DESAUTELS Faculty of Management
Faculté de gestion

INSY- 669

Text Analytics

Group Project

Review Radar

Submitted to: Prof. Taha Havakhor, Ph.D

(Group Members)

Yash Sethi

Jimmy Chu

Ashraf Elrufaie

Kevin Wang

Clement Orcibal

Review Radar – Extracting Attribute Ratings from Customer Reviews with NLP and ML

Problem Statement

99.9%* of consumers read reviews when they shop online. However, these reviews are unfocused and unstructured text, making it difficult to extract quantifiable insights. Shoppers looking for a phone that excels in one aspect while compromising on others struggle with decision-making, as reading numerous reviews is overwhelming. Review Radar addresses this challenge by transforming discussions from Reddit, Flipkart, and Amazon into numeric scores, allowing consumers to compare phones based on the features that matter most to them. Phone shoppers are to benefit from Review Radar by leveraging authentic consumer insights, helping shoppers in discerning misleading claims and saving them from arduous research on online reviews, ultimately aiding in their purchasing decisions. Phone makers and E-commerce platforms can also reap the value from Review Radar by increased traffic or sales thanks to leveled trust.

Methodology and Data Retrieval

To define a feasible project scope, the phone reviews and data pertaining to the analysis are limited to 5 flagship models of some of the most prevailing brands. The primary data source of the consumer-generated content was Reddit subforums, that are specific to those phone models in lieu of general forums of brands, to ensure the quality and focus of the texts. Praw (Python Reddit API Wrapper) was used to extract posts and comments from relevant subforums based on defined timeframe and specified keywords.

In addition, Amazon and Flipkart were scraped to collect actual customer reviews as these sources provided structured feedback on key phone attributes. By integrating multiple platforms, Review Radar ensures a well-rounded assessment of each phone's strengths and trade-offs.

Text Preprocessing

The extracted texts were first preprocessed by which URLs, stop words, special characters, punctuations and extra spaces were removed to retain only alphanumeric and single space in between. The texts were converted into lowercases and their root forms via lemmatization of NLTK to ensure consistency.

Extraction of Attributes and Adjectives Associated with Phone Features

We identified key features most relevant to phone shoppers, including battery, camera, screen, performance, design, software, display, and speaker. However, as consumers use varied terminology to describe these features, we utilized the Sentence-BERT model to extract the most relevant words from the reviews, aligning them with our predefined feature set. SBERT were implemented in the following steps: 1) Upon tokenizing the reviews, unique words representing the entire text body of reviews were extracted. 2) SBERT model *all-MiniLM-L6-v2,* which is a lighter and faster version of BERT was loaded to encode each unique word to render word embeddings, in which each word was converted into a numerical vector (embedding) that captures its meaning. So were the predefined features being encoded into embeddings by SBERT. 3) Cosine similarity between the feature embeddings and all unique word embeddings were being computed to find the closest attributes to each predefined feature.

After identifying the closely related words to each phone feature, the descriptive words and phrases related to the features were further extracted using SpaCy for POS tagging and dependency parsing amongst words. The process of extracting adjectives began with constructing a dictionary of each feature being keys and the list of corresponding related words being values, which are essentially synonyms to the features. Secondly, each review was then preprocessed, tokenized and converted into a SpaCy Doc object. Thirdly, each token within Doc object was being examined to check if there exists a match with any of the synonyms to the features, using *token.lemma* . If a match was found, SpaCy's dependency parsing was used to extract the associated adjectives. Here we specified the dependency relation to be of either an adjective modifier (amod) that directly modifies a noun, or an adjective complement (acomp) that is linked to a noun via a verb. Not only were single descriptive words extracted, but bi-grams and trigrams were extracted by identifying left-side modifiers. Table 1. shows part of the associated single words and phrases (N-grams) to some of the predefined features for demonstration purpose.

The justification of using Sentence-BERT in lieu of traditional NLP techniques, i.e. Word2Vec, TF-IDF, etc, was based on the advantages BERT offers in better understanding of the context, handling synonyms and capturing sentence meaning.

Table 1.

	Battery	**Camera**	**Performance**
Single Words	‘smaller', 'sustained',	'main', 'previous',	'better', 'slow', 'high', 'best
N-Grams	'quick battery','terrible battery'	'nice camera', 'different cameras'	‘slow speeds', 'high efficiency',

Sentiment Analysis on Phone Reviews

A variety of models were used to perform sentiment analysis, including BERT, VADER, TextBlob and Hugging Face. Figure 1. (Appendix) shows the sentiment scores, which were converted to scale of 1 to 5 as the ratings on the features, attained from the SBERT model. The numerical comparison of each feature amongst phones enlightens shoppers in choosing the phone that meets the need, showcasing the business value of text analytics. Figure 3. (Appendix) shows the correlation of sentiment scores from different models, which indicates moderate to strong agreement among VADER, BERT, and TextBlob, though no model is perfectly aligned. VADER and TextBlob rely on lexicons, while BERT captures contextual meaning, leading to variations in sentiment interpretation. This suggests that no single model is universally optimal, as each has strengths in different contexts. Future work may also explore ensemble techniques combining multiple models that could improve sentiment score accuracy.

The Alternatives to Sentiment Analysis – LLM Model with Structured Prompting

Apart from using conventional text analytics techniques, such as BERT in our primary analysis, we explored an alternative of leveraging OpenAI API in sentiment analysis and scoring on the features of the phones. The prompt given to Open AI (GPT 3.5) is structured to ask for firstly identifying and extracting the features of concern, which is the same list of phone features used in SBERT, from the reviews. Secondly, GPT was asked to assign a general sentiment score on each review, with the score ranging from -1 to 1. Thirdly, for each of the reviews, GPT was asked to rate the features mentioned therein on the scale of 1 to 5, based on sentiment analysis performed previously. Lastly, the structured prompt aimed to render the responses from GPT in unified format so that parsing would be of more ease. Upon receiving the responses from GPT, they were parsed by regular expression to extract the features, overall sentiment scores and ratings on features. Figure 2. (Appendix) below shows the feature ratings derived from sentiment scores on the scale of 1-5 graded by GPT for each phone model. By the visual, shopper can conveniently discern what phone might better accommodate the needs according to the features that concern them.

Classification Model

An alternative approach was explored, in which the focus was placed on classification of either positive or negative labeling rather than sentiment scores to derive ratings for the attribute. K-Nearest Neighbors (KNN) was used to assign positive and negative labels to the reviews due to the absence of predefined sentiment labels. Naïve Bayes (NB) and Support Vector Machine (SVM) classifiers were then implemented to categorize the reviews. This approach enables sentiment-based rating assignment on a 1 to 5 scale, determined by the proportion of positive reviews.

Combinations of TF-IDF, KNN and Word2Vec, SVM were tested. For the KNN mmodel;; all reviews were labeled based on a set of keywords. The classifier was initialized to consider the 5 most similar reviews. The sentiment scores of reviews were further aggregated to model level. Word2Vec was used to capture the meaning of each word by embeddings, and SVM classifier was trained on the embeddings and sentiment labels. This model is classified according to the detail of feature per phone model. Table 2. shows the classification results from KNN.

Table 2.

	Accuracy	Positive Ratio	Negative Ratio
Google Pixel	0.72	14.13%	85.87%
Oppo Find X3 Pro	0.71	17.70%	82.30%
Samsung S24 Ultra	0.82	8.76%	91.24%
Xiaomi 14 Ultra	0.72	10.23%	89.77%
iPhone 16 Pro	0.76	12.79%	87.21%

All models demonstrated high negative ratios and low positive ratios, which might be because the general public tends to have slightly negative sentiments towards products. However, by comparing the number of positive sentiments within each model, we can more accurately reflect whether the reviews' overall sentiment towards a product is good or bad.

Conclusion and Challenges

Review Radar uses a range of NLP tools and techniques, machine learning (or GenAI) to provide potential phone shoppers with a clear comparison amongst the most prevailing and newest phone models on the market at the time of the project. As seen in the correlation matrix and given BERT's strength, we recommend using BERT to expand into a broader scope including more phone models and analyze the texts from other sources of user-generated content. This can also be extended to reviews for other products.

Thus far, we have seen the ratings for the features are not so much differentiated across phone models, posing limitation on driving to the conclusion of which phone is superior to the rest. Nevertheless, the project showcased that people generally have mixed opinions on their phones, especially when it comes down to each feature, albeit iPhone's sales consistently trump others throughout the years.

With the advances made in LLM space, we experimented and found its certain limits and shortcomings. Firstly, LLMs fail to adhere to a strict output format despite clear instructions, making parsing unpredictable. Secondly, API calls of LLMs can incur huge expenses when analyzing datasets with the size exceeding thousands of reviews. (Using it on 10k reviews cost us 8 CAD) Lastly, due to the lack of specialized domain training, sentiment analysis might be misclassified due to overly generalized understanding of language.

References:

- *"Online Review Statistics: How Many People Read & Write Reviews?" *Search Engine Journal*, 2021, https://www.searchenginejournal.com/online-review-statistics/329701/.

Figure 1.

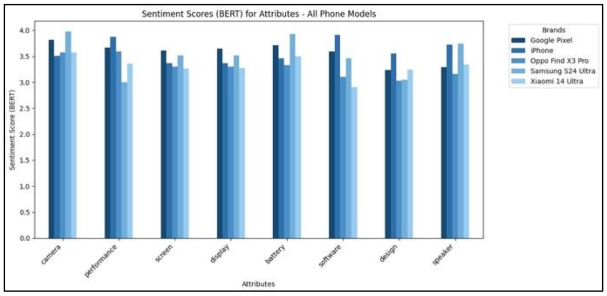


Figure 2.

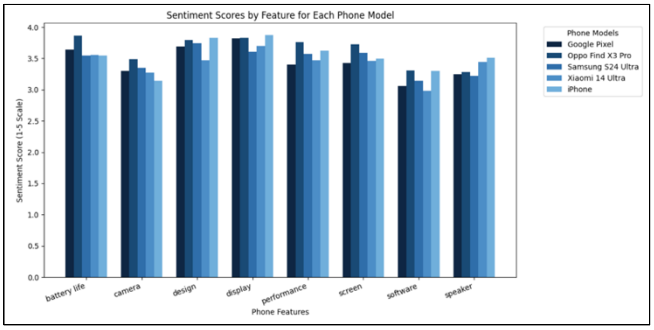
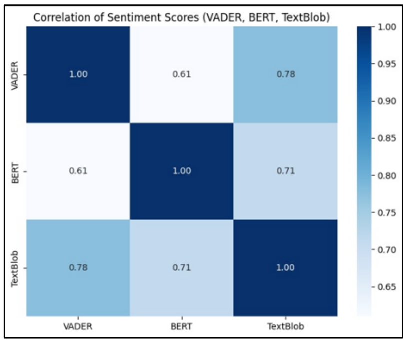


Figure 3.


Correlation of Sentiment Scores (VADER, BERT, TextBlob)
VADER
BERT
TextBlob
VADER
BERT
TextBlob