Word Embeddings

Applied Text Mining

Dr. Maryam Movahedifar

14–17 July 2025

University of Bremen, Germany
movahedm@uni-bremen.de

Universität Bremen

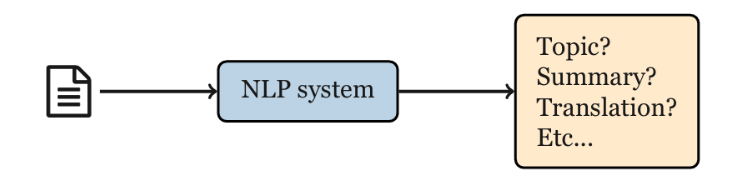
DATA SCIENCE CENTER

Outline

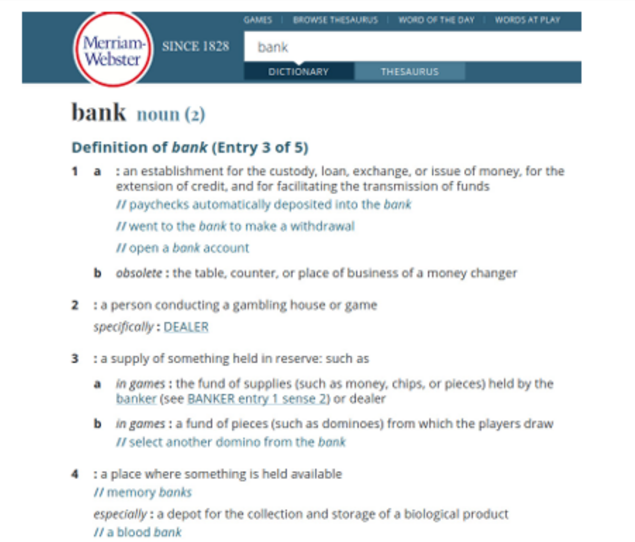
Introduction to Word Representations

What is NLP and Why Represent Words?

- NLP enables tasks like summarization, translation, and classification.
- Key challenge: How do we represent the meaning of words computationally?
- Use cases:
    - Compute similarity between words (e.g., *cat* vs. *dog*)
    - Understand documents and sentences



Traditional Dictionaries



GAMES | BROWSE THESAURUS | WORD OF THE DAY | WORDS AT PLAY

Merriam-Webster SINCE 1828

bank

DICTIONARY THESAURUS

bank noun (2)

Definition of *bank* (Entry 3 of 5)

1 a : an establishment for the custody, loan, exchange, or issue of money, for the extension of credit, and for facilitating the transmission of funds

// paychecks automatically deposited into the *bank*

// went to the *bank* to make a withdrawal

// open a *bank* account

b *obsolete* : the table, counter, or place of business of a money changer

2 : a person conducting a gambling house or game

specifically : DEALER

3 : a supply of something held in reserve: such as

a *in games* : the fund of supplies (such as money, chips, or pieces) held by the banker (see BANKER entry 1 sense 2) or dealer

b *in games* : a fund of pieces (such as dominoes) from which the players draw

// select another domino from the *bank*

4 : a place where something is held available

// memory *banks*

especially : a depot for the collection and storage of a biological product

// a blood *bank*

bank Noun

- **bank** (sloping land near water) — "they pulled the canoe up on the bank"; "he sat on the bank of th
- Depository financial i t the bank"; "that bank holds the mortgage o
- ...

Verb

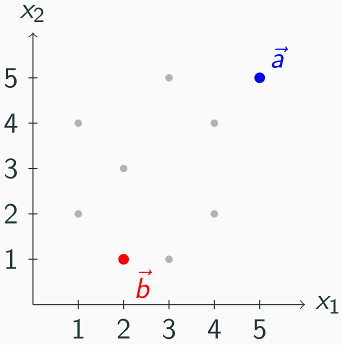
- **bank** (tip laterally) — ft"
- **bank** (do business wi in this town?"
- ...

Unfortunately, dictionaries and knowledge bases are hard to maintain and have limited coverage

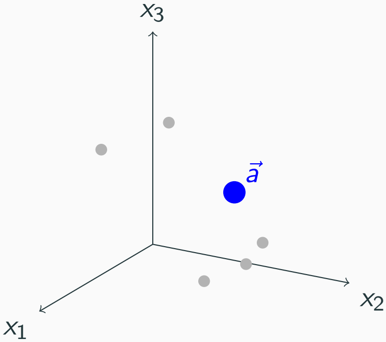
https://wordnet.princeton.edu

Vector Space Models

Vector Representations



2D space: vectors $\vec{a} = [5, 5]$, $\vec{b} = [2, 1]$



3D space: $\vec{a} = [2, 4, 3]$ (scaled here)

Words as Vectors

Key idea: Represent words as vectors to capture meaning.

- Similar words have similar vectors—close together in space.
- Vector representations should:
  - Capture meaning (semantics)
  - Reflect relationships (e.g., analogies)
  - Be efficient and interpretable
- Example vectors: `cat = [0.5, 0.8, ...]`, `dog = [0.3, 0.7, ...]`
- **Cosine similarity** measures how close vectors are.

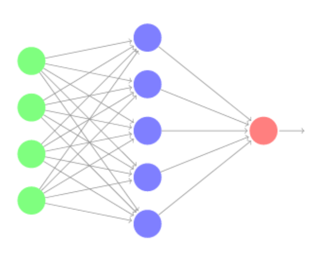
Similarity Example (SimLex-999)

smart vs. **intelligent** $\rightarrow$ **9.2** (very similar) (0 = not similar, 10 = very similar)
easy vs. **big** $\rightarrow$ **1.12** (not similar)

How Are Word Vectors Used?

In Neural Networks

- Text classification
- Sequence tagging
- Machine translation



As Research Objects

- Word meaning
- Semantic change
- Language variation

cat	0.52	0.48	-0.01	$\cdots$	0.28
dog	0.32	0.42	-0.09	$\cdots$	0.78

Cosine similarity helps find similar words:

dog $\rightarrow$ cat, cow, horse **car** $\rightarrow$ vehicle, driver, race

Exercise: Exploring Word Vectors (5 min)

- Go to https://projector.tensorflow.org/
- The site should load the **Word2Vec 10K** vectors by default (check left panel).
- Use the search bar (top right) to explore word neighborhoods:
  - What are the 5 nearest words to **cat**?
  - What are the 5 nearest words to **computer**?

One-Hot Encoding of Words

What is One-Hot Encoding?

Each word is assigned a unique integer ID. For example, `cat` (3), `dog` (5).
The vector representation is mostly zeros, except a 1 at the position of the word's ID.

cat	0	0	1	0	0	0	0
dog	0	0	0	0	1	0	0
car	0	0	0	0	0	0	1

Limitations:

- No semantic meaning: similar words have completely different vectors.
- Very high-dimensional and sparse vectors.
- No relationships or patterns captured between words.

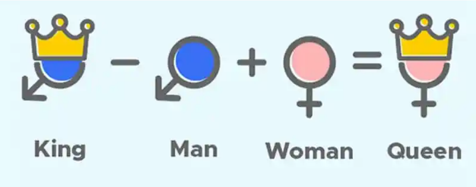
Word embeddings

What Are Word Embeddings?

Word Embeddings: dense, low-dimensional vectors capturing word meanings and relationships.

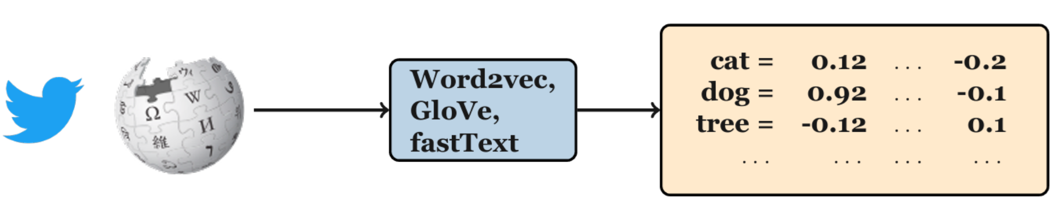
Key characteristics:

- Map words to continuous vectors (e.g., 100–300 dimensions)
- Similar words have similar vectors (close in vector space)
- Capture semantic and syntactic relationships (e.g., analogies)
- Learned from large text corpora using models like Word2Vec or GloVe



Learning Word Embeddings

Popular models to learn word embeddings include Word2Vec, GloVe, and fastText.
These models map words like `cat`, `dog`, and `tree` to dense vectors.



Training Word Embeddings

How can we train a model to learn word meanings?

- **Key idea:** Use text itself as training data — a form of self-supervision.
- Train a neural network to predict the next word given previous words (language modeling).
- This approach lets the model learn word meanings and relationships without labeled data.

Exercise: Word Prediction Task

- Yesterday I went to the **?**
- A new study has highlighted the positive **?**

Question

Which word comes next?

Word2Vec: Training Tasks and Methods (Context-based)

CBOW (Continuous Bag of Words)

Predicts the current word using the surrounding context words.
Example: Given "The cat __ on the mat," predict the missing word.

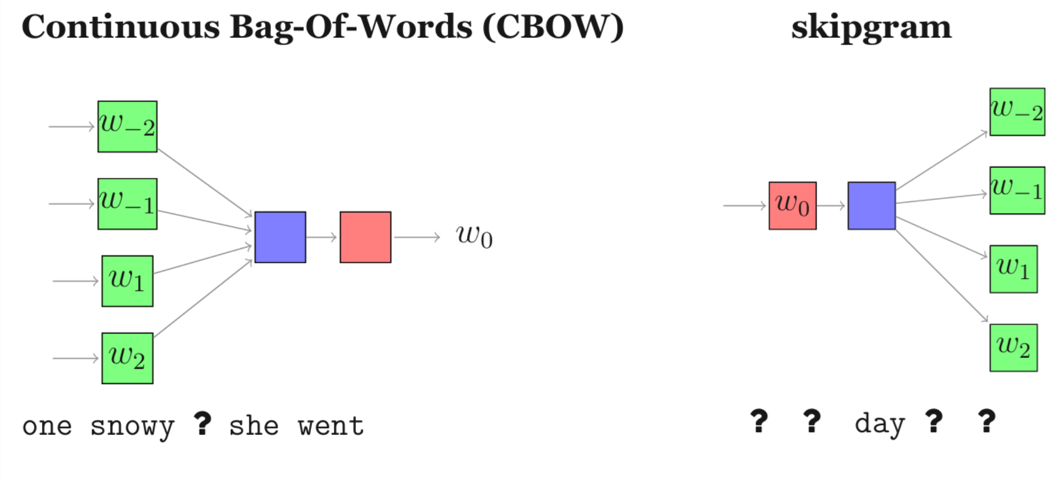
Skip-gram

Predicts surrounding context words given the current word.
Example: Given "cat," predict words like "the," "sat," "on."

Training Regimes

- **Hierarchical Softmax:** Uses a tree structure to efficiently compute probabilities, reducing computation for large vocabularies.
- **Negative Sampling:** Samples a small number of "negative" words instead of all, speeding up training.

Word2Vec Overview



Visual summary of Word2Vec training tasks: CBOW and Skip-gram.

Word2Vec: Skip-gram Model and Probability

Goal: Given a target word (e.g., **cat**), predict the surrounding **context words** within a window (e.g., size = 5) in the following example.

The	domestic	**cat**	is	a	small,	typically	furry
c_1	c_2	w	c_3	c_4	c_5	c_6	c_7

The model learns by computing similarity and converting it to a probability:

- For each word pair (w, c), compute similarity using dot product: $w \cdot c$
- Convert similarity into probability with a sigmoid, which gives the probability that c is a true context word of w:

$$P(+|w, c) = \frac{1}{1 + e^{-w \cdot c}}$$

- The model adjusts vectors to increase this probability for true pairs and decrease it for false ones

fastText: Subword-Level Embeddings

Problem with Word2Vec:

Struggles with rare or unseen words — it treats every word as an atomic unit.

> **fastText's Solution:** Leverage subword information!
> Each word = the sum of its character **n-gram embeddings** + the word itself.

How does it work? Word boundaries are marked with < and >, then split into overlapping character n-grams.

Example: word = *where*, $n = 3$

- Character n-grams: `<wh`, `whe`, `her`, `ere`, `re>`
- Also includes: `<where>`

Final Word Embedding:

Sum of all n-gram vectors → captures word structure and generalizes better to unseen words.

GloVe: Learning from the Big Picture

What's different about GloVe? Instead of just focusing on local context like Word2Vec, GloVe captures how words relate across the *entire corpus*, using **global co-occurrence statistics**.

Step 1: Build a **word-word co-occurrence matrix**

- Each cell counts how often word i appears near word j
- Captures broad patterns, e.g., "ice" co-occurs with "cold," and "fire" with "hot"

Step 2: Train word vectors so that:

$$w_i^\top w_j \approx \log(\text{co-occurrence}(i, j))$$

Why it works: Combines **meaning** and **frequency**, great for learning analogies and capturing rare word relationships.

Properties of Word Embeddings: Analogies (Conceptual)

We can explore **semantic relationships** in the vector space through analogies:

$$king - man + woman \approx queen$$

This reflects the idea that embeddings capture meaning through geometric patterns. Similar relationships (gender, tense, plural forms) often form consistent vector directions.

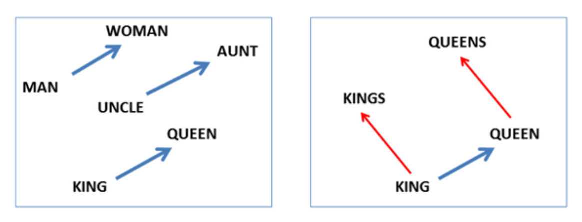


Figure: Visualizations from Mikolov et al. (2013) showing analogy patterns.

Properties of Word Embeddings: Analogies (Numeric Example)

Let's break down the vector arithmetic of an analogy:

$$king - man + woman = queen$$

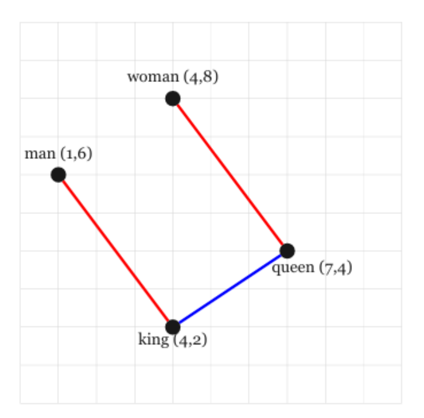
Step-by-step:

$$\text{king} - \text{man} = [4, 2] - [1, 6] = [3, -4]$$

$$[3, -4] + \text{woman} = [3, -4] + [4, 8] = [7, 4] \approx \text{queen}$$

$\Rightarrow$ The vectors used:

- king $=$ [4, 2]
- man $=$ [1, 6]
- woman $=$ [4, 8]



Evaluation

Intrinsic Evaluation: How Good Are Our Embeddings?

Goal: Test embeddings directly before plugging them into full tasks. These are quick checks to see what kind of information the vectors capture.

1. Similarity — Do similar words have similar vectors?
Example: "car" and "automobile" should be close together.
How? Compare cosine similarity with human ratings.

2. Analogies — Can the model solve word puzzles?
Example: king - man + woman = queen
How? If the relationships are encoded in the vectors, simple arithmetic should reveal them.

3. Probing Classifiers — What linguistic features are inside?
Example: Can a tiny model guess POS tag from the embedding?
Why? Tests if grammar or syntax info is encoded.

Intrinsic Evaluation: Similarity

Cosine similarity measures the angle between two vectors — not their length:

$$\text{cosine}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

Interpretation:

- 1: same direction $\rightarrow$ high similarity
- 0: orthogonal $\rightarrow$ no relation
- -1: opposite direction $\rightarrow$ opposite meaning

Why cosine instead of Euclidean distance?

- Cosine focuses on direction, not magnitude
- Word vectors differ in length — cosine captures semantic similarity better

Intrinsic Evaluation: Spearman Correlation

To evaluate how well embeddings reflect human intuition, we use **Spearman correlation**:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Definitions:

- d_i: difference between ranks (human vs model)
- n: number of word pairs

Procedure:

1. Collect human similarity scores for word pairs
2. Compute cosine similarities from embeddings
3. Rank both sets and compute Spearman's ρ

Higher ρ means better alignment with human judgments.

Intrinsic Evaluation: Analogies

Analogies test if embeddings capture semantic and syntactic relationships:

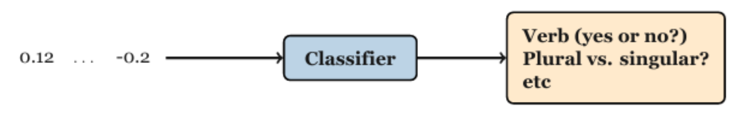
$$\text{king} - \text{man} + \text{woman} \approx \text{queen}$$

Procedure:

- Compute vector arithmetic on word embeddings
- Find the closest word vector to the resulting vector
- Evaluate accuracy on benchmark analogy datasets (e.g., Google Analogy Test Set)

Intrinsic Evaluation: Probing Classifiers

Also called **diagnostic classifiers**



Mostly used to evaluate **sentence embeddings**, but sometimes also for analyzing **word embeddings**.

Caution: Performance might seem high, but the classifier may learn unrelated signals (e.g., word frequency, part-of-speech) instead of the intended linguistic property.

Biases in word embeddings

Biases in Word Embeddings

What is bias?

Bias in word embeddings means unfair associations learned from data — for example, associating the word **"doctor"** more with **"he"** and **"nurse"** more with **"she"**.

Why measure bias?

- Quantify and understand social biases (e.g., gender, race) in embeddings.
- Evaluate effectiveness of bias mitigation methods.
- Examine how NLP models reflect or amplify societal prejudices.
- Reveal societal trends captured in text data.

Common methods: Measure associations between target and attribute words using tests like WEAT or SEAT.

Biases in Word Embeddings: Gender Analogies

We analyze biases by finding **gender analogies** aligned with a *seed direction* (e.g., *she–he*).

$$S_{(a,b)}(\mathrm{x},\mathrm{y}) = cos(\underbrace{\mathrm{a}}_{\text{embedding}_{she}} - \underbrace{\mathrm{b}}_{\text{embedding}_{he}}, x-y) \quad if \quad \underbrace{\|x-y\|_2}_{L_2 \text{ distance}} \leq \delta$$

Goal:

Find word pairs whose vector difference aligns with the gender direction and are semantically close.

Gender-appropriate analogies

queen	king
sister	brother
ovarian cancer	prostate cancer
mother	father
convent	monastery

Gender-stereotype analogies

nurse	surgeon
sassy	snappy
cupcakes	pizzas
lovely	brilliant
vocalist	guitarist

Application: analysis of semantic change

Application: Semantic Change Analysis

Goal: Detect how word meanings evolve over time using word embeddings trained on historical corpora.

Method Overview:

- Train embeddings on texts from different time periods (e.g., 1900s vs. 2000s)
- Align embeddings across time using *orthogonal Procrustes*
- Measure change via cosine distance between word vectors

Applications:

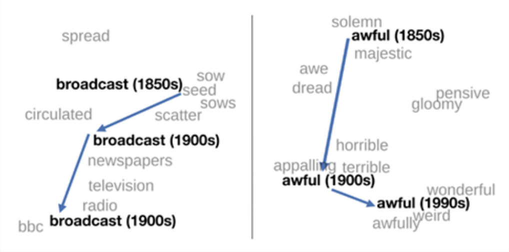
- Track technological shifts: *cloud, tablet, mouse*
- Study ideological or cultural change in media
- Support historical linguistics and lexicography

Semantic Change: Visual Examples

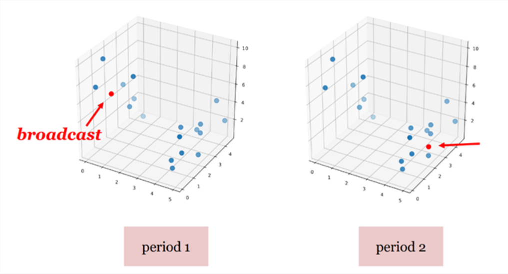
Examples of meaning change detected via word embeddings.

- Words like *broadcast* and *awful* exhibit strong shifts in meaning over decades.
- 2D projection (e.g., PCA or t-SNE) helps visualize drift in the embedding space.

Semantic Shift: Broadcast & Awful



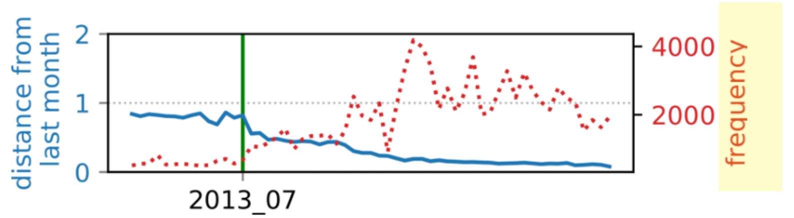
Tracking Change in Embedding Space



Semantic Change Case Study: glo

Example: Analyzing the emergence of new meaning for the word *glo* using word embeddings.

- *glo* gained new usage linked to rapper Chief Keef's 2013 track: "Gotta Glo Up One Day".
- Embedding-based methods can detect such emerging senses by measuring shifts over time.



Contextual word embeddings

Contextual Word Embeddings

Why Context Matters

Traditional word embeddings assign a *single vector per word type*, regardless of how the word is used. This limitation makes it hard to capture different meanings for words with multiple senses.

Key Idea

Contextual word embeddings generate a unique representation for each **word token** based on its surrounding context — enabling models to capture the precise meaning in every situation.

- **Static embeddings (e.g., Word2Vec):** One fixed vector per word, ignoring context nuances.
- **Contextual embeddings (e.g., BERT):** Dynamic vectors that adjust meaning based on context.

Contextual Word Embeddings: BERT

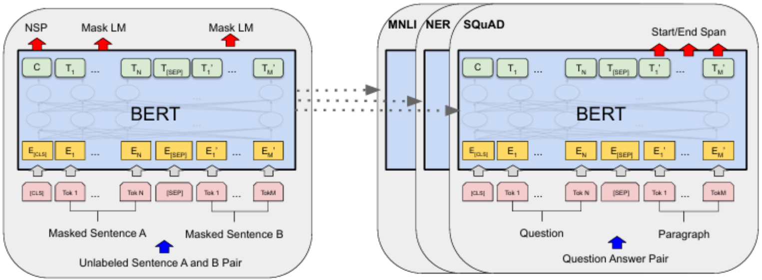
Key idea: Assign a unique embedding to each **word token**, derived from its **context**.

Traditional word embeddings use one vector per word type:

- "He went to the **bank** to deposit a check."
- "She sat by the **bank** of the river."

Training Objectives:

- Masked Language Modeling (MLM)
- Next Sentence Prediction (NSP)



Pre-training

Fine-Tuning

Practical