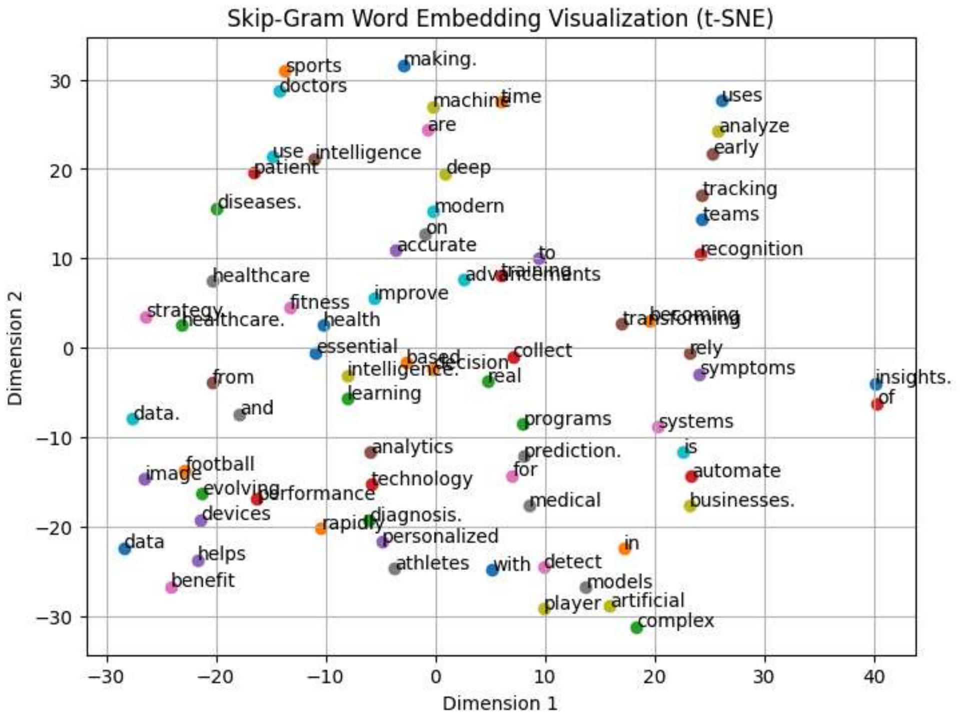
```python
# Install and Import Required Libraries
!pip install gensim
import gensim
from gensim.models import Word2Vec
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
import numpy as np

# -------- NEW CORPUS --------
sentences = [
    "Technology is evolving rapidly with advancements in artificial intelligence.",
    "Healthcare systems rely on accurate data for patient diagnosis.",
    "Sports analytics uses data to improve player performance and strategy.",
    "Artificial intelligence helps automate complex decision making.",
    "Doctors analyze medical data to detect early symptoms of diseases.",
    "Football teams use machine learning for performance prediction.",
    "Fitness tracking devices collect real time health data.",
    "Deep learning models are transforming image recognition in healthcare.",
    "Data analytics is becoming essential in modern businesses.",
    "Athletes benefit from personalized training programs based on data insights."
]

# Tokenize
tokenized_sentences = [sentence.lower().split() for sentence in sentences]

# Train Skip-Gram Model
skipgram_model = Word2Vec(
    sentences=tokenized_sentences,
    vector_size=100,
    window=3,
    min_count=1,
    sg=1,
    epochs=100
)

# Inspect Word Vectors
word = "data"
vector = skipgram_model.wv[word]
print(f"Vector size for '{word}':", len(vector))

# Similarity
similarity_score = skipgram_model.wv.similarity("data", "health")
print("Similarity between 'data' and 'health':", similarity_score)

# Most Similar Words
similar_words = skipgram_model.wv.most_similar("data", topn=5)
print("Words similar to 'data':")
for word, score in similar_words:
    print(word, ":", score)

# ---- UPDATED PAIRS ----
pairs = [
    ("data", "health"),
    ("machine", "learning"),
    ("sports", "analytics")
]

scores = [skipgram_model.wv.similarity(w1, w2) for w1, w2 in pairs]
print("Average Similarity Score:", np.mean(scores))

# Visualization using t-SNE
words = list(skipgram_model.wv.index_to_key)
vectors = np.array([skipgram_model.wv[word] for word in words])

tsne = TSNE(n_components=2, random_state=42, perplexity=5)
reduced_vectors = tsne.fit_transform(vectors)

# Plot Word Embeddings
plt.figure(figsize=(8, 6))

for i, word in enumerate(words):
    plt.scatter(reduced_vectors[i, 0], reduced_vectors[i, 1])
    plt.annotate(word, (reduced_vectors[i, 0], reduced_vectors[i, 1]))

plt.title("Skip-Gram Word Embedding Visualization (t-SNE)")
plt.xlabel("Dimension 1")
plt.ylabel("Dimension 2")
plt.grid(True)
plt.show()
```

```
Requirement already satisfied: gensim in /usr/local/lib/python3.12/dist-packages (4.4.0)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.12/dist-packages (from gensim) (2.0.2)
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.12/dist-packages (from gensim) (1.16.3)
Requirement already satisfied: smart_open>=1.8.1 in /usr/local/lib/python3.12/dist-packages (from gensim) (7.5.0)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart_open>=1.8.1->gensim) (2.0.1)
Vector size for 'data': 100
Similarity between 'data' and 'health': 0.48025241
Words similar to 'data':
benefit : 0.5446097254753113
evolving : 0.5314252972602844
for : 0.5287097096443176
to : 0.5243873000144958
on : 0.4924267530441284
Average Similarity Score: 0.33346006
```


Skip-Gram Word Embedding Visualization (t-SNE)
Dimension 1
Dimension 2
sports
doctors
making.
machine
time
are
uses
analyze
early
use
intelligence
patient
deep
diseases.
modern
on
accurate
tracking
teams
recognition
to
training
advancements
healthcare
improve
fitness
strategy
healthcare.
health
transforming
becoming
essential
based
decision
intelligence.
collect
real
rely
symptoms
insights.
of
from
learning
data.
and
programs
systems
analytics
prediction.
is
football
image
for
automate
evolving
performance
technology
medical
businesses.
devices
diagnosis.
rapidly
personalized
in
data
helps
athletes
with
detect
models
benefit
player
artificial
complex