Summary of the sentiment_analysis application

The python application in this Capstone Project on NLP Applications performs sentiment analysis on a dataset of product reviews from amazon.

Dataset Used

The dataset has been downloaded from Kaggle via the link below:
https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products

It contains the reviews from 5000 users and has 24 columns or fields such as the ID, date added and name of product. The reviews cover 23 unique products largely of either Amazon Fire, Amazon Kindle or Amazon Echo. In terms of the datatypes, 20 fields are objects, 2 are integers, 1 is a float and 1 is a Boolean.

The field that we use for our analysis is “reviews.text” where we use the spaCy and Textblob models along with the similarity and polarity functions to measure how positive or negative the reviews are. We stored this data in the reviews_data variable.

Preprocessing Steps

In order to make the dataset ready for use we performed data cleaning doing the following:

1. Removed any rows with missing data using the dropna() function – there was no missing data as 5000 reviews remained after doing this step.
2. Created a function to remove stopwords called “remove_stopwords” from the text. Stopwords are words that don’t add a lot of meaning to the sentence such as “the” or “is”. We then applied this function to the dataset and stored the data in a new variable called reviews_without_stopwords.

Evaluation of Results

We created a function using TextBlob to take the reviews without stopwords and apply a polarity score to them to give an indication of sentiment. A polarity score of 1 indicates a very positive sentiment, while a polarity score of -1 indicates a very negative sentiment. A polarity score of 0 indicates a neutral sentiment. We adjusted the neutral category to be a score of between -0.05 and 0.05 in case there were rounding errors from the model. We then tested this model on 3 random reviews to make sure it is working correctly.

Having run this function on all 5000 reviews we noted there were 4,289 (86%) positive summaries, 173 (3%) negative summaries and 538 (11%) neutral summaries. This indicates that the amazon reviews for these technology products were largely positive.

We were also able to create functions to calculate similarity scores between two reviews. This was done by first creating a function to tokenise the reviews using nlp and then creating a second function to calculate the similarity scores between the reviews using the similarity function in spaCy. A similarity score of 1 indicates that two reviews are similar, while a similarity score of 0 indicates the two reviews are not similar. Testing this on the first two reviews in the dataset yielded a similarity score of 0.7 indicating high similarity.

Insights into the model's strengths and weaknesses

The model created can give a good overview and summary statistics of public sentiment to a particular product. As there are multiple products in the dataset, it may be necessary to group the sentiments by each product type or particular model for more granular analysis.

It is relatively easy to use and is based on pre-trained models with predefined lists of words reflecting their associated polarity scores, so has good consistency in application. However, the model may not be as accurate as a result of its simplicity as it may not be able to capture all the complexities of language such as sarcasm, irony or ambiguity. The accuracy of the model may also be affected by the contextualisation of words, while performance may not be optimal for large scale application. But in general, it is a convenient and easy-to-use tool for sentiment analysis.