Word Embeddings

Marcel Bollmann

Department of Computer and Information Science (IDA)

LINKÖPING UNIVERSITY

Today's lecture

1. Semantics
   - Lemmas and Lexemes
   - Semantic Relations

2. Distributional Semantics
   - Vector Representations
   - Collocations

3. Vector Semantics
   - Cosine Similarity
   - Analogies

4. Learning Word Embeddings
   - Skip-gram
   - Example

5. Outlook

Semantics
or: The Meaning of Words

1

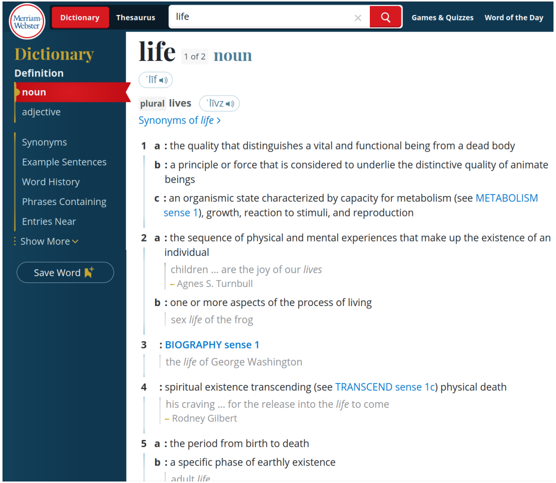
Lemmas and lexemes

- Different word forms can have the same fundamental **meaning**.

RUN : *run*, *runs*, *ran*, *running*

- A **lexeme** is the abstract meaning represented by a set of word forms.
  - "word sense"
- A **lemma** is the word form chosen to represent a given lexeme.
  - "dictionary form"

What is the meaning of “life”?



- **Word sense ambiguity**: One lemma can represent multiple lexemes.
- The lemma *life* in Merriam-Webster has:
  - **20 different meanings** as a noun
  - **4 different meanings** as an adjective

Source: Merriam-Webster

Polysemy and homonymy

- **Polysemy**: a word has multiple, semantically **related** meanings.
  - LIFE^1: "the quality that distinguishes a vital and functional being from a dead body"
  - LIFE^5: "the period from birth to death"
  - LIFE^8: "a vital or living being"

- **Homonymy**: a word has multiple, semantically ***un*related** meanings.
  - BASS^1: a type of fish
  - BASS^2: "the lowest adult male singing voice"

Semantic relations between word senses

- **Synonymy**
  two senses are (nearly) identical

$$\text{COUCH} \xleftrightarrow{\text{is synonym of}} \text{SOFA}$$

- **Antonymy**
  two senses are opposites of each other

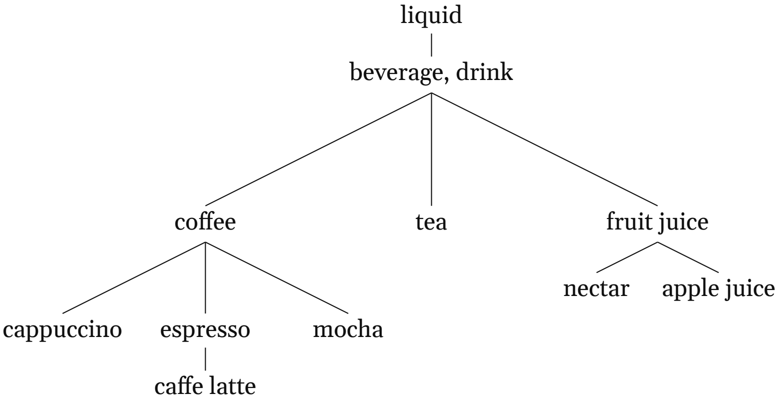
$$\text{HOT} \xleftrightarrow{\text{is antonym of}} \text{COLD}$$

- **Hyponymy**
  a sense is *more* specific than the other

- **Hypernymy**
  a sense is *less* specific than the other

$$\begin{tikzcd} \text{COFFEE} \arrow[rr, bend left, "\text{is hyponym of}"] & & \text{BEVERAGE} \arrow[ll, bend left, "\text{is hypernym of}"] \end{tikzcd}$$

A hierarchy of hypernyms



Source: WordNet

Semantic networks

- A **semantic network** – also called **knowledge graph** – is a collection of words and semantic relations between them.

- An example of a multilingual knowledge graph is ConceptNet.
  - Covers ten "core" languages with a combined vocabulary of 9.5 million entries.
  - Contains "words and phrases and common-sense relationships between them."

- The basic unit of ConceptNet is a **string**, i.e. a word or phrase.
  - Doesn't distinguish between different word senses.

Example relations for the word *coffee* in ConceptNet

Synonyms	Hyponyms	Hypernyms	Antonyms
en coffee bean	en cappuccino	en beverage	en tea
es café	en espresso	en drink	
sv kaffe	en instant coffe	en stimulant	
zh 咖啡	en mocha	en tree	

Source: ConceptNet 5.8

Important concepts

- lemma, lexeme
- polysemy, homonymy
- semantic relations
  - synonymy, antonymy, hypernymy, hyponymy

Distributional Semantics

2

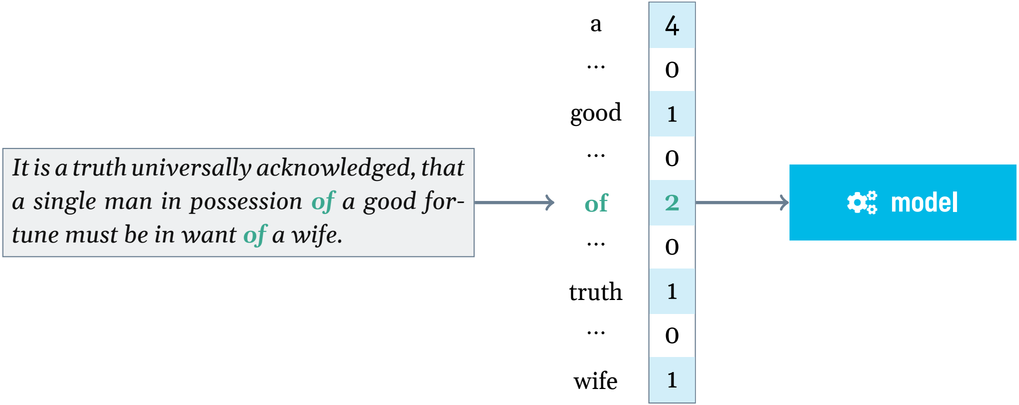
Reminder: Bag of words

- For most machine learning algorithms, we first need to **convert text into numerical vectors**.

It is a truth universally acknowledged,
that a single man in possession of a
good fortune must be in want of a wife.

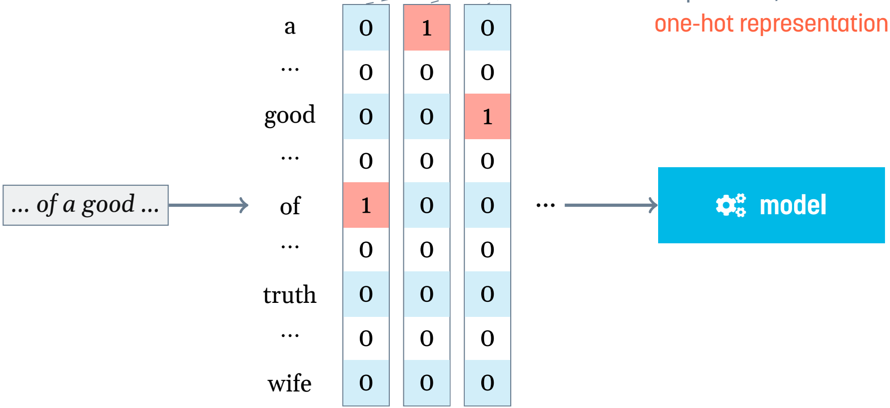
Earlier, we learned about the **bag-of-words** representation.

Bag-of-words as a numerical vector



Sequence representation with one-hot vectors

Each vector indicates which word it represents; also called one-hot representation



Reminder: How do we represent text?

- So far, we used either **bag-of-words** or simply **individual words**.
  - each vector dimension corresponds to a word in the vocabulary
- We also learned about **feature vectors**.
  - each vector dimension corresponds to a feature that we define by hand

Problem

None of these encode anything about the **meaning** of words.

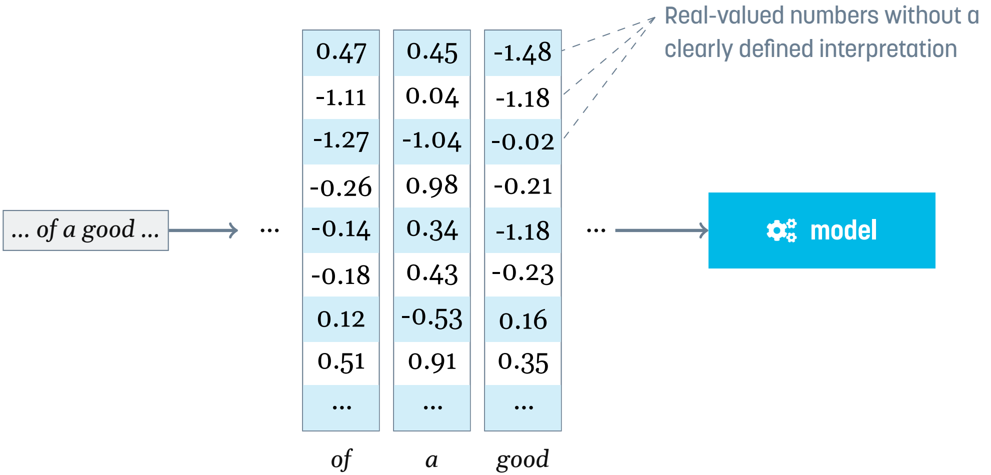
Dense vector representations

💡 **Idea #1**

Vectors should encode the **meaning** of words, so that words with similar meanings are **closer to each other** in the vector space.

- This type of vector is called a **word embedding**.
  - "embedded" into a vector space
  - **dense** vectors: all values are typically non-zero!
- **Dimensions** (axes) of the vector now have no clearly defined interpretation.

Sequence representation with word embeddings



Distributional semantics

Idea #2

We can look at the **context** of words to learn something about their meaning.

- Popularized by English linguist John R. Firth in the 1950s.
  - *"You shall know a word by the company it keeps."*
- Two words that frequently occur together are called **collocations**.

...trying to rebuild his **life** after the tragic **death** of his wife ...
...sometimes dark and all about **life**, love and **death**, the stories are ...
...to understand the **life**, work and **death** of Jesus of Nazareth ...

Collocations

Corpus of Contemporary American English

SEARCH | WORD | CONTEXT

COLLOCATES BOOK NOUN See also as: VERB Advanced options Coll

+ NOUN		NEW WORD	?
7596	3.86	author	
3671	3.39	review	
2868	3.41	library	
2385	2.58	club	
2136	2.80	title	
2040	3.45	copy	
2008	3.54	chapter	
1949	3.62	description	
1836	2.51	reader	
1757	3.09	cover	
1592	2.70	reading	

+ ADJ		NEW WORD	?
5289	6.77	comic	
1804	2.84	favorite	
1382	6.06	best-selling	
618	4.74	forthcoming	
587	6.25	self-help	
502	3.58	audio	
429	4.25	printed	
351	2.62	upcoming	
259	3.36	published	
255	6.60	self-published	
205	3.51	award-winning	

+ VERB		NEW WORD	?
29765	4.32	read	
23719	3.81	write	
6349	4.23	publish	
1660	2.95	recommend	
1306	2.86	review	
875	4.39	title	
559	2.80	entitle	
467	2.55	research	
442	4.13	kindle	
355	5.09	author	
301	5.63	co-author	

Source: COCA *(requires registration)*

What can we learn from collocations?

- What can we learn about ***Garrotxa*** from the following sentences?
  - *Garrotxa is made from milk.*
  - *Garrotxa pairs well with crusty country bread.*
  - *Garrotxa is aged in caves to enhance mold development.*
- The **distributional hypothesis** states that words with **similar distributions** have **similar meanings**.
  - "distributions" ≈ what contexts a word appears in

Paraphrased from Wikipedia

Important concepts

- word embeddings
- collocations
- distributional semantics
- distributional hypothesis

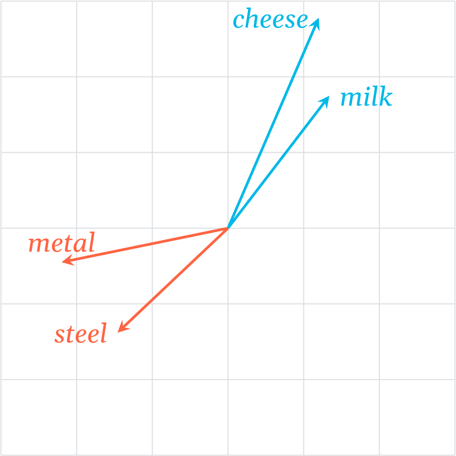
Vector Semantics

3

Vector semantics

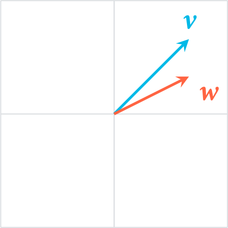
- **Pre-trained word embeddings** can be downloaded for many languages.
  - NLPL word embeddings repository
  - ConceptNet Numberbatch
- How can we **analyze the information** encoded in these vectors?
  - *Idea:* “words with similar meanings should be closer to each other in the vector space”
- What else can we do with these vectors?

Word embeddings, intuition



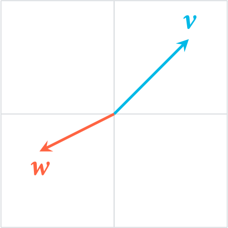
The dot product

$$\boldsymbol{v} = (+2, +2)$$
$$\boldsymbol{w} = (+2, +1)$$



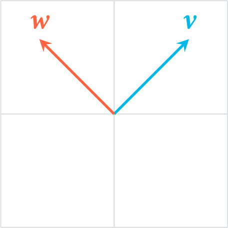
$$\boldsymbol{v} \cdot \boldsymbol{w} = +6$$

$$\boldsymbol{v} = (+2, +2)$$
$$\boldsymbol{w} = (-2, -1)$$



$$\boldsymbol{v} \cdot \boldsymbol{w} = -6$$

$$\boldsymbol{v} = (+2, +2)$$
$$\boldsymbol{w} = (-2, +2)$$



$$\boldsymbol{v} \cdot \boldsymbol{w} = \pm 0$$

Cosine similarity

- The dot product is sensitive to the **length** of the vectors.

- The **cosine similarity** of two vectors is the **length-normalized dot product**:

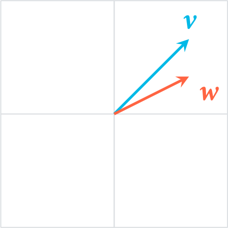
$$\cos(\boldsymbol{v}, \boldsymbol{w}) = \frac{\boldsymbol{v}}{|\boldsymbol{v}|} \cdot \frac{\boldsymbol{w}}{|\boldsymbol{w}|} = \frac{\boldsymbol{v} \cdot \boldsymbol{w}}{|\boldsymbol{v}| \cdot |\boldsymbol{w}|}$$

$$= \frac{\sum_{i=1}^{d} \boldsymbol{v}_{\boldsymbol{i}} \boldsymbol{w}_{\boldsymbol{i}}}{\sqrt{\sum_{i=1}^{d} \boldsymbol{v}_{\boldsymbol{i}}^2} \cdot \sqrt{\sum_{i=1}^{d} \boldsymbol{w}_{\boldsymbol{i}}^2}}$$

- Cosine similarity ranges from **−1** *(opposite)* to **+1** *(identical)*.

Cosine similarity

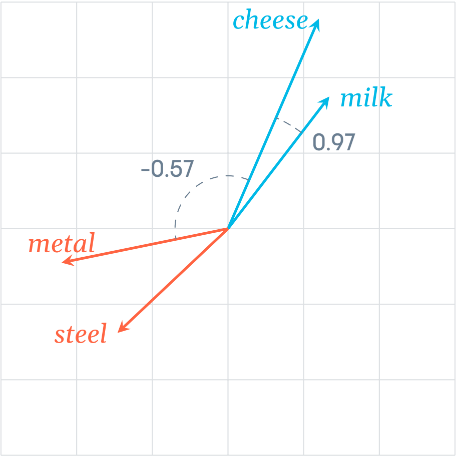
$$v = \quad (+2, +2)$$
$$w = \quad (+2, +1)$$



$$v \cdot w = +6$$

$$\begin{aligned}
\cos(v, w) &= \frac{v \cdot w}{|v||w|} \\
&= \frac{6}{\sqrt{2^2 + 2^2} \cdot \sqrt{2^2 + 1^2}} \\
&= \frac{6}{\sqrt{8} \cdot \sqrt{5}} \\
&\approx \frac{6}{6.3246} \\
&\approx 0.9487
\end{aligned}$$

Cosine similarity on word embeddings



Word analogies

- **Word analogies** are one way to "probe" the information encoded in word vectors.

$$man : woman :: king : queen$$

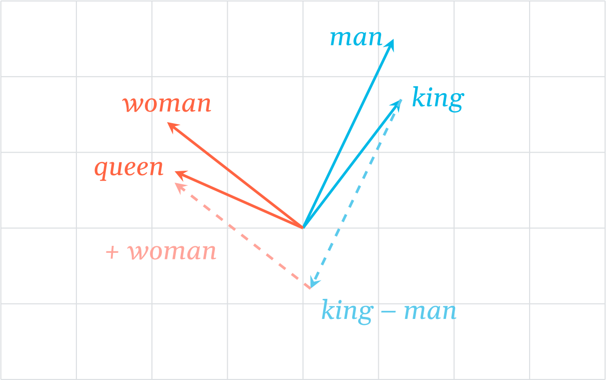
man is to *woman* as *king* is to *queen*

- **Idea**: Use vector semantics to find the last word of the analogy.

$$\boldsymbol{v}_{king} - \boldsymbol{v}_{man} + \boldsymbol{v}_{woman} \approx \boldsymbol{v}_{queen}$$

This was originally proposed by Mikolov et al. (2013)

Word analogies, intuitively



Important concepts

- cosine similarity
- word analogies *(with embedding vectors)*

Learning Word Embeddings

4

Intuition: Learning word embeddings

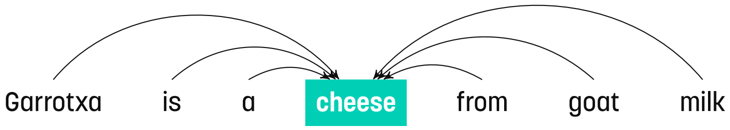
- Word embeddings are typically produced by **training neural networks**.
- Similar to the perceptron, neural networks have **weight matrices** that they "learn."

$$\hat{y} = f(\underbrace{\boldsymbol{x}}_{\text{input vector} \in \mathbb{R}^n} \underbrace{\boldsymbol{W}}_{\text{weight matrix} \in \mathbb{R}^{n \times k}})$$

- If $\boldsymbol{x}$ is an **indicator vector** for a word w, then $\boldsymbol{x} \cdot \boldsymbol{W}$ is the **word embedding** for w.

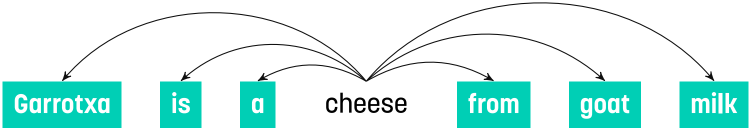
Continuous bag-of-words model

- Train a classifier to **predict a word from its context**:



Continuous skip-gram model

- Train a classifier to **predict context from a given word**:



- Both methods were originally implemented as Google's **word2vec**.

Skip-gram model as binary classification

- What's the probability that ***milk*** is a **real context word** of ***cheese***?

$$P(+ \mid milk, cheese)$$

- If *milk* and *cheese* are **semantically similar**, we want this probability to be high.

- What's the probability that ***robot*** is **not a real context word** of ***cheese***?

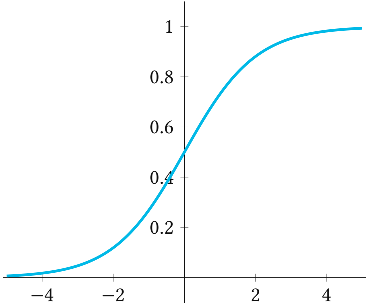
$$P(- \mid robot, cheese) = 1 - P(+ \mid robot, cheese)$$

- If *robot* and *cheese* are **semantically different**, we want this probability to be low.

From dot product to probability

- The dot product takes values in the range $[-\infty, +\infty]$.
- We can use the **logistic function** to map this to the range $[0, 1]$.

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$



Negative sampling

- We can get **positive examples** from training data.
- We can get **negative examples** using **negative sampling**.
  - randomly sample words from the entire vocabulary

$$
\begin{array}{rrr}
P(+ \mid is, cheese) & P(- \mid wicked, cheese) & P(- \mid mattress, cheese) \\
P(+ \mid from, cheese) & P(- \mid doubts, cheese) & P(- \mid headers, cheese) \\
P(+ \mid goat, cheese) & P(- \mid hell, cheese) & P(- \mid therapy, cheese) \\
P(+ \mid milk, cheese) & P(- \mid metal, cheese) & P(- \mid packages, cheese)
\end{array}
$$

Learning embeddings with the skip-gram model

1. Initialize all word vectors with **random values**.
2. **Compute probabilities** for both positive and negative examples.
3. Apply a **learning algorithm** to update the word vectors.
   - probability should be high for positive examples, low for negative examples
   - common algorithm: stochastic gradient descent (SGD) → advanced material!
4. Repeat steps 2 & 3 several times.

Example: Learning embeddings with the skip-gram model

Step 1: Initialize vectors with random values.

-1.71	0.36
-0.50	-0.04
-0.80	1.59
0.68	0.12
-1.31	-0.63
-0.17	-0.26
0.99	0.03
-0.37	-0.40
...	...
milk	*cheese*

Example: Learning embeddings with the skip-gram model

$$P(+ \mid milk, cheese) = \sigma\left(\begin{bmatrix} -1.71 \\ -0.50 \\ -0.80 \\ 0.68 \\ -1.31 \\ -0.17 \\ 0.99 \\ -0.37 \\ \cdots \end{bmatrix}_{milk} \cdot \begin{bmatrix} 0.36 \\ -0.04 \\ 1.59 \\ 0.12 \\ -0.63 \\ -0.26 \\ 0.03 \\ -0.40 \\ \cdots \end{bmatrix}_{cheese} \right)$$

Step 2: Compute probability of a positive example.

Example: Learning embeddings with the skip-gram model

$$P(+\mid milk, cheese) = \sigma\left(\begin{bmatrix} -1.71 \\ -0.50 \\ -0.80 \\ 0.68 \\ -1.31 \\ -0.17 \\ 0.99 \\ -0.37 \\ \dots \end{bmatrix}_{milk} \cdot \begin{bmatrix} 0.36 \\ -0.04 \\ 1.59 \\ 0.12 \\ -0.63 \\ -0.26 \\ 0.03 \\ -0.40 \\ \dots \end{bmatrix}_{cheese} \right) \approx \sigma(-0.73) \approx 0.33$$

Step 3: Update the vectors so that their dot product **increases**.

Important concepts

- skip-gram model
- logistic function
- negative sampling

Outlook

5

Using word embeddings for classifiers

- In sequence labelling, word embeddings can **replace feature vectors**.
  - simply represent each word by its embedding
- Word embeddings can also **replace bag-of-words** in classification.
  - *e.g.* average the embeddings of all words in a sentence
- Mapping words to embeddings is the first step in any **neural network** model.
  - includes all state-of-the-art NLP models, like ChatGPT

Static vs. dynamic embeddings

- **Static** embeddings: a word will always get the **same vector** regardless of context.
  - *e.g.* “bass” the instrument vs. “bass” the fish
- **Dynamic** (also: **contextualized**) embeddings solve this problem.
  - require more advanced neural networks

LINKÖPING
UNIVERSITY