Understanding Sentiment Lifecycles in Electronics Categories: Temporal Evolution of User Opinions on Amazon

Kumar Sanskar, Vedant Dave, Saarthak Jindal, Pragnya Dandavate
MSc. Data Science, Dhirubhai Ambani University, Gandhinagar, India
{202418027, 202418014, 202418046, 202418065}@daiict.ac.in

Problem Statement

In the digital era, user-generated reviews play a pivotal role in shaping consumer choices and influencing business strategies. These reviews are not only indicative of product performance, but also reflect evolving customer expectations, satisfaction levels, and pain points over time. While sentiment analysis has been widely adopted to extract insights from textual reviews, most existing methods treat sentiment as a static classification problem, often ignoring the rich temporal dynamics embedded in sequential user feedback. This results in a narrow understanding of customer sentiment that fails to capture trends, shifts, or anomalies that unfold throughout the life-cycle of a product.

This limitation is particularly pronounced in Electronics, a domain characterized by rapid innovation cycles, short product shelf lives, and constant user engagement. A laptop, for instance, may receive glowing reviews around its launch due to hype and new features, only to face disillusionment months later due to battery degradation or software issues. Similarly, sentiment may spike again due to firmware updates or decline as better alternatives emerge. These shifts are not isolated incidents tied to individual products—they often reflect broader, recurring patterns across product categories. Therefore, analyzing sentiment at the category level, rather than focusing on isolated products, offers a scalable and insightful approach to uncovering behavioral trends and life-cycle patterns.

The core objective of this study is to develop a comprehensive framework for modeling the temporal evolution of sentiment across product categories within the Electronics domain, using Amazon reviews as a primary data source. We aim to identify and interpret sentiment trends such as prelaunch excitement, post-purchase disillusionment, mid-life stability, and end-of-life dissatisfaction, and understand their implications on product development and customer satisfaction. By transforming raw review data into structured sentiment time series, we seek to build forecasting models capable of detecting inflection points and predicting future sentiment directions.

Our long-term goal is to bridge the gap between sentiment analysis and time series modeling in a way that supports strategic decision making, customer support planning, and market response forecasting. The outcome of this project will not only contribute novel insights to the fields of natural language processing and time series analysis but also offer practical tools for businesses to proactively manage their product lifecycles and consumer relationships.

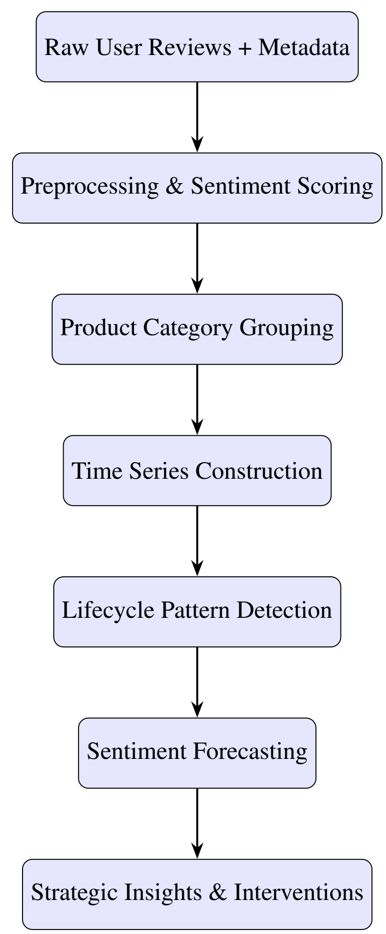

Figure 1: Problem Formulation Pipeline: Temporal Sentiment Evolution in Product Categories

Literature Review

Understanding the *temporal evolution of sentiment* in product reviews requires robust modeling techniques that capture long-term sentiment trends, contextual dependencies, and support large-scale data processing. Prior research has revealed important gaps that our work aims to bridge.

Sentiment Evolution and Narrative Arcs Traditional supervised sentiment analysis models suffer from data scarcity and limited domain generalization, especially for long-form narrative text. SentimentArcs[1] reveals that even state-of-the-art Transformer and deep learning models can be unstable when analyzing sentiment arcs over time. To address this, the authors propose a self-supervised ensemble-based framework for diachronic sentiment analysis, producing synthetic ground truth using multiple models to optimize corpus-model alignment. This method emphasizes the need for narrative-aware and context-sensitive sentiment analysis—an approach our study aligns with by tracking lifecycle sentiment trajectories within product categories.

Fine-Grained Sentiment Classification in Short Texts Agarwal et al.[2] investigate sentiment classification in microblogs like Twitter, moving beyond traditional binary polarity detection. Their study emphasizes the importance of 4-way classification (objective, neutral, positive, negative) and explores cascaded classification architectures. They further analyze the trade-offs between classifier confidence and prediction accuracy (F1 score), an often-overlooked metric in prior work. This granularity is especially useful in our task of temporal sentiment modeling where subtle shifts in consumer tone over time need to be identified and interpreted accurately.

Large-Scale Joint Topic and Sentiment Modeling Topic-sentiment-preference models such as TSPRA are often computationally limited due to the use of Gibbs sampling on large-scale corpora. Wang et al.[3] address this by introducing scalable alternatives—vTSPRA, svTSPRA, and ovTSPRA—based on variational inference and stochastic variational inference. These methods allow for real-time sentiment and preference tracking across millions of reviews. Our methodology draws inspiration from these scalable designs to model category-level sentiment evolution while preserving the semantic structure and latent temporal dynamics across product lifecycles.

Temporal Dynamics in Aspect-Level Sentiment Xia et al.[4] highlight that traditional sentiment analysis models often fail to account for the dynamic evolution of user opinions over time, particularly at the aspect level. To address this, they propose the Aspect-Based Sentiment Dynamic Prediction (ASDP) model, which captures sentiment progression using both uniform and non-uniform time intervals. This approach offers improved granularity in tracking shifts in user sentiment. The improved ASDP+ model further introduces a bin segmentation algorithm that identifies both gradual and sudden sentiment transitions, enhancing the system's ability to reflect meaningful behavioral changes over time. These temporal modeling insights directly inform our methodology for constructing category-level sentiment timelines and detecting lifecycle stages such as surges in post-launch excitement or declines during product obsolescence.

By combining narrative-aware sentiment modeling, fine-grained polarity classification, and large-scale scalable architectures, our work contributes a structured pipeline for tracking the lifecycle of sentiment in Amazon Electronics reviews—capturing transitions from product launch hype to post-purchase feedback and eventual decline.

Dataset Selection

Identification

The dataset used for this study contains over ˜17 million user reviews and product metadata from Amazon, specifically focusing on the **Electronics** category. The dataset provides rich temporal and textual information about customer feedback, enabling us to track sentiment changes over time. Key attributes include:

- **User Reviews Data:**
  - `rating`: Numerical rating (1.0 to 5.0) reflecting user satisfaction.
  - `text`: Review text containing user opinions and feedback.
  - `timestamp`: Time at which the review was posted (Unix format), essential for constructing time series.
  - `verified_purchase`: Indicates whether the user actually bought the product, helping to filter genuine reviews.
  - `helpful_vote`: Number of users who found the review helpful, serving as a proxy for review influence.
  - `asin`: Product ID, used to link reviews to metadata.
- **Product Metadata:**
  - `main_category`: Ensures that we are analyzing only **Electronics** products.
  - `average_rating`: Provides an aggregated sentiment score for each product over time.
  - `description` & `features`: Contain keywords that might correlate with sentiment shifts.
  - `price`: Changes in price could influence sentiment evolution.
  - `categories`: Helps in grouping products into meaningful subcategories (e.g., smartphones, wearables).

Justification

This dataset is particularly well-suited for our study due to the following reasons:

- **Temporal Evolution of Sentiment:**
  - The availability of `timestamps` in user reviews allows for time series analysis of sentiment.
  - By binning the data (weekly/monthly), we can analyze long-term sentiment trends.
- **Sentiment-Rich Text Data:**
  - The `text` field provides direct user opinions, making it a crucial feature for sentiment analysis.
  - The combination of `rating` and `text` sentiment scores allows for multi-modal sentiment evaluation.
- **Data Integrity & Quality Control:**
  - The `verified_purchase` attribute ensures that only genuine buyers are considered for sentiment modeling.
  - `helpful_vote` helps weigh reviews by importance.
- **Category-Specific Trends:**
  - The `main_category` and `categories` attributes ensure that our analysis remains focused on **Electronics**, preventing contamination from unrelated domains.

Preprocessing

To transform raw review data into meaningful sentiment time series aligned with our objective of tracking sentiment evolution across product categories, we employ a structured preprocessing pipeline. The key steps are:

- **Filtering by Category:** We isolate reviews specifically under the **Electronics** category and further refine the data by identifying consistent subcategories such as *headphones, laptops, wearables*, etc. This ensures category-level granularity and allows for comparative sentiment lifecycle analysis across different product types.
- **Time Binning:** To analyze sentiment evolution as a time series, we group reviews into uniform temporal bins — **weekly or monthly** — based on their timestamps. This step is critical for identifying lifecycle phases such as *pre-launch excitement, post-launch feedback surges, and eventual sentiment decline*. The bin size is tuned to balance granularity and stability.
- **Text Cleaning:** Review texts are normalized through common preprocessing steps, like **lowercasing**. These steps enhance tokenization quality and sentiment extraction by reducing noise and linguistic variation.
- **Tokenization:** Cleaned review texts are tokenized into structured units (words or subwords), which serve as inputs for downstream models such as **BERT-based sentiment scorers**. Tokenization is crucial for leveraging deep learning and attention-based architectures.
- **Sentiment Scoring:** Each review is assigned a sentiment score using robust sentiment classification techniques. We employ a **transformer-based** (e.g., *fine-tuned RoBERTa*) models to capture both surface-level polarity and contextual sentiment. These scores are aggregated at the time-bin level to form **category-wise sentiment timelines**.
- **Grouping and Aggregation:** Sentiment scores are grouped and averaged by product **category or sub-category** using product metadata. This aggregation enables us to create category-level sentiment arcs that reflect collective user opinion, rather than isolated product-specific signals.
- **Handling Data Sparsity and Imbalance:** A key challenge in category-level sentiment modeling is *temporal sparsity*, where certain products or time bins may lack sufficient review volume, especially in early or late lifecycle phases. To address this:
  - We apply **time-series smoothing** (e.g., rolling averages, exponential moving averages) to reduce volatility and highlight trends.

Methodology

The primary objective of this research is to perform **temporal based sentiment analysis** on a large-scale dataset of Amazon electronics product reviews. The analysis encompasses diverse product categories, including electronics, computers and wearables. By leveraging state-of-the-art natural language processing (NLP) and machine learning methodologies, this study aims to generate detailed, sentiment time-series. The resulting granular insights will enhance understanding of product lifecycle dynamics, facilitating early detection of critical shifts in consumer sentiment and enabling data-driven decisions for product management, marketing strategies, and customer engagement initiatives.

The methodological framework proposed consists of five distinct yet integrated phases:

- **Pre-training**: This phase involves fine-tuning a BERT-based language model on the complete Amazon review corpus. This step ensures the BERT encoder captures domain-specific linguistic patterns, terminology, and semantics, enhancing downstream task performance.
- **Temporal Aggregation and Data Storage**: The outputs, structured as tuples containing product identifiers, sentiment polarities, and temporal bins, are systematically stored in efficient analytical formats (Parquet files). This approach supports rapid querying and real-time analytical exploration.
- **Temporal Predictive Analytics**: Predictive modeling using Long Short-Term Memory (LSTM) networks, Deep LSTM and Transformers will enable predictions of future sentiment trends, thus providing actionable forecasts for strategic business planning.

Collectively, this methodological framework positions our research at the intersection of advanced NLP methodologies and practical, temporal sentiment analytics, aiming to significantly contribute to both theoretical advancements and real-world business intelligence applications.

Proposed Model Architecture

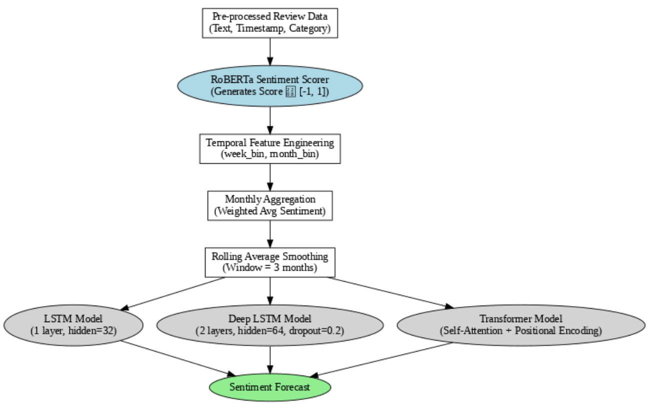

Figure 2: Model Architecture

Training Strategy

To ensure optimal model performance and accurate temporal sentiment analysis, the training strategy is methodically structured into distinct, sequential phases. Each phase is carefully designed to leverage progressive fine-tuning, iterative evaluation, and rigorous hyperparameter optimization.

Phase I: Domain-Specific Sentiment Scoring

In the first phase, we apply a pretrained sentiment analysis model to assign polarity scores to a large corpus of Amazon product reviews. Rather than training or fine-tuning a model on domain-specific data, we utilize the publicly available model, which has been trained on social media data but performs robustly across informal text.

The input data consists of structured reviews with separate title and body fields. For each record, we concatenate the title and text into a unified input string, tokenize it to a maximum length of 128 tokens, and run inference in mini-batches of 64. Inference is conducted on the GPU where available, with automatic mixed precision for improved throughput.

The model predicts one of three sentiment classes: negative, neutral, or positive. These are mapped to numerical values of -1, 0, and 1, respectively, and recorded in a new column. Processing is done in streaming chunks of 200,000 rows to minimize memory usage, enabling scalable inference on large datasets.

Phase II: Temporal Analysis of Subcategory-Level Sentiment Trends

In the second phase, we perform a temporal analysis of sentiment trends for selected product subcategories using the sentiment scores computed in Phase I. This analysis provides insights into how customer sentiment evolves over the product lifecycle.

We begin by filtering the dataset to include only reviews from three key subcategories: *Headphones, Earbuds & Accessories*, *Television & Video*, and *Car & Vehicle Electronics*. The filtered dataset is saved in Parquet format for efficient access.

Each review is timestamped and associated with its product's launch date. We compute the number of days since launch for each review and exclude any reviews submitted before the product release. Using this time delta, we bin the data into weekly and monthly intervals.

We first compute the average sentiment per week (weighted by review count) and then aggregate these weekly values into monthly averages to stabilize short-term fluctuations. To further enhance interpretability, we apply a weighted rolling average (smoothing window of 3 months) to the monthly sentiment scores.

The final output consists of smoothed sentiment trajectories plotted over time, with months since launch on the x-axis and sentiment score (ranging from -1 to 1) on the y-axis. These plots reveal the long-term sentiment dynamics of each product category and serve as a foundation for downstream tasks like product performance modeling, sentiment forecasting, or anomaly detection.

Phase III: Realistic Time-Series Forecasting

In this phase, we aimed to simulate real-world forecasting conditions by training models solely on historical sentiment data and evaluating their predictions on unseen future data. The three models used were:

- **Standard LSTM Model**
- **Deep LSTM Model (with Layer Normalization)**
- **Transformer-based Regressor**

Each model was evaluated using the best hyperparameters obtained from Phase 1 tuning. The core objective was to predict future sentiment trends for different subcategories based on smoothed sentiment sequences.

Data Preparation and Sequence Splitting For each subcategory, the sentiment series was normalized using MinMaxScaler. A 70-30 split was used for training and testing. Sliding window sequences were created with a tuned sequence length for all models.

Model Architectures and Training

(a) Standard LSTM: A basic LSTM with tuned hidden size h, number of layers l, dropout d, and weight decay λ was trained. The final hidden state was passed through a dense layer to predict the sentiment. Early stopping with patience p was used based on validation loss.

(b) Deep LSTM: An enhanced LSTM model with layer normalization after the LSTM output. This model used the same hyperparameters as above, but was found to generalize slightly better due to improved internal gradient flow and regularization.

(c) Transformers: A custom transformer encoder with:

- Input projection: linear mapping to model
- Encoder: 2 layers, 4 heads, feedforward dimension 128
- Output: predicted sentiment from final encoder state

The transformer was trained using the Adam optimizer and MSE loss with early stopping.

Evaluation All models were evaluated on the same test split using two metrics:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- The Deep LSTM generally outperformed the vanilla LSTM on longer sequences and noisier subcategories.
- The Transformer model showed competitive performance despite fewer iterations, particularly in subcategories with clearer seasonality.
- Overfitting was controlled via dropout and early stopping, although smaller datasets limited generalization.

This phase demonstrated the capability of modern sequence models to capture temporal sentiment dynamics under realistic constraints.

Algorithm: Sentiment Analysis and Forecasting Pipeline

- **Phase I: Domain-Specific Sentiment Scoring**
  - Load pretrained sentiment analysis model
  - For each product review:
    * Concatenate title and body into a single text string
    * Tokenize the text with a maximum length of 128 tokens
    * Run inference on the model in mini-batches of 64 (on GPU with mixed precision)
    * Map the sentiment classes (negative: -1, neutral: 0, positive: 1) to numeric values
    * Store the sentiment score in a new column
  - Process data in chunks of 200,000 rows for memory efficiency
- **Phase II: Temporal Analysis of Subcategory-Level Sentiment Trends**
  - Filter reviews for the key subcategories: Headphones, Earbuds & Accessories, Television & Video, Car & Vehicle Electronics
  - Save the filtered dataset in Parquet format
  - For each review:
    * Compute the number of days since product launch
    * Exclude reviews submitted before product release
    * Bin reviews into weekly and monthly intervals based on time delta
  - Compute the average sentiment per week (weighted by review count)
  - Aggregate weekly sentiment values into monthly averages
  - Apply a weighted rolling average (3 months smoothing window) to monthly sentiment scores
  - Plot smoothed sentiment trajectories with months since launch on the x-axis and sentiment score on the y-axis
- **Phase III: Realistic Time-Series Forecasting**
  - Normalize sentiment series using MinMaxScaler for each subcategory
  - Split data into training (70%) and testing (30%) sets
  - Create sliding window sequences with tuned sequence length for all models
- **Model Architectures and Training**
  - Train Standard LSTM:
    * Use hidden size h, number of layers l, dropout d, and weight decay λ
    * Use early stopping with patience p based on validation loss
    * Final hidden state passed through dense layer to predict sentiment
  - Train Deep LSTM (with layer normalization):
    * Use the same hyperparameters as Standard LSTM
    * Improved gradient flow and regularization due to layer normalization
  - Train Transformer-based Regressor:
    * Input projection: linear mapping
    * Encoder: 2 layers, 4 heads, feedforward dimension 128
    * Output: predicted sentiment from final encoder state
    * Train with Adam optimizer and MSE loss with early stopping
- **Evaluation**

Experiment Setup

Data Selection and Filtering

The dataset comprises Amazon product reviews across various categories. To focus on electronics with substantial volume and diverse feedback, we selected the different subcategories.

Using the `Polars` library for efficient data processing, only reviews from these subcategories were retained. The filtered dataset was stored in `Parquet` format for optimized I/O performance.

Date Parsing and Cleaning

Each review includes:

- `timestamp`: Date of the review
- `Date First Available`: Product launch date

Both fields were parsed into ISO 8601 datetime format. Reviews with missing or malformed dates were removed. Additionally, any review dated prior to the product launch was excluded to maintain consistency.

Temporal Feature Engineering

To analyze sentiment evolution, the following features were engineered:

- **Days since launch:** $\texttt{review\_date} - \texttt{product\_launch\_date}$
- **Week bin:** $\lfloor \frac{\texttt{days\_since\_launch}}{7} \rfloor$
- **Month bin:** $\lfloor \frac{\texttt{days\_since\_launch}}{30} \rfloor$

These features enable grouping sentiment by consistent temporal intervals.

Sentiment Score Aggregation: Weekly Analysis

Each review had an associated sentiment score in the range $[-1, 1]$. For each week and subcategory, we computed:

- **Weekly average sentiment**
- **Weekly review count**

This step reduces data noise and highlights temporal sentiment patterns.

Weighted Monthly Sentiment Computation

Monthly sentiment scores were calculated by weighting weekly scores by their respective review counts. Let s_i be the sentiment score for week i and n_i its review count. The monthly sentiment is given by:

$$\texttt{monthly\_sentiment} = \frac{\sum_i n_i \cdot s_i}{\sum_i n_i}$$

Gap Filling and Temporal Continuity

To preserve continuity in the time series:

1. A complete range of month bins was generated.
2. Missing months were filled with zero-valued entries for sentiment and review count.

This regularization ensures consistent input for smoothing and modeling.

Trend Smoothing with Rolling Average

A 3-month weighted rolling average was applied to reduce short-term fluctuations:

$$\texttt{smoothed}(t) = \frac{\sum_{i=t-2}^{t} n_i \cdot s_i}{\sum_{i=t-2}^{t} n_i}$$

where n_i and s_i are review count and sentiment for month i.

LSTM-Based Time Series Forecasting

We implemented a single-layer LSTM model with:

- Hidden size: Tuned via grid search
- Output: Linear layer predicting the next month's sentiment

Inputs were sliding windows of fixed length from smoothed sentiment values. Data was normalized to $[0, 1]$ using `MinMaxScaler`.

Training Setup and Early Stopping

- Train/test split: 70% / 30%
- Loss function: Mean Squared Error (MSE)
- Optimizer: Adam, learning rate $= 0.01$
- Early stopping: Based on validation loss, patience = 5 epochs

Hyperparameter Tuning Grid search was performed over:

- `sequence_length`: $\{3, 6\}$
- `hidden_size`: $\{32, 64\}$
- `dropout`: $\{0.0, 0.2\}$
- `weight_decay`: $\{0.0, 0.001\}$
- `num_layers`: $\{1, 2\}$

Deep LSTM-Based Time Series Forecasting

We implemented a deep LSTM model with:

- Multi-layer LSTM: 1 or 2 stacked layers
- Hidden size: Tuned via grid search
- Layer Normalization: Applied to the final LSTM output
- Output: Fully connected layer predicting the next time step's sentiment

Inputs were sliding windows of fixed length constructed from smoothed sentiment values. All values were normalized to $[0, 1]$ using `MinMaxScaler`.

Training Setup and Early Stopping

- Train/test split: 70% / 30%
- Loss function: Mean Squared Error (MSE)
- Optimizer: Adam with learning rate $= 0.01$
- Weight decay: Applied as L2 regularization
- Early stopping: Based on validation loss, using a configurable patience threshold

Hyperparameter Tuning Grid search was conducted over the following configuration space:

- `sequence_length`: $\{5\}$
- `hidden_size`: $\{32, 64\}$
- `dropout`: $\{0.0, 0.2\}$
- `weight_decay`: $\{0.0, 1\text{e-}5\}$
- `patience`: $\{5, 10\}$
- `num_layers`: $\{1, 2\}$

Dropout was only applied when the number of layers was greater than one. Each configuration was evaluated using MAE and MSE on the test set, with best model weights selected via early stopping.

Transformer-Based Time Series Forecasting

We implemented a transformer-based regressor to predict the next time step's sentiment score using:

- Input projection: Linear layer mapping from input dimension to model dimension
- Transformer encoder: Stack of self-attention layers with feedforward blocks
- Output: Final time step representation passed through a linear layer for regression

The model learns temporal dependencies from sliding windows of normalized smoothed sentiment values. All inputs were scaled to $[0, 1]$ using `MinMaxScaler`.

Training Setup and Early Stopping

- Train/test split: 70% / 30%
- Loss function: Mean Squared Error (MSE)
- Optimizer: Adam with learning rate $= 0.001$
- Early stopping: Based on validation loss with patience = 10 epochs

Model Configuration The transformer architecture was configured as follows:

- `input_dim`: 1
- `d_model`: 64 (embedding size)
- `nhead`: 4 (number of attention heads)
- `num_layers`: 2 (transformer encoder layers)
- `dim_feedforward`: 128 (inner layer size of feedforward networks)
- `dropout`: 0.1
- `sequence_length`: 5

The model was trained for each subcategory independently, with evaluation using MAE and MSE on the test set. Plots of actual vs. predicted sentiment were generated to visualize forecasting performance.

Experiment Results

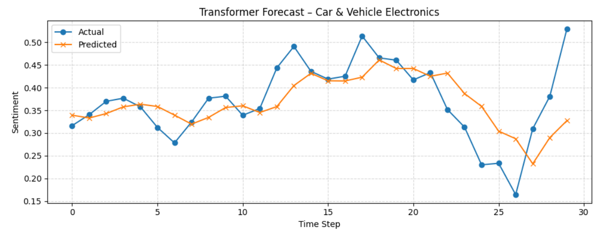

Figure 3: Transformers Forecast- Car & Vehicle Electronics

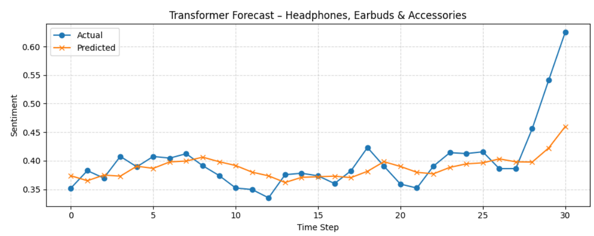

Figure 4: Transformers Forecast- Headphones, Earbuds & Accessories

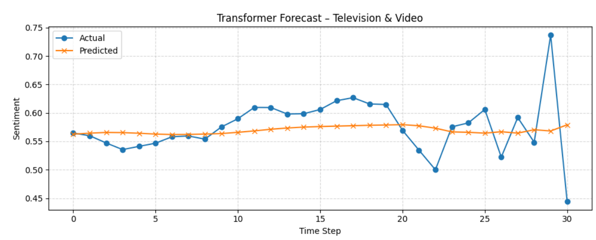

Figure 5: Transformers Forecast- Television & Video

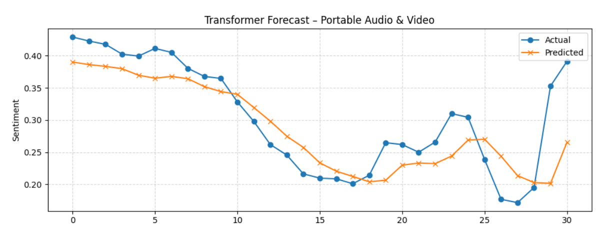

Figure 6: Transformers Forecast- Portable Audio & Video

Table 1: Validation and Test Performance of Different Models Across Subcategories

Subcategory	Baseline		Transformer Test MSE
	LSTM Test Loss	Deep LSTM Test Loss	(MSE)
Car & Vehicle Electronics	0.0364	0.0369	0.0046
Headphones, Earbuds & Accessories	0.0223	0.0198	0.0019
Portable Audio & Video	0.0677	0.1796	0.0023
Television & Video	0.0591	0.0393	0.0024

Conclusion

The study focuses on the limitation of traditional static sentiment analysis, which fails to capture the temporal evolution of sentiment throughout a product's lifecycle. This is particularly relevant in the Electronics domain, where user opinions on products like laptops change over time due to factors like hype, post-purchase experience, and updates or alternatives. The main goal was to build a comprehensive framework to model this temporal sentiment evolution across product categories using Amazon reviews and to develop forecasting models capable of predicting future sentiment trends and identifying inflection points. The methodology involved processing raw Amazon review data by filtering for Electronics categories and consistent subcategories, binning reviews into weekly or monthly intervals based on timestamps, and cleaning review text. Transformer-based models were used to assign sentiment scores to each review, which were then aggregated at the time-bin and category level to form sentiment timelines. Time-series smoothing techniques, such as rolling averages, were applied to handle data sparsity and highlight trends. For forecasting, the research implemented and evaluated Standard LSTM, Deep LSTM (with Layer Normalization), and Transformer-based regressors. These models were trained on historical sentiment data to predict future sentiment trends. Evaluation using MAE and MSE showed that the Deep LSTM generally outperformed the vanilla LSTM, and the Transformer model also demonstrated competitive performance. The outcome of this research provides valuable insights into product lifecycle dynamics by identifying sentiment trends at different stages. This framework offers practical tools for businesses to make data-driven decisions regarding product management, marketing strategies, and customer engagement by understanding and predicting how consumer sentiment evolves over time. The work bridges the gap between sentiment analysis and time series modeling, offering actionable intelligence for proactively managing product lifecycles.

References

[1] Chun, Jon. *SentimentArcs: A Novel Method for Self-Supervised Sentiment Analysis of Time Series Shows SOTA Transformers Can Struggle Finding Narrative Arcs*. arXiv preprint arXiv:2110.09454, 2021.
`https://arxiv.org/abs/2110.09454`

[2] Agarwal, Apoorv and Sabharwal, Jasneet Singh. *End-to-End Sentiment Analysis of Twitter Data*. Proceedings of the Workshop on Information Extraction and Entity Analytics on Social Media Data, 2012, pp. 39–44.
`https://aclanthology.org/W12-5504.pdf`

[3] Wang, Hao; Lu, Yulei; Zhai, ChengXiang; Han, Jiawei. *Large-Scale Joint Topic, Sentiment & User Preference Analysis for Online Reviews*. arXiv preprint arXiv:1901.04993, 2019.
`https://arxiv.org/abs/1901.04993`

[4] Xia, Peike; Jiang, Wenjun; Wu, Jie; Xiao, Surong; Wang, Guojun. *Exploiting Temporal Dynamics in Product Reviews for Dynamic Sentiment Prediction at the Aspect Level*. ACM Transactions on Knowledge Discovery from Data, Vol. 15, No. 4, Article 68, April 2021, pp. 1–29.
`https://doi.org/10.1145/3441451`

Limitations

While the models perform well across product subcategories, several limitations remain. First, the approach is limited to textual sentiment and does not incorporate structured metadata or images. Second, sentiment trends toward later months suffer from data sparsity, where each review significantly impacts the trend due to low review volume. Third, the Transformer model was trained on short sequences because of memory constraints, limiting its ability to capture long-range dependencies. Finally, the use of smoothed sentiment reduces noise but may obscure short-term fluctuations that are relevant for real-time forecasting.

Future Scope

Future work can address these limitations by incorporating additional signals like product ratings, price variations, and image features into the forecasting pipeline. Extending the analysis to include cross-product interactions or hierarchical relationships (e.g., brand-level trends) may uncover richer patterns. Transformer architectures could be further optimized using efficient attention mechanisms or pretraining on a domain-specific corpus to enhance generalization. Moreover, integrating external factors like holidays, sales events, or supply chain delays could improve the temporal accuracy of sentiment-driven demand forecasts. Finally, deployment of the models in a live setting could validate their predictive utility through A/B testing and continuous feedback.