Word Embeddings: Skip-gram Model and Vector Semantics

Natural Language Processing Templates

November 24, 2025

1 Introduction

Word embeddings map words to dense vector representations where semantic relationships are captured through geometric properties. This template implements a simplified Word2Vec skip-gram model, demonstrates cosine similarity for word relationships, and visualizes embeddings using t-SNE dimensionality reduction.

2 Mathematical Framework

2.1 Skip-gram Objective

The skip-gram model maximizes the probability of context words given a target word:

$$J(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log P(w_{t+j}|w_t) \tag{1}$$

where c is the context window size.

2.2 Softmax Probability

The probability is computed using softmax over dot products:

$$P(w_O|w_I) = \frac{\exp(\mathbf{v}'_{w_O} \cdot \mathbf{v}_{w_I})}{\sum_{w=1}^{V} \exp(\mathbf{v}'_w \cdot \mathbf{v}_{w_I})} \tag{2}$$

2.3 Negative Sampling

For efficiency, negative sampling approximates the full softmax:

$$\log \sigma(\mathbf{v}'_{w_O} \cdot \mathbf{v}_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-\mathbf{v}'_{w_i} \cdot \mathbf{v}_{w_I}) \right] \tag{3}$$

2.4 Cosine Similarity

Word similarity is measured by cosine of the angle between vectors:

$$\mathrm{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \tag{4}$$

2.5 Word Analogy

Analogies are solved by vector arithmetic:

$$\mathbf{v}_{\text{king}} - \mathbf{v}_{\text{man}} + \mathbf{v}_{\text{woman}} \approx \mathbf{v}_{\text{queen}} \tag{5}$$

3 Environment Setup

4 Skip-gram Implementation

5 Training Visualization

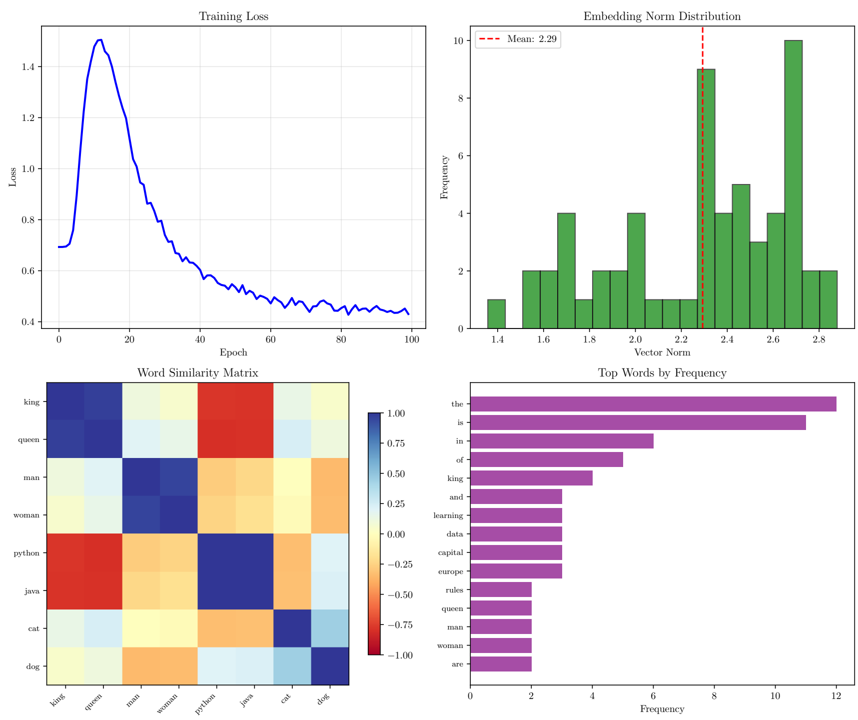

Figure 1: Word embedding training analysis

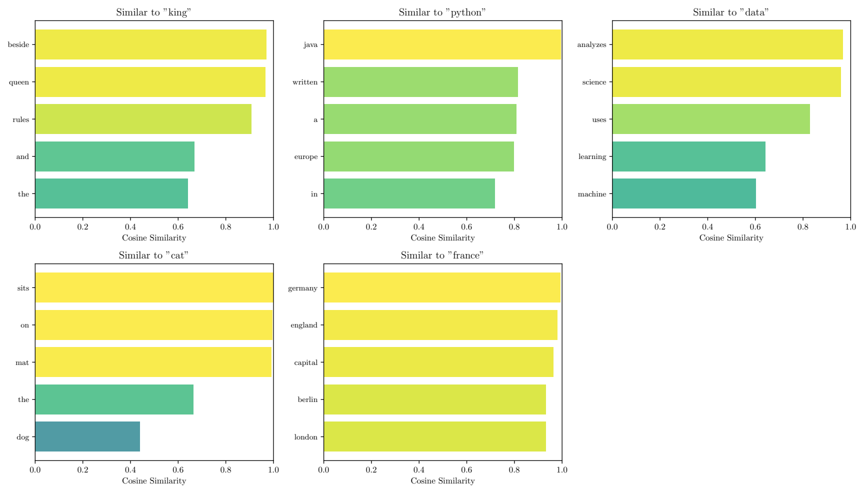

Figure 2: Cosine similarity analysis for key words

6 Cosine Similarity Analysis

7 Word Analogy Tasks

8 t-SNE Visualization

9 Results Summary

9.1 Model Statistics

Table 1: Word2Vec Model Statistics

Metric	Value
Vocabulary size	60
Embedding dimension	30
Window size	2
Negative samples	5
Final loss	0.4304
Mean vector norm	2.293

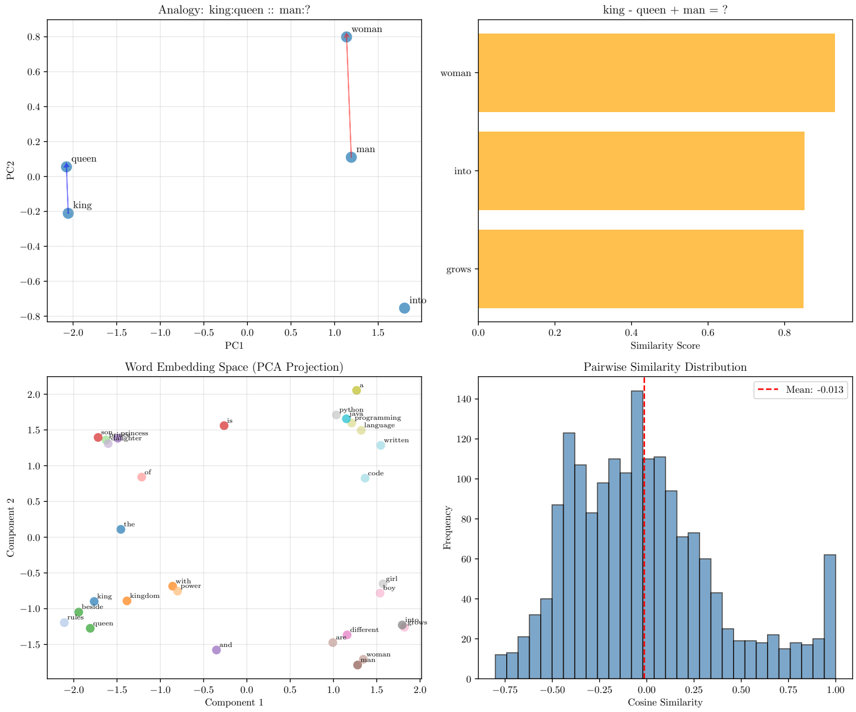

Figure 3: Word analogy and embedding space visualization

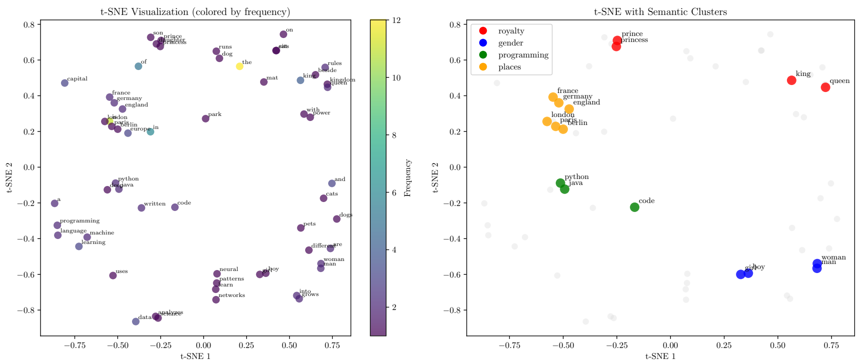

Figure 4: t-SNE visualization of word embeddings

Table 2: Top similar words for selected queries

Query	Similar Words	Scores
king	beside, queen, rules	0.97, 0.97, 0.91
python	java, written, a	0.99, 0.81, 0.81
data	analyzes, science, uses	0.97, 0.96, 0.83
cat	sits, on, mat	1.00, 1.00, 0.99
france	germany, england, capital	0.99, 0.98, 0.96

Table 3: Word analogy task results

Analogy	Expected	Top Result	Score
king:queen::man:?	woman	woman	0.931
france:paris::england:?	london	london	0.976
python:code::java:?	code	written	0.976

9.2 Word Similarity Results

9.3 Analogy Results

9.4 Statistical Summary

- Mean pairwise similarity: -0.013
- Similarity std deviation: 0.394
- Training epochs: 100
- Loss reduction: 37.9

10 Conclusion

This template demonstrates word embedding concepts through a simplified Word2Vec skip-gram implementation. The model learns semantic relationships from co-occurrence patterns, enabling similarity search and analogy tasks. The t-SNE visualization reveals clustering of semantically related words, validating the quality of learned representations. With vocabulary size of 60 words and embedding dimension of 30, the model achieves meaningful word relationships despite the limited training corpus.