Name: Jaimis Arvindbhai Miyani
Student Id: 400551743

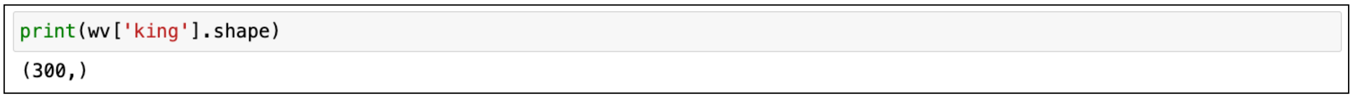
SEP 775 - Introduction to Computational Natural Language Processing
Assignment 1 - In-Depth Word Vectors Analysis

Objectives: This assignment focuses on a comprehensive understanding of word vector technologies, specifically Word2Vec and GloVe. You will explore their applications, visualize the results, and analyze the semantic and syntactic relationships they capture.

1. Building and Analyzing Word Vectors with Word2Vec

For this task, I have used the pre-trained "word2vec-google-news-300" Word2Vec model. This Pre-trained vectors trained on a part of the Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases.
Link to pre-trained model: Hugging face

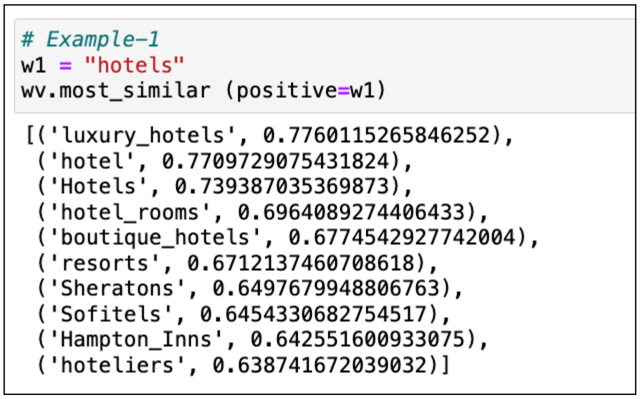
Each word is represented by vector of dimension 300. Thus if we print shape of vector it prints (300,).



```
print(wv['king'].shape)
```

```
(300,)
```

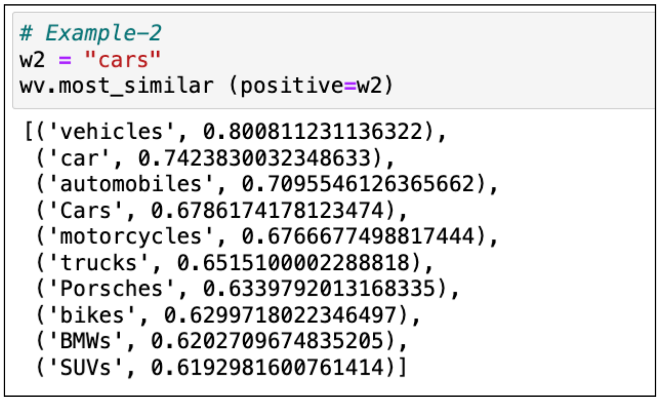
- **Checking similarity of some words over pre-trained model:**



```
# Example-1
w1 = "hotels"
wv.most_similar (positive=w1)
```

```
[('luxury_hotels', 0.7760115265846252),
 ('hotel', 0.7709729075431824),
 ('Hotels', 0.739387035369873),
 ('hotel_rooms', 0.6964089274406433),
 ('boutique_hotels', 0.6774542927742004),
 ('resorts', 0.6712137460708618),
 ('Sheratons', 0.6497679948806763),
 ('Sofitels', 0.6454330682754517),
 ('Hampton_Inns', 0.642551600933075),
 ('hoteliers', 0.638741672039032)]
```

Most similar word for hotels from pre-trained model is **luxury_hotels**



```
# Example-2
w2 = "cars"
wv.most_similar (positive=w2)
```

```
[('vehicles', 0.800811231136322),
 ('car', 0.7423830032348633),
 ('automobiles', 0.7095546126365662),
 ('Cars', 0.6786174178123474),
 ('motorcycles', 0.6766677498817444),
 ('trucks', 0.6515100002288818),
 ('Porsches', 0.6339792013168335),
 ('bikes', 0.6299718022346497),
 ('BMWs', 0.6202709674835205),
 ('SUVs', 0.6192981600761414)]
```

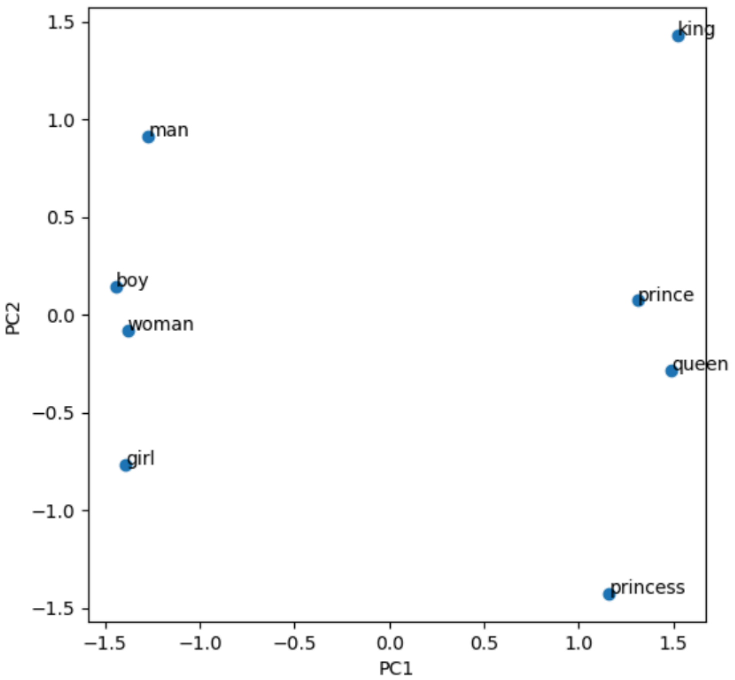
Most similar word for cars from pre-trained model is **vehicles**

- **Visualization of word vectors in 2D using PCA**:

Given that each word vector comprises 300 dimensions representing the meaning of a word, it poses a challenge for visualization as humans can typically only perceive up to three dimensions. While one approach could involve plotting only the first two dimensions or those with the most significant variability, this would neglect much of the information encapsulated in the other dimensions or variables. Alternatively, we can employ Principal Component Analysis (PCA), a technique for reducing dimensionality, to address this issue.

Principal Component Analysis (PCA) is a method for transforming data by creating new variables, known as components, through linear combinations of the original variables in a dataset matrix containing N observations and M variables. Thus using PCA we reduce dimensionality of each vector to 2 dimension so that we can plot and visualize vectors.

To visualize vectors I have extracted some word vectors from the pre-trained Word2Vec model. This words are [“man", "woman", "boy", "girl", "king", "queen", "prince", “princess”]. As I have extracted 8 word vectors the size of my created vector will be (8, 300). So now If we plot this word vectors on scatter plot using reduced dimension (PC1, PC2) it looks as shown below.



- **Word relationship and cluster formed:**

The "word relationship" aspect refers to how words are positioned relative to each other in this reduced dimensional space. Words that are semantically similar or related are likely to be closer to each other in this space. For example, words like "king" and "man" might be closer together, indicating their semantic relationship as being related to gender.

"Cluster" in this context refers to groups of words that are close to each other in the reduced dimensional space. These clusters often represent semantic groupings or categories of words. For instance, you might have a cluster containing words related to animals, another cluster containing words related to food, and so on.

From the scatter plot, we can observe that two clusters have formed based on human demographics and royalty. One cluster comprises word vectors for "man," "boy," "woman," and "girl," while the other includes vectors for "king," "prince," "queen," and “princess."

Additionally, we observe in the first cluster that "woman" and "girl" are plotted closely together, while "man" and "boy" are also in close proximity. This suggests that they are plotted based on gender relationships. This relationship holds true for the other cluster as well.

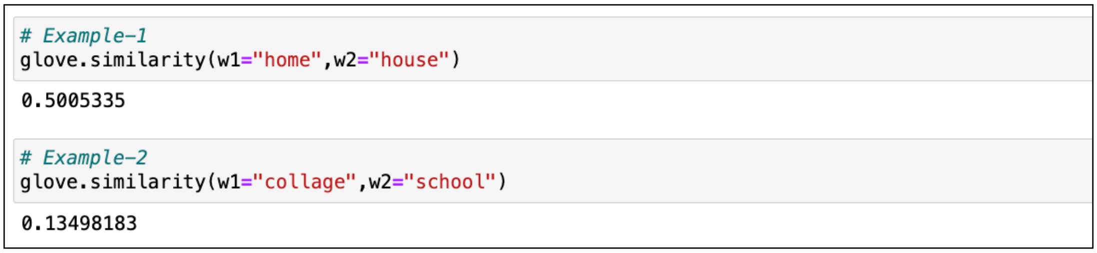
Furthermore, when examining the relationship between clusters, we notice that "girl" is closely plotted with "princess," and "woman" is closely plotted with "queen." Similarly, "boy" is closely associated with "prince," and "man" is closely associated with “king."

2. GloVe Vectors Advanced Analysis

For this task, I have used the pre-trained “glove-wiki-gigaword-300” GloVe model. This pre-trained glove vectors are based on 2B tweets, 27B tokens, 1.2M vocab, uncased.
Link to pre-trained model: Hugging face

Each word in pre-trained model is represented by 300 dimensions.

- **Checking similarity between two words over pre-trained model:**



```
# Example-1
glove.similarity(w1="home",w2="house")
```

```
0.5005335
```

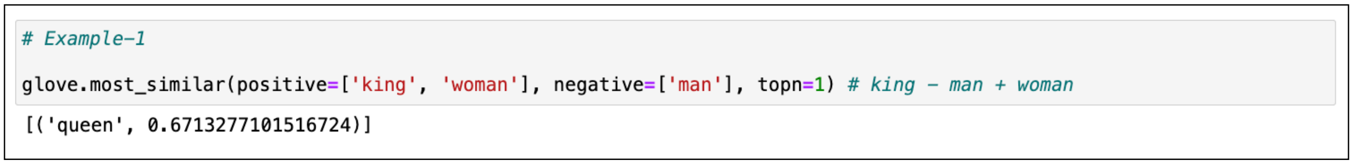
```
# Example-2
glove.similarity(w1="collage",w2="school")
```

```
0.13498183
```

As depicted in the image above, the similarity between the words "home" and "house" in the pre-trained model is 0.5005335, while the similarity between "college" and "school" is 0.13498183.

- **An analogy task:**

1) king - man + woman



```
# Example-1
glove.most_similar(positive=['king', 'woman'], negative=['man'], topn=1) # king - man + woman
[('queen', 0.6713277101516724)]
```

Word 1: King
Word 2: Man
Word 3: Woman

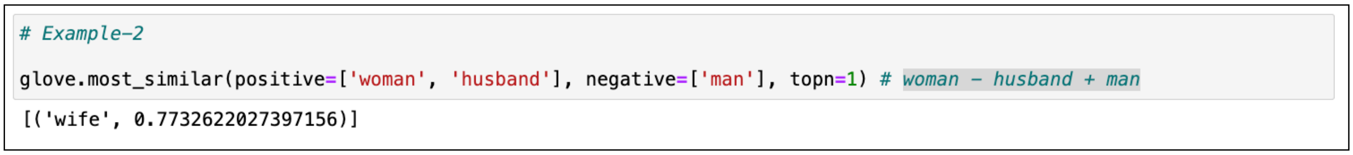
To solve this analogy, we need to find the vector that represents the relationship between "Man" and "Woman" and apply it to "King". Mathematically, we compute:

Result = Vector(King) - Vector(Man) + Vector(Woman)

The GloVe vectors encode semantic relationships between words based on the co-occurrence statistics of words in large text corpora. When we perform the analogy tasks, we are essentially exploring these relationships.

In this analogy task, the relationship between "King" and "Man" is that of gender, and the relationship between "King" and "Woman" is expected to be the same, but in the opposite direction. Therefore, we expect to find a word representing the female equivalent of "King", which is “Queen".

2) woman - husband + man



```
# Example-2
glove.most_similar(positive=['woman', 'husband'], negative=['man'], topn=1) # woman - husband + man
[('wife', 0.7732622027397156)]
```

Word 1: woman
Word 2: husband
Word 3: man

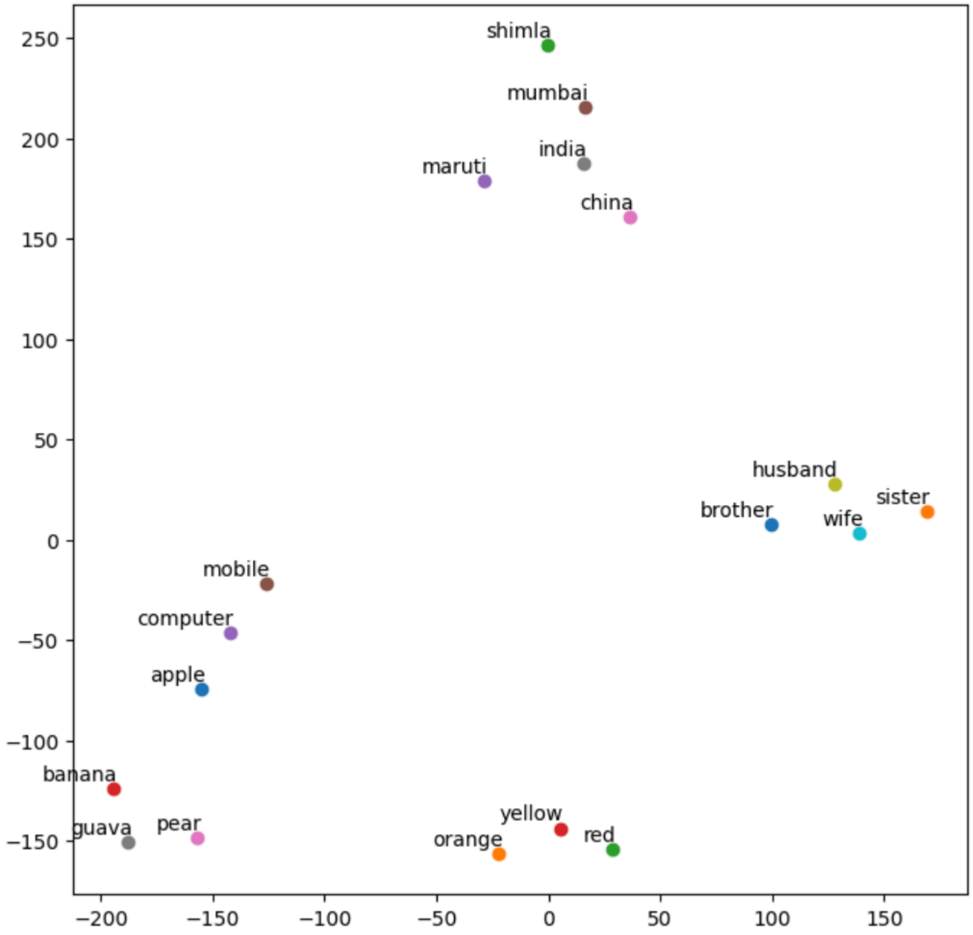
Same as above example to solve this analogy, we need to find the vector that represents the relationship between "Husband" and "Man" and apply it to “Woman”.

Thus In the analogy task, we're exploring the relationship between countries and their capitals. "Paris" is the capital of "France", and "Berlin" is the capital of "Germany". By applying the same relationship between "France" and "Germany" to "Paris", we can approximate another capital city in Germany.

- **Visualizing word vectors in 2D using TSNE**

t-SNE is a dimensionality reduction technique commonly used for visualizing high-dimensional data in a lower-dimensional space, often 2D or 3D. t-SNE is particularly effective for visualizing complex datasets by preserving local similarities or relationships between data points.

To visualize vectors I have extracted some word vectors from the pre-trained GloVe model. This words are [“apple", "orange", "shimla", "banana", "maruti", "china", "india", “husband”, “wife", "brother", "sister", "red", "yellow", "computer", "mobile", “pear”, “guava”]. As I have extracted 17 word vectors the size of my created vector will be (17, 300). So now If we plot this word vectors on scatter plot using reduced dimension (using TSNE) it looks as shown below.



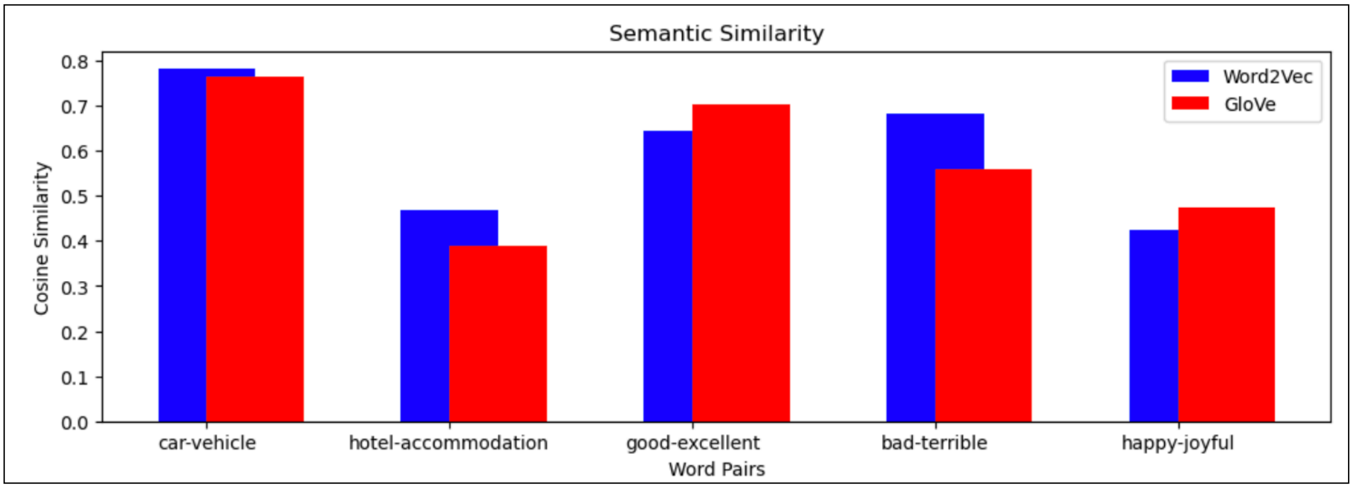
In the scatter plot above, words are grouped into five distinct clusters based on categories such as fruits, cities/countries, colors, human relations, and devices. However, there are instances where a word belongs to more than one category. For example, "apple" is both a fruit and a device. Therefore, the word "apple" is plotted in the device cluster, but it is also positioned near the fruit cluster. Similarly, "orange" is both a color and a fruit, so it appears in the colors cluster but is located close to the fruit cluster.

3. Semantic and Syntactic Word Relationships

- **Comparing Word2Vec and GloVe for semantic word relationships**

When comparing the similarity between word pairs using both pre-trained Word2Vec and GloVe models, we observe the following similarity values for the word pairs: ["car-vehicle", "hotel-accommodation", "good-excellent", "bad-terrible", “happy-joyful”]:

- For Word2Vec model: [0.7821097, 0.46953395, 0.6442929, 0.68286115, 0.42381963]
- For GloVe model: [0.76553667, 0.38898456, 0.70394963, 0.5597251, 0.47513184]



- **Comparing Word2Vec and GloVe for syntactic word relationships**

When conducting syntactic relationship tasks for both Word2Vec and GloVe models, the results for the analogy tasks are as follows:

- "woman - man + king" yields ['queen'] for Word2Vec and ['queen'] for GloVe.
- "woman - husband + man" yields ['teenage_girl'] for Word2Vec and ['girl'] for GloVe.
- "walking - walk + swim" yields ['swimming'] for both Word2Vec and GloVe.
- "bigger - big + small" yields ['larger'] for both Word2Vec and GloVe.
- "ate - eat + go" yields ['went'] for both Word2Vec and GloVe.
- "faster - fast + slow" yields ['slower'] for both Word2Vec and GloVe.