Aim

To develop and evaluate a Word2Vec model for analyzing product reviews from the Cell Phones and Accessories category.

Objective

1. To preprocess review text data for training a Word2Vec model.
2. To train a Word2Vec model on product review texts.
3. To evaluate the semantic similarity of various terms based on the trained model.

Summary

This project involves the application of Word2Vec, a popular word embedding technique, to a dataset of product reviews. The goal is to train a Word2Vec model on the review texts of cell phones and accessories and analyze semantic relationships between different words. The dataset is processed and cleaned using Gensim's utilities, and the Word2Vec model is trained and evaluated to understand word similarities and relationships.

Tools and Libraries Used

- **Google Colab**: Platform for running the code.
- **Gensim**: For training the Word2Vec model.
- **pandas**: For data manipulation and processing.
- **python-Levenshtein**: Optional library for optimizing distance calculations (though not used in the provided code).

Procedure

1. **Installation and Importing Libraries**:

CODE:

```
!pip install gensim

!pip install python-Levenshtein

import gensim

import pandas as pd
```

2. **Loading Data**:

CODE:

```
df = pd.read_json("/content/drive/MyDrive/gensim/gensim/Cell_Phones_and_Accessories_5.json",
lines=True)
```

3. **Preprocessing Text**:

CODE:

```
review_text = df.reviewText.apply(gensim.utils.simple_preprocess)
```

4. **Training the Word2Vec Model**:

CODE:

```
model = gensim.models.Word2Vec(
  window=10,
  min_count=2,
  workers=4,
)
model.build_vocab(review_text, progress_per=1000)
model.train(review_text, total_examples=model.corpus_count, epochs=model.epochs)
model.save("/content/drive/MyDrive/gensim/gensim/word2vec-amazon-cell-accessories-reviews-short.model")
```

5. **Evaluating the Model**:

CODE:

```
model.wv.most_similar("bad")
model.wv.similarity(w1="cheap", w2="inexpensive")
model.wv.similarity(w1="great", w2="good")
model.wv.similarity(w1="horrible",w2="awful")
```

Highlights

- **Model Training**: The Word2Vec model is configured with a context window of 10, a minimum word count of 2, and 4 worker threads for parallel processing.
- **Vocabulary Building**: The model builds vocabulary from the processed review texts and trains on this data to create word embeddings.
- **Semantic Analysis**: The model evaluates semantic similarities between different words, showing how well it captures the relationships between words based on the training data.

Conclusion

The project successfully demonstrates the application of Word2Vec for understanding semantic similarities in product reviews. By preprocessing review texts and training the model, it is possible to extract meaningful relationships between different terms. The evaluation of word similarities highlights the effectiveness of the model in capturing semantic meanings, which can be useful for various natural language processing tasks.