Analysis of sentiment evolution in e-Commerce reviews

Misty Roy 202418033
Jalak Vyas 202418062
Kaustav Mitra 202418023

Dhirubhai Ambani University
Gandhinagar, Gujarat

Abstract

In today's world, e-Commerce plays an important role for customers who do not need to step out of their house to purchase something of their choice. With millions of online transactions occurring daily, customer reviews have become a crucial source of feedback, influencing purchasing decisions and brand reputations. In this context, Sentiment analysis, also known as opinion mining, is the process of analyzing customer reviews, to determine the sentiment or emotional tone expressed. Examining the temporal evolution of sentiments can help understand trends, seasonal patterns, and shifts in customer perceptions, providing valuable insights for companies to refine their strategies and enhance customer satisfaction in an increasingly competitive digital marketplace.

Introduction

E-Commerce platforms such as Amazon receive reviews from millions of customers around the world , providing valuable information on product quality, customer satisfaction, and brand satisfaction. These reviews reflect real-time customer experience and play a crucial role in influencing potential buyers. By analyzing the sentiment of these reviews over time, patterns in customer perception can be identified, along with shifts in sentiment due to product updates or policy changes. This temporal sentiment analysis helps businesses optimize their offerings, improve customer engagement, and adapt to evolving market and consumer demands.

Problem Definition

In the highly competitive e-commerce environment, understanding customer feedback is vital for maintaining product quality, enhancing user satisfaction, and reducing churn. Platforms like Amazon host millions of product reviews, each containing valuable information about customer experiences. However, most traditional sentiment analysis methods reduce this rich feedback to a single sentiment label per review. This oversimplification overlooks the nuanced opinions customers express about specific product attributes such as delivery, packaging, performance, pricing, and usability. To address this gap, our project centers on developing a robust Aspect-Based Sentiment Analysis (ABSA) system tailored for e-commerce reviews. Instead of treating reviews as monolithic text blocks, ABSA breaks them down into individual aspects and determines the sentiment expressed toward each. This fine-grained approach provides actionable insights into exactly what customers like or dislike, helping businesses make targeted improvements to specific product features. However, implementing ABSA at scale presents several challenges. E-commerce reviews are often unstructured, noisy, and domain-specific, making manual annotation of aspect-sentiment pairs both time-consuming and costly. These challenges underscore the importance of automation—not just in sentiment classification, but also in the creation of labeled datasets required to train such systems. Our problem statement, therefore, goes beyond standard sentiment analysis. It focuses on automating the process of extracting and labeling aspect-sentiment pairs from raw, unstructured customer reviews. To this end, we propose a semi-supervised learning framework integrated with active learning. This enables the model to selectively query the most informative and uncertain examples for manual labeling, significantly reducing annotation effort while improving learning efficiency. For aspect extraction, we employ SpaCy, a fast and customizable NLP library suited for domain-specific pattern recognition. For sentiment classification, we use a BERT-based model with multi-head attention, capable of capturing the nuanced context around each aspect. The active learning component strategically guides annotation by identifying samples where the model is least confident, ensuring that human labeling efforts are used where they matter most. By automating both the extraction and efficient labeling of aspect-level sentiment data, this approach delivers a scalable, real-world ABSA pipeline—bridging the gap between raw customer feedback and actionable business intelligence. To make the system forward-looking, we also incorporate time series forecasting on historical aspect-level sentiment data. This allows us to predict future sentiment trends, enabling businesses to proactively address potential issues and adapt to shifting customer preferences.

Problem Formulation

Data Setup

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ be the dataset of customer reviews, where:

- $x_i \in \mathcal{X}$: raw text review

- $y_i = \{(a_{ij}, s_{ij})\}_{j=1}^{m_i}$: set of annotated aspect-sentiment pairs in x_i, where:
  - a_{ij}: aspect term
  - $s_{ij} \in \{\text{negative}, \text{neutral}, \text{positive}\}$: sentiment label

Initially, only a small subset $D_L \subset D$ is labeled, while the rest $D_U = D \setminus D_L$ remains unlabeled.

Aspect Extraction Using spaCy

Given input text x, the aspect extractor f_a identifies spans $\hat{A}(x) = \{\hat{a}_1, \hat{a}_2, ..., \hat{a}_k\}$ with start-end character offsets:

$$\hat{a}_j = (\text{start}_j, \text{end}_j)$$

Sentiment Classification Using BERT

Given extracted aspect $\hat{a}_j \in x$, a classifier $f_s(x, \hat{a}_j) \to \hat{s}_j$ predicts the sentiment label.

Loss Function (Cross-Entropy):

$$\mathcal{L}_{\text{sentiment}} = -\sum_{i=1}^{N}\sum_{j=1}^{m_i} \log P(s_{ij}|x_i, a_{ij})$$

Note: Sentiment classification is only evaluated if the aspect is correctly extracted:

$$\text{If } \hat{a}_j \neq a_{ij}, \text{ then } \hat{s}_j \text{ is not evaluated.}$$

Active Learning Loop

Algorithm 1 Iterative Active Learning with Aspect-Based Sentiment Classification

Input: Unlabeled dataset $\mathcal{U}$, Labeled set $\mathcal{L} = \emptyset$, Budget B, Model $\mathcal{M}$ (initialized with zero-shot weights)
while $|\mathcal{L}| < B$ **do**
  Use $\mathcal{M}$ to predict sentiment probabilities $\hat{y}$ for all $x \in \mathcal{U}$
  for each $x \in \mathcal{U}$ **do**
    Compute entropy $\mathcal{H}(x) = -\sum_{i=1}^{3} \hat{y}_i \log(\hat{y}_i)$
    Compute margin $\mathcal{M}(x) = \hat{y}_{[1]} - \hat{y}_{[2]}$
    Compute attention dispersion score $\mathcal{A}(x)$ from BERT attention maps
    Combine scores to compute uncertainty score $\mathcal{U}(x)$
  end for
  Select top-k most uncertain samples: $\mathcal{S} \subset \mathcal{U}$ where $|\mathcal{S}| = k$
  Query human annotator to label $\mathcal{S}$ with true sentiment labels y
  Update labeled dataset: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x, y) \mid x \in \mathcal{S}\}$
  Remove annotated samples from pool: $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{S}$
  Fine-tune model $\mathcal{M}$ on updated labeled set $\mathcal{L}$ using cross-entropy loss
end while
Return: Fine-tuned model $\mathcal{M}$

Time Series Forecasting with LSTM and Smooth Quantile Regression (Rolling Forecast)

Let $t \in \{1, 2, ..., T\}$ represent monthly time steps for a selected aspect a^*, and let:

$r_t^{(s)}$ is the sentiment ratio at time t for $s \in \{-1, 0, 1\}$

We adopt a **rolling forecast strategy**, where predictions for each future time step are made using a sliding window of the past w observations. For each sentiment class s, define the input sequence at time t as:

$$X_t = \{r_{t-w+1}^{(s)}, r_{t-w+2}^{(s)}, ..., r_t^{(s)}\}$$

LSTM Model:

$$h_t = \text{LSTM}(X_t, h_{t-1})$$
$$\hat{r}_{t+1}^{(s,q)} = W_q h_t + b_q, \quad q \in \{0.025, 0.5, 0.975\}$$

Smooth Quantile Loss: To provide robustness against outliers while maintaining asymmetric quantile regression properties, we combine the quantile loss with the Huber loss:

$$\mathcal{L}_{\text{smooth}} = \mathcal{L}_{\text{quantile}} + \lambda \cdot \mathcal{L}_{\text{huber}}$$

Where:

- Quantile loss for quantile $q \in (0, 1)$:

$$\mathcal{L}_{\text{quantile}} = \sum_t \max\left(q(r_t - \hat{r}_t), (q-1)(r_t - \hat{r}_t)\right)$$

- Huber loss with threshold $\delta > 0$:

$$\mathcal{L}_{\text{huber}} = \sum_t \begin{cases} \frac{1}{2}(r_t - \hat{r}_t)^2 & \text{if } |r_t - \hat{r}_t| \leq \delta \\ \delta(|r_t - \hat{r}_t| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

- λ: regularization coefficient to control the influence of the Huber loss.

This smoothed loss function improves robustness to outliers and stabilizes learning, particularly when the target sentiment ratios are noisy or sparse.

Dataset Selection

Primary dataset - Amazon reviews dataset
Secondary dataset - Laptop reviews dataset

Description

User Reviews:
rating: Rating of the product (from 1.0 to 5.0).
title: Title of the user review.
text: Text body of the user review.
images: Images of that product.
asin: ID of the product.
parent_asin: Parent ID of the product.
user_id: ID of the reviewer.
timestamp: Time of the review.
verified_purchase: User purchase verification.

helpful_vote: Helpful votes of the review.

Item Metadata:
main_category: Main category of the product.
title: Name of the product.
average_rating: average rating of the product.
rating_number: Number of ratings for the product.
features: Bullet-point format features of the product.
description: Description of the product.
price: Price in US dollars.
images: Images of the product.
videos: Videos of the product including title and URL.
store: Store name of the product.
categories: Hierarchical categories of the product.
details: Product details, including materials, brand, sizes.
parent_asin: Parent ID of the product.
bought_together: Recommended bundles.

Justification

The Amazon Reviews dataset is one of the most comprehensive collections available, featuring a vast amount of review data coupled with detailed product metadata. Notably, every review comes with a timestamp, making it possible to break down and analyze data year by year. This is especially valuable when incorporating time series analysis to understand how consumer sentiment evolves over time.

Pre-Processing

1. Lowercasing Converts all characters in the text to lowercase to ensure consistency and avoid treating the same words differently due to casing.

2. Removing Punctuation Eliminates punctuation marks which usually do not contribute meaningfully to text analysis and can be considered noise.

3. Tokenization Splits the text into smaller units (tokens), typically words or subwords, to prepare it for further processing.

4. Removing Numbers Filters out numeric values if they do not add contextual value, helping to simplify the dataset.

5. Removing Extra Whitespace Cleans up unnecessary spaces, tabs, or line breaks to standardize text formatting.

Scope for Augmentation

Since the dataset is large, augmentation may not be necessary for volume, but it can enhance generalization. In NLP, augmentation can simulate linguistic variability and reduce overfitting to specific phrases or token patterns.It helps models become less sensitive to surface-level changes and more focused on semantic meaning. Especially in sentiment analysis and aspect extraction, it can expose the model to varied phrasings of opinions and aspect mentions. Thus, even when data volume is sufficient, augmentation serves as a regularization strategy to improve performance on unseen data. *Although we haven't done any data augmentation, it can be incorporated if needed to improve generalization.*

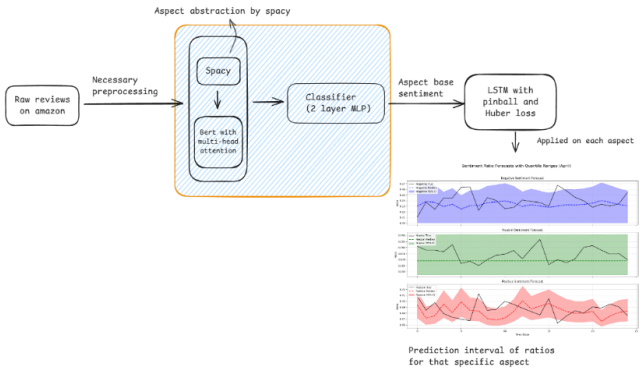

Figure 1: End to end pipeline

Model architecture

1. Aspect Extraction using spaCy

- spaCy language model initialization: The core architecture of this pipeline begins with a pre-trained language model (en_core_web_sm) from spaCy.
  - Tokenization: Breaks text into tokens (words, punctuation, etc.).
  - Tagging: Parts of speech tagging (e.g., nouns, verbs).
  - Parser: Syntactic dependency parsing to understand sentence structure.
  - NER (Named Entity Recognition): Detects and classifies entities such as names, dates, and in this case, aspects.
- Entity Recognition Pipeline
  - The NER component is designed to detect entities and categorize them into predefined classes. In this architecture, the ner component is responsible for identifying product aspects in the text.
  - Adding custom entity labels: The custom label ASPECT is added to the NER component. This label represents the entities the model will classify during training, specifically for product aspects in customer reviews.

2. BERT Model with MultiheadAttention

- Pretrained BERT
  - Pretrained BERT Model Initialization: A pretrained BERT model (bert-base-uncased) is used as the base model. It is fine-tuned for the Aspect-Based Sentiment Analysis (ABSA) task. The core functionality of BERT lies in its ability to capture the contextual relationships between words by processing the input text in both directions.
  - Bert Embedding: BERT uses its built-in tokenizer from the transformers library to process the input text. This tokenizer splits the sentence into smaller units called tokens and converts them into numerical representations. The resulting sequence of tokens is fixed-length and fed into the BERT model for further processing.

- BERT Pipeline
  - Multihead Attention Layer: The multihead attention mechanism is introduced to allow the model to focus on specific parts of the input sequence, especially the aspect tokens. This layer computes attention scores to decide which tokens should be given more importance during sentiment prediction. The attention mechanism ensures that the model attends more effectively to the aspect while still considering the entire context of the sentence.
  - Layer Normalization: After the attention layer, a layer normalization step is applied to the output. This helps in stabilizing the training process, ensuring smoother convergence and improving model performance by reducing internal covariate shift.
  - Classifier Layer: The final layer consists of a multilayer perceptron (MLP) that performs sentiment classification:
    * Linear Layer: The output of the attention layer is passed through a linear transformation.
    * ReLU Activation: The linear output is then passed through a ReLU activation function to introduce nonlinearity.
    * Dropout Layer: A dropout layer is used for regularization, preventing the model from overfitting.
    * Output Layer: The final layer produces the sentiment predictions, which are classified into three categories: Positive, Negative, or Neutral.

3. Active Learning Framework

The active learning framework is designed to reduce manual annotation costs while iteratively improving the performance of a BERT-based sentiment classifier. The process is structured as follows:

- Initial Setup:
  - Start with a small basically test dataset for manual annotation.
  - Run inference using the spaCy and BERT model on the training dataset.
- Low confidence sampling:
  - Use confidence scores to identify uncertain samples.
  - Select a batch of highly uncertain examples for human labeling.
- Manual Annotation:
  - Human annotators review and label the selected uncertain examples, for both aspect and sentiments.
  - These new labels are used for finetuning.
- Model Retraining:
  - Retrain spaCy and BERT on the rest of the training dataset.
  - Evaluate the model using test dataset.
  - Use F1 score to monitor performance improvements.
- Iteration and Convergence:
  - Repeat the sampling, annotation, and retraining process until:
    * The model reaches a target performance.
    * Or the annotation budget is exhausted.

4. LSTM with Quantile regression

- LSTM Network:
  - A deep LSTM with 5 layers and a hidden size of 256.
  - Capable of modeling long-term dependencies and temporal dynamics in the input sequence.
  - Outputs the final hidden state summarizing the sequential information.
- Dropout Layer:
  - Introduced after the LSTM to reduce overfitting by randomly zeroing out some neurons during training.
- Fully Connected Layer:
  - Maps the LSTM output to a 3-dimensional vector.
  - Each output corresponds to a specific quantile:
    * 2.5% quantile (Q025)
    * 50% quantile (median, Q50)
    * 97.5% quantile (Q975)

Experimental Setup and Result

Data Collection and Sampling

We utilized the Cell Phones and Accessories subset of the Amazon Reviews '23 dataset. Reviews spanning from 2011 to 2020 were selected for training and testing purposes. From an initial pool of 60,000 training samples, data was selected in batches using review ratings as pseudo-labels. Specifically, ratings of 1–2 were treated as negative, 3 as neutral, and 4–5 as positive. This labeling ensured a balanced distribution across sentiment classes in the training set. The test set comprised 1,800 samples from the same time period and was sampled to match the sentiment distribution of the training set. However, due to computational constraints, the full dataset could not be used for model training. Therefore, the training set was downsampled to 9,000 samples, and the test set was reduced to 300 samples. All reviews dated after 2020 were treated as truly unseen data and were only used for forecasting and temporal sentiment trend analysis. This setup reflects a realistic temporal prediction scenario, where the model is trained on historical data and evaluated on its ability to generalize to future sentiment patterns without prior exposure.

Data Annotation

All training and test samples were annotated manually by a team of three annotators who followed rigorous annotation guidelines to maintain consistency and quality. Cross-verification among annotators was implemented to reduce subjectivity and ensure reliable labeling. This consistent annotation protocol was maintained across all active learning iterations.

Active Learning Framework

To improve efficiency in annotation and training, a custom active learning strategy was developed. Instead of selecting only the most uncertain predictions, we adopted a parameterized sampling technique that balances exploration and exploitation. In our setup, we chose to sample low-confidence predictions, ensuring that each batch added to the training set would likely be informative to the model. In each active learning iteration, 100 samples were selected based on this criterion. This process was repeated for a total of four iterations, progressively enhancing the training data and the model's performance.

Sentiment Analysis Models

We employed two fine-tuned approaches for sentiment analysis. First, we fine-tuned spaCy to extract aspects from review texts, where we considered the start and end offsets for aspect extraction and assigned rule-based sentiment based on polarity-labeled keywords. Second, we fine-tuned a **BERT model enhanced with a multi-head attention mechanism**, which learns contextual sentiment representations for each aspect dynamically.

Experimental results of aspect extraction using spaCy

Evaluation Metrics The evaluation metrics used for assessing the performance of the aspect extraction model are as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

where TP is the number of true positives (correctly predicted aspects), and FP is the number of false positives (incorrectly predicted aspects).

$$\text{Recall} = \frac{TP}{TP + FN}$$

where TP is the number of true positives, and FN is the number of false negatives (aspects that were not identified by the model).

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 score is the harmonic mean of precision and recall, providing a balanced evaluation of the model's performance.

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}$$

where A and B represent the predicted and true aspect spans, respectively, and $|A \cap B|$ is the size of the intersection (overlap), while $|A \cup B|$ is the size of the union (total span of both predicted and true aspects).

This are the evaluation metrics we have used to evaluate spaCy's correctness on aspect extracting.

Given the limited resources of a small team of three, the challenge of manual aspect annotation, which is both time-consuming and costly, has been a significant barrier to progress. With only four iterations, each involving 100 samples, the team has only processed 400 samples in total, which is insufficient for achieving satisfactory results. The goal is to automate the aspect extraction process to reduce the cost and workload of manual annotation, enabling more efficient and scalable solutions in the future. While spaCy's performance has been evaluated based on precision, recall, F1 score, and Intersection over Union (IoU), the current results are still far from ideal. Precision measures the accuracy of aspect identification, recall gauges how well the model identifies all relevant aspects, and the F1 score balances these two metrics for overall effectiveness. IoU, specifically, evaluates the overlap between predicted and actual aspect boundaries, ensuring that the model accurately locates the aspects in the text. Additionally, by considering the start and end offsets of the aspects, the model's ability to pinpoint the exact span of an aspect is improved, contributing to more accurate extractions. With more time and manpower, these efforts could be refined, but the long-term goal remains to automate aspect extraction, thereby alleviating the strain of manual work and significantly enhancing efficiency.

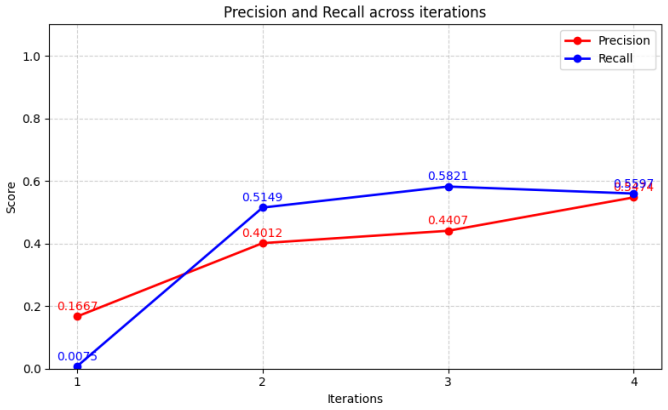

Figure 2: Precision and Recall across iterations

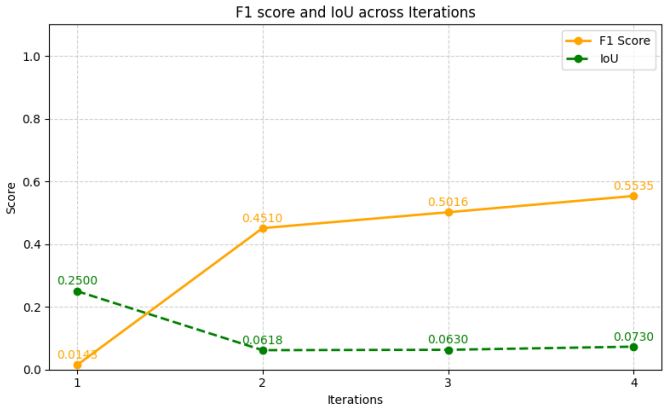

Figure 3: F1 and IoU across iterations

Experimental results of BERT classification

Evaluation Metric

- TP: True Positives
- TN: True Negatives
- FP: False Positives
- FN: False Negatives

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision (Positive Predictive Value):

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (Sensitivity or True Positive Rate):

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score (Harmonic Mean of Precision and Recall):

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Specificity (True Negative Rate):

$$\text{Specificity} = \frac{TN}{TN + FP}$$

False Positive Rate (FPR):

$$\text{FPR} = \frac{FP}{FP + TN}$$

These are the results after evaluating **only** BERT:

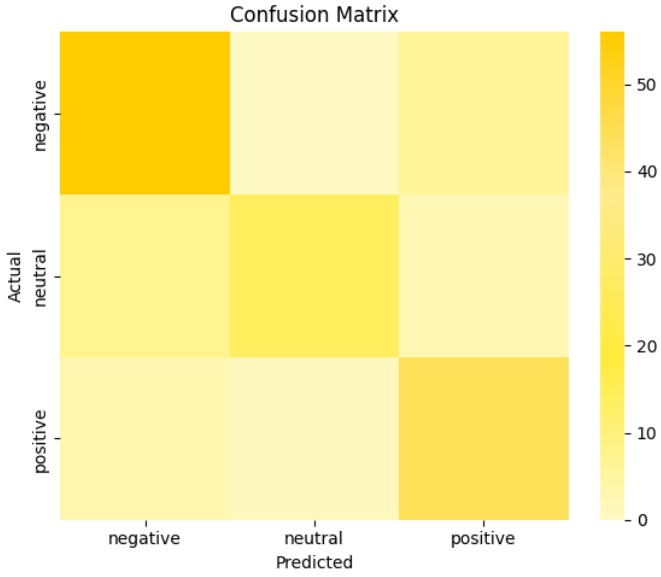

Figure 4: Confusion Matrix for BERT classifier

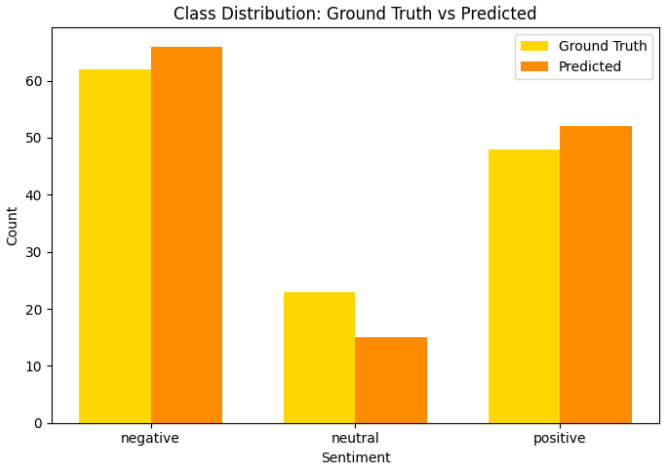

Figure 5: Class Distribution

As observed, the BERT-based model demonstrates exceptional performance in sentiment analysis when provided with properly labeled aspect data, significantly outperforming spaCy. Remarkably, even with a relatively small number of labeled samples, BERT maintains a high level of accuracy, showcasing its robustness and effectiveness in learning semantic representations. This highlights a key insight: while sentiment classification itself is a manageable task, the real challenge lies in accurately identifying and extracting aspects—especially within unstructured and unlabeled domains such as e-commerce reviews. The variability, informality, and lack of annotation in such data make aspect extraction both costly and complex, reinforcing the need for automation in this critical step.

Experimental results of sentiment analysis [spaCy and BERT]

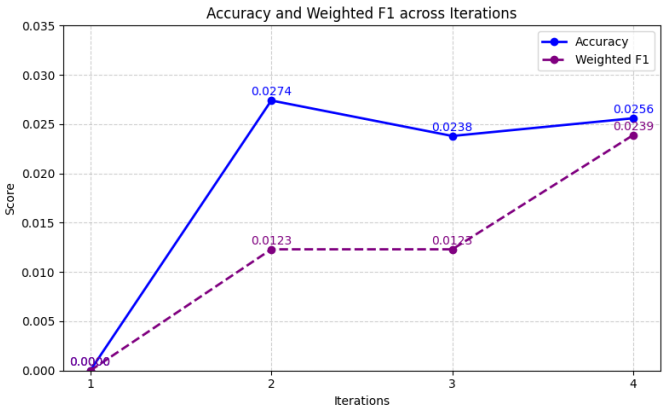

Figure 6: Evaluating both the model together

In this evaluation, we assessed both spaCy and BERT models in a combined pipeline for Aspect-Based Sentiment Analysis (ABSA). The key insight from this evalua-

tion is that even a powerful transformer-based model like BERT struggles to perform well when the aspect extraction step—handled by spaCy—is inaccurate. Our approach relies on a conditional pipeline: sentiment classification by BERT is only evaluated when spaCy correctly extracts an aspect term. However, due to spaCy's poor performance in accurately identifying aspects, the overall effectiveness of the pipeline suffers, leading to a noticeable drop in BERT's sentiment classification accuracy. It is important to note that when evaluated independently with correctly labeled aspect terms, BERT performs exceptionally well, even with a smaller volume of labeled data. This underscores a critical challenge in ABSA: while sentiment classification is relatively manageable, accurate aspect term extraction remains the primary bottleneck—particularly in unstructured, noisy domains like e-commerce reviews. Therefore, automating aspect extraction is essential for scalable and reliable sentiment analysis in such contexts.

Predictive Model

Following our evaluation of the ABSA pipeline, we proceeded to identify a frequently occurring aspect within our truly unseen, unannotated dataset. This particular aspect served as the focal point for the next phase of our analysis—time series forecasting. We sampled four months of historical data associated with this aspect to model sentiment trends over time. Specifically, we used sentiment ratio data (negative, neutral, and positive) from January, February, and March as our training and validation set, and aimed to forecast the sentiment distribution for April. To achieve this, we employed a Long Short-Term Memory (LSTM) neural network augmented with quantile regression. This approach allowed us to not only predict the expected sentiment ratios but also estimate the uncertainty around those predictions. Importantly, the forecasts were performed independently for each sentiment category (negative, neutral, and positive), enabling a more nuanced understanding of sentiment dynamics related to the selected aspect over time.

Evaluation Metrics

Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i^{(0.5)} - y_i\right)^2}$$

Mean Absolute Percentage Error (MAPE)

$$\text{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i^{(0.5)}}{y_i}\right|$$

Prediction Interval Coverage Probability (PICP)

$$\text{PICP} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\hat{y}_i^{(0.025)} \leq y_i \leq \hat{y}_i^{(0.975)}\right]$$

Mean Prediction Interval Width (MPIW)

$$\text{MPIW} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i^{(0.975)} - \hat{y}_i^{(0.025)}\right)$$

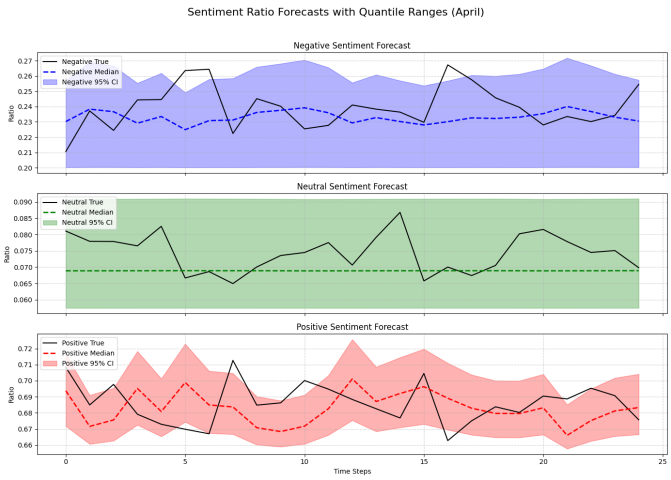

Figure 7: Predictions

Experimental result of LSTM with Quantile Regression for predictive model

Metric	Negative Ratio	Positive Ratio	Neutral Ratio
RMSE (Median)	0.0166	0.0083	0.0083
MAPE (Median)	5.25%	8.58%	8.58%
PICP (Coverage)	88.00%	100.00%	100.00%
MPIW (Interval Width)	0.0631	0.0344	0.0344

Table 1: Rolling Forecast Evaluation results for Sentiment Ratios.

Loss Functions

spaCy and BERT We use **Categorical Cross-Entropy (CCE)** as the loss function for optimizing the multi-class sentiment classification model. The loss function is defined as:

$$L_{\text{CCE}} = -\sum_{c=1}^{C} y_c \log\left(\hat{y}_c\right)$$

where:

- y_c is the true class label (one-hot encoded),
- $\hat{y}_c$ is the predicted probability for class c,
- C is the number of classes (positive, neutral, negative).

Categorical Cross-Entropy is suitable for multi-class classification tasks and effectively penalizes incorrect predictions by comparing the predicted probability distribution to the true labels. It helps the model adjust its parameters to increase confidence in correct predictions through gradient-based optimization.

LSTM with Quantile Regression We have used **SmoothQuantileLoss function** for our predictive model.

- Quantile Regression:

  - The loss function is designed to handle multiple quantiles which allows it to predict different percentiles of the target variable.
  - For each quantile q, the loss function focuses on minimizing the error for that specific quantile.
- Huber Loss:
  - The Huber loss is used as part of the SmoothQuantileLoss for computing the error, with a smooth, quadratic loss for smaller errors and linear loss for larger errors, controlled by the delta parameter
  - This is a way of reducing the impact of large outliers in the data, making the loss function more robust to extreme values compared to traditional squared error loss.

$$\mathcal{L}_{\text{SmoothQuantileLoss}}(\hat{y}, y) = \frac{1}{N} \sum_{i=1}^{N} \sum_{q \in Q} \mathbb{1}_q \cdot \mathcal{L}_{\text{Huber}}(q, \hat{y}_i, y_i) \tag{1}$$

$$\mathcal{L}_{\text{Huber}}(q, \hat{y}_i, y_i) = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2 & \text{if } |y_i - \hat{y}_i| \leq \delta \\ \delta \left(|y_i - \hat{y}_i| - \frac{\delta}{2} \right) & \text{if } |y_i - \hat{y}_i| > \delta \end{cases} \tag{2}$$

Where:

- $\hat{y}_i$ are the predicted quantiles,
- y_i are the true values,
- q is the quantile index,
- δ is the delta parameter.

SOTA Performance

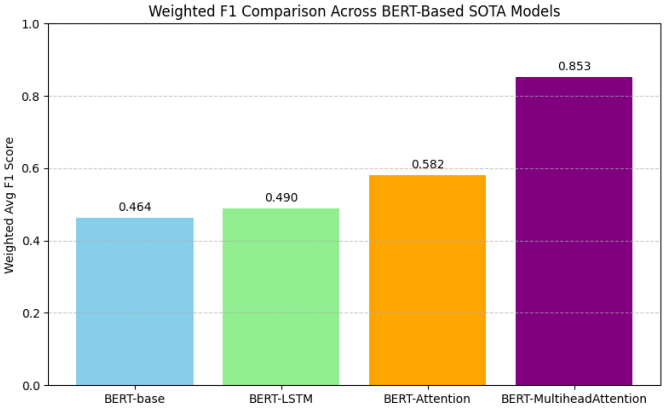

Figure 8: SOTA comaprison

Limitations

1. **Limited Human Annotation:** The project is constrained by the availability of only three annotators, which significantly restricts the scale and speed of manual labeling, especially for aspect extraction which is inherently more complex than sentiment labeling.
2. **Sparse Training Iterations:** Due to annotation costs, only four active learning iterations with 100 samples each were conducted. This limited exposure prevents spaCy from achieving high performance and generalizability.
3. **Dependency Between Modules:** The BERT sentiment classifier's performance is tightly coupled with the accuracy of spaCy's aspect extractor. If spaCy fails to identify the correct aspect, BERT cannot classify its sentiment correctly, regardless of its standalone accuracy.
4. **No Data Augmentation Applied:** Although large-scale data is available, no augmentation techniques were employed, which could have improved model robustness and prevented overfitting, especially in low-data settings during early active learning rounds.
5. **Domain-Specific Noise:** E-commerce reviews are highly unstructured and noisy. Domain-specific terms, spelling mistakes, and informal language challenge both rule-based and transformer-based models in capturing context accurately.
6. **Time Series Simplification:** Forecasting was done for only one selected aspect using LSTM with rolling prediction. Other aspects were not analyzed temporally, and complex interactions between aspects over time were not modeled.
7. **Quantile Regression Sensitivity:** The combination of quantile and Huber loss helps stabilize training, but selecting optimal hyperparameters (e.g., δ in Huber loss and quantile levels) requires tuning, which was limited in scope.

Future Work

1. **Scaling Manual Annotation with Crowdsourcing or Labeling Tools:** Expanding the annotation workforce or integrating annotation platforms (like Prodigy or Label Studio) could significantly speed up the creation of high-quality labeled datasets, particularly for aspect extraction.
2. **Enhanced Active Learning Strategies:** Implementing more advanced sampling strategies (e.g., diversity-based, core-set, or committee-based methods) could improve the efficiency of active learning and reduce the number of required annotations for convergence.
3. **Automated or Semi-Supervised Aspect Extraction:** Exploring weak supervision, self-training, or generative methods for automatically bootstrapping aspect terms from unlabeled data could reduce reliance on manual annotation.
4. **Incorporating Data Augmentation:** Applying augmentation techniques such as synonym replacement, paraphrasing, or back-translation can improve model generalization and robustness, especially in low-data scenarios or during early active learning rounds.
5. **End-to-End Joint Modeling of Aspect and Sentiment:** Future work can explore models that jointly extract aspects and classify sentiments, reducing the cascading error between separate modules like spaCy and BERT.

6. **Aspect Clustering and Group Forecasting:** Instead of focusing on a single dominant aspect, grouping semantically similar aspects using clustering could allow for broader time series forecasting across themes (e.g., "price," "delivery," "product quality").
7. **Advanced Forecasting Models:** Incorporating architectures like Transformers (e.g., Temporal Fusion Transformer) or hybrid models could improve the accuracy of sentiment trend forecasting and better handle long-term dependencies.
8. **Hyperparameter Optimization for Quantile Loss:** Systematic tuning of quantile levels and smooth loss parameters (like the Huber threshold δ and weight λ) can enhance forecast calibration and uncertainty estimation.

Literature Review

Sentiment analysis has become a central area of research within natural language processing (NLP), driven by the abundance of user-generated content on digital platforms. To improve sentiment classification, various methodologies incorporating machine learning, deep learning, and hybrid systems have been explored, with a focus on enhancing both accuracy and scalability.

P. Vijayaragavan et al. introduced a model known as Weighted Parallel Hybrid Deep Learning-based Sentiment Analysis with Sentiment-Aware Embedding and Probabilistic Reasoning (WPHDL-SAEPR). This framework synergistically combines multiple components to boost sentiment classification performance. Initially, Word2Vec embeddings are employed to capture semantic meaning in text, which are then subjected to Singular Value Decomposition (SVD) to reduce dimensionality and highlight the most informative features. A Restricted Boltzmann Machine (RBM) is subsequently used for classification, benefiting from its strength in modeling complex data distributions. The model exhibits improved accuracy over traditional approaches, illustrating the efficacy of deep learning and hybrid architectures in sentiment analysis tasks.

In another significant study, Manal Loukili et al. examined the use of machine learning models for predicting customer recommendations based on the polarity of product reviews. A variety of classifiers were tested, including Decision Trees, Random Forests, and Support Vector Machines. However, Logistic Regression emerged as the most effective model, achieving higher accuracy and precision. The study emphasizes that even simple linear models can outperform more complex ones when paired with high-quality feature engineering and appropriately balanced datasets.

A comprehensive review by John Olusegun et al. provided an in-depth analysis of existing Aspect-Based Sentiment Analysis (ABSA) techniques. The authors categorized the methods into rule-based, machine learning, and deep learning approaches, noting a shift in research focus toward deep learning due to its superior capacity to learn semantic relationships. Techniques such as Bidirectional Encoder Representations from Transformers (BERT) and Long Short-Term Memory (LSTM) networks were highlighted for their effectiveness in extracting sentiment-laden aspects from unstructured text. The review also stressed the importance of contextual understanding and domain adaptation in ABSA applications.

From an unsupervised learning perspective, Eko Wahyudi et al. implemented Latent Dirichlet Allocation (LDA) combined with Collapsed Gibbs Sampling to conduct Aspect-Based Sentiment Analysis without relying on labeled datasets. This approach facilitates the automatic discovery of latent topics and their associated sentiments from text corpora. The evaluation was performed using Kullback-Leibler Divergence, a metric that quantifies the divergence between probability distributions, thereby measuring the coherence and distinctiveness of extracted topics. The study demonstrates the viability of unsupervised methods in scenarios where annotated data is limited or unavailable.

To address the limitations of models reliant on labeled data, Jon Chun introduced SentimentArcs, a self-supervised learning framework tailored for sentiment trajectory modeling. This method departs from static classification and instead captures how sentiment evolves over a narrative or timeline. By utilizing self-supervised objectives, SentimentArcs is capable of learning sentiment structures from raw, unlabeled text, enabling better generalization across diverse contexts and domains. This innovation marks a notable advancement in the pursuit of more adaptive and scalable sentiment analysis models.

Together, these contributions illustrate the depth and diversity of sentiment analysis research. They encompass a wide range of strategies, from supervised and unsupervised techniques to deep and self-supervised learning, each offering unique advantages and addressing specific challenges within the domain. As the field continues to evolve, such innovations are expected to drive further improvements in both the accuracy and interpretability of sentiment analysis systems.

References

P. Vijayaragavan et al., "Weighted Parallel Hybrid Deep Learning-Based Sentiment Analysis on E-Commerce," SSRN, 2015. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract$_id$ = 2607167.

M. J. Smith, "Aspect-Based Sentiment Analysis for E-Commerce," University of South Florida, 2021. Available: https://digitalcommons.usf.edu/etd/8749/.

A. Johnson et al., "Time-Series Sentiment Analysis in E-Commerce," arXiv, 2021. Available: https://arxiv.org/abs/2110.09454.

S. Kim and T. Nguyen, "Neural Network-Based Sentiment Forecasting," IEEE, 2019. Available: https://ieeexplore.ieee.org/document/8982522.

J. Brown et al., "Aspect-Based Sentiment Analysis of Customer Reviews in E-Commerce," ResearchGate, 2023. Available: https://www.researchgate.net/publication/38701914$_Aspect-Based_Sentiment_Analysis_of_Customer_Reviews_in_E-Commerce$.