READ ME

**Instructions**

**1. Install Dependencies**

Ensure you have Python 3.x installed. Then, install the required libraries using the following command:

bash

Copy code

pip install gensim

**2. Load Pre-trained Models**

The project uses **FastText** and **Word2Vec** models from Gensim’s API. Load the models as shown below:

python

Copy code

import gensim.downloader as api

# Load FastText and Word2Vec models

fasttext\_model = api.load('fasttext-wiki-news-subwords-300')

word2vec\_model = api.load('word2vec-google-news-300')

**3. Find Word Analogies**

To find word analogies, use the find\_analogies() function. Provide three words in the analogy form "A : B :: C : ?":

python

Copy code

def find\_analogies(model, word\_a, word\_b, word\_c):

result = model.most\_similar(positive=[word\_b, word\_c], negative=[word\_a], topn=1)

return result[0][0]

Example:

python

Copy code

result = find\_analogies(fasttext\_model, 'king', 'man', 'woman')

print(result) # Output: 'queen'

**4. Interactive Testing**

You can test the word analogy function interactively by running:

python

Copy code

def interactive\_testing():

while True:

word\_a = input("Enter word A (or 'exit' to quit): ")

if word\_a.lower() == 'exit':

break

word\_b = input("Enter word B: ")

word\_c = input("Enter word C: ")

result\_ft = find\_analogies(fasttext\_model, word\_a, word\_b, word\_c)

print(f"FastText result: {result\_ft}")

result\_w2v = find\_analogies(word2vec\_model, word\_a, word\_b, word\_c)

print(f"Word2Vec result: {result\_w2v}")

This allows users to input any word analogy and get predictions in real-time.