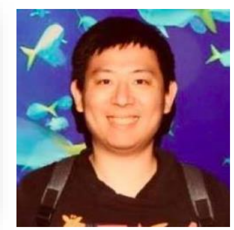
Deep Learning for NLP

Lecture 4 – Word Embeddings 1: CBOW & Skip-Gram

Dr. Steffen Eger
Wei Zhao
Niraj Dev Pandey


Natural Language Learning Group (NLLG)
Technische Universität Darmstadt

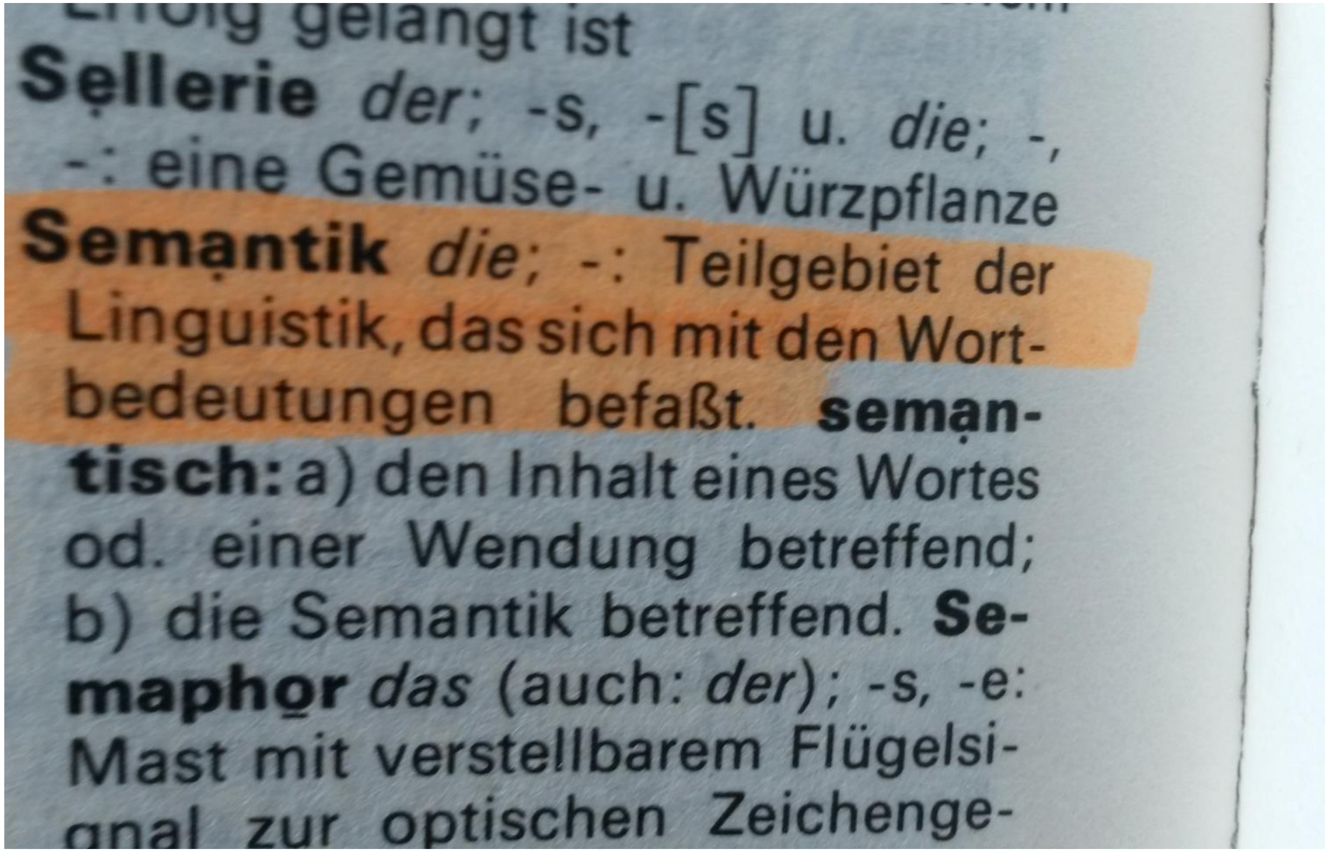
Outline

Word meaning

How can we represent words?

TECHNISCHE
UNIVERSITÄT
DARMSTADT

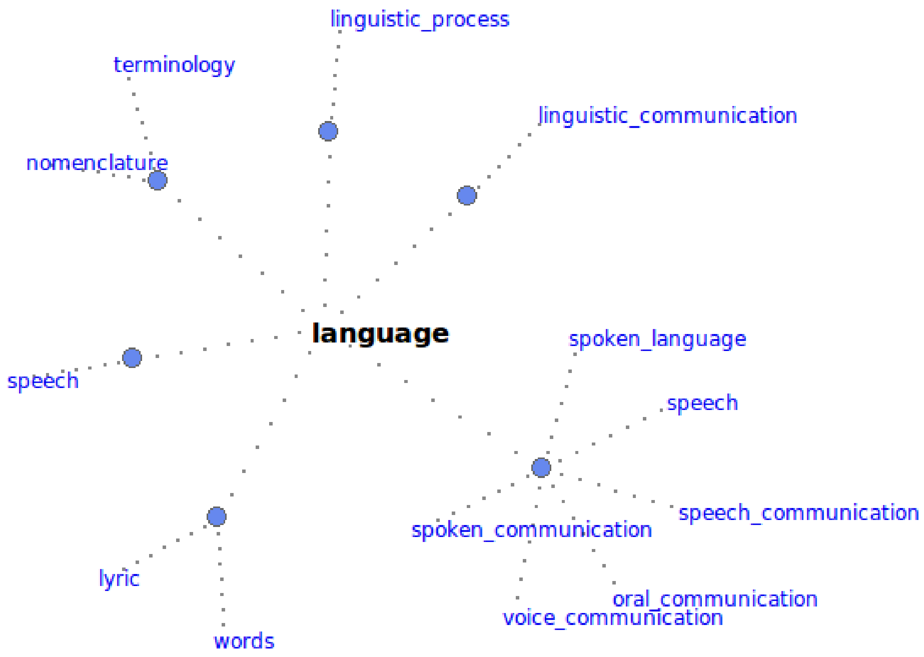
- As a dictionary entry


gelangt ist
Sellerie *der;* -s, -[s] u. *die;* -, -: eine Gemüse- u. Würzpflanze
Semantik *die;* -: Teilgebiet der Linguistik, das sich mit den Wortbedeutungen befaßt. **semantisch:** a) den Inhalt eines Wortes od. einer Wendung betreffend; b) die Semantik betreffend. **Semaphor** *das* (auch: *der*); -s, -e: Mast mit verstellbarem Flügelsignal zur optischen Zeichenge-

NLLG

Taxonomy of words

- Represent words by their **relations to other words**



Picture from: http://kylescholz.com/projects/wordnet/, based on representation from WordNet: https://wordnet.princeton.edu

Word vectors

- “One-hot” vector, sparse representation

$$\begin{array}{cccccc}
\text{der} & \text{die} & \text{und} & \text{in} & \text{....} & \text{für} \\
\begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\\ldots\end{bmatrix} &
\begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\\0\\0\\\ldots\end{bmatrix} &
\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\\ldots\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\\ldots\end{bmatrix} &
 &
\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\1\\\ldots\end{bmatrix}
\end{array}$$

- Dimensionality of vector equals size of vocabulary

Word vectors

- “One-hot” vector, sparse representation

$$
\overset{\text{der}}{\begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\\\dots\end{bmatrix}}
\quad
\overset{\text{die}}{\begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\\0\\0\\\dots\end{bmatrix}}
\quad
\overset{\text{und}}{\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\\0\\\dots\end{bmatrix}}
\quad
\overset{\text{in}}{\begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\\0\\\dots\end{bmatrix}}
\quad \overset{\dots}{} \quad
\overset{\text{für}}{\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\1\\\dots\end{bmatrix}}
$$

Problem: relations between words are not represented

Word meaning can be inferred from context

- Famous example by McDonald and Ramscar (2001):

    1. He filled the **wampimuk**, passed it around and we all drank some.

Word meaning can be inferred from context

- Famous example by McDonald and Ramscar (2001):

  1. He filled the **wampimuk**, passed it around and we all drank some.

     jar

     cup

     glass

     goblet

     …

Word meaning can be inferred from context

- Famous example by McDonald and Ramscar (2001):

1. He filled the wampimuk, passed it around and we all drank some.

   jar

   cup

   glass

   goblet

   …

2. We found a little hairy **wampimuk** sleeping behind the tree.

Word meaning can be inferred from context

TECHNISCHE UNIVERSITÄT DARMSTADT

- Famous example by McDonald and Ramscar (2001):

1. He filled the wampimuk, passed it around and we all drank some.

   jar
   cup
   glass
   goblet
   …

2. We found a little hairy **wampimuk** sleeping behind the tree.

   cat
   bear
   racoon
   mole
   …

Distributional hypothesis

- Firth (1957): ***“You shall know a word by the company it keeps.”***

Outline

Computational semantics: Count models

How can we model the distributional hypothesis?

- By calculating co-occurrence counts
  - capture in which contexts a word appears
- Context is modeled using a window over the words

- Consider the following example by Richard Socher:
- Corpus
  - *I like deep learning .*
  - *I like NLP .*
  - *I enjoy flying .*
- Window size = 1, left and right neighbor
  - In real tasks, window size is usually bigger (5-10)

How can we model the distributional hypothesis?

- By calculating co-occurrence counts
  - capture in which contexts a word appears
- Context is modeled using a window over the words

Such models have been called „count models“ in the literature
See: Baroni et al. Don‘t count predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In: ACL 2014

- Consider the following example by Rich
- Corpus
  - *I like deep learning .*
  - *I like NLP .*
  - *I enjoy flying .*
- Window size = 1, left and right neighbor
  - In real tasks, window size is usually bigger (5-10)

Co-occurrence matrix

- Example by Richard Socher:
  - *I like deep learning .*
  - *I like NLP .*
  - *I enjoy flying .*

Window size = 1

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Co-occurrence matrix

- Example by Richard Socher:
  - *I like deep learning .*
  - *I like NLP .*
  - *I enjoy flying .*

Window size = 1

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Co-occurrence matrix

TECHNISCHE UNIVERSITÄT DARMSTADT

- Example by Richard Socher:
  - *I like deep learning .*
  - *I like NLP .*
  - *I enjoy flying .*

Window size = 1

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

NLLG

Co-occurrence matrix

TECHNISCHE UNIVERSITÄT DARMSTADT

- Example by Richard Socher:
  - *I like deep learning .*
  - *I like NLP .*
  - *I enjoy flying .*

Window size = 1

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Co-occurrence counts

- Assumption: If we collect co-occurrence counts over thousands of sentences, the vectors for “enjoy” and “like” will have similar vector representations.

Co-occurrence counts

- Assumption: If we collect co-occurrence counts over thousands of sentences, the vectors for “enjoy” and “like” will have similar vector representations.

- Problem:
  - Vectors become very large with real data
    → We need to apply dimensionality reduction

Outline

Computational semantics: NN models

Background idea: language models

- Based on the concept of language modeling:
- Common problem in NLP, popular application is auto-completion
  - Given a sequence of words, predict the following word
  - *The same procedure as every* ______

- Idea: Language modeling is too restrictive because it only considers the left context. What about the right context?

word2vec

- Most popular toolkit for training word representations:

word2vec

https://code.google.com/archive/p/word2vec/

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean:
Distributed Representations of Words and Phrases and their Compositionality
In Proceedings of NIPS, 2013.

- Two different setups/auxiliary tasks: CBOW and Skip-gram

Auxiliary Tasks

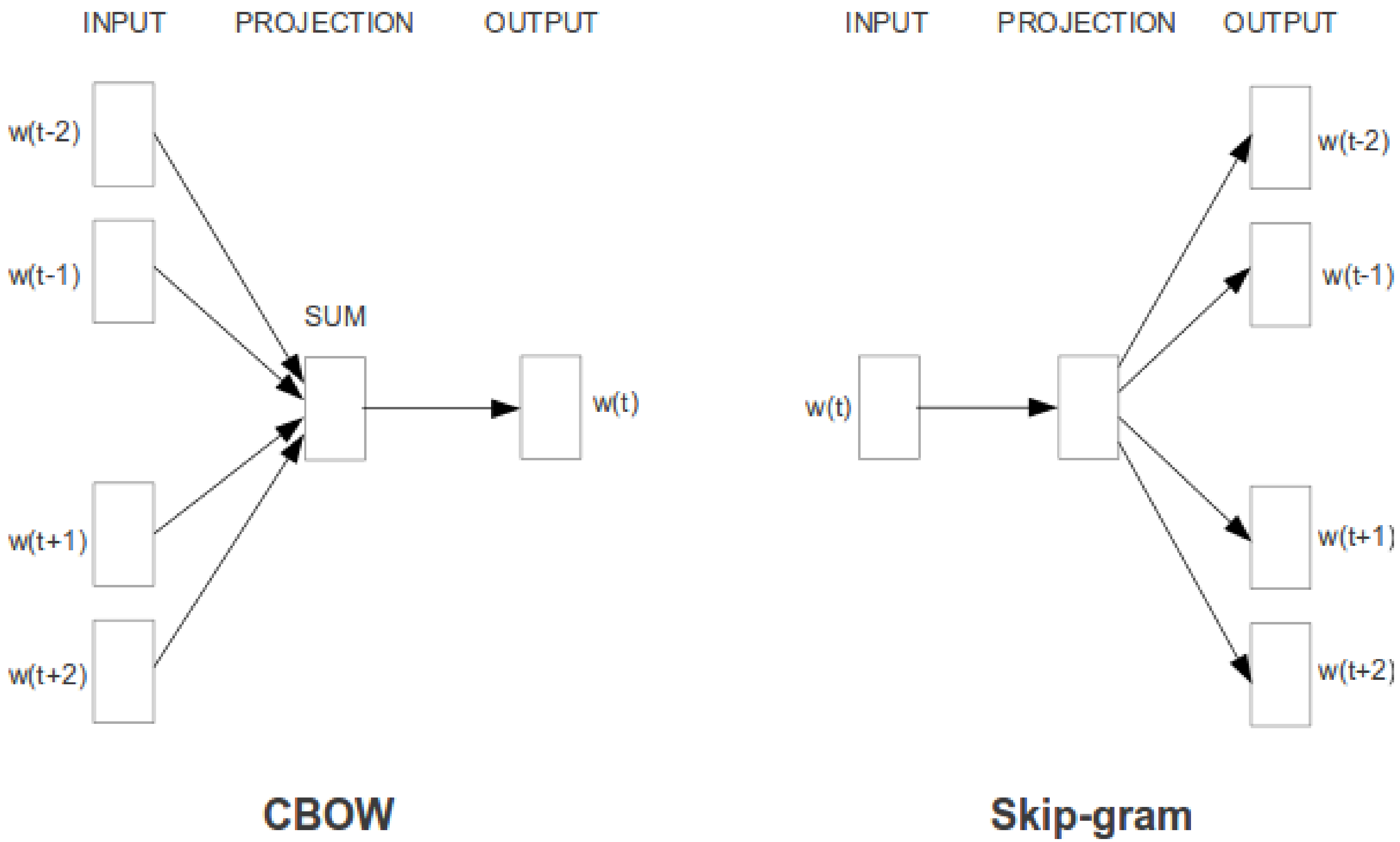
- CBOW: Given a context, predict the missing word
  - *same procedure ____ every year*
  - *as long ___ you sing*
  - *please stay ___ you are*

- Skip-gram: given a word, predict the context words
  - ____ _______ *as* ______ _______
  - If window size is two, we aim to predict:

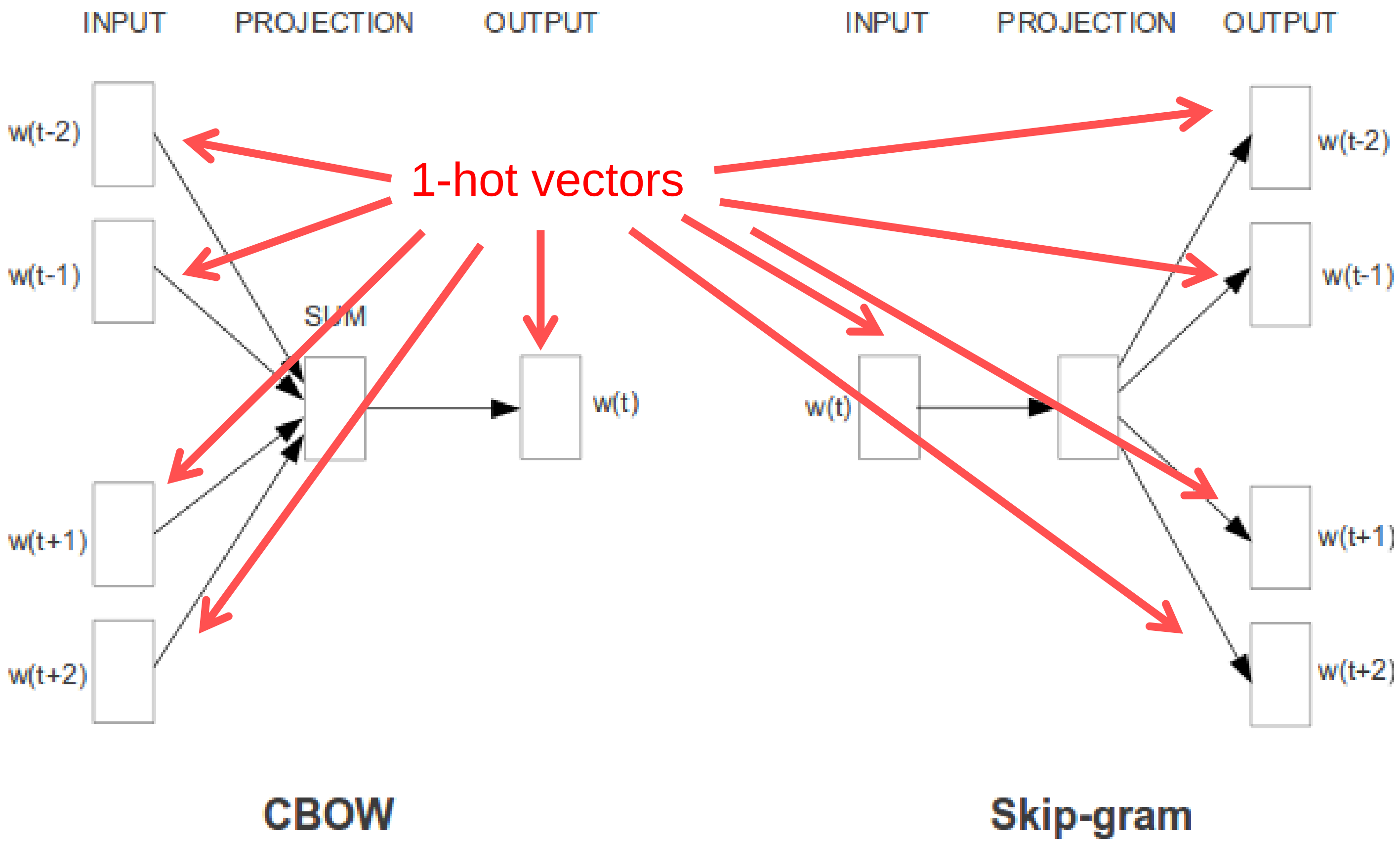
    (w,c_{-2}), (w,c_{-1}), (w,c_{1}) and (w,c_{2})

- Note that order information is not preserved, i.e. we do not distinguish whether a word is more likely to occur before or after the word

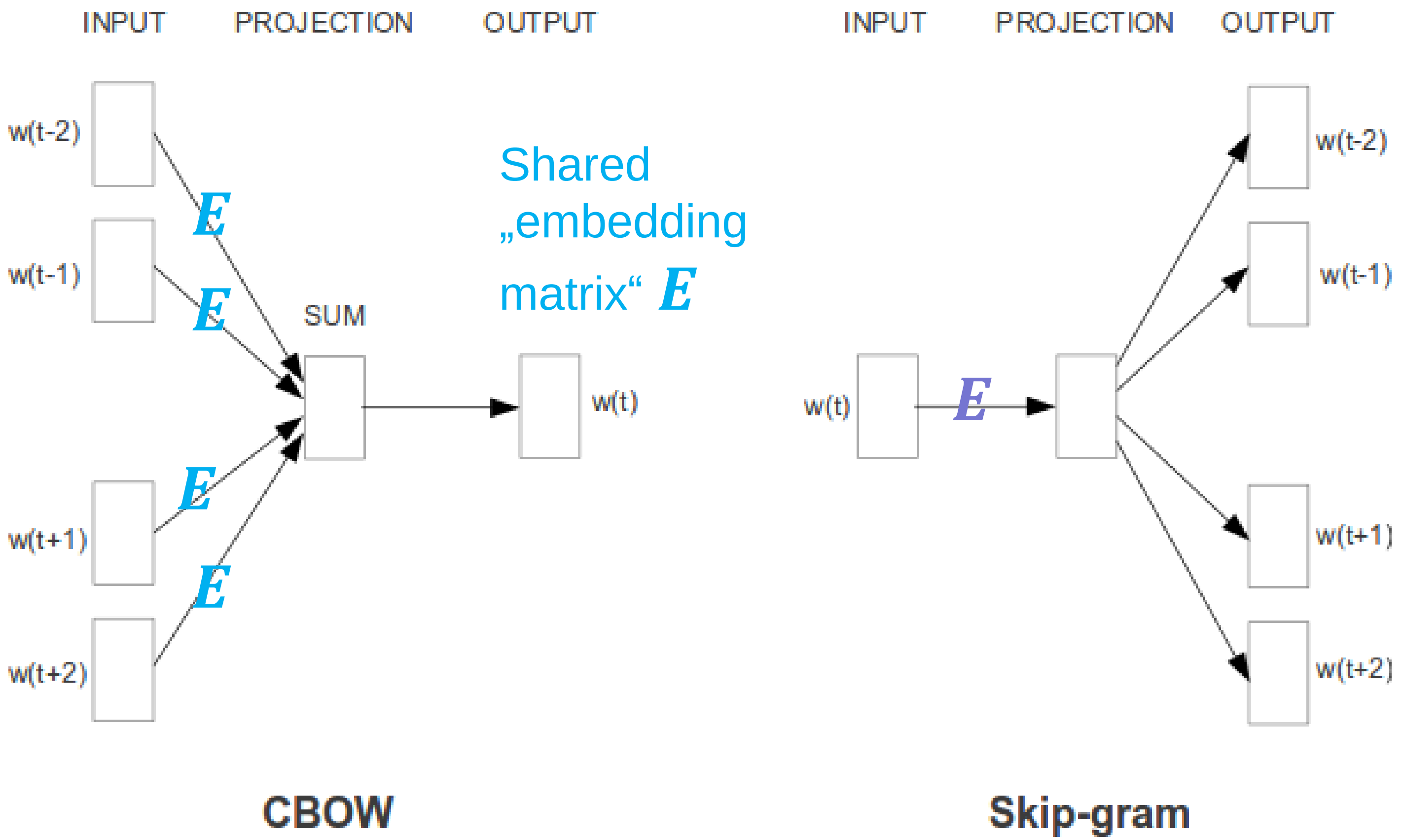
CBOW vs Skip-gram



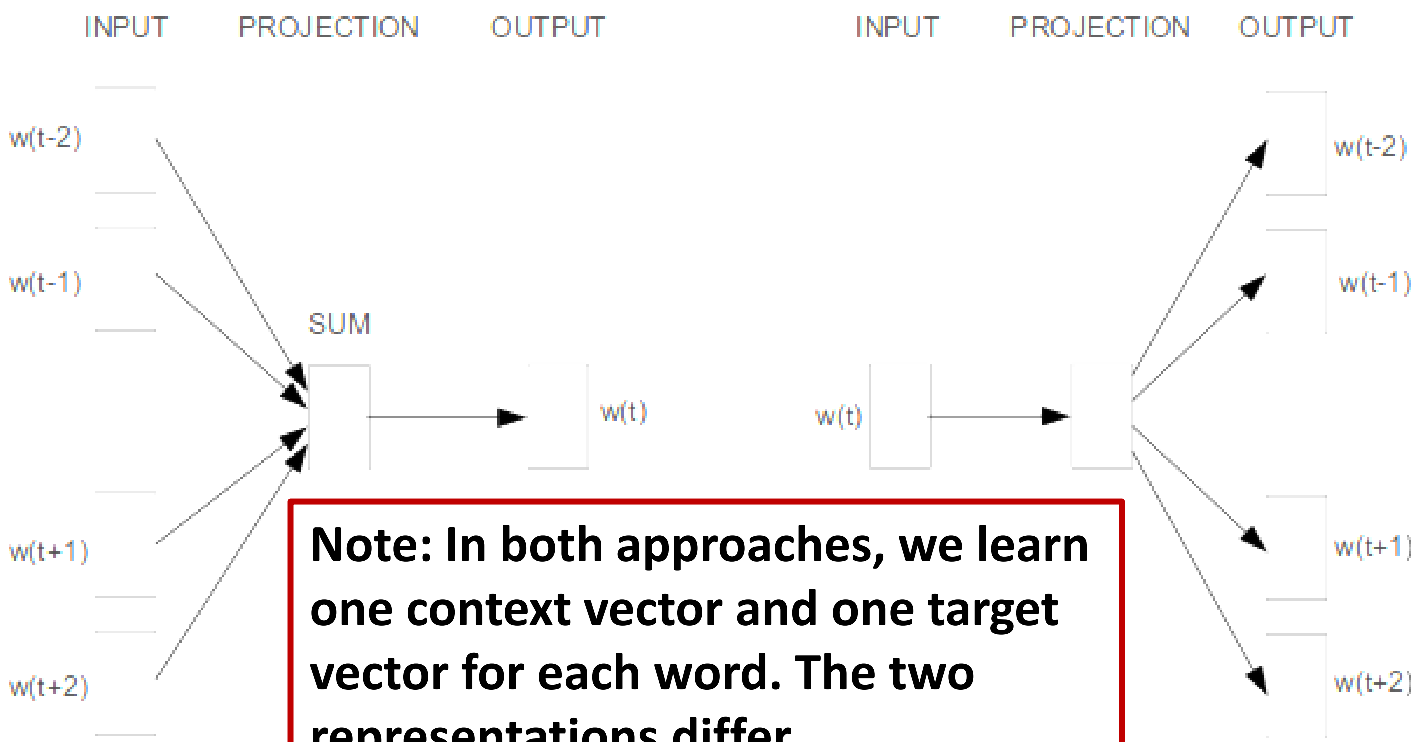
CBOW vs Skip-gram



CBOW vs Skip-gram



Tasks:



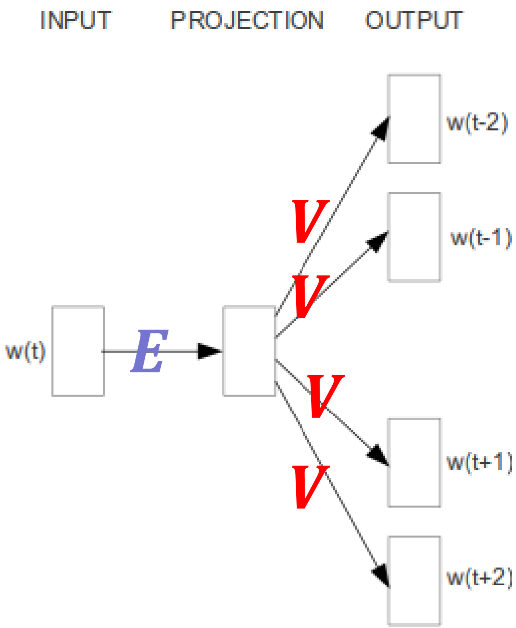
Note: In both approaches, we learn one context vector and one target vector for each word. The two representations differ.

CBOW Skip-gram

The Skip-gram model: Theory

- We'll take a closer look at the Skip-Gram model

  - $\boldsymbol{E}$ has dimension $N \times d$
  - $\boldsymbol{V}$ has dimension $d \times N$
  - N is number of words in the vocabulary
  - d is the embedding dimension

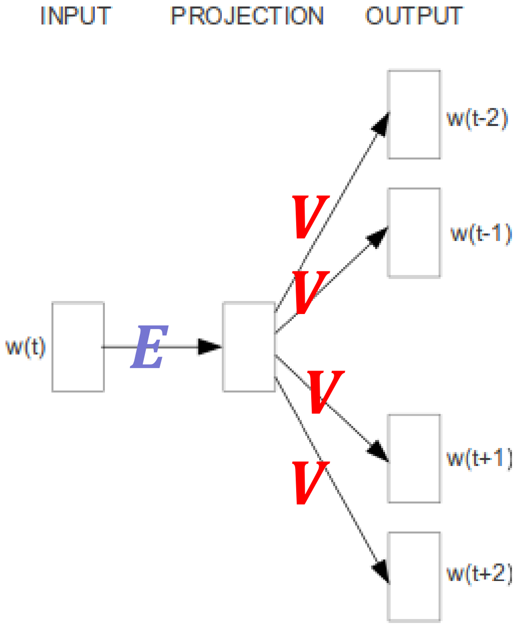


Skip-gram

The Skip-gram model: Theory

- We'll take a closer look at the Skip-Gram model

  - $\boldsymbol{E}$ has dimension $N \times d$
  - $\boldsymbol{V}$ has dimension $d \times N$
  - N is number of words in the vocabulary
  - d is the embedding dimension



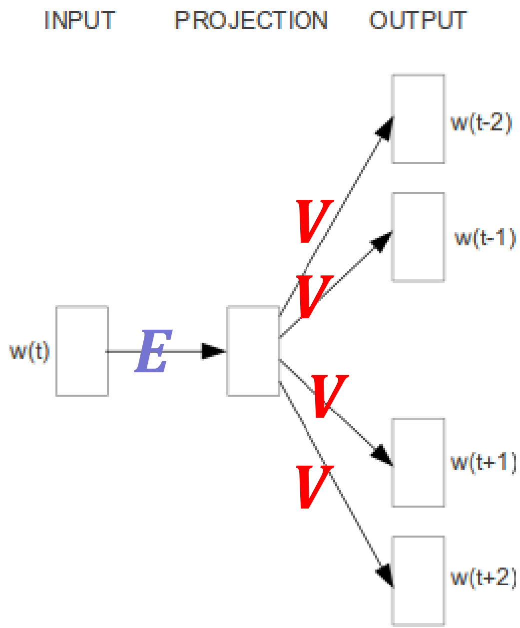
Skip-gram

How can we predict different words when V is always the same?

The Skip-gram model: Theory

- We'll take a closer look at the Skip-Gram model

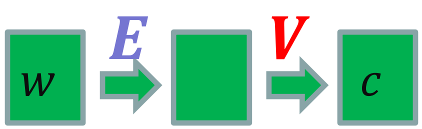
  - $\boldsymbol{E}$ has dimension $N \times d$
  - $\boldsymbol{V}$ has dimension $d \times N$
  - N is number of words in the vocabulary
  - d is the embedding dimension



Skip-gram

How can we predict different words when V is always the same?

Turns out the original model is actually this:



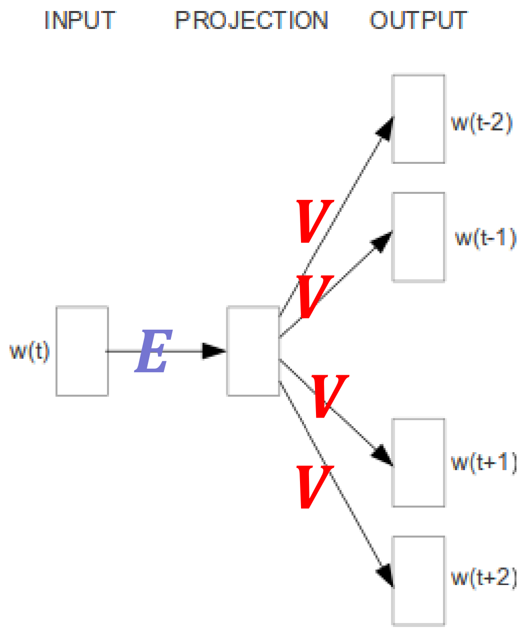
The Skip-gram model: Theory

- We'll take a closer look at the Skip-Gram model

  - $\boldsymbol{e} = \boldsymbol{e}(\boldsymbol{w}) = \boldsymbol{wE}$ has dimension $1 \times d$: It's the embedding of word w

  - The activation of the projection layer is *linear (= identity: f(x)=x)*

  - $\boldsymbol{eV}$ has dimension $1 \times N$
  - $\boldsymbol{V} = [\boldsymbol{v}_1 \cdots \boldsymbol{v}_N]$: each $\boldsymbol{v}_i$ can be seen as an(other) embedding of a vocab. word

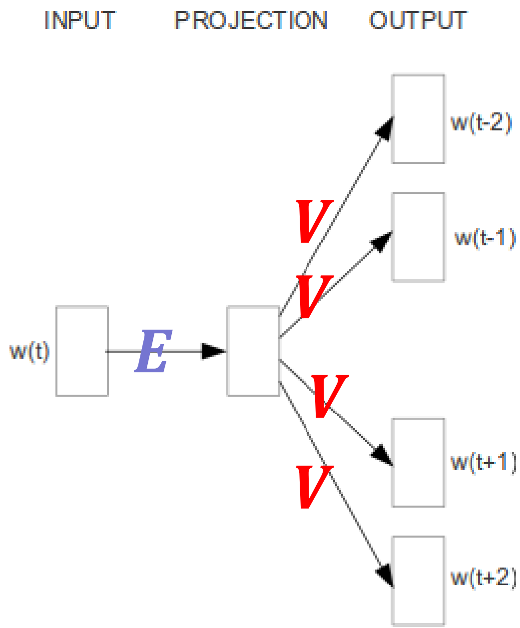


Skip-gram

The Skip-gram model: Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

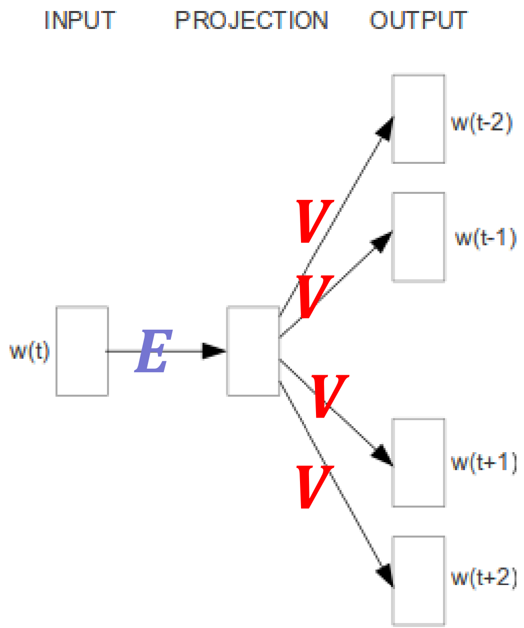
- We‘ll take a closer look at the Skip-Gram model
  - $\boldsymbol{eV} = [\boldsymbol{ev}_1, \dots, \boldsymbol{ev}_N]$ has dimension $1 \times N$
  - $\boldsymbol{V} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_N]$: each $\boldsymbol{v}_i$ can be seen as an(other) embedding of a vocab. word
  - The output layer has softmax activation function
  - softmax($\boldsymbol{eV}$)= $[\exp(\boldsymbol{ev}_1), \dots, \exp(\boldsymbol{ev}_N)]$ / Z, where Z is normalizer



Skip-gram

The Skip-gram model: Theory

- We‘ll take a closer look at the Skip-Gram model
  - We could just run this model with SGD
  - With the methods we learned
  - When all matrices are learned, we‘re interested in the $\boldsymbol{E}$ matrix, which holds the word embeddings
  - **However, the practical implementation is different from this (see below)**



Skip-gram

The Skip-gram model: Illustration

TECHNISCHE UNIVERSITÄT DARMSTADT

- Preparation:

  - Download a lot of (**unlabeled**) data, e.g. all the poems of W.S.

All the world's a stage and all the men and women merely players. They totter …

  - Tokenize it

All the world 's a stage and all the men and women merely players . They totter …

  - For word2vec: Define either one of two **auxiliary** tasks: predict middle words or predict contexts

For skip-gram:

All the world 's a stage and all the men and women merely players . They totter …

- Training data (maybe this is our first batch):
  - x=world t=the
  - x=world t=All
  - x=world t=‘s
  - x=world t=a

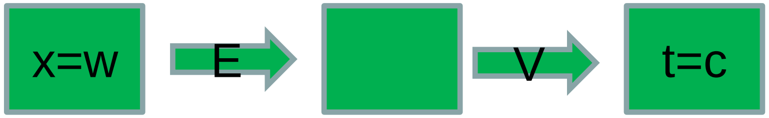
  - Of course, the 1-hot vectors of this

For skip-gram:

All the world 's a stage and all the men and women merely players . They totter …

- Training data (maybe this is our first batch):
  - x=world t=the
  - x=world t=All
  - x=world t='s
  - x=world t=a

Feed in to the network; update params



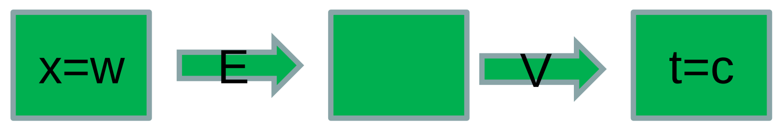
  - Of course, the 1-hot vectors of this

For skip-gram:

All the world 's a stage and all the men and women merely players . They totter …

- Training data (maybe this is our first batch):
  - x=world   t=the
  - x=world   t=All
  - x=world   t=‘s
  - x=world   t=a

Feed in to the network; update params



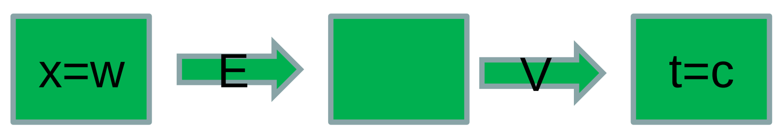
  - Of course, the 1-hot vectors of this
- Notation: We write (x,t) or (w,c) for a training data sample

For skip-gram:

All the world 's a stage and all the men and women merely players . They totter …

- Training data (maybe this is our second batch):
  - x='s t=world
  - x='s t=the
  - x='s t=a
  - x='s t=stage

Feed in to the network; update params



  - Of course, the 1-hot vectors of this
- Notation: We write (x,t) or (w,c) for a training data sample

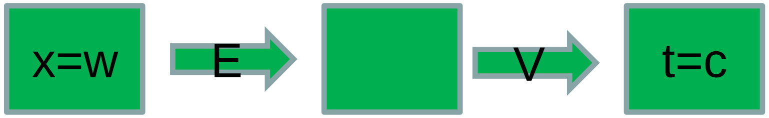
For skip-gram:

All the world 's a stage and all the men and women merely players . They totter …

Context window size is a hyperparam.

- Training data (maybe this is our second batch):
  - x='s t=world
  - x='s t=a

Feed in to the network; update params



  - Of course, the 1-hot vectors of this
- Notation: We write (x,t) or (w,c) for a training data sample

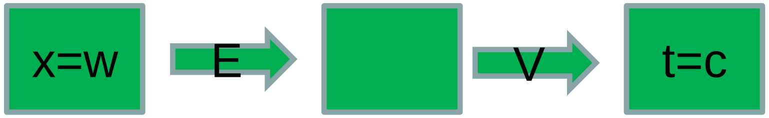
For skip-gram:

All the world 's a stage and all the men and women merely players . They totter …

Context window size is a hyperparam.

- Training data (maybe this is our second batch):
  - x=‘s t=All
  - x=‘s t=the
  - x=‘s t=world
  - x=‘s t=a
  - x=‘s, t=stage, x=‘s, t=and
  - Of course, the 1-hot vectors of this
- Notation: We write (x,t) or (w,c) for a training data sample

Feed in to the network; update params



Toolkits for training word representations

word2vec

https://code.google.com/archive/p/word2vec/

GloVe

http://nlp.stanford.edu/projects/glove/

- GloVe aims at reconciling the advantages of global co-occurrence counts and local context windows
- Applies additional trick: take the sum of the target/center vector **e(w)** and the context vector $\mathbf{v}_c$ of each word as representation
- Many more, but these are two popular ones
- Terminology:
  - context-counting vs context-predicting representations, sparse vs dense
  - matrix-factorization methods vs shallow window-based approaches
  - **word representations ≈ word embeddings ≈ word vectors**

Pre-Trained Embeddings

TECHNISCHE UNIVERSITÄT DARMSTADT

- Word2vec
  - trained on Google news (100 billion tokens)
  - vectors with Freebase naming, trained on news (100 billion tokens)
- GloVe
  - trained on Wikipedia (6 billion tokens)
  - trained on CommonCrawl (42 and 840 billion tokens)
  - trained on Twitter (27 billion tokens)

- Omer Levy: dependency-based embeddings trained on Wikipedia
  https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/

- There are many embeddings nowadays, in all possible languages
  https://fasttext.cc/docs/en/crawl-vectors.html

Outline

Evalution of word embeddings

Evaluating word representations

- Extrinsic

  by the performance of a model that uses the word representations for solving a task

  - Named entity recognition (accuracy), machine translation (BLEU score), summarization (ROUGE score), information retrieval (coverage)…
  - Compare performance of two models that only differ in the word representations they use

- Intrinsic

  by using the representations directly

  - Word Similarity Task
  - Word Analogy Task
  - Word Intrusion Task

Extrinsic Evaluation: Setup

- Say, our task is POS tagging

- Our labeled training data

Word	Label
The	DET
cat	NN
on	PREP
the	DET
mat	NN
.	PUNC

Extrinsic Evaluation: Setup

- Say, our task is POS tagging

- Our labeled training data; replace words with their embeddings

x	t
E(The)	1-hot(DET)
E(cat)	1-hot(NN)
E(on)	1-hot(PREP)
E(the)	1-hot(DET)
E(mat)	1-hot(NN)
E(.)	1-hot(PUNC)

Extrinsic Evaluation: Setup

- Say, our task is POS tagging

- Our labeled training data; replace words with their embeddings; usually add some context

x	t
E(SOS);E(The);E(cat)	1-hot(DET)
E(The);E(cat);E(on)	1-hot(NN)
E(cat); E(on); E(on)	1-hot(PREP)
E(on); E(the); E(mat)	1-hot(DET)
E(the); E(mat); E(.)	1-hot(NN)
E(mat); E(.); E(EOS)	1-hot(PUNC)

Extrinsic Evaluation: Setup

- Say, our task is POS tagging

- Our labeled training data; replace words with their embeddings; usually add some context [not necessary with other architectures such as RNN; see later lectures]

x	t
E(SOS);E(The);E(cat)	1-hot(DET)
E(The);E(cat);E(on)	1-hot(NN)
E(cat); E(on); E(on)	1-hot(PREP)
E(on); E(the); E(mat)	1-hot(DET)
E(the); E(mat); E(.)	1-hot(NN)
E(mat); E(.); E(EOS)	1-hot(PUNC)

- Now train with different embeddings and look which one performs best

Intrinsic: Word Similarity Task

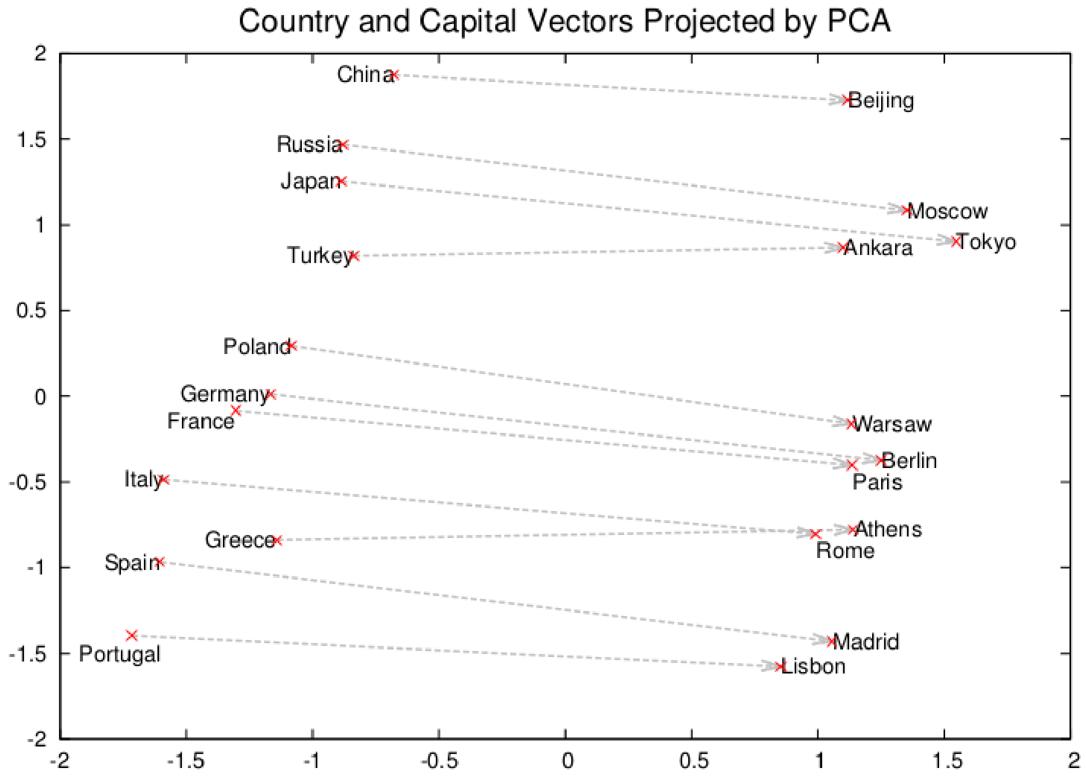
- Similar words should have similar representations
  - Similarity Dataset: http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/
    Scores from 0 to 10 by human raters
  - Intrinsic evaluation of embeddings:
    → quantify similarity by distance between word vectors
    → evaluate correlation with juman judgements

Word 1	Word 2	Human (mean)	Learned vectors
tiger	cat	7.35	dist(tiger, cat)
book	paper	7.46	dist(book, paper)
plane	car	5.77	dist(plane, cat)
smart	student	4.62	dist(smart, student)
stock	phone	1.62	dist(stock, phone)
....			
			....

Relations between word vectors

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mikolov et al. (2013)



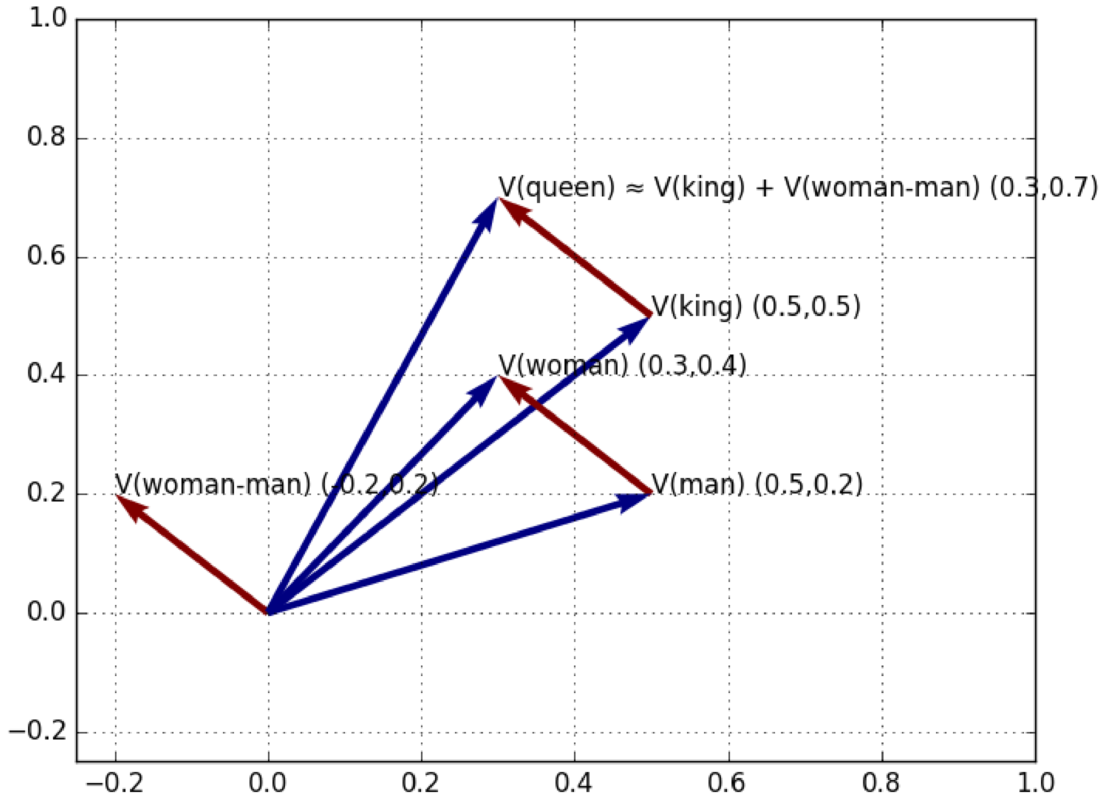
How to find analogies?

- A is to B as C to ?
- Germany is to Berlin as France to x
- Find x such that:
  - vec(x) = vec(“Berlin”) - vec(“Germany”) + vec(“France”)

- Most famous example:

$$\text{KING} - \text{MAN} + \text{WOMAN} = \text{QUEEN}$$

KING-MAN+WOMAN=QUEEN



Semantic analogies

- All examples from: //code.google.com/p/word2vec/source/browse/trunk/questions-words.txt
- capital-common-countries
  - *Athens Greece Baghdad* ***Iraq***
  - *Athens Greece Berlin* ***Germany***
- currency
  - *Denmark krone Croatia* ***kuna***
  - *Europe euro Hungary* ***forint***
- family
  - *boy girl brother* ***sister***
  - *brother sister dad* ***mom***

Syntactic analogies

- adjective-to-adverb
  - *amazing amazingly apparent* ***apparently***
- comparative
  - *bad worse big* ***bigger***
- present-participle
  - *code coding dance* ***dancing***
- past-tense
  - *dancing danced decreasing* ***decreased***
- plural
  - *banana bananas bird* ***birds***
- 3rd person verbs
  - *decrease decreases eat* ***eats***

Practical Guidelines

- Word2Vec and Glove are pretty good tools
- Fast, give good word embeddings
- However, many other embeddings out there (see next lectures)
- Always try out different embeddings --- consider them as another hyperparameter
  - Results may vary drastically with different embeddings

Summary

- Vectors are useful for representing words
  - Dense vs sparse representations
  - Projecting co-occurrence counts to low-dimensional vectors vs directly learning low-dimensional vectors
- Learning low-dimensional vectors
  - Inspired by neural language modeling
  - CBOW and Skip-gram model
  - Negative sampling
- Evaluating word representations
  - Extrinsic vs intrinsic evaluation

- Terminology:

word representations ≈ word embeddings ≈ word vectors

Mandatory reading

- Faruqui et al. (2016), Problems With Evaluation of Word Embeddings Using Word Similarity Tasks

References

- Levy & Goldberg: word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method, preprint on arxiv: *arXiv:1402.3722,* 2014.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J.: Distributed representations of words and phrases and their compositionality, in *Advances in neural information processing systems*: 3111-3119, 2013.
- McDonald, S., & Ramscar, M.: Testing the distributional hypothesis: The influence of context on judgements of semantic similarity, in *Proceedings of the 23rd annual conference of the Cognitive Science Society*, 2001.
- Bengio, Y., Schwenk, H., Senécal, J. S., Morin, F., & Gauvain, J. L.: Neural probabilistic language models, in *Innovations in Machine Learning*: 137-186, 2006.
- Collobert, R., & Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning, in *Proceedings of the 25th international conference on Machine learning*: 160-167, 2008.
- Derweester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., & Harshman, R.: Indexing by latent semantic analysis , in *Journal of the American Society for Information Science 41*: 391-407, 1990.
- Harris, Z. S.: Distributional structure. *Word 10(2-3)*: 146-162, 1954.
- John R. Firth: A synopsis of linguistic theory 1930–55, in *Studies in Linguistic Analysis* (special volume of the Philological Society), 1–32, 1957.

References

- Baroni and, M., Dinu, B. & Kruszewski, G.: Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014
- Levy, O. & Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization, in *Advances in Neural Information Processing Systems 27*, 2014
- Arora, S., Li, Y., Liang, Y., Ma, T. & Risteski, A.: A Latent Variable Model Approach to PMI-based Word Embeddings, in *Transactions of the Association for Computational Linguistics*, 2016
- Stratos, K., Collins, M. & Hsu, D.: Model-based Word Embeddings from Decompositions of Count Matrices, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015
- Faruqui, M., Tsvetkov, Y., Rastogi, P., Dyer, C.: Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. CoRRabs/1605.02276, 2016
- Ma, X. & Hovy, E.: End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, *arXiv:1603.01354*, 2016