L06: Embeddings & RL

Text Representations and Sequential Decision Making

Methods and Algorithms

MSc Data Science

Spring 2026

Outline

Learning Objectives

By the end of this lecture, you will be able to:

1. **Derive** the Skip-Gram objective and analyze the negative sampling approximation
2. **Evaluate** static vs. contextual embeddings for domain-specific NLP tasks (e.g., FinBERT)
3. **Analyze** the convergence properties of Q-learning and the role of the exploration–exploitation tradeoff
4. **Critique** RL-based trading strategies and their limitations (transaction costs, non-stationarity, overfitting)

Bloom's taxonomy levels 4–5: Analyze, Evaluate, Derive, Critique

The Business Problem

Text Data Challenge

- Financial news, reports, social media contain valuable signals
- Text is unstructured—how to feed it to ML models?
- Need to capture semantic meaning ("bullish" similar to "positive")

Sequential Decision Challenge

- Trading requires sequences of buy/sell/hold decisions
- Actions have delayed consequences (profit realized later)

Embeddings solve text, RL solves sequential decisions

Why Is This Hard?

"There are only two hard problems in NLP: understanding language, and getting your regex to work." — adapted from Phil Karlton

This Lecture:

- Part 1: Turn text into numbers that capture meaning (Embeddings)
- Part 2: Learn to make good decisions over time (RL)

XKCD #1838 by Randall Munroe (CC BY-NC 2.5): "Machine Learning" — relevant to both topics

Key Equations

Embeddings — Skip-Gram Objective:

$$\max \sum_{t=1}^{T} \sum_{\substack{-c \leq j \leq c \\ j \neq 0}} \log p(w_{t+j} \mid w_t)$$

Cosine Similarity:

$$\text{sim}(u, v) = \frac{u \cdot v}{\|u\| \, \|v\|}$$

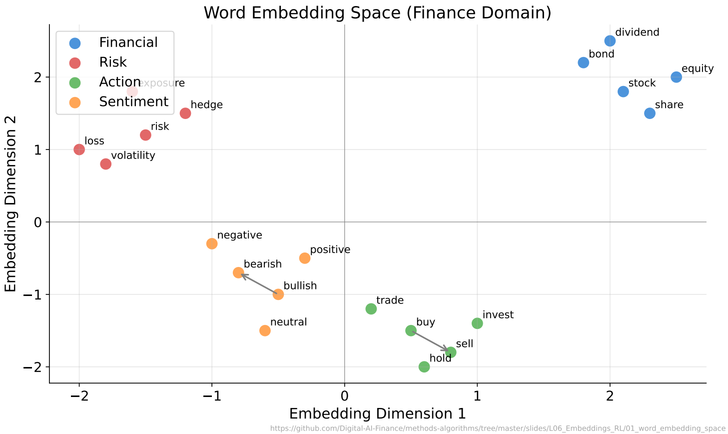
Reinforcement Learning — Bellman Equation:

$$Q^*(s, a) = \mathbb{E}\left[r + \gamma \max_{a'} Q^*(s', a') \mid s, a\right]$$

TD Update (Q-Learning):

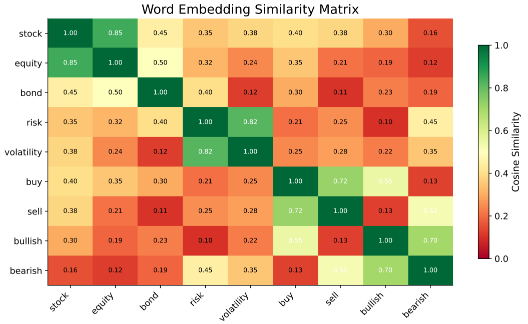
$$Q(s, a) \leftarrow Q(s, a) + \alpha\left[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]$$

These four equations are the mathematical backbone of this lecture



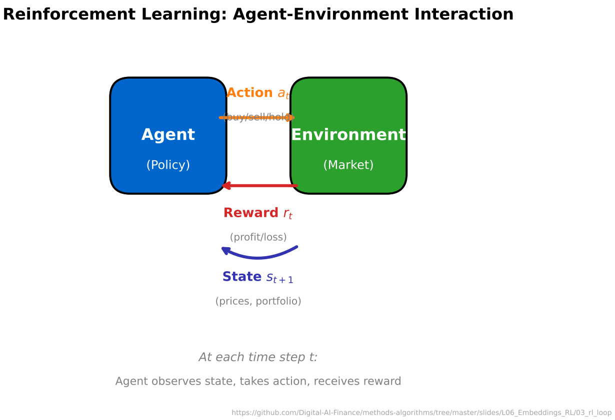
Similar words cluster together in embedding space

Embedding Similarity



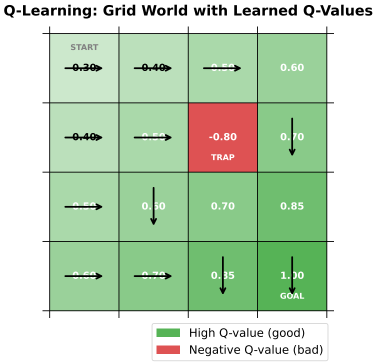
https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L06_Embeddings_RL/02_similarity_heatmap

Cosine similarity captures semantic relationships



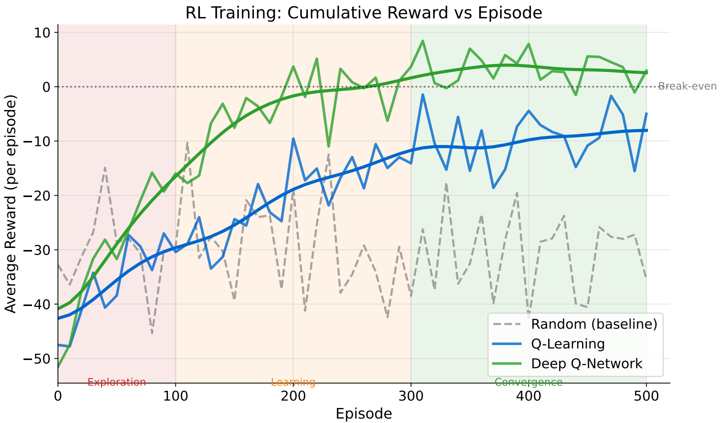
Agent takes actions, receives rewards, learns optimal policy

Q-Learning: Value Function



https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L06_Embeddings_RL/04_q_learning_grid

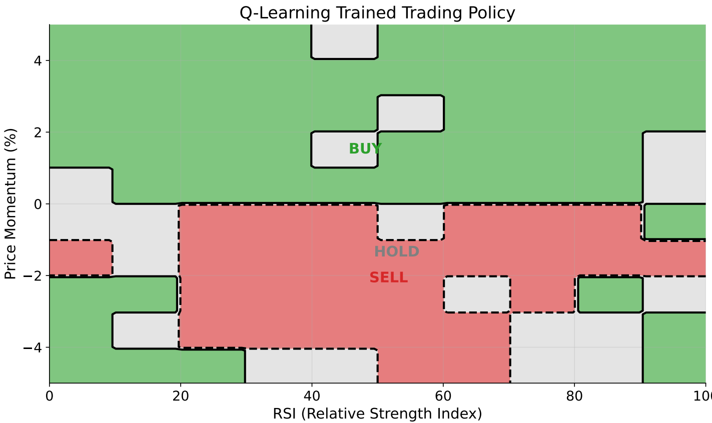
Q-values show expected reward from each state-action



https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L06_Embeddings_RL/05_reward_curves

RL agents improve through exploration and exploitation

Learned Trading Policy



https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L06_Embeddings_RL/06_policy_viz

Policy maps states to actions (when to buy/sell/hold)

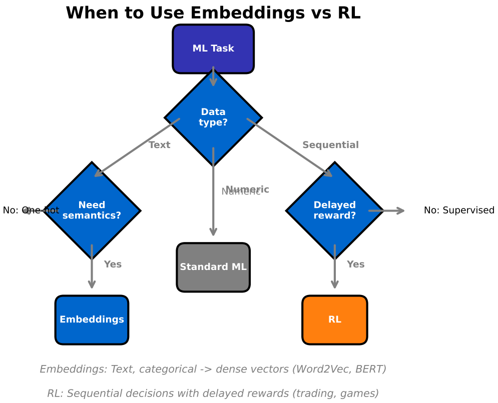
Hands-on Exercise

Open the Colab Notebook

- Exercise 1: Explore word embeddings with Word2Vec
- Exercise 2: Implement basic Q-learning
- Exercise 3: Apply RL to a simple trading environment

Link: `https://colab.research.google.com/github/Digital-AI-Finance/methods-algorithms/blob/master/notebooks/L06_embeddings_rl.ipynb`

Decision Framework



https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L06_Embeddings_RL/07_decision_flowchart

Embeddings for text, RL for sequential decisions with delayed rewards

References

- Mikolov et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv.
- Sutton, R. & Barto, A. (2018). *Reinforcement Learning: An Introduction*. MIT Press.
- Jurafsky & Martin (2024). *Speech and Language Processing*. `https://web.stanford.edu/~jurafsky/slp3/`