

```

file_path = "CBOW.txt"
with open(file_path, 'r') as file:
    file_contents = file.read()
file_contents

import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
sentences = file_contents.split('.')
tokenizer = Tokenizer()
tokenizer.fit_on_texts(sentences)
total_words = len(tokenizer.word_index) + 1
window_size = 3
tokenized_sentences = tokenizer.texts_to_sequences(sentences)
data = pad_sequences([[sentence[j] for j in range(i - window_size, i +
window_size + 1) if j != i and 0 <= j < len(sentence)]]
for sentence in tokenized_sentences
for i, _ in enumerate(sentence)])
labels = np.array([target_word for sentence in tokenized_sentences for
target_word in sentence])
model = models.Sequential([
    layers.Embedding(input_dim=total_words, output_dim=50,
input_length=window_size * 2),
    layers.GlobalAveragePooling1D(),
    layers.Dense(total_words, activation='softmax')
])
model.compile(optimizer='adam',

```

```

loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train the model

model.fit(data, labels, epochs=200)

word_embeddings = model.layers[0].get_weights()[0]

from sklearn.metrics.pairwise import cosine_similarity

target_word = 'influenza'

target_embedding =

word_embeddings[tokenzier.word_index[target_word]]

similarities = cosine_similarity(target_embedding.reshape(1, -1),

word_embeddings)[0]

most_similar_indices = similarities.argsort()[-5:][::-1]

most_similar_words = [word for word, idx in

tokenizer.word_index.items() if idx in most_similar_indices]

print(f"Most similar words to '{target_word}': {most_similar_words}")

```