UtilitasMathematica

ISSN 0315-3681 Volume 122, 2025

Aspect Based Sentiment Analysis using Deep Learning Models

Bhagyashree Bappanna
Assistant Professor, CSE - Artificial Intelligence and Machine learning, VNR Vignana Jyothi of Institute of Engineering and Technology
Ambika Shetkar
Assistant Professor, Computer science and engineering, Vnr Vignana Jyothi of Institute of Engineering and Technology
Bavana. Hanisha
Assistant Professor, Computer science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology
Manchem Asha priya
Assistant Professor, Artificial Intelligence and Machine learning, Gokaraju Rangaraju Institute of Engineering and Technology

Abstract. Sentiment analysis is an important task in natural language understanding and has a wide range of real-world applications. In sentiment analysis, the opinion is evaluated to its positivity, negativity and neutrality with respect to the complete document or object. But this level of analysis does not provide the necessary detailed information for many applications. To obtain more fine-grained analysis, Aspect Based Sentiment Analysis is introduced. Aspect Based Sentiment analysis introduces a suite of problems which require deeper NLP capabilities and also produces a rich set of results. This paper discusses about the aspect-based sentiment analysis classification using Deep learning models – Keras sequential model, LSTM model, BERT model using ktrain and multilabel multiclass BERT model using Hugging face transformers for sentiment and aspect classification.

Keywords: ASBS, LSTM, BERT, Keras, K train

1.Introduction

Aspect based sentiment analysis (ABSA) is a technique that takes into consideration the terms related to the aspects and identifies the sentiment associated with each aspect. ABSA model requires aspect categories and its corresponding aspect terms to extract sentiment for each aspect from a piece of text. For example, Analysing the restaurant based on the aspect food mentioned in the reviews. Deep Learning is at the cutting edge of what machines can do, and developers and business leaders absolutely need to understand what it is and how it works. This unique type of algorithm has far surpassed any previous benchmarks for classification of images, text, and voice. Deep learning starts with the humble perceptron. Similar to how a "neuron" in a human brain transmits electrical puls-es throughout our nervous system, the perceptron receives a list of input signals and transforms them into output signals. Deep Learning Algorithms use some-thing called a neural network to find associations between a set of inputs and outputs. There are many types of Deep learning - Convolutional Neural Networks (CNNs), Multi-Layer Perceptron's (MLPs) Recurrent Neural Networks (RNNs) etc.

2. Related Work

Aspect-Based Sentiment Analysis (ABSA) has gained significant attention in recent years due to its ability to capture more fine-grained sentiments about specific aspects of entities, such as products or services, from textual data. Unlike traditional sentiment analysis, which assigns a global sentiment label to the entire text, ABSA identifies and analyzes sentiments related to individual aspects, making it particularly useful for applications in product reviews, customer feedback, and social media analysis (Pontiki et al., 2014). Traditional methods, such as rule-based approaches and classical machine learning models like Support Vector Machines (SVMs), Naive Bayes, and Maximum Entropy, were initially employed for ABSA tasks. However, these methods often fell short in dealing with complex linguistic nuances and the variability of sentiment expressions in different contexts (Hu & Liu, 2004). With the advent of deep learning, ABSA has seen substantial improvements in performance. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been widely used for aspect-based sentiment classification due to their ability to capture sequential dependencies in text (Tai et al., 2015). LSTMs, specifically, help mitigate the vanishing gradient problem, making them more effective for handling long-range dependencies in sentence structures. However, a limitation of RNNs and LSTMs is that they often struggle to capture complex relationships between words that are far apart in a sentence. To address this, Convolutional Neural Networks (CNNs) have also been employed, as they can efficiently capture local n-gram patterns in text, which is particularly beneficial for sentiment analysis (Kim, 2014). A major breakthrough in ABSA came with the introduction of transformer-based models, particularly BERT (Bidirectional Encoder Representations from Transformers) (Vaswani et al., 2017). BERT and its variants have revolutionized natural language processing tasks, including ABSA, due to their self-attention mechanisms and the ability to process text bidirectionally, capturing contextual relationships between words regardless of their position in the sentence. Researchers have adapted BERT for ABSA tasks by fine-tuning pre-trained BERT models to perform aspect extraction and sentiment classification simultaneously (Zhang et al., 2019). The success of BERT has led to the development of other transformer-based models like RoBERTa, ALBERT, and XLNet, which have further improved performance on ABSA tasks by addressing some of BERT's limitations, such as training inefficiencies and handling of longer text sequences.

Moreover, multitask learning (MTL) has emerged as an effective technique for ABSA. By training models to simultaneously perform aspect extraction and sentiment classification, MTL takes advantage of shared representations between these tasks, leading to better generalization and improved performance (Zhang et al., 2019). Hybrid models that combine LSTMs or CNNs with attention mechanisms or integrate BERT with other deep learning techniques have shown to outperform single models by effectively capturing both local and global context (Xu et al., 2020). Despite these advances, ABSA still faces challenges, especially in the areas of domain adaptation and multilingual ABSA. Models trained on one domain (e.g., restaurant reviews) often do not generalize well to other domains (e.g., electronics reviews) due to differences in vocabulary, expression styles, and aspect categories (Liu et al., 2020). Domain adaptation techniques, such as transfer learning, are critical in overcoming this issue. Additionally, while much of the ABSA research has focused on English data, there is growing interest in developing multilingual models that can handle multiple languages with limited labeled data (Santos et al., 2021). This involves overcoming challenges related to linguistic diversity, sentiment expression, and aspect extraction in non-English languages. In conclusion, deep learning models have significantly advanced ABSA by providing more accurate and efficient solutions. Transformer-based architectures, especially BERT, along with multitask learning and hybrid models, have set new performance standards. However, challenges remain in domain adaptation, aspect-dependent sentiment, and multilingual ABSA, suggesting that future research should focus on improving model generalization, addressing linguistic diversity, and developing more robust approaches for fine-grained sentiment analysis across various domains and languages. Prasanna Pabba (2025) to automate human body measurements using computer vision methods, the Media pipe framework and the body matrix framework. The system measures body dimensions using images as inputs, and uses Media pipe's pose estimation and tracking as a way to identify key landmarks on the human body. The landmarks are then used in conjunction with computer vision algorithms to accurately calculate the measurements of interest

3. Methods and Methodology

This section describes the step wise approach in developing deep learning model for Aspect Based Sentiment Analysis. The flowchart is presented followed by data collection, data pre-processing, feature extraction, building deep learning models and evaluating the accuracy

The main objectives of this topic are mentioned below:

- To collect the restaurant reviews dataset (SEMEVAL 2014).
- To perform data pre-processing on the collected data.
- To perform feature extraction.
- To design and build the DL models for sentiment analysis.
- To test and analyse the developed deep learning model.
- To build a UI for Aspect Based Sentiment Analysis.

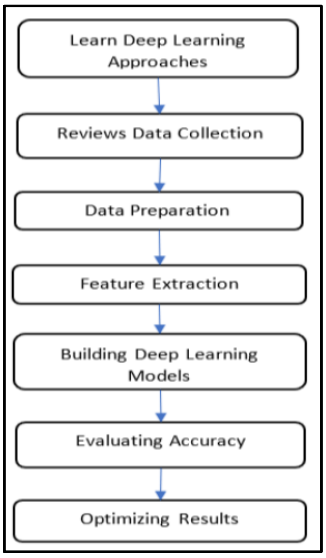

Fig. 1. Deep learning model for ABSA

3.1 Data Pre-processing

In NLP, text pre-processing is the first step in the process of building a model. The various text pre-processing steps are:

1. Tokenization: Tokens are the building blocks of Natural language. Tokenization is a way of separating a piece of text into similar units called tokens. Here tokens can be either words, characters, or sub words.

For example, let us consider “smarter”:

1. Character tokens: s-m-a-r-t-e-r
2. Sub word tokens: smart-er

2. Stop word removal: A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

3. Stemming and Lemmatization: Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming and Lemmatization both generate the root form of the inflected word. The difference is that stem might not be an actual word whereas, lemma is an actual language word. Stemming follows an algorithm with steps to perform on the words which makes it faster. Whereas, in lemmatization, WordNet corpus and a corpus for stop words as well to produce lemma is used which makes it slower than stemming.

For example: Consider words happy, happier and happiest; during stemming or lemmatization process these words are converted into a root word happy.

3.2 Feature extraction:

Feature extraction step to extract and produce feature representations that are appropriate for type of model that is used.

1. TF-IDF (Term Frequency-Inverse Document Frequency):

- TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.
- TF-IDF can be done by multiplying two metrics: term frequency, and the inverse document frequency of the word across a set of documents:
- TF: Term Frequency, which measures how frequently a term occurs in a document.

$$F(t) = \frac{\text{Number of times term t appears in a document}}{\text{Total number of terms in the document}}$$

IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, IDF is computed as below,

$$IDF(t) = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents with term t in it}}\right)$$

$$TF\ IDF = TF(t) \times IDF(t)$$

Keras Tokenizer - tf.keras.pre processing. text. Tokenizer

This class allows to vectorize a text corpus, by turning each text into either a sequence of integers (each integer being the index of a token in a dictionary) or into a vector where the coefficient for each token could be binary, based on word count, based on TF-IDF.

After feature extraction, the dataset is split into 80% for training and 20% for testing.

Building Models:

1. Keras Model

Keras is a high-level neural networks API, capable of running on top of Tensor flow, Theano, and CNTK. It enables fast experimentation through a high level, user-friendly, modular and extensible API. Keras can also be run on both CPU and GPU. The Keras deep learning library helps to develop the neural network models fast and easy. There are two ways to create Keras models such as sequential and functional.

The easiest way of creating a model in Keras is by using the sequential API, which lets to stack one layer after the other. The problem with the sequential API is that it doesn't allow models to have multiple inputs or outputs, which are needed for some problems. Nevertheless, the sequential API is a perfect choice for most problems. To create a convolutional neural network, a Sequential object is created and the add function is used to add layers.

Alternatively, the functional API allows to create the same models but offers more flexibility at the cost of simplicity and readability. It can be used with multiple input and output layers as well as shared layers, which enables to build really complex network structures. When using the functional API, always need to pass the previous layer to the current layer. It also requires the use of an input layer.

LSTM Model

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. Feed-forward networks are neural networks where inputs are multiplied by a weight and then bias is added to that and so on and the last layer is output layer. But the problem with these types of networks is they do not store memory and cannot be used in sequential data. Even the input and output of this type of net-work is fixed. RNN was designed in a way such that it can catch the sequential / time series data. But RNN suffers from a vanishing gradient problem that is very significant changes in the weights that do not help the model learn. To overcome this LSTM was introduced.

BERT Model

BERT (Bidirectional Encoder Representations from Transformers) is a Natural Language processing Model proposed by researchers at Google Research in 2018. When it was proposed it achieve state -of -the -art accuracy on many NLP and NLU tasks such as:

- General Language Understanding Evaluation
- Stanford Q/A dataset SQuAD v1.1 and V2.0
- Situation with Adversarial Generations

BERT makes use of transformers, that learns contextual relations between words in a text. Transformer includes two separate mechanisms namely: Encoder that reads the input text and decoder that produces the prediction of the task.

The transformer encoder reads the entire sequence of words at once. hence it is considered as bidirectional. This characteristic allows the model to learn the context of a word based on its surroundings (both left and right of the word).

BERT Model Architecture:

BERT is released in two sizes $BERTbase$ and $BERTlarge$. The BASE model is used to measure the performance of the architecture comparable to another architecture and the LARGE model produces state -of -the-art result. BERT is basically an Encoder stack of transformer architecture. A transformer architecture is an encoder- decoder network that uses self-attention on the encoder side and attention on the decoder side.

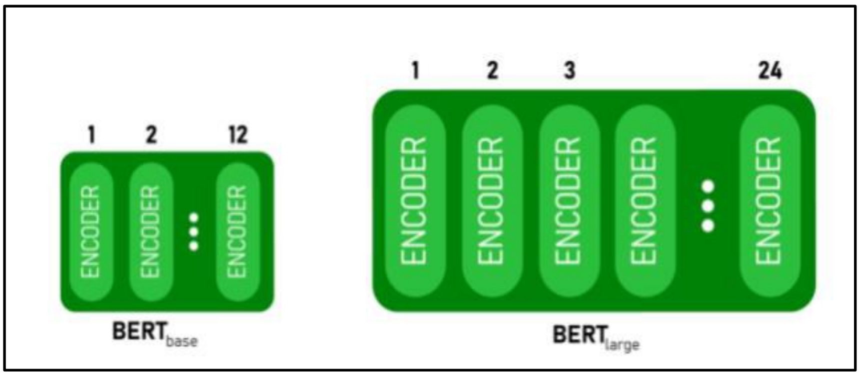

Fig. 2 BERT Model Architecture: $BERTbase$ and $BERTlarge$

A multilabel multiclass Bert has been developed using Hugging face transformers and Keras. The Multilabel, Multiclass text classification model figure developed has 4 layers. The first layer is

input layer, the second layer is TFBert Main Layer. The third layer is dropout layer. The last layer has two dense layers which are sentiment and aspect.
These models are then compiled, and trained using training data. Once the models are fitted, the model can be evaluated based on accuracy using the testing data. An application is developed using python Flask wherein user can enter a review and the model predicts the aspect sentiment and displays.

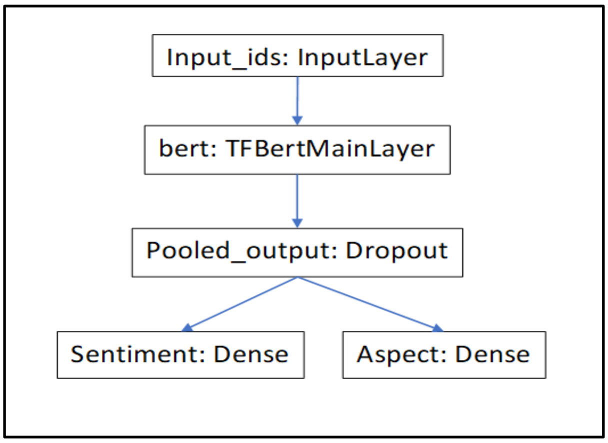

Fig 3: Multiclass text classification model

4 Results

4.1 Keras Model
Sentiment Model using Keras:
The figure 1 describes the sentiment model using Keras with test accuracy of 0.6765

```
smodel_score = sentiment_model.evaluate(X_test,Y_test, batch_size=64, verbose=1)
print('Test accuracy:', smodel_score[1])

12/12 [==============================] - 0s 3ms/step - loss: 0.8402 - accuracy: 0.6765
Test accuracy: 0.6765498518943787
```

Fig. 4 Sentiment model using Keras Sequential model

Aspect Model using Keras:
Figure 2 describes the Aspect model using Keras with test accuracy being 0.61.

```
amodel_score = absa_model.evaluate(X_test,Y_test, batch_size=64, verbose=1)
print('Test accuracy:', amodel_score[1])

12/12 [==============================] - 0s 3ms/step - loss: 1.0716 - accuracy: 0.6105
Test accuracy: 0.6105121374130249
```

Fig. 5 Aspect Model using keras Sequential model

LSTM Model
Sentiment Model using LSTM:
Figure 3 describes the Sentiment model using LSTM with test accuracy of 0.66.

```
smodel_score = s_model.evaluate(X_test,Y_test, batch_size=64, verbose=1)
print('Test accuracy:', smodel_score[1])

12/12 [==============================] - 0s 15ms/step - loss: 0.8566 - accuracy: 0.6604
Test accuracy: 0.6603773832321167
```

Fig. 6 Sentiment Model using LSTM

Aspect Model using LSTM:

Figure 4 describes the Aspect model using LSTM with test accuracy of 0.62.

```
amodel_score = a_model.evaluate(X_test,Y_test, batch_size=64, verbose=1)
print('Test accuracy:', amodel_score[1])

12/12 [==============================] - 0s 17ms/step - loss: 1.0954 - accuracy: 0.6213
Test accuracy: 0.6212937831878662
```

Fig. 7 Aspect Model using LSTM

BERT Model

BERT Model using Ktrain

Visualizing the graph of Loss Vs Learning rate of Sentiment and Aspect Model

Sentiment Model:

```
[ ]  learner.lr_plot()
```

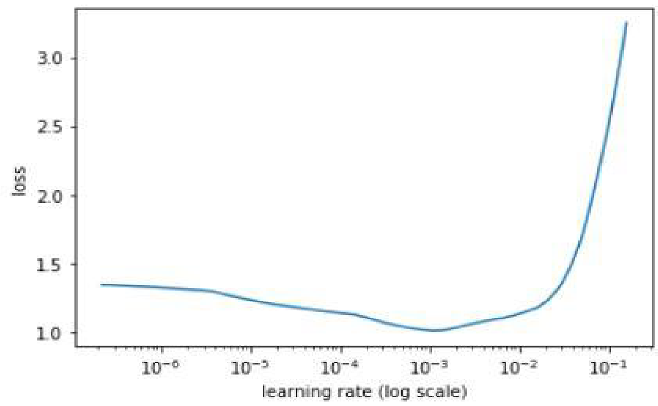

Fig. 8 Loss vs Learning Rate of Sentiment Model

Aspect Model:

```
[ ]  learner1.lr_plot()
```

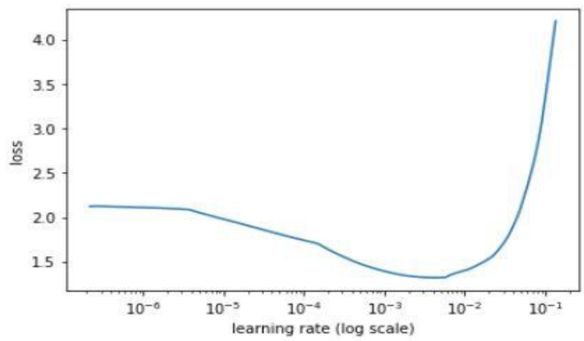

Fig. 9 Loss vs Learning rate of Aspect Model

Multi Label, Multiclass BERT

Training the models with optimized learning rate

Sentiment Model with Optimized learning Rate:

```
[ ] learner.fit_onecycle(lr = 1e-4, epochs = 1)

    begin training using onecycle policy with max lr of 0.0001...
    47/47 [==============================] - 40s 853ms/step - loss: 0.7626 - accuracy: 0.7031 - val_loss: 0.8267 - val_accuracy: 0.6707
    <tensorflow.python.keras.callbacks.History at 0x7f54423033d0>
```

Fig. 10 Sentiment Model with Optimized learning Rate

Aspect Model with Optimized Learning Rate:

```
[ ] learner1.fit_onecycle(lr = 1e-4, epochs = 1)

    begin training using onecycle policy with max lr of 0.0001...
    47/47 [==============================] - 41s 873ms/step - loss: 0.7834 - accuracy: 0.7017 - val_loss: 0.8024 - val_accuracy: 0.6856
    <tensorflow.python.keras.callbacks.History at 0x7f5442143890>
```

Fig. 11 Aspect Model with Optimized Learning Rate

Testing Model with optimized Learning rate Multi Label, Multiclass BERT using Hugging face transformers:

```
[ ] data= ['the pizza is good']

[ ] predictor = ktrain.get_predictor(learner.model, preproc)
    predictor1 = ktrain.get_predictor(learner1.model, preproc1)
    sentiment=predictor.predict(data)
    aspect= predictor1.predict(data)
    print([sentiment, aspect])

    [['positive'], ['food']]
```

Fig. 12 Testing Model with optimized Learning rate

Multi Label, Multiclass BERT using Hugging face transformers

Multi Label, Multiclass BERT using Hugging face transformers Comparing Training and Testing Accuracy of all developed Models:

```
24/24 [==============================] - 5s 209ms/step - loss: 1.9517 - aspect_loss: 1.0207 - sentiment_loss: 0.9310
```

```
aspect_accuracy: 0.6415 - sentiment_accuracy: 0.6712
```

Fig. 13 Multi Label, Multiclass BERT using Hugging face transformers

Table 2. Comparing Training and Testing Accuracy of all developed Models

	Keras Sequential		LSTM		ktrain Bert		Multilabel Multiclass Bert	
Accuracy	Aspect	Sentiment	Aspect	Sentiment	Aspect	Sentiment	Aspect	Sentiment
Training	69.30	72.83	62.89	67.88	70.17	70.31	81.84	82.64
Testing	61.05	67.65	62.13	66.04	68.56	67.07	64.15	67.12

Demonstration of UI

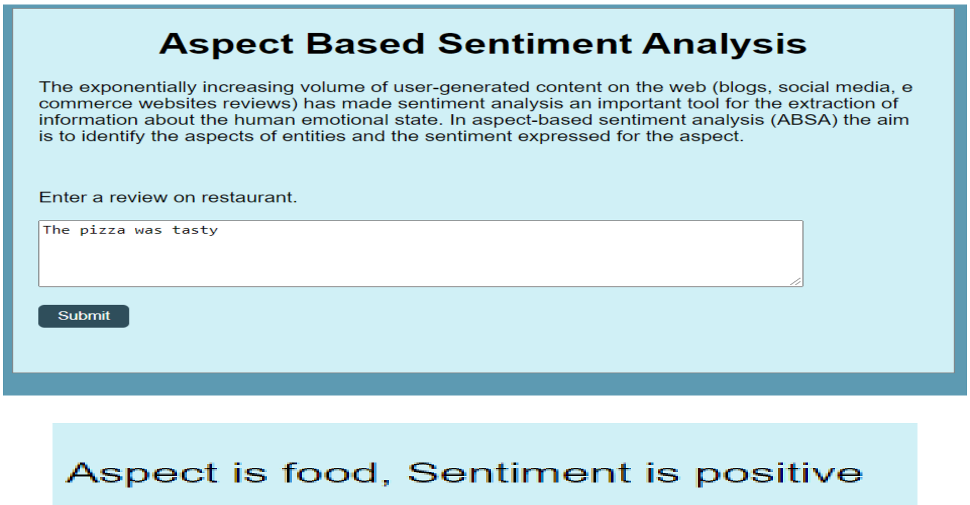

Fig. 14 Demonstration of UI in Aspect based sentiment Analysis

Disclosure of Interests.

Future work in the area of ABSA should construct even more benchmark datasets, from a greater variety of domains, to allow safer conclusions to be drawn. A frame work can be designed for Aspect Based Sentiment Analysis through which sentiment analysis can be done for any domain datasets. So, for most of the work carried with respect to Sentiment Analysis, Opinion Mining and Aspect Based Sentiment Analysis with machine learning techniques only, moreover Deep Learning is becoming a prominent area for the research.

Suggestions for future work include developing a multi label classification model to label the sentence reviews into multiple aspects. Example a user enters a paragraph as review describing various aspects (food, service, price), and the model predicts many aspects and the associated sentiment. Extending the model to predict reviews in other language also for example French, Hindi, German. Developing a dashboard application where the user can visualise the aspect-based sentiment analysis which will be helpful for businesses to have insights and improve.

References

1. Dong, L., Wei, F., & Xu, J. (2014). *A Hybrid Deep Learning Model for Aspect-Based Sentiment Analysis in Twitter*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.
2. Hu, M., & Liu, B. (2004). *Mining and Summarizing Customer Reviews*. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
3. Kim, Y. (2014). *Convolutional Neural Networks for Sentence Classification*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.
4. Liu, B., & Zhang, L. (2020). *Deep Learning for Aspect-Based Sentiment Analysis: A Survey*. Proceedings of the 2020 International Conference on Natural Language Processing and Chinese Computing.
5. Pontiki, M., et al. (2014). *SemEval-2014 Task 4: Aspect-Based Sentiment Analysis*. Proceedings of the 8th International Workshop on Semantic Evaluation.
6. Santos, R., Araujo, R., & Liao, S. (2021). *Multilingual Aspect-Based Sentiment Analysis: A Survey and Research Directions*. Journal of Artificial Intelligence Research, 71, 1-40.
7. Tai, K. S., Socher, R., & Manning, C. D. (2015). *Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks*. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics.
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017). *Attention Is All You Need*. Advances in Neural Information Processing Systems.
9. Xu, J., et al. (2020). *Hybrid Deep Learning for Aspect-Based Sentiment Analysis*. Proceedings of the 2020 Conference on Neural Information Processing Systems.
10. Zhang, Z., Zhao, L., & LeCun, Y. (2019). *BERT for Aspect-Based Sentiment Analysis*. Proceedings of the 2019 International Conference on Computational Linguistics.
11. Mrs. Prasanna Pabba, E. Kaeith Emmanuel, G. Gopi Harishitha, G. Sahasra, & A. Sai Pranavi. (2025). Automatic Human Body Measurement Using Computer Vision & Media Pipe. The Bioscan, 20(Supplement 2), 253–258.