Check list

- [ ] Microphone turned on
- [ ] Zoom room open
- [ ] MyBinder launched
- [ ] Sound Volume on

SSIE 419/519: Applied Soft Computing

Lecture 01: Introduction & Word Embeddings

Sadamori Kojaku

Binghamton University

EngiNet™

State University of New York

WARNING

Contact Information:

EngiNet Office Staff:
Janice Kinzer
Email: enginet@binghamton.edu
Phone: 1-800-478-0718 or 607-777-4965

Media Production Operator: Ryan Luo

Instructor: Sadamori Kojaku
Email: skojaku@binghamton.edu

Course Overview

- **Instructor:** Sadamori Kojaku (幸若完壮)
- **Email:** skojaku@binghamton.edu
- **Office Hours:** Tue & Thu 15:00-17:00
- **Lecture Note:** https://skojaku.github.io/applied-soft-comp
- **GitHub:** https://github.com/skojaku/applied-soft-comp

Why Soft Computing?

Jason Allen
Pueblo West
Théâtre D'opéra Spatial

$750

Colorado State Fair

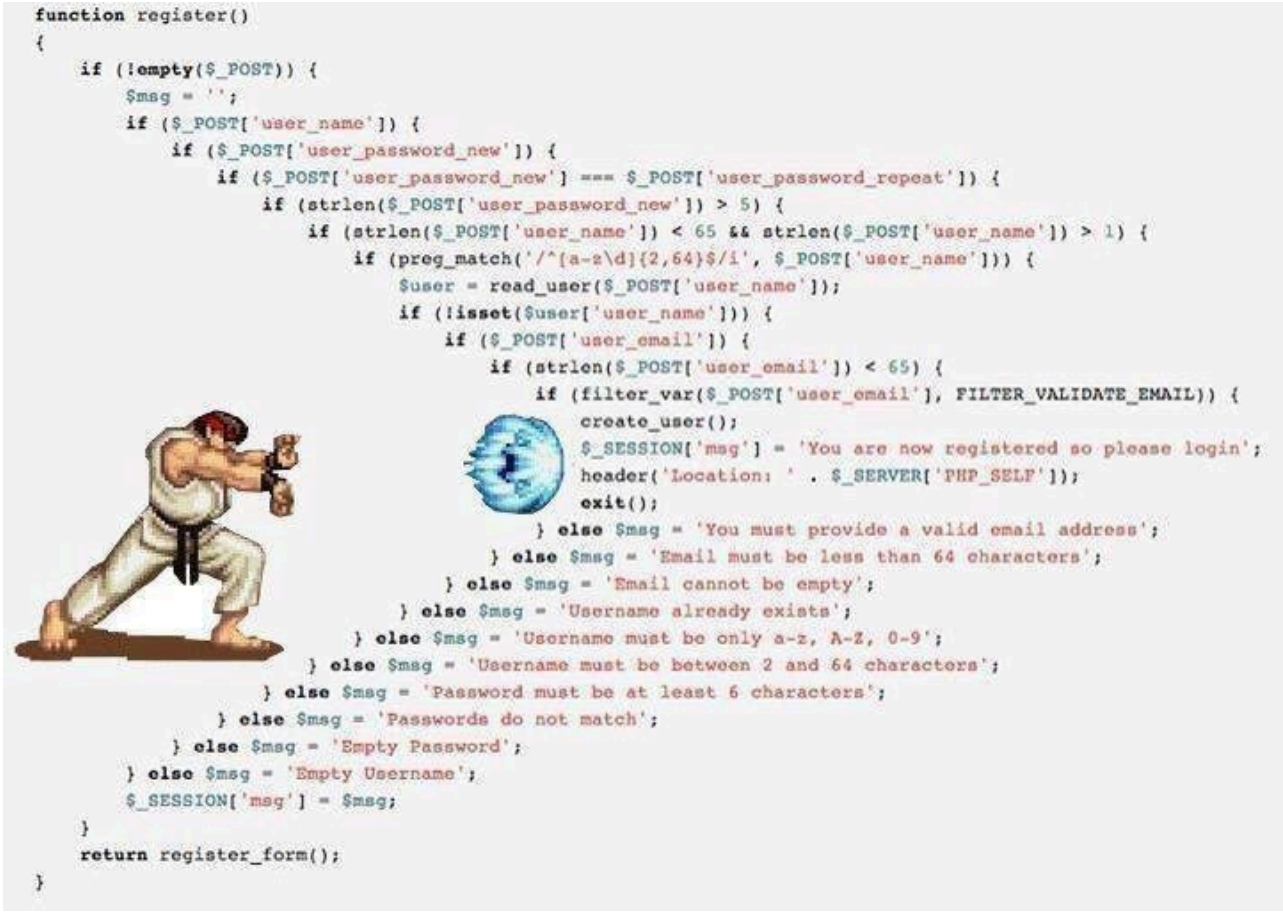
People that do not exist

ThisPersonDoesNotExist.com

How is it possible?

60s-80s: Expert System

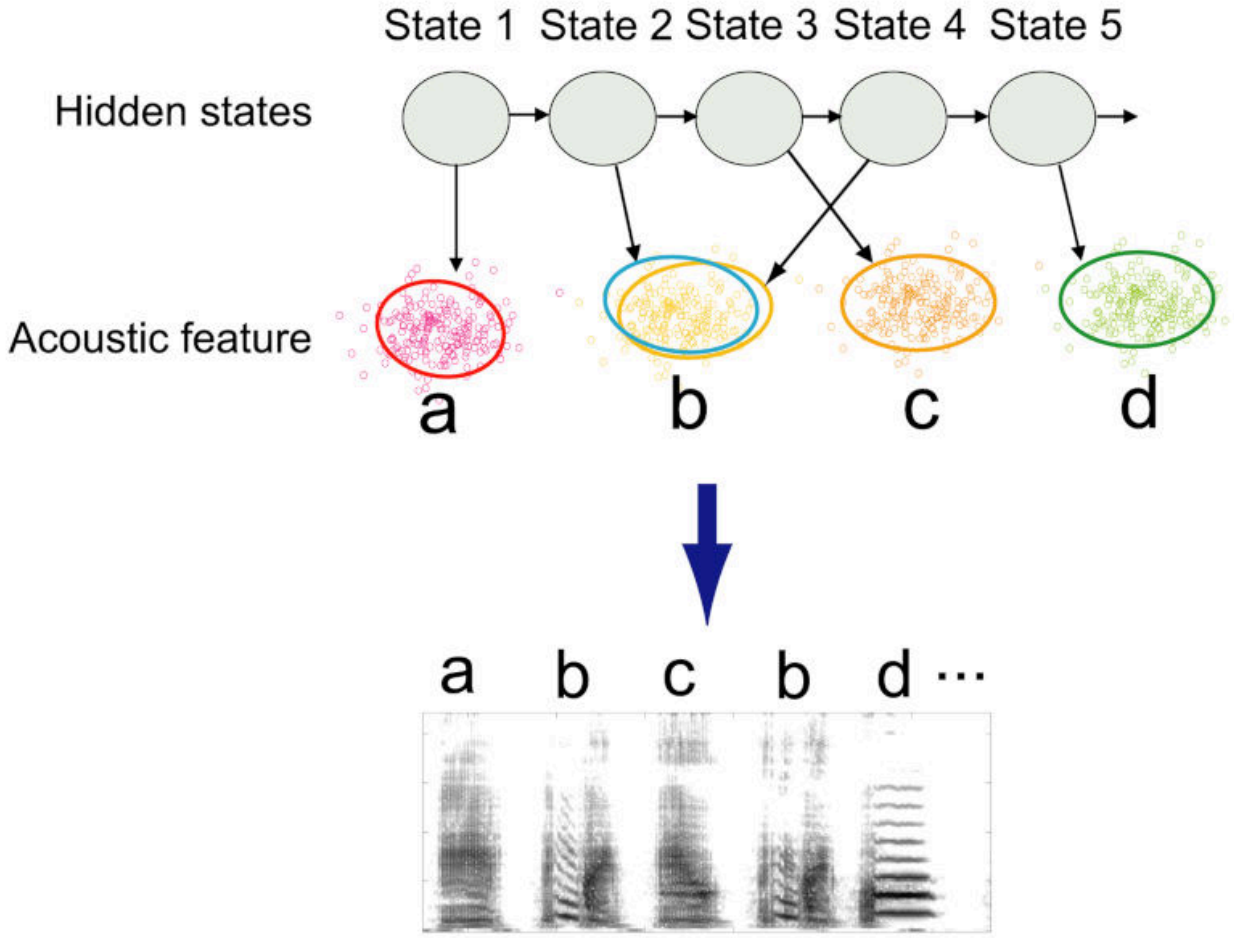
- Explicit rules
- Examples:
  - MYCIN (1976) for medical diagnosis
- Issues:
  - No adaptability
  - Need a lot of rules for complex tasks



```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```

90s-2000s: Statistical Learning

- Allow some randomness in real-world data
- Examples:
  - Support Vector Machine
  - Random Forest
- Issues:
  - Need a lot of data
  - No cross-domain generalization



2010s-2020s: Deep Learning 🧠

- Learn patterns from examples
- Adapt to new situations
- No need for explicit rules
- Handle complexity naturally

Neural networks are not new

- Warren McCulloch and Walter Pitts to develop the concept of the McCulloch-Pitts (MCP) neuron in 1943.
- Frank Rosenblatt later introduced the perceptron learning rule for the MCP.

Perceptron Research from the 50's & 60's, clip

Neural networks were virtually dead until 2012

- Computing limitations - computers were too slow to train meaningful neural networks
- Data scarcity - Neural networks require massive datasets that weren't available
- Vanishing gradient - Prevented training deep architectures effectively
- Better alternatives (e.g., SVMs) outperformed neural networks with less overhead
- AlexNet (2012) represents a breakthrough in deep learning

Language Understanding: ChatGPT 💭

- Natural conversation abilities
- Multiple capabilities:
  - Poetry and creative writing
  - Technical explanations
  - Code assistance
  - Understanding context and humor

Scientific Breakthrough: AlphaFold 🧬

- Solved 50-year protein folding problem
- Near-perfect accuracy
- Accomplished in days what took months in labs
- Revolutionary impact on biology and medicine

Game AI: The Divine Move 🎮

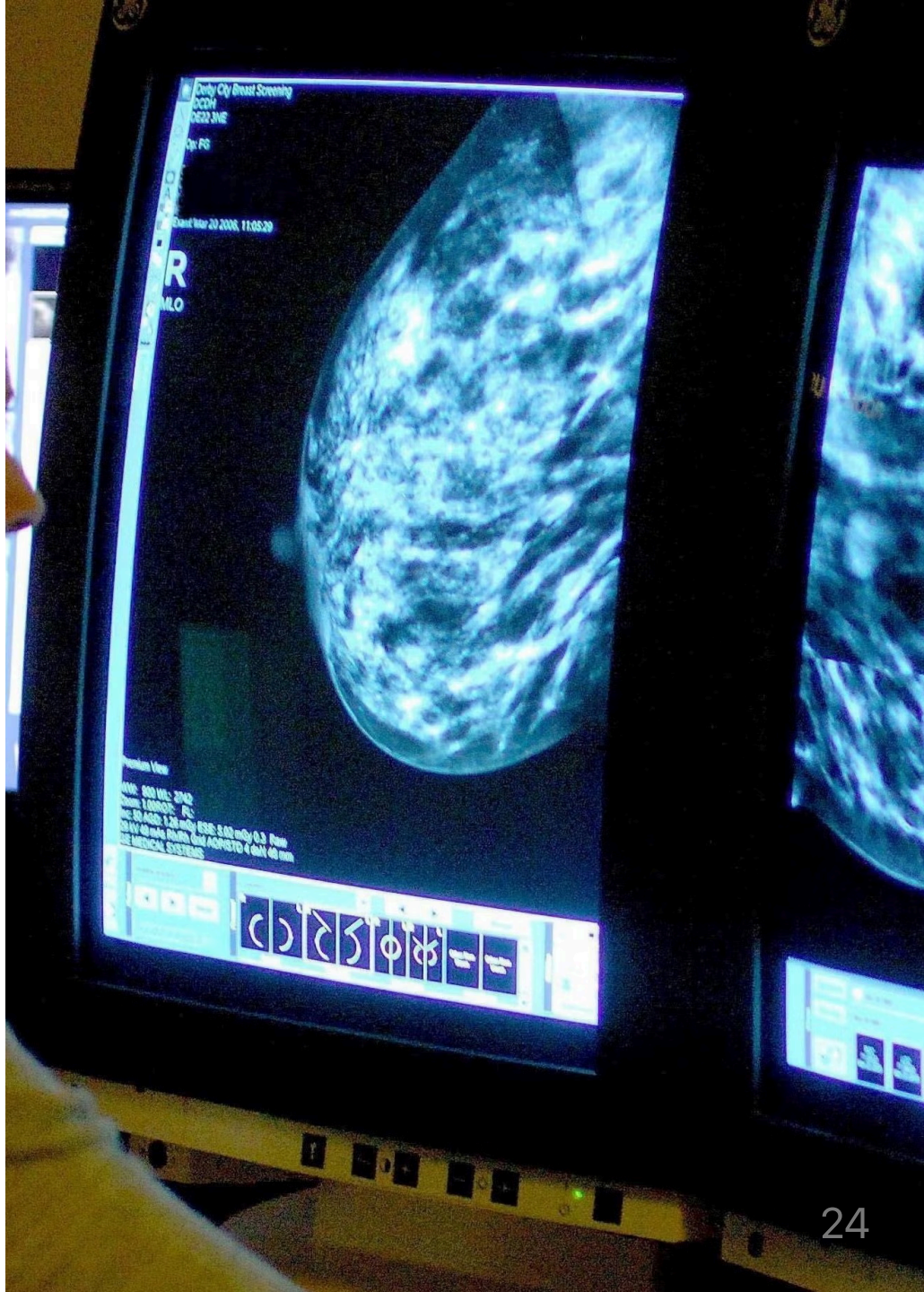
- AlphaGo vs Lee Sedol
- Famous "Move 37"
- Demonstrated creative thinking
- Showed AI can surpass human intuition

Video Generation: Sora 🎬

- Text-to-video technology
- 60-second realistic videos
- Physics-accurate scenes
- Multiple moving elements
- Photorealistic quality

Medical Diagnosis: AI Assistant 👨‍⚕️

- Surpasses human accuracy in cancer detection
- Reduces:
  - Missed cases
  - False alarms
- Augments doctor's capabilities



Autonomous Systems: Self-Driving 🚗

- Real-time sensor processing
- Faster than human reactions
- Constant alertness
- Improved safety in many conditions

Why This Matters 🌟

- Unprecedented pace of breakthroughs
- Solving decades-old problems
- Surpassing human capabilities
- Transforming multiple fields:
  - Healthcare
  - Transportation
  - Scientific research
  - Creative arts

Join the revolution in applied soft computing!

About this course

Course Structure

"Don't think! Feeeeeel" - Bruce Lee

- 🎓 Lectures
- 🛠️ Hands-on exercises
- 📝 Weekly quizzes
- 💻 Biweekly coding assignments
- 🎓 Final project
- 📝 Exam

Final Project 🎓

- Individual project (30% of grade) 📊
- Timeline 📅
  - 03/09: Project Proposal
  - 05/06: Project Presentation
  - 05/09: Project Final Paper
- Requirements 📋
  - Apply concepts to real problem 🌍
  - Show course integration 🧠
  - Clear presentation 🗣️

Exam

- 📚 Final exam on all topics (weight: 30%)
- 📅 During exam week
- 📝 Theory + practical problems
- 🌍 Apply concepts to real scenarios
- 📚 Review sessions before exam

Weekly Quiz on Brightspace

- 📊 Quizzes: A tool to identify misconceptions (weight: 20%)
- 🧠 Covers previous week's topics
- 🏁 Deadline: before final exam
- 🔄 Unlimited attempts until correct

Assignment

- 📅 One assignment per module (weight: 20%)
- 💻 Coding exercises
- 🤖 Autograded assignments
- 🏁 Deadline: before final exam
- 🔄 Unlimited attempts until correct

Lecture note

- 📚 Interactive Jupyter book
- 💻 Run code directly on the page
  - ⏳ First-time loading may take 2-3 mins
- 🔄 Or download as Jupyter notebook
  - ☁️ Use on cloud (Google Colab, Kaggle) or locally
  - 📦 Install packages from `environment.yml` for local use
  - See The course GitHub repo for details
- Slides: The course GitHub repo

Policy

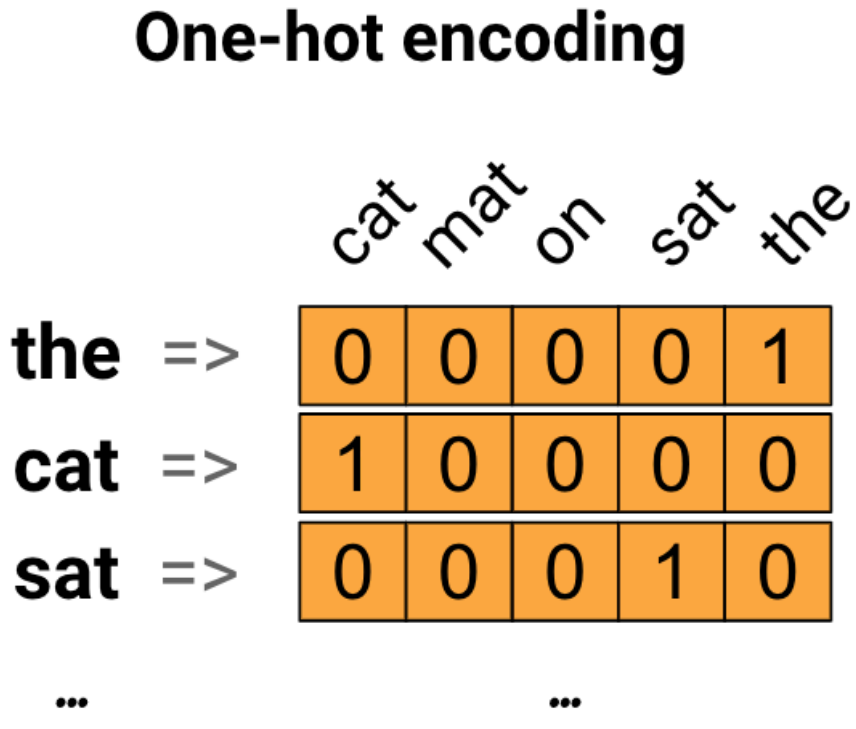
- 📚 3-credit course: 6.5+ hours of work/week outside class
- 🤖 AI tools allowed for learning, but cite if used in assignments
- 💾 Back up all data and code (loss not an excuse for late work)
- ♿ Accommodations available for students with disabilities
- 🚫 Zero tolerance for academic dishonesty

Questions?

Teaching computers how to understand words

The Challenge: Teaching Computers Language 🧮

- Computers only understand numbers
- Words need to be translated into numerical form
- Early approach: One-hot encoding
  - Each word gets a unique binary vector
  - Example: cat → [1,0,0], dog → [0,1,0]
- Problem: No semantic meaning captured

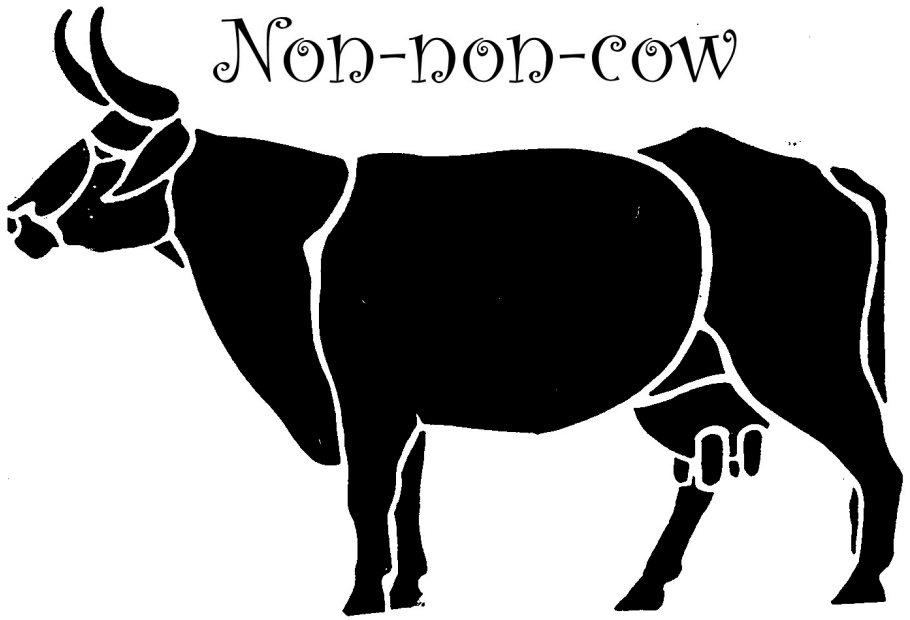


Distributional Hypothesis

Words that appear in similar contexts have similar meanings

A trivia

- Ancient Buddhist concept of Apoha (5th-6th century CE):
  - We understand concepts by what they are not
  - Example: A "cow" is defined by everything that is not a cow
- Parallels with modern distributional semantics:
  - Both define meaning through relationships between concepts
  - Words understood by their contrasts with other words



Another trivia

My grandma called me in many different ways

- Sadamori
- My son
- My nephew
- My niece
- My cousin
- Her dog

How can we build such a distributional representation?

Pen and Paper exercise

From Words to Numbers: TF-IDF 📊

- Term Frequency (TF):

$$TF(t, d) = \frac{\text{count of term t in doc d}}{\text{total terms in doc d}}$$

- Inverse Document Frequency (IDF):

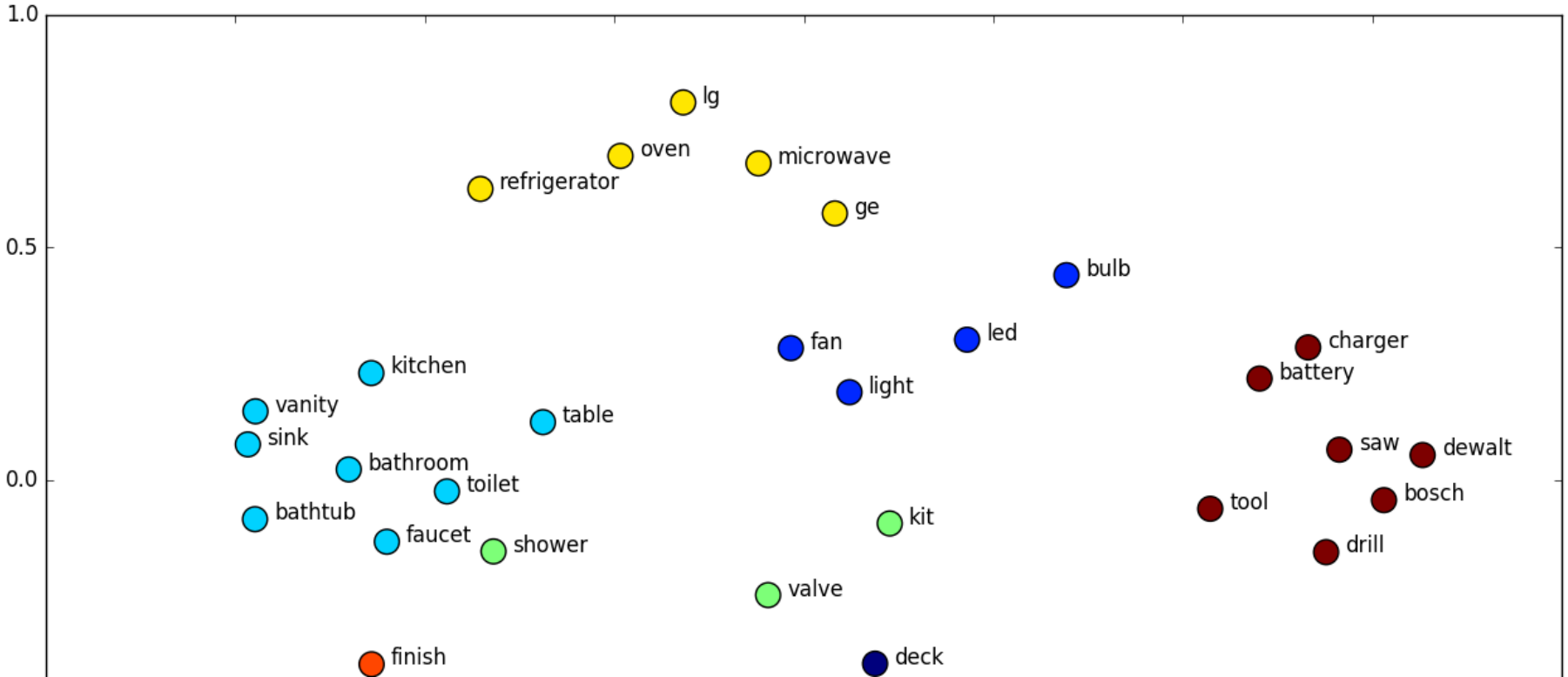
$$IDF(t) = \log \frac{\text{total documents}}{\text{docs containing t}}$$

- Combined score:

$$TF\text{-}IDF(t, d) = TF(t, d) \times IDF(t)$$

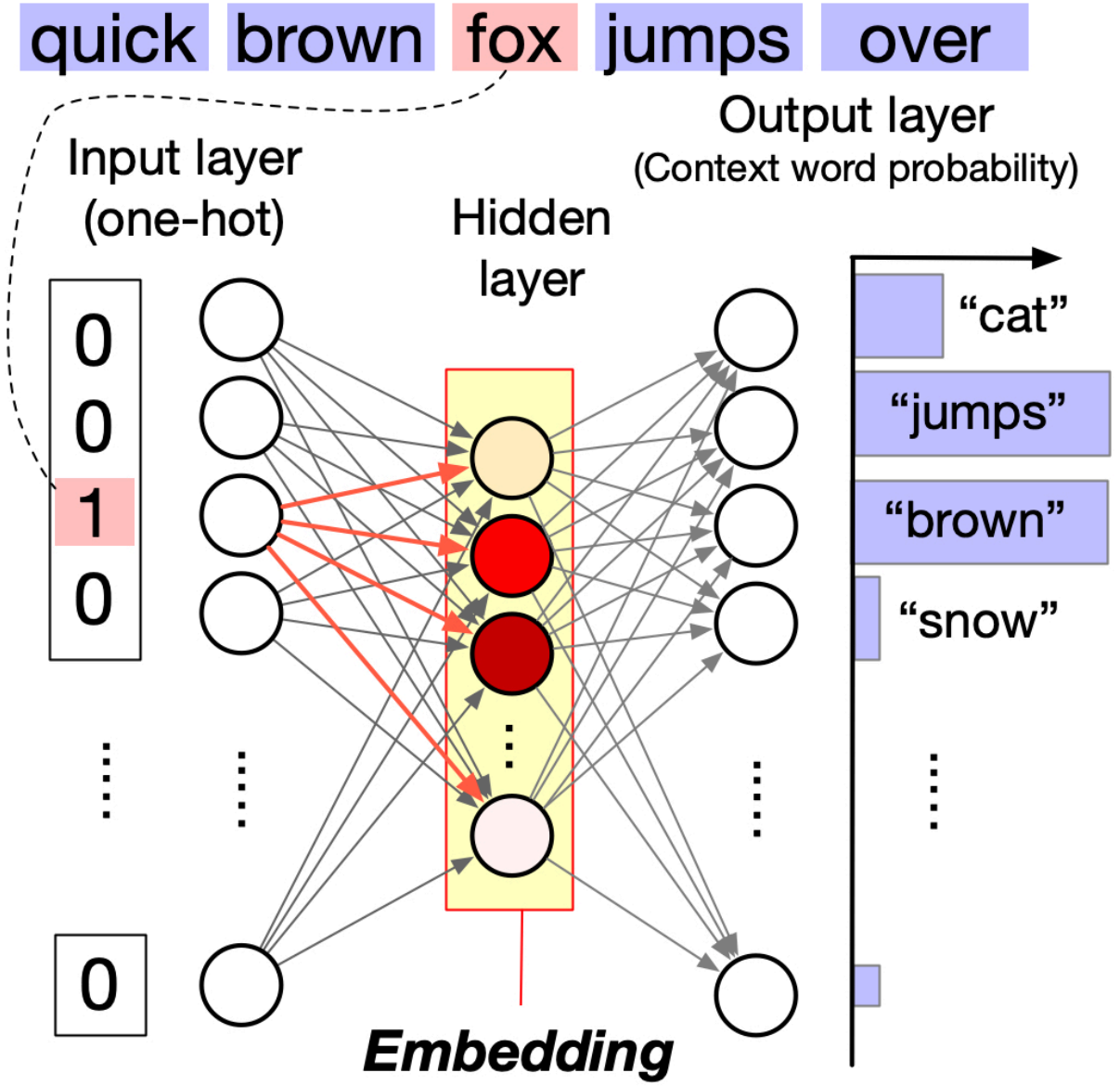
The Distributional Hypothesis 🌐

> "You shall know a word by the company it keeps"
> ~Words appearing in similar contexts have similar meanings~



Word2Vec: Neural Word Embeddings 🧠

1. Consit of one hidden layer without non-linear activation
2. Output layer is a softmax layer
3. Two models: Skip-gram and CBOW depending on the input and output

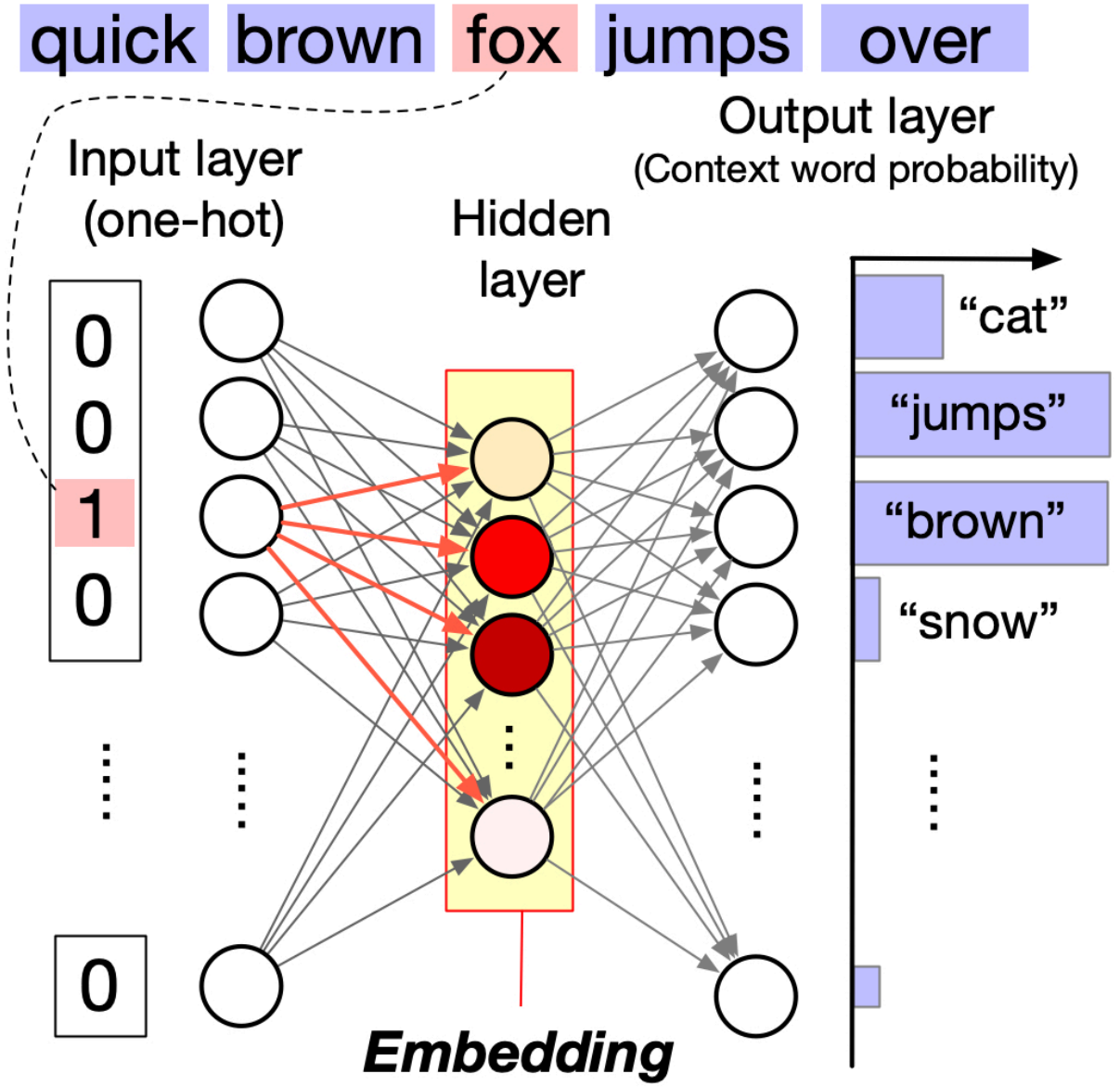


Model (Skip-gram)

$$P(\text{context word } j \mid \text{center word } i) = \frac{1}{Z}\exp(v_j^T u_i)$$

Each word has **two vectors**, v and u:

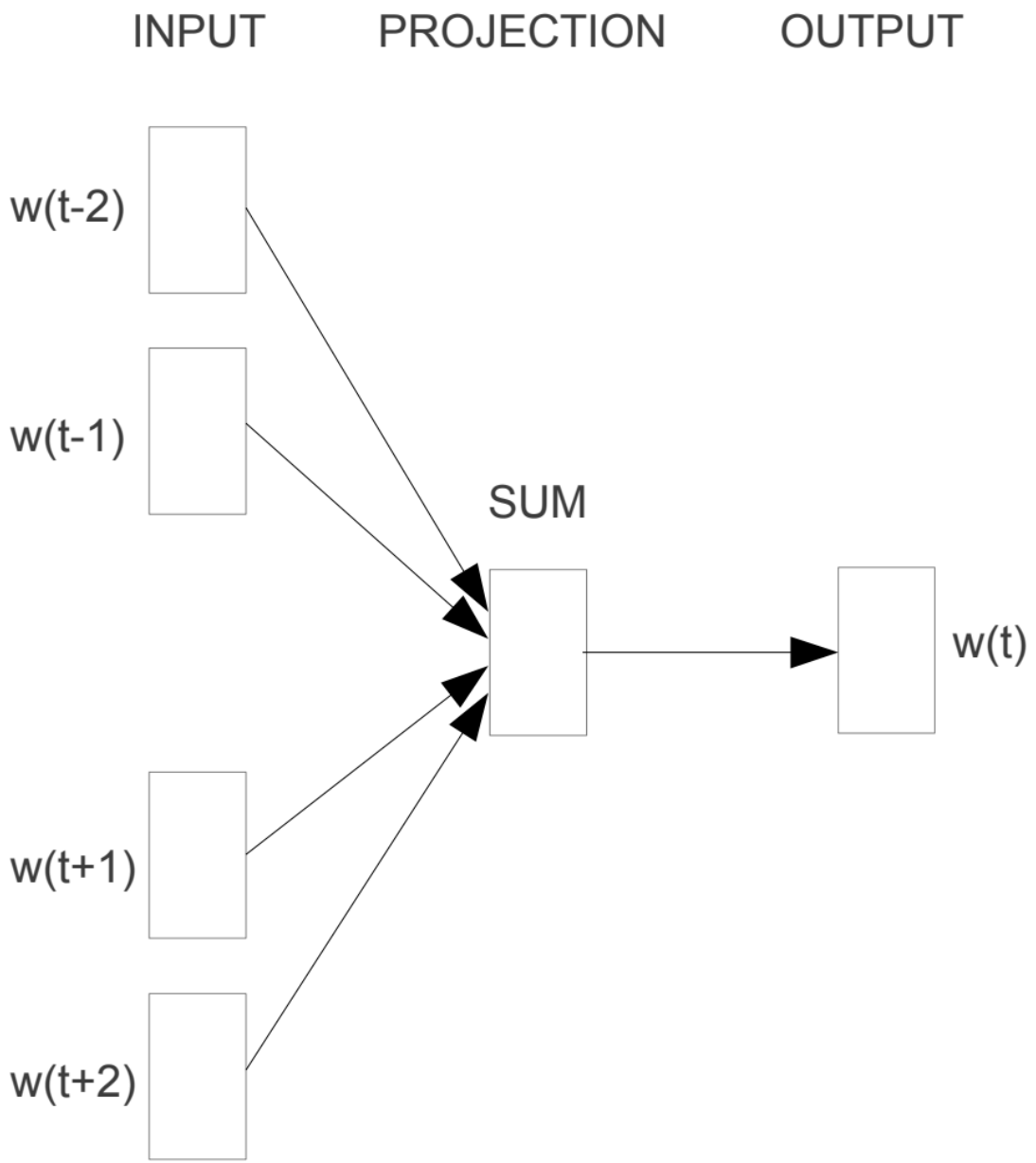
- In-vector: u_i represents word i as a center word
- Out-vector: v_j represents word j as a context word
- Z is normalization constant



Model (CBOW)

$$P(w_c|w_1,\ldots,w_C) = \frac{\exp(v_{w_c}^T \bar{v})}{\sum_{w\in V}\exp(v_w^T \bar{v})},$$

where $\bar{v} = \frac{1}{C}\sum_{i=1}^{C} v_{w_i}$ is the average of the context word vectors.



CBOW

Matrix Factorization Connection 🔢

Word2vec implicitly factorizes a Pointwise Mutual Information (PMI) matrix:

$$M_{ij} = \log \frac{P(w_i, w_j)}{P(w_j)P(w_j)}$$

Properties:

- Low when words appear independently
- High when words co-occur frequently
- Similar effect to TF-IDF normalization

Word embeddings preserve PMI values:

$$v_{w_i}^\top v_{w_j} \approx M_{ij}$$

Key insight: Words that frequently appear in similar contexts will have similar embeddings!

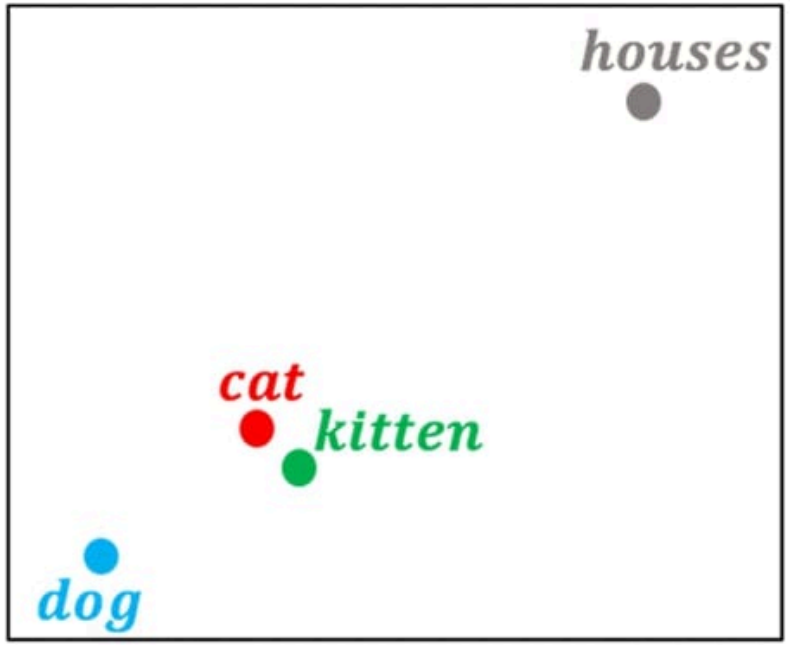
GloVe: Global Vectors 🌐

GloVe explicitly factorizes PMI matrix:

$$M_{ij} = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$$

	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8

Dimensionality reduction of word embeddings from 7D to 2D



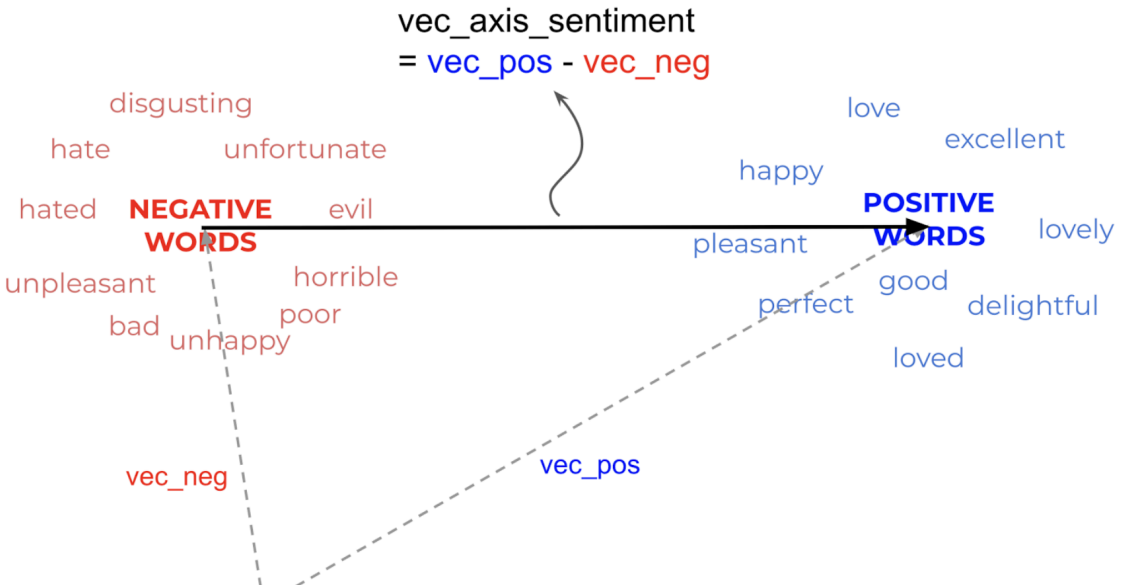
Let's build the first word embedding using TF-IDF and word2vec

Jupyter notebook

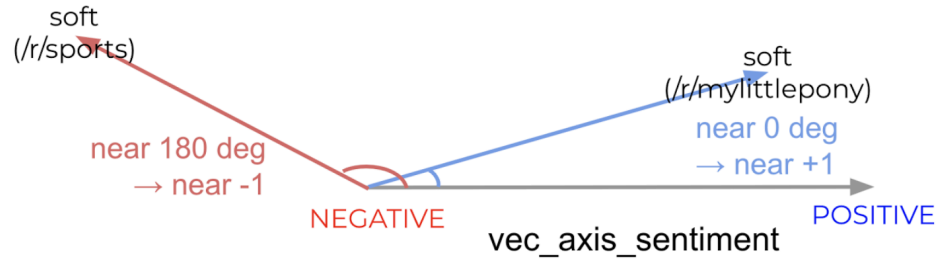
SemAxis: Understanding Word Relationships 📐

- Identify two semantically interpretable clusters
- SemAxis: An axis between the cluster centroids, e.g., good-bad, soft-hard
- Project word vectors onto the SemAxis

Step 1: Define a semantic axis



Step 2. Locate words on the semantic axis

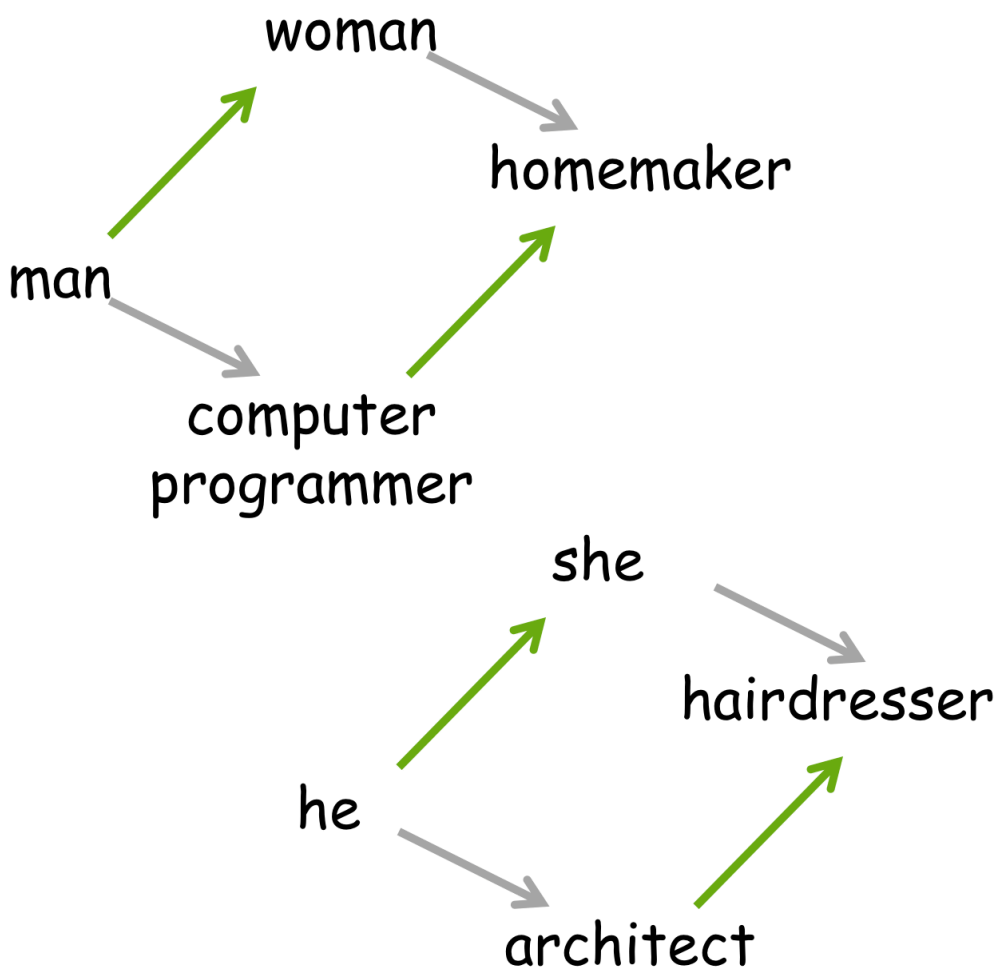


Compute a cosine similarity between a word vector and axis vector

Bias in Word Embeddings ⚖️

Gender bias example:

- man : doctor :: woman : nurse
- he : programmer :: she : homemaker



Let's build SemAxis

Jupyter notebook

Doc2Vec: From Words to Documents 📄

Two models:

1. PV-DM (Distributed Memory)
2. PV-DBOW (Distributed Bag of Words)

Doc2Vec (PV-DM)

PV-DM probability:

$$P(w_i|w_{i-k},\ldots,w_{i-1},d) = \frac{\exp(u_{w_i}^T h)}{\sum_{w\in V}\exp(u_w^T h)}$$

Where h is either:

- Average: $h = \frac{1}{k+1}(v_d + \sum_{j=i-k}^{i-1} v_{w_j})$
- Concatenation: $h = (v_d, \sum_{j=i-k}^{i-1} v_{w_j})U$

Doc2Vec (PV-DBOW)

PV-DBOW probability:

$$P(w_i|d) = \frac{\exp(u_{w_i}^T v_d)}{\sum_{w \in V} \exp(u_w^T v_d)}$$

Where v_d is the document vector.

Let's build Doc2Vec

Jupyter notebook

Summary: Evolution of Word Embeddings 🎓

1. One-hot encoding → No semantics
2. TF-IDF → Document-level patterns
3. Word2Vec → Local context patterns
4. Doc2Vec → Document embeddings

Assignment on GitHub: