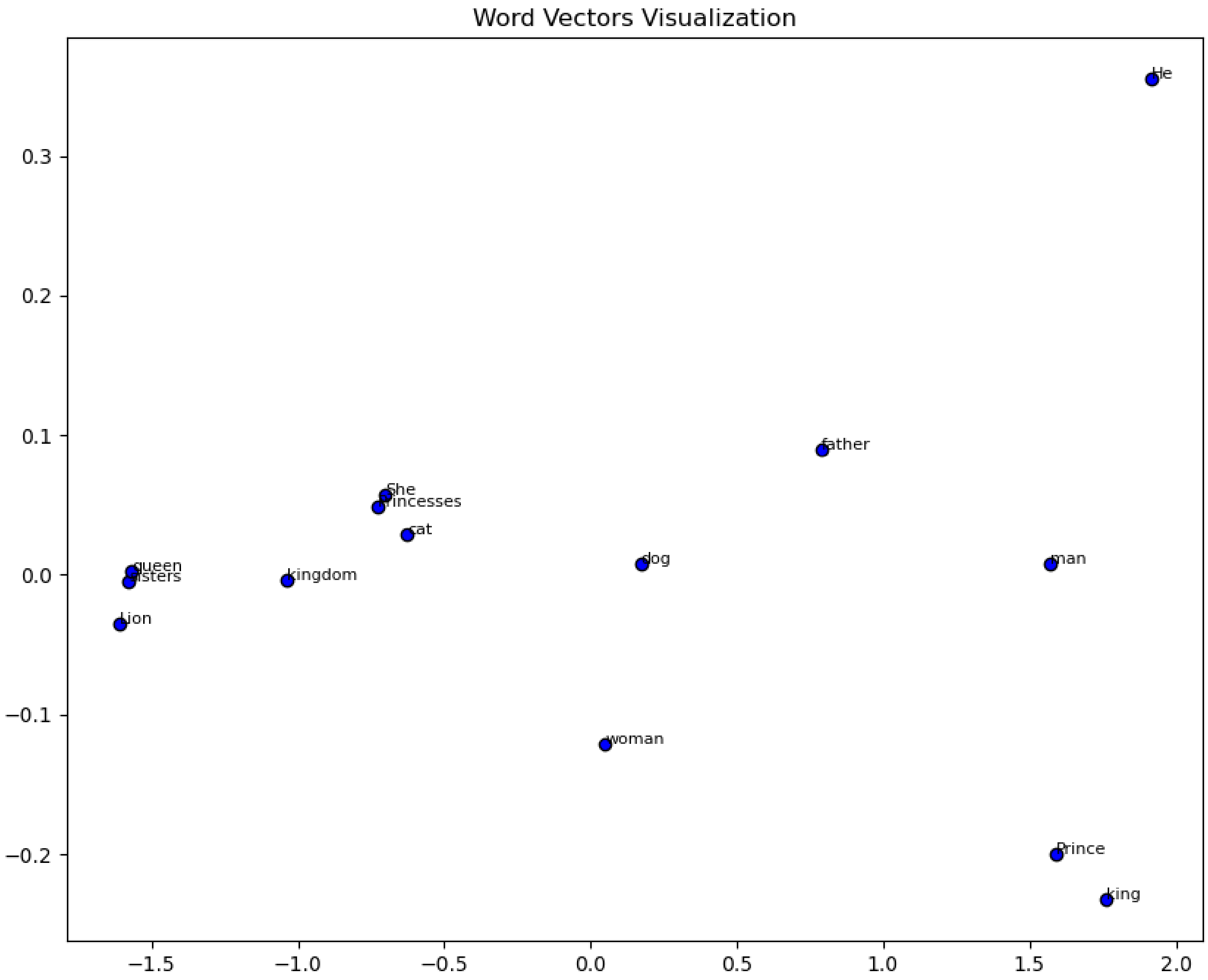
Assignment 1

Q 1. Building and Analyzing Word Vectors with Word2Vec

(Analysis: Discuss the word relationships and clusters formed.)

Ans:



- The above image shows the representation of vectors of some words in 2-D space (obtained using word2vec model and reduced dimensions of vectors using PCA). For this task, I used gensim library to train word2vec model ("myword2vec.model") on a specific corpus of words that was prepared from a textbook (web-version) available on open-source website (https://www.gutenberg.org/)

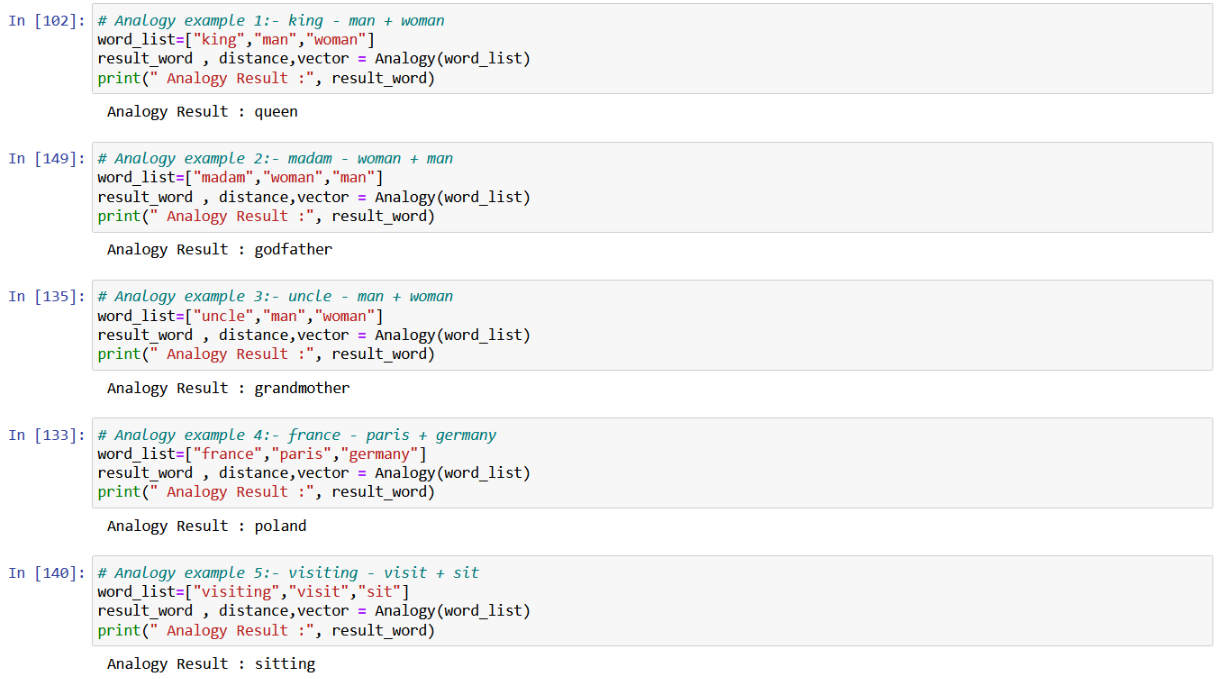
- from the image, it can be observed that words such as "queen", "sister"," she"," princess" are very close to another and also belongs to female gender type. These words create a particular cluster where almost all words relate to a particular gender type.

- Apart from this, “father”, “man”, “king and “prince” are also very close to each other and create a small cluster at bottom right corner. These words belongs to male gender type.

Q 2. GloVe Vectors Advanced Analysis

(Explanation: Explain the results of the analogy tasks.)

Ans:



```
In [102]: # Analogy example 1:- king - man + woman
          word_list=["king","man","woman"]
          result_word , distance,vector = Analogy(word_list)
          print(" Analogy Result :", result_word)

           Analogy Result : queen

In [149]: # Analogy example 2:- madam - woman + man
          word_list=["madam","woman","man"]
          result_word , distance,vector = Analogy(word_list)
          print(" Analogy Result :", result_word)

           Analogy Result : godfather

In [135]: # Analogy example 3:- uncle - man + woman
          word_list=["uncle","man","woman"]
          result_word , distance,vector = Analogy(word_list)
          print(" Analogy Result :", result_word)

           Analogy Result : grandmother

In [133]: # Analogy example 4:- france - paris + germany
          word_list=["france","paris","germany"]
          result_word , distance,vector = Analogy(word_list)
          print(" Analogy Result :", result_word)

           Analogy Result : poland

In [140]: # Analogy example 5:- visiting - visit + sit
          word_list=["visiting","visit","sit"]
          result_word , distance,vector = Analogy(word_list)
          print(" Analogy Result :", result_word)

           Analogy Result : sitting
```

- For analogy task, I used a pretrained glove model (one of the words to vectors conversion models). It is a count-based (word to word co-occurrence) model.

- Analogy tasks (performed using "glove.6B.model" - Trained on 6 billions tokens) helps to evaluate quality and structure of any kind of word embedding (word to vector) model. These tasks are based on the idea that word vectors can show linear relationships that correspond to semantic or syntactic analogies.

  - For example, if we have a word vector for "king", we can subtract the vector for "man" and add the vector for "woman" to get a vector that is close to the vector for "queen". This is because the word vectors capture the gender difference between "king" and "queen" as well as their semantic similarity

    e.g. : king – women + man =~ queen

  - Similarly, if we have a word vector for "visiting", we can subtract the vector for "visit" and add the vector of "sit" to get the vector which is close to the vector of word "sitting". It means that word vector can also capture the grammatical relation between "visiting" and "sitting" as well as their syntactic similarities.

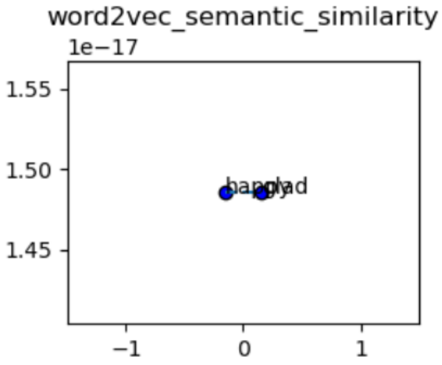
    e.g. visiting – visit + sit =~ sitting

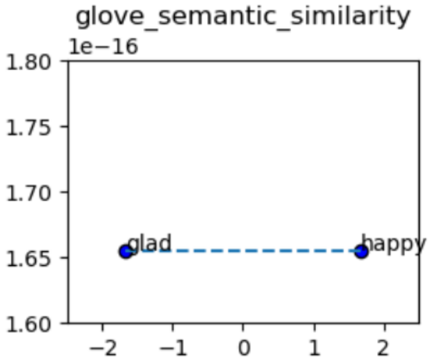
- These results demonstrate that GloVe vectors can capture complex linguistic patterns and similarities between words in a lower-dimensional vector space.

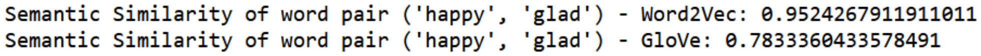
Q 3. Semantic and Syntactic Word Relationships

Ans:

Semantic Similarity Result :-





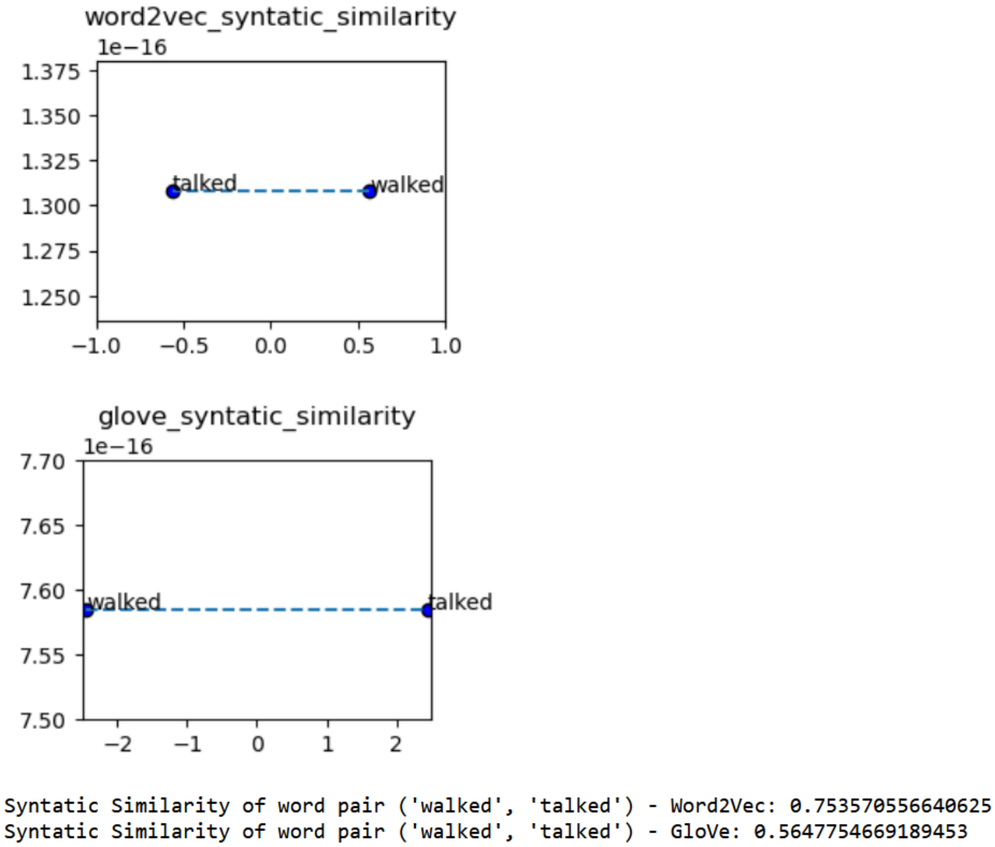


```
Semantic Similarity of word pair ('happy', 'glad') - Word2Vec: 0.9524267911911011
Semantic Similarity of word pair ('happy', 'glad') - GloVe: 0.7833360433578491
```

- From the result, it can be observed that the word2vec word embedding method is good in capturing semantic similarity between "Happy" and "Glad". If I represent the vectors obtained using the word2vec method into 2D space, then they are more closer than that calculated using GloVe model.

- Overall, it can not be said that "word2vec" is good in capturing semantic similarity in comparison with "GloVe" at all the times. It requires testing of more such examples and take average of all observation to make final decision.

Syntactic Similarity Result :-



```
Syntatic Similarity of word pair ('walked', 'talked') - Word2Vec: 0.753570556640625
Syntatic Similarity of word pair ('walked', 'talked') - GloVe: 0.5647754669189453
```

- Similar to the semantic similarity, word2vec performed better in capturing syntactic similarity between “walked” and “Talked”. To decide which one is better in capturing syntactic similarity in general, both models need to be tested with more examples.