TITLE : Word embedding: Implement word embeddings for the English language to analyze word representations and their semantic relationships (Word2Vec/GloVe/fastText).

NAME : Shinde Shubham Dnyandev,

ROLL NO : 23107121,

BATCH : B.

In [1]:
```
import json
import re
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from gensim.models import Word2Vec, FastText
```

In [3]:
```
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /home/admin1/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /home/admin1/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[3]: True

In [5]:
```
file_path = "/home/admin1/Downloads/Cell_Phones_and_Accessories_5.json"

reviews = []
with open(file_path, "r", encoding="utf-8") as f:
    for line in f:
        reviews.append(json.loads(line))

print("Total reviews:", len(reviews))
```

```
Total reviews: 194439
```

In [7]:
```
stop_words = set(stopwords.words("english"))
sentences = []

for r in reviews:
    text = r.get("reviewText", "")
    text = text.lower()
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    text = re.sub(r'\s+', ' ', text).strip()
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]

    if len(tokens) > 3:
        sentences.append(tokens)

print(sentences[0])
```

```
['look', 'good', 'stick', 'good', 'dont', 'like', 'rounded', 'shape', 'always', 'bumpi
ng', 'siri', 'kept', 'popping', 'irritating', 'wont', 'buy', 'product', 'like']
```

In [8]:
```
word2vec_model = Word2Vec(sentences=sentences, vector_size=100, window=5, min_count=5
```

In [11]:
```
word2vec_model.wv.most_similar("battery")
```

Out[11]:
```
[('batter', 0.9141750335693359),
 ('batt', 0.7636744976043701),
 ('batterys', 0.7532073259353638),
 ('batteries', 0.7236005067825317),
 ('batterythe', 0.697971761226654),
 ('batteryi', 0.644496500492096),
 ('batteryit', 0.6087470650672913),
 ('itbattery', 0.5861104726791382),
 ('juice', 0.5802252888679504),
 ('batterythis', 0.5608657002449036)]
```

In [13]:
```
word2vec_model.wv.similarity("movie", "netflix")
```

Out[13]:
```
0.8910148
```

In [15]:
```
fasttext_model = FastText(sentences=sentences, vector_size=100, window=5, min_count=5
```

In [16]:
```
fasttext_model.wv.most_similar("battrey")
```

Out[16]:
```
[('battey', 0.8683770298957825),
 ('batttery', 0.8314153552055359),
 ('battry', 0.8280643224716187),
 ('batteryits', 0.8202927112579346),
 ('batterys', 0.8199542164802551),
 ('batterythese', 0.8185259103775024),
 ('battery9658', 0.8178174495697021),
 ('batteryis', 0.8176612257957458),
 ('batterypack', 0.8168357014656067),
 ('battery3', 0.8153510689735413)]
```

In [19]:
```
fasttext_model.wv.similarity("movie", "netflix")
```

Out[19]:
```
0.82810014
```

In [21]:
```
word2vec_model.save("cellphones_word2vec.model")
fasttext_model.save("cellphones_fasttext.model")
```