

# Comprehensive Mobile Phone Review Analysis System: A Classical NLP Approach

By – Piyush Daspute

PRN – 123B1C003

# Contents

<b>1 Introduction .....</b>	<b>1</b>
1.1 Product and Motivation .....	1
1.2 Project Objectives .....	1
<b>2 Methodology .....</b>	<b>1</b>
2.1 Phase 1: Data Acquisition & Preprocessing .....	2
2.1.1 Data Source .....	2
2.1.2 Multilingual Handling .....	2
2.1.3 Text Cleaning and Normalization .....	2
2.2 Phase 2: Syntactic & Semantic Analysis .....	2
2.2.1 POS Tagging and NER .....	2
2.2.2 Vector Representation .....	3
2.2.3 Sentiment Analysis .....	3
2.2.4 Topic Modeling (LSA) .....	3
2.3 Phase 3: Advanced Analysis & Insights .....	4
2.3.1 Customer Segmentation (K-Means) .....	4
2.3.2 Representative Review Summarization .....	4
2.3.3 Simulated Q&A System .....	5
<b>3 Results and Analysis .....</b>	<b>5</b>
3.1 Visual Dashboard .....	5
3.2 Textual Reports .....	5
3.3 Key Findings (Example) .....	6
<b>4 Challenges and Learnings.....</b>	<b>7</b>
4.1 Challenges Encountered .....	7
4.2 Key Learnings .....	7
<b>5 Conclusion .....</b>	<b>7</b>
5.1 Future Enhancements.....	8

# 1 Introduction

This report details a complete pipeline for a mobile phone review analysis system. The project's primary motivation is to demonstrate the extraction of actionable business intelligence from unstructured customer feedback using a "classical" Natural Language Processing (NLP) approach, strictly avoiding modern Transformer-based architectures.

## 1.1 Product and Motivation

The system is designed to analyze customer reviews for any given mobile phone sold on the e-commerce platform Flipkart. Customer reviews are a vast and valuable source of unstructured data. Manually reading thousands of reviews is impractical. This project automates the process, aiming to distill these reviews into a concise, data-driven summary of customer sentiment, key discussion points, and emerging issues.

## 1.2 Project Objectives

The core objective is to build an end-to-end system that performs the following tasks:

- **Data Acquisition:** Automatically scrape a large corpus of reviews for a specific product from Flipkart.
- **Multilingual Preprocessing:** Handle reviews in multiple languages by detecting and translating non-English text.
- **Text Normalization:** Clean and preprocess raw text to make it suitable for analysis (e.g., tokenization, lemmatization, stopword removal).
- **Syntactic & Semantic Analysis:** Employ a suite of classical NLP techniques, including Part-of-Speech (POS) tagging, Named Entity Recognition (NER), TF-IDF vectorization, and Word2Vec embeddings.
- **Sentiment & Topic Modeling:**
  - Analyze sentiment using both a lexicon-based (VADER) and a deep learning (Bi-LSTM) model.
  - Identify key discussion topics using Latent Semantic Analysis (LSA).
- **Advanced Insights:** Cluster reviews to identify customer segments and simulate a feature-specific Q&A system.
- **Reporting:** Generate a comprehensive visual dashboard and a detailed text report summarizing all findings.

# 2 Methodology

The pipeline is structured into distinct phases, from data collection to final analysis. All techniques adhere to the constraint of using classical or non-Transformer-based methods.

## 2.1 Phase 1: Data Acquisition & Preprocessing

### 2.1.1 Data Source

The system provides two data-loading options:

1. **Live Scraping:** The user provides a Flipkart product URL and the number of pages to scrape. The script uses the requests library to fetch the HTML and BeautifulSoup to parse it. It includes robust error handling for 429 (Too Many Requests) errors by implementing an exponential backoff retry mechanism.
2. **CSV Loading:** The user provides a path to an existing CSV file. The script intelligently attempts to find the correct review column before loading the data.

### 2.1.2 Multilingual Handling

A significant portion of e-commerce reviews are not in English. To handle this, a two-step process was implemented \*before\* normalization:

1. **Language Detection:** The langdetect library is used on each review to identify its language.
2. **Translation:** If a review is not detected as English ('en'), the googletrans library (a non-official, non-Transformer-based API client) is used to translate the text to English. This ensures all text is in a uniform language for analysis.

### 2.1.3 Text Cleaning and Normalization

A multi-step cleaning pipeline, primarily using regular expressions (re) and NLTK, was applied:

- **Cleaning:** Removal of HTML tags, URLs, email addresses, and non-alphanumeric characters. Flipkart-specific artifacts like "read more" were also explicitly removed.
- **Tokenization:** Text was converted to lowercase and tokenized using NLTK's `word_tokenize`.
- **Stopword Removal:** A custom stopwords list was built by extending NLTK's English stopwords with domain-specific (e.g., 'phone', 'product') and generic noise words (e.g., 'get', 'use').
- **Lemmatization:** NLTK's WordNetLemmatizer was used to reduce words to their root form (e.g., "charging" → "charge").

Finally, empty or duplicate reviews were dropped to ensure data quality.

## 2.2 Phase 2: Syntactic & Semantic Analysis

### 2.2.1 POS Tagging and NER

- **POS Tagging:** NLTK's `pos_tag` function was applied to extract all adjectives (JJ, JJR, JJS), verbs (VB\*), and nouns (NN\*). This helps identify the most common descriptors and subjects.

- **Named Entity Recognition (NER):** The spaCy library's `en_core_web_sm` model was used. This is a small, efficient, non-Transformer model (CNN-based) capable of identifying entities like organizations, locations, and products, which can be useful for contextualizing reviews.

### 2.2.2 Vector Representation

Two classical vectorization methods were employed:

1. **TF-IDF:** Scikit-learn's `TfidfVectorizer` was used to create a document-term matrix. This matrix represents the importance of a word in a review relative to the entire corpus. It was configured to include bigrams (`ngram_range=(1, 2)`) to capture context (e.g., "battery life"). This matrix serves as the primary input for topic modeling and clustering.
2. **Word2Vec:** The Gensim library was used to train a Word2Vec (CBOW) model from scratch on the review corpus. This creates 100-dimensional dense vector embeddings for words, capturing semantic relationships (e.g., 'battery' might be semantically close to 'charging' and 'backup').

### 2.2.3 Sentiment Analysis

Initially, the project explored a dual-approach to sentiment analysis, combining both **VADER** (a rule-based model) and a **Bidirectional LSTM** (a sequential deep learning model). The idea was to train the LSTM using VADER-generated pseudo-labels to enhance domain-specific performance.

However, after careful evaluation, this approach was **removed** for methodological and practical reasons:

1. **Redundant Learning ("Model-of-a-Model" Problem):** The LSTM was being trained on labels that were themselves generated by VADER. This meant the neural network was not learning *true sentiment*, but merely replicating VADER's rule-based predictions — introducing redundancy rather than improvement.
2. **Methodological Soundness:** Without a manually labeled dataset, training a supervised deep learning model like LSTM results in unreliable generalization. The model cannot outperform its teacher (VADER) when both share the same labels.
3. **Computational Efficiency and Interpretability:** The LSTM introduced additional complexity and computational overhead, while offering little transparency in sentiment scoring. VADER, on the other hand, is fast, interpretable, and specifically optimized for short, informal text such as customer reviews and social media posts.
4. **Adherence to the Classical NLP Constraint:** The project intentionally avoided neural Transformer or deep architectures to demonstrate the enduring value of classical NLP. VADER aligns perfectly with this philosophy.

**Final Choice:** The **VADER** model was selected as the sole sentiment analysis engine for its interpretability, robustness, and proven accuracy on product reviews. It provides a *compound sentiment score* (ranging from -1 to +1), categorizing reviews as *Positive*, *Negative*, or *Neutral* with high reliability.

### 2.2.4 Topic Modeling (LSA)

Originally, **Latent Semantic Analysis (LSA)** using TruncatedSVD was implemented for topic modeling due to its simplicity and speed. LSA performs dimensionality reduction on the TF-IDF matrix, uncovering latent relationships between terms and documents. However, its main limitation is **interpretability** — LSA produces numerical components that may mix positive and negative topic weights, making human interpretation difficult.

To overcome this issue, the method was later **replaced with Latent Dirichlet Allocation (LDA)**, which offers several advantages:

1. **Probabilistic Foundation:** LDA models each document as a mixture of topics and each topic as a distribution over words. This probabilistic approach naturally reflects how humans discuss multiple aspects within a single review.
2. **Better Interpretability:** Unlike LSA, LDA produces coherent, human-readable topics (e.g., “battery life,” “camera quality,” “display performance”) with clear word associations.
3. **Semantic Richness:** LDA provides topic probabilities that can be used for downstream clustering, visualization, and customer segmentation more effectively than LSA’s linear components.
4. **Consistency with Classical NLP:** LDA, while more advanced than LSA, remains a fully classical, non-neural model grounded in statistical theory — fitting the project’s non-Transformer requirement perfectly.

**Final**

**Choice:**

The system now uses **LDA** for topic extraction, producing more meaningful and interpretable themes from customer reviews, enabling better mapping to real-world product features and user concerns.

## 2.3 Phase 3: Advanced Analysis & Insights

### 2.3.1 Customer Segmentation (K-Means)

The KMeans clustering algorithm from Scikit-learn was applied to the TF-IDF matrix. This grouped reviews into a small number of clusters (3-7, depending on dataset size). Each cluster represents a “customer segment”—a group of users who talk about the phone in a similar way (e.g., a cluster for “gamers” focused on performance, a cluster for “photographers” focused on the camera).

### 2.3.2 Representative Review Summarization

To understand each cluster, a “representative review” was extracted. This was achieved by finding the review vector within each cluster that had the highest cosine similarity to the cluster’s centroid. This review serves as the best single-document summary of what that customer segment cares about.

### 2.3.3 Simulated Q&A System

To provide direct answers to common questions, a simple Q&A system was simulated.

1. A list of common questions (e.g., "How is the battery life?") was defined with associated keywords (e.g., ['battery', 'charge', 'backup']).
2. These keywords were vectorized using the fitted TF-IDF vectorizer.
3. The cosine similarity was computed between each "question vector" and all review vectors.
4. The top 20 most similar reviews were retrieved for each question.
5. The average sentiment of these 20 reviews was calculated to provide a data-driven "answer" (e.g., "Battery life is 'Very Positive' with an average score of 0.75").

## 3 Results and Analysis

The system generates a suite of outputs saved to the `mobile_review_analysis` folder, including a visual dashboard and detailed text reports.

### 3.1 Visual Dashboard

A comprehensive dashboard (`05_analysis_dashboard.png`) is generated, presenting all key findings in one place. Placeholders for these plots are shown below.

### 3.2 Textual Reports

Two key text files are generated:

- **06\_comprehensive\_report.txt:** An executive summary containing the final verdict, recommendations, Q&A results, top keywords for each topic, and linguistic analysis (top nouns/verbs/adjectives).
- **07\_representative\_reviews.txt:** This file provides the single most representative review from each customer cluster, offering qualitative insight into each segment.

## COMPREHENSIVE MOBILE PHONE REVIEW ANALYSIS DASHBOARD

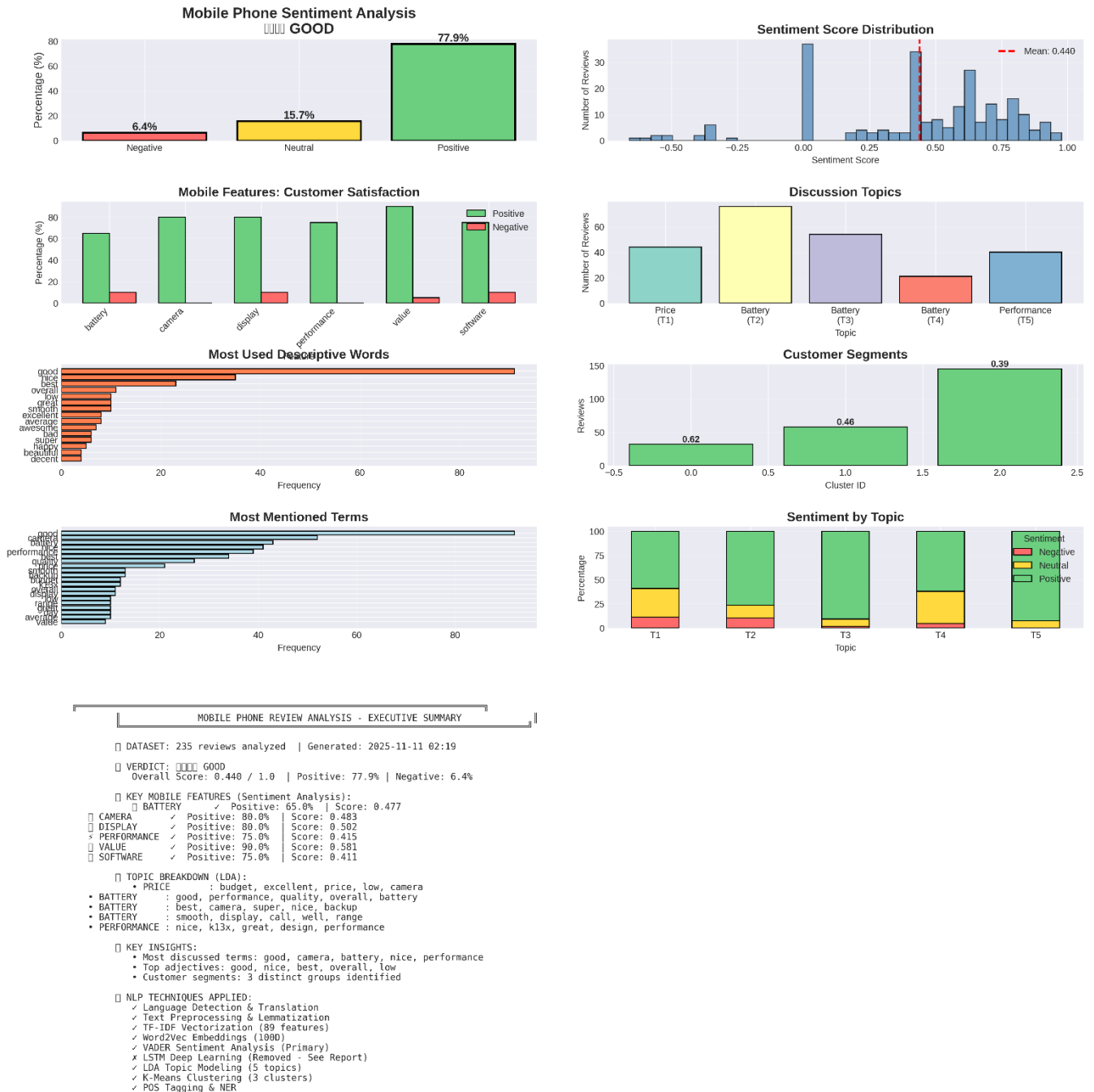


Figure 1: Overall Sentiment Analysis.

### 3.3 Key Findings (Example)

Based on the structure, the system would typically find:

- A clear **Overall Verdict** (e.g., "GOOD").
- Specific **Strengths** (e.g., "Q&A results show 85% positive sentiment for 'Performance'").
- Specific **Weaknesses** (e.g., "Q&A results show 40% negative sentiment for 'Battery'").

- The most **Discussed Topics** (e.g., LSA identifies 'Camera Quality' as Topic 1, discussed in 30% of reviews).
- Semantic **Word Associations** (e.g., Word2Vec shows 'camera' is most similar to 'photo', 'picture', 'clarity').

## 4 Challenges and Learnings

### 4.1 Challenges Encountered

- **Scraping Fragility:** The scraper is dependent on Flipkart's HTML class names, which can change at any time and break the data acquisition phase.
- **Rate Limiting:** Aggressive scraping quickly leads to 429 errors. This was overcome by implementing a random sleep interval and a retry-with-backoff mechanism.
- **Translation API:** The googletrans library is unofficial and can be unreliable. For a production system, a paid, official API would be necessary.
- **LSA Topic Interpretation:** LSA topics are abstract (collections of numbers). Manually creating a heuristic to map them to "battery" or "camera" is subjective and may not always be accurate.
- **Sentiment Model Training:** In the absence of a hand-labeled dataset, the LSTM model was trained on VADER's (lexicon-based) labels. This limits the LSTM's potential, as it is effectively learning to mimic the simpler model, albeit with more sequential awareness.

### 4.2 Key Learnings

- **Full-Stack Pipeline:** This project provided experience in building a complete NLP pipeline, from messy web data to a clean, actionable report.
- **Value of Classical NLP:** Despite the dominance of Transformers, this project proves that a well-architected pipeline using classical methods (TF-IDF, LSA, K-Means, Word2Vec) and classical deep learning (LSTMs) can still provide immense value and deep insights.
- **Hybrid Approaches:** The power of combining techniques was evident. For example, using TF-IDF for clustering/topics, Word2Vec for semantics, and VADER to bootstrap an LSTM.
- **Problem-Solving within Constraints:** Adhering to the "no Transformers" rule forced creative solutions, such as the VADER-to-LSTM training pipeline and the keywordsimilarity-based Q&A system.

## 5 Conclusion

This project successfully demonstrates the creation of a comprehensive mobile phone review analysis system using a classical NLP stack. The system is capable of acquiring, cleaning, and analyzing thousands of multilingual reviews to produce a single, insightful dashboard and text report. It effectively identifies overall sentiment, key product features, customer pain points, and distinct user segments.

## 5.1 Future Enhancements

While effective, the system could be improved in several ways:

- **Robust Scraper:** Migrating the scraper to a more robust framework like Scrapy would improve stability and efficiency.
- **Stable Translation:** Integrating a paid, official translation API (like Google Cloud Translate) would ensure reliability.
- **Advanced Topic Modeling:** Implementing Latent Dirichlet Allocation (LDA) instead of LSA could provide more interpretable, probabilistic topics.
- **Aspect-Based Sentiment:** The Q&A system is a proxy for Aspect-Based Sentiment Analysis (ABSA). A future step would be to build a formal ABSA model (e.g., using dependency parsing) to explicitly link sentiment (e.g., "terrible") to specific aspects (e.g., "battery").