Capstone Project – NLP Applications – Sentimental Analysis

1. Dataset Description

The dataset used was of product reviews from Amazon (https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products)

The dataset was a csv file to be imported into python for our analysis. It contained 34661 rows of data with 21 different fields. My focus was on the 'reviews.text' field that contained the written reviews for the product for me to perform the sentimental analysis of.

2. Preprocessing

Once the dataset had been imported into python I performed some basic preprocessing of the data. This included removing any rows in the column that contained missing or blank values. I also then made sure to remove any stop words from the text before then performing the sentiment analysis of the text.

3. Evaluation of Results

Initially I tried to used the .sentiment, but this wouldn't work with the 'en_core_web_sm' model that I was using with spaCy and would have required further training for it to be able to perform the analysis. Instead I combined it with the 'spacytextblob' that then allowed me to used 'doc._.polarity' to perform the Sentimental Analysis. A value of 1 would have been a very positive review, -1 a very negative review and 0 a neutral review. I test tested this on a couple of reviews and when categorising into positive, neutral or negative it appears to work very well to gauge the sentiment even on such a small model.

I then also compared the similarity of some of the reviews using '.similarity()' and this appeared to give varied results, but again hinting at how reviews were similar if it could match up certain words or tokens.

4. Model Insights – Strengths and Limitations

Strengths:

1. **Efficiency:** Being a small model, it is lightweight and loads quickly. It's great for tasks that require speed and aren't heavily dependent on accuracy.
2. **Versatility:** It supports a wide range of tasks like part-of-speech tagging, named entity recognition, and dependency parsing.
3. **Ease of Use:** The model is easy to load and use for processing text, making it user-friendly for beginners.

Limitations:

1. **Accuracy:** Since it's a small model, it may not be as accurate as its larger counterparts (en_core_web_md or en_core_web_lg) for certain tasks.
2. **Lack of Word Vectors:** This model doesn't include word vectors, meaning it won't be able to leverage semantic similarities between words.
3. **Language-Specific:** As the name suggests, it's specifically trained for English language text. It may not perform well with text in other languages.