```python
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pandas as pd

# Sample dataset of items with descriptions
data = {
    'title': [
        'The Matrix',
        'Inception',
        'Interstellar',
        'The Prestige',
        'The Dark Knight',
        'Memento',
        'Tenet'
    ],
    'description': [
        'A hacker discovers reality is a simulation and joins a rebellion.',
        'A thief steals corporate secrets through dream-sharing technology.',
        'Explorers travel through a wormhole in space to save humanity.',
        'Two magicians engage in a bitter rivalry with deadly consequences.',
        'A vigilante in Gotham fights crime and faces his nemesis Joker.',
        'A man with short-term memory loss uses notes and tattoos to hunt a killer.',
        'A secret agent manipulates time to prevent World War III.'
    ]
}

# Create DataFrame
df = pd.DataFrame(data)

# Vectorize the text using TF-IDF
vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = vectorizer.fit_transform(df['description'])

# Compute cosine similarity
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)

# Function to recommend similar items
def recommend(title, df=df, cosine_sim=cosine_sim):
    if title not in df['title'].values:
        return "Title not found in the dataset."

    idx = df.index[df['title'] == title][0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:4]
    movie_indices = [i[0] for i in sim_scores]

    return df['title'].iloc[movie_indices].tolist()

# Example usage
if __name__ == "__main__":
    user_input = input("Enter a movie title: ")
    recommendations = recommend(user_input)
    print("Recommendations:")
print(recommendations)
```