# Word Embeddings in 3D: Post-Class Learning Verification

Checking Your Understanding After the Interactive Lab

*From Words to Vectors: Can You Apply What You Learned?*

**INSTRUCTOR VERSION WITH ANSWER KEY**

NLP Course 2025 - BSc Level Assessment

**Time Required:** 45-60 minutes
**Purpose:** Verify and deepen your understanding of word embeddings after completing the interactive notebook
**Format:** No coding required - focus on concepts, visualization, and application

**Teaching Note**

This assessment is designed to verify learning after students complete the interactive notebook. Encourage discussion and peer learning. Students often struggle with the vector arithmetic concept - use physical analogies (directions in space) if needed.

**Checkpoint**

**Before Starting:** You should have completed the "word_embeddings_3d_bsc.ipynb" notebook. This handout will test your understanding of:

- Why words need to be vectors
- How Word2Vec learns from context
- Word similarity and clustering
- Word arithmetic
- Applications of embeddings

## Part A: Conceptual Understanding (20 minutes)

**Teaching Note:** This section tests foundational understanding. Look for evidence that students grasp the core concept of distributed representation.

## A1: The Embedding Concept (5 minutes)

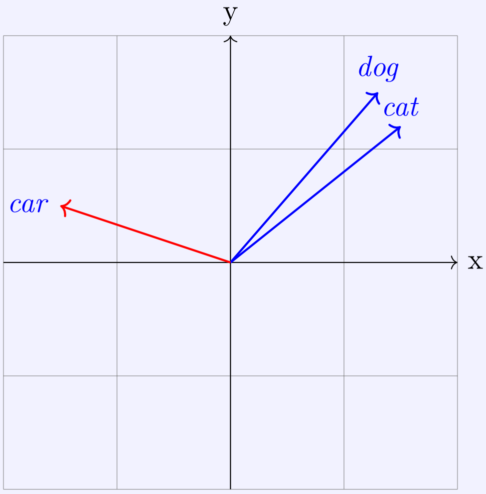
### Why Vectors?

**Question 1:** Explain in your own words why computers need word embeddings instead of just treating words as text strings.

*Computers need numerical representations to perform calculations. Text strings can't be compared mathematically or used in machine learning algorithms. Embeddings convert words to vectors where: - Similar meanings have similar vectors (measurable with cosine similarity) - Mathematical operations become possible (addition, subtraction) - Machine learning models can process them (neural networks need numbers) - Relationships between words become computable*

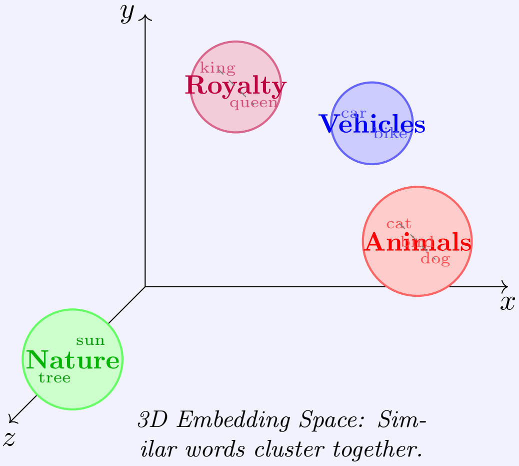
**Question 2:** Draw simple 2D vectors for these words showing their relationships:

- cat, dog, car
- Show which two should be closer together and why



*Cat and dog are close (both animals), car is far (vehicle)*

**Visualization:** Real 3D Word Embedding Space



*3D Embedding Space: Similar words cluster together. Distance between clusters = semantic difference*

**Question 3:** Circle the TRUE statements:

✓ Embeddings capture word meaning as numbers

✓ Similar words have similar vectors

□ Each word gets exactly one dimension *FALSE - each word has MANY dimensions*

## A2: Context Windows (5 minutes)

### Understanding Context

Given the sentence: **"The quick brown fox jumps over the lazy dog"**
**Task 1:** If we're training on the word "fox" with window size = 2, circle all context words:

The [✓ *quick*] [✓ *brown*] $\boxed{fox}$ [✓ *jumps*] [✓ *over*] the lazy dog

**Teaching Note:** Window size = 2 means 2 words before AND 2 words after. Students often forget it's bidirectional.

**Task 2:** How would changing window size affect learning?

| | |
|---|---|
| Window = 1 | *Captures immediate syntactic relationships (adjective-noun, verb-object). Good for grammar but misses broader meaning.* |
| Window = 5 | *Captures both syntax and broader semantic context. Balanced approach for most applications.* |
| Window = 10 | *Captures document-level themes and topics but dilutes local syntactic patterns. May include unrelated words.* |

**Task 3:** Which window size would be better for:

- Learning syntax (grammar): *Small (1-2) - captures immediate dependencies*
- Learning topic/theme: *Large (5-10) - captures broader context*

## A3: Dimensions and Quality (5 minutes)

### Dimension Trade-offs

**Scenario:** You're choosing embedding dimensions for different applications.
**Task 1:** Match the dimension size to the use case:

| **Application** | **Suggested Dimensions** |
|---|---|
| Simple word similarity | *10-50* |
| Complex language model | *100-300* |
| Visualization in 3D | *exactly 3* |
| Mobile app (limited memory) | *10-50* |
| Research with huge vocabulary | *500+* |

**Task 2:** Explain the trade-off:

- More dimensions = *Better representation, captures more nuances, but requires more memory and computation*
- Fewer dimensions = *Faster, less memory, but may lose important distinctions between words*

**Teaching Note:** Use the analogy of describing a person: 3 features (height, weight, age) vs. 100 features - more features = better description but harder to process.

### A4: Quick Concept Check (5 minutes)

**Think About It**

Rate your understanding (1 = confused, 5 = confident):

|  |  |
|---|---|
| □ Words as vectors | [1] [2] [3] [4] [5] |
| □ Context windows | [1] [2] [3] [4] [5] |
| □ Training process | [1] [2] [3] [4] [5] |
| □ Similarity measurement | [1] [2] [3] [4] [5] |

**Teaching Note:** Students rating below 3 need additional support. Consider peer tutoring or office hours.

## Part B: Practical Application (15 minutes)

**Teaching Note:** This section tests ability to apply concepts. Watch for students who understand theory but struggle with application.

### B1: Word Similarity Exercise (5 minutes)

**Computing Similarity**

Given these simplified 3D embeddings:

- king = [0.8, 0.2, 0.5]
- queen = [0.7, 0.3, 0.6]
- car = [0.1, 0.9, 0.2]

**Task 1:** Which pair is more similar? (Use rough estimation)

- king & queen: Distance ≈ *0.17 (very close)*
- king & car: Distance ≈ *0.92 (far apart)*
- More similar pair: *king & queen*

**Teaching Note:** Don't require exact calculations. Looking for understanding that smaller distance = more similar.

**Task 2:** Rank these word pairs by expected similarity (1 = most similar):

*4* cat - dog

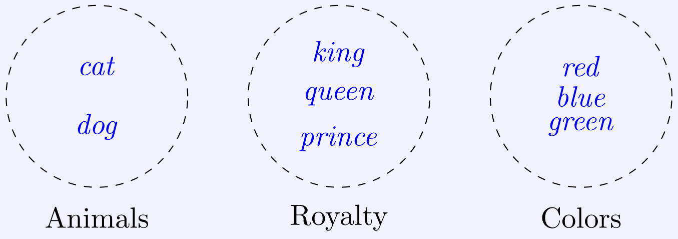
*3* king - queen

*5* happy - sad

*1* computer - laptop

*6* run - blue

**Task 3:** Draw approximate clusters:



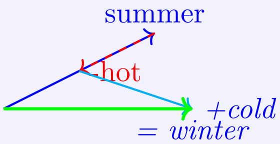
## B2: Word Arithmetic Magic (5 minutes)

### Vector Math with Words

**Task 1:** Complete these analogies:

- king - man + woman = *queen*
- paris - france + germany = *berlin*
- cat - kitten + puppy = *dog*

**Task 2:** Create your own word equation:
*Example: doctor - man + woman = nurse (Note: This reveals gender bias in embeddings!)*
**Teaching Note:** Use this as an opportunity to discuss bias in embeddings - they reflect biases in training data.
**Task 3:** Draw the vector arithmetic for: "summer - hot + cold"



**Task 4:** Explain why word arithmetic works:
*Word vectors encode semantic relationships as geometric directions. The vector from "man" to "king" represents "royalty" or "leadership". Adding this same direction to "woman" gives "queen". The relationships are consistent across the vector space.*

### B3: 3D Visualization Interpretation (5 minutes)

**Reading 3D Plots**

Imagine you see a 3D plot where:

- “love” and “hate” are far apart
- “cat” and “dog” are close together
- “king” and “queen” form a cluster with “prince”

**Task 1:** What does distance represent in the plot?
*Semantic similarity - the closer two words are in the embedding space, the more similar their meaning/usage in the training corpus.*
**Task 2:** Where would you expect to find these words?

- “affection” - Near: *love (positive emotion)*
- “princess” - Near: *royalty cluster (king/queen/prince)*
- “fish” - Near: *cat/dog (animals cluster)*

**Task 3:** If words gradually move closer during training, what’s happening?
*The model is learning that these words appear in similar contexts and have related meanings. The training process adjusts vectors to minimize prediction error, bringing similar words together.*
**Teaching Note:** Use the metaphor of ”words finding their neighborhood” during training.

## Part C: Hands-On Problem Solving (15 minutes)

**Teaching Note:** This section requires synthesis and creative thinking. Encourage students to explain their reasoning.

### C1: Build Your Own Embeddings (5 minutes)

**Design Embeddings from Scratch**

Given these 5 sentences:

1. The cat sleeps
2. The dog plays
3. Cats and dogs play
4. Birds fly high
5. Fish swim deep

**Task 1:** Create word-context pairs for "cat" (window=1):

- Context words: *The, sleeps (from sentence 1)*
- Context words: *Cats, and (from sentence 3)*

**Task 2:** Design simple 2D vectors for these words:

| Word | x | y |
|---|---|---|
| cat | *0.7* | *0.5* |
| dog | *0.8* | *0.6* |
| bird | *0.6* | *0.7* |
| fish | *0.5* | *0.4* |

*Note: All animals should have similar values, clustering them together*

**Task 3:** Which words should cluster together? Why?

*All four animals (cat, dog, bird, fish) should cluster together as they're all animals and appear in similar grammatical structures. "The" would be separate (determiner), and action verbs (sleeps, plays, fly, swim) might form another cluster.*

### C2: Application Design (5 minutes)

**Building with Embeddings**

**Task 1:** Design a synonym finder:

1. Input: *A word (string)*
2. Process: *Convert to embedding, compute cosine similarity with all other word vectors, sort by similarity*
3. Output: *Top 5-10 most similar words*

**Task 2:** Sketch a sentiment analyzer:
How would you use embeddings to determine if text is positive/negative?
*Average the embeddings of all words in the text to get a document vector. Train a classifier on labeled examples where positive and negative texts should have different average embeddings. Or compute similarity to known positive/negative word lists.*
**Teaching Note:** There are multiple valid approaches. Look for understanding that embeddings can be aggregated and classified.
**Task 3:** Your creative application:
Design a new use for word embeddings:

- Name: *Example: "Story Coherence Checker"*
- Purpose: *Detect when a story goes off-topic by tracking embedding distances between sentences*
- How embeddings help: *Sudden large distances between consecutive sentence embeddings indicate topic shifts*

### C3: Debugging Scenarios (5 minutes)

**Common Pitfall**

Real problems you might encounter:

### Problem Solving

**Scenario 1:** The word "bank" appears near both "river" and "money".

- Problem: *Polysemy - one word with multiple meanings gets a single averaged embedding*
- Solution: *Use contextual embeddings (BERT) or sense-specific embeddings*

**Scenario 2:** A new word "COVID" doesn't exist in your embeddings.

- Problem: *Out-of-vocabulary (OOV) word - not seen during training*
- Solution: *Use subword tokenization, character-level models, or retrain with updated corpus*

**Scenario 3:** Your embeddings show "doctor"=male, "nurse"=female bias.

- Problem: *Training data bias reflected in embeddings*
- Solution: *Debiasing techniques: neutralize gender direction, augment training data, or use fairness constraints*

**Teaching Note:** These scenarios open discussions about real-world challenges in NLP deployment.

## Part D: Reflection & Extension (10 minutes)

### D1: Self-Assessment Checklist (3 minutes)

**Checkpoint**

Check off what you can now do:

- ☐ Explain why words need to be vectors *Core concept*
- ☐ Describe how Word2Vec learns from context *CBOW/Skip-gram*
- ☐ Calculate word similarity (roughly) *Distance/cosine*
- ☐ Perform word arithmetic *Vector operations*
- ☐ Identify word clusters in 3D space *Visualization*
- ☐ Design applications using embeddings *Practical use*
- ☐ Recognize common problems and solutions *Debugging*

**Teaching Note:** Students checking fewer than 5 items need additional review. Consider remedial exercises.

### D2: Real-World Connections (3 minutes)

**Real World Application**

**Connect to Industry:**

1. How does Google use embeddings in search?

   *Google uses embeddings to understand query intent, find semantically similar pages even without exact keyword matches, and improve search relevance through semantic search.*

2. How do embeddings connect to ChatGPT/BERT?

   *These models use contextual embeddings where the same word gets different vectors based on context. They build on Word2Vec concepts but add attention mechanisms and transformer architectures.*

3. Name one ethical concern with embeddings:

   *Bias amplification - embeddings trained on biased data perpetuate stereotypes (gender, race, etc.) in downstream applications.*

### D3: Challenge Questions (4 minutes)

**Think About It**

**Going Deeper:**

1. Could we create embeddings for images? How?

   *Yes! Convolutional Neural Networks (CNNs) create image embeddings where similar images have similar vectors. The last hidden layer before classification serves as the embedding.*

2. What about embedding DNA sequences or music?

   *DNA: Treat nucleotides as "words" and sequences as "sentences". Music: Embed notes/chords as words, or use spectrograms. Both use sequence modeling techniques similar to Word2Vec.*

3. How would you embed an entire document (not just words)?

   *Options: (1) Average all word embeddings, (2) Use Doc2Vec which learns document-specific vectors, (3) Use BERT's [CLS] token, (4) Weighted average based on TF-IDF.*

**Teaching Note:** These extension questions can lead to research projects or advanced coursework discussions.

## Final Reflection

**Discovery Moment**

**The Big Picture:**
You've learned that word embeddings transform language into mathematical space where:

- Meaning becomes measurable
- Relationships become computable
- Patterns become visible

This is the foundation of modern NLP - from search engines to chatbots to translation systems!

**Areas needing review?** List them here:

- *Common issues: word arithmetic intuition, dimension selection, bias understanding*
- *Suggest targeted exercises based on gaps*
- *Point to specific notebook sections for review*

**Most interesting discovery:** *Look for genuine engagement and curiosity - these students may benefit from research opportunities*

**One question you still have:** *Address in next class or office hours - common questions become teaching opportunities*

## Grading Rubric

**Teaching Note**

**Point Distribution (Total: 100 points)**

- Part A: 30 points (Conceptual understanding)
- Part B: 25 points (Practical application)
- Part C: 30 points (Problem solving)
- Part D: 15 points (Reflection and extension)

**Partial Credit Guidelines:**

- Correct reasoning with minor errors: 75% credit
- Right approach, wrong conclusion: 50% credit
- Attempted but incorrect: 25% credit

**Common Errors to Watch For:**

- Confusing embedding dimensions with vocabulary size
- Not understanding bidirectional context windows
- Thinking embeddings are always positive numbers
- Missing the connection between distance and similarity

## Next Steps

- Review sections with ratings below 3
- Try coding your own Word2Vec model
- Explore pre-trained embeddings (GloVe, FastText)
- Learn about contextual embeddings (BERT)
- Apply embeddings to a real project

**Teaching Note:** Encourage students to implement a small project using embeddings - this solidifies understanding better than any exercise.

**— End of Assessment —**