

Image: <https://developers.google.com>

# The World of Word Vectors

Michael Heck

Dialog Systems and Machine Learning

# INTRODUCTION

## Words as atomic units

- +: Intuitive, simple, efficient

Popular example: n-gram models

$$p(\text{„I live in New York“}) = p(\text{York|New}) * p(\text{New|in}) * p(\text{in|live}) * p(\text{live|I})$$

$$p(\text{„I live in New York“}) > p(\text{„I live in New Haven“})$$

- „Integerization“

$$p(\text{„2 156 38 56 528“}) = p(528|56) * p(56|38) * p(38|156) * p(156|2)$$

- -: Scalability issues, lack of similarity modeling
  - Similar properties can not be shared
  - Testing for word identity is often too strict

## Words as vectors

- 1-hot vectors (equivalent to integerization):

0	1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

„cat“

0	0	0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---

„dog“

- Designed feature vectors:

15	12	4	1	25	0	3	16	25	0
----	----	---	---	----	---	---	----	----	---

108	35	4	1	48	1	0	8	36	0
-----	----	---	---	----	---	---	---	----	---

Size, Weight, #Legs, Fur?, ...

- Learned feature vectors:

.05	-.3	.75	-.24	.08	.37	-.01	.54
-----	-----	-----	------	-----	-----	------	-----

.11	-.8	.34	-.23	-.01	.32	.21	.44
-----	-----	-----	------	------	-----	-----	-----

?, ? ,?, ...

## Distributed representations

- Connectionist perspective:

„Incorporating new knowledge about a concept affects the knowledge of other concepts that are represented by similar activity patterns.“ – Hinton (1986)

- Connectionism approach of the 70s and 80s

„Information processing takes place through the interactions of a large number of simple processing elements called units.“ – McClelland et al. (1986)



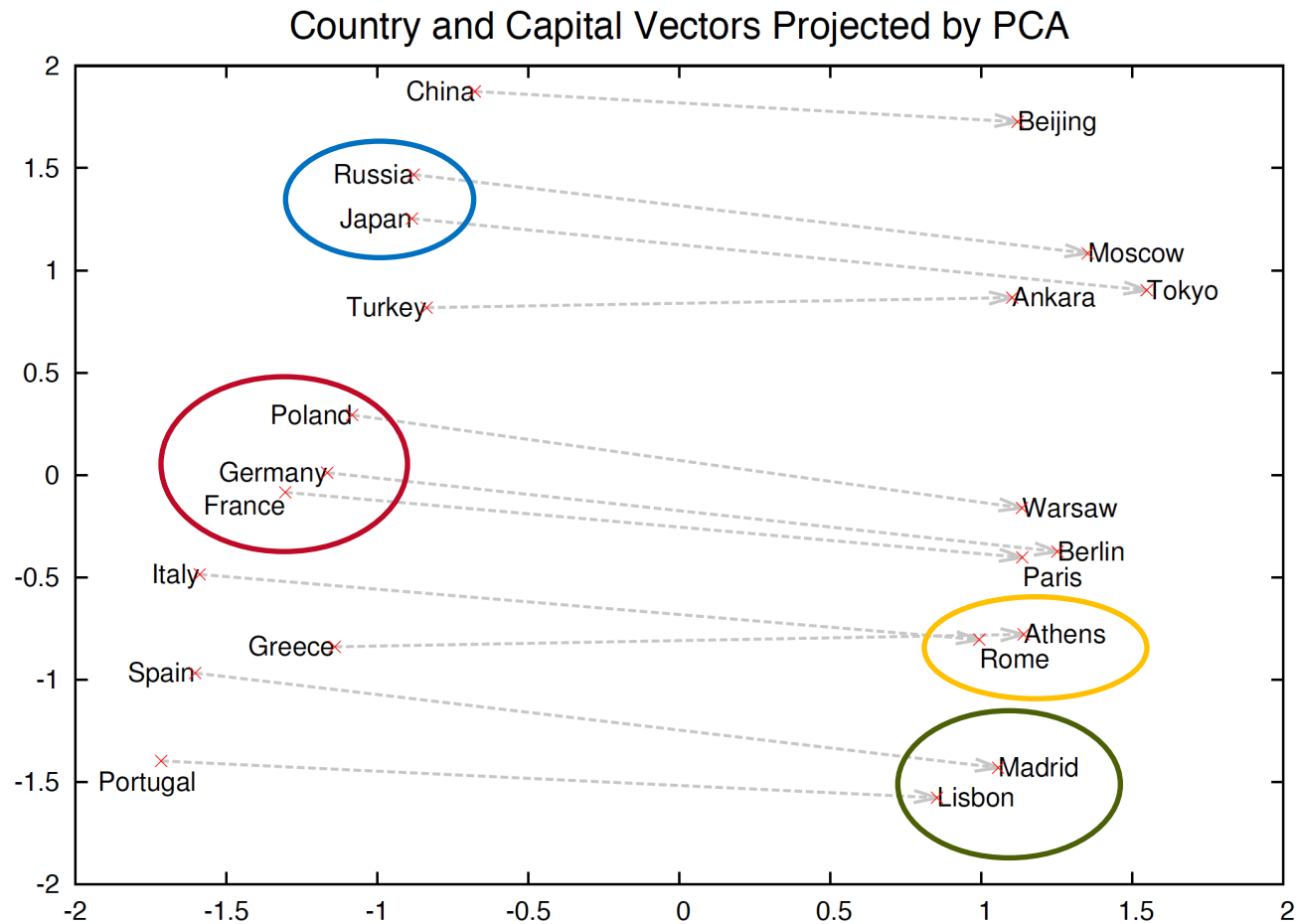
„artificial neural networks“

„distributed representations“



„Concepts can be represented by distributed patterns of activity in networks of neuron-like units.“ – Hinton (1986)

## Distributional word vectors



## Distributional hypothesis

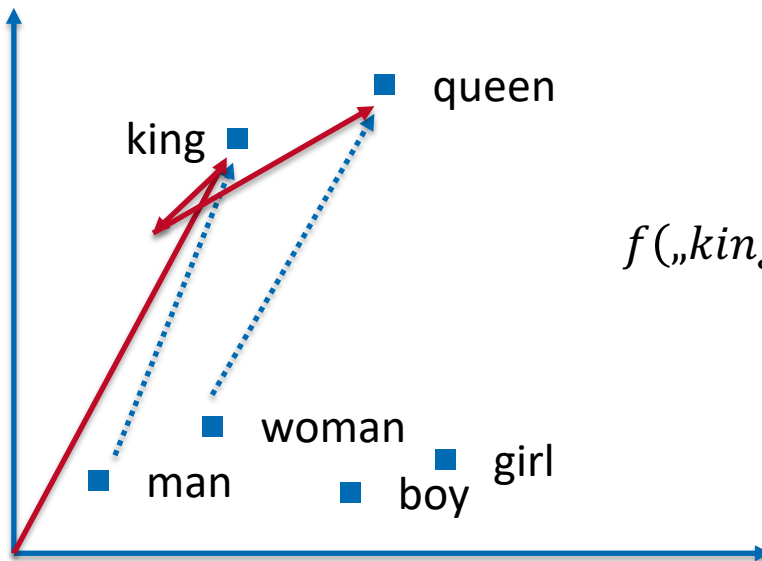
„Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache.“ („The meaning of words lie in their use.“) – Wittgenstein (1953, postum)

„All elements in a language can be grouped into classes whose relative occurrence can be stated exactly. However, for the occurrence of a particular member of one class relative to a particular member of another class it would be necessary to speak in terms of **probability**, based on the **frequency** of that **occurrence** in a **sample**.“ – Harris (1954)

- Provides basis for **statistical/distributional semantics**
  - Data driven statistical study of word meanings

## Distributional word vectors

- Real valued vectors that represent words
- Semantically related words have similar vectors
  - Powerful tool for knowledge representation and reasoning



$$f(\text{„king“}) - f(\text{„man“}) + f(\text{„woman“}) = f(\text{„queen“})$$



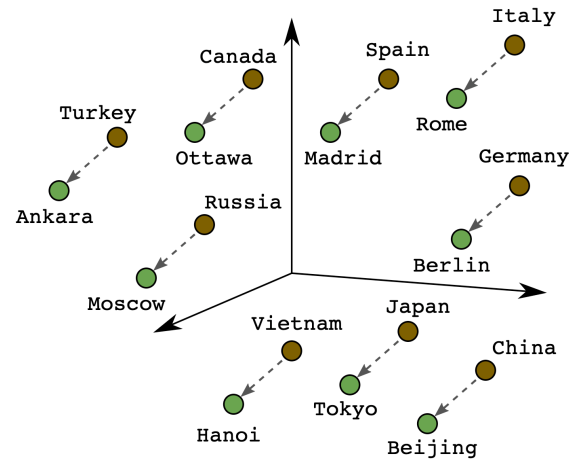
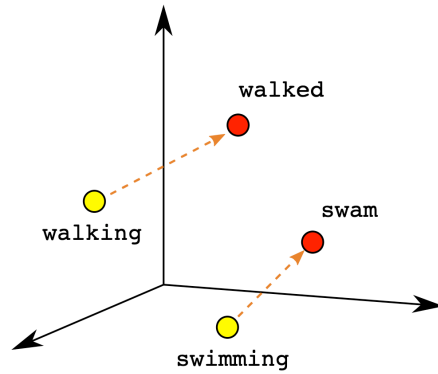
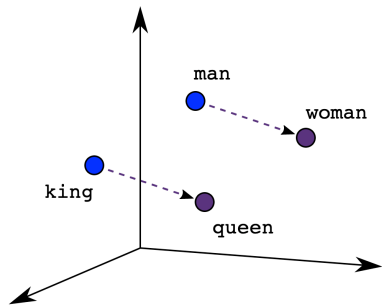


Image: <https://developers.google.com>

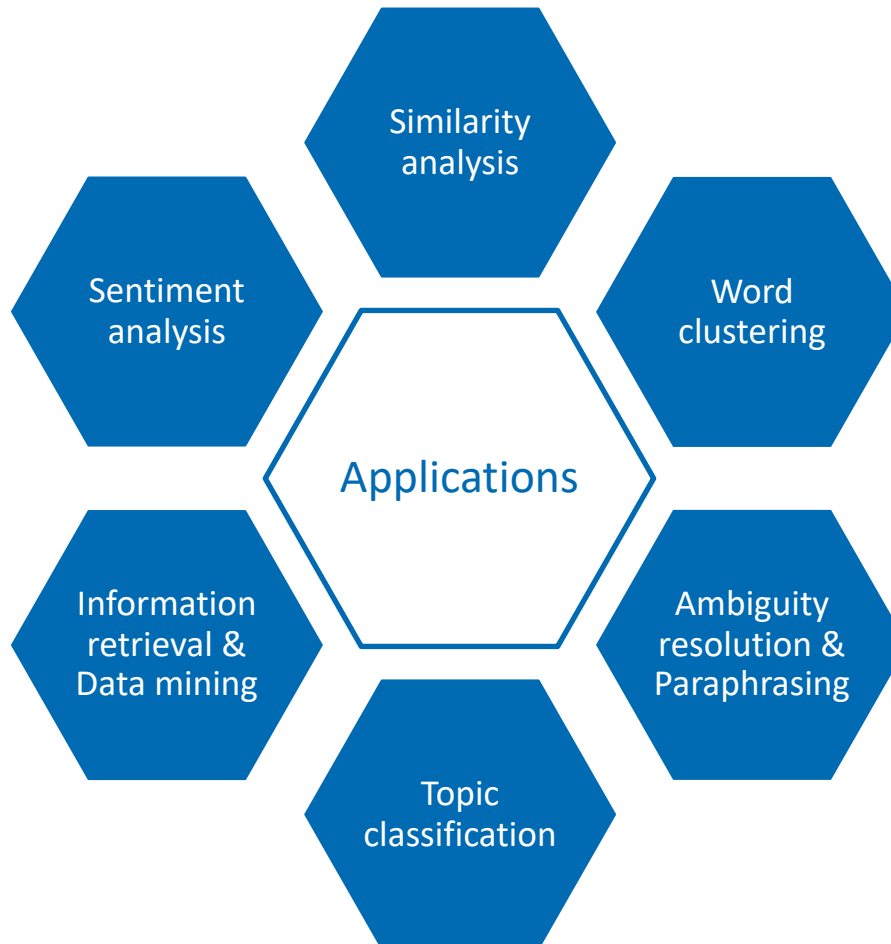
*Space*

# The World of Word Vectors

Michael Heck

Dialog Systems and Machine Learning

## Distributional semantics and its application



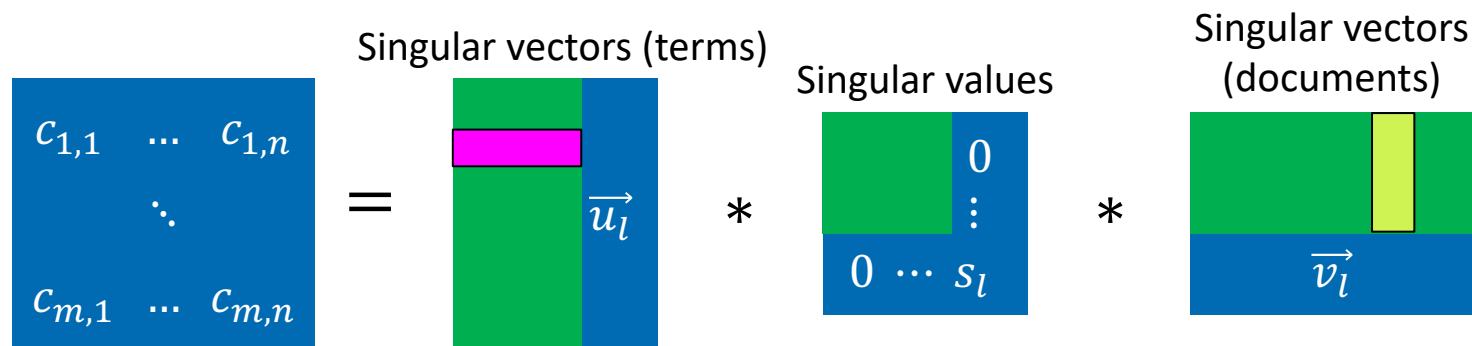
# WORD VECTORS & EMBEDDINGS

## Latent semantic analysis



### ■ Matrix factorization method

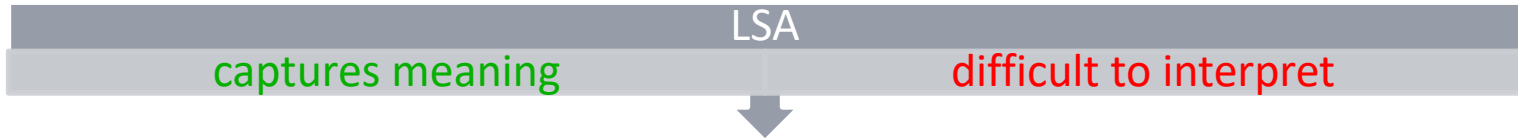
- Compute a term-document matrix  $A$

- Use singular value decomposition:  $A = U * S * V^T$



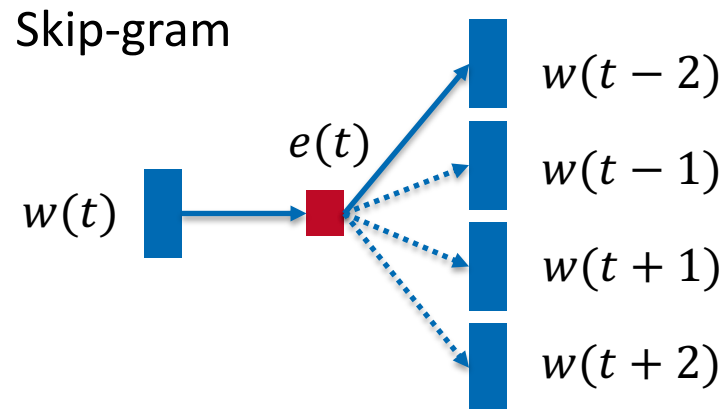
- Dimensional reduction by omitting singular values
- Term and document vectors can be treated as semantic spaces

 corresponds to a term  
 corresponds to a document



## word2vec

- Early popular **neural network based** method
  - Learns word vectors by learning to predict the context of words
    - Word **embeddings** are a by-product of solving a prediction task



$w(t-2)$	$w(t-1)$	$w(t)$	$w(t+1)$	$w(t+2)$	
a	refreshing	pool	of	cold	water

## Analogical reasoning task

- Perform operations with vectors to answer questions

$A - B + C = ?$  („What is to C in the same sense as B is to A?“)

- Word closest according to cosine distance is taken as answer

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
big - bigger	small: larger	cold: colder	quick: quicker
Copper - Cu	Zinc: Zn	Gold: Au	Uranium: Plutonium
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

*Table: Examples for 300 dimensional embeddings, trained on 783M words with skip-gram model.*

## Analogical reasoning task

- Perform operations with vectors to answer questions

$A - B + C = ?$  („What is to C in the same sense as B is to A?“)

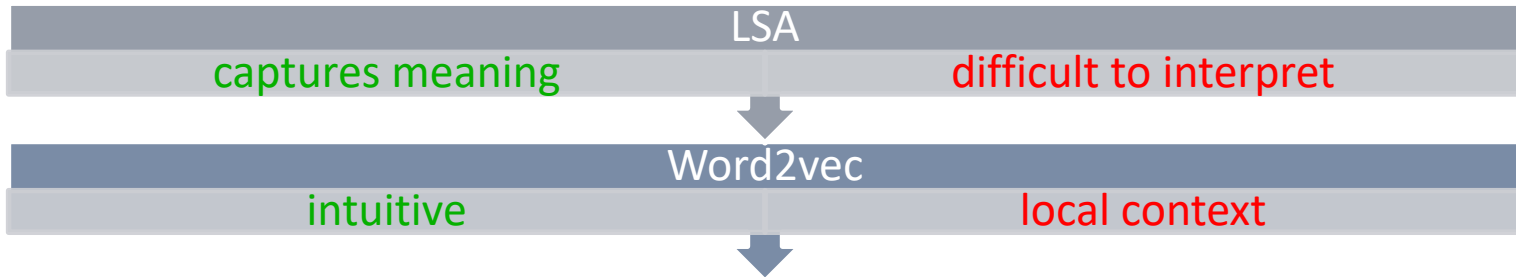
Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
big - bigger	small: larger	cold: colder	quick: quicker
Copper - Cu	Zinc: Zn	Gold: Au	Uranium: Plutonium
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Table: Examples for 300 dimensional embeddings, trained on 783M words with skip-gram model.

- Vector compositionality

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De





## GloVe („Global Vectors“)

- GloVe considers **global context**
  - Utilizes a co-occurrence matrix to capture global statistics
- Question: How is meaning generated from corpus statistics?
  - How might word vectors represent that meaning?
- Observation: word/word co-occurrence probability ratios have potential to encode meaning

Probability and Ratio	$k = \text{solid}$	$k = \text{gas}$	$k = \text{water}$	$k = \text{fashion}$
$P(k \text{ice})$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k \text{steam})$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k \text{ice})/P(k \text{steam})$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

Table: Co-occurrence probabilities for target words and context words.

## GloVe („Global Vectors“)

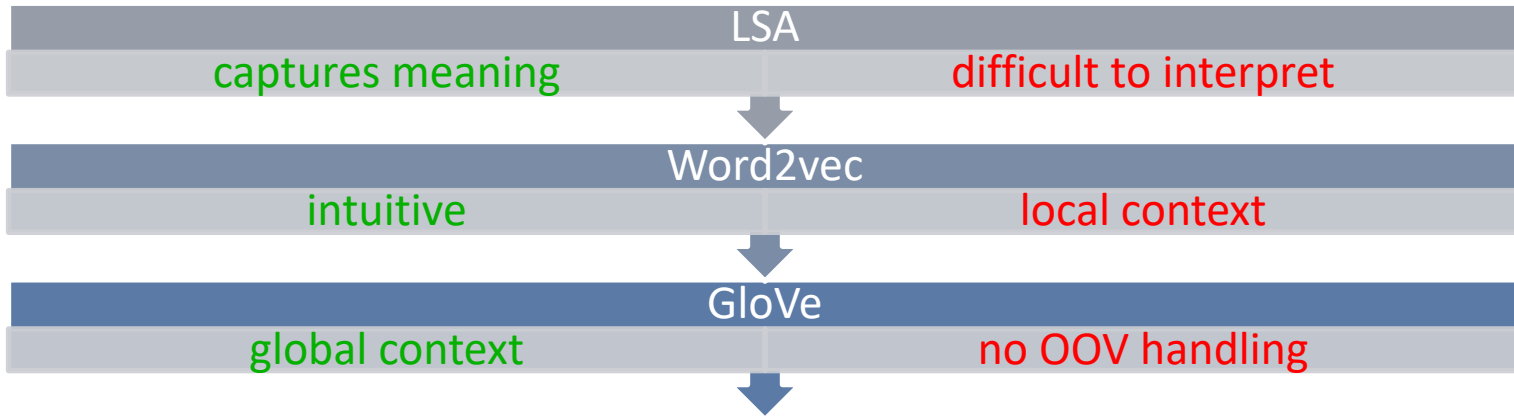
- Log-bilinear model with weighted least-squares objective:

$$J = \sum_{i,j=1}^V (w_i^T \tilde{w}_j - \log(P_{ij}))^2$$

product of word vector and  
context word vector

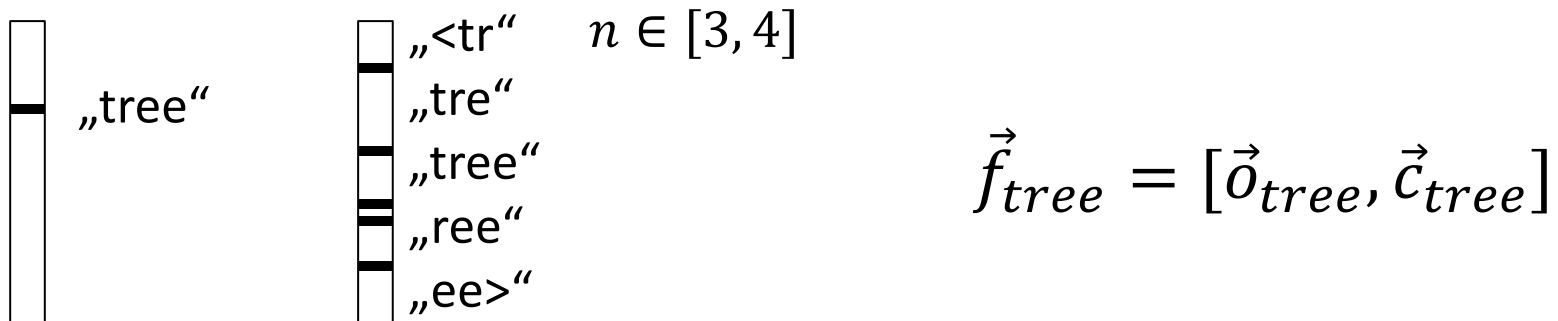
co-occurrence probabilities

- Goal: Learn word vectors such that their product equals the log of their co-occurrence probability
  - $J$  associates co-occurrence probability ratios with vector differences
  - Ratios encode meaning → **vector differences encode meaning**



## fastText


- Word2vec and GloVe consider „words“ as smallest unit
- fastText sees words as being composed of character  $n$ -grams



- Generates better embeddings for rare words
- Can construct vectors for unseen (OOV) words
- Hyperparameter choice is critical for performance

## fastText

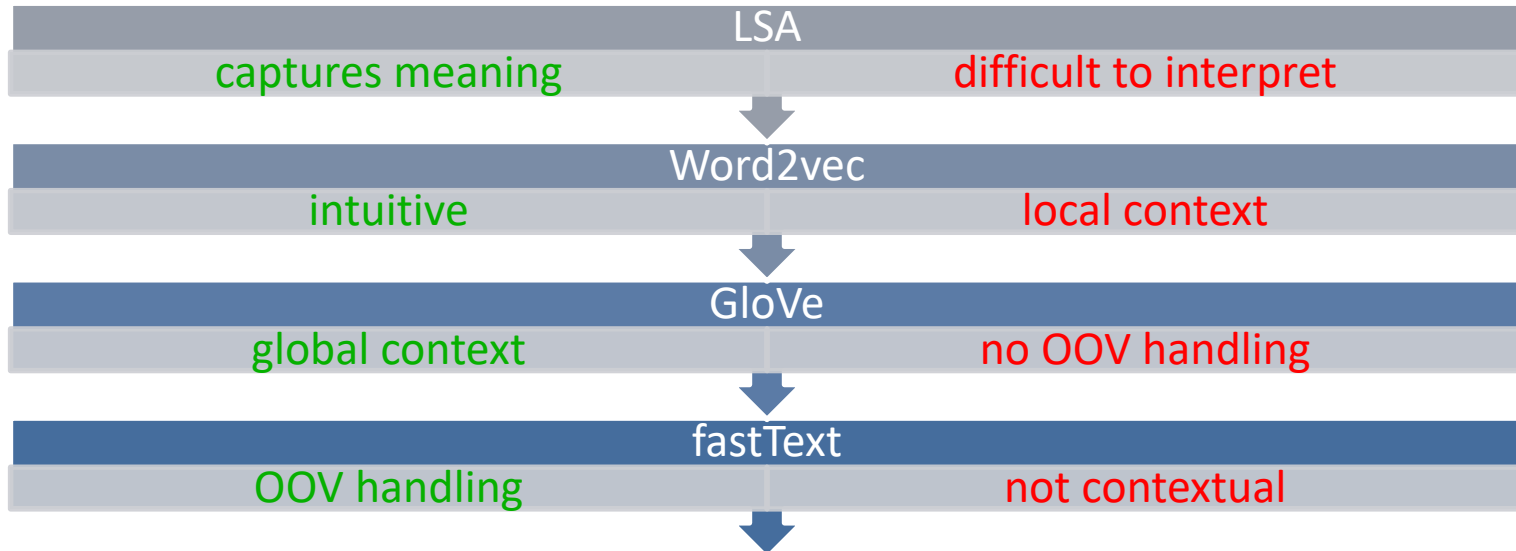
- Importance of subword information

word2vec:  typo

Query word? acomodation  
sunnhordland 0.775057  
acomodations 0.769206  
administrational 0.753011  
laponian 0.752274  
ammenities 0.750805  
dachas 0.75026  
vuosaari 0.74172  
hostelling 0.739995  
greenbelts 0.733975  
asserbo 0.732465

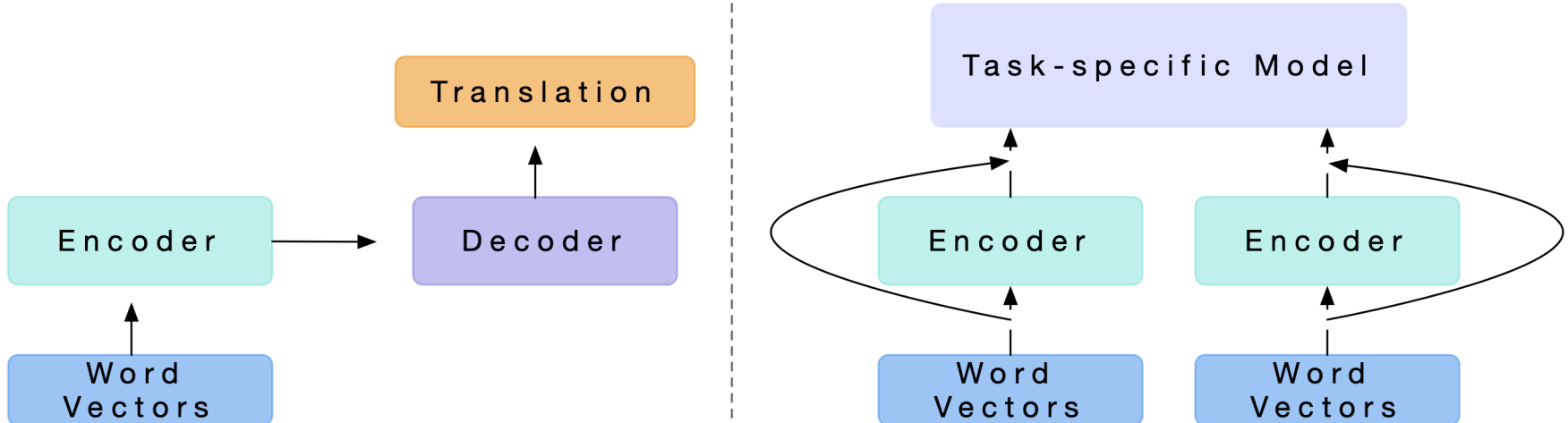
fastText:

Query word? acomodation  
acomodations 0.96342  
accommodation 0.942124  
accommodations 0.915427  
accommodative 0.847751  
accommodating 0.794353  
acomodated 0.740381  
amenities 0.729746  
catering 0.725975  
acomodate 0.703177  
hospitality 0.701426



## CoVe („Context Vectors“)

- Leverages MT to learn **contextualized** word vectors
  - Enables sense-specific representations for homographs
  - Assumes MT is general enough to capture „meanings“ of words



*Left: Training of encoder-decoder architecture for MT; Right: Utilize encoder to generate word vectors*



## CoVe („Context Vectors“)

- Sequence of context vectors produced by encoder:

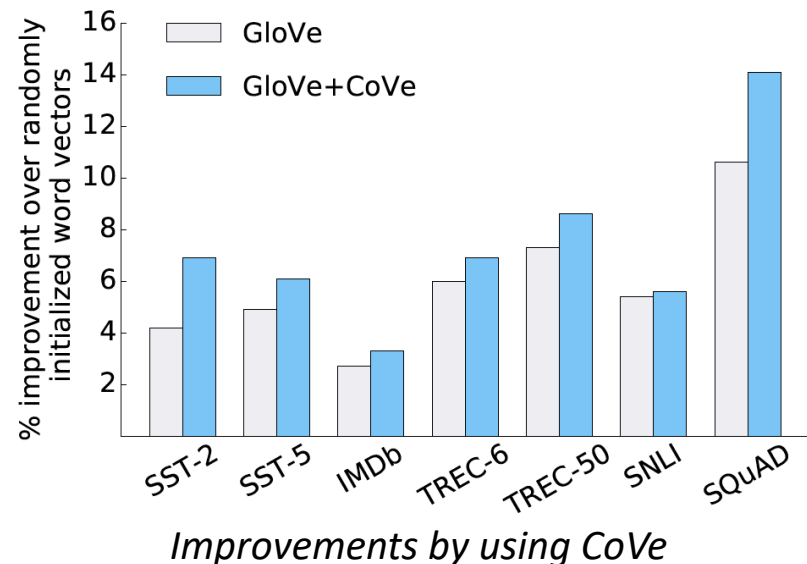
$$\text{CoVe}(w) = \text{MT-LSTM}(\text{GloVe}(w))$$

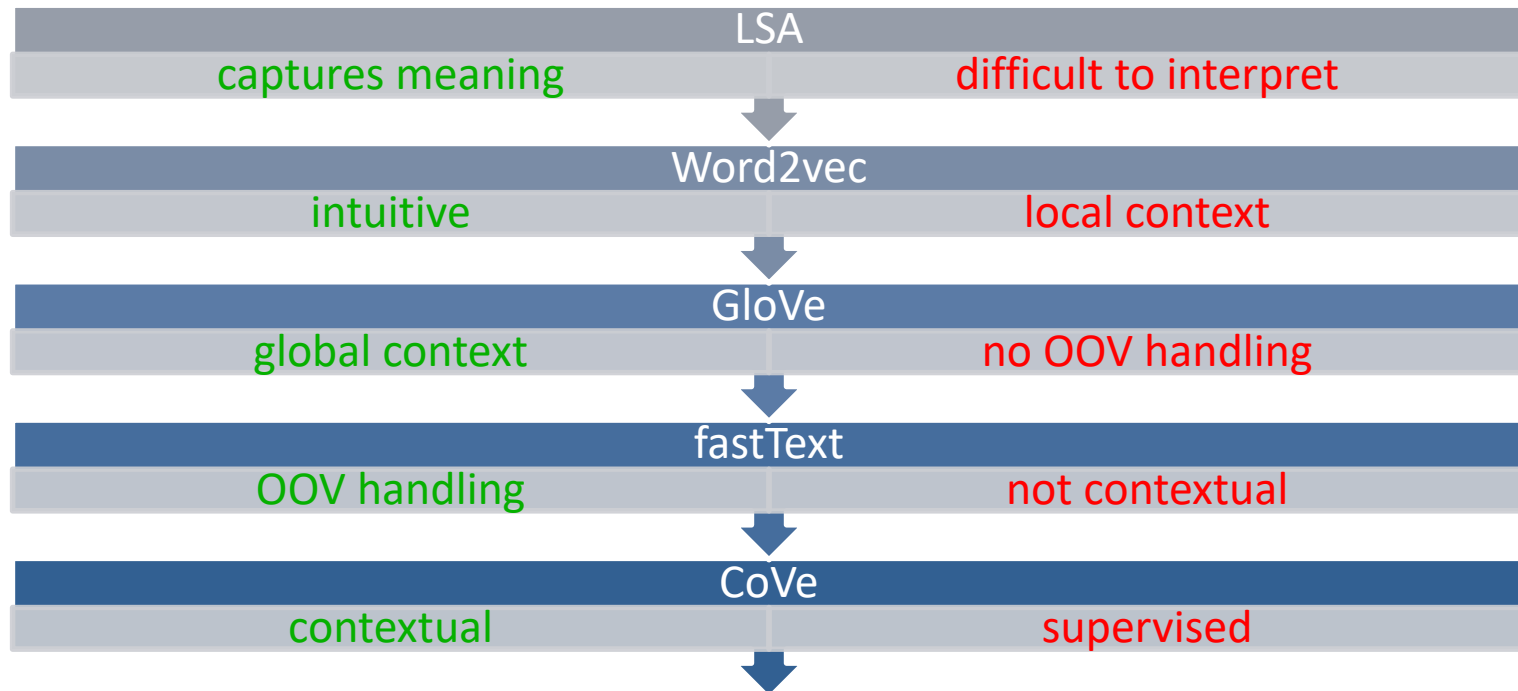
Sequence of words

Sequence of GloVeS

- Input for downstream tasks:

$$\tilde{w} = [\text{GloVe}(w); \text{CoVe}(w)]$$





## ELMo („Embeddings from Language Models“)

- Deep **contextualized** word representation
  - Learned function of internal states of a deep bidirectional LM
    - Each token representation is a function of the entire input sentence

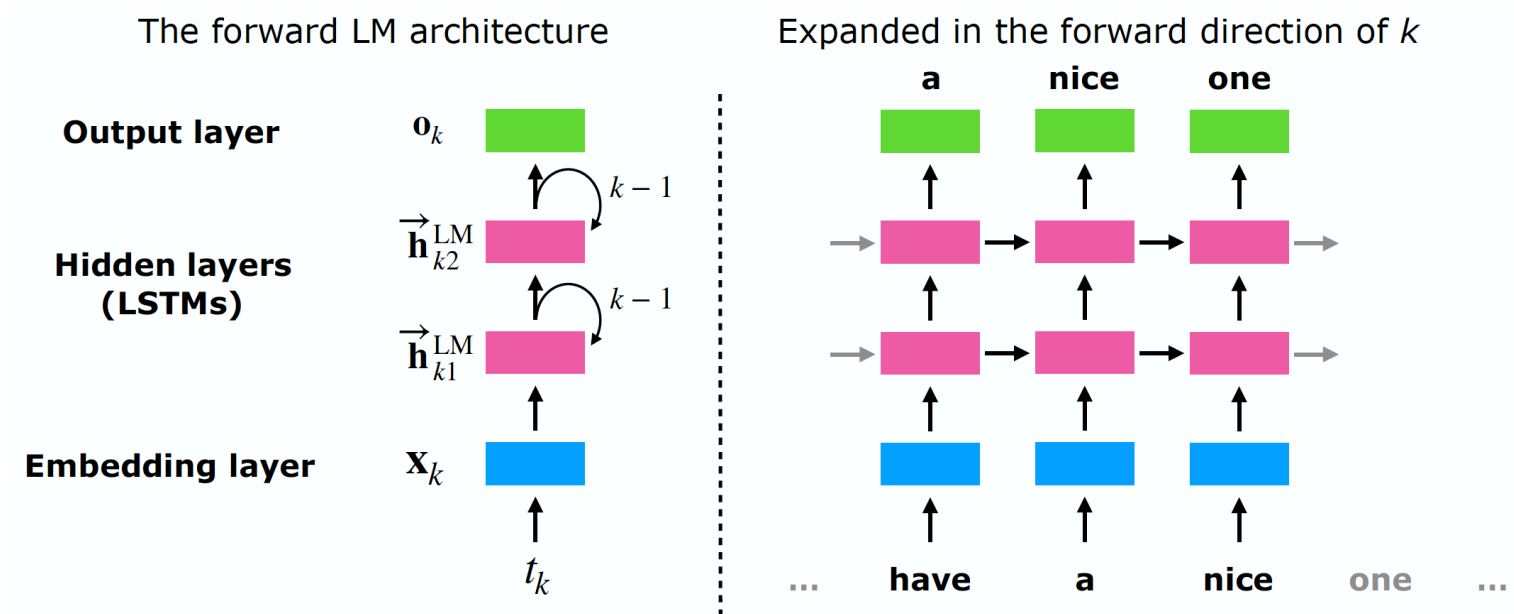


NLP running gag incoming...

Alammar, Jay (2018). <https://jalammar.github.io/illustrated-transformer/>

## ELMo („Embeddings from Language Models“)

- biLMs consist of a forward and a backward LM



- Forward:

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, t_2, \dots, t_{k-1})$$

- Backward:

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, t_{k+2}, \dots, t_N)$$

Yada, Shuntaro (2018). <https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018>

## ELMo („Embeddings from Language Models“)

- **Token** are represented by combinations of hidden layers
  - biLM parameters are fixed, weighting & scaling is learned
    - Higher layers seem to capture semantics, lower layers syntactics

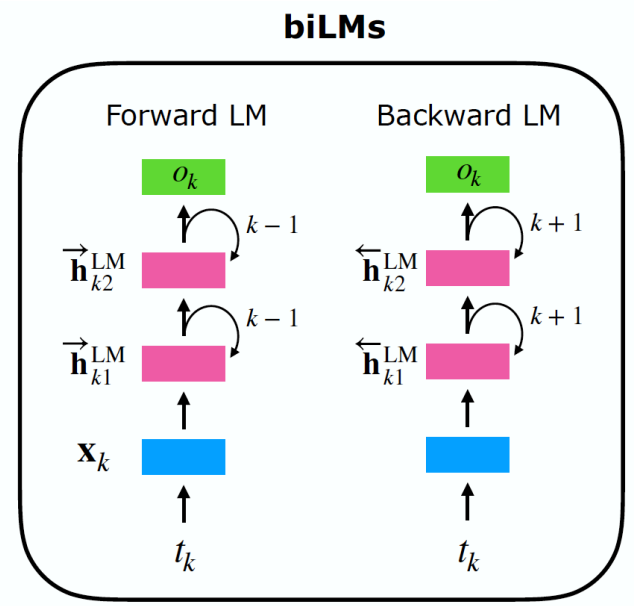
ELMo is a task specific representation. A down-stream task learns weighting parameters

$$\mathbf{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \times \sum \left\{ \begin{array}{l} s_2^{\text{task}} \times \mathbf{h}_{k2}^{\text{LM}} \\ s_1^{\text{task}} \times \mathbf{h}_{k1}^{\text{LM}} \\ s_0^{\text{task}} \times \mathbf{h}_{k0}^{\text{LM}} \end{array} \right. \left[ \vec{\mathbf{h}}_{kj}^{\text{LM}}; \overleftarrow{\mathbf{h}}_{kj}^{\text{LM}} \right]$$

Concatenate hidden layers

( $\mathbf{x}_k; \mathbf{x}_k$ )

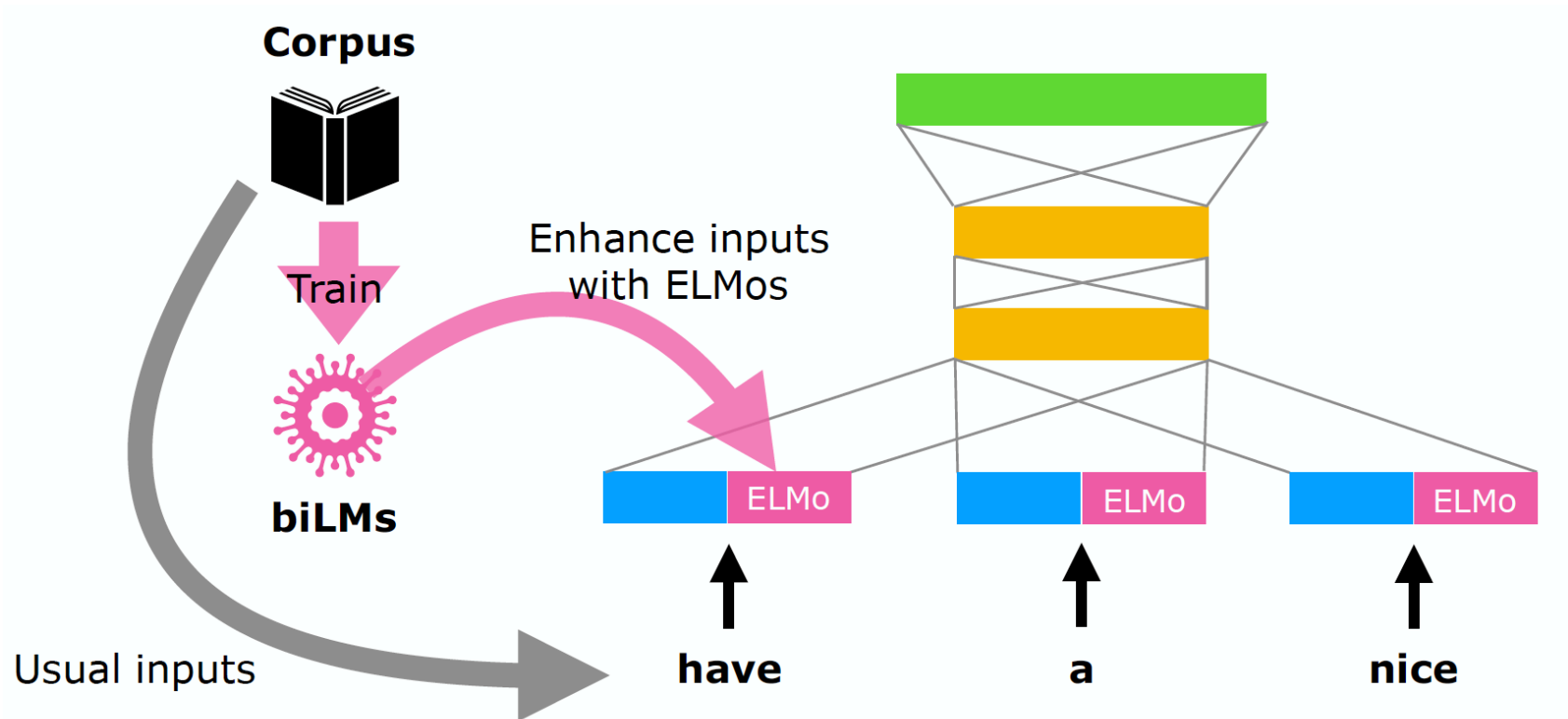
Unlike usual word embeddings, ELMo is assigned to every *token* instead of a *type*



Yada, Shuntaro (2018). <https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018>

## ELMo („Embeddings from Language Models“)

- ELMo vectors are used as **additional** features in NLP tasks



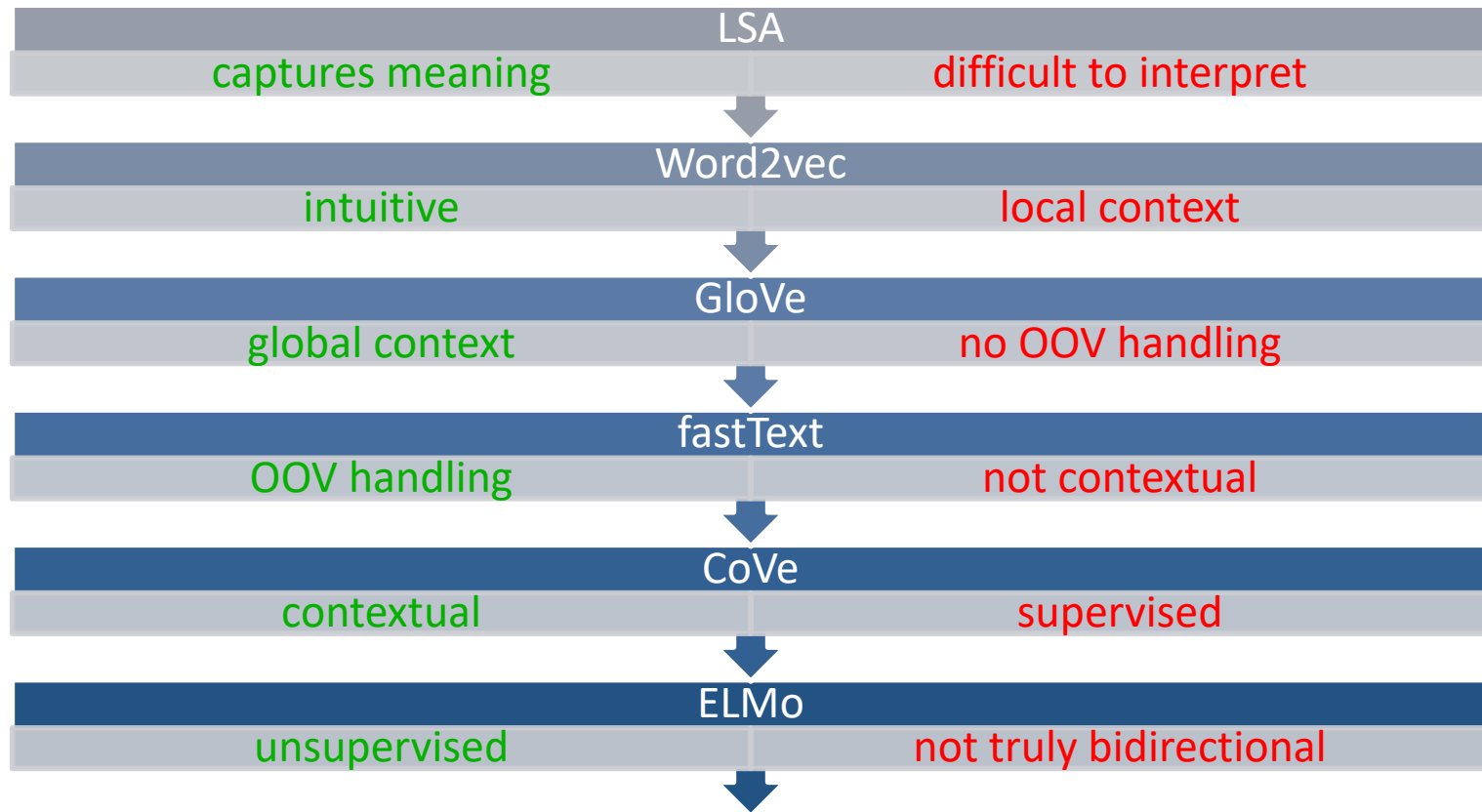
Yada, Shuntaro (2018). <https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018>

## ELMo („Embeddings from Language Models“)

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular <u>play</u> on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent <u>play</u> .
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}	{...} they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement .

*Table: Nearest neighbors to the token „play“ using GloVe or ELMo.*

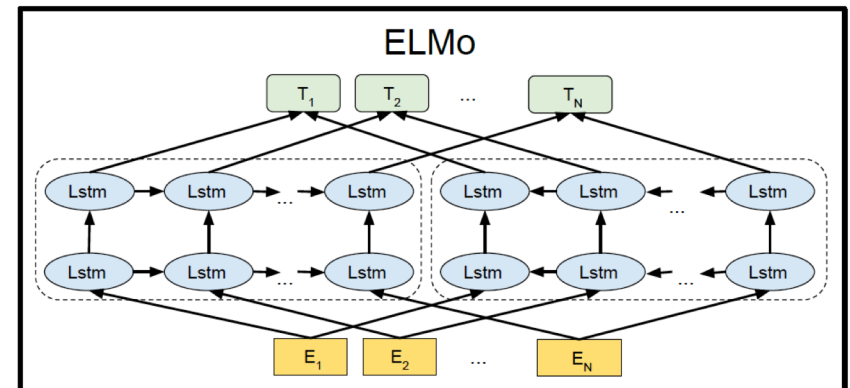
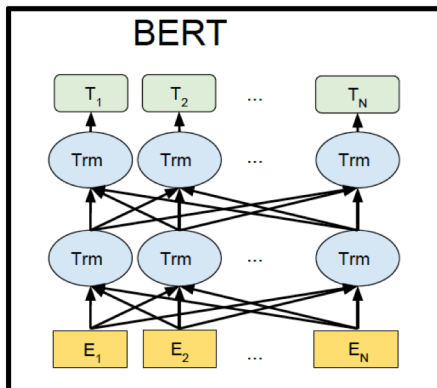
- Unsupervised learning problem (no labels needed)
- Out-of-vocabulary words can be accepted as input
- Utilizes various levels of deep knowledge





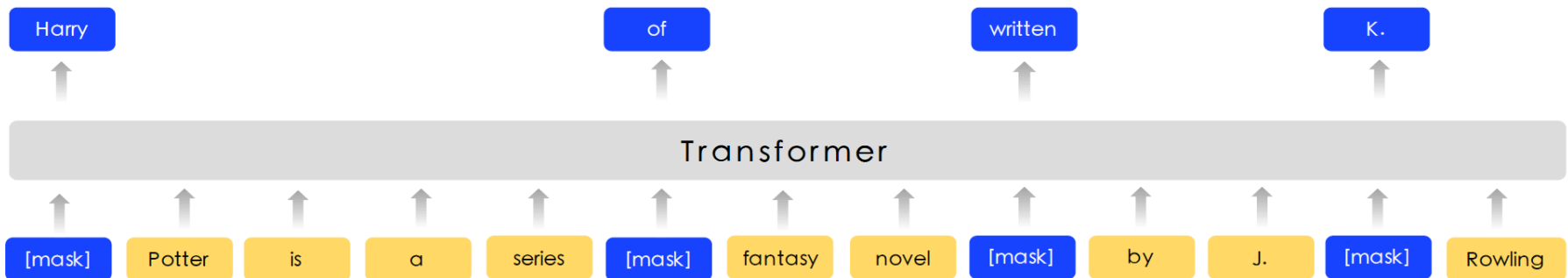
## BERT („Bi-directional Encoder Representations from Transformers“)

- Pre-trains deep bi-directional representations
  - **Jointly** conditioned on left **and** right context in all layers
  - Fine-tune additional output layer to solve specific NLP tasks



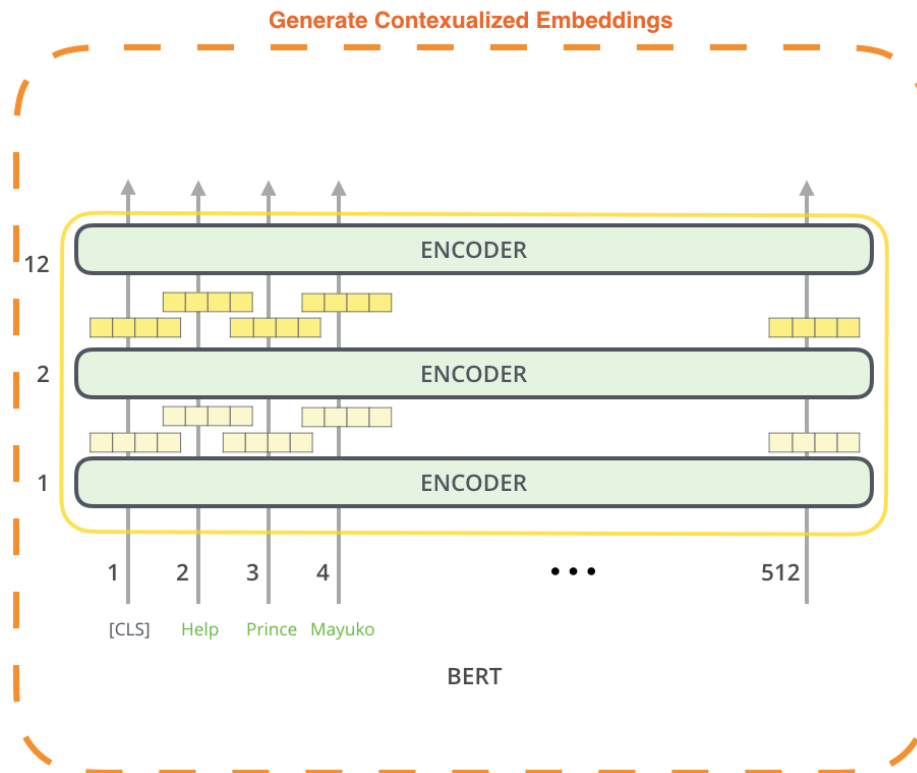
## BERT („Bi-directional Encoder Representations from Transformers“)

- Objective: „Masked language modeling“
  - Words are randomly masked (probability 15%)
  - Masked words are predicted given their context
  - Process sentence pairs to tackle various NLP tasks

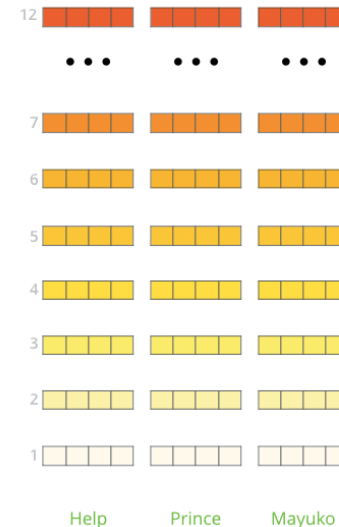


## BERT („Bi-directional Encoder Representations from Transformers“)

- Can also be used to create „ELMo-like“ word embeddings

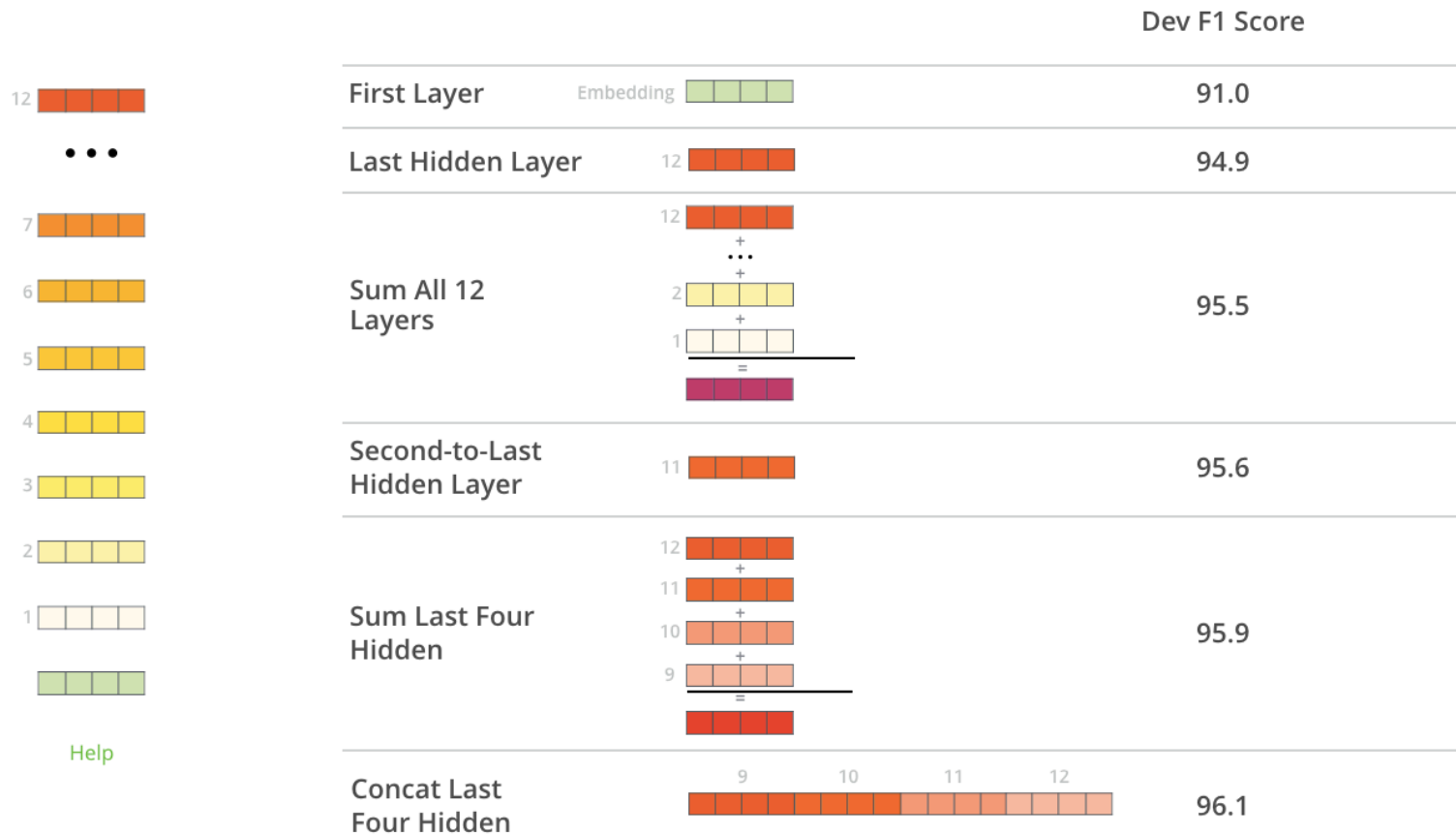


The output of each encoder layer along each token's path can be used as a feature representing that token.

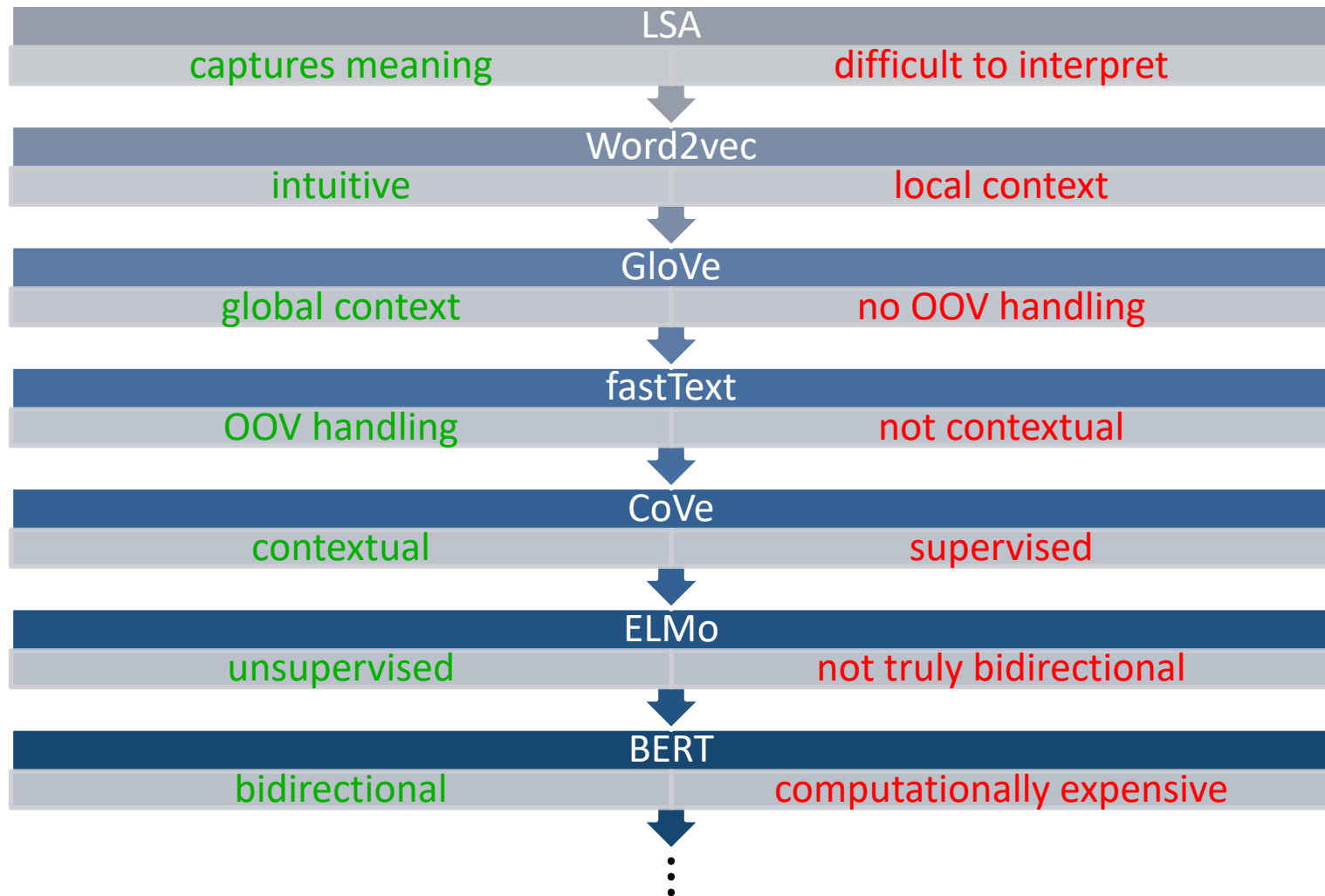


## BERT („Bi-directional Encoder Representations from Transformers“)

- Optimal representation is task-dependent



Alammar, Jay (2018). <https://jalammar.github.io/illustrated-transformer/>



## Other transformer-based models

- ERNIE („Enhanced Representation through kNowledge IntEgration“)
  - Learns to fill in phrases (token groups)
    - Demonstrably works well for Chinese
- KERMIT („Kontextuell Encoder Representations Made by Insertion Transformations“)
  - Learns to insert token into incomplete sequences
    - Instead of masking the input, token are removed completely
  - Generates text in arbitrary order
    - Applicable to sentence completion, translation, etc.

## Complex word relations

- Context can not convey all possibly relevant word relations

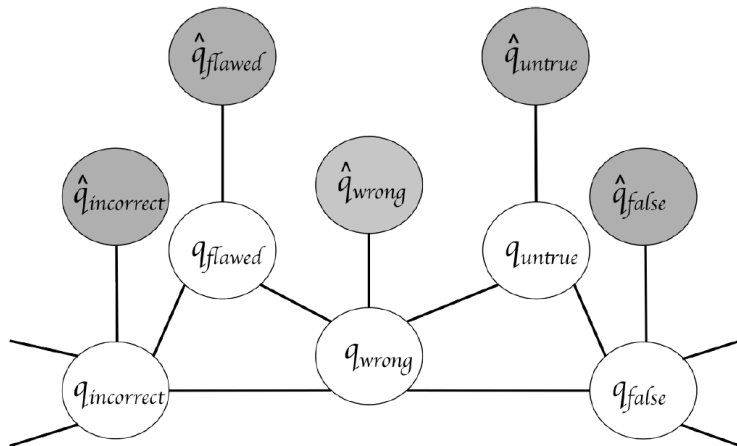
Example: Antonymy

„cheap“ and „expensive“ share most contexts, but their meanings differ greatly

- Retrofitting & counter-fitting
  - Refine vector spaces using relational information
  - Post-processing steps (independent of vector types)
  - Task-dependent

## Retrofitting

- Moves related words (according to ontology) closer together
  - Related words: synonyms, hypernyms, hyponyms



Word graph with vectors to be retrofitted.

- Objective:

$$\Psi(Q) = \sum_{i=1}^n \left[ \alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j) \in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

keep old and new vectors  
as close as possible

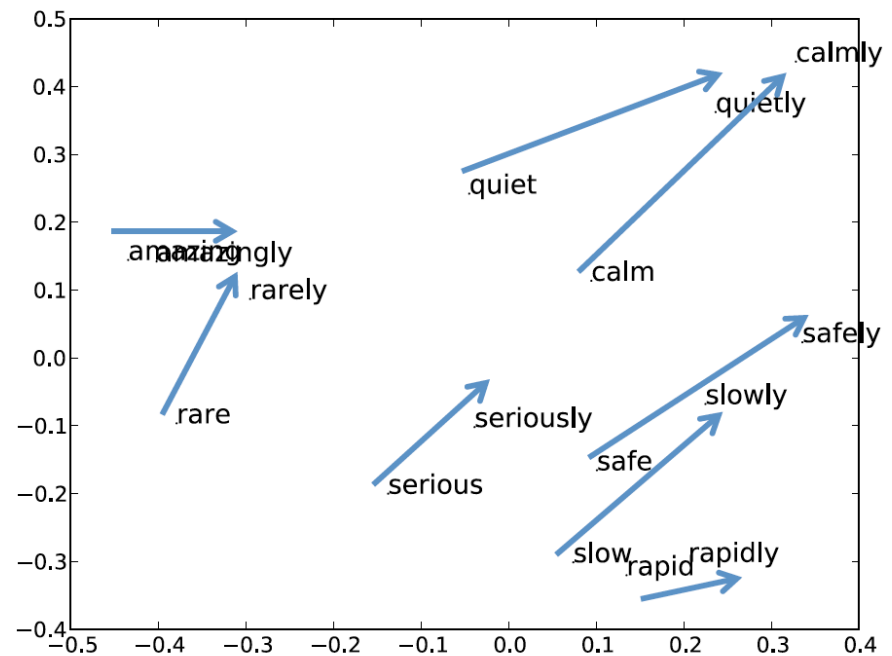
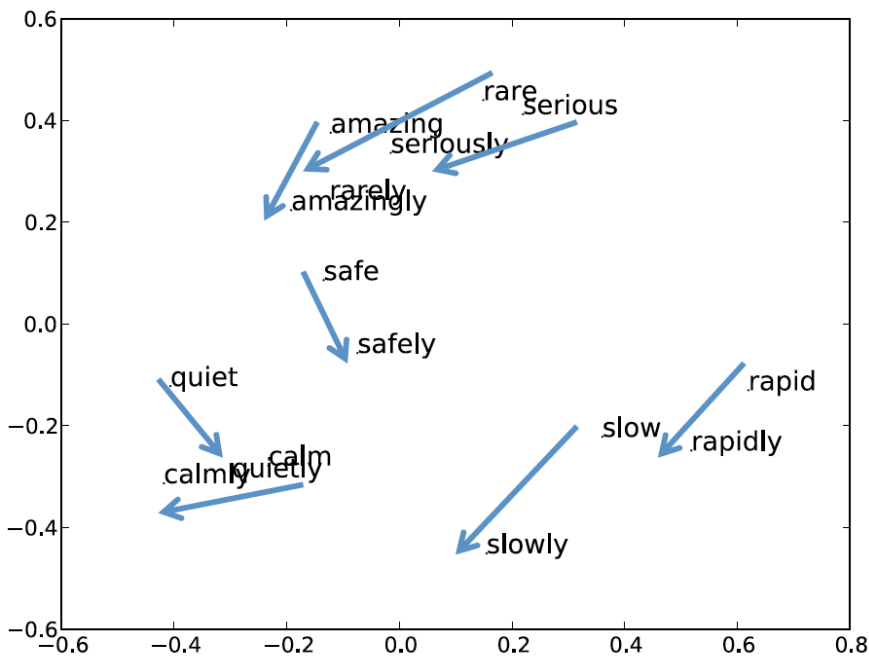
keep vectors of related words  
as close as possible

- Competitive with joint optimization using „semantic prior“
  - Constraints among words are used as regularization term



## Retrofitting

### ■ Visualization



2D PCA projections of 100-dimensional skip-gram word vectors of words with adjective/adverb relations.

Left: **before** retrofitting; right: **after** retrofitting.

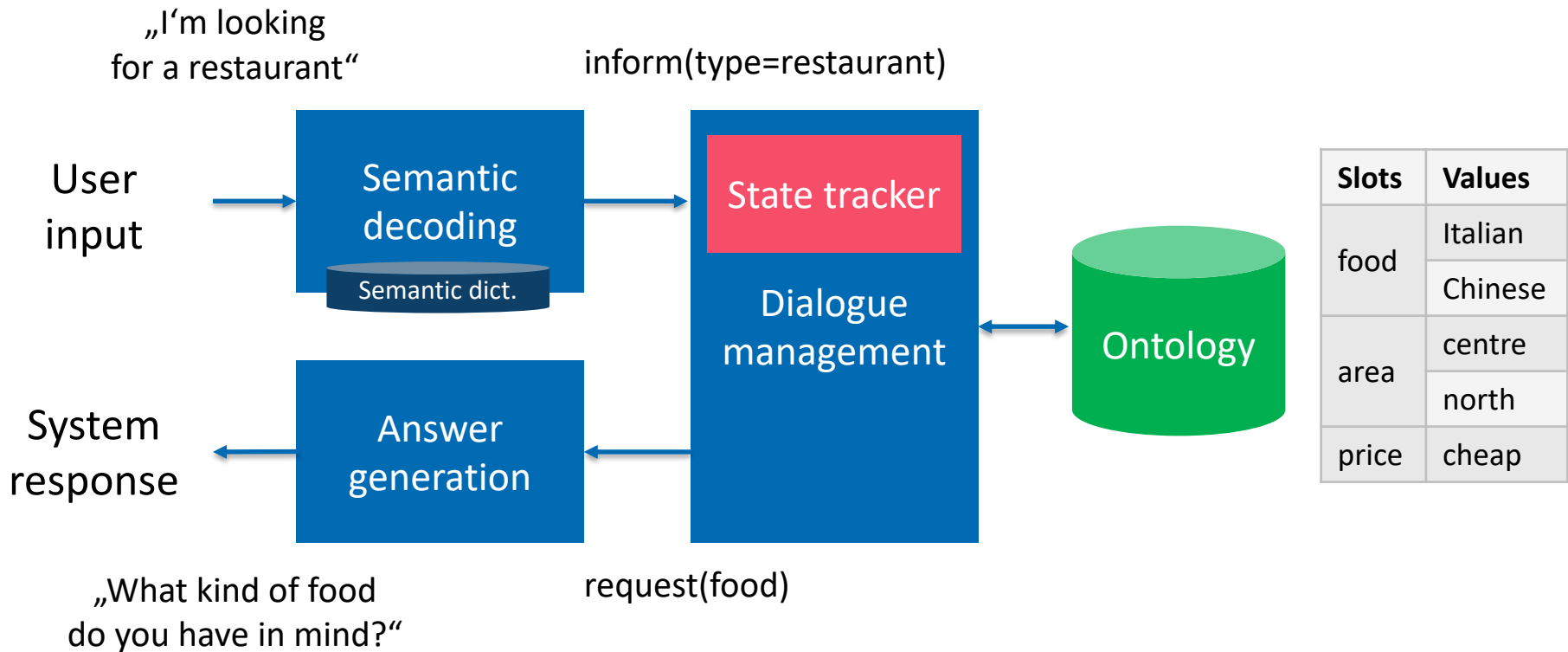
## Counter-fitting

- Injects antonymy/synonymy constraints into vector spaces
  - Objective analogous to retrofitting

	<b>east</b>	<b>expensive</b>	<b>British</b>
Before	west	pricey	American
	north	cheaper	Australian
	south	costly	Britain
	southeast	overpriced	European
	northeast	inexpensive	England
After	eastward	costly	Brits
	eastern	pricy	London
	easterly	overpriced	BBC
	-	pricey	UK
	-	afford	Britain

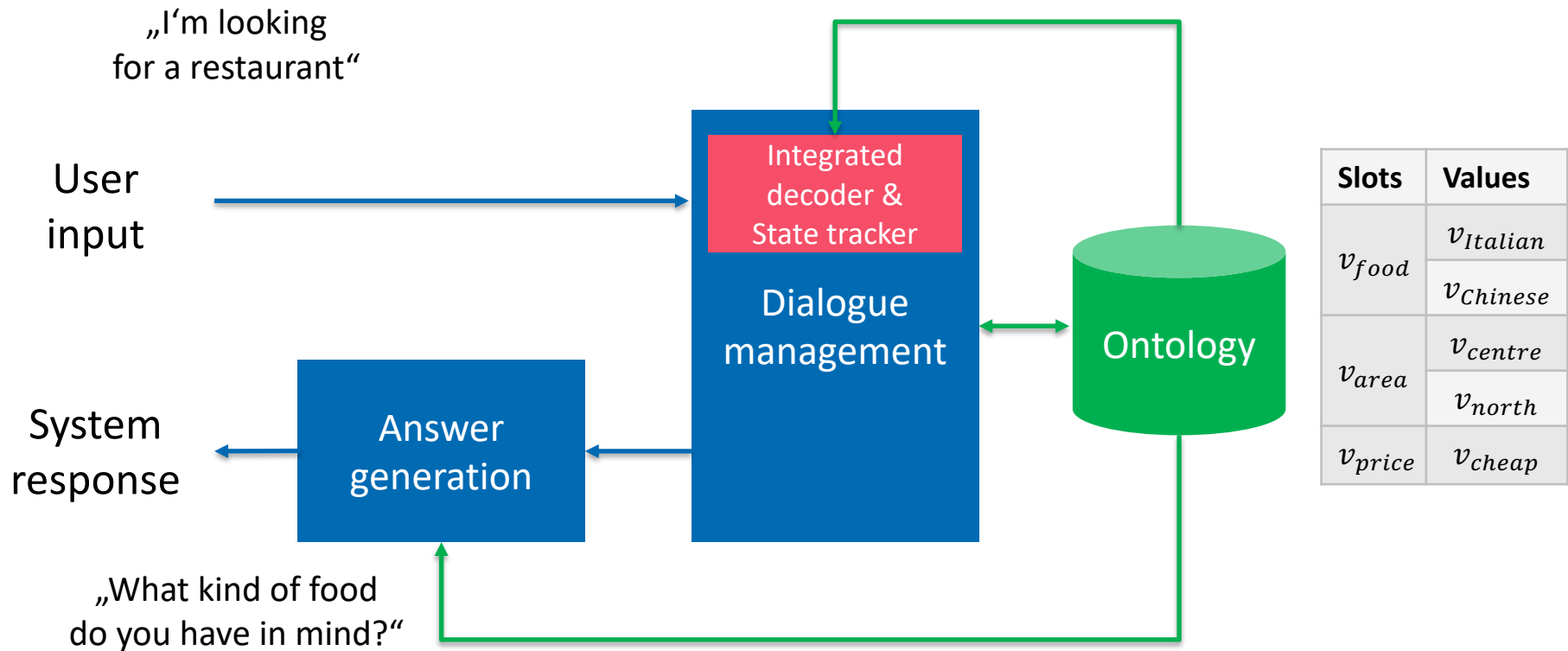
*Nearest neighbors of GloVe-embedded words before and after counter-fitting.*

## Statistical dialogue systems 101



- Discrete representation of concepts limits capacities

## Word vectors in statistical dialogue systems



- Word vectors mitigate semantic decoding problem
  - Similarity measures replace exact matching

## Word vectors in statistical dialogue systems

- Ontology is the core of all DS components
- Designed ontologies are inflexible
  - Hard to build, hard to expand, never complete
- Distributed representations help deal with complex ontologies
  - Dynamic adjustment (growth, hierarchy, etc.)
  - Tighter integration of DS components
  - Motivates truly statistical solutions instead of heuristics

## Today & the future

- Sensible substructure in vector space
  - Consistent relations between token and token classes
- Preservation of complex semantic relations
  - Antonymy/synonymy, hypernymy/hyponymy, meronymy, etc.
- Free of semantic biases
  - Stereotypes, historical, gender, racial, etc.
- Robustness and flexibility
  - Rare words, unseen words, new concepts

## Select references

- Bojanowski, 2017, Enriching Word Vectors with Subword Information
- Chan, 2019, KERMIT - Generative Insertion-Based Modeling for Sequences
- Devlin, 2018, BERT - Pre-training of Deep Bidirectional Transformers for Language Understanding
- Faruqui, 2015, Retrofitting Word Vectors to Semantic Lexicons
- Harris, 1954, Distributional Structure
- Hinton, 1986, Learning Distributed Representations of Concepts
- McCann, 2018, Learned in Translation - Contextualized Word Vectors
- McClelland, 1986, The Appeal of Parallel Distributed Processing
- Mikolov, 2013, Distributed Representations of Words and Phrases
- Mikolov, 2013, Efficient Estimation of Word Representations in Vector Space
- Mrksic, 2016, Counter-fitting Word Vectors to Linguistic Constraints
- Mrksic, 2017, Neural Belief Tracker - Data-Driven Dialogue State Tracking
- Pennington, 2014, GloVe - Global Vectors for Word Representation
- Peters, 2018, Deep Contextualized Word Representations
- Smith, 2019, Contextual Word Representations - A Contextual Introduction
- Sun, 2019, ERNIE - Enhanced Representation through Knowledge Integration
- Wittgenstein, 1953, Philosophische Untersuchungen