05_embeddings

February 25, 2026

1 05 — Embeddings

Generate and compare embeddings from 4 models: - Word2Vec (trained on our corpus) - SBERT multilingual - SBERT MiniLM (English) - SBERT MPNet (English, best quality)

Goal: pick the best model for the recommender based on similarity quality.

[1]:
```
import sys
sys.path.append('../src')

import pandas as pd
import numpy as np
from embeddings import train_word2vec, embed_corpus, load_sbert,␣
 ↪load_embeddings, SBERT_MODELS

df_cnhpsx = pd.read_csv('../data/processed/cnhpsx_clean.csv')
df_news   = pd.read_csv('../data/processed/pakistan_news_clean.csv')

print('CNH-PSX:', df_cnhpsx.shape)
print('Pakistan News:', df_news.shape)
print('\nAvailable SBERT models:', SBERT_MODELS)
```

```
CNH-PSX: (8858, 5)
Pakistan News: (25912, 5)

Available SBERT models: {'multilingual': 'paraphrase-multilingual-
MiniLM-L12-v2', 'minilm': 'all-MiniLM-L6-v2', 'mpnet': 'all-mpnet-base-v2'}
```

1.1 1. Word2Vec

[2]:
```
w2v_corpus = list(df_cnhpsx['headline_clean'].dropna()) +␣
 ↪list(df_news['text_clean'].dropna())
print(f'Training corpus size: {len(w2v_corpus)} texts')

w2v_model = train_word2vec(
    texts=w2v_corpus,
    vector_size=100,
    window=5,
    min_count=2,
```

```
    epochs=10,
    save_path='../data/processed/word2vec.model'
)
```

```
Training corpus size: 34770 texts
[Word2Vec] Tokenizing 34770 texts…
[Word2Vec] Training (vector_size=100, window=5, epochs=10)…
[Word2Vec] Vocabulary size: 15243 words
[Word2Vec] Model saved to ../data/processed/word2vec.model
```

[3]:
```
# Sanity check - similar words
for word in ['karachi', 'bank', 'oil', 'market', 'stock']:
    if word in w2v_model.wv:
        similar = w2v_model.wv.most_similar(word, topn=4)
        print(f"'{word}' -> {[(w, round(s,3)) for w, s in similar]}")
```

```
'karachi' -> [('depleting', 0.765), ('lahore', 0.765), ('islamabad', 0.735),
('karachis', 0.704)]
'bank' -> [('bank's', 0.792), ('nbp', 0.791), ('fardan', 0.789), ('hbl', 0.788)]
'oil' -> [('spiral', 0.864), ('tola', 0.854), ('urea', 0.836), ('offtake',
0.821)]
'market' -> [('trading', 0.905), ('gains', 0.884), ('points', 0.881), ('psx',
0.881)]
'stock' -> [('investor', 0.823), ('modest', 0.813), ('loss', 0.81), ('buying',
0.81)]
```

[4]:
```
w2v_embeddings = embed_corpus(
    texts=list(df_cnhpsx['headline_clean']),
    method='word2vec',
    model=w2v_model,
    save_path='../data/processed/embeddings_w2v.npy'
)
```

```
[embed_corpus] Embedding 8858 texts with word2vec…

Word2Vec:
100%|                                        |
8858/8858 [00:01<00:00, 8176.12it/s]

[embed_corpus] Done. Shape: (8858, 100)
[embed_corpus] Saved to ../data/processed/embeddings_w2v.npy
```

1.2 2. SBERT — 3 models

[5]:
```
# SBERT Multilingual
sbert_multi = load_sbert('multilingual')
emb_multi = embed_corpus(
    texts=list(df_cnhpsx['headline_clean']),
    method='sbert',
    model=sbert_multi,
```

```
    save_path='../data/processed/embeddings_sbert_multilingual.npy'
)
```

```
[SBERT] Loading: paraphrase-multilingual-MiniLM-L12-v2…

Warning: You are sending unauthenticated requests to the HF Hub. Please set a
HF_TOKEN to enable higher rate limits and faster downloads.

modules.json:   0%|          | 0.00/229 [00:00<?, ?B/s]

config_sentence_transformers.json:   0%|          | 0.00/122 [00:00<?, ?B/s]

README.md: 0.00B [00:00, ?B/s]

sentence_bert_config.json:   0%|          | 0.00/53.0 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/645 [00:00<?, ?B/s]

model.safetensors:   0%|          | 0.00/471M [00:00<?, ?B/s]

Loading weights:   0%|          | 0/199 [00:00<?, ?it/s]

BertModel LOAD REPORT from: sentence-transformers/paraphrase-
multilingual-MiniLM-L12-v2
Key                     | Status     |  |
------------------------+------------+--+-
embeddings.position_ids | UNEXPECTED |  |

Notes:
- UNEXPECTED        :can be ignored when loading from different
task/architecture; not ok if you expect identical arch.

tokenizer_config.json:   0%|          | 0.00/526 [00:00<?, ?B/s]

tokenizer.json:   0%|          | 0.00/9.08M [00:00<?, ?B/s]

special_tokens_map.json:   0%|          | 0.00/239 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/190 [00:00<?, ?B/s]

[SBERT] Ready - embedding dim: 384
[embed_corpus] Embedding 8858 texts with sbert…

Batches:   0%|          | 0/139 [00:00<?, ?it/s]

[embed_corpus] Done. Shape: (8858, 384)
[embed_corpus] Saved to ../data/processed/embeddings_sbert_multilingual.npy
```

[6]:
```
# SBERT MiniLM (English, fast)
sbert_minilm = load_sbert('minilm')
emb_minilm = embed_corpus(
    texts=list(df_cnhpsx['headline_clean']),
    method='sbert',
    model=sbert_minilm,
    save_path='../data/processed/embeddings_sbert_minilm.npy'
```

```
)
```

```
[SBERT] Loading: all-MiniLM-L6-v2…
modules.json:   0%|          | 0.00/349 [00:00<?, ?B/s]
config_sentence_transformers.json:   0%|          | 0.00/116 [00:00<?, ?B/s]
README.md: 0.00B [00:00, ?B/s]
sentence_bert_config.json:   0%|          | 0.00/53.0 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/612 [00:00<?, ?B/s]
model.safetensors:   0%|          | 0.00/90.9M [00:00<?, ?B/s]
Loading weights:   0%|          | 0/103 [00:00<?, ?it/s]
BertModel LOAD REPORT from: sentence-transformers/all-MiniLM-L6-v2
Key                     | Status     |  |
------------------------+------------+--+-
embeddings.position_ids | UNEXPECTED |  |

Notes:
- UNEXPECTED        :can be ignored when loading from different
task/architecture; not ok if you expect identical arch.
tokenizer_config.json:   0%|          | 0.00/350 [00:00<?, ?B/s]
vocab.txt: 0.00B [00:00, ?B/s]
tokenizer.json: 0.00B [00:00, ?B/s]
special_tokens_map.json:   0%|          | 0.00/112 [00:00<?, ?B/s]
config.json:   0%|          | 0.00/190 [00:00<?, ?B/s]
[SBERT] Ready - embedding dim: 384
[embed_corpus] Embedding 8858 texts with sbert…
Batches:   0%|          | 0/139 [00:00<?, ?it/s]
[embed_corpus] Done. Shape: (8858, 384)
[embed_corpus] Saved to ../data/processed/embeddings_sbert_minilm.npy
```

[2]:
```
# SBERT MPNet (English, best quality)
sbert_mpnet = load_sbert('mpnet')
emb_mpnet = embed_corpus(
    texts=list(df_cnhpsx['headline_clean']),
    method='sbert',
    model=sbert_mpnet,
    save_path='../data/processed/embeddings_sbert_mpnet.npy'
)
```

```
[SBERT] Loading: all-mpnet-base-v2…
```

```
Warning: You are sending unauthenticated requests to the HF Hub. Please set a
HF_TOKEN to enable higher rate limits and faster downloads.

model.safetensors:   0%|          | 0.00/438M [00:00<?, ?B/s]

Loading weights:   0%|          | 0/199 [00:00<?, ?it/s]

MPNetModel LOAD REPORT from: sentence-transformers/all-mpnet-base-v2
Key                     | Status     |  |
------------------------+------------+--+-
embeddings.position_ids | UNEXPECTED |  |

Notes:
- UNEXPECTED        :can be ignored when loading from different
task/architecture; not ok if you expect identical arch.

tokenizer_config.json:   0%|          | 0.00/363 [00:00<?, ?B/s]

vocab.txt: 0.00B [00:00, ?B/s]

tokenizer.json: 0.00B [00:00, ?B/s]

special_tokens_map.json:   0%|          | 0.00/239 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/190 [00:00<?, ?B/s]

[SBERT] Ready - embedding dim: 768
[embed_corpus] Embedding 8858 texts with sbert…

Batches:   0%|          | 0/139 [00:00<?, ?it/s]

[embed_corpus] Done. Shape: (8858, 768)
[embed_corpus] Saved to ../data/processed/embeddings_sbert_mpnet.npy
```

1.3 3. Comparison — cosine similarity on test headlines

[9]:
```python
import pandas as pd

# Reload dataframes that are no longer in memory
df_stocks = pd.read_csv('../data/processed/psx_stocks_clean.csv')
df_news   = pd.read_csv('../data/processed/pakistan_news_clean.csv')
df_cnhpsx = pd.read_csv('../data/processed/cnhpsx_clean.csv')

print('Stocks:', df_stocks.shape)
print('News:', df_news.shape)
print('CNH-PSX:', df_cnhpsx.shape)
```

```
Stocks: (813588, 10)
News: (25912, 5)
CNH-PSX: (8858, 5)
```

[6]:
```python
from embeddings import load_embeddings, load_word2vec, embed_corpus
# Reload the SBERT models into memory
```

```
sbert_multi  = load_sbert('multilingual')
sbert_minilm = load_sbert('minilm')
sbert_mpnet  = load_sbert('mpnet')
# Load all saved embeddings
w2v_embeddings = load_embeddings('../data/processed/embeddings_w2v.npy')
emb_multi      = load_embeddings('../data/processed/
 ↪embeddings_sbert_multilingual.npy')
emb_minilm     = load_embeddings('../data/processed/embeddings_sbert_minilm.
 ↪npy')
emb_mpnet      = load_embeddings('../data/processed/embeddings_sbert_mpnet.npy')

print("All embeddings loaded!")
```

```
[SBERT] Loading: paraphrase-multilingual-MiniLM-L12-v2…

Warning: You are sending unauthenticated requests to the HF Hub. Please set a
HF_TOKEN to enable higher rate limits and faster downloads.

Loading weights:   0%|          | 0/199 [00:00<?, ?it/s]

BertModel LOAD REPORT from: sentence-transformers/paraphrase-
multilingual-MiniLM-L12-v2
Key                     | Status     |  |
------------------------+------------+--+-
embeddings.position_ids | UNEXPECTED |  |

Notes:
- UNEXPECTED        :can be ignored when loading from different
task/architecture; not ok if you expect identical arch.

[SBERT] Ready - embedding dim: 384
[SBERT] Loading: all-MiniLM-L6-v2…

Loading weights:   0%|          | 0/103 [00:00<?, ?it/s]

BertModel LOAD REPORT from: sentence-transformers/all-MiniLM-L6-v2
Key                     | Status     |  |
------------------------+------------+--+-
embeddings.position_ids | UNEXPECTED |  |

Notes:
- UNEXPECTED        :can be ignored when loading from different
task/architecture; not ok if you expect identical arch.

[SBERT] Ready - embedding dim: 384
[SBERT] Loading: all-mpnet-base-v2…

Loading weights:   0%|          | 0/199 [00:00<?, ?it/s]

MPNetModel LOAD REPORT from: sentence-transformers/all-mpnet-base-v2
Key                     | Status     |  |
------------------------+------------+--+-
```

```
embeddings.position_ids | UNEXPECTED |  |

Notes:
- UNEXPECTED        :can be ignored when loading from different
task/architecture; not ok if you expect identical arch.

[SBERT] Ready - embedding dim: 768
[load_embeddings] Loaded (8858, 100) from ../data/processed/embeddings_w2v.npy
[load_embeddings] Loaded (8858, 384) from
../data/processed/embeddings_sbert_multilingual.npy
[load_embeddings] Loaded (8858, 384) from
../data/processed/embeddings_sbert_minilm.npy
[load_embeddings] Loaded (8858, 768) from
../data/processed/embeddings_sbert_mpnet.npy
All embeddings loaded!
```

[10]:
```python
from sklearn.metrics.pairwise import cosine_similarity
from embeddings import get_sbert_embedding, get_word2vec_embedding
import pandas as pd
import numpy as np

results = {}

#
# TEST 1 - CNH-PSX: same category vs different
#

idx_a = df_cnhpsx[df_cnhpsx['category'] == 'Market'].index[0]
idx_b = df_cnhpsx[df_cnhpsx['category'] == 'Market'].index[1]
idx_c = df_cnhpsx[df_cnhpsx['category'] != 'Market'].index[0]

print("=== TEST 1: CNH-PSX headlines ===")
print('A (Market):', df_cnhpsx.loc[idx_a, 'headline'])
print('B (Market):', df_cnhpsx.loc[idx_b, 'headline'])
print('C (Other): ', df_cnhpsx.loc[idx_c, 'headline'])

# Word2Vec
a, b, c = w2v_embeddings[idx_a], w2v_embeddings[idx_b], w2v_embeddings[idx_c]
results['[CNH] Word2Vec (clean)'] = {
    'A-B (same)': round(cosine_similarity([a], [b])[0][0], 4),
    'A-C (diff)': round(cosine_similarity([a], [c])[0][0], 4),
    'Δ': round(cosine_similarity([a], [b])[0][0] - cosine_similarity([a],␣
 ↪[c])[0][0], 4)
}

for name, model, emb in [
    ('SBERT-Multi',  sbert_multi,  emb_multi),
    ('SBERT-MiniLM', sbert_minilm, emb_minilm),
```

```
    ('SBERT-MPNet',  sbert_mpnet,  emb_mpnet),
]:
    # Clean
    a_c, b_c, c_c = emb[idx_a], emb[idx_b], emb[idx_c]
    sim_ab = cosine_similarity([a_c], [b_c])[0][0]
    sim_ac = cosine_similarity([a_c], [c_c])[0][0]
    results[f'[CNH] {name} (clean)'] = {
        'A-B (same)': round(sim_ab, 4),
        'A-C (diff)': round(sim_ac, 4),
        'Δ': round(sim_ab - sim_ac, 4)
    }
    # Raw
    a_r = get_sbert_embedding(df_cnhpsx.loc[idx_a, 'headline'], model)
    b_r = get_sbert_embedding(df_cnhpsx.loc[idx_b, 'headline'], model)
    c_r = get_sbert_embedding(df_cnhpsx.loc[idx_c, 'headline'], model)
    sim_ab_r = cosine_similarity([a_r], [b_r])[0][0]
    sim_ac_r = cosine_similarity([a_r], [c_r])[0][0]
    results[f'[CNH] {name} (raw)'] = {
        'A-B (same)': round(sim_ab_r, 4),
        'A-C (diff)': round(sim_ac_r, 4),
        'Δ': round(sim_ab_r - sim_ac_r, 4)
    }


#
# TEST 2 - Pakistan News: same section vs different
#

# Pick a section with enough articles
top_section = df_news['section'].value_counts().index[0]
idx_d = df_news[df_news['section'] == top_section].index[0]
idx_e = df_news[df_news['section'] == top_section].index[1]
idx_f = df_news[df_news['section'] != top_section].dropna(subset=['section']).
 ↪index[0]

print(f"\n=== TEST 2: Pakistan News (section='{top_section}') ===")
print('D (same section):', df_news.loc[idx_d, 'heading'])
print('E (same section):', df_news.loc[idx_e, 'heading'])
print('F (diff section):', df_news.loc[idx_f, 'heading'])

for name, model in [
    ('SBERT-Multi',  sbert_multi),
    ('SBERT-MiniLM', sbert_minilm),
    ('SBERT-MPNet',  sbert_mpnet),
]:
    # Raw headlines (Pakistan News has full text, no need for cleaned version)
    d_r = get_sbert_embedding(df_news.loc[idx_d, 'text_combined'], model)
```

```
    e_r = get_sbert_embedding(df_news.loc[idx_e, 'text_combined'], model)
    f_r = get_sbert_embedding(df_news.loc[idx_f, 'text_combined'], model)
    sim_de = cosine_similarity([d_r], [e_r])[0][0]
    sim_df = cosine_similarity([d_r], [f_r])[0][0]
    results[f'[News] {name} (raw)'] = {
        'A-B (same)': round(sim_de, 4),
        'A-C (diff)': round(sim_df, 4),
        'Δ': round(sim_de - sim_df, 4)
    }


#
# TEST 3 - PSX Stocks: same ticker name in headline vs different
#

# Find a ticker that appears in CNH-PSX headlines
common_tickers = df_stocks['symbol'].unique()
ticker_found = None
for ticker in common_tickers:
    mask = df_cnhpsx['headline'].str.contains(ticker, case=False, na=False)
    if mask.sum() >= 2:
        ticker_found = ticker
        break

if ticker_found:
    ticker_headlines = df_cnhpsx[df_cnhpsx['headline'].str.
 ↪contains(ticker_found, case=False, na=False)]
    other_headlines  = df_cnhpsx[~df_cnhpsx['headline'].str.
 ↪contains(ticker_found, case=False, na=False)]

    idx_g = ticker_headlines.index[0]
    idx_h = ticker_headlines.index[1]
    idx_i = other_headlines.index[0]

    print(f"\n=== TEST 3: PSX Stocks - ticker '{ticker_found}' in headlines␣
 ↪===")
    print('G (mentions ticker):', df_cnhpsx.loc[idx_g, 'headline'])
    print('H (mentions ticker):', df_cnhpsx.loc[idx_h, 'headline'])
    print('I (other headline): ', df_cnhpsx.loc[idx_i, 'headline'])

    for name, model in [
        ('SBERT-Multi',  sbert_multi),
        ('SBERT-MiniLM', sbert_minilm),
        ('SBERT-MPNet',  sbert_mpnet),
    ]:
        g_r = get_sbert_embedding(df_cnhpsx.loc[idx_g, 'headline'], model)
        h_r = get_sbert_embedding(df_cnhpsx.loc[idx_h, 'headline'], model)
```

```
        i_r = get_sbert_embedding(df_cnhpsx.loc[idx_i, 'headline'], model)
        sim_gh = cosine_similarity([g_r], [h_r])[0][0]
        sim_gi = cosine_similarity([g_r], [i_r])[0][0]
        results[f'[Stocks] {name} (raw)'] = {
            'A-B (same)': round(sim_gh, 4),
            'A-C (diff)': round(sim_gi, 4),
            'Δ': round(sim_gh - sim_gi, 4)
        }
else:
    print("\nNo common ticker found between stocks and headlines - skipping␣
 ↪test 3")


#
# Final results table
#

df_results = pd.DataFrame(results).T
print('\n\n=== FULL COMPARISON ACROSS ALL DATASETS ===')
print(df_results.to_string())
print('\n→ Best model overall: highest Δ')
best = df_results['Δ'].idxmax()
print(f'→ Winner: {best} with Δ = {df_results.loc[best, "Δ"]}')
```

```
=== TEST 1: CNH-PSX headlines ===
A (Market): ['KSE index plunges by 83 points']
B (Market): ['Karachi stocks record mixed trend,,,,,By our correspondent']
C (Other):  ['Oil prices fall']

=== TEST 2: Pakistan News (section='Pakistan') ===
D (same section): Chinese national held for beating traffic police constable in
Karachi
E (same section): Sarmad Khoosat reveals why Zindagi Tamasha's trailer was
removed from YouTube
F (diff section): Iraqi paramilitaries call for withdrawal from US embassy

=== TEST 3: PSX Stocks - ticker 'ABL' in headlines ===
G (mentions ticker): ['Bulls remain on drive, KSE gains 118 points', 'KSE to be
more profitable in 2007, says report']
H (mentions ticker): ['KSE dips as investors book profit on available margins']
I (other headline):  ['KSE index plunges by 83 points']


=== FULL COMPARISON ACROSS ALL DATASETS ===
                            A-B (same)  A-C (diff)       Δ
[CNH] Word2Vec (clean)          0.8135      0.5888  0.2248
[CNH] SBERT-Multi (clean)       0.2294      0.2065  0.0230
[CNH] SBERT-Multi (raw)         0.2751      0.3147 -0.0397
```

```
[CNH] SBERT-MiniLM (clean)       0.2994      0.2682  0.0312
[CNH] SBERT-MiniLM (raw)         0.4256      0.3558  0.0697
[CNH] SBERT-MPNet (clean)        0.3387      0.3328  0.0059
[CNH] SBERT-MPNet (raw)          0.4185      0.4235 -0.0050
[News] SBERT-Multi (raw)        -0.0215     -0.1026  0.0810
[News] SBERT-MiniLM (raw)        0.0575     -0.0195  0.0770
[News] SBERT-MPNet (raw)         0.0668     -0.0103  0.0772
[Stocks] SBERT-Multi (raw)       0.4744      0.4122  0.0622
[Stocks] SBERT-MiniLM (raw)      0.6056      0.5786  0.0269
[Stocks] SBERT-MPNet (raw)       0.7068      0.5967  0.1101

→ Best model overall: highest Δ
→ Winner: [CNH] Word2Vec (clean) with Δ = 0.2248000055551529
```