Homework 4: Word Embeddings with Word2Vec

Points: 20 | **Due:** Sunday, February 22, 2026 @ 11pm Pacific

Author: Richard Young, Ph.D. | UNLV Lee Business School

Compute: CPU (free tier)

Learning Objectives

1. **Understand** how words become numbers (and why it matters)
2. **Train** your own Word2Vec model on domain-specific data
3. **Document** embedding successes AND failures
4. **Create** business-relevant word analogies
5. **Compare** your embeddings to pre-trained models

Grading

Component	Points	What We're Looking For
Data Prep	3	Text preprocessed for Word2Vec
Model Training	4	Word2Vec model trained successfully
Similarities	4	Word similarities explored and analyzed
Failures	4	At least 1 embedding failure documented
Analogies	3	3 business-relevant analogies tested
Comparison	2	Compared to pre-trained embeddings
Total	**20**	

The Big Picture

Before embeddings, computers saw words as arbitrary symbols. **Word2Vec changed everything** by learning that similar words should have similar numbers.

The magic: king - man + woman ≈ queen

Instructions

1. Open `MIS769_HW4_Word_Embeddings.ipynb` in Google Colab
2. Load your dataset (need 10,000+ documents for good embeddings)
3. Preprocess text for Word2Vec training
4. Train your model and explore word similarities
5. Hunt for embedding failures (antonyms, rare words, polysemy)
6. Create and test business-relevant analogies
7. Compare your embeddings to pre-trained GloVe

Questions to Answer

- **Q1:** Which similarities made sense? Which surprised you?
- **Q2:** Document an embedding failure: what happened and why?
- **Q3:** Which analogies worked? Why do some fail?
- **Q4:** How do your embeddings differ from pre-trained ones?

Submission

Upload to Canvas: - Your completed `.ipynb` notebook with all cells executed

---

— *Richard Young, Ph.D.*