Word Embeddings

Nils Reimers

Course-Website: www.deeplearning4nlp.com

Possible Representations for Words

TECHNISCHE UNIVERSITÄT DARMSTADT

- Many NLP systems regards words as atomic symbols
- In vector terms, this is a vector with one 1 and a lot of zeros:

Hotel	[1 0 0 0]	Dog	[0 0 1 0]
Motel	[0 1 0 0]	Cat	[0 0 0 1]

- Its problem:

Hotel	[1 0 0 0]	AND	
Motel	[0 1 0 0]	=	0

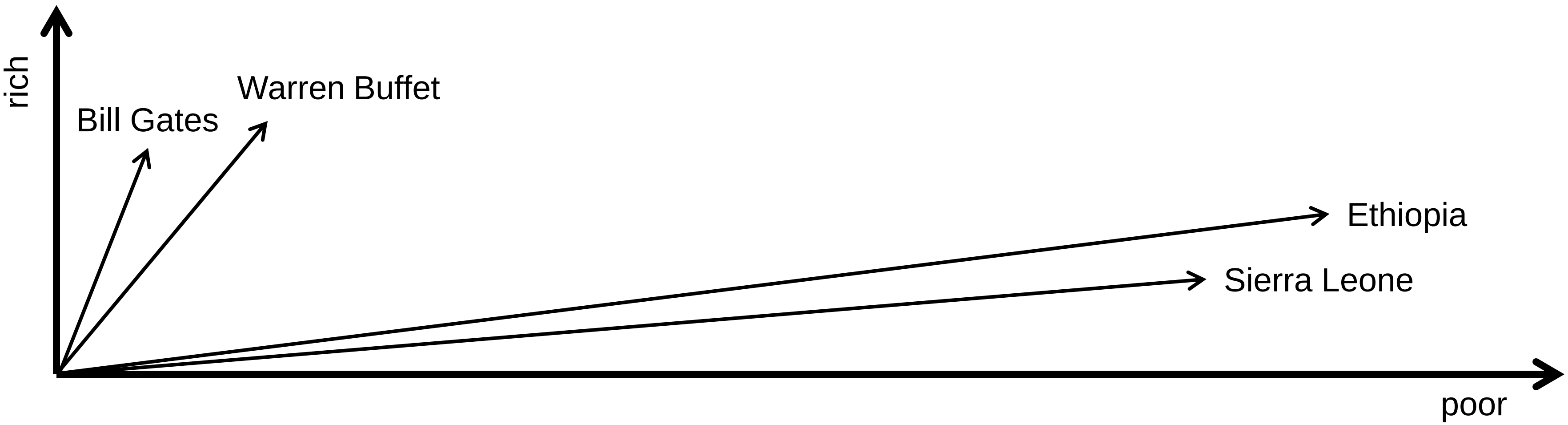
- Machine learning systems cannot derive useful information from this 1-hot representation
  - Impossible to keep-up with synonyms and new words

Distributional Hypothesis

- *Words that occur in the same contexts tend to have similar meanings* (Harris, 1954)

- One of the most successful ideas in modern NLP
- Hugely boost the performance if used correctly

- Idea: Count the co-occurrence of tokens:



Co-occurence Matrix

TECHNISCHE UNIVERSITÄT DARMSTADT

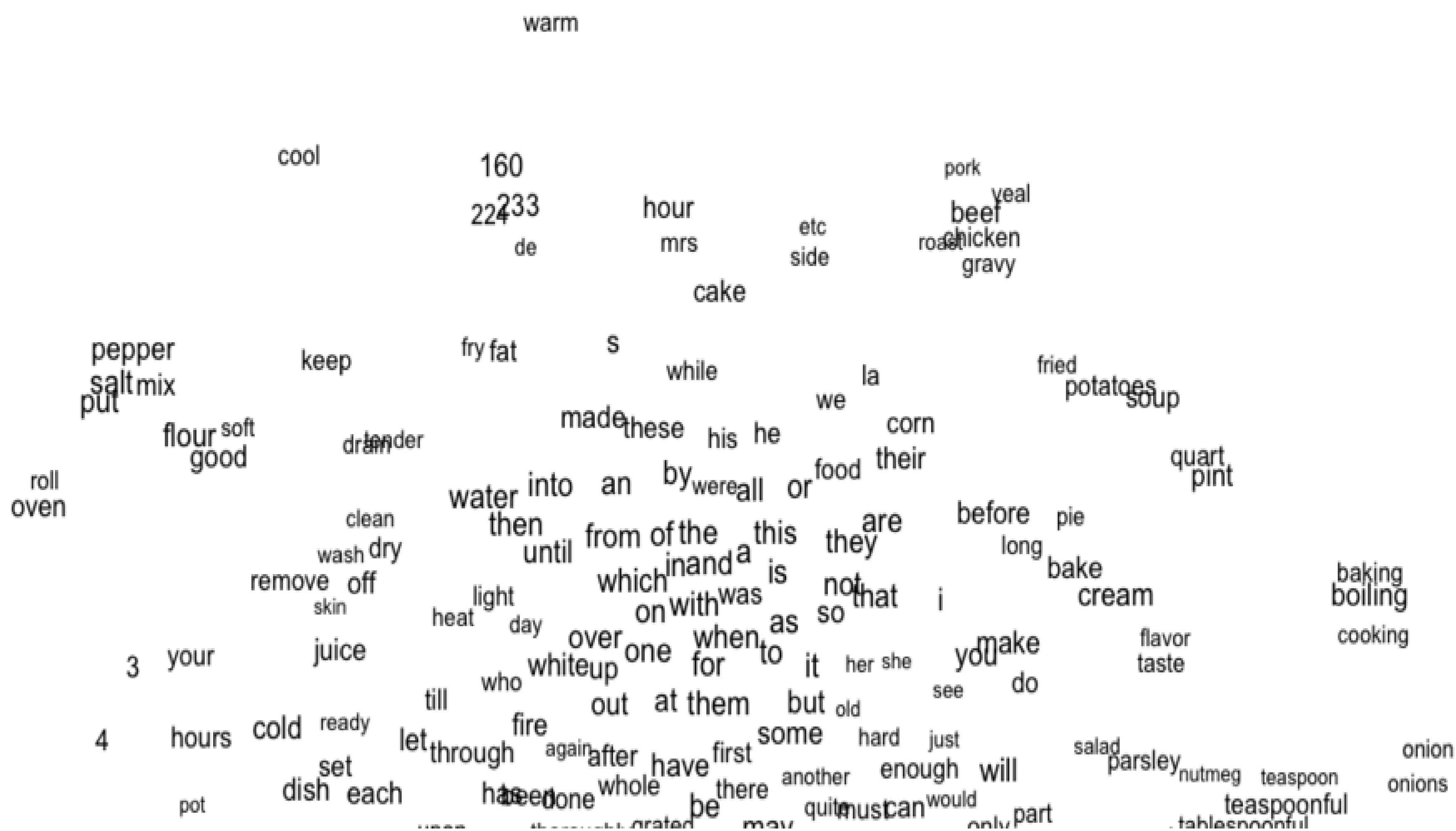
- Create a matrix of the co-occurences for all words

	word1	word2	word3	word4	word5	word6	word7
word1	0	2	0	3	5	0	1
word2		0	1	5	2	0	3
word3			0	1	0	0	1
word4				0	6	0	1

- Problems:
  - Increases with the size of vocabulary
  - High dimensional – requires a lot memory
  - Subsequent steps have sparsity issues
- Solution:
  - Use dimensionality reduction to store only the most important information

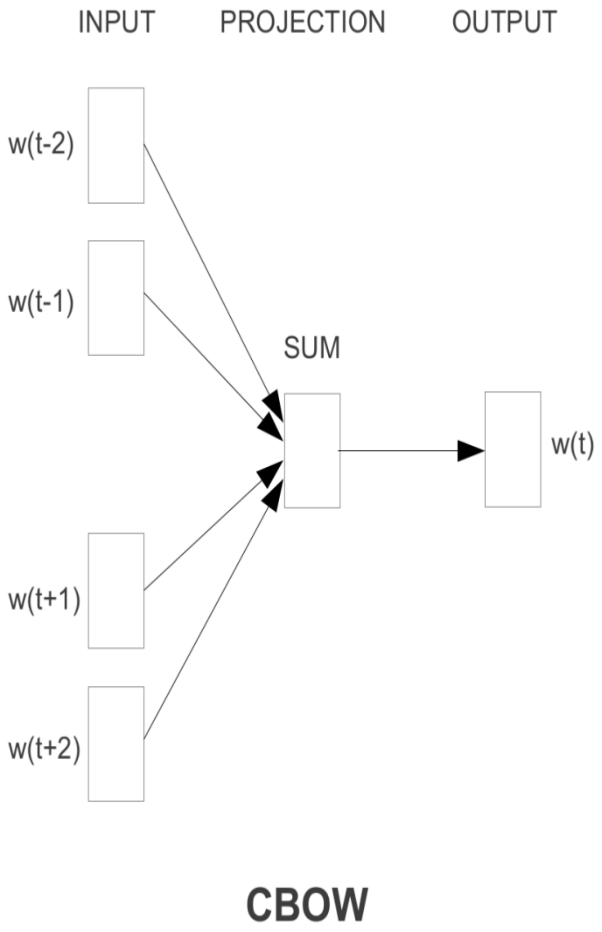
Low Dimensionality Representation of Words

TECHNISCHE UNIVERSITÄT DARMSTADT



UKP

Representation Learning for Words
Word2vec - CBOW



- Word2vec (Mikolov et al.): Instead of creating co-occurrence matrix, create low-dimensional vectors directly
- Highly efficient for large corpora (>100 TB)

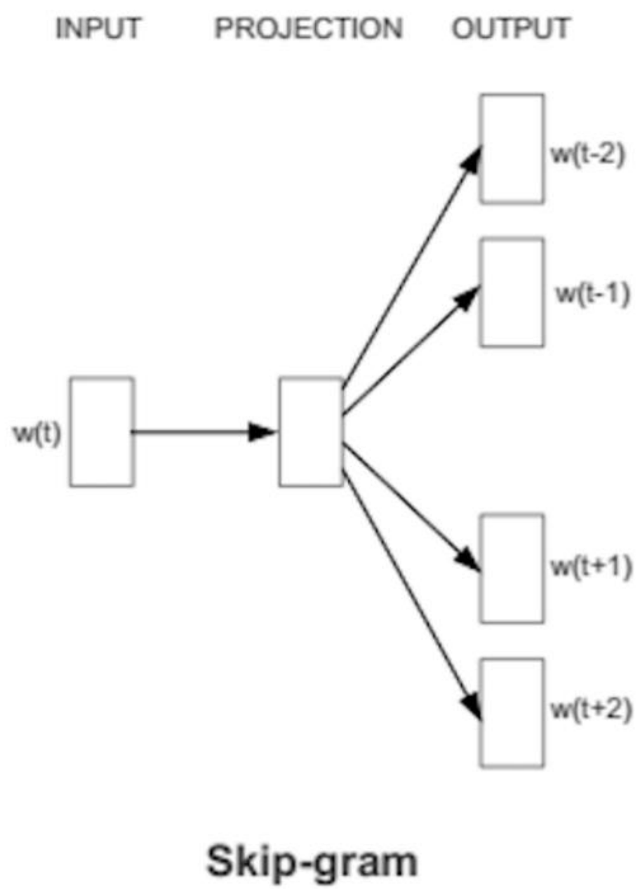
CBOW-Model:

- Given the surrounding words,
  try to predict word: w(t)
- Idea:
  score(cat chills **on** a mat) >
  score(cat chills **French** a mat)
- Maximize the distance between **on** and **French** for the given phrase
- Result: Similar words are close, dissimilar words are far apart in vector space

Mikolov et al., 2013

Representation Learning for Words

Word2vec – Skip-Gram

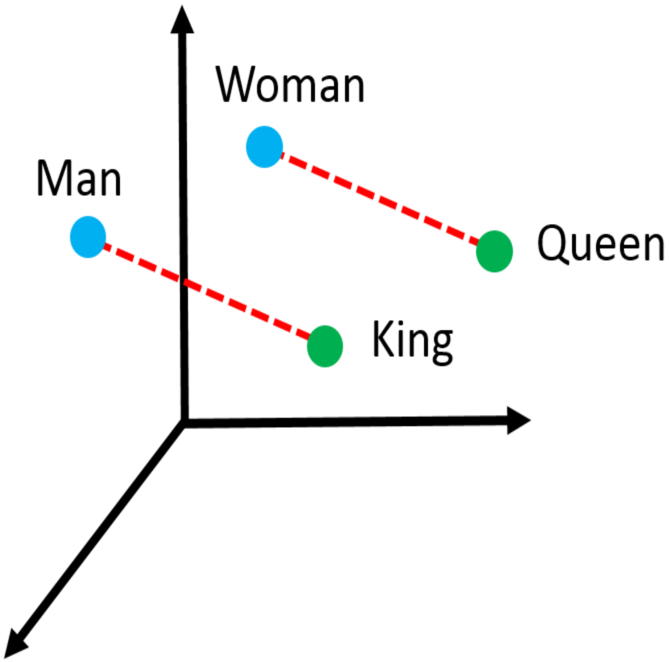


- Given center word w(t), try to predict context words w(t-2), …, w(t+2)
- Usually works better than CBOW
- What a context word is, can be quite arbitrarily:
  - Words to the left & right of the word
  - Dependency relations
  - Relations in FreeBase / WikiData etc.
- Different contexts create different embeddings
  - Which is most suitable depends on the task

Mikolov et al., 2013

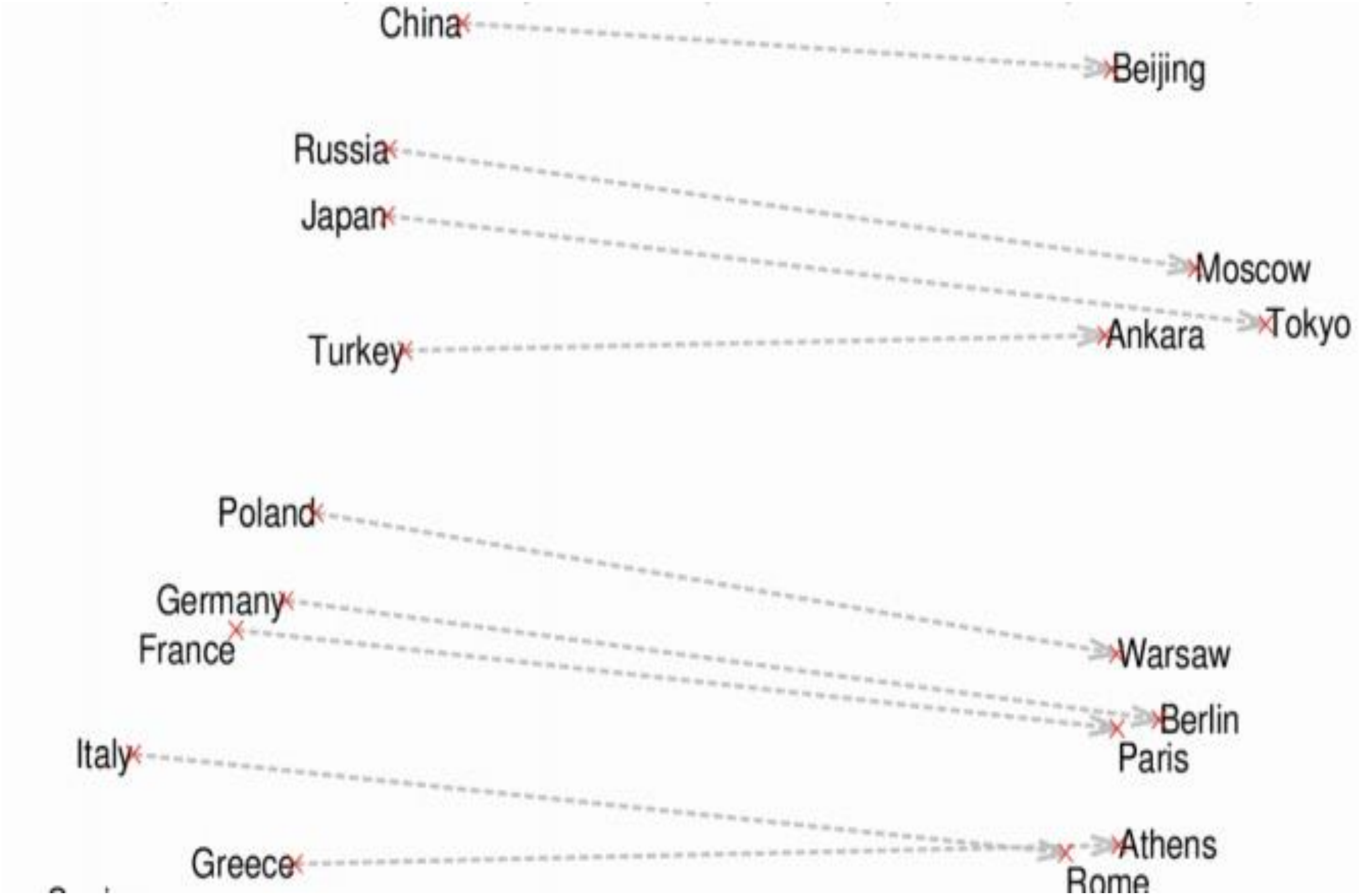
Representation of Words

- Syntactic and semantic properties are captured in this vector space

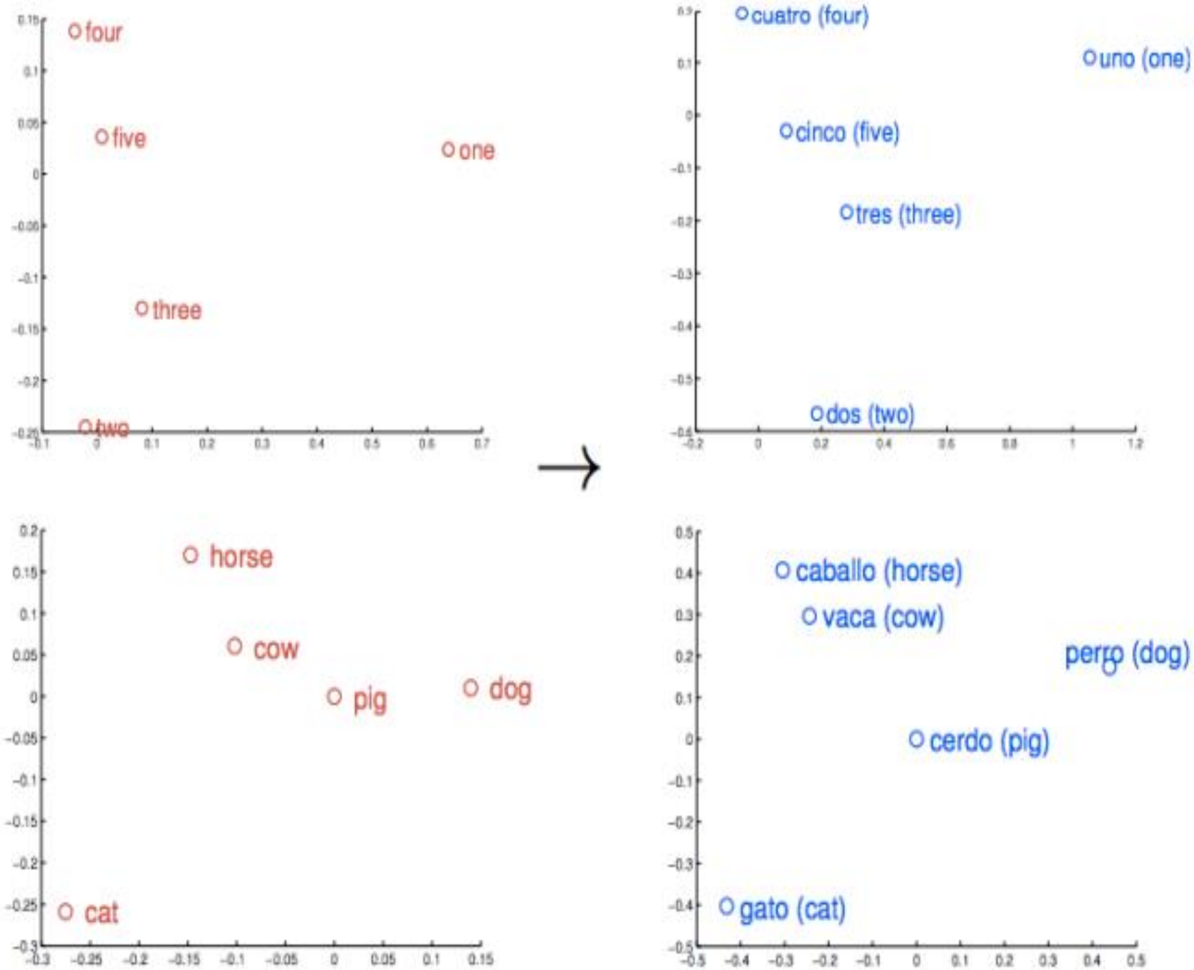


King - Man + Woman = Queen

Representation of Words

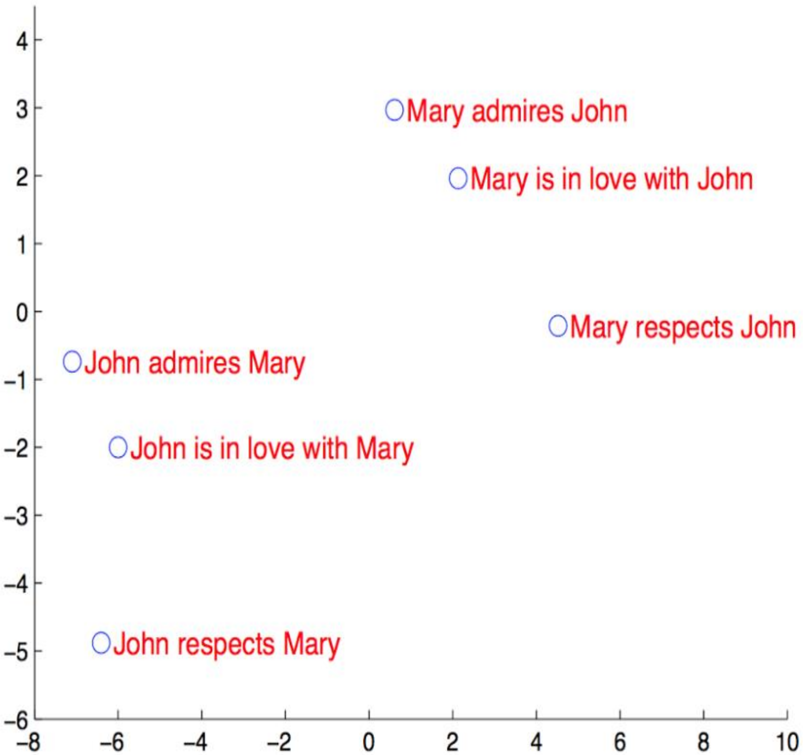


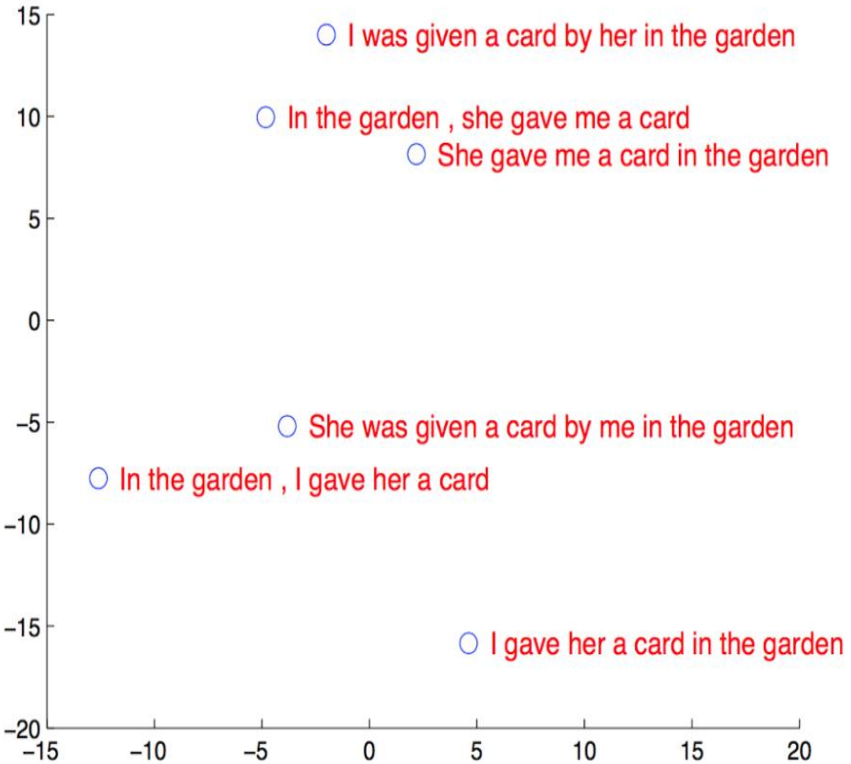
Representation of Words



Representation of Sentences

- Sentences can be represented as well as a dense vector space





How Powerful are Dense Representations?

- Google Neural Machine Translation system learned a *lingua franca*
- Similar sentences are mapped to the same area *independent of the language*
- Allows translating between unseen language pairs!
- Otherwise 10.000 bilingual corpora would be need for supporting 103 languages

Source: https://research.googleblog.com/2016/11/zero-shot-translation-with-googles.html

Embeddings & Deep Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- Word vectors form the basis of most deep learning approaches
- They provide basic knowledge about the meaning
- Neural Networks are able to propagate information into them
  - Linear models like naïve bayes / SVM cannot do that
- The quality of the embeddings has huge effect on the performance

- Quality of embeddings depends on:
  - Dataset (quality & quantity)
  - Pre-processing & cleaning of the dataset
  - Definition of the *context words*
  - Hyperparameters
- The algorithm (word2vec, GloVe) is often of minor importance