Write a brief report or summary in a PDF file:

5.1. A Description of the Dataset Used

The dataset comprises customer reviews of Amazon products. Each record holds data that relates a particular product, such as product name, category review ratings, review dates, brand and the actual reviews from customers. For this analysis, the primary focus was on this review text column, named *review.text*, where customers shared their thoughts and feedback on products. These reviews serve as valuable data for sentiment analysis and similarity comparison to gauge customer sentiment and review patterns.

5.2. Details of the Preprocessing Steps

The pre-processing steps used in preparing the data for analysis are:

Handling missing data: Removed any rows with missing reviews using the dropna() method. Dealing with missing data helps to prevent errors during processing.

Text Normalization: Converted all reviews to lowercase using the lower() method to maintain uniformity. Leading and trailing whitespaces are removed using the strip() function to clean the text further.

Stop Word Removal: Removed common stop words such as "and," "the," "is" using spaCy's .is_stop attribute to focus on meaningful content.

Tokenization: Each review is processed through the spaCy NLP pipeline to break it down into tokens.

Ensuring Text Consistency: The str() function is used to handle non-string data types and ensure all inputs are processed as strings.

5.3. Evaluation of Results

The 2 outputs that analysis produced are:

Sentiment Polarity

Using the SpacyTextBlob pipeline, the model calculated the polarity of each review as a floating-point value between -1 (negative sentiment) and +1 (positive sentiment). Sample reviews showed accurate polarity scores, with positive reviews yielding high positive values and negative ones showing low or negative values. For instance, a review stating "This product is amazing and exceeded my expectations!" would have a high positive polarity, demonstrating effective sentiment detection.

Similarity Score

The similarity between pairs of reviews was computed using spaCy's Doc.similarity() function. The scores are on a spectrum from 0-1with 0 representing complete dissimilarity to 1, which is identical; depending on the semantic overlap between reviews. For instance, two reviews discussing similar features of a product yielded higher similarity scores.

5.4. Insights into the model's strengths and limitations.

Strengths

The model captures sentiment polarity effectively, identifying both negative and positive sentiments accurately in most cases. Example: The negative sentiment of "definitely recommend paperwhite instead" was correctly detected, reflecting dissatisfaction with the original Kindle.

The similarity score (0.72798) accurately reflects the semantic overlap between two reviews. Both reviews discuss lightweight and portable qualities, resulting in a relatively high similarity score. The preprocessing steps (e.g., lowercase conversion, stop-word removal, and handling missing values) allowed the model to clean and process the data efficiently, even with user-generated noise like fragmented sentences.

The use of polarity scores provides numerical outputs that are easy to interpret for understanding the sentiment of reviews.

Limitations

While the sentiment analysis works well for straightforward positive and negative phrases, nuanced cases or mixed sentiment may challenge the model. E.g. "m happy, little dark" is classified as Positive, but the sentiment might be mixed due to "little dark," which could reflect dissatisfaction.

Also the model's performance heavily relies on effective preprocessing. For example, certain abbreviations like "nt" (for "didn't") might not be accurately interpreted without advanced text cleaning. Sentiment analysis results are binary (positive or negative), without additional layers of emotion or gradation such as joy and sadness.

The model cannot handle sarcasm or ironic statements effectively, potentially leading to incorrect sentiment classification.

Semantic similarity might favour reviews with overlapping terms, even if the context differs. For example, two reviews mentioning "beach" may score high similarity even if their sentiments differ.

In Conclusion

The model demonstrates significant strengths in performing sentiment analysis and computing review similarities, providing meaningful insights into consumer feedback however, its limitations around contextual nuance, preprocessing dependency, and nuanced emotions suggest room for improvement. Integrating larger models, incorporating sarcasm detection, or employing advanced preprocessing techniques could enhance the model's performance for real-world applications.