Project Brief: Analysing Spotify Customer Feedback Using NLP and Deep learnig Techniques

Nicholas Lombard and Ben Morton

1 Introduction

In the digital age, companies like Spotify receive vast amounts of customer feedback that can drive product enhancements and strategic decisions. However, the unstructured nature and sheer volume of this data make it challenging to analyze manually. Our project, *Review^2*, aims to transform complex review data into actionable insights using Natural Language Processing (NLP) and Deep learning techniques. By analyzing Spotify's Google Play Store reviews, we sought to determine customer sentiment and extract key themes related to user satisfaction and dissatisfaction.

1.1 Objective

- **Topic Identification**: Extract insights on specific aspects users are satisfied or dissatisfied with.
- **Actionable Reporting**: Provide summarized reports highlighting areas of excellence and improvement that are indistinguishable from human generated ones.

2 NLP Implementation

2.1 Data Collection and Preprocessing

- **Data Acquisition**: The dataset was downloaded from Kaggle.
- **Data Cleaning**: Removed missing values and very short reviews (five characters or fewer).
- **Language Filtering**: Focused on English reviews using the *langdetect* library with a threshold to exclude reviews with less than 60% English content.

2.2 Text Normalization

- **Standardization**: Converted text to lowercase; removed URLs, HTML tags, and non-ASCII characters.
- **Slang Replacement**: Replaced abbreviations and slang with standard English (e.g., ‘u’ to ‘you’, ‘idk’ to ‘I don’t know’).
- **Word Standardization**: Unified similar words (e.g., ’ad’ and ’ads’ to ’advertisement’).

2.3 Custom Dictionaries and Stopwords

- **Dictionary Building**: Combined words from NLTK and WordNet.
- **Stopword Expansion**: Removed frequent but non-informative words and sentiment words (e.g., ‘great’, ‘bad’) to focus on substantive content.

2.4 Lemmatization and Tokenization

- **Lemmatization**: Reduced words to their base forms (e.g., ’playing’ to ’play’).
- **Tokenization**: Split text into individual words.
- **Filtering**: Removed non-English words and custom stopwords.

2.5 Sentiment Labeling

- **Classification**: Labeled reviews as *Positive* (ratings of 4 or 5) or *Negative* (ratings of 1, 2, or 3).
- **Rationale**: Included 3-star ratings in ‘Negative’ to capture nuanced dissatisfaction.

2.6 Topic Modeling with LDA

- **Model Selection**: Chose Latent Dirichlet Allocation (LDA) for its balance of performance and interpretability.
- **Separate Models**: Built LDA models for positive and negative reviews separately.
- **Optimal Topics**: Determined the optimal number of topics using Coherence Scores.
- **Topic Assignment**: Assigned dominant topics to each review based on probability.

3 Deep Learning

For our deep learning section, we approached this problem in two distinct ways, which can be broken down into the following key steps:

3.1 Approach 1

- **Loading Reviews**: The process begins by ingesting reviews related to a specific topic.
- **Iterative Abstractive Summarization**: The reviews are batched and then summarized in a divide-and-conquer algorithm using a summarization BART model.
- **Generation**: The summary is passed to a large language model (LLM) and converted into a well-worded paragraph. A heading is generated to capture the paragraph's content.
- **Repeat**: This process is repeated for both positive and negative topics, and at the end of each section, an action paragraph is generated based on the section's context.
- **Format**: The generated topic paragraphs, along with an introduction, are formatted into a report sent to the client.

3.2 Approach 2

- **Load Topic Data**: Load reviews from CSV files into a list of strings per topic.
- **Tokenization**: Takes a batch of reviews and tokenizes it.
- **Embedding Generation**: Converts reviews into embeddings.
- **Clustering with K-Means**: Groups reviews into clusters based on their embedding similarities.
- **Selecting Representative Reviews**: The most representative reviews from each cluster are selected by comparing their similarity to the cluster centroid.
- **Summarization**: A summarization model condenses the selected reviews into a concise and coherent form.
- **Generation**: The summary is passed to an LLM and converted into a well-worded paragraph. A heading is generated to capture the paragraph's content.
- **Repeat**: This process is repeated for both positive and negative topics, and at the end of each section, an action paragraph is generated.

- **Format**: The generated topic paragraphs, along with an introduction, are formatted into a report sent to the client.

4 Technical Summary of Pre-trained Models Used

4.1 Llama 3.2 90b (Generation)

The Llama 3.2 90b model employs a transformer architecture which utilizes multi-head self-attention mechanisms to capture complex dependencies between tokens across the entire input, allowing it to understand context and meaning more deeply. This model was trained on a diverse mixture of publicly available datasets and licensed datasets, encompassing a wide range of sources including books, websites, Wikipedia articles, scientific papers, and other written content. This diverse training corpus enables the model to generate high-quality text outputs, making it suitable for various natural language processing tasks. Additionally, the model's large parameter count (90 billion parameters) allows it to leverage vast amounts of information and learn intricate patterns. I was able to access this model through a software service called 'groq', which allowed me programmatic inference access through calls to their api.

4.2 all-MiniLM-L6-v2 (Embedding)

The all-MiniLM-L6-v2 model builds on the Transformer architecture introduced by the original BERT model. It retains BERT's bidirectional attention mechanism, which allows it to capture context from both directions in a sentence. This is in contrast to autoregressive models like GPT, which only consider past tokens. MiniLM's self-attention mechanism enables a more comprehensive understanding of the relationships between all tokens in the input.

Developed through knowledge distillation, MiniLM is a smaller model (the "student") trained to mimic the outputs of a larger model (the "teacher"), achieving efficiency without sacrificing performance. Its training involved tasks similar to those used for BERT, such as randomly masking words in a sentence and predicting those masked words based on context, as well as determining whether a pair of sentences are contiguous in the original text.

It has approximately 22 million parameters with 6 transformer layers and 384 hidden dimensions, MiniLM is significantly smaller than BERT, yet it maintains competitive performance on tasks like semantic similarity and paraphrase identification. I chose this model specifically because of its lower computational cost and it's easy to deploy nature using Huggin faces transformer library.

4.3 Facebook BART (Summarization)

Bidirectional and Auto-Regressive Transformers combine the strengths of both BERT and GPT, featuring a flexible encoder-decoder architecture.The model has 406 million parameters. This includes 12 encoder layers and 12 decoder layers. The encoder processes the input text, capturing contextual information through bidirectional attention, while the decoder generates the output sequence in an autoregressive manner, attending to previous tokens. BART is pre-trained on a large corpus of text with a denoising autoencoder objective, where random parts of the input text are masked or corrupted (for example, through shuffling or token deletion), and the model learns to reconstruct the original text. This approach not only enhances the model's robustness but also allows it to generalize well across a variety of tasks. The "facebook/bart-large-cnn" model is fine-tuned specifically for extractive and abstractive summarization tasks, using datasets that include news articles and their corresponding summaries. Again, we implemented this model with Hugginfaces transformers package.

5 Our Approaches' Limitations

In our first approach, we encountered high computational costs due to the repeated calls to the BART model. Another issue was the loss of viewpoint diversity, as multiple rounds of summarization compressed information, losing less common insights. Lastly, while we weren't able to confirm any instances, there's always the potential risk of hallucination, where iterative abstractive summarization could introduce inaccuracies, raising issues that don't actually exist. We found that the second approach produced better results. Our only concern would be the dependency on the quality of clustering but incorporating evaluation metrics and dynamically adjusting the number of clusters or using multiple randomized cluster initializations and then averaging them would help ensure the best clustering solutions.

6 Our Report's Core Results and Findings

An addendum of our Generated report is attached but some of our core findings are as follows.

6.1 Positive Themes

1. Exceptional Music Discovery: Users praise the vast and diverse music library.

2. User-Friendly Interface: The intuitive design enhances the listening experience.

3. Personalized Access: Immediate access to favorite tracks and personalized playlists.

6.2 Negative Themes

1. Intrusive Advertising: Frequent ads disrupt the listening experience.

2. Feature Limitations: Restrictions on free users reduce app value.

3. Aggressive Premium Promotion: Users feel pressured to upgrade.
4. Technical Issues: Problems like app freezing hinder usability.

7 Conclusions

Our NLP and deep learning pipeline effectively transformed unstructured review data into meaningful insights. Addressing the identified issues—such as ad frequency, feature limitations, and technical problems—can enhance Spotify's user satisfaction and maintain its competitive advantage. The positive feedback highlights areas like music discovery and user interface improvements that should be prioritized.

8 Ethical Considerations

8.1 Data Privacy and Anonymity

- **Public Data Usage**: Utilized publicly available reviews from the Google Play Store.
- **Anonymization**: Ensured no personally identifiable information was included.

8.2 Bias and Fairness

- **Language Bias**: Acknowledged potential bias by focusing on English reviews.
- **Sentiment Accuracy**: Recognized that rating-based sentiment may not capture all nuances.