WordEmbeddingsPart1 Assignment

Yuhuan Huang, Aug 2025

- Console outputs:

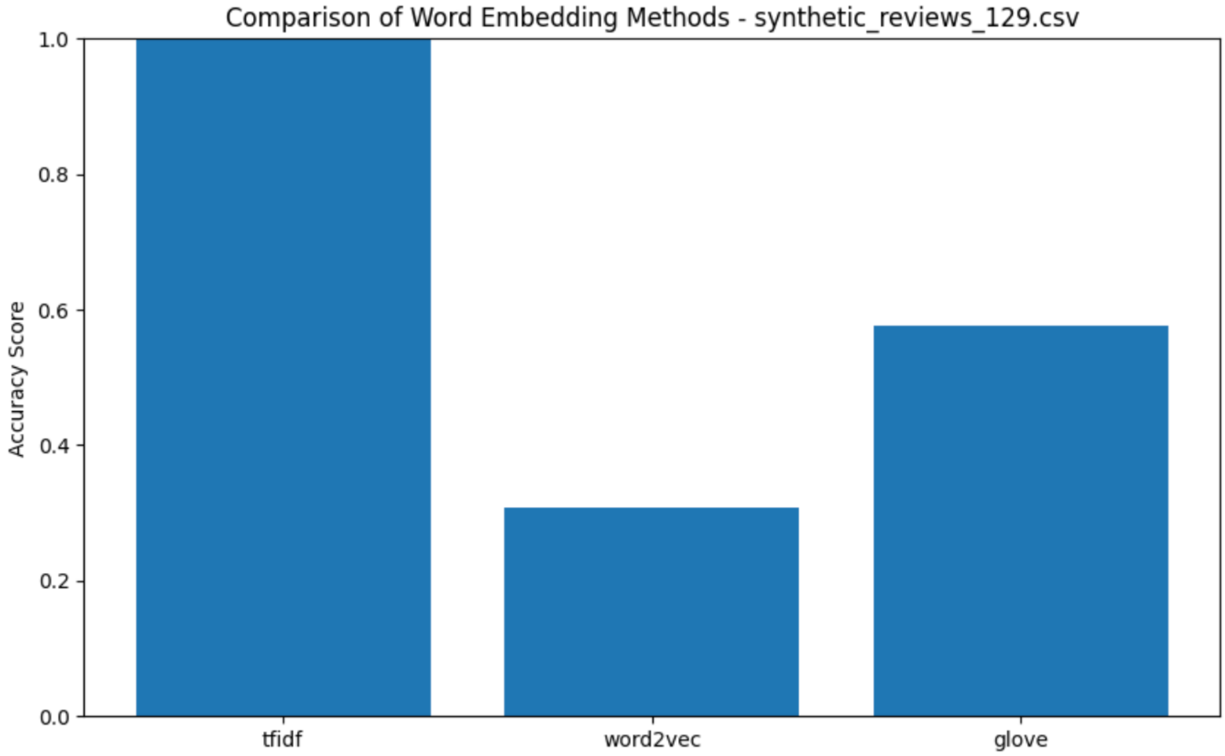
```
[ ] #Run this cell to install gensim and restart the runtime
    !pip install gensim --quiet
    import os
    os.kill(os.getpid(), 9)  # Forces a runtime restart
```

```
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.0/61.0 kB 3.2 MB/s eta 0:00:00
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.6/60.6 kB 2.5 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 26.6/26.6 MB 79.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 18.0/18.0 MB 93.7 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 38.2/38.2 MB 17.4 MB/s eta 0:00:00
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This beha
opencv-python-headless 4.12.0.88 requires numpy<2.3.0,>=2; python_version >= "3.9", but you have numpy 1.26.4 which
opencv-contrib-python 4.12.0.88 requires numpy<2.3.0,>=2; python_version >= "3.9", but you have numpy 1.26.4 which i
opencv-python 4.12.0.88 requires numpy<2.3.0,>=2; python_version >= "3.9", but you have numpy 1.26.4 which is incomp
thinc 8.3.6 requires numpy<3.0.0,>=2.0.0, but you have numpy 1.26.4 which is incompatible.
tsfresh 0.21.0 requires scipy>=1.14.0; python_version >= "3.10", but you have scipy 1.13.1 which is incompatible.
```

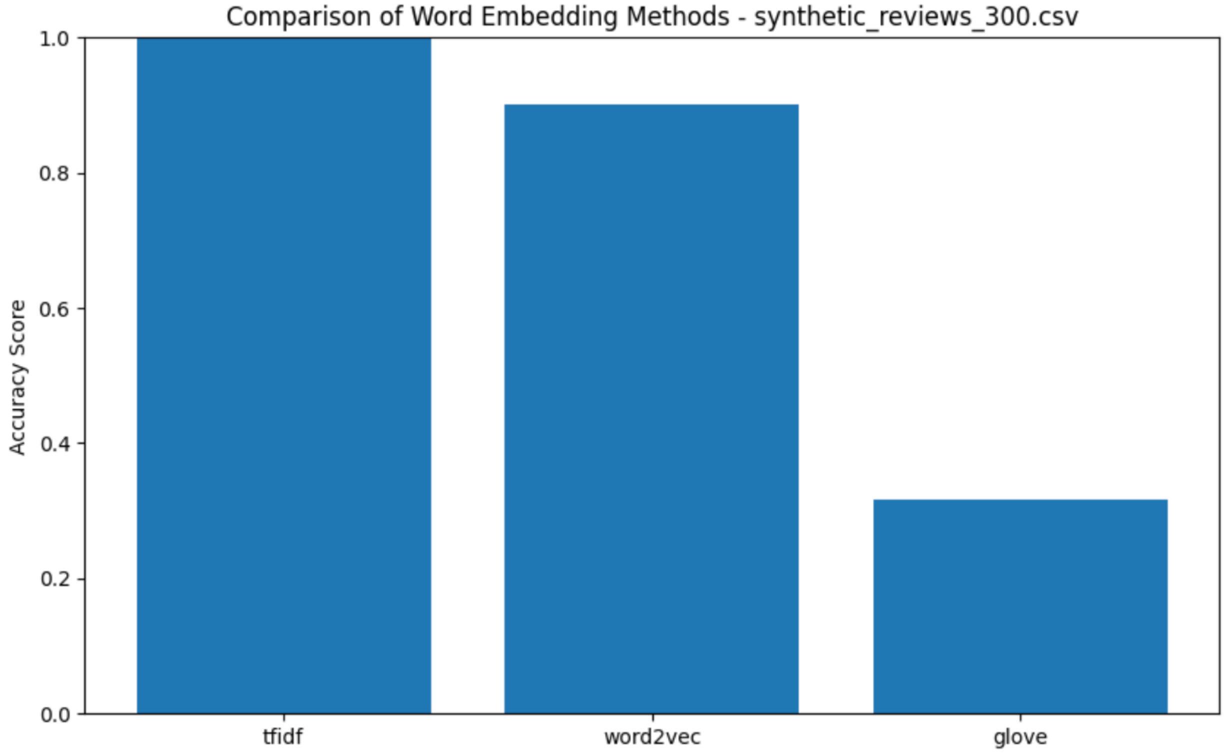
I encountered some problems here of the numpy version, but I am still able to successfully run the following cells.

1.Example Usage and Visualization - Synthetic Data

```
Processing synthetic_reviews_129.csv
Loaded 129 reviews
Sentiment distribution:
sentiment
 0    43
-1    43
 1    43
Name: count, dtype: int64
Generating embeddings...
TFIDF accuracy: 1.00
WORD2VEC accuracy: 0.31
GLOVE accuracy: 0.58
```



```
Processing synthetic_reviews_300.csv
Loaded 300 reviews
Sentiment distribution:
sentiment
 1    100
 0    100
-1    100
Name: count, dtype: int64
Generating embeddings...
TFIDF accuracy: 1.00
WORD2VEC accuracy: 0.90
GLOVE accuracy: 0.32
```




```
Accuracies over all test files TFIDF :  [1. 1.]
Accuracies over all test files WORD2VEC :  [0.30769231 0.9        ]
Accuracies over all test files GLOVE :  [0.57692308 0.31666667]
Average accuracies:  [1.         0.60384615 0.44679487]
TestResults(failed=0, attempted=4)
```

2.Example Predictions - Synthetic Data

These are the reviews I created for testing:

```
example_reviews = [
    "This product is fantastic, works exactly as described!",
    "I hate it.  terrible design!",
    "Don't buy this shit.",
    "I really enjoy using the product. Would definitely recommend to my friends!",
    "Not as good as described, but still works.",
    "Total waste of money.",
    "It was terrible, and it ran out of battery so quickly.",
    "Great product, good price.",
    "Not a wise choice.",
    "It is easy to use, but the material looks cheap.",
    "Awesome! It is such a lovely dress.",
    "It is fine considering its price, but I would not buy it again."
]
```

and the results are:

```
Loaded 129 reviews

Sentiment distribution:
sentiment
 0    43
-1    43
 1    43
Name: count, dtype: int64
Generating embeddings...

Example Predictions:

Review: This product is fantastic, works exactly as described!
TF-IDF prediction: 1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: 1 (1=positive, 0=neutral, -1=negative)

Review: I hate it.  terrible design!
TF-IDF prediction: -1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: -1 (1=positive, 0=neutral, -1=negative)

Review: Don't buy this shit.
TF-IDF prediction: -1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: -1 (1=positive, 0=neutral, -1=negative)

Review: I really enjoy using the product. Would definitely recommend to my friends!
TF-IDF prediction: 1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: 1 (1=positive, 0=neutral, -1=negative)

Review: Not as good as described, but still works.
TF-IDF prediction: 0 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: -1 (1=positive, 0=neutral, -1=negative)

Review: Total waste of money.
TF-IDF prediction: -1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: 1 (1=positive, 0=neutral, -1=negative)
```

```
Review: Total waste of money.
TF-IDF prediction: -1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: 1 (1=positive, 0=neutral, -1=negative)

Review: It was terrible, and it ran out of battery so quickly.
TF-IDF prediction: -1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: -1 (1=positive, 0=neutral, -1=negative)

Review: Great product, good price.
TF-IDF prediction: 1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: 0 (1=positive, 0=neutral, -1=negative)

Review: Not a wise choice.
TF-IDF prediction: -1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: -1 (1=positive, 0=neutral, -1=negative)

Review: It is easy to use, but the material looks cheap.
TF-IDF prediction: 0 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: -1 (1=positive, 0=neutral, -1=negative)

Review: Awesome! It is such a lovely dress.
TF-IDF prediction: -1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: -1 (1=positive, 0=neutral, -1=negative)

Review: It is fine considering its price, but I would not buy it again.
TF-IDF prediction: -1 (1=positive, 0=neutral, -1=negative)
Word2Vec prediction: -1 (1=positive, 0=neutral, -1=negative)
GloVe prediction: -1 (1=positive, 0=neutral, -1=negative)
```

Reflection:

Answer the following questions:

1. Implementation Analysis: Compare the sentiment analysis results using TF-IDF, Word2Vec, and GloVe embeddings in your implementation. Which performed better for the 'synthetic_reviews_129.csv' product review dataset and why do you think that was the case? Support your answer with specific examples from your results.

   My Ans: For the 129 review, I think TF-IDF is the better method, since from the bar chart we can see it has the highest accuracy score.

2. Compare the sentiment analysis results for the 2 different review files provided ('synthetic_reviews_129.csv', 'synthetic_reviews_300.csv'). Which embeddings technique consistently performed best and which performed worst. Provide

possible reasons why.

My Ans: Comparint the 129 review and the 300 review, I think TF-TDF consistently perfrom better, while Glove can be considered as the worst model.

3. Technical Understanding: Suppose you encountered a new product review containing words not present in your training vocabulary. Explain how each embedding method (TF-IDF, Word2Vec, GloVe) handles out-of-vocabulary words, and propose a strategy to improve handling of such cases.

   My Ans: For the TF-TDF method, it just simply ignores the oov words; for the Word2Vec and Glove, we can assign the unknown words to a separare group: 'UNK' or 'OOV'.

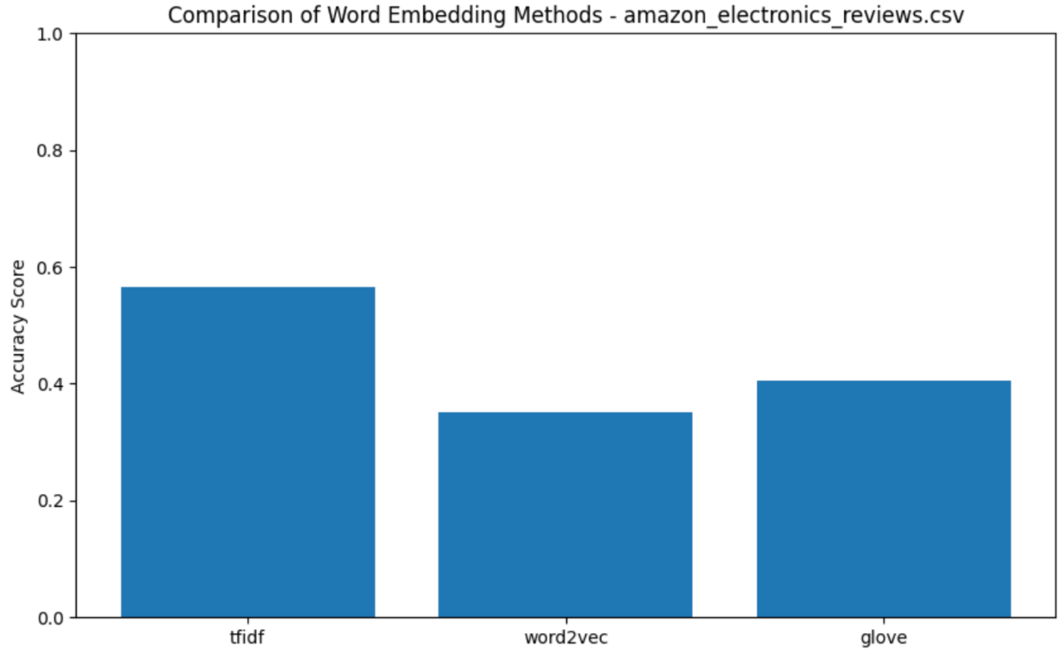
4. Practical Application: Suppose your model shows high accuracy but poor real-world performance on certain product categories. How could you use the word embedding visualizations to diagnose potential biases or gaps in your model's understanding of product-specific terminology? Include specific visualization techniques in your answer.

My Ans: Maybe it is because the embeddings does not perform well in the new scenario. We may use a heatmap to see the cosine similarity and check whether the embeddings follow the reasonable categories.

3. Assignment Extension

Console output:

```
Processing amazon_electronics_reviews.csv
Loaded 1251 reviews
Sentiment distribution:
sentiment
 0    417
 1    417
-1    417
Name: count, dtype: int64
Generating embeddings...
TFIDF accuracy: 0.57
WORD2VEC accuracy: 0.35
GLOVE accuracy: 0.41
```




```
Accuracies over all test files TFIDF :  [0.56573705]
Accuracies over all test files WORD2VEC :  [0.35059761]
Accuracies over all test files GLOVE :  [0.4063745]
Average accuracies:  [0.56573705 0.35059761 0.4063745 ]
```

Reflection:

Answer the following questions:

1. Compare the sentiment analysis results using TF-IDF, Word2Vec, and GloVe embeddings for a real dataset. Which performed best for the dataset you used and why do you think that was the case? Support your answer with specific examples from your results.

My Ans: From the graphs, RF-TDF still perform (accuracy: 0.5657) better than the remaining models. RF-TDF has advantage in small-scale and sparse texts, such as reviews; while Word2Vec and Glove may be more suitable in larger data or richer sentimental scenario.

2. Why do you think the results for the real dataset differed from the synthetic one used in the previous cell?

My Ans:I think in the synthetics one, some reviews are duplicate, and also real-world customers may not write reviews like the synthetic ones.