Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 8, No. 1, February, 2024

Extracting Explicit and Implicit Aspects Using Deep Learning

Mikail Muhammad Azman Busst [1], Kalaiarasi Sonai Muthu Anbananthen [1*], Subarmaniam Kannan [1], Rajkumar Kannan [2]

[1] *Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia.*

[2] *Bishop Heber College (Autonomous), Tiruchirappalli 620017, India.*

Abstract

The proliferation of user-generated content on social networks and websites has heightened the significance of sentiment analysis, also known as opinion mining, as a critical tool for comprehending people's attitudes toward various topics. Aspect-level sentiment analysis, which considers specific aspects or features of texts, provides a more comprehensive view of sentiment analysis. The aspect-level approach encompasses both explicit and implicit aspects, where explicit aspects are readily mentioned in texts while implicit aspects are implied or inferred from contextual clues. Despite the significance of implicit aspects in the overall review, previous research has predominantly focused on explicit aspect extraction. Limited attention has been given to the extraction of implicit aspects, despite their potential impact on capturing the complete sentiment picture of texts. Therefore, this study aims to find an aspect extraction solution capable of identifying and extracting both explicit and implicit aspects from texts. This study compares various machine and deep learning models on the SemEval-2014 and SemEval-2016 restaurant datasets. The experimental analysis demonstrates that the proposed Aspect-BiLSTM model emerged as the best-performing model, achieving high accuracy in classifying both explicit and implicit aspects, with 92.9% accuracy for the 2014 and 90.7% accuracy for the 2016 datasets. Notably, the proposed solution was able to capture multiple aspects of texts, making it more robust and versatile. This study highlights the efficacy of the Aspect-BiLSTM model for aspect extraction, which will give valuable insights into the advancement of aspect-level sentiment analysis.

Keywords:

Aspect Extraction;
Explicit Aspects;
Implicit Aspects;
Deep Learning.

Article History:

Received:	02	October	2023
Revised:	11	January	2024
Accepted:	19	January	2024
Published:	01	February	2024

1- Introduction

Sentiment analysis, or opinion mining, plays a vital role in identifying opinions expressed in written texts [1]. The analysis can be carried out at the document, sentence, or aspect/feature levels. According to Jonnalagadda et al. (2019) [2], document-level and sentence-level sentiment analysis allow for identifying the overall sentiment polarities of documents and sentences, respectively, as shown in Figure 1. However, according to Medhat et al. (2014) [3], these two levels of sentiment analysis are insufficient for capturing the sentiment information of the features or aspects in documents and sentences. Hence, this limitation highlights the need for deeper analysis of sentiments in unstructured textual data. Therefore, aspect-level sentiment analysis becomes increasingly significant in such scenarios.

Aspect-level sentiment analysis provides a more comprehensive view of sentiment analysis by capturing the sentiment information related to specific aspects or features of texts [4]. Identifying these aspects is a crucial step in sentiment analysis. Additionally, aspect-level sentiment analysis is the most granular form since it captures multiple sentiments expressed in texts instead of being limited to just one sentiment per document or sentence.

* **CONTACT**: kalaiarasi@mmu.edu.my

DOI: http://dx.doi.org/10.28991/ESJ-2024-08-01-05

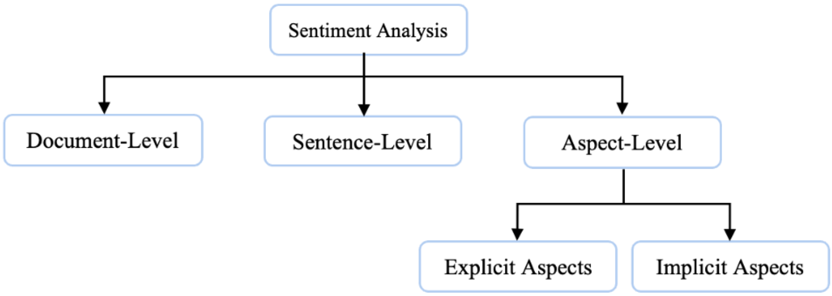

Figure 1. Classification of Sentiment Analysis Levels

Aspect-level sentiment analysis can be further classified into two aspects present in texts: explicit and implicit, as shown in Figure 1. Explicit aspects are directly present in texts and include words or phrases that describe an entity's characteristics. On the other hand, implicit aspects are more challenging to extract as they are not directly mentioned in texts but are inferred from other words associated with them [5].

Despite the extensive research on explicit aspect extraction, implicit aspect extraction has received more limited attention from the research community than explicit aspect extraction. This is due to the different approaches proposed in previous works to extract aspectual features from texts. Certain aspect extraction or aspect-level sentiment analysis solutions, like those proposed by Singh Chauhan et al. (2020) [6], He et al. (2023) [7], and Karimi et al. (2021) [8], treat the problem as sequence labeling problems. This approach to aspect extraction involves the process of labeling aspect terms in documents or sentences. The problem with this approach is that it cannot be used for implicit aspect extraction, as implicit aspects are not directly mentioned in texts and thus cannot be labeled.

Nevertheless, implicit aspects are equally significant and contribute substantially to the meaning of texts, as Maitama et al. (2020) [9] established. However, the extraction of implicit aspects is not as straightforward as explicit ones due to their absence from texts. While human beings can identify implicit aspects of texts through their prior knowledge, aspect extraction models lack the knowledge to make these inferences. Instead, this knowledge must be introduced to these models through handcrafted rules from rule-based algorithms or learned weights for approaches utilizing machine learning techniques.

Besides this, current aspect-level sentiment analysis studies do not heavily focus on extracting multiple aspects from texts. This can be attributed to the architectures of the solutions proposed in them that only produce one set of sentiment output labels for each document or sentence, such as the solutions proposed by Zhang et al. (2020) [10] and Zhou et al. (2022) [11]. This poses an issue as written texts may include multiple explicit and implicit aspects; thus, it is imperative for an optimal aspect extraction solution to consider the presence of all aspects in them. As a result, the main objective of this study is to propose an aspect extraction solution capable of identifying and extracting multiple aspects from texts, comprising both explicit and implicit aspects.

This paper is organized as follows: Section 2 highlights the existing studies conducted for explicit and implicit aspect extraction, followed by the research framework adopted to develop the proposed aspect extraction models in Section 3. Section 4 presents the experiment to evaluate the developed aspect extraction models and analyze their results. Section 5 concludes this study.

2- Background Study

As illustrated in Figure 2, aspect extraction solutions can be classified as either being rule-based, unsupervised, or supervised.

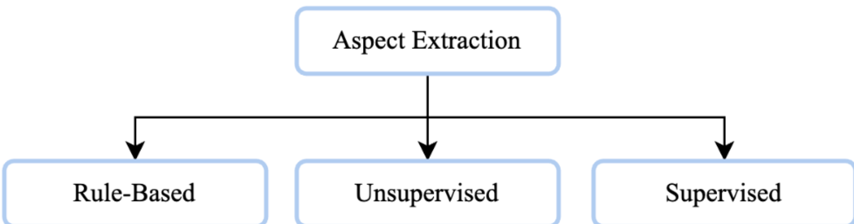

Figure 2. Types of methods used for aspect extraction

2-1- Rule-Based Approaches

Rule-based algorithms for aspect extraction involve the use of manually crafted rules [12]. These approaches do not require any form of training, and they provide transparency and flexibility for modifications. Prior studies on rule-based aspect extraction include those conducted by Zainuddin et al. (2017) [13], Shouten et al. (2017) [14], Rana et al. (2020) [15], Venugopalan & Gupta (2020) [16], and Li et al. (2020) [17].

Zainuddin et al. (2017) [13] employed a Stanford dependency parser to extract aspects from texts by breaking them down into distinct grammatical components and determining the implicit aspect words or phrases based on the grammatical dependencies extracted. However, this method may not be suitable for texts with incorrect grammar. Schouten et al. (2017) [14], on the other hand, employed a co-occurrence association rule mining approach to extract aspects from textual data. Their method involved mining association rules using the spread activation algorithm to generate associative networks based on the word relationships depicted in their co-occurrence matrix, which captured the frequencies of words occurring in texts referring to each aspect category.

The aspect extraction solution proposed by Rana et al. (2020) [15] adopted two sets of rules for explicit and implicit aspect extraction. The first set of rules adopted was a set of sequential pattern rules that utilized each word's Part of Speech (POS) tags and the use of opinion and concept lexicons to extract explicit aspect terms from their input texts. The second set of rules utilized the synonyms and co-occurrences of explicit aspect terms. It adopted the use of the Normalized Google Distance (NGD) [18] algorithm to extract implicit aspects from the solution's input texts.

Venugopalan & Gupta (2020) [16] proposed an aspect extraction solution that utilized multiple sets of rules that took advantage of the different features of its input texts. It first extracted the initial aspect terms of its input texts by considering each word's POS and Named Entity Recognition (NER) tags and their grammatical dependencies. It then utilized a set of rules to capture aspects that were expressed using compound terms and a set of heuristic rules to capture aspects expressed in phrases. The extracted aspects then went through a couple of pruning processes based on the presence of single-word aspects in the extracted multi-word aspects as well as on the word embedding similarities of the extracted aspects with certain aspect seed terms.

Lastly, the rule-based aspect extraction solution proposed by Li et al. (2020) [17] adopted a framework based on the Answer Set Programming (ASP) [19] logic programming paradigm. Their solution first modeled the relationships between the words in its input texts using Abstract Meaning Representation (AMR) graphs [20]. AMR subgraphs of the words were then extracted using the AMR Subgraph Extraction (AMR-SE) algorithm to capture the semantic relationships between aspects and their respective opinion words. The researchers then adopted a rule-mining technique to model the various relationships of aspects and other words in the subgraphs. They applied syntactic rules to capture the parts of speech of each word in them. The relationships modeled by the semantic and syntactic rules were then represented using the Answer Set Programming (ASP) logic programming paradigm. The texts' final aspects were extracted using an ASP resolver.

2-2- Unsupervised Approaches

Similar to rule-based models, unsupervised machine learning models do not require training, as they cluster or represent data points based on their features. Studies that have adopted unsupervised machine-learning approaches for aspect extraction include those conducted by Tulkens & Cranenburgh (2020) [21], Luo et al. (2019) [22], Venugopalan & Gupta (2022) [23], and Singh Chauhan et al. (2020) [6].

Tulkens & Cranenburgh (2020) [21] utilized the semantic similarities of words captured by word embeddings, specifically Word2Vec [24] embeddings, to encode aspect terms and labels to determine their semantic similarities. They also employed an attention mechanism to assign weights to the words in documents, prioritizing those that are more similar to aspects. However, the limitation of this model was that it only searched for aspect terms explicitly mentioned in texts. Luo et al. (2019) [22] used an unsupervised neural network to generate sentence representations of aspects in texts. Their solution did so by learning the features of texts at a sememe level, which can be described as minimum semantic units of word meanings [3]. Sequential information from their input text was extracted using Long Short-Term Memory (LSTM) [25] layers. Aspects were then inferred from the model's output representations in the embedding space used by the model to find the closest aspect categories related to them. However, using Word2Vec to generate initial word vectors posed a problem, as it cannot generate embeddings for Out-of-Vocabulary words.

The unsupervised solution proposed by Venugopalan & Gupta (2022) [23] utilized a guided Latent Dirichlet Allocation (LDA) [26] model to determine the aspect categories present in its input texts. The guided LDA model used in the researchers' proposed solution required two sets of inputs. The first set consisted of sequences of initial aspect terms that were extracted and filtered from the input texts. The second input set, on the other hand, consisted of common aspect seed terms used in each aspect category. The use of these two sets of input data improved their LDA model's ability to form both topic-word and review-topic distributions. The unsupervised aspect extraction solution proposed by Singh Chauhan et al. (2020) [6] utilized a Bidirectional Long Short-Term Memory (BiLSTM) [27] model that was trained using the labels extracted by their hybrid algorithm (rule-based + unsupervised). The initial aspect terms were first extracted using a set of rules based on the parts of speech of words in the input texts as well as their grammatical dependencies. Once extracted, the irrelevant aspects were then pruned based on their frequencies and similarities with certain prominent aspect terms. The final extracted aspects were then encoded for aspect term extraction based on the IOB labels used in similar works.

2-3- Supervised Approaches

In contrast to unsupervised and rule-based approaches, supervised machine learning models are trained using input data samples and their corresponding output labels [28]. Studies exploring this approach include those conducted by Ray & Chakrabarti (2019) [29], Karimi et al. (2021) [8], He et al. (2023) [7], and Cai et al. (2020) [30].

Ray & Chakrabarti (2019) [29] developed a supervised deep-learning approach to aspect extraction in which a Convolutional Neural Network (CNN) [31] was utilized to extract aspects from texts. The CNN consisted of multiple convolution and max pooling layers with an attention mechanism to extract aspect features from input texts.

The supervised aspect extraction solution proposed by Karimi et al. (2021) [8] utilized a fine-tuned BERT [32] model. This model consisted of a BERT encoder and a fully connected output layer and was trained using the adversarial training technique. This technique generated adversarial training features that attempted to make the model form incorrect predictions. This technique was adopted during training to improve a model's generalization of its training data, resulting in better prediction performance.

The aspect extraction solution proposed by He et al. (2023) [7], on the other hand, utilized a self-training mechanism by employing a teacher model, a meta-weighter, as well as a student model. The training data used for this solution consisted of a set of gold-annotated features that have been manually labeled and a set of unlabeled features. The teacher model, which consisted of a Multi-Layer Perceptron (MLP) [33], was trained on the gold-annotated training data to generate accurate pseudo-labels for the remaining features. The student model and the meta-weighter were then trained on both the gold-annotated and pseudo-annotated features to overcome the influence of the imbalanced datasets.

Lastly, the supervised aspect extraction solution proposed by Cai et al. (2020) [30] utilized a Hierarchical Graph Convolutional Neural Network (Hier-GCNN) to identify the aspect categories present in its input texts. The model adopted by the researchers learned the co-occurrence relationships between different aspect categories in a hierarchical design, enabling it to capture interactions between them at different levels.

Ideal aspect extraction solutions should not only be able to extract both explicit and implicit aspects from texts but should also be able to extract multiple instances of them as well. Based on the previous aspect extraction studies highlighted in Table 1, it can be seen that while both rule-based and unsupervised methods include certain advantages, they may not be the most suitable methods for developing ideal aspect extraction solutions. This is because rule-based algorithms struggle with handling linguistic nuances in texts, affecting their aspect extraction capabilities. On the other hand, unsupervised algorithms are not suited to extract multiple aspects from texts [34]. Pontiki et al. (2014) [34] have demonstrated the effectiveness of supervised aspect extraction solutions in explicit and implicit aspect extraction by comparing their performances to certain unsupervised approaches. Their findings have shown that the evaluated supervised approaches obtained higher aspect extraction performance than their unsupervised counterparts based on their accuracy and F1 scores.

Based on the context of our research, supervised aspect extraction solutions can effectively handle the nuances of the task by utilizing domain-specific knowledge, certain linguistic cues, and contextual information. Besides this, they tend to outperform their rule-based and unsupervised counterparts in terms of extraction accuracy and are more suited for extracting multiple aspects from single documents or sentences. These advantages are crucial for our current research and for identifying aspects when performing sentiment analysis in our future work. Therefore, this study will assess the effectiveness of various supervised machine and deep learning models in extracting multiple explicit and implicit aspects from texts.

Table 1. Various research on Aspect Extraction Studies with Performance Metrics (Accuracy, Precision, Recall, and F1 Score)

Study	Methods	Datasets	P	R	F1
		Rule-Based Approaches			
Zainuddin et al. (2017) [13]	Dependency parsing	Stanford Twitter Sentiment [353 samples] Hate Crime Twitter Sentiment [1,078 samples] Sanders Twitter Corpus [1,091 samples]	-	-	-
Schouten et al. (2018) [14]	Co-occurrence association rule mining	SemEval-2014 Restaurant [3,844 samples]	70.00	64.70	64.00
Li et al. (2020) [17]	AMR-SE + semantic rule mining + syntactic rules + ASP solver	SemEval-2014 Restaurant [3,844 samples]	83.80	86.60	85.10
Rana et al. (2020) [15] (implicit aspect extraction)	Sequential pattern rules + co-occurrences & NGD	Customer Reviews [3,945 samples]	77.00	79.00	78.00
Venugopalan & Gupta (2020) [16]	Grammatical dependencies-POS-NER rules + multi-word aspect term rules + heuristic rules + pruning mechanism	SemEval-2014 Restaurant [3,844 samples]	53.00	81.90	64.35

Unsupervised Approaches					
Chauhan et al. (2020) [6]	BiLSTM (trained on aspects extracted from rule-based + unsupervised approach)	SemEval-2016 Restaurant [2,676 samples]	81.02	77.09	79.01
Tulkens & Cranenburgh (2020) [21]	Word embeddings similarities	Citysearch corpus [1,490 samples]	86.50	86.40	86.40
Luo et al. (2019) [22]	Aspect embeddings generation	Citysearch corpus [1,490 samples]	82.50	81.80	82.10
Venugopalan & Gupta (2022) [23]	Guided LDA	SemEval-2014 Restaurant [3,844 samples]	75.32	86.61	80.57
Supervised Approaches					
Ray & Chakrabarti (2019) [29]	Supervised Deep Learning + Rule-Based Hybrid (CNN)	SemEval-2014 Restaurant [3,844 samples]	-	-	-
Cai et al. (2020) [30] (average)	Hier-GCNN	SemEval-2015 Restaurant [1,647 samples]	70.38	55.58	61.93
He et al. (2023) [7]	MLP	SemEval-2014 Restaurant [3,844 samples]	-	-	88.95
Karimi et al. (2021) [8]	Fine-tuned BERT	SemEval-2014 Laptop [3,845 samples]	-	-	85.57

3- Research Methodology

An aspect extraction framework was developed in this study to determine the most optimal method of extracting explicit and implicit aspects from unstructured texts. It involved the evaluation of several machines and deep learning models in identifying both types of aspects, as well as the development of a novel aspect extraction model that utilized the aspect feature extraction pipeline of the best-performing model. The proposed aspect extraction framework includes pre-processing, text encoding, aspect feature extraction methods, and classification modules, as shown in Figure 3. The pre-processing, text encoding, and classification modules remained constant while different aspect feature extraction methods were used.

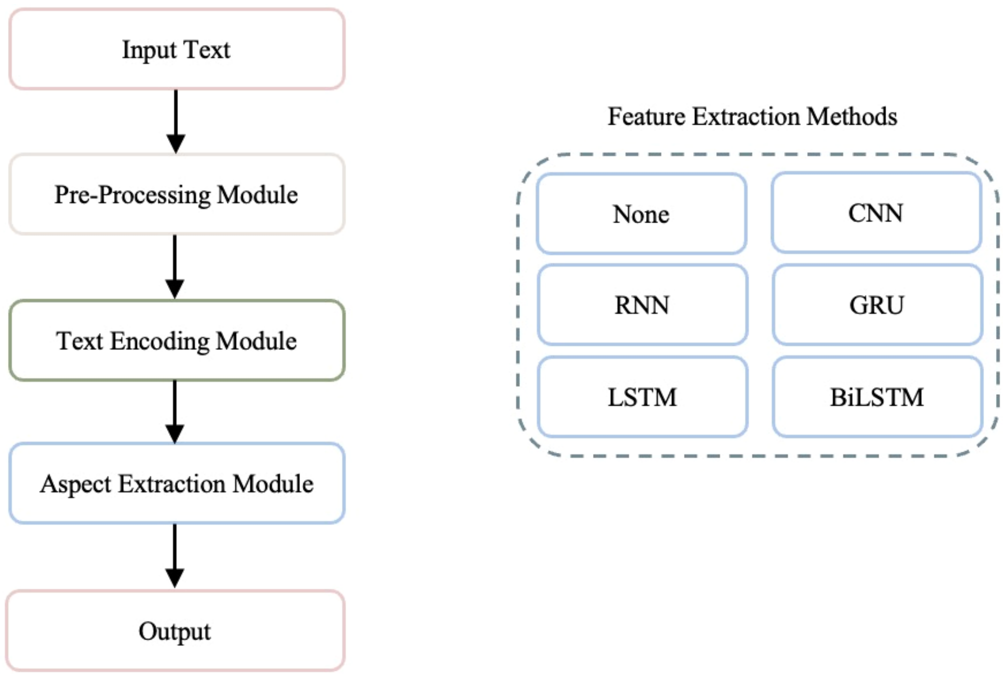

Figure 3. Research framework

3-1- Pre-Processing Module

Pre-processing is first conducted on raw input texts, which may contain redundant and noisy features. It can be described as the process of cleaning up noise from raw data and preparing it for machine or deep learning models [35]. In this research, we performed text normalization and removed all the punctuation, numbers, and stopwords from them. Text normalization was conducted to prevent the text encoding model from misinterpreting the same words written in different character cases as having different features. Punctuations, numbers, and stopwords were removed as they did not provide useful information for aspect extraction. Table 2 highlights the pre-processing steps used on the input data and their effects on them.

Table 2. Pre-processing steps and their effects on input data

Pre-Processing Step	Data Before Being Pre-Processed	Data After Being Pre-Processed
Text normalization	Loved the risotto here but I didn't love waiting 1 hour for it!	Loved the risotto here but I didn't love waiting 1 hour for it!
Punctuations removal	Loved the risotto here but I didn't love waiting 1 hour for it!	loved the risotto here but I didn't love waiting 1 hour for it
Numbers removal	loved the risotto here but I didn't love waiting 1 hour for it	loved the risotto here but I didn't love waiting hour for it
Stopwords removal	loved the risotto here but I didn't love waiting hour for it	loved risotto I didn't love waiting hour

3-2- Text Encoding Module

After pre-processing, the text data needed to be converted into numerical representations to enable machine and deep learning models to analyze and process them effectively, as they cannot perform arithmetic operations on text data. Therefore, texts were converted into word embeddings before being sent to the model. Word embeddings are vector representations of words in documents containing their syntactic and semantic properties and, in the case of language models, their contextual properties as well.

The present study employed a pre-trained BERT model, as it has been observed that contextual embeddings generated by BERT are endowed with richer contextual properties than other neural text embeddings and language models, as previously demonstrated in Wang et al. (2019) [36]. Specifically, the $\text{BERT}_{\text{BASE}}$ model, which incorporates 12 layers of transformer encoders, was utilized in this investigation. Notably, this model can generate representations for Out-of-Vocabulary terms. Hence, none of the word representations were manually generated. The text-encoding process is illustrated in Figure 4.

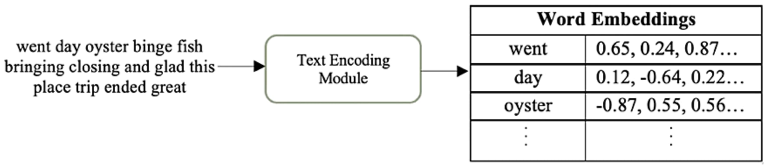

Figure 4. Text encoding process

3-3- Aspect Extraction Models

Once the texts were encoded, the implicit and explicit features/aspects were extracted. Several supervised machine and deep learning aspect feature extraction models were evaluated in this study, which included k-nearest Neighbour (kNN), Decision Tree (DT), Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM) models, as well as a newly proposed model in this study. The evaluated models and their descriptions are listed below.

Supervised Machine Learning Models

k-Nearest Neighbours (kNN): k-Nearest Neighbour classifiers are machine learning models that classify input samples to their respective output classes based on their proximity to previously seen examples. An unseen data sample is assigned to the class with the most data points closest to it out of k data points. The value of k is an integer which is pre-determined before the training process [37].

Decision Tree (DT): Decision Tree classifiers are supervised machine learning models that classify unseen input samples based on sets of logic conditions organized sequentially like a directed rooted tree graph. The tree's root will represent the input data, while each node (except the leaf nodes) represents a logic condition for the input data to meet. The tree's leaf nodes represent the input features' final output classes [38].

Supervised Deep Learning Models

Multi-Layer Perceptron (MLP): MLPs are deep learning models that consist of multiple artificial neurons. Each neuron transforms the input features passed to it using non-linear functions and then passes the outputs from those functions to its activation functions. These models act as the foundation for various deep learning models [33].

Convolutional Neural Network (CNN): CNNs are deep learning models that remove noisy and redundant features from their input data. They accomplish this by passing their input data through one or more convolution layers that capture more focused representations of them and then through a pooling layer, which further reduces the size of the convolved features [31].

Recurrent Neural Network (RNN): RNNs are deep learning models that extract sequential properties from time series data. They accomplish this by using specialized neural networks that contain feedback loops to generate representations of their input data at every step by utilizing their respective feature(s) along with the hidden state of the input data from the previous step [39].

Gated Recurrent Unit (GRU) & Long Short-Term Memory (LSTM): GRU and LSTM models are specialized versions of RNN models that aim to solve the vanishing gradient problem. They use specialized logic gates to dynamically retain or remove features at certain layers in their architectures [40, 25].

Bidirectional Long Short-Term Memory (BiLSTM): BiLSTM models are variants of the LSTM model that capture the sequential properties of texts in both forward and backward directions. Therefore, each sequential representation (hidden state) is generated based on the data that came before and after each time step [27].

Aspect-BiLSTM (Proposed Model)

In this research, we developed the Aspect-BiLSTM aspect extraction model, which integrates two distinct techniques. Specifically, we have leveraged the semantic and contextual properties of texts extracted from the flattened BERT word embeddings in conjunction with the sequential representations produced by a BiLSTM model. These two features were merged into singular aspect features for the Aspect-BiLSTM model. This resulted in a more robust algorithm utilizing features that better represented aspects present in unstructured texts.

Output Module

Following the application of the Aspect Extraction Module, the resultant output comprises a comprehensive set of aspects, which includes both explicit and implicit aspects. This is a crucial feature of our model, which enhances the overall effectiveness of the analysis.

3-4- Proposed Solution

The proposed Aspect-BiLSTM model is a supervised deep learning model containing a BiLSTM layer, two hidden layers, and a multi-label output layer indicating the presence of each aspect in its input texts. The framework of the proposed solution is depicted in Figure 5.

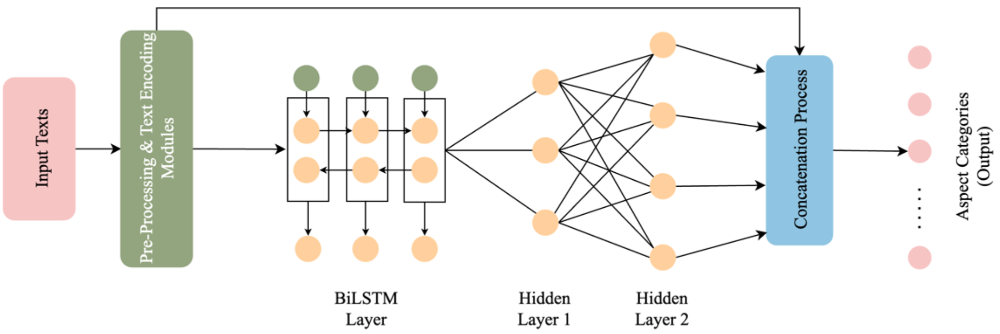

Figure 5. Framework of proposed solution

Once the initial word embeddings were generated, they were sent to the model's BiLSTM layer, where their forward and backward sequential properties were extracted. This layer captures richer contextual features of words in its input texts with respect to their aspect features. These sequential representations are then sent to the model's hidden layers, where the final aspect representations of the input texts are generated. These representations were computed using non-linear functions, which are depicted in the formula below.

$$h_j = \sum_{i=1}^{n} W_i X_i + b \tag{1}$$

where h_j represents the j^{th} element in the aspect representation, n represents the total number of input features for the layer, W_i represents the weight for feature X_i, and b represents the bias term used.

Once the aspect representations of the input texts were generated, they were then concatenated with the initial BERT word embeddings and were then sent to its output layer, where the final aspect output labels were generated. The output values generated by this layer are computed as follows:

$$\hat{y}_j = \sigma\left(\sum_{i=1}^{n} W_i X_i + b\right) \tag{2}$$

where $\hat{y}_j$ represents the j^{th} value in the output label, σ represents the sigmoid non-linear function, n represents the number of elements in the extracted sequential representations of the input feature, W_i represents the weight of feature X_i, and b represents the bias term used.

4- Experiment

4-1- Dataset

The SemEval-2014 [34] and SemEval-2016 [41] restaurant datasets were used in this research. These datasets consisted of restaurant reviews, their aspects, and sentiment polarities. From the statistics highlighted in Table 3, the SemEval-2014 Restaurant dataset contained 3,844 reviews, with 3,044 training reviews, while the remaining 800 reviews were allocated for testing. On the other hand, the SemEval-2016 Restaurant dataset contained 2,676 reviews, with 2,000 allocated to its training dataset and the remaining 676 reviews allocated to its testing dataset. All the reviews in this dataset were annotated with their respective aspect labels. In this experiment, we have extracted the labels from the dataset, as shown in Figure 6. A standardized set of aspect label classes, namely ambiance, food, general, location, price, and service, were employed as outputs for the reviews in this study. It should be noted that the general label used in the experiment covered the anecdotes/miscellaneous class from the SemEval-2014 Restaurant dataset as well as the RESTAURANT class from the SemEval-2016 Restaurant dataset, thereby ensuring consistency in the aspect label classes across the two datasets.

Table 3. Statistics of the datasets

Dataset	Total Reviews	Training Reviews	Testing Reviews	Training Aspects	Testing Aspects
SemEval-2014 Restaurant reviews	3,844	3,044	800	3,714	1,025
SemEval-2016 Restaurant reviews	2,676	2,000	676	2,227	729
Total	6,520	5,044	1,476	5,941	1,754

Figure 6. Original vs. modified aspect label (SemEval-2016 Restaurant reviews)

The 3,044 training reviews in the SemEval-2014 Restaurant contained 3,714 aspects, while the 800 testing reviews in the dataset had 1,025 aspects. On the other hand, the 2,000 training reviews in the SemEval-2016 Restaurant dataset contained 2,227 aspects, while the 676 testing reviews contained 729 aspects. It is worth noting that the number of aspects can vary from the number of reviews, as there were cases where multiple aspects were extracted from a single review, and there were also cases where some reviews did not have any aspects.

4-2- Aspect Extraction Models Implementation

The machine learning models used in the experiment were developed using scikit-learn [42], while the deep learning models used in this experiment were developed using the Tensorflow framework [43]. Hyperparameter tweaking was done to find the ideal hyperparameter values that produced the optimum performance. Samples from the SemEval-2014 Restaurant and SemEval-2016 Restaurant datasets were used to tweak the hyperparameters of the models. Table 4 outlines the hyperparameters used in the experiment, including the tested values and the best value for each hyperparameter. Tables 5 and 6 display the optimal hyperparameters for each model, while Table 7 lists the constant hyperparameters used for the hidden and output layers across all deep learning models evaluated in the study. In addition, the maximum number of training epochs was set to 100 for all deep-learning models.

Table 4. Possible hyperparameter configurations

Hyperparameter	Tested Values
Neighbours (kNN)	3, 4, 5, 6, 7
Weights (kNN)	Uniform, Distance
Distance measuring (DT)	Manhattan, Euclidean
Quality of split (DT)	Gini, Log Loss, Entropy
Splitting strategy (DT)	Best, Random
Max features (DT)	None, Square Root, Log Base 2
Class weights (DT)	None, Balanced
Numbers of filters (CNN)	64, 128, 256
Stride (CNN)	2, 3, 4
Pool size (CNN)	2, 3, 4
Neurons (RNN, GRU, LSTM, BiLSTM)	Adam, Stochastic Gradient Descent (SGD), Root Mean Squared Propagation (RMSProp)

Table 5. Optimal hyperparameters of the machine learning models

Hyperparameter	kNN	DT
Neighbours	6	-
Weights	Distance	-
Distance measuring	Manhattan	-
Quality of split	-	Gini
Splitting strategy	-	Best
Max features	-	None
Class weights	-	None

Table 6. Optimal hyperparameters of the deep learning models

Hyperparameter	MLP	CNN	RNN	GRU	LSTM	BiLSTM	Aspect-BiLSTM
Number of filters		128					
Strides		2					
Pool size		4					
Neurons			128	32	128	64	
Optimiser	SGD	RMSProp	Adam	SGD	Adam	Adam	SGD
Learning rate	0.001	0.00001	0.00001	0.00001	0.0001	0.001	0.001

Table 7. Hidden and output layer hyperparameters

Hyperparameter	Value
Neurons	120 (hidden_1) / 480 (hidden_2) / 6 (output)
Activations	None (hidden_1 & hidden_2) / sigmoid (output)

The hyperparameter settings listed in Tables 5 and 6 were optimal, as they produced the highest testing F1 scores during the hyperparameter tuning process. The k-nearest Neighbour model performed optimally with 6 neighbours and distance weights, as well as using the Manhattan distance formula. In contrast, the Decision Tree model used the Gini function to measure the quality of the split and used the best splitting strategy to do so. Besides this, the CNN model performed optimally with 128 filters, a stride of 2, and a pool size of 4. Meanwhile, the RNN and LSTM models achieved the highest scores with 128 neurons, while the GRU and BiLSTM models performed best with 32 and 64 neurons, respectively.

Regarding optimizers, the models achieved optimal performance with the following settings: Stochastic Gradient Descent (SGD) for the MLP, GRU, and Aspect-BiLSTM; Root Mean Squared Propagation (RMSProp) for CNN; and Adam for RNN, LSTM, and BiLSTM. In terms of learning rates, the MLP, LSTM, BiLSTM, and Aspect-BiLSTM performed optimally with a learning rate of 0.001, while the CNN and RNN models achieved optimal results with a learning rate of 0.00001. On the other hand, the GRU model achieved the best results with a learning rate of 0.0001.

4-3- Baseline Solutions

The evaluation results of the aspect extraction solutions proposed in the studies listed below acted as baseline results for the comparative analysis conducted in this study. These solutions were selected from recently published works on either aspect extraction or aspect-level sentiment analysis.

He et al. (2023) [7]: The aspect extraction solution proposed in this study consisted of an MLP model, which was trained using a self-training mechanism. This mechanism consisted of a teacher model that was trained on gold-annotated data to generate pseudo labels for the remaining unannotated data. The student model and a meta-weighter were then trained on both the gold-annotated and pseudo-annotated data.

Karimi et al. (2021) [8]: The solution proposed in this study consisted of a fine-tuned BERT model, which was trained using the adversarial training technique. The fine-tuning process in this study consisted of adding an additional fully connected MLP layer to the BERT encoder's architecture, which predicted the presence of aspects in its input texts.

Khan et al. (2023) [44]: The supervised aspect extraction solution proposed in this study consisted of a CNN model for extracting significant aspect features from its input word embeddings, a modified LSTM model for extracting the sequential features of its input texts, and an attention mechanism that identified the most significant words in its input texts as well as assigned the appropriate weights to them.

Kumar et al. (2021) [45]: The solution proposed in this study consisted of a supervised deep-learning model that fused multiple information types together. First, the solution concatenated its input texts' general and domain-specific word embeddings before passing it to its BiLSTM layer. The solution then utilized an attention mechanism to generate the global contextual word embeddings from the generated BiLSTM hidden states and concatenated them before passing them through another attention mechanism.

Wan et al. (2020) [46]: The aspect extraction solution proposed in this study consisted of a fine-tuned BERT model for aspect term extraction. The fine-tuning process consisted of adding two fully connected MLP layers to the encoder's architecture and two separate output layers. The first output layer indicated the presence of aspects and sentiments in the solution's input sentences, while the second output layer generated an output sequence highlighting the aspect terms in them.

4-4- Results

This subsection highlights the experimental results of the selected machine and deep learning models along with the proposed solution when compared against each other. Table 8 highlights the metrics used to evaluate the performance of these models along with their descriptions which are based on those provided by Anbananthen et al. (2022) [47].

Table 8. Evaluation metrics

Metric	Description
Accuracy (A)	The percentage of correct samples obtained out of the total number of samples. $A = \frac{TP + TN}{N}$
Precision (P)	The percentage of correct positive results out of the total number of positive predictions. $P = \frac{TP}{TP + FP}$
Recall (R)	The percentage of correct positive results out of the total number of true positive samples. $R = \frac{TP}{TP + FN}$
F1 Score (F1)	The harmonic mean between the precision and recall scores. $F1 = 2 \times \frac{P \times R}{P + R}$

N: Total number of samples
TP: Number of true positive predictions
FP: Number of false positive predictions
FN: Number of false negative predictions
TN: Number of true negative predictions

These metrics were selected due to their frequent use in similar aspect extraction studies, enabling the performance comparison between the baseline solutions and the proposed aspect extraction solution in this study. Tables 9 and 10 as well as Figures 7 and 8, highlight the performance comparison of the selected machine and deep learning models when evaluated on the SemEval-2014 Restaurant and SemEval-2016 Restaurant datasets, respectively.

Table 9. SemEval-2014 Restaurant experimental results

Model	Accuracy	Precision	Recall	F1 Score
kNN	87.9	78.4	59.5	67.7
DT	79.2	51.3	50.5	50.9
MLP	91.8	82.8	77.9	80.2
CNN	91.4	80.1	79.3	79.7
RNN	91.0	79.9	77.5	78.7
GRU	91.7	82.9	77.3	80.0
LSTM	91.7	82.7	77.3	79.9
BiLSTM	92.5	**83.5**	80.9	82.2
Aspect-BiLSTM	**92.9**	83.0	**84.0**	**83.5**

Table 10. SemEval-2016 Restaurant experimental results

Model	Accuracy	Precision	Recall	F1 Score
kNN	86.2	64.4	51.6	57.3
DT	78.8	41.2	42.2	41.7
MLP	90.7	**72.8**	68.4	70.6
CNN	90.1	**72.8**	71.3	72.1
RNN	88.9	69.2	68.6	68.9
GRU	72.8	34.5	31.3	32.8
LSTM	88.7	68.1	70.0	69.0
BiLSTM	88.3	65.7	72.8	69.1
Aspect-BiLSTM	**90.7**	72.5	**77.6**	**75.0**

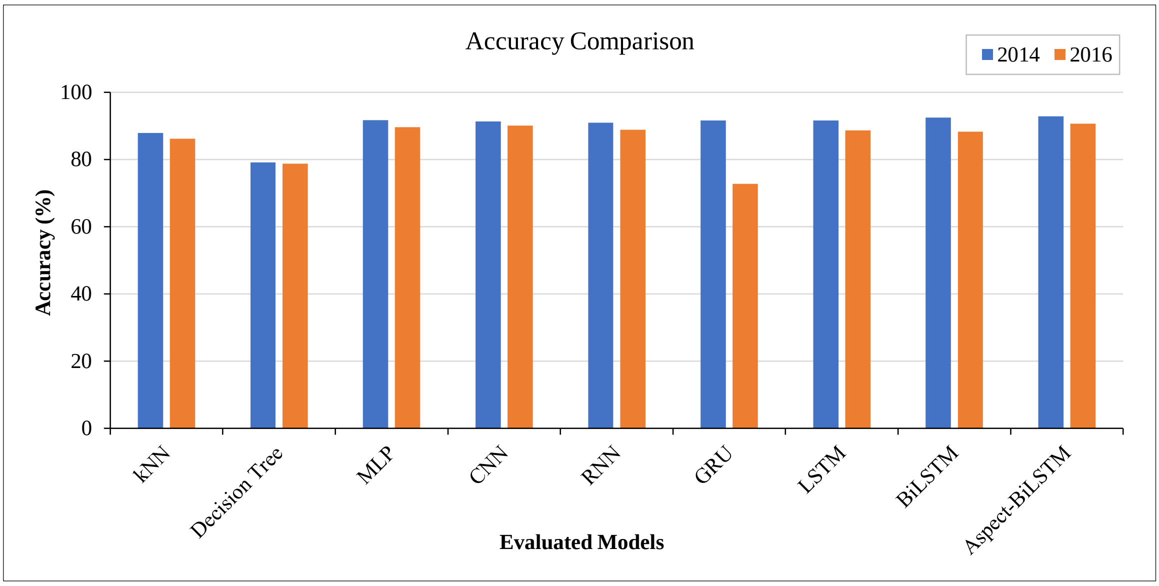

Figure 7. Accuracy comparison between all the models evaluated on both datasets

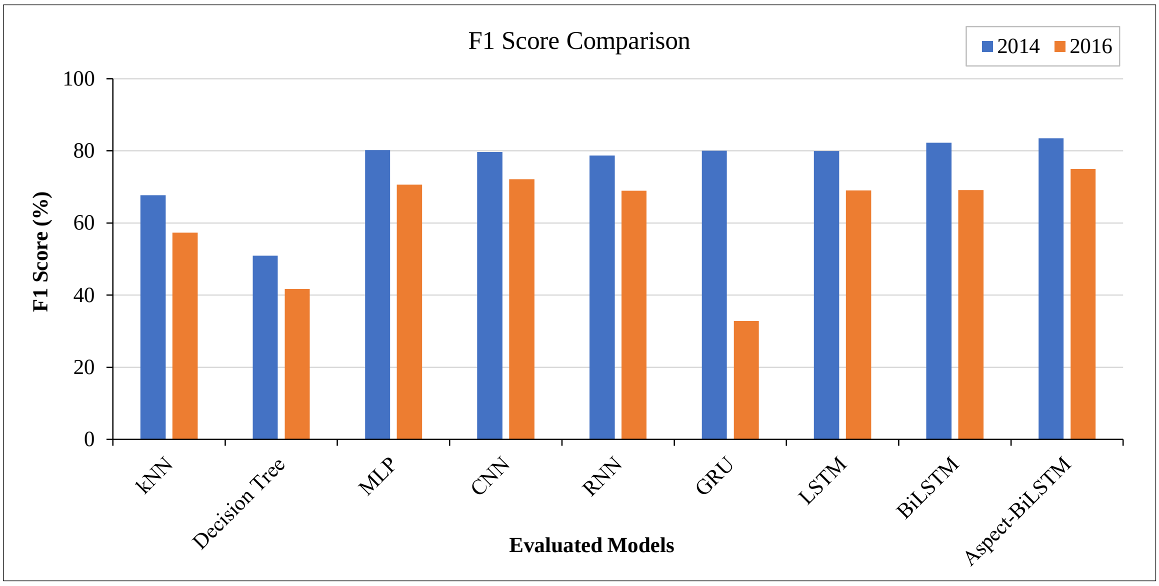

Figure 8. F1 score comparison between all the models evaluated on both datasets

Several observations can be made from the evaluation results of the models depicted in Table 9 (SemEval-2014 Restaurant). Firstly, the best-performing model in terms of accuracy, recall, and F1 score was the proposed Aspect-BiLSTM model, as it obtained scores of 92.9%, 84.0%, and 83.5%, respectively. However, it did not obtain the highest precision score, as the evaluated BiLSTM model was able to outscore it by half a percent with a score of 83.5%. The worst-performing model was the Decision Tree model, which obtained the worst metrics scores. It only obtained an accuracy of 79.25%, a precision score of 51.3%, a recall score of 50.5%, and an F1 score of 50.9%. The kNN model, on the other hand, outperformed the Decision Tree model as it obtained an accuracy of 87.9%, a precision score of 78.4%, a recall score of 59.5%, and an F1 score of 67.7%. However, it was not able to achieve the same level of performance as its deep learning counterparts, which shows a discrepancy in performance between the evaluated machine and deep learning models.

On the other hand, it can be observed in Table 10 (SemEval-2015 Restaurant) that the best-performing model was also the proposed Aspect-BiLSTM model, as it was able to obtain an accuracy of 90.7%, a recall score of 77.6%, and an F1 score of 75.5%. Again, it did not obtain the highest precision score, as the MLP and CNN models were able to outperform it with scores of 72.8%. The Decision Tree model was again considered to be the worst-performing model on this dataset, as it was only able to achieve an accuracy of 78.8%, a recall score of 42.2%, and an F1 score of 41.7%. However, it did not obtain the worst precision score, as that was obtained by the evaluated GRU model with 34.5% precision. The results in this table have also shown a discrepancy in performance between the evaluated machine and deep learning models, as the kNN and Decision Tree models did not match the performance of their deep learning counterparts.

4-5- Discussion

The experimental evaluation conducted has demonstrated the effectiveness of several machines and deep learning models in extracting aspects from unstructured texts. The first observation that can be made from the results is that both of the machine learning models (kNN and Decision Tree) were not as effective as their deep learning counterparts in extracting aspects from texts. This can be attributed to their lack of aspect feature extraction pipelines in their architectures, which prevented them from extracting vital aspect features from their input texts. The lack of an aspect extraction pipeline also hindered their generalization abilities as well.

On the other hand, the deep learning models that have been the most effective in extracting aspects from texts (besides the proposed Aspect-BiLSTM solution) were the BiLSTM and CNN models, as they obtained the highest F1 scores on the SemEval-2014 and SemEval-2016 restaurant datasets, respectively. Their effectiveness in the task has highlighted the importance of extracting aspect features from its input texts, particularly the convolved and sequential features from them. This was what led to the idea of proposing an additional model that leverages these types of features for the task of aspect extraction.

The proposed Aspect-BiLSTM model was developed to extract the sequential properties of its input texts while retaining their semantic, syntactic, and contextual properties presented in the encoded BERT word embeddings. The model accomplished this by concatenating the extracted aspect features to the word embedding features, which acted as a weight indicating the presence of both explicit and implicit aspects in its input texts.

To demonstrate the proposed model's effectiveness in extracting both types of aspects, a comprehensive manual analysis of the model's predicted output labels was conducted. The data samples used in this analysis consisted of 50 reviews, with 25 of them each being selected from both the SemEval-2014 Restaurant dataset as well as the SemEval-2016 Restaurant dataset. The selected reviews consisted of 300 tokens, with 57 classified as aspects. Out of these 57 aspect tokens, 34 of them were explicit aspects, while the remaining 23 of them were implicit aspects.

The performance of the Aspect-BiLSTM model in terms of explicit and implicit aspect extraction can be seen in the confusion matrix presented in Table 11. This confusion matrix presents the numbers of explicit, implicit, and non-aspects that were truly present in the dataset (actual) and predicted (predicted) by the model. Non-aspects in the context of this analysis were tokens that were considered to be non-aspects. Based on the values presented in the matrix above, it can be seen that Aspect-BiLSTM was able to accurately predict both explicit and implicit aspects within texts, as it was able to predict 33 out of the 34 explicit aspects as well as 18 out of the 23 implicit aspects in the dataset. The model, however, made 6 false negative predictions (1 for explicit and 5 for implicit) and 11 false positive (6 for explicit and 5 for implicit) predictions, highlighting the potential areas of improvement that can be made to the model's performance. Despite this, however, the overall performance of the model in this analysis indicates its high reliability in extracting aspects.

Table 11. Confusion matrix of the Aspect-CNN-BiLSTM model for explicit and implicit aspect extraction

		Actual		
		Explicit	**Implicit**	**Non-Aspects**
Predicted	Explicit	33	0	6
	Implicit	0	18	5
	Non-Aspects	1	5	232

The performance comparison of the model for explicit and implicit aspect extraction, as depicted in Table 12, has demonstrated how Aspect-BiLSTM has excelled in extracting both types of aspects. It was able to achieve accuracy scores of 97.10% in explicit aspect extraction and 78.30% in implicit aspect extraction. Besides this, the proposed solution obtained F1 scores of 93.00% in explicit aspect extraction and 83.70% in implicit aspect extraction. The findings suggest that the proposed Aspect-BiLSTM model is an effective algorithm for aspect-level sentiment analysis tasks with potential practical applications. Furthermore, the model demonstrated consistent and accurate predictions of the aspects in texts that do not mention entities, which is challenging. These findings provide valuable insights into the potential of the proposed model for aspect-level sentiment analysis tasks.

Table 12. Performance comparison for explicit and implicit aspect extraction

Aspect Type	Accuracy	F1 Score
Explicit	97.1	93.0
Implicit	78.3	83.7

Lastly, a comparative analysis was conducted between the performance of the baseline solutions evaluated on the same datasets and the performance of Aspect-BiLSTM. Tables 13 and 14 provide the evaluation F1 scores of the baseline solutions and the proposed solution when evaluated on the SemEval-2014 and SemEval-2016 Restaurant datasets, respectively.

Table 13. Performance comparison between the baseline solutions and proposed solution on the SemEval-2014 Restaurant dataset

Aspect Extraction Solution	F1
He et al. (2023) [7]	88.95
Karimi et al. (2021) [8]	85.57
Kumar et al. (2021) [45]	**89.96**
Aspect-BiLSTM (Proposed Solution)	83.50

Table 14. Performance comparison between the baseline solutions and proposed solution on the SemEval-2016 Restaurant dataset

Aspect Extraction Solution	F1
Karimi et al. (2021) [8]	81.50
Khan et al. (2020) [44]	79.10
Wan et al. (2020) [46]	**82.77**
Aspect-BiLSTM (Proposed Solution)	75.00

Although the proposed Aspect-BiLSTM solution did not outperform the baseline solutions in terms of prediction F1 score, the findings from this study can act as a foundation for future work into the aspect extraction process as it explores the types of features that best represent both explicit and implicit aspects in text data.

While Aspect-BiLSTM can effectively extract explicit and implicit aspects from unstructured texts, certain limitations could be further enhanced. The main limitation of the model is its inability to fully capture the complex nature of written languages, making it difficult to accurately identify certain aspects, especially when implicit aspects are subtle. Further research will need to be conducted to enhance the contextual representations captured by the model, especially when handling domain-specific nuances in texts, as well as to improve the model's ability to identify implicit aspects.

5- Conclusion

This study has presented a comprehensive investigation into the efficacy of certain machine and deep learning models in extracting explicit and implicit aspects from text data. It can be observed from the experimental evaluation that the evaluated deep learning models generally performed better than their machine learning counterparts, which can be attributed to their aspect feature extraction pipelines. This was what led to the development of the proposed Aspect-BiLSTM model for aspect extraction, which utilized the syntactic, semantic, and contextual properties of words depicted in BERT word embeddings along with additional sequential properties of its input texts that were extracted using the evaluated BiLSTM model. This has resulted in a solution that could effectively extract both explicit and implicit aspects from its input texts, given its F1 scores of 83.5% on the SemEval-2014 Restaurant dataset and 75.0% on the SemEval-2016 Restaurant dataset. These findings have profound implications for researchers and practitioners in natural language processing, particularly sentiment analysis. Future work for this research could involve mapping the extracted aspects to their corresponding polarity (positive, negative, or neutral) for sentiment analysis. This could provide a more nuanced understanding of the sentiment expressed towards specific aspects of a subject, which could be valuable for various applications, such as product or service reviews. Additionally, exploring the effectiveness of the Aspect-BiLSTM model in other domains and datasets could further validate its potential as a robust and efficient algorithm for aspect-based sentiment analysis.

6- Declarations

6-1- Author Contributions

Conceptualization, M.M.A.B. and K.S.M.A.; methodology, M.M.A.B.; validation, S.K. and R.K.; formal analysis, M.M.A.B.; investigation, R.K.; writing—original draft preparation, M.M.A.B. and K.S.M.A; writing—review and editing, S.K. and R.K.; supervision, K.S.M.A. and S.K.; funding acquisition, K.S.M.A. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available in the article.

6-3- Funding

This work was supported by MMU & Telekom Malaysia (TMR&D) under Grant MMUE/210038.

6-4- Institutional Review Board Statement

Not applicable.

6-5- Informed Consent Statement

Not applicable.

6-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Anbananthen, K. S. M., & Elyasir, A. M. H. (2013). Evolution of opinion mining. Australian Journal of Basic and Applied Sciences, 7(6), 359-370.

[2] Jonnalagadda, P., Hari, K. P., Batha, S., & Boyina, H. (2019). A rule based sentiment analysis in Telugu. International Journal of Advance Research, Ideas and Innovations in Technology, 2(5), 387-390.

[3] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093–1113. doi:10.1016/j.asej.2014.04.011.

[4] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A Survey. WIREs Data Mining and Knowledge Discovery, 8(4), e1253. doi:10.1002/widm.1253.

[5] Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 168-177. doi:10.1145/1014052.1014073.

[6] Singh Chauhan, G., Kumar Meena, Y., Gopalani, D., & Nahta, R. (2020). A two-step hybrid unsupervised model with attention mechanism for aspect extraction. Expert Systems with Applications, 161, 113673–113686. doi:10.1016/j.eswa.2020.113673.

[7] He, K., Mao, R., Gong, T., Li, C., & Cambria, E. (2023). Meta-Based Self-Training and Re-Weighting for Aspect-Based Sentiment Analysis. IEEE Transactions on Affective Computing, 14(3), 1731–1742. doi:10.1109/TAFFC.2022.3202831.

[8] Karimi, A., Rossi, L., & Prati, A. (2021). Adversarial Training for Aspect-Based Sentiment Analysis with BERT. 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy. doi:10.1109/icpr48806.2021.9412167.

[9] Maitama, J. Z., Idris, N., Abdi, A., Shuib, L., & Fauzi, R. (2020). A Systematic Review on Implicit and Explicit Aspect Extraction in Sentiment Analysis. IEEE Access, 8, 194166–194191. doi:10.1109/access.2020.3031217.

[10] Zhang, M., & Qian, T. (2020). Convolution over Hierarchical Syntactic and Lexical Graphs for Aspect Level Sentiment Analysis. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 3540-3549. doi:10.18653/v1/2020.emnlp-main.286.

[11] Zhou, J., Huang, J. X., Hu, Q. V., & He, L. (2020). SK-GCN: Modeling Syntax and Knowledge via Graph Convolutional Network for aspect-level sentiment classification. Knowledge-Based Systems, 205. doi:10.1016/j.knosys.2020.106292.

[12] Anbananthen, S. K., Sainarayanan, G., Chekima, A., & Teo, J. (2006). Data Mining using Pruned Artificial Neural Network Tree (ANNT). IEEE 2nd International Conference on Information & Communication Technologies, Damascus, Syria. doi:10.1109/ICTTA.2006.1684577.

[13] Zainuddin, N., Selamat, A., & Ibrahim, R. (2018). Hybrid sentiment classification on twitter aspect-based sentiment analysis. Applied Intelligence, 48(5), 1218–1232. doi:10.1007/s10489-017-1098-6.

[14] Schouten, K., van der Weijde, O., Frasincar, F., & Dekker, R. (2018). Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis with Co-occurrence Data. IEEE Transactions on Cybernetics, 48(4), 1263–1275. doi:10.1109/TCYB.2017.2688801.

[15] Rana, T. A., Cheah, Y.-N., & Rana, T. (2020). Multi-level knowledge-based approach for implicit aspect identification. Applied Intelligence, 50(12), 4616–4630. doi:10.1007/s10489-020-01817-x.

[16] Venugopalan, M., & Gupta, D. (2020). An Unsupervised Hierarchical Rule Based Model for Aspect Term Extraction Augmented with Pruning Strategies. Procedia Computer Science, 171, 22–31. doi:10.1016/j.procs.2020.04.303.

[17] Li, X., Wang, B., Li, L., Gao, Z., Liu, Q., Xu, H., & Fang, L. (2020). Deep2s: Improving Aspect Extraction in Opinion Mining with Deep Semantic Representation. IEEE Access, 8, 104026–104038. doi:10.1109/ACCESS.2020.2999673.

[18] Cilibrasi, R. L., & Vitányi, P. M. B. (2007). The Google similarity distance. IEEE Transactions on Knowledge and Data Engineering, 19(3), 370–383. doi:10.1109/TKDE.2007.48.

[19] Dimopoulos, Y., Nebel, B., Koehler, J. (1997). Encoding planning problems in non-monotonic logic programs. Recent Advances in AI Planning. ECP 1997. Lecture Notes in Computer Science, Volume 1348. Springer, Berlin, Germany. doi:10.1007/3-540-63912-8_84.

[20] Langkilde, I., & Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1, 704–710. doi:10.3115/980845.980963.

[21] Tulkens, S., & van Cranenburgh, A. (2020). Embarrassingly simple unsupervised aspect extraction. arXiv Preprint, arXiv:2004.13580. doi:10.48550/arXiv.2004.13580.

[22] Luo, L., Ao, X., Song, Y., Li, J., Yang, X., He, Q., & Yu, D. (2019). Unsupervised Neural Aspect Extraction with Sememes. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), 5123-5129. doi:10.24963/ijcai.2019/712.

[23] Venugopalan, M., & Gupta, D. (2022). An enhanced guided LDA model augmented with BERT based semantic strength for aspect term extraction in sentiment analysis. Knowledge-Based Systems, 246. doi:10.1016/j.knosys.2022.108668.

[24] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv Preprint, arXiv:1301.3781. doi:10.48550/arXiv.1301.3781.

[25] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[26] Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). Inference of population structure using multilocus genotype data. Genetics, 155(2), 945–959. doi:10.1093/genetics/155.2.945.

[27] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5–6), 602–610. doi:10.1016/j.neunet.2005.06.042.

[28] Akhtar, M. S., Garg, T., & Ekbal, A. (2020). Multi-task learning for aspect term extraction and aspect sentiment classification. Neurocomputing, 398, 247–256. doi:10.1016/j.neucom.2020.02.093.

[29] Ray, P., & Chakrabarti, A. (2022). A Mixed approach of Deep Learning method and Rule-Based method to improve Aspect Level Sentiment Analysis. Applied Computing and Informatics, 18(1–2), 163–178. doi:10.1016/j.aci.2019.02.002.

[30] Cai, H., Tu, Y., Zhou, X., Yu, J., & Xia, R. (2020). Aspect-Category based Sentiment Analysis with Hierarchical Graph Convolutional Network. Proceedings of the 28th International Conference on Computational Linguistics, 833-843. doi:10.18653/v1/2020.coling-main.72.

[31] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2323. doi:10.1109/5.726791.

[32] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv Preprint, arXiv:1810.04805. doi:10.48550/arXiv.1810.04805.

[33] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408. doi:10.1037/h0042519.

[34] Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2014). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. 8th International Workshop on Semantic Evaluation (SemEval-2014), 27–35. doi:10.3115/v1/S14-2004.

[35] Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. Procedia Computer Science, 17, 26–32. doi:10.1016/j.procs.2013.05.005.

[36] Wang, S., Zhou, W., & Jiang, C. (2020). A survey of word embeddings based on deep learning. Computing, 102(3), 717–740. doi:10.1007/s00607-019-00768-7.

[37] Fix, E., & Hodges, J. L. (1989). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. International Statistical Review / Revue Internationale de Statistique, 57(3), 238. doi:10.2307/1403797.

[38] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). Classification and Regression Trees. Routledge, New York, United States. doi:10.1201/9781315139470.

[39] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. doi:10.1038/323533a0.

[40] Cho, K., Merrienboer, B.v., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. arXiv. doi:10.48550/arXiv.1409.1259.

[41] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., Clercq, O. D., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., & Eryiğit, G. (2016). SemEval-2016 Task 5: Aspect Based Sentiment Analysis. 10th International Workshop on Semantic Evaluation (SemEval-2016), 19–30. doi:10.18653/v1/S16-1002.

[42] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. The Journal of Machine Learning Research, 12, 2825-2830.

[43] Tensorflow. (2022). Create production-grade machine learning models with TensorFlow. Version v2.15.0. Available online: https://zenodo.org/records/10126399 (accessed on February 2024).

[44] Khan, M. U., Javed, A. R., Ihsan, M., & Tariq, U. (2023). A novel category detection of social media reviews in the restaurant industry. Multimedia Systems, 29(3), 1825–1838. doi:10.1007/s00530-020-00704-2.

[45] Kumar, A., Veerubhotla, A. S., Narapareddy, V. T., Aruru, V., Neti, L. B. M., & Malapati, A. (2021). Aspect term extraction for opinion mining using a Hierarchical Self-Attention Network. Neurocomputing, 465, 195–204. doi:10.1016/j.neucom.2021.08.133.

[46] Wan, H., Yang, Y., Du, J., Liu, Y., Qi, K., & Pan, J. Z. (2020). Target-Aspect-Sentiment Joint Detection for Aspect-Based Sentiment Analysis. Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 9122–9129. doi:10.1609/aaai.v34i05.6447.

[47] Anbananthen, K. S. M., Busst, M. B. M. A., Kannan, R., & Kannan, S. (2023). A Comparative Performance Analysis of Hybrid and Classical Machine Learning Method in Predicting Diabetes. Emerging Science Journal, 7(1), 102–115. doi:10.28991/ESJ-2023-07-01-08.