other reported methods. The model demonstrated exceptional accuracy of 91.27% and achieved a 43.16 F-score, underscoring its effectiveness. The BiLSTM units played a pivotal role in capturing the intricate relationships between aspect terms and the surrounding words within the sentences. The attention layer focuses on essential input elements and improves prediction accuracy and computational efficiency in aspect term extraction. A few domains with few reviews show statistically insignificant results when viewed domain-wise. The outcomes could be enhanced by employing unsupervised techniques or by including a few domain-specific features. In order to enhance model performance especially F-score, we would like to look at domain-specific characteristics for aspect-term extraction and work on handling multi-word.

5 Acknowledgement

The authors would like to thank the editors and reviewers. Ashwani Gupta is grateful to MJP Rohilkhand University, Bareilly, Uttar Pradesh, India for granting him study leave under QIP scheme to pursue PhD program at Tezpur University, Tezpur, Assam, India. He is also thankful to AICTE for providing fellowship for pursing Ph.D. at Tezpur University, Tezpur, Assam, India.

References

1) Orvieto A, Smith S, Gu A, Fernando A, Gülçehre Ç, Pascanu R, et al. Resurrecting Recurrent Neural Networks for Long Sequences. 2023. Available from: https://doi.org/10.48550/arXiv.2303.06349.
2) Bidirectional LSTM in NLP. 2023. Available from: https://www.geeksforgeeks.org/bidirectional-lstm-in-nlp/.
3) Kumar D, Gupta A, Gupta VK, Gupta A. Aspect-Based Sentiment Analysis using Machine Learning and Deep Learning Approaches. *International Journal on Recent and Innovation Trends in Computing and Communication*. 2023;11(5s):118–138. Available from: https://doi.org/10.17762/ijritcc.v11i5s.6636.
4) Bordoloi M, Biswas SK. Sentiment analysis: A survey on design framework, applications and future scopes. *Artificial Intelligence Review*. 2023;56(11):12505–12560. Available from: https://doi.org/10.1007/s10462-023-10442-2.
5) Shrivash BK, Verma D, Pandey P. An Effective Framework for Sentiment Analysis of Hindi Sentiments Using Deep Learning Technique. *Wireless Personal Communications*. 2023;132:2097–2110. Available from: https://doi.org/10.1007/s11277-023-10702-y.
6) Pathak A, Kumar S, Roy PP, Kim BG. Aspect-Based Sentiment Analysis in Hindi Language by Ensembling Pre-Trained mBERT Models. *Electronics*. 2021;10(21):1–15. Available from: https://doi.org/10.3390/electronics10212641.
7) Sahoo C, Wankhade M, Singh BK. Sentiment analysis using deep learning techniques: a comprehensive review. *International Journal of Multimedia Information Retrieval*. 2023;12(2). Available from: https://doi.org/10.1007/s13735-023-00308-2.

8) Dhasarathan C, Shrestha H. An NLP Based Sentimental Analysis and Prediction: A Dynamic Approach. In: International Conference on Communication, Networks and Computing CNC 2020;vol. 1502 of Communications in Computer and Information Science. Springer, Singapore. 2022;p. 343–353. Available from: https://doi.org/10.1007/978-981-16-8896-6_28.
9) Akhtar MS, Ekbal A, Bhattacharyya P. Aspect based sentiment analysis in Hindi: resource creation and evaluation. In: Proceedings of the tenth international conference on language resources and evaluation (LREC'16). European Language Resources Association (ELRA). 2016;p. 2703–2709. Available from: https://aclanthology.org/L16-1429.
10) Gandhi H, Attar V. Extracting Aspect Terms using CRF and Bi-LSTM Models. *Procedia Computer Science*. 2020;167:2486–2495. Available from: https://doi.org/10.1016/j.procs.2020.03.301.
11) Bhattacharya A, Debnath A, Shrivastava M. Enhancing Aspect Extraction for Hindi. In: and others, editor. Proceedings of The 4th Workshop on e-Commerce and NLP. Association for Computational Linguistics. 2021;p. 140–149. Available from: https://doi.org/10.18653/v1/2021.ecnlp-1.17.
12) Shrivastava K, Kumar S. A Sentiment Analysis System for the Hindi Language by Integrating Gated Recurrent Unit with Genetic Algorithm. *The International Arab Journal of Information Technology*. 2020;17(6):954–964. Available from: https://iajit.org/portal/PDF/November%202020,%20No.%206/18975.pdf#:~:text=It%20has%20been%20observed%20that,based%20and%20machine%20learning%20approaches.
13) Vandana Yadav, Verma P, Katiyar V. Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages. *International Journal of Information Technology*. 2023;15(2):759–772. Available from: https://doi.org/10.1007/s41870-022-01010-y.
14) Rani S, Kumar P. Aspect-based Sentiment Analysis using Dependency Parsing. *ACM Transactions on Asian and Low-Resource Language Information Processing*. 2021;21(3):1–19. Available from: https://doi.org/10.1145/3485243.
15) Khurana D, Koli A, Khatter K, Singh S. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*. 2023;82(3):3713–3744. Available from: https://doi.org/10.1007/s11042-022-13428-4.
16) Salmony MY, Faridi AR, Masood F. Leveraging attention layer in improving deep learning models performance for sentiment analysis. *International Journal of Information Technology*. 2023. Available from: https://doi.org/10.1007/s41870-023-01570-7.
17) Jurafsky D, Martin JH. Sequence Labeling for Parts of Speech and Named Entities. In: Speech and Language Processing;vol. 71. 2020. Available from: https://web.stanford.edu/~jurafsky/slp3/8.pdf.
18) Deep Learning | Introduction to Long Short Term Memory. 2019. Available from: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.
19) Chen PH, Yu H, Dhillon IS, Hsieh C. DRONE: Data-aware Low-rank Compression for Large NLP Models. In: 35th Conference on Neural Information Processing Systems (NeurIPS 2021). 2021;p. 1–14. Available from: https://proceedings.neurips.cc/paper_files/paper/2021/file/f56de5ef149cf0aedcc8f4797031e229-Paper.pdf.
20) Garbin C, Zhu X, Marques O. Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia Tools and Applications*. 2020;79:12777–12815. Available from: https://doi.org/10.1007/s11042-019-08453-9.
21) Hodson TO. Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development* . 2022;15(14):5481–5487. Available from: https://doi.org/10.5194/gmd-15-5481-2022.
22) Plevris V, Solorzano G, Bakas NP, Seghier MB. Investigation of performance metrics in regression analysis and machine learning-based prediction models. In: ECCOMAS Congress 2022 - 8th European Congress on Computational Methods in Applied Sciences and Engineering. CIMNE. 2022. Available from: https://www.scipedia.com/public/Plevris_et_al_2022a.