UNIVERSIDADE CATÓLICA PORTUGUESA

BRAGA

Behavior Analysis Technologies

Session 15

Word Embeddings

Applied Data Science

2024/2025

Word Meaning via Language Use

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- The meaning of a word can be given by its distribution in language usage:
    - One way to define "usage": words are defined by their environments
        - Neighboring words or grammatical environments

- Intuitions: Zellig Harris (1954):
    - “oculist and eye-doctor … occur in almost the same environments”
    - “If A and B have almost identical environments we say that they are synonyms.”

Words Representations

- How can we represent the meaning of words?

- So we can ask:
  - How similar is cat to dog, or Paris to London?
  - How similar is document A to document B?

Words as Vectors

- Can we represent words as vectors?

- The vector representations should:
  - Capture semantics
    - similar words should be close to each other in the vector space
    - relation between two vectors should reflect the relationship between the two words
  - Be efficient (vectors with fewer dimensions are easier to work with)
  - Be interpretable

Words as Vectors

- How similar are the following two words? (not similar 0–10 very similar)
    - Smart and intelligent
    - Easy and big
    - Easy and difficult
    - Hard and difficult

Words as Vectors

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- How similar are the following two words? (not similar 0–10 very similar)
    - Smart and intelligent: 9.20
    - Easy and big: 1.12
    - Easy and difficult: 0.58
    - Hard and difficult: 8.77
- (SimLex-999 dataset, https://fh295.github.io/simlex.html)

Recap: One-Hot-Encoding

- Map each word to a unique identifier (e.g. cat (3) and dog (5))

- Vector representation: all zeros, except 1 at the 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

Recap: One-Hot-Encoding

- Map each word to a unique identifier (e.g. cat (3) and dog (5))

- Vector representation: all zeros, except 1 at the 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

What are the limitations of one-hot-encoding?

Recap: One-Hot-Encoding

- Map each word to a unique identifier (e.g. cat (3) and dog (5))

- Vector representation: all zeros, except 1 at the 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

High number of dimensions!

Related words have distinct vectors.

Distributional Hypotesis

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Words that occur in similar contexts tend to have similar meanings.

J.R.Firth 1957

- "You shall know a word by the company it keeps"
- One of the most successful ideas of modern statistical NLP!

Distributional Hypotesis

“tejuino”

C1: A bottle of ____ is on the table.

C2: Everybody likes ____.

C3: Don’t have ____ before you drive.

C4: We make ____ out of corn.

Distributional Hypotesis

C1: A bottle of ___ is on the table.

C2: Everybody likes ___.

C3: Don't have ___ before you drive.

C4: We make ___ out of corn.

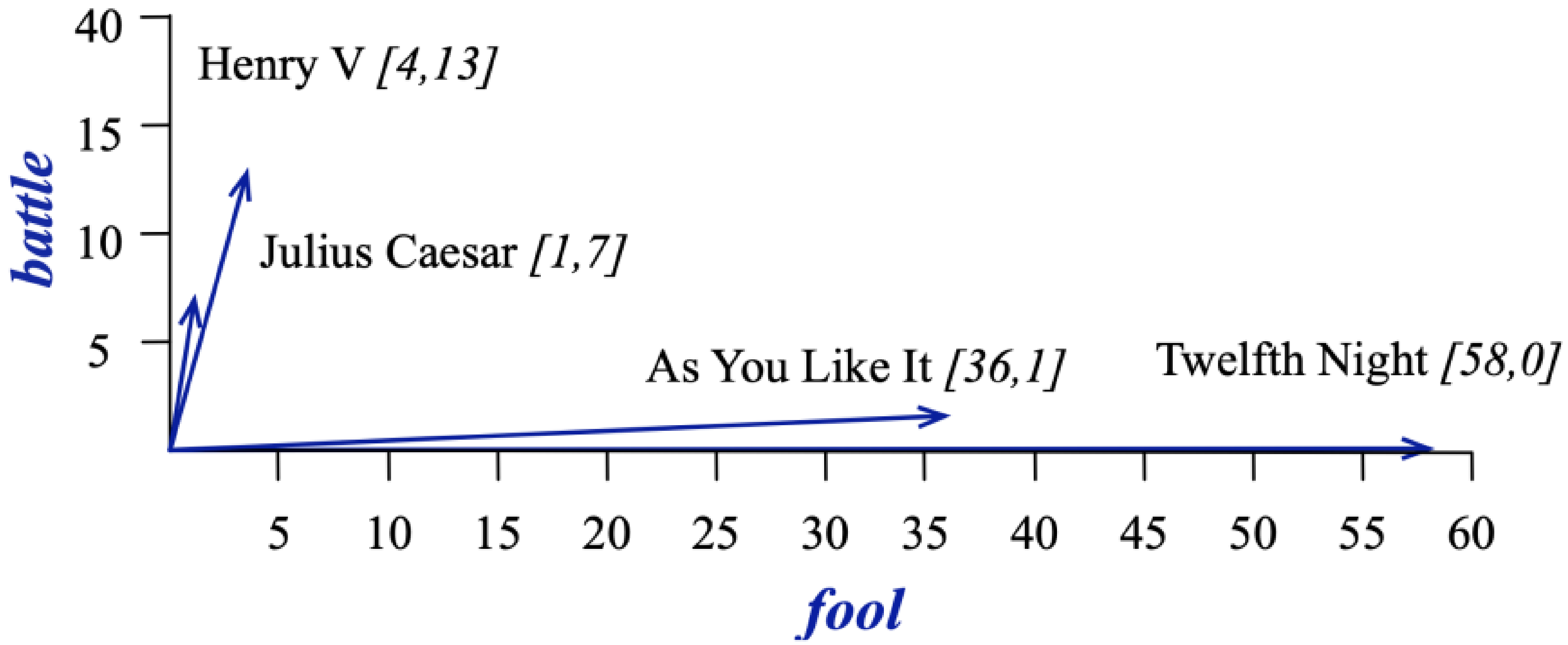
	C1	C2	C3	C4
tejuino	1	1	1	1
loud	0	0	0	0
motor-oil	1	0	0	0
tortillas	0	1	0	1
choices	0	1	0	0
wine	1	1	1	0

"words that occur in similar contexts tend to have similar meanings"

Term Document Matrix and Document Vectors

Each **document** is represented by a vector of words

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3



- Vectors are similar for the two comedies
- Comedies are different from the other two (tragedies)
  - More fools, less battle

Word-Word Co-Occurrence Matrix

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

Context Window

is traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

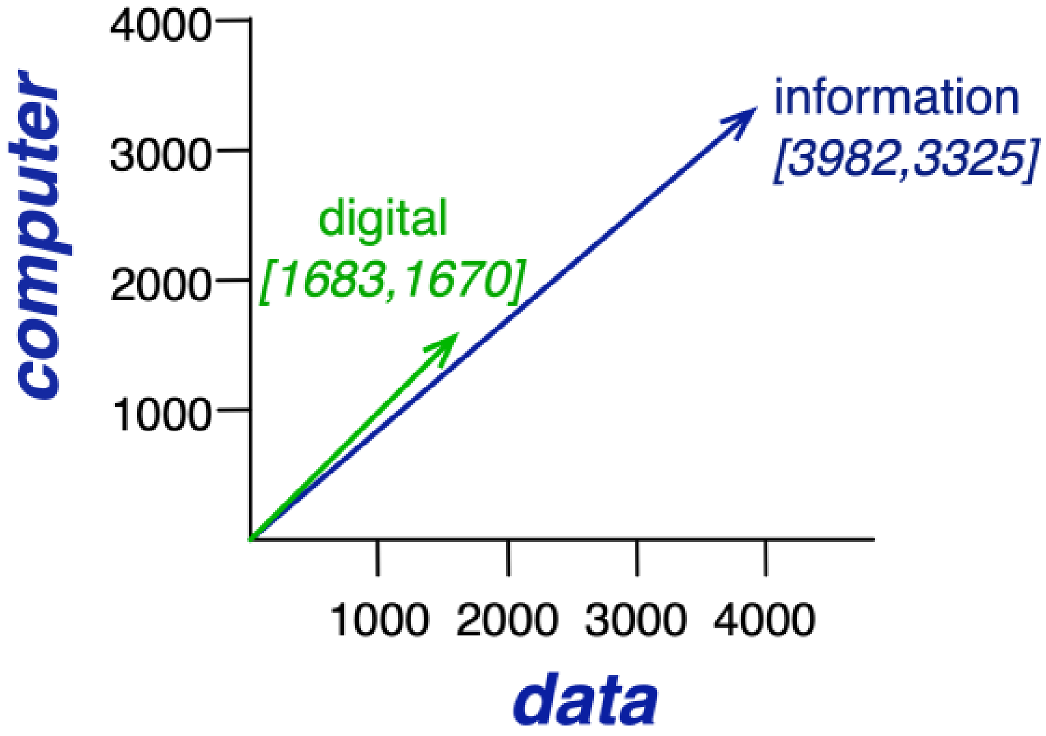
Two words are similar in meaning if their context vectors are similar

Words, not documents

	aardvark	...	**computer**	**data**	**result**	**pie**	**sugar**	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Word-Word Co-Occurrence Matrix

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...



Choice of features matters!

Not every word's raw frequency matters!

Word Vectors Based on Co-Occurrences

- There are many variants:
    - Context (words, documents, which window size, etc.)
    - Weighting (raw frequency, etc.)

- **Vectors are sparse:** Many zero entries.
    - Therefore: Dimensionality reduction is often used.

- These methods are sometimes called count-based methods as they work directly on co-occurrence counts.

Sparse vs Dense Vectors

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- As we have seen, count-based methods are sparse (most are 0's) and long (high dimensionality).

- Alternatively, we want to represent words as **short** (50-1024 dimensional) and **dense** (real-valued) vectors

- On the other hand, individual dimensions are **less interpretable**!

cat	0.52	0.48	-0.01	...	0.28
dog	0.32	0.42	-0.09	...	0.78

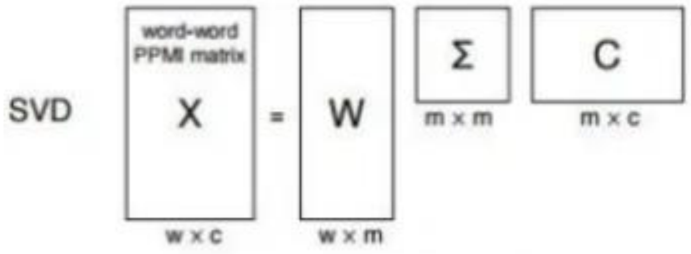
Dense Vectors

$$\text{employees} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 10.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix}$$

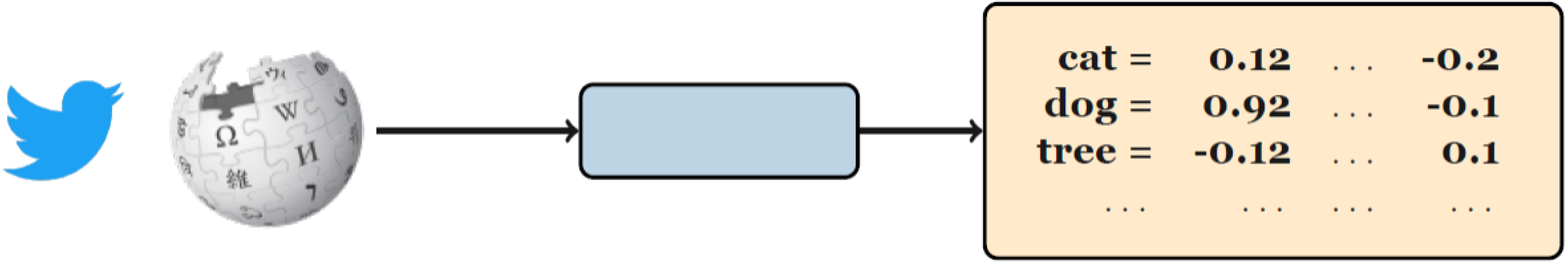
Why Dense Vectors?

- Short vectors are **easier to use as features** in ML systems
- Dense vectors may **generalize better** than storing explicit counts
- They do better at capturing **synonyms**

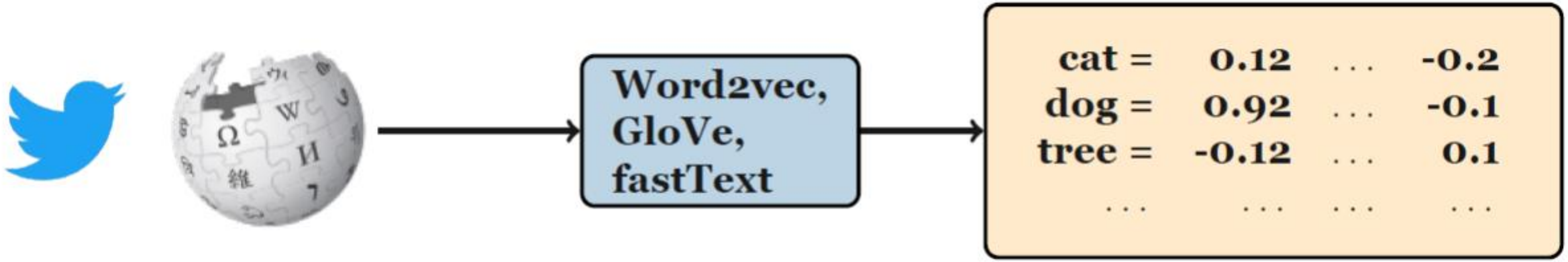
- Different methods for getting dense vectors:
  - Singular value decomposition (SVD)
  - word2vec and friends: **“learn” the vectors**!



Learning Word Embeddings



Learning Word Embeddings



Training Data for Word Embeddings

- Use text itself as training data for the model!
  - A form of self-supervision.

- Train a classifier (neural network, logistic regression, or SVM, etc.) to predict the next word given previous words.

Exercise: Word Prediction Task

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

Which word comes next?

Word2Vec

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Popular embedding method
- Very fast to train
- Idea: predict rather than count
- https://projector.tensorflow.org/

Word2Vec

The domestic **cat** is a small, typically furry carnivorous mammal

w_{-2} w_{-1} w_0 w_1 w_2 w_3 w_4 w_5

- We have **target** words (cat) and **context** words (here: window size = 5).

Word2Vec

- Instead of **counting** how often each word w occurs near a target word
  - Train a classifier on a binary **prediction** task:
    - Is w likely to show up near target?

- We don't actually care about this task
  - But we'll take the learned classifier weights as the word embeddings

- Big idea: **self-supervision**
  - A word c that occurs near target in the corpus as the gold "correct answer" for supervised learning
  - **No need for human labels**
  - Bengio et al. (2003); Collobert et al. (2011)

Word2Vec

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Input: a large text corpora, V, d
  - V: a pre-defined vocabulary
  - d: dimension of word vectors (e.g. 300)
  - Text corpora:
    - Wikipedia + Gigaword 5: 6B
    - Twitter: 27B
    - Common Crawl: 840B

- Output: $f : V \to \mathbb{R}^d$

$$v_{\text{cat}} = \begin{pmatrix} -0.224 \\ 0.130 \\ -0.290 \\ 0.276 \end{pmatrix} \quad v_{\text{dog}} = \begin{pmatrix} -0.124 \\ 0.430 \\ -0.200 \\ 0.329 \end{pmatrix}$$

$$v_{\text{the}} = \begin{pmatrix} 0.234 \\ 0.266 \\ 0.239 \\ -0.199 \end{pmatrix} \quad v_{\text{language}} = \begin{pmatrix} 0.290 \\ -0.441 \\ 0.762 \\ 0.982 \end{pmatrix}$$

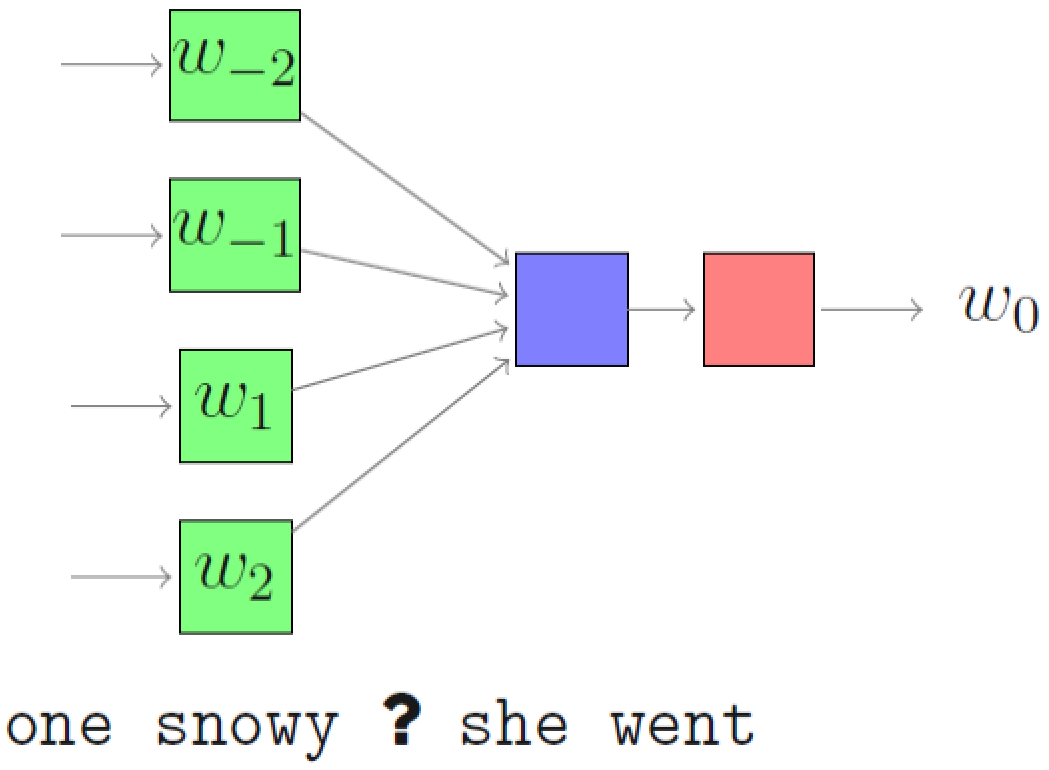
Word2Vec

- **Two different tasks (context):**
  - Continuous Bag-of-Words
  - Skipgram

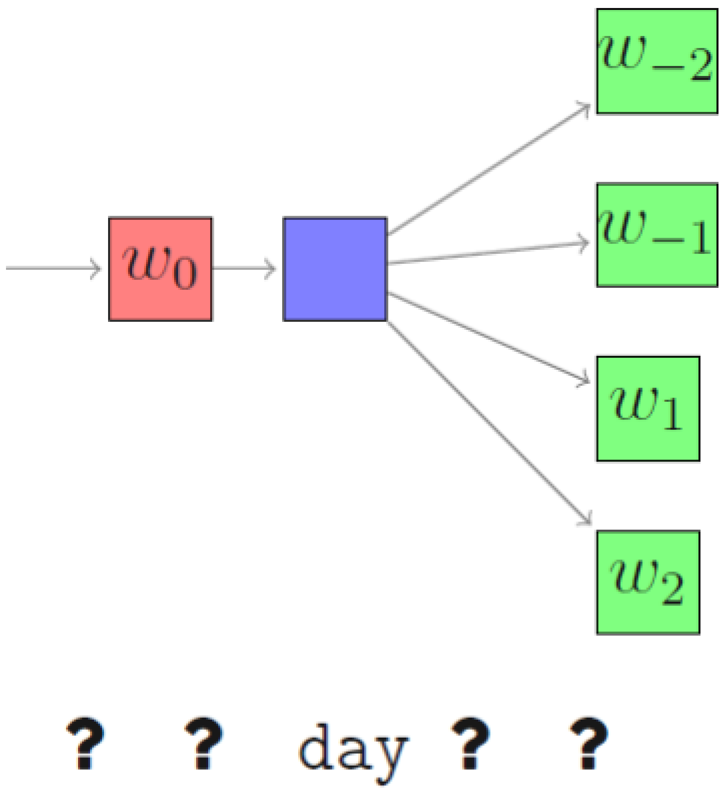
- **Two training regimes (reducing compute cost):**
  - Hierarchical Softmax
  - Negative Sampling

Word2Vec Algorithms

Continuous Bag-Of-Words (CBOW)



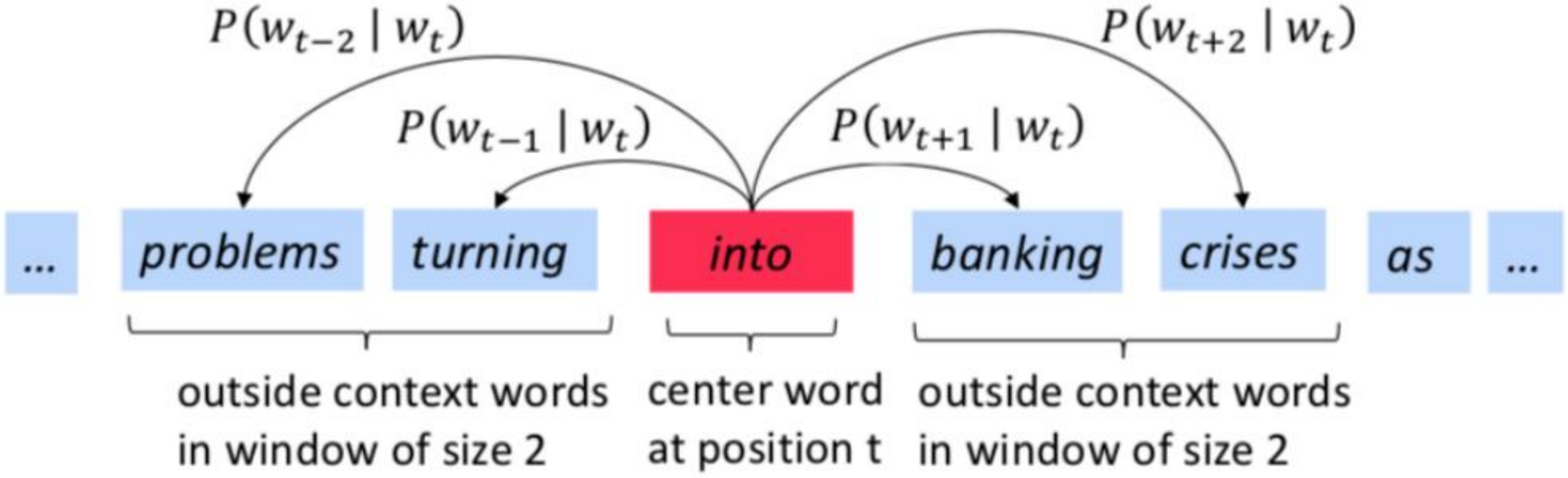
skipgram



The Skipgram Model

- **The idea:** we want to use words to predict their context words

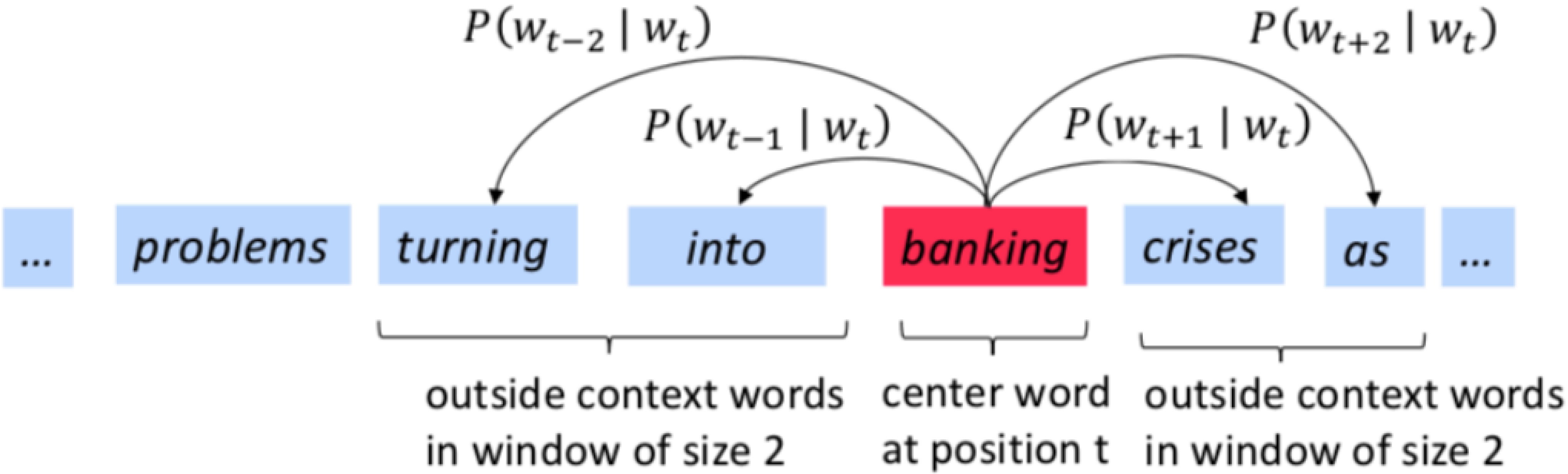
- **Context:** a fixed window of size 2m



The Skipgram Model

- **The idea:** we want to use words to predict their context words

- **Context:** a fixed window of size 2m



Skipgram: Overview

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

1. **Create examples:**
   - Positive examples: target word and neighboring context
   - Negative examples: target word and randomly sampled words from the lexicon (**negative sampling**)

The domestic **cat** is a small, typically furry carnivorous mammal

word (w)	context (c)	label
cat	small	1
cat	furry	1
cat	car	0
...	...	...

2. **Train a logistic regression model** to distinguish between the positive and negative examples
3. **The resulting weights are the embeddings!**

Skipgram: Overview

The domestic **cat** is a small, typically furry carnivorous mammal

$c1 \quad c2 \quad w \quad c3 \; c4 \; c5 \quad c6 \quad c7$

- We have **target** words (cat) and **context** words (here: window=5).

- The probability that c is a real context word:

$$P(+|w, c)$$

- The probability that c is not a real context word:

$$P(-|w, c)$$

Skipgram: Overview

UNIVERSIDADE CATÓLICA PORTUGUESA BRAGA

- Intuition: A word c is likely to occur near the target if its embedding is similar to the target embedding.

$$\approx w \cdot c$$

- Turn this into a probability using the sigmoid function

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

Skipgram: Training

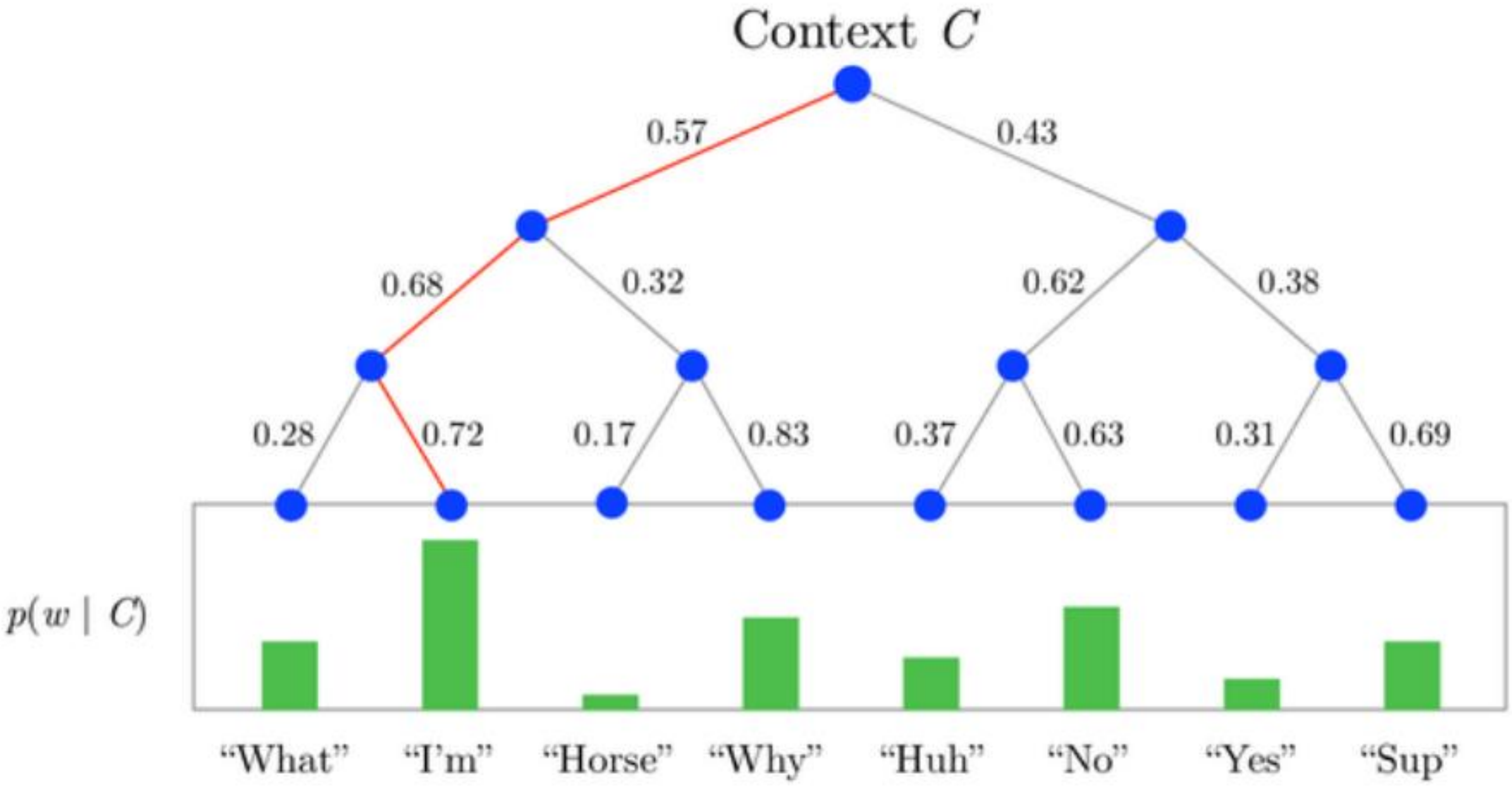
- We **initialize** the embeddings with random values

- **During training:**
  - Maximize the similarity between the embeddings of the target word and context words from the positive examples
  - Minimize the similarity between the embeddings of the target word and context words from the negative examples

- **After training:**
  - frequent word-context pairs in data: $w \cdot c$ high
  - not word-context pairs in data: $w \cdot c$ low

- So: Words occurring in same contexts are close to each other

Hierarchical Softmax



$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = \mathrm{ch}(n(w,j))]\!] \cdot {v'_{n(w,j)}}^\top v_{w_I} \right)$$

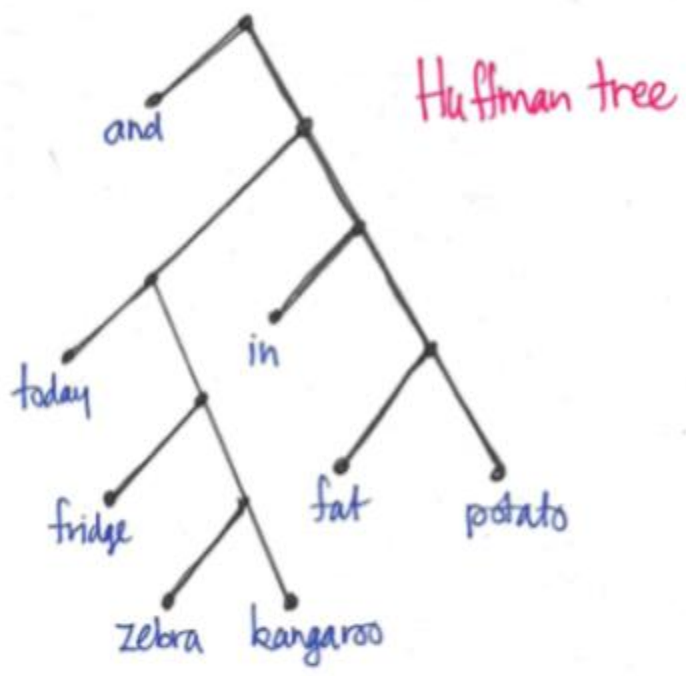
Hierarchical Softmax

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

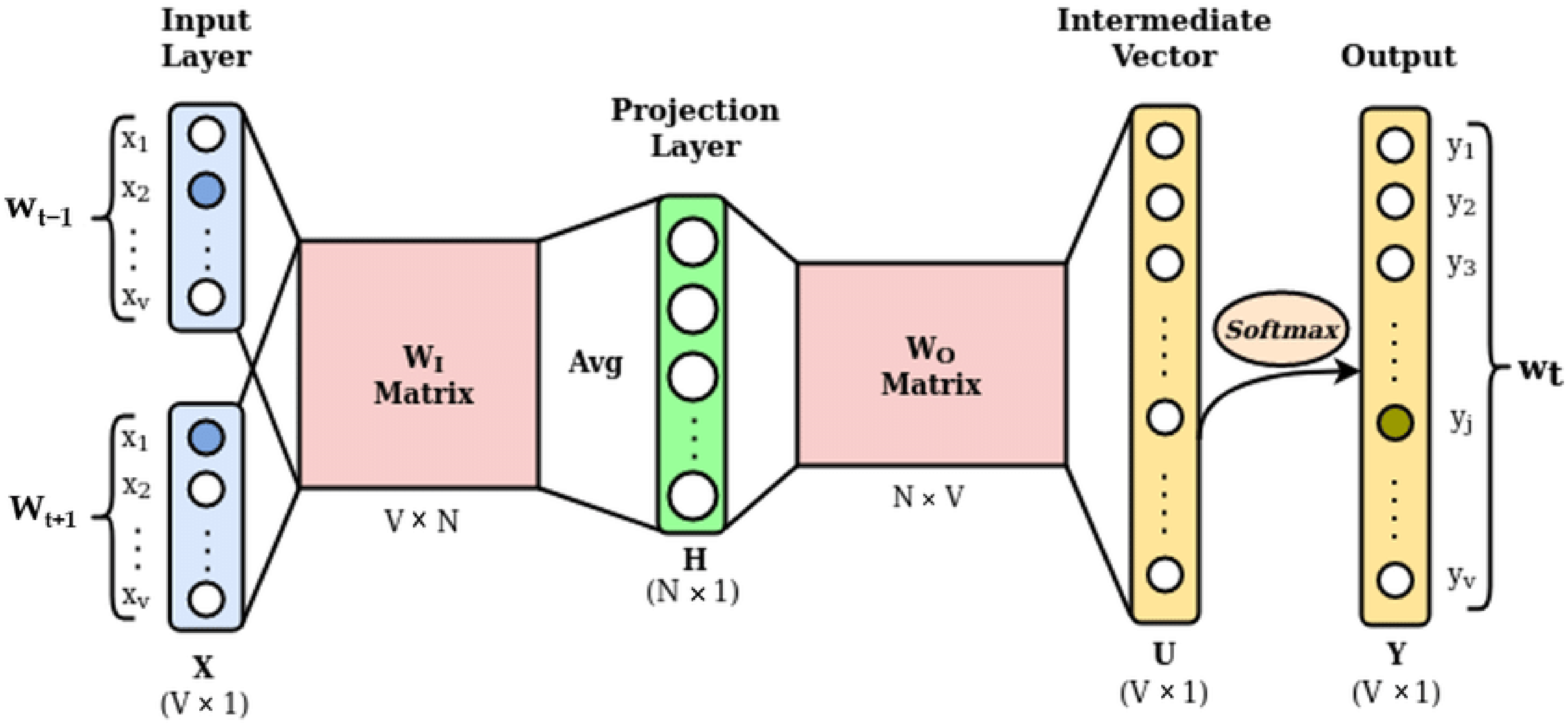
- Huffman tree:

word	count
fat	3
fridge	2
zebra	1
potato	3
and	14
in	7
today	4
kangaroo	2

→



Continuous Bag-of-Words



Other Embeddings: fastText

- **Limitation of word2vec**: Can't handle unknown words

- fastText is very similar to word2vec, but each word is represented as a **bag of character n-grams** (+ the word itself). ≤ and ≥ mark word boundaries.

- Example: where with n = 3: <wh, whe, her, ere, re> and <where>

- Representation of a word: The sum of the vector representations of its n-grams.

Other Embeddings: GloVe

- First create a global **word-word co-occurrence matrix** (how frequent pairs of words occur with each other). Requires a pass through the entire corpus at the start!

- **Training objective:** learn word embeddings so that their dot products equals the log of the words' co-occurrence probability.

Pre-Trained Embeddings

- I want to build a system to solve a task (e.g. sentiment analysis)
  - Use pre-trained embedddings. Should I fine-tune?
    - Lots of data: yes
    - Just a small dataset: no

- Analysis (e.g. bias, semantic change):
  - Train embeddings from scratch

Properties of Word Embeddings

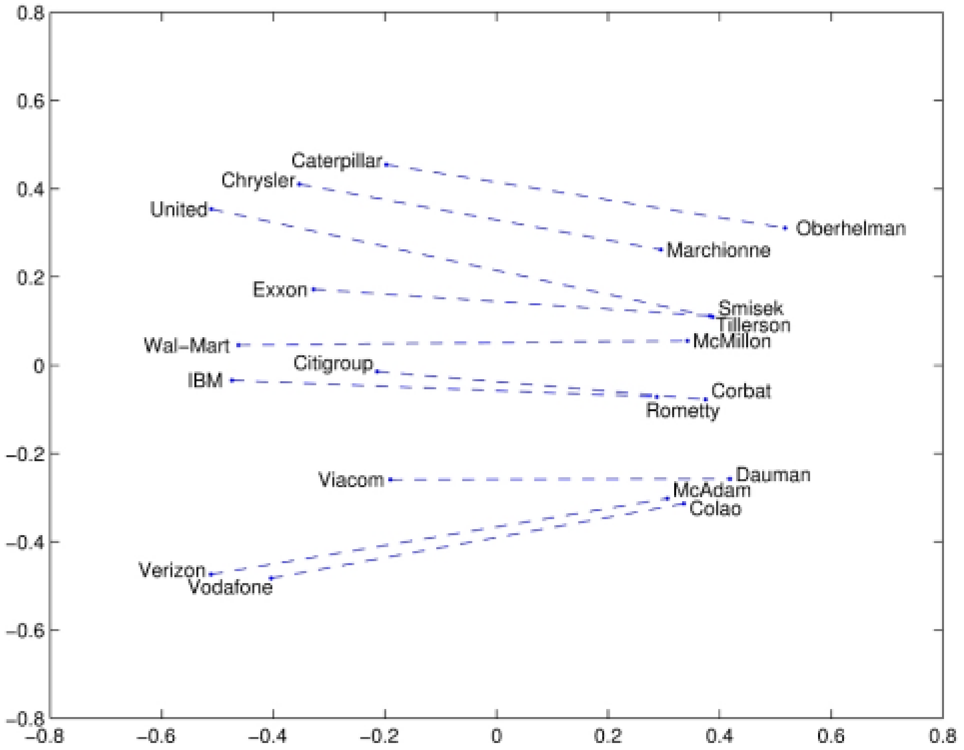


Figure: company - ceo

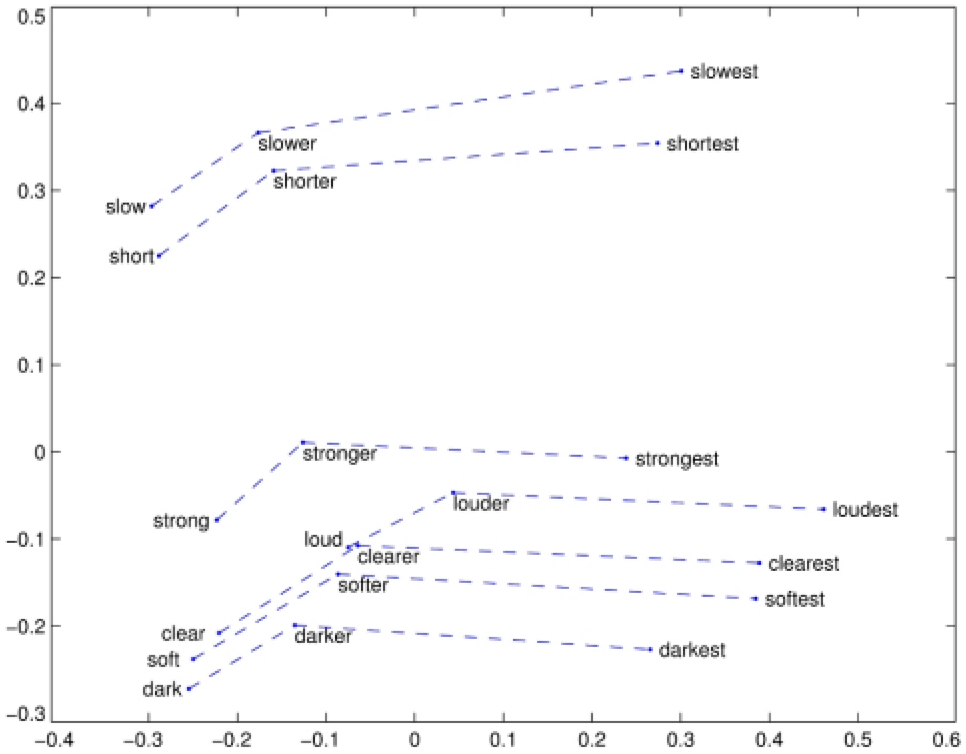
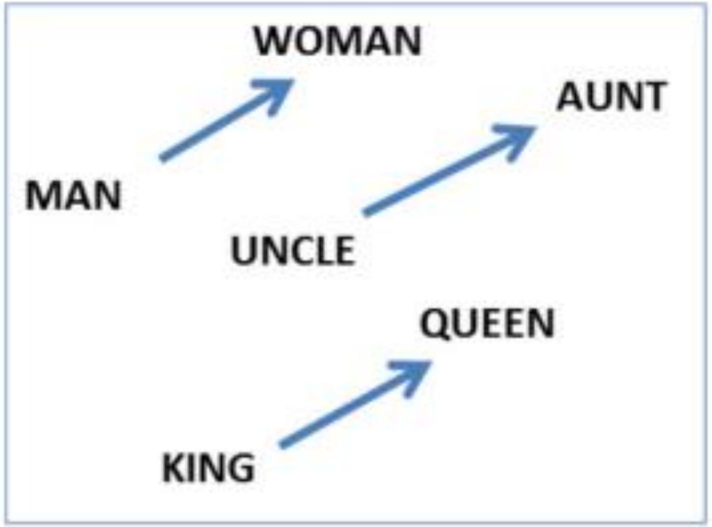
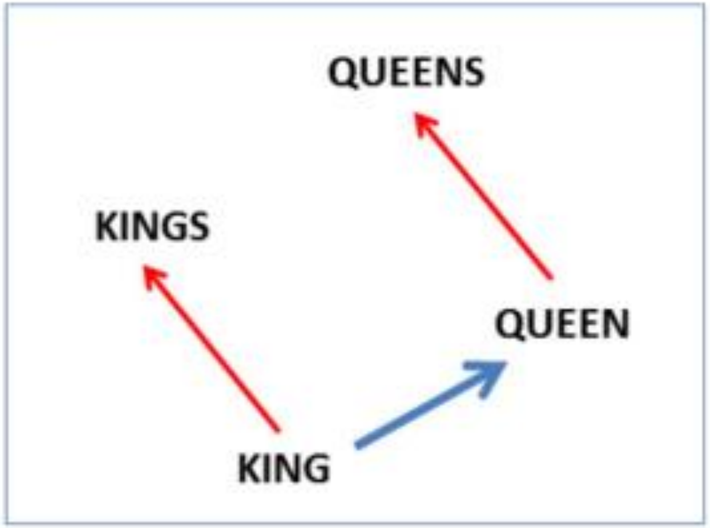


Figure: comparative - superlative

Properties of Word Embeddings: Analogies

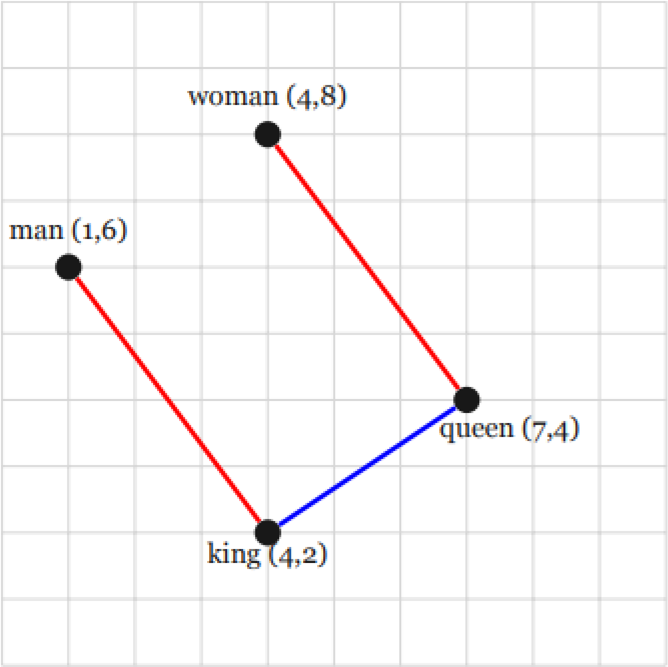
- We can look at analogies in the vector space, for example:
  - king - man + woman ≈ queen





Properties of Word Embeddings: Analogies

- We can look at analogies in the vector space, for example:
  - king - man + woman ≈ queen




```
king-man = [4,2] - [1,6] = [3,-4]
king-man + woman = [3,-4] + [4,8] = [7,4]
```

Dong Nguyen (2024)

Embeddings Evaluation

- How would you evaluate word embeddings?

- E.g., how do you know whether a new word embedding algorithm is an improvement over previous ones?

Embeddings Evaluation

- Types of evaluation:
  - Extrinsic evaluation
  - Intrinsic evaluation

Embeddings Evaluation

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Types of evaluation:
  - Extrinsic evaluation
  - Intrinsic evaluation

- **Extrinsic Evaluation:**
  - Evaluation based on performance on external tasks (e.g., part of speech tagging, sentiment analysis)
  - I.e., plug in different embeddings into the same NLP system and measure difference in task performance.

Embeddings Evaluation

- Types of evaluation:
  - Extrinsic evaluation
  - Intrinsic evaluation

- **Intrinsic Evaluation:**
  - Evaluation based only on the embeddings
    - Similarities
    - Analogies
    - Probing classifiers

Intrinsic Evaluation: Similarities

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Input: Dataset with relatedness or similarity scores for pairs of words.

- Goal: High (pearson or spearman) correlation between scores and the cosine similarity of the embeddings for the two words.

- Example from WordSim353:
  - wood and forest: 7.73
  - money and cash: 9.15
  - month and hotel: 1.81

Intrinsic Evaluation: Analogies

- Base/3rd Person Singular Present
  - see:sees return: ?
- Singular/Plural
  - year:years law: ?
- Meronyms
  - player:team fish: ?
- UK city county
  - york:yorkshire Exeter: ?

Intrinsic Evaluation: Analogies

This method is referred to by Levy and Goldberg (2014) as **3COSADD**
$\mathbf{a} - \mathbf{a}^* \approx \mathbf{b} - \mathbf{b}^*$. We can find $\mathbf{b}^*$ as follows:

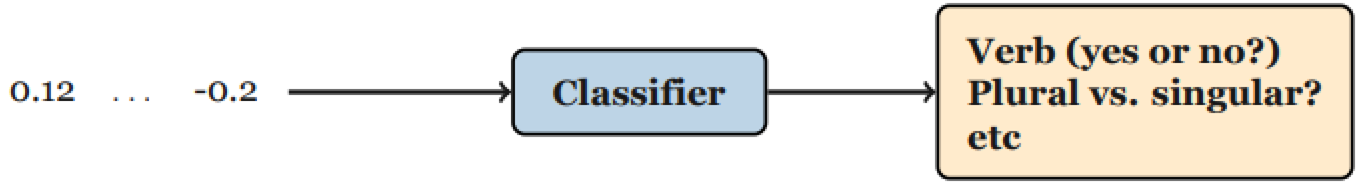
$$\underset{\mathbf{b}^* \in V}{\operatorname{argmax}}\, cos(\mathbf{b}^*, \mathbf{b} - \mathbf{a} + \mathbf{a}^*)$$

Linzen 2016 notes that results can be misleading:
The offsets are often very small, so that often just the nearest neighbor to **b** is returned.
Control setting: Just return the nearest neighbor of **b**.

Intrinsic Evaluation: Probing Classifiers

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Also called diagnostic classifiers

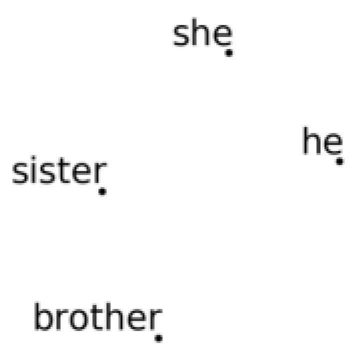


- Mostly used to evaluate sentence embeddings, but sometimes also used for analyzing word embeddings.

- But, be careful! Performance might seem high, but classifier might learn other signals (e.g. word frequency, part of speech classes) than what you focus on.

Biases in Word Embeddings

- Measuring gender bias:
  - To assess NLP models and investigate the impact of ‘bias mitigation’ techniques
  - To study societal trends

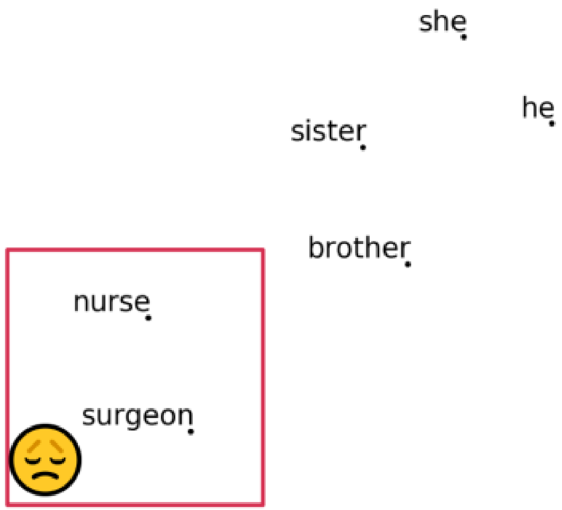


Biases in Word Embeddings

- Measuring gender bias:
  - To assess NLP models and investigate the impact of ‘bias mitigation’ techniques
  - To study societal trends

Gender appropriate she-he analogies
queen–king
sister–brother
ovarian cancer–prostate cancer
mother–father
convent–monastery

Gender stereotype she-he analogies
nurse–surgeon
sassy–snappy
cupcakes–pizzas
lovely–brilliant
vocalist–guitarist



Pre-trained GloVe model on Twitter

man is to *computer programmer* as *woman* is to ? : x = homemaker
father is to *doctor* as *mother* is to ? : x = nurse

Perpetuation of Bias in Sentiment Analysis

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

"I had tried building an algorithm for sentiment analysis based on word embeddings [..]. When I applied it to restaurant reviews, I found it was ranking Mexican restaurants lower. The reason was not reflected in the star ratings or actual text of the reviews. It's not that people don't like Mexican food. ***The reason was that the system had learned the word "Mexican" from reading the Web.****"*

Contextual Word Embeddings

The hut is located near the bank of the river

Tokens	Types
The	the
hut	hut
is	is
located	located
near	near
the	bank
bank	of
of	river
the	
river	

- So far: an embedding for **each word (type)**.

Contextual Word Embeddings

- So far: an embedding for **each word (type)**.

- Today, I went to the **bank** to deposit a check.

bank	0.52	0.48	-0.01	$\cdots$	0.28

- The hut is located near the **bank** of the river.

bank	-0.27	0.28	-0.07	$\cdots$	0.82

Key idea in NLP:
Can we have an embedding for each **word token?**

Contextualized Word Embeddings

- **Key idea:** Have embeddings for each **word token**

- Previously:
  - One embedding for each **word type**
  - A table where each word is mapped to a vector.

- Now:
  - One embedding for each **work token**
  - Embeddings for a token are created based on the context
  - There is no single embedding for a word anymore

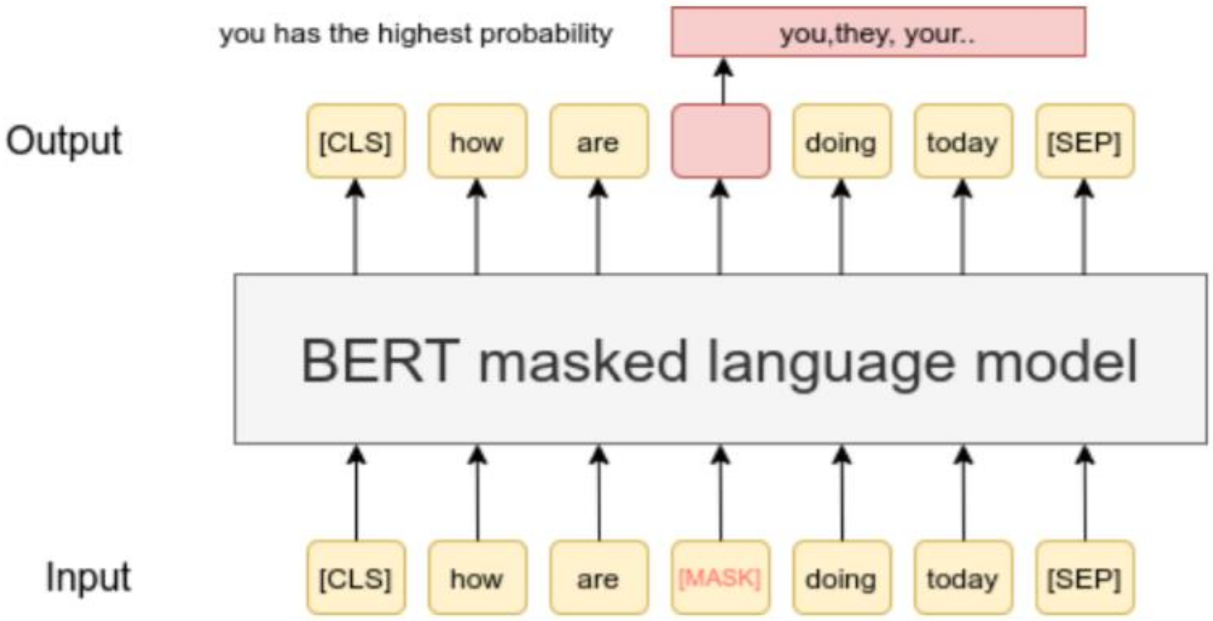
BERT

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Two tasks:
    - **Masked Language Model**
    - **Next Sentence Prediction**

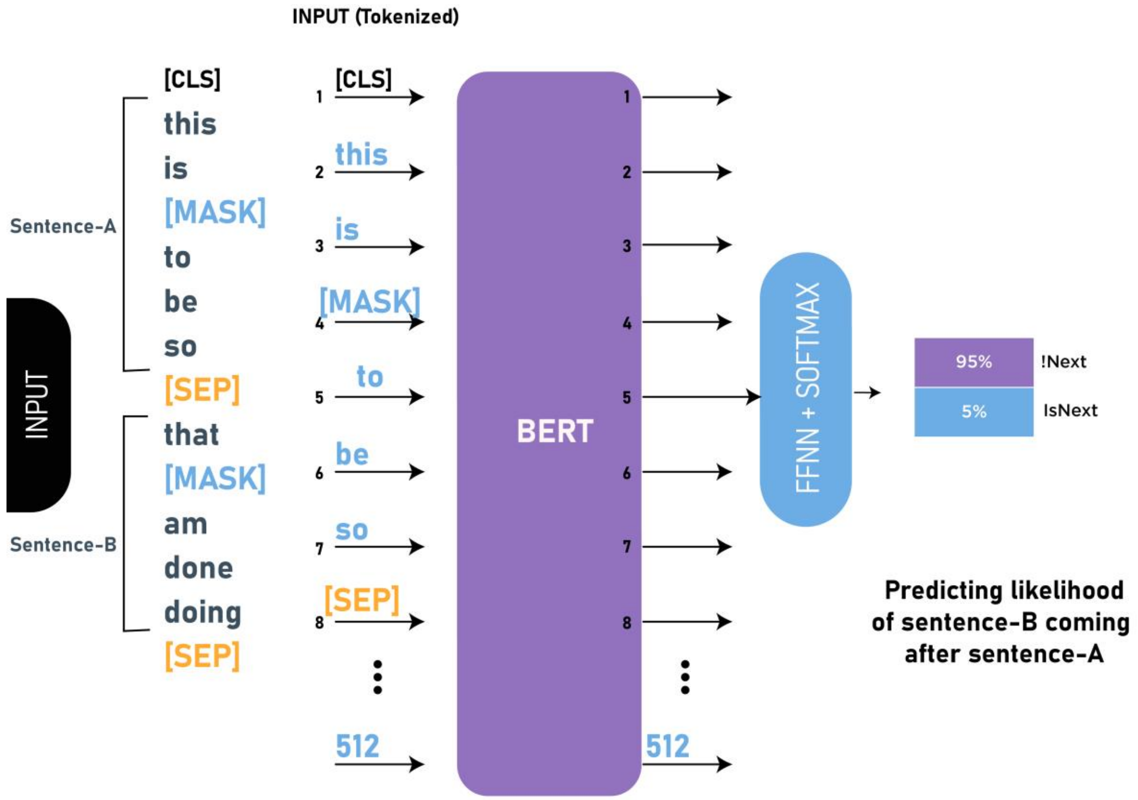
BERT

- Two tasks:
    - **Masked Language Model**



BERT

- Two tasks:
  - **Next Sentence Prediction**



Software

- word2vec: gensim (https://radimrehurek.com/gensim/) and official implementation (https://code.google.com/archive/p/word2vec/).
- fasttext: official implementation (https://fasttext.cc/)
- GloVe: official implementation (https://nlp.stanford.edu/projects/glove/)
- Hugging Face: for BERT and other transformer models (https://huggingface.co/)