《多模态机器学习》

第四章 文本模态与递归神经网络

黄文炳
中国人民大学高瓴人工智能学院
hwenbing@126.com
2024年秋季

内容提纲

① 词表示

② 递归神经网络

③ 语言模型

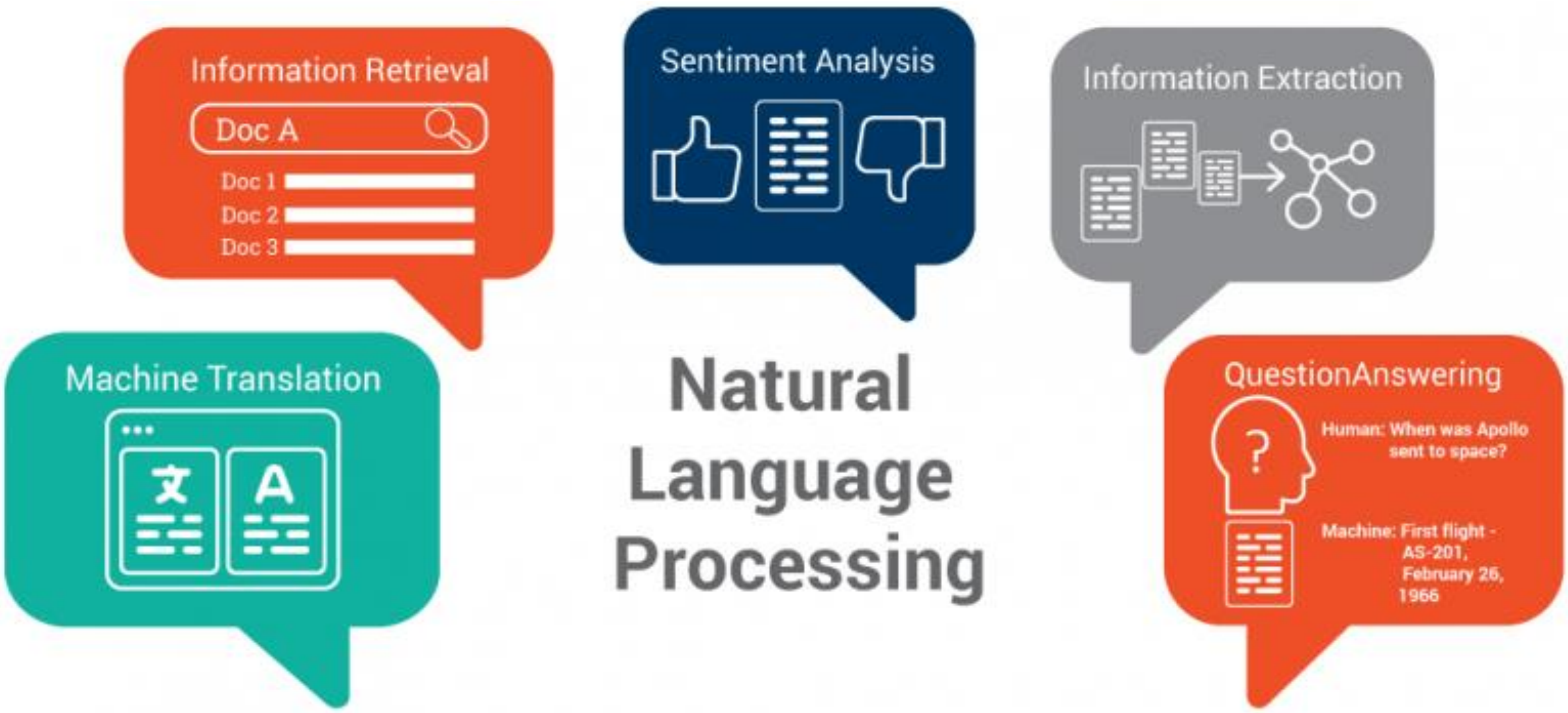
内容提纲

① 词表示

② 递归神经网络

③ 语言模型

Natural Language Processing Tasks



ChatGPT

Default (GPT-3.5)

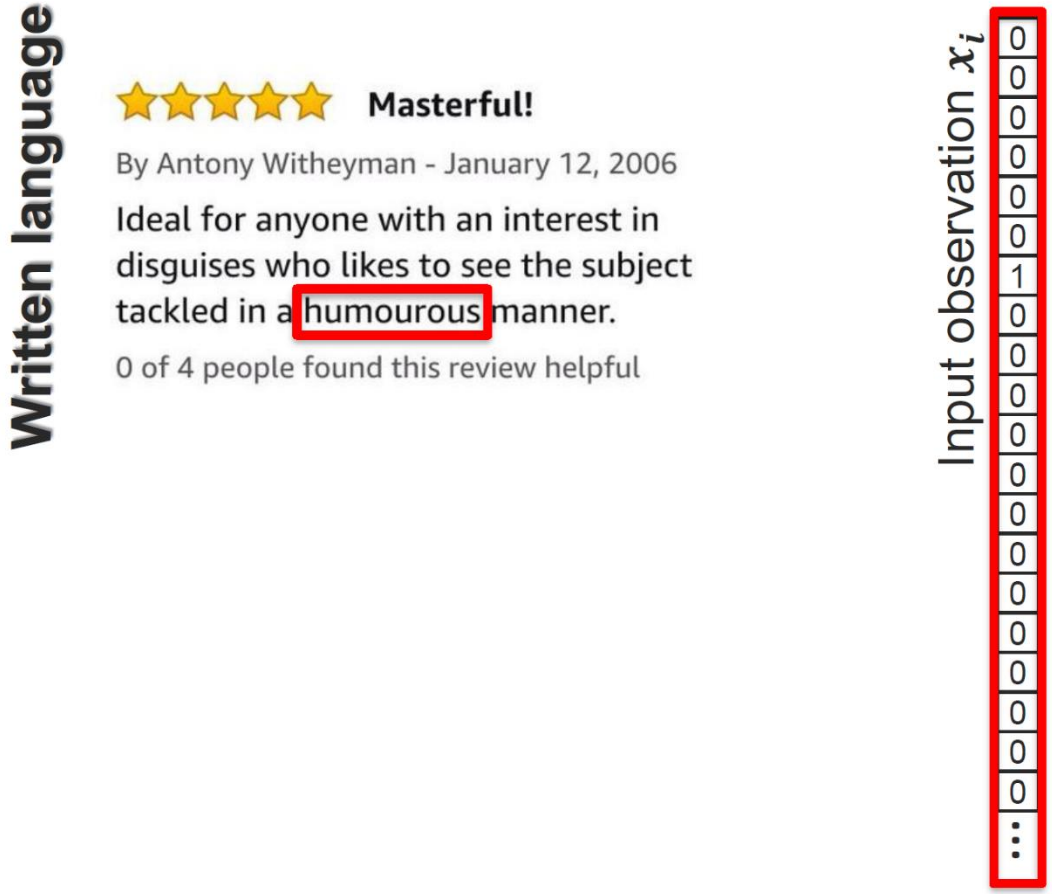
W

Please tell me what chatgpt can't do in natural language processing

ChatGPT, like other natural language processing models, has its limitations. Here are some of the things that ChatGPT may struggle with or cannot do effectively:

1. **Generating False or Misleading Information:** ChatGPT can generate text based on the patterns it has learned from its training data, which means it can inadvertently produce false or misleading information, especially when asked about topics with controversial or disputed facts.
2. **Understanding Context Over Long Conversations:** While ChatGPT can maintain context over a short conversation, it may lose track of the topic or context in longer interactions, leading to responses that don't make sense or are off-topic.
3. **Lack of Common Sense Reasoning:** ChatGPT may struggle with common sense reasoning. It might provide answers that are logically correct but lack real-world common sense. For

Simple Word Representation



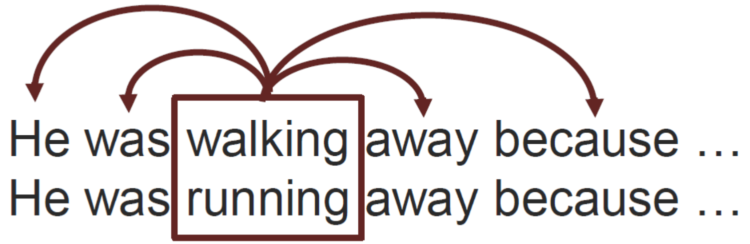
"one-hot" vector

$|x_i|$ = number of words in dictionary

How to learn (word) features/representations?

- **Distribution hypothesis:** Approximate the word meaning by its surrounding words

- Words used in a similar context will lie close together



- **Instead of capturing co-occurrence counts directly, predict surrounding words of every word**

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

Geometric Interpretation

统计dog这个词和其他词一起出现在同一句句子中的次数。这样一来就获得了一个dog的R^n的表示（假设词典的大小是n）

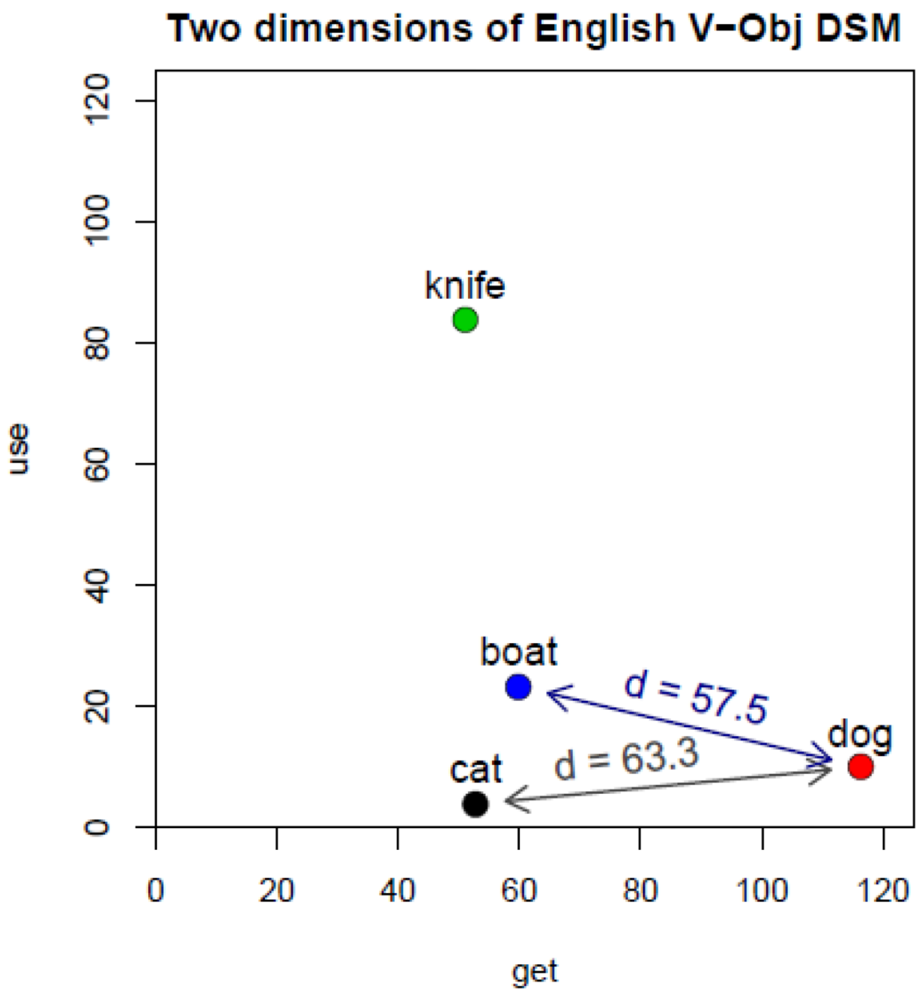
- row vector $\mathbf{x}_{dog}$ describes usage of word *dog* in the corpus
- can be seen as coordinates of point in *n*-dimensional Euclidean space R^n

	get	see	use	hear	eat	kill
knife	51	20	84	0	3	0
cat	52	58	4	4	6	26
dog	115	83	10	42	33	17
boat	59	39	23	4	0	0
cup	98	14	6	2	1	0
pig	12	17	3	2	9	27
banana	11	2	2	0	18	0

co-occurrence matrix M

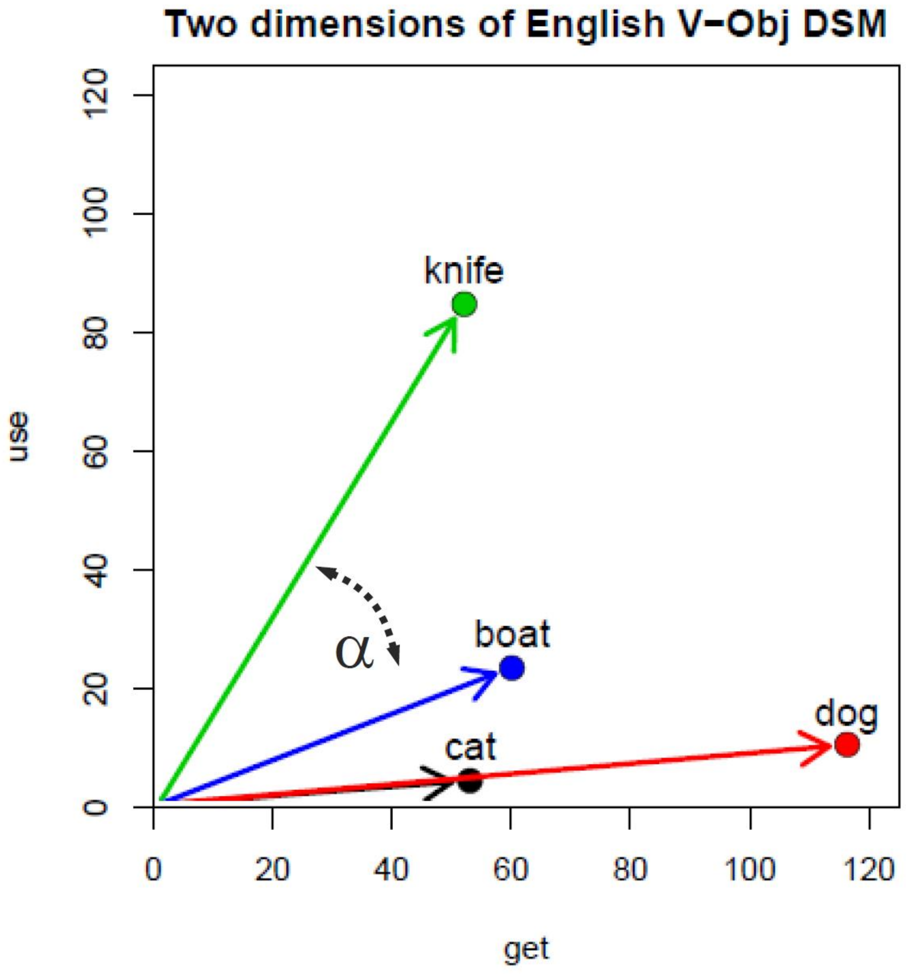
Distance and Similarity

- illustrated for two dimensions: *get* and *use*: $\mathbf{x}_{dog} = (115, 10)$
- similarity = spatial proximity (Euclidean distance)
- location depends on frequency of noun ($f_{dog} \approx 2.7 \cdot f_{cat}$)

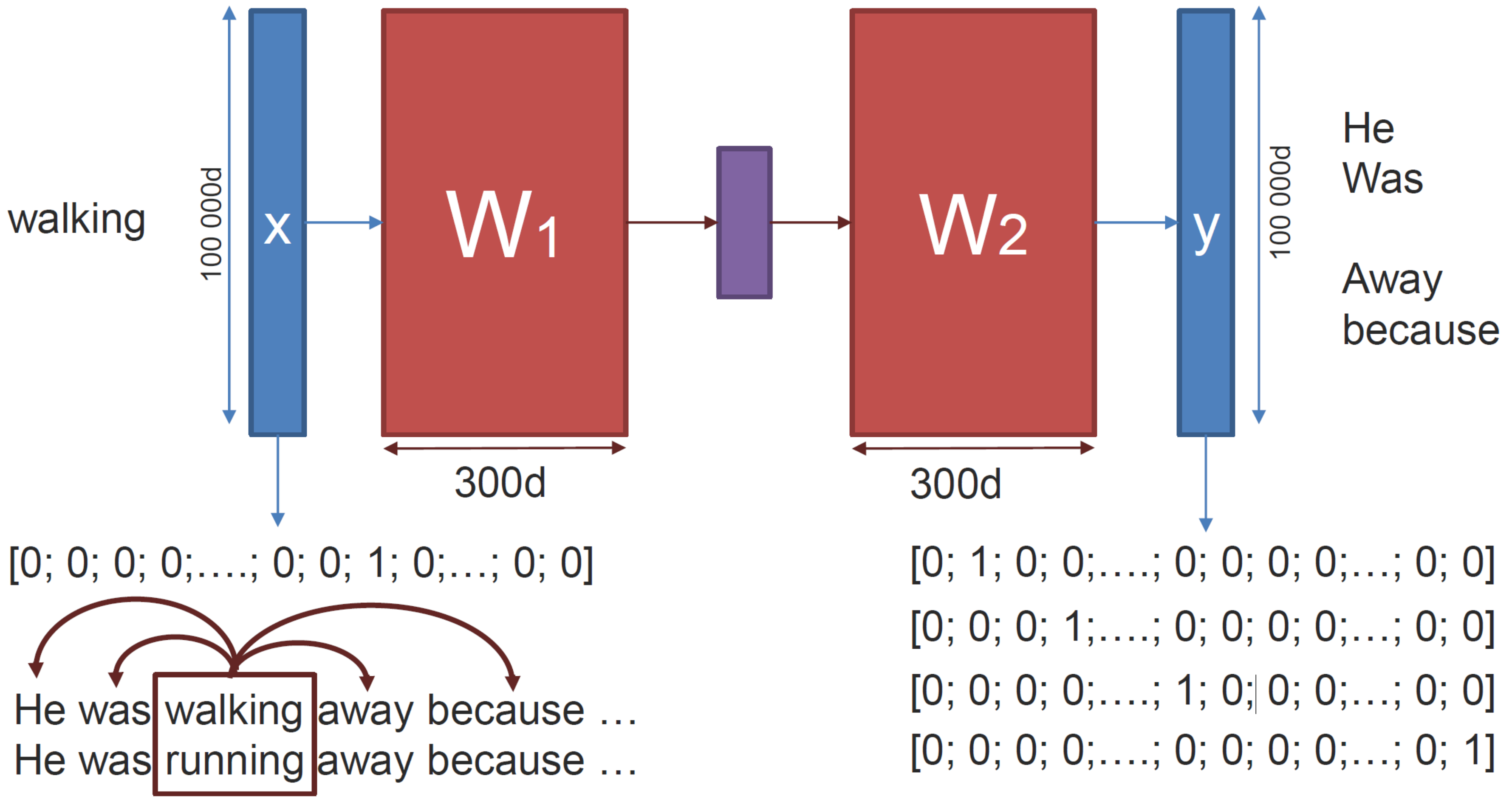


Angle and Similarity

- direction more important than location
- normalise "length" $||\mathbf{x}_{dog}||$ of vector
- or use angle α as distance measure



How to learn (word) features/representations?



Word2vec algorithm: https://code.google.com/p/word2vec/

How to use these word representations

If we would have a vocabulary of 100 000 words:

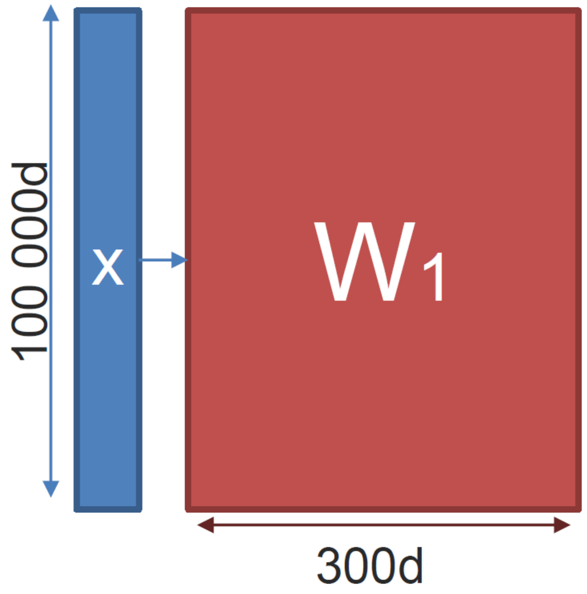
Classic NLP: ← 100 000 dimensional vector →

Walking: [0; 0; 0; 0;….; 0; 0; 1; 0;…; 0; 0]

Running: [0; 0; 0; 0;….; 0; 0; 0; 0;…; 1; 0]

➡ Similarity = 0.0

⬇ Transform: x'=x*W



Goal: ← 300 dimensional vector →

Walking: [0,1; 0,0003; 0;….; 0,02; 0.08; 0,05]

Running: [0,1; 0,0004; 0;….; 0,01; 0.09; 0,05]

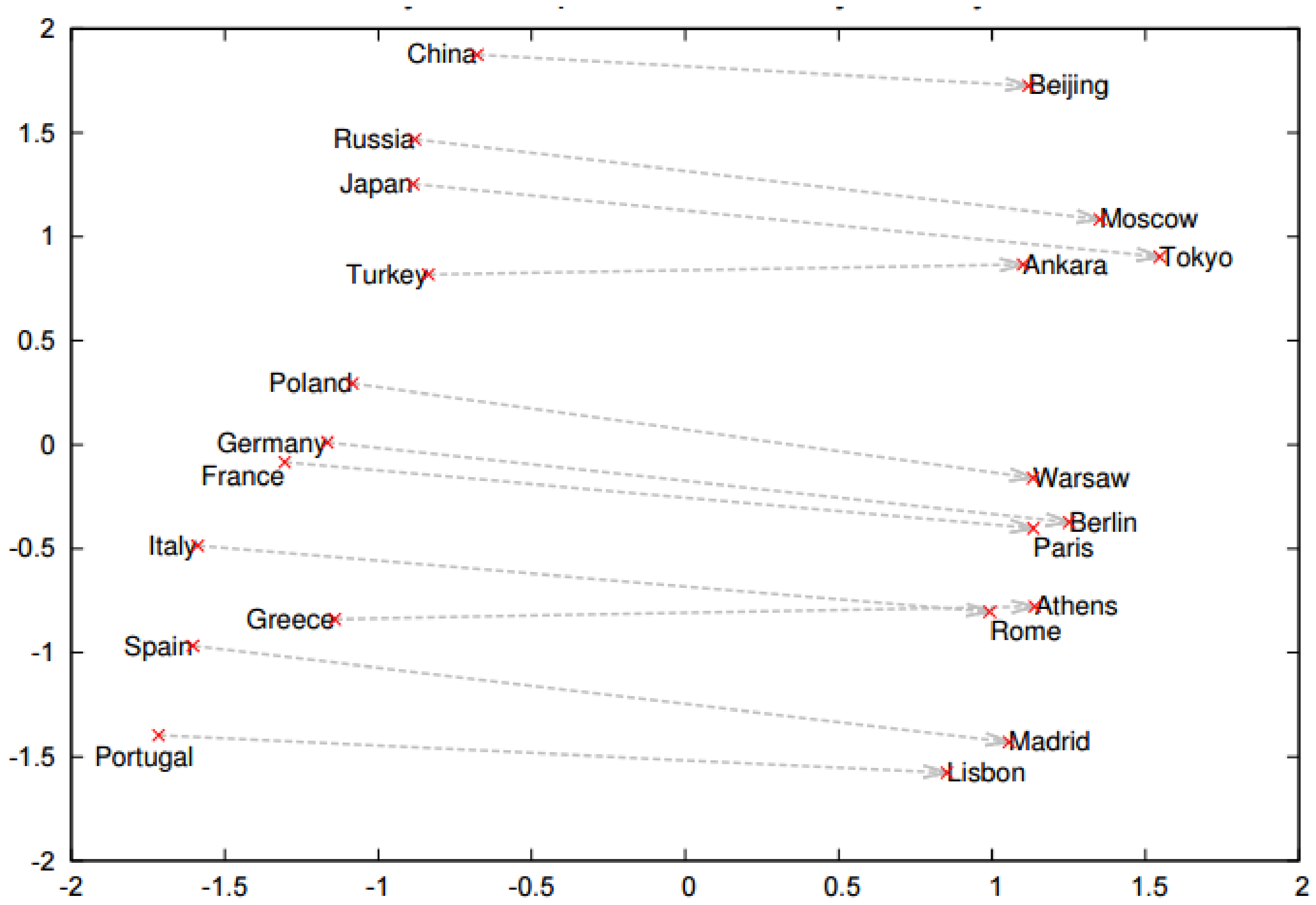
➡ Similarity = 0.9

Vector Space Models of Words

- While learning these word representations, we are actually building a vector space in which all words reside with certain relationships between them
- Encodes both syntactic and semantic relationships
- This vector space allows for algebraic operations:

$$\text{Vec(king)} - \text{vec(man)} + \text{vec(woman)} \approx \text{vec(queen)}$$

Vector Space Models of Words: Semantic Relationships



Trained on the Google news corpus with over 300 billion words

Word Representation Resources

Word-level representations:

Word2Vec (Google, 2013)
https://code.google.com/archive/p/word2vec/
Glove (Stanford, 2014)
https://nlp.stanford.edu/projects/glove/
FastText(Facebook, 2017)
https://fasttext.cc/

Sentence-level representations:

ELMO (Allen Institute for AI, 2018)
https://allennlp.org/elmo
BERT (Google, 2018)
https://github.com/google-research/bert
RoBERTa(Facebook, 2019)
https://github.com/pytorch/fairseq

Word representations are contextualized using all the words in the sentence.

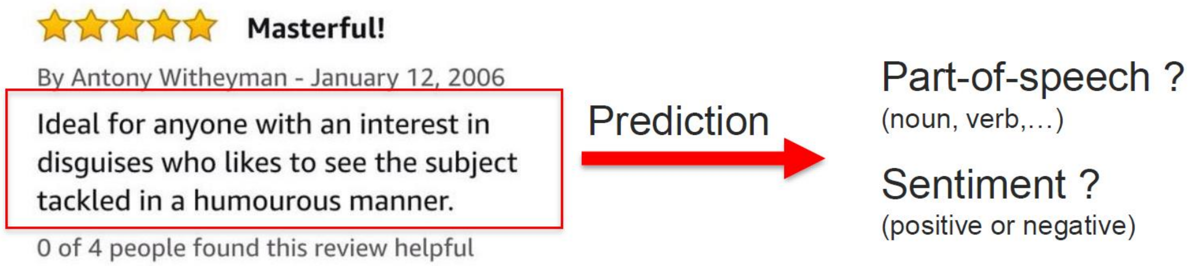
内容提纲

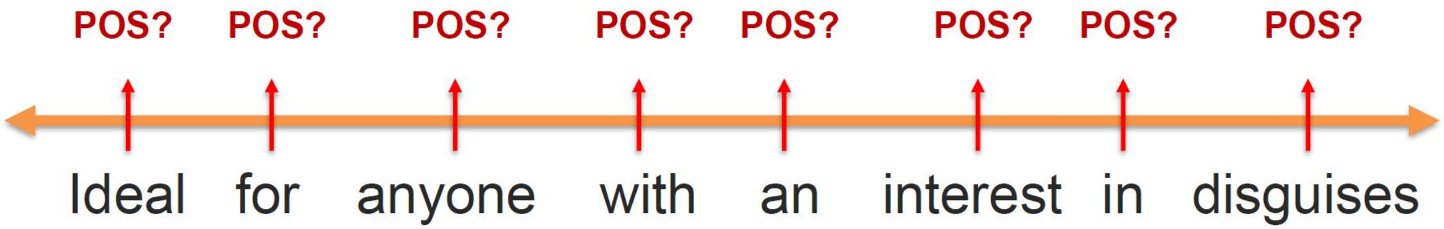
① 词表示

② 递归神经网络

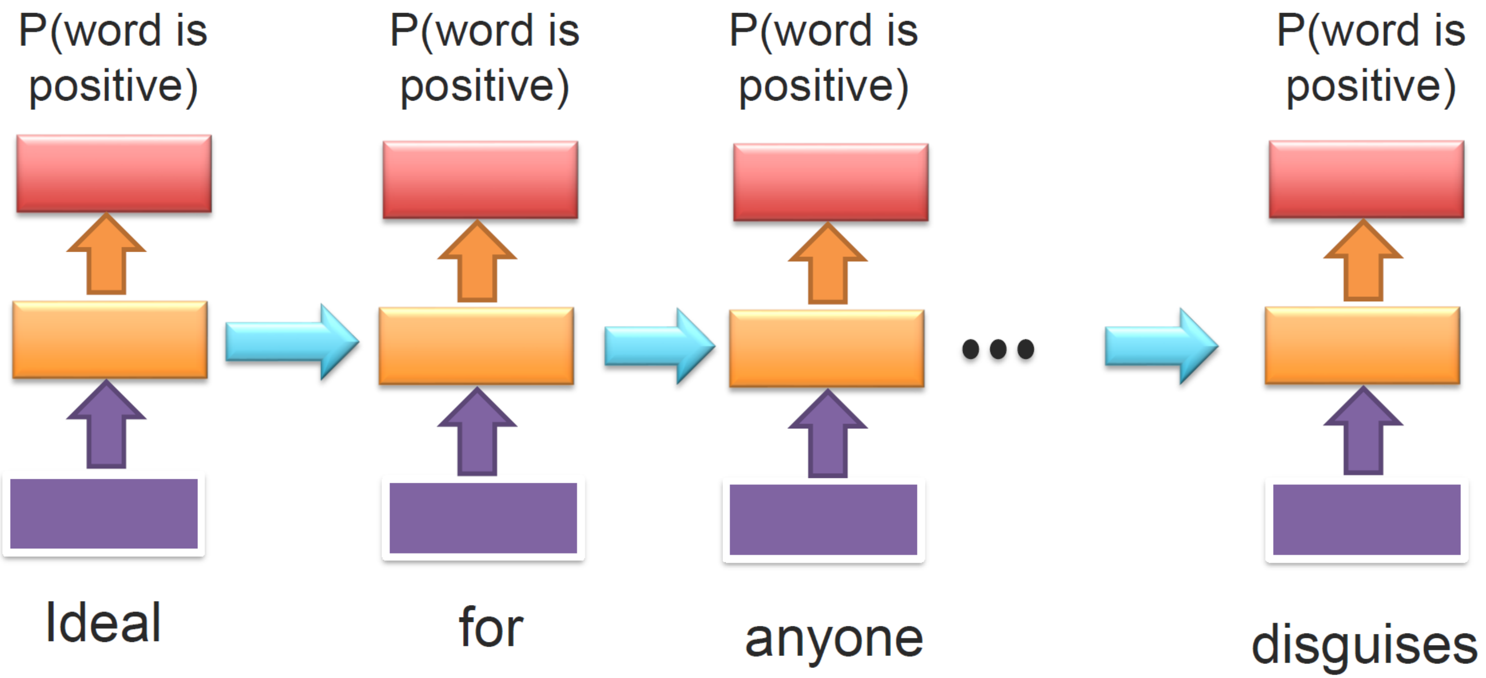
③ 语言模型

Sentence Modeling: Sequence Prediction





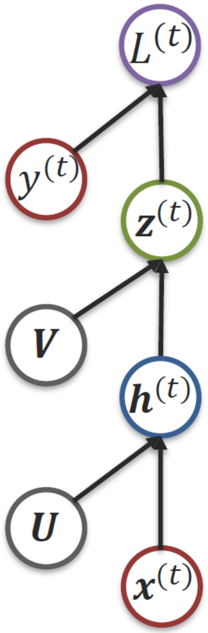
RNN for Sequence Prediction



What is the loss? $L = \frac{1}{N}\sum_{t} L^{(t)} = \frac{1}{N}\sum_{t} -\log P(Y = y^{(t)} | z^{(t)})$

Recurrent Neural Network

Feedforward Neural Network



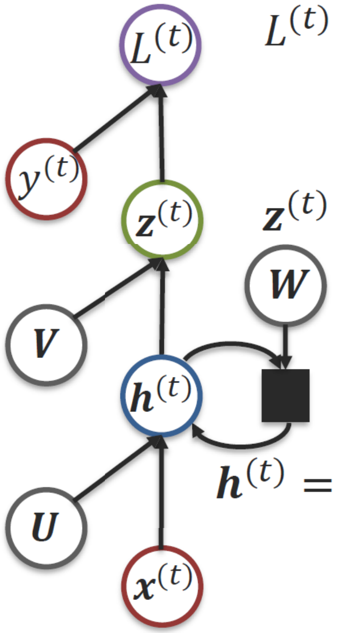
$$L^{(t)} = -logP(Y = y^{(t)} | \boldsymbol{z}^{(t)})$$

$$\boldsymbol{z}^{(t)} = matmult(\boldsymbol{h}^{(t)}, \boldsymbol{V})$$

$$\boldsymbol{h}^{(t)} = tanh(\boldsymbol{U}\boldsymbol{x}^{(t)})$$

Recurrent Neural Network

$$L = \sum_t L^{(t)}$$



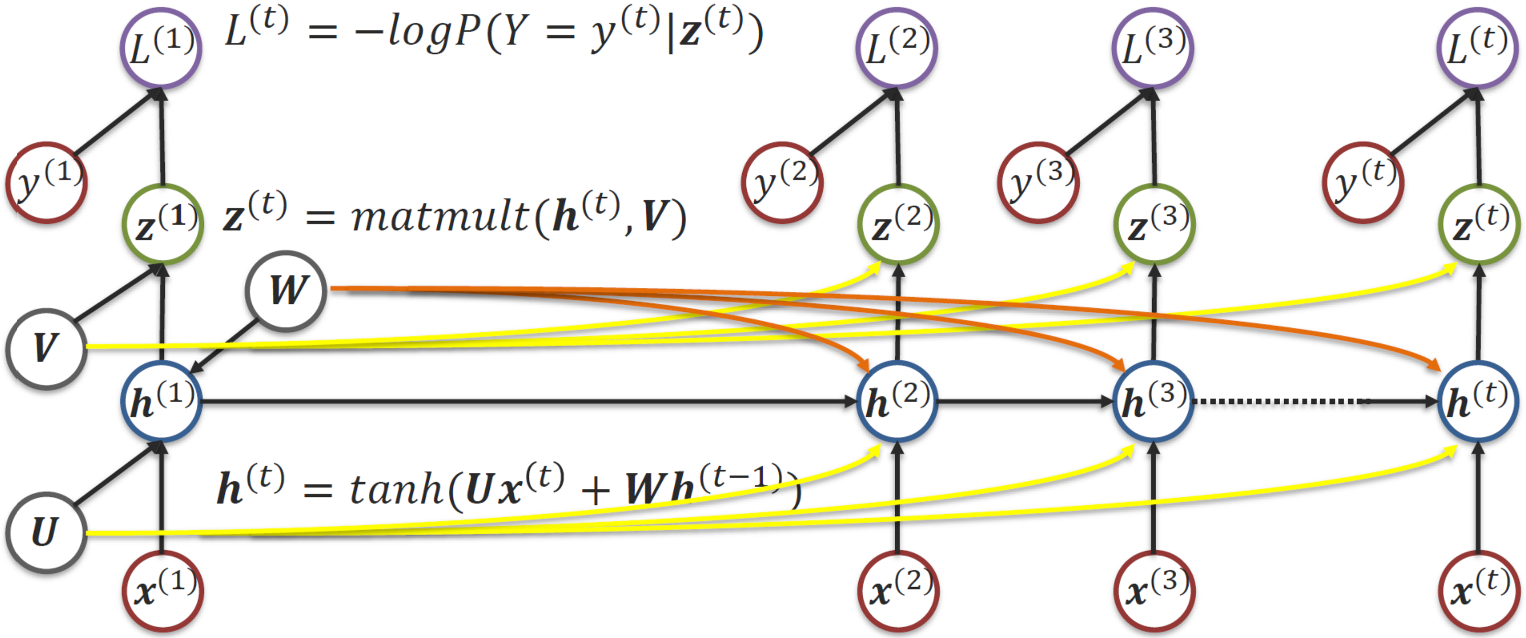
$$L^{(t)} = -logP(Y = y^{(t)} | \boldsymbol{z}^{(t)})$$

$$\boldsymbol{z}^{(t)} = matmult(\boldsymbol{h}^{(t)}, \boldsymbol{V})$$

$$\boldsymbol{h}^{(t)} = tanh(\boldsymbol{U}\boldsymbol{x}^{(t)} + \boldsymbol{W}\boldsymbol{h}^{(t-1)})$$

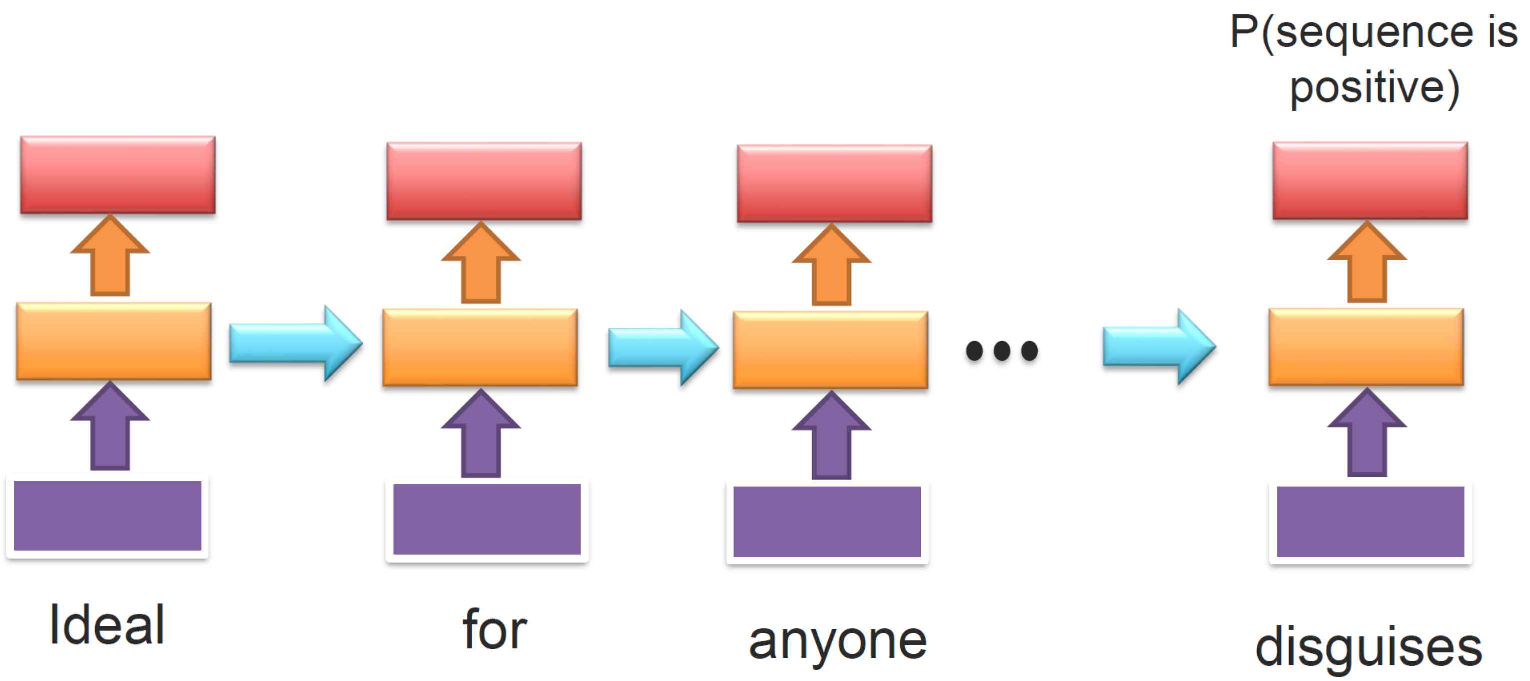
Recurrent Neural Network

$$L = \sum_t L^{(t)}$$



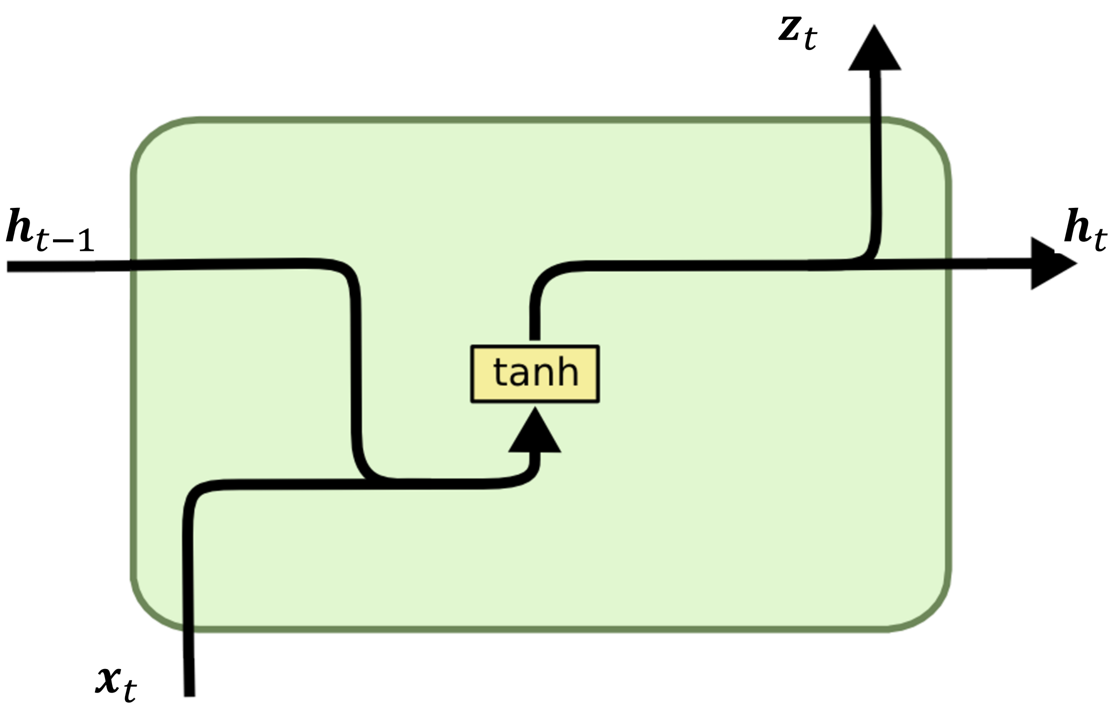
Same model parameters are used for all time parts.

RNN for Sequence Prediction



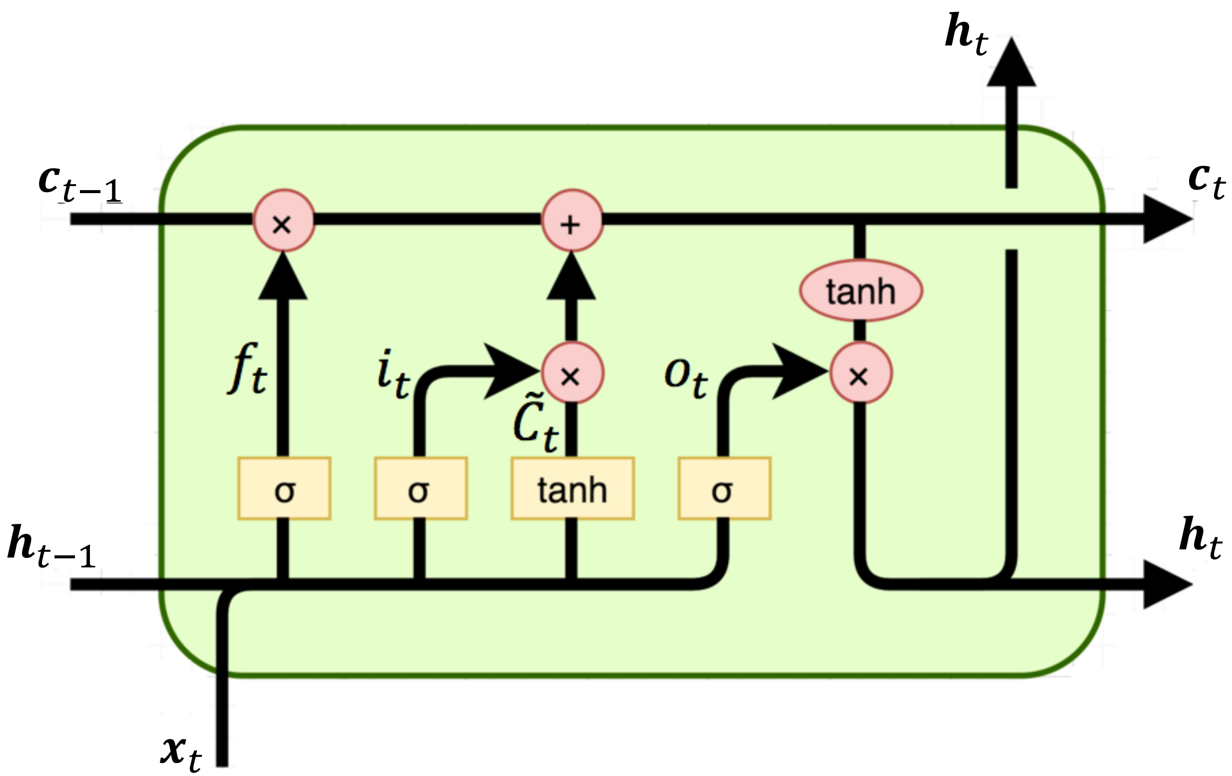
What is the loss? $L = L^{(N)} = -logP(Y = y^{(N)} | z^{(N)})$

Recurrent Neural Network

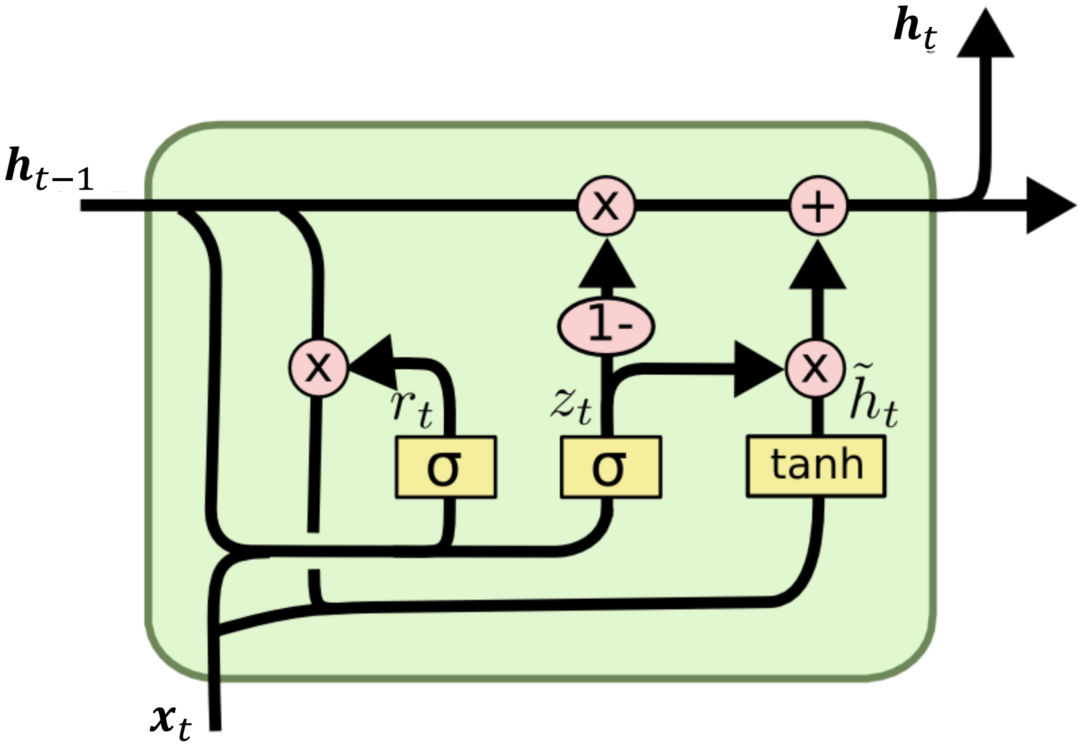


RNN suffers from gradient vanishment for long sequence

LSTM: Long Short Term Memory



GRU: Gated Recurrent Unit



内容提纲

① 词表示

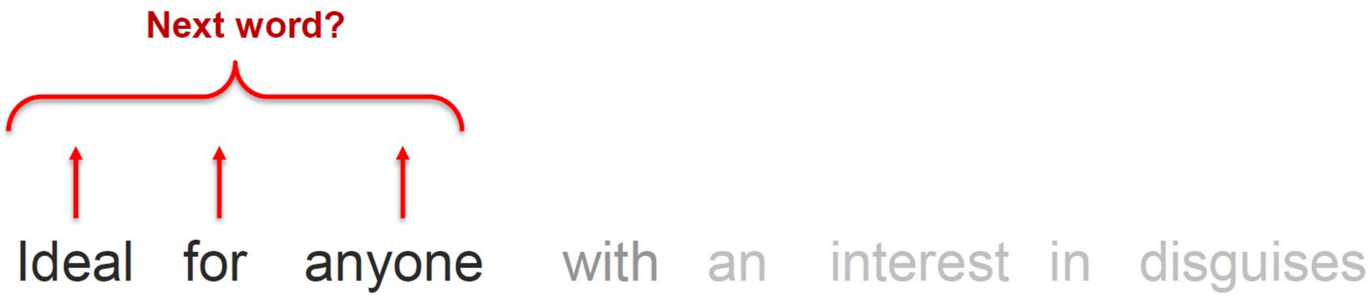
② 递归神经网络

③ 语言模型

Sentence Modeling: Language Model


Masterful!
By Antony Witheyman - January 12, 2006
Ideal for anyone with an interest in disguises who likes to see the subject tackled in a humourous manner.
0 of 4 people found this review helpful

Prediction → Next word



Language Model Application: Speech Recognition

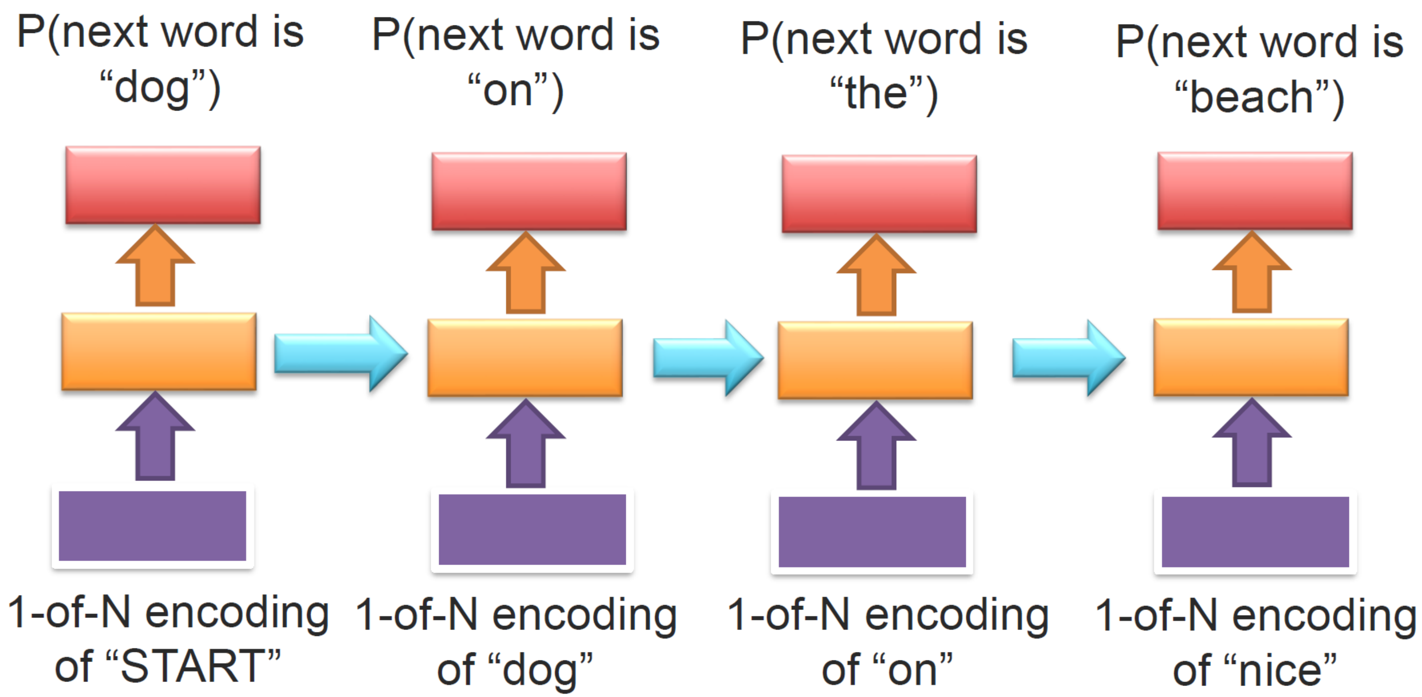
$$\arg\max_{wordsequence} P(wordsequence \mid acoustics) =$$

$$\arg\max_{wordsequence} \frac{P(acoustics \mid wordsequence) \times P(wordsequence)}{P(acoustics)}$$

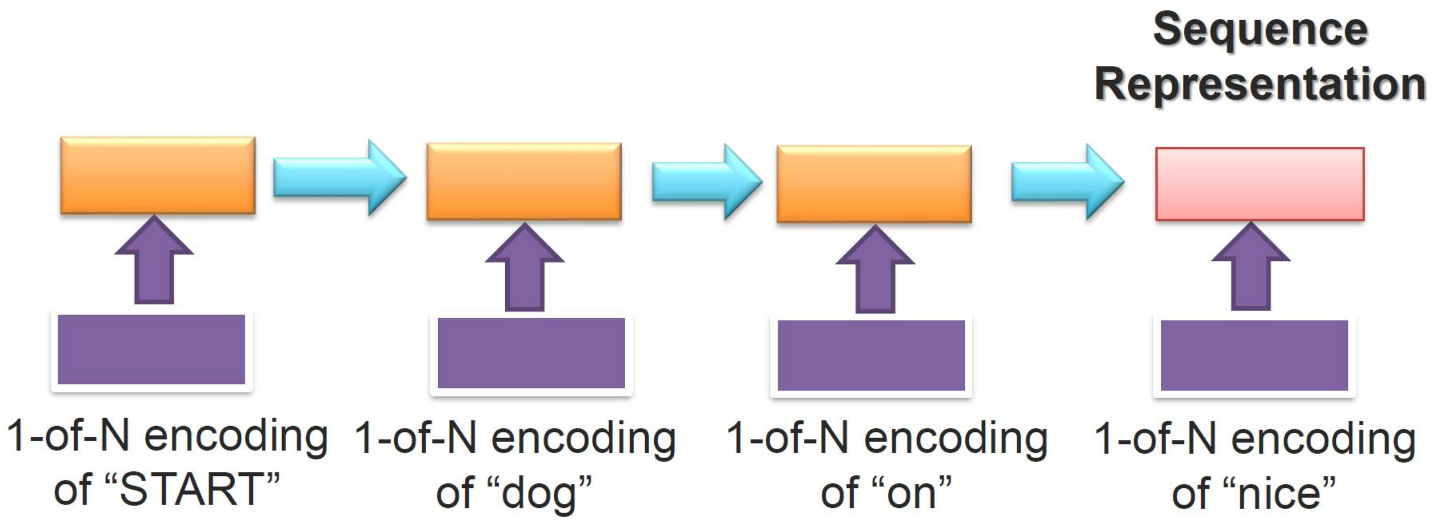
$$\arg\max_{wordsequence} P(acoustics \mid wordsequence) \times P(wordsequence)$$

Language model

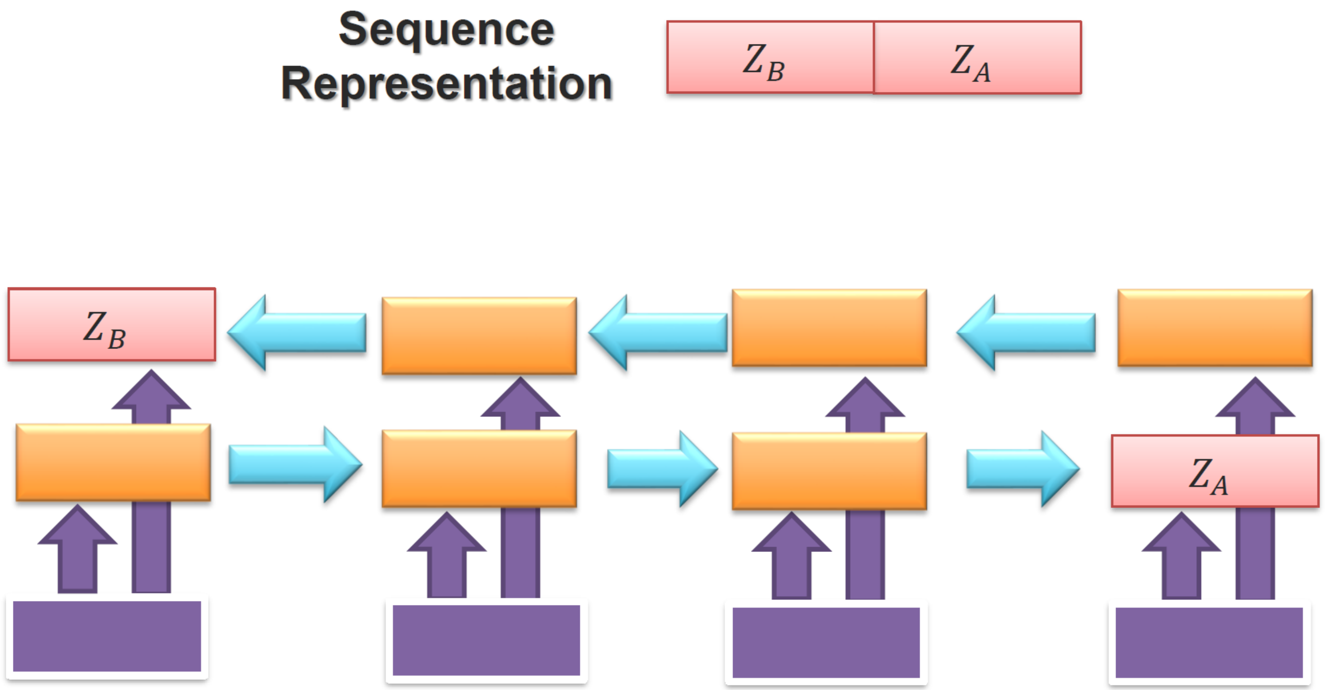
RNN for Language Model



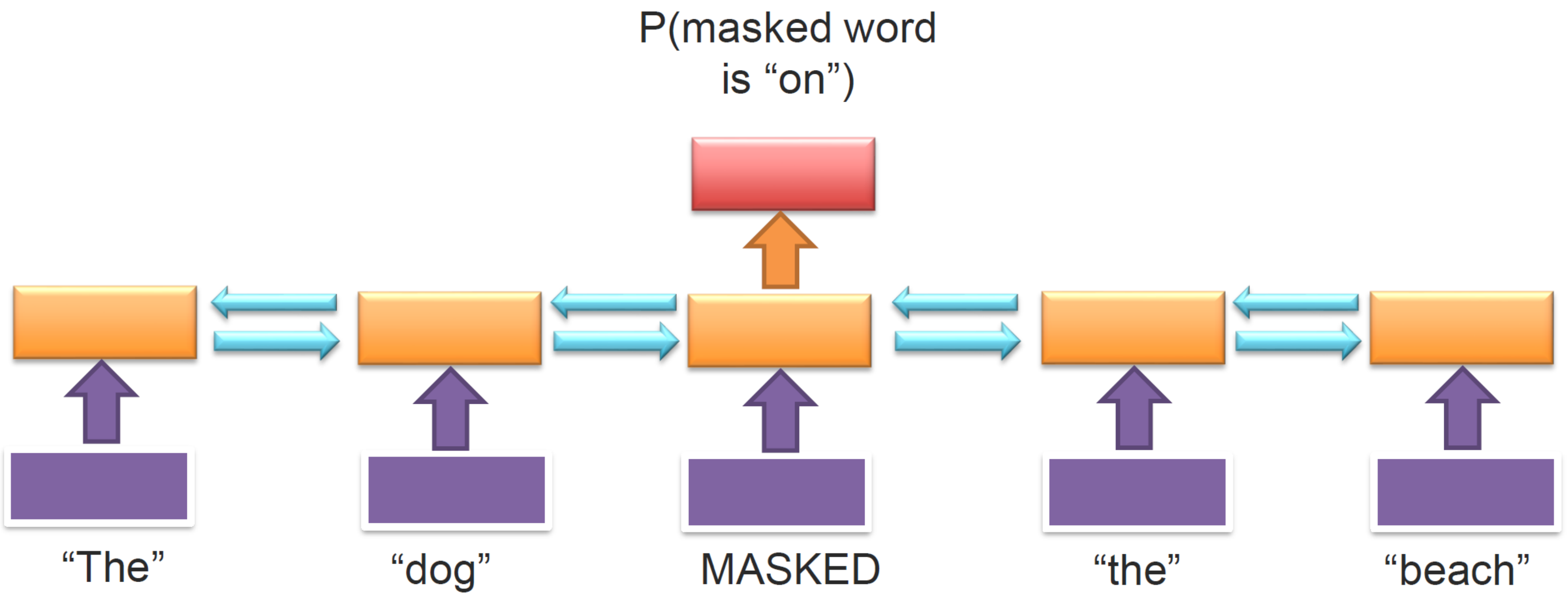
RNN for Sequence Representation (Encoder)



Bi-Directional RNN



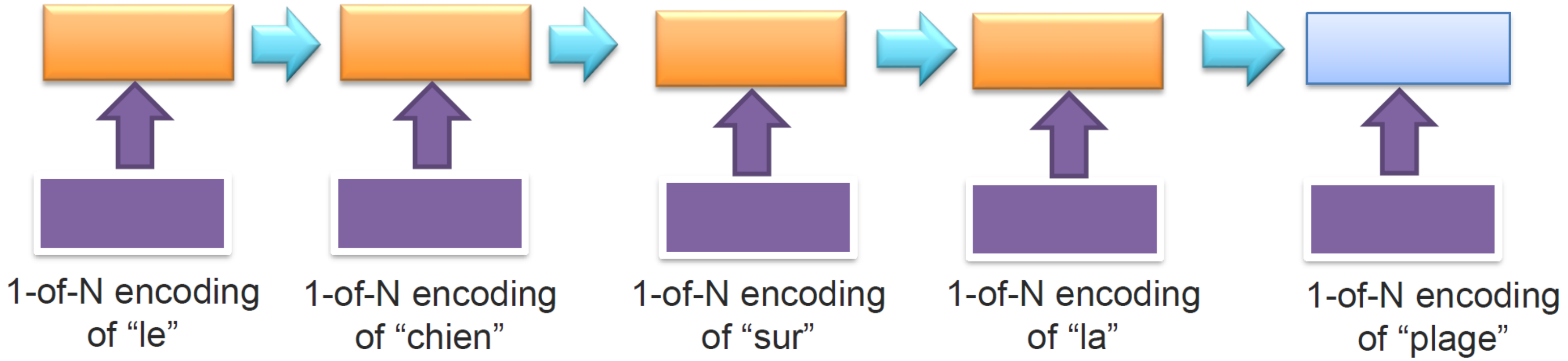
Pre-training and “Masking”



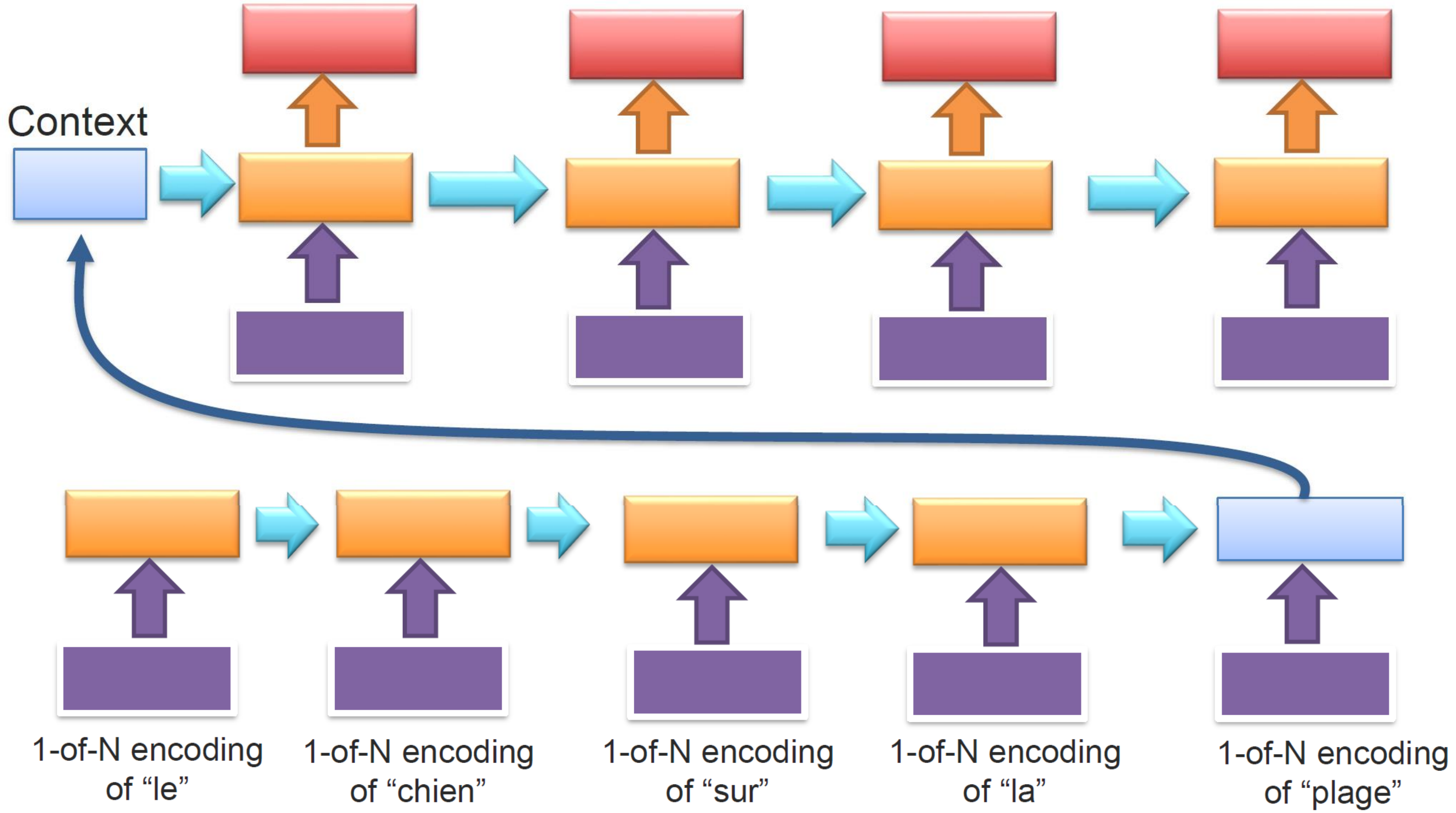
(short-lived) ELMO was a bi-directional pretrained language model

RNN-based for Machine Translation

Le chien sur la plage → The dog on the beach



Encoder-Decoder Architecture



And There Are More Ways To Model Sequences…

BERT & GPT

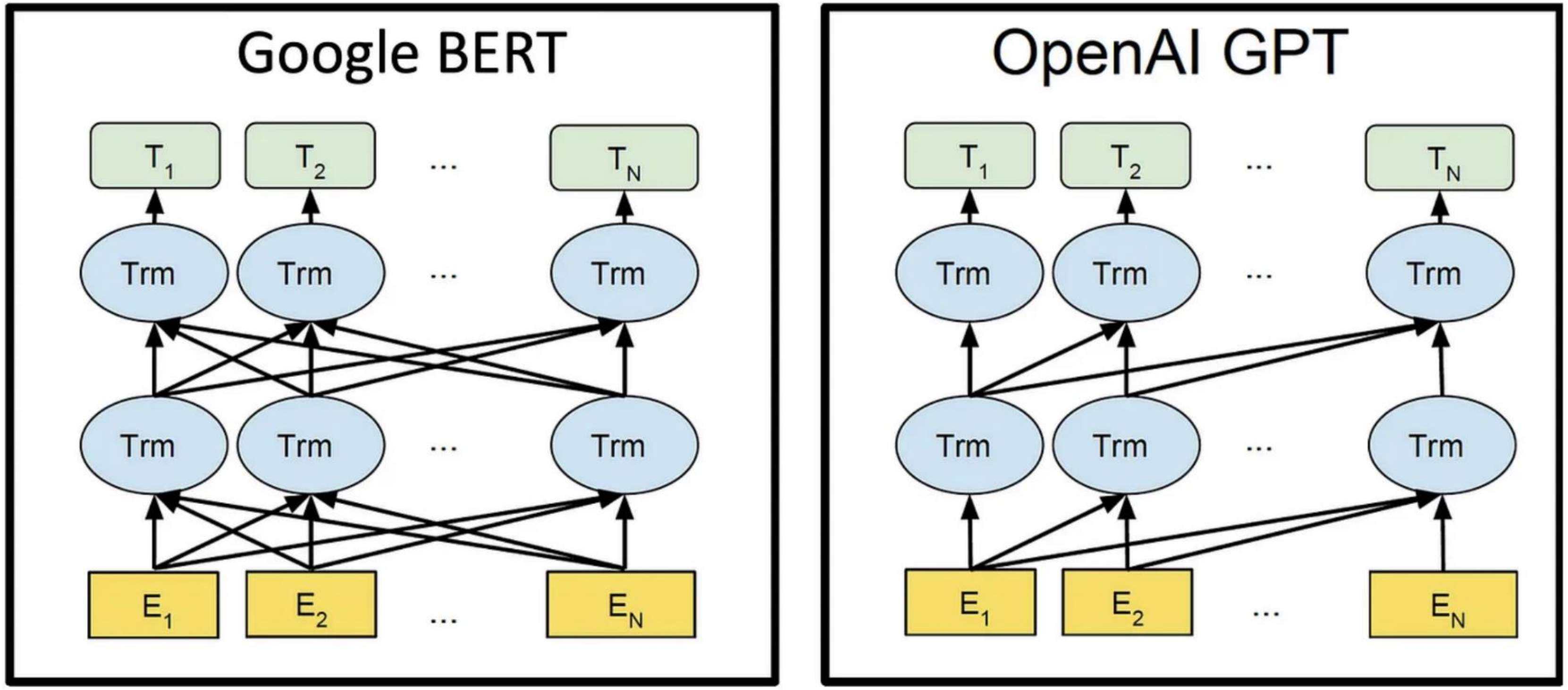


Fig 9: BERT vs GPT. BERT: transformer **encoder-based**, **bidirectional.** GPT: transformer **decoder-based**, **left-to-right. Image Source:** Devlin, et al., 2018