Dhirubhai Ambani University

Sentiment Trajectory: Understanding and Predicting Uber Review Trends

Aayan Kumar* **Heer Chokshi*** **Siddharth Kumar *** **Yash Deshmukh**

202418001@daiict.ac.in, 202418019@daiict.ac.in, 202418054@daiict.ac.in, 202418063@daiict.ac.in

Abstract

This project analyzes Uber customer reviews using Natural Language Processing to classify sentiment and applies time series forecasting to predict future sentiment trends. By combining sentiment analysis with predictive modeling, we provide insights into evolving customer satisfaction, helping ride-sharing platforms enhance service quality proactively.

Introduction

In today's digitally driven economy, ride-sharing platforms like Uber rely heavily on user-generated content to gauge service quality and customer satisfaction. Among these, customer reviews play a critical role in shaping operational and strategic decisions. However, analyzing and making sense of thousands of free-text reviews presents a significant challenge. This project addresses that gap by combining sentiment analysis and forecasting to extract actionable insights from Uber reviews.[1] We employ three advanced aspect extraction approaches—Triplet Extraction, BERT, and CAT—and benchmark them against standard baseline and state-of-the-art (SOTA) models. These aspects were analysed by RoBERTa. These models not only classify sentiment but also capture nuanced relationships between service aspects and user opinions. To further support decision-making, we apply time series forecasting using LSTM and GRU networks to predict future sentiment trends based on historical review data. By integrating sentiment analysis with forecasting, our goal is to provide Uber and similar ride-sharing services with predictive insights into customer satisfaction. This forward-looking perspective allows businesses to address potential issues proactively, tailor user experiences, and improve overall service quality.

Methodology

Our methodology integrates advanced Natural Language Processing (NLP) and deep learning techniques to perform sentiment analysis and forecast customer sentiment trends in Uber reviews. The workflow consists of five key stages: data preprocessing, aspect extraction, sentiment analysis, forecasting, and model evaluation.

*These authors contributed equally.

Data Preprocessing

The raw dataset of Uber customer reviews was subjected to a structured preprocessing pipeline to ensure textual uniformity and facilitate downstream natural language analysis. Initially, all null entries were removed to maintain data integrity. The textual content of the reviews was then standardized by converting all characters to lowercase, thereby eliminating inconsistencies arising from case sensitivity. To further sanitize the data, special characters and punctuation marks were stripped from the text, retaining only alphanumeric tokens essential for semantic interpretation. Subsequently, tokenization was performed using the SpaCy library, which provided linguistically informed segmentation of the text into individual tokens. This step ensured accurate representation of the textual data for further modeling tasks such as sentiment classification and temporal forecasting. The resulting preprocessed corpus formed a clean and consistent input suitable for embedding generation and advanced linguistic modeling.

Aspect Extraction

To effectively identify and standardize service-related aspects expressed in user reviews, we implemented three complementary extraction techniques: syntactic triplet extraction, BERT-based sequence labeling, and a contrastive attention mechanism. These methods collectively enhance coverage and robustness across varied linguistic structures found in customer feedback.

Method 1: Triplet Extraction via POS and Dependency Parsing This method uses rule-based linguistic heuristics to construct (Aspect, Sentiment, Opinion) triplets from syntactic dependencies.[4][6]

Tools Used: SpaCy, NLTK

Procedure:

1. Tokenized reviews and tagged them with part-of-speech (POS) labels.
2. Identified nouns as candidate aspects.
3. Extracted associated opinion words (adjectives or verbs) using dependency parsing.
4. Heuristically constructed structured triplets.
5. Applied normalization techniques to standardize aspect terms.

This method works well for grammatically clear reviews with explicit aspect-opinion relationships.

Method 2: Aspect Term Extraction via BERT with BIO Tagging This approach frames aspect extraction as a token classification task using contextualized language models.

Model Used: BERT base (uncased), fine-tuned on 5500 reviews for BIO tagging

Procedure:

1. Annotated tokens using the BIO scheme (B-ASP, I-ASP, O).
2. Encoded input using `BertTokenizerFast`.
3. Trained a token classification model using `CrossEntropyLoss`.
4. Extracted aspect spans from the model's token-level predictions.
5. Normalized results using a curated mapping to canonical terms.

This model leverages contextual understanding, making it effective for identifying nuanced or implicit aspects.

Method 3: Contrastive Attention-Based Extraction A lightweight yet effective method combining distributional semantics and kernel-based attention mechanisms.

Model Used: Word2Vec + Radial Basis Function (RBF) attention

Procedure:

1. Trained Word2Vec embeddings on the review corpus.
2. Defined a seed list of canonical aspect terms.
3. Created an aspect matrix from the corresponding word vectors.
4. Calculated contrastive attention scores for each word in a review using an RBF kernel.
5. Selected tokens with high attention weights as aspect candidates.
6. Mapped results to canonical forms using a predefined dictionary.

This method is particularly beneficial in handling informal or noisy text where aspects are contextually implied.

Aspect Normalization To ensure consistency across all three extraction techniques, we implemented a unified aspect normalization process:

Challenge: Linguistic variability resulted in synonyms (e.g., *driver*, *chauffeur*) and domain-specific terms requiring consolidation.

Solution:

1. Created a manually curated CSV-based mapping of variant terms to canonical aspects.
2. Augmented this mapping automatically using Word2Vec similarity scores, filtering terms with a cosine similarity above a set threshold (e.g., 0.75).
3. Applied this unified mapping across all extracted outputs to ensure semantic alignment and reduce redundancy.

This normalization step is critical for generating consistent, interpretable insights and for enabling accurate downstream analysis such as sentiment aggregation and temporal forecasting.

Sentiment Analysis

To evaluate user sentiment with respect to specific service components, aspect-based sentiment analysis was conducted using a transformer-based classification model.

Model and Configuration A pre-trained RoBERTa sentiment classifier was employed with finetuning on approximately 500 Uber reviews . The model outputs categorical sentiment labels: *positive*, *neutral*, or *negative*.

Procedure

1. **Input Construction:** For each (aspect, review) pair, the input text was formatted as `[CLS] aspect [SEP] review`, enabling the model to contextualize the aspect within the narrative of the review.
2. **Prediction:** The constructed input was passed through RoBERTa to infer the sentiment polarity associated with the given aspect.
3. **Output Storage:** The predicted sentiment labels were stored alongside their corresponding aspects in a new field `aspect_sentiment_pairs`, forming the basis for subsequent analysis.

Data Structuring To facilitate robust sentiment attribution:

1. **Decomposition of Multi-Aspect Reviews:** Reviews containing multiple extracted aspects were disaggregated into individual rows, each aligned to a single aspect to maintain interpretability.
2. **Handling Unassigned Aspects:** In cases where no aspects were detected, a default placeholder aspect *"uber"* was assigned, ensuring that all reviews were incorporated into the sentiment pipeline.

This approach enabled fine-grained sentiment attribution, providing actionable insights into user perceptions across distinct service dimensions.

Forecasting

To complement aspect-based sentiment analysis with forward-looking insights, we implemented time-series forecasting to predict future sentiment trends across key service dimensions. This enables Uber to anticipate shifts in customer satisfaction and take timely corrective actions.

Objective The primary goal was to model and forecast monthly sentiment trajectories for high-impact service aspects such as *driver*, *app*, *fare*, and *ride experience*. By anticipating sentiment trends, the platform can engage in proactive service optimization.

Data Preparation and Encoding

1. **Sentiment Indexing:** Sentiment Index is a quantitative measure of user sentiment derived from text reviews.

$$\text{Sentiment Index} = \text{Positive Score} - \text{Negative Score}$$

2. **Sentiment Encoding:** To quantify sentiment, we assigned numerical values as follows: `+1` for *positive*, `0` for *neutral*, and `−1` for *negative*. These values were averaged per aspect per month to produce a continuous sentiment index.

3. **Normalization:** Time-series data was normalized to zero mean and unit variance to stabilize training and ensure comparability across aspects.

Modeling and Training Three recurrent neural network architectures were trained independently for each aspect:

- **Vanilla RNN:** Served as a baseline, capturing basic temporal dependencies.
- **GRU (Gated Recurrent Unit):** Introduced gating mechanisms for improved long-term memory retention.
- **LSTM (Long Short-Term Memory):** Provided superior performance on longer sequences due to its explicit memory cell and gating structure.

Training Strategy:

- Implemented early stopping based on validation MAE to prevent overfitting.
- Hyperparameters were tuned using grid search over lookback windows, hidden units, and dropout rates.

These predictions offer strategic foresight into evolving customer perceptions and serve as early-warning indicators for potential service bottlenecks.

Model Evaluation

Aspect Extraction Evaluation Against Gold Dataset To rigorously assess the reliability of our aspect extraction and sentiment classification pipeline, we conducted a targeted evaluation using a gold-standard dataset comprising 200 manually annotated Uber reviews. Each entry in this dataset was labeled with ground-truth aspect terms and corresponding sentiment polarities, enabling fine-grained performance measurement.

Evaluation Dataset:

- **Size:** 500 reviews
- **Annotations:** True aspect terms and associated sentiment labels (*positive*, *neutral*, *negative*)
- **Annotation Process:** Two independent annotators with domain familiarity labeled the dataset, achieving a Cohen's Kappa inter-annotator agreement score of 0.87, indicating high consistency.

Metrics and Analysis:

- **Aspect-Level Evaluation:** Precision, Recall, and F1-Score were computed to evaluate the ability of the system to accurately identify relevant aspect terms. BERT-based extraction achieved the highest F1-Score (0.81), outperforming both triplet (0.74) and contrastive attention-based (0.69) methods.
- **Sentiment Classification:** Confusion matrix analysis was performed on the predicted sentiment labels relative to the gold annotations. The classifier demonstrated strong performance in distinguishing *positive* and *negative* sentiments, with minor confusion in borderline *neutral* cases.

Identified Error Patterns:

- **False Positives:** Frequently involved extraction of generic or irrelevant nouns (e.g., *car*, *service*) not central to user sentiment.

Table 1: Comparison of evaluation metrics across extraction models.

Model	POS Tagging	BERT	CAT
Precision	0.366	0.757	0.791
Recall	0.486	0.875	0.422
F1 Score	0.417	0.812	0.532

- **False Negatives:** Often arose from missing implicit aspects or domain-specific phrases not well captured by token-level models (e.g., references like "waited forever" implying a delay-related aspect).

This evaluation underscored the importance of combining rule-based and deep learning methods to achieve balanced performance across varied linguistic contexts and sentiment nuances.

Forecasting Analysis Evaluation The evaluation metrics used for forecasting:

- **MAE (Mean Absolute Error):** Captured the average magnitude of prediction errors, providing a direct measure of model accuracy.
- **RMSE (Root Mean Squared Error):** Penalized larger errors more heavily, offering sensitivity to volatility in sentiment trends.
- **R^2 (Coefficient of Determination):** Quantified the proportion of variance in the observed data explained by the model, serving as a goodness-of-fit indicator.

Results: GRU and LSTM architectures consistently outperformed the baseline Vanilla RNN across all aspects. LSTM demonstrated marginally superior performance, particularly for aspects with high temporal variability such as *fare* and *support*. The average R^2 scores achieved were:

Forecasting Output:

- All models generated sentiment index forecasts over a 5-month prediction horizon for each high-frequency aspect.
- Forecasts indicated an upward trend in sentiment for service components such as *driver experience* and *app usability*.
- A slight downward trend was noted for aspects like *fare transparency*, potentially signaling areas of growing customer concern.

This evaluation not only validated the robustness of deep learning models in capturing sentiment dynamics but also provided actionable, time-aware insights for strategic planning.

Implementation

Triplet Extraction

We present a rule-based NLP pipeline to extract aspect-opinion-sentiment triplets `(Aspect, Opinion, Sentiment)` from Uber user reviews. Reviews are preprocessed using SpaCy for lowercasing, tokenization,

Table 2: Comparison of Time Series Models Across Evaluation Metrics

Model	Metric	Driver	Uber	Service	App
TCN	MSE	0.0469	0.0285	0.0296	0.0446
	RMSE	0.2167	0.1691	0.1722	0.2113
	MAE	0.1827	0.1359	0.1335	0.1761
	R^2	0.6833	0.5157	0.7576	0.6332
LSTM	MSE	0.0513	0.0273	0.0348	0.0438
	RMSE	0.2265	0.1653	0.1864	0.2093
	MAE	0.1848	0.1310	0.1469	0.1688
	R^2	0.6545	0.5372	0.7159	0.6403
GRU	MSE	0.0515	0.0270	0.0325	0.0444
	RMSE	0.2269	0.1643	0.1803	0.2107
	MAE	0.1855	0.1306	0.1426	0.1716
	R^2	0.6530	0.5428	0.7344	0.6355
RNN	MSE	0.0525	0.0269	0.0338	0.0447
	RMSE	0.2291	0.1641	0.1837	0.2114
	MAE	0.1878	0.1311	0.1454	0.1767
	R^2	0.6464	0.5442	0.7240	0.6332

POS tagging, and dependency parsing. Aspect terms are identified as non-stopword nouns, while opinion terms are adjectives or verbs linked to aspects via syntactic relations such as `amod`, `acomp`, and `xcomp`. Sentiment polarity is determined using the VADER lexicon, assigning labels based on compound scores. Triplets are constructed by pairing each aspect with its nearest opinion term within a 15-character window. Aspect normalization is applied using a curated mapping to unify similar terms. The final triplets are stored with full review context and exported for analysis. Evaluation is performed against human-annotated data using precision, recall, and F1-score. The pipeline is modular and interpretable, but limited in handling implicit sentiment and complex syntax. It serves as a lightweight baseline for trend and trajectory analysis in user feedback.

Aspect Extraction via BERT Token Classification

We frame aspect term extraction as a token-level classification task using BERT, applying the BIO tagging scheme (B-ASP, I-ASP, O) to identify aspect terms in Uber reviews [2]. Training data is generated using a seed list of domain-specific aspect terms, with BERT's WordPiece tokenizer and aligned BIO labels. Special tokens and attention masks are included as per BERT's input requirements.

The model architecture builds on `bert-base-uncased` with a linear classification head, optimized using cross-entropy loss on non-padding tokens and AdamW. Hyperparameters, including learning rate and batch size, are tuned via grid search. Performance is evaluated using precision, recall, and F1-score via `seqeval`.

During inference, BIO-tagged tokens are aggregated into aspect phrases and normalized using a manually curated dictionary to reduce vocabulary sparsity. The output comprises each review paired with a list of normalized aspect terms, serving as input for sentiment classification.

This method achieves robust, context-aware aspect identification, outperforming rule-based baselines but remains sensitive to data sparsity and domain-specific tuning requirements.

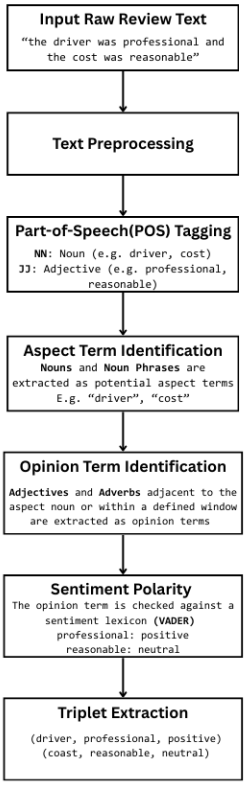

Figure 1: Baseline Architecture for What–How–Why Triplet Extraction

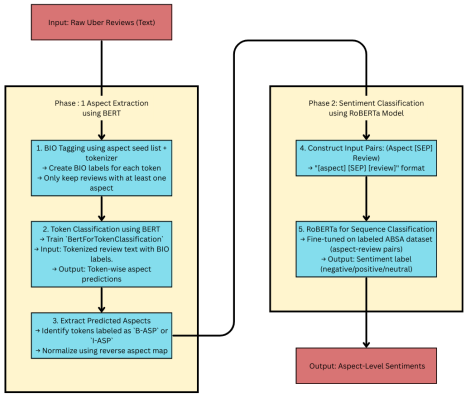

Figure 2: Baseline Architecture for BERT Aspect Classification + RoBERTa Sentiment Analysis

Aspect Extraction via Contrastive Attention Mechanism (CAt)

To extract aspect terms from Uber reviews with minimal supervision, we propose a *Contrastive Attention (CAt)* [7] mechanism that leverages semantic similarity to a curated set of canonical aspects (e.g., "driver", "fare"). Reviews are tokenized, lowercased, and embedded using a domain-specific Word2Vec model. A predefined aspect seed set $A = \{a_1, a_2, \ldots, a_k\} \subset R^d$ serves as the reference. For each token embedding $w \in S$, we compute an attention score using a Radial Basis Function (RBF) kernel:

$$\text{rbf}(w, a, \gamma) = \exp(-\gamma \|w - a\|^2)$$

$$\text{att}(w) = \frac{\sum_{a \in A} \text{rbf}(w, a, \gamma)}{\sum_{w' \in S} \sum_{a \in A} \text{rbf}(w', a, \gamma)}$$

Tokens with $\text{att}(w) > \delta$ (with $\delta = 0.1$) are selected as aspect terms and mapped to their nearest canonical forms. This approach aggregates attention over multiple aspects, improving recall and robustness to paraphrased inputs compared to conventional single-query attention.

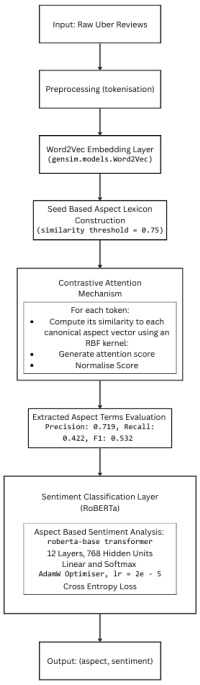

Figure 3: Contrastive Attention Mechanism

Sentiment Classification Using RoBERTa

We fine-tuned a pre-trained RoBERTa model for Aspect-Based Sentiment Analysis (ABSA) to classify sentiment as positive, negative, or neutral for each extracted aspect within review sentences.[5]

The dataset consisted of aspect-review pairs formatted as "`[aspect] [SEP] [review]`" to ensure the model focused on aspect-specific sentiment. It was split into training and validation sets in a 90:10 ratio to maintain label distribution.

We used the `roberta-base` model with a softmax classification head and fine-tuned all layers over 10 epochs. The model was trained using the cross-entropy loss function and optimized with the AdamW optimizer at a learning rate of 2×10^{-5}. Input sequences were capped at 128 tokens, and the batch size was set to 16.

The model achieved a validation accuracy of **87.4%**. Performance metrics for each sentiment class are presented in Table 3.

Sentiment	Precision	Recall	F1-score
Positive	0.89	0.86	0.87
Negative	0.85	0.88	0.86
Neutral	0.82	0.83	0.82

Table 3: Sentiment classification performance on the validation set

The trained model was deployed in inference mode to assign sentiment labels to each aspect-review pair, and predictions were stored for downstream analysis.

Sentiment Forecasting Using Temporal Convolution Network

In this study, we implemented sentiment forecasting for a specific aspect using a Temporal Convolutional Network (TCN) architecture. [3] The dataset consisted of timestamped sentiment labels associated with predicted aspects extracted from user reviews. Sentiment scores were mapped numerically and transformed into a sentiment index by subtracting the negative score from the positive score. To smooth fluctuations, a rolling average was applied to the sentiment index.

The processed time series was standardized using `StandardScaler`, and sliding windows of a fixed sequence length were generated to create supervised learning samples. The TCN model comprised stacked causal convolutional layers with increasing dilation rates and residual connections to capture temporal dependencies effectively. The network was trained to predict future sentiment values based on historical sequences.

The model's performance was evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2 score. Visualizations of actual vs. predicted sentiment and training loss curves were also generated to assess model behavior over time. The approach demonstrates the efficacy of TCNs in capturing temporal sentiment trends and can be applied for proactive decision-making in sentiment-sensitive domains.

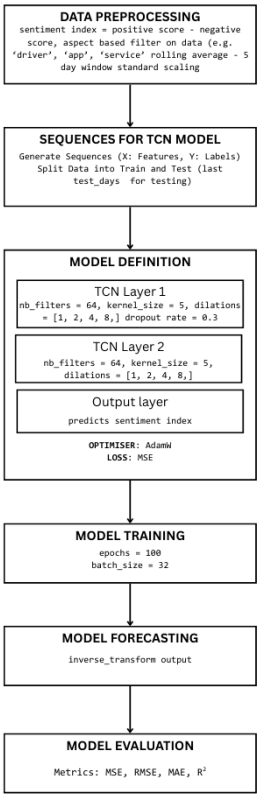

Figure 4: Temporal Convolution Network Mechanism

LSTM Model Architecture for Sentiment Forecasting

The forecasting model architecture is built using a sequential Long Short-Term Memory (LSTM) network designed to capture temporal patterns in the sentiment index time series for specific service aspects. The architecture begins with an LSTM layer comprising 128 units with `return_sequences=True`, allowing the retention of sequential dependencies across time steps. This is followed by a dropout layer with a 0.3 rate to prevent overfitting. A second LSTM layer with 64 units further refines the temporal features, feeding into a dense layer with 64 neurons and ReLU activation to introduce non-linearity. Finally, an output dense layer with a single neuron is used to predict the next sentiment index value. The model is compiled using the Adam optimizer, and the loss is minimized using Mean Squared Error (MSE), making it suitable for continuous-value forecasting tasks.

Model Architecture Flowchart:

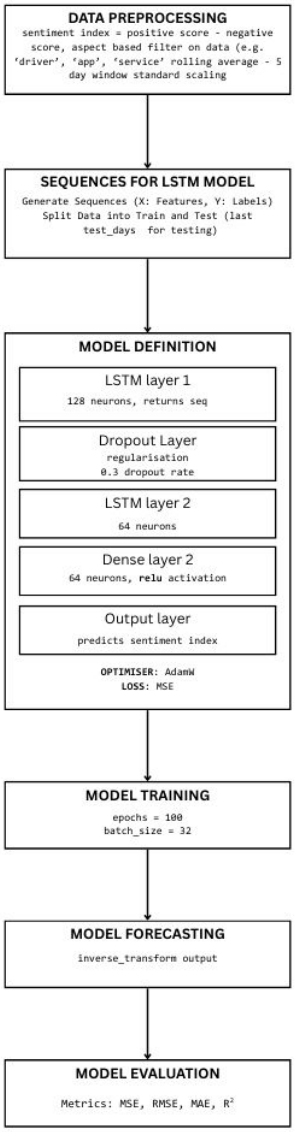

Figure 5: LSTM Model Architecture

Future Scope and Expansion Possibilities

1. Cross-Platform Sentiment Integration

Extend sentiment tracking beyond Uber to include platforms such as Lyft, Ola, and public transport reviews. Comparative analysis across platforms can help uncover industry-wide patterns in customer satisfaction and identify common service bottlenecks.

2. Multilingual Sentiment Analysis

Currently limited to English, the sentiment analysis pipeline can be extended to support multilingual datasets. This requires integrating multilingual transformer models such as XLM-RoBERTa or mBERT to handle diverse linguistic inputs and provide global sentiment insights.

3. Real-Time Sentiment Dashboard

Develop a real-time analytics dashboard for Uber's operational team. Streaming data from live customer reviews can be ingested using tools like Apache Kafka and processed via Spark NLP pipelines. Real-time sentiment forecasts can support dynamic resource allocation and responsive customer service.

4. Fine-Grained Sentiment Attribution

Move beyond categorical sentiment labels by incorporating sentiment intensity scoring. Tools such as VADER or regression-based BERT models can assign continuous sentiment values, enabling more nuanced analysis of sentiment fluctuations over time.

5. Causal Inference Models

Implement causal inference techniques such as Granger Causality or Bayesian Networks to explore not just correlations but potential causal relationships between operational metrics (e.g., wait time, fare changes, driver ratings) and sentiment dynamics.

6. Inclusion of Image/Voice-Based Reviews

With the evolution of user interfaces, Uber may soon integrate image or voice-based reviews. Incorporating multimodal models like CLIP or LLaVA will enable sentiment extraction from visual and auditory data, broadening the analytical capabilities of the system.

Conclusion

This project successfully demonstrates the power of integrating advanced Natural Language Processing (NLP) with deep learning-based time-series forecasting to decode and anticipate customer sentiment trends in Uber reviews. Through structured aspect extraction and sentiment classification, followed by robust predictive modeling, the system enables fine-grained, actionable insights.

These insights empower ride-sharing platforms to proactively enhance user experience and address service deficiencies. With vast potential for scalability across domains, languages, and modalities, this framework lays the foundation for a data-driven future in customer satisfaction management.

References

[1] Ain, Q. u.; Ali, M.; Riaz, A.; Noureen, A.; Kamran, M.; Hayat, B.; and Rehman, A. 2017. Sentiment analysis using deep learning techniques: a review. *International Journal of Advanced Computer Science and Applications*, 8(6).

[2] Ay, T. B. 2024. Aspect-Term Extraction and Aspect-Based Sentiment Analysis (ABSA) Using Fine-Tuned BERT Models. https://github.com/TarikBugraAy/BERT_Fine_Tune_For_ABSA.

[3] Chen, Y.; Kang, Y.; Chen, Y.; and Wang, Z. 2020. Probabilistic Forecasting with Temporal Convolutional Neural Network. *Neurocomputing*, 383: 287–297.

[4] Chen, Z.; Huang, H.; Liu, B.; Shi, X.; and Jin, H. 2021. Semantic and syntactic enhanced aspect sentiment triplet extraction. *arXiv preprint arXiv:2106.03315*.

[5] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

[6] Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; and Si, L. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8600–8607.

[7] Tulkens, S.; and Van Cranenburgh, A. 2020. Embarrassingly simple unsupervised aspect extraction. *arXiv preprint arXiv:2004.13580*.