Methods and Algorithms

Spring 2026

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

1. Explain word embeddings and their applications
2. Apply pre-trained embeddings for text analysis
3. Understand the reinforcement learning framework
4. Implement basic Q-learning for decision problems

Finance Applications: Sentiment analysis, algorithmic trading

From text to numbers, from decisions to optimal policies

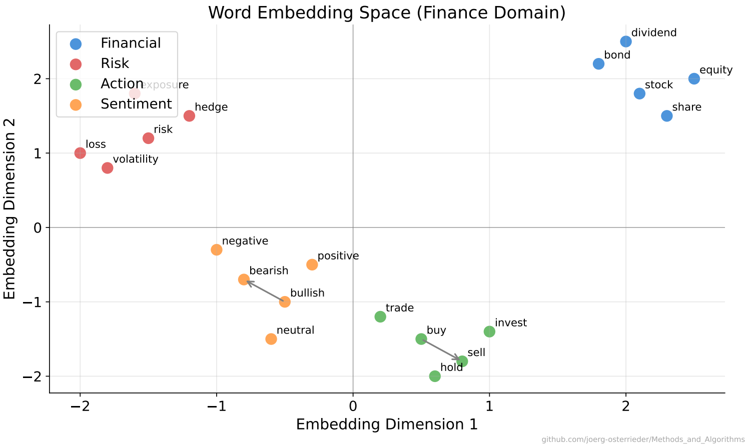
Text Data Challenge

- 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”)

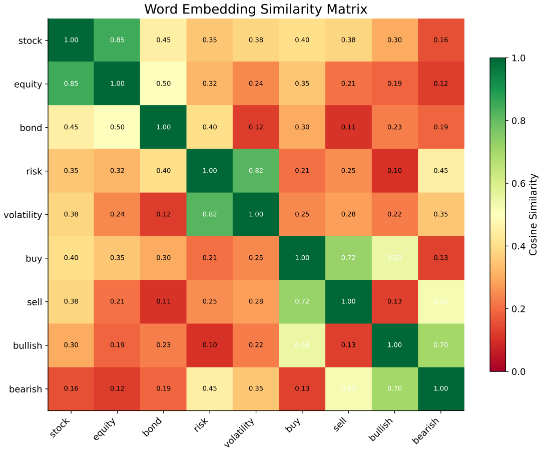
Sequential Decision Challenge

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

Embeddings solve text, RL solves sequential decisions



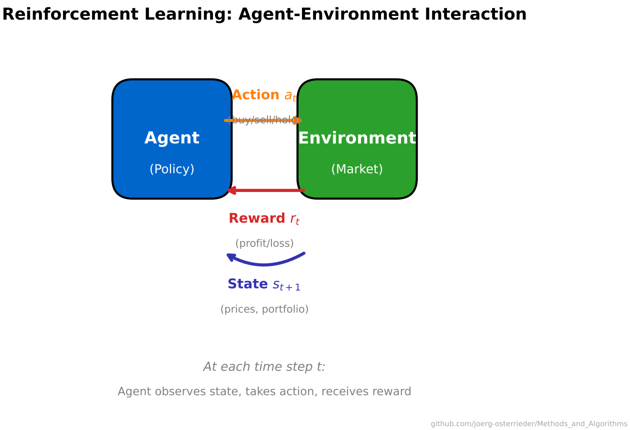
Similar words cluster together in embedding space



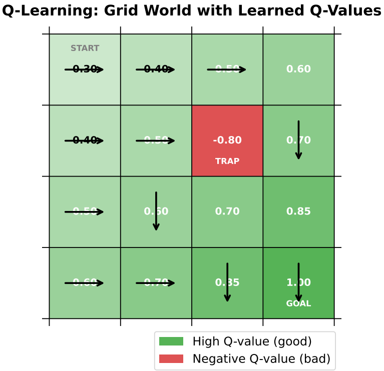
	stock	equity	bond	risk	volatility	buy	sell	bullish	bearish
stock	1.00	0.85	0.45	0.35	0.38	0.40	0.38	0.30	0.16
equity	0.85	1.00	0.50	0.32	0.24	0.35	0.21	0.19	0.12
bond	0.45	0.50	1.00	0.40	0.12	0.30	0.11	0.23	0.19
risk	0.35	0.32	0.40	1.00	0.82	0.21	0.25	0.10	0.45
volatility	0.38	0.24	0.12	0.82	1.00	0.25	0.28	0.22	0.35
buy	0.40	0.35	0.30	0.21	0.25	1.00	0.72	0.55	0.13
sell	0.38	0.21	0.11	0.25	0.28	0.72	1.00	0.13	0.52
bullish	0.30	0.19	0.23	0.10	0.22	0.55	0.13	1.00	0.70
bearish	0.16	0.12	0.19	0.45	0.35	0.13	0.52	0.70	1.00

github.com/joerg-osterrieder/Methods_and_Algorithms

Cosine similarity captures semantic relationships

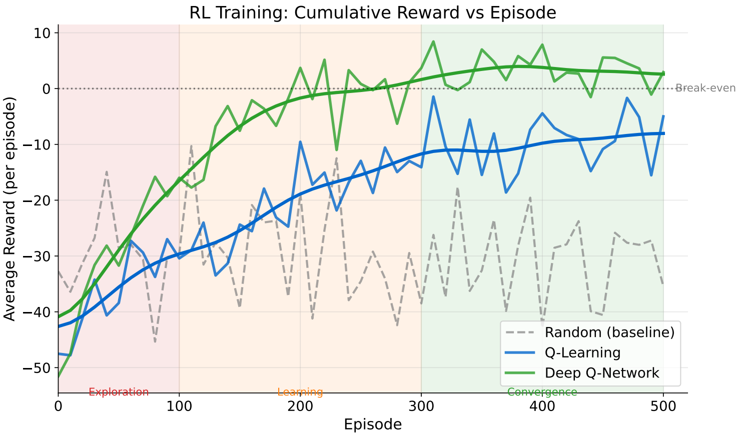

Reinforcement Learning: Agent-Environment Interaction
Agent
(Policy)
Environment
(Market)
Action a_t
buy/sell/hold
Reward r_t
(profit/loss)
State s_{t+1}
(prices, portfolio)
At each time step t:
Agent observes state, takes action, receives reward
github.com/joerg-osterrieder/Methods_and_Algorithms

Agent takes actions, receives rewards, learns optimal policy



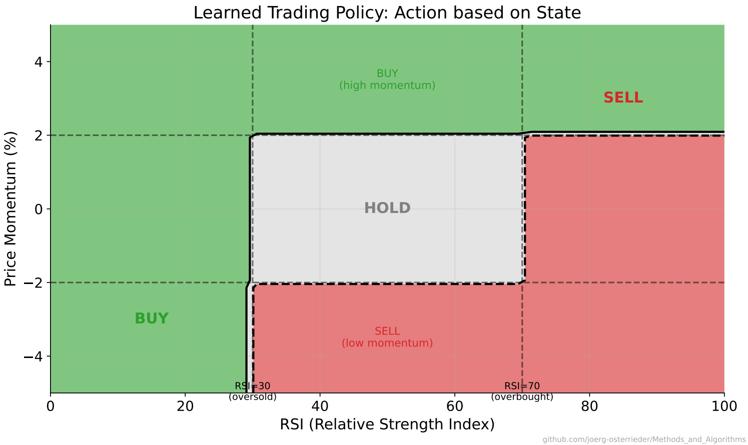
github.com/joerg-osterrieder/Methods_and_Algorithms

Q-values show expected reward from each state-action



github.com/joerg-osterrieder/Methods_and_Algorithms

RL agents improve through exploration and exploitation



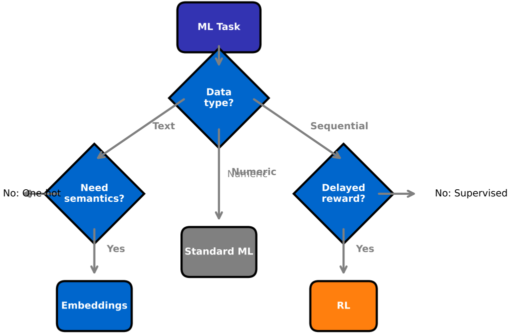
Policy maps states to actions (when to buy/sell/hold)

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/` [TBD]

When to Use Embeddings vs RL



Embeddings: Text, categorical -> dense vectors (Word2Vec, BERT)

RL: Sequential decisions with delayed rewards (trading, games)

github.com/joerg-osterrieder/Methods_and_Algorithms

Embeddings for text, RL for sequential decisions with delayed rewards