

Natural Language Processing

Shangbin Feng

Xi'an Jiaotong University

wind_binteng@stu.xjt.edu.cn

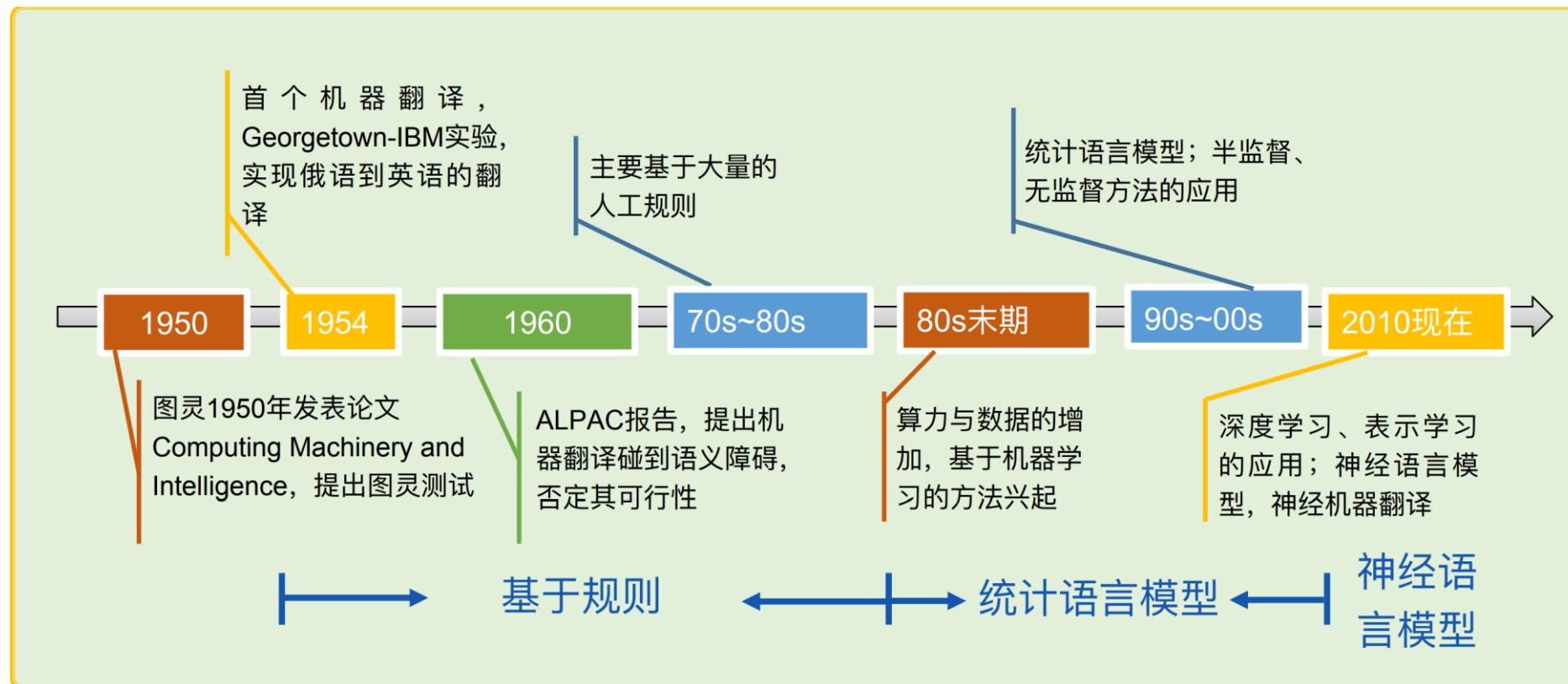
October, 2021

Contents

- Introduction to NLP
- Word Embedding and RNNs
- Attention and Transformers
- Self-supervised Learning
- Pre-trained Language Models

What is NLP?

- Natural Language Processing



Task 1: PoS Tagging

A dog is a very common four-legged animal that is often kept by people
as a pet or to guard or hunt .

Adjective

Adverb

Conjunction

Determiner

Noun

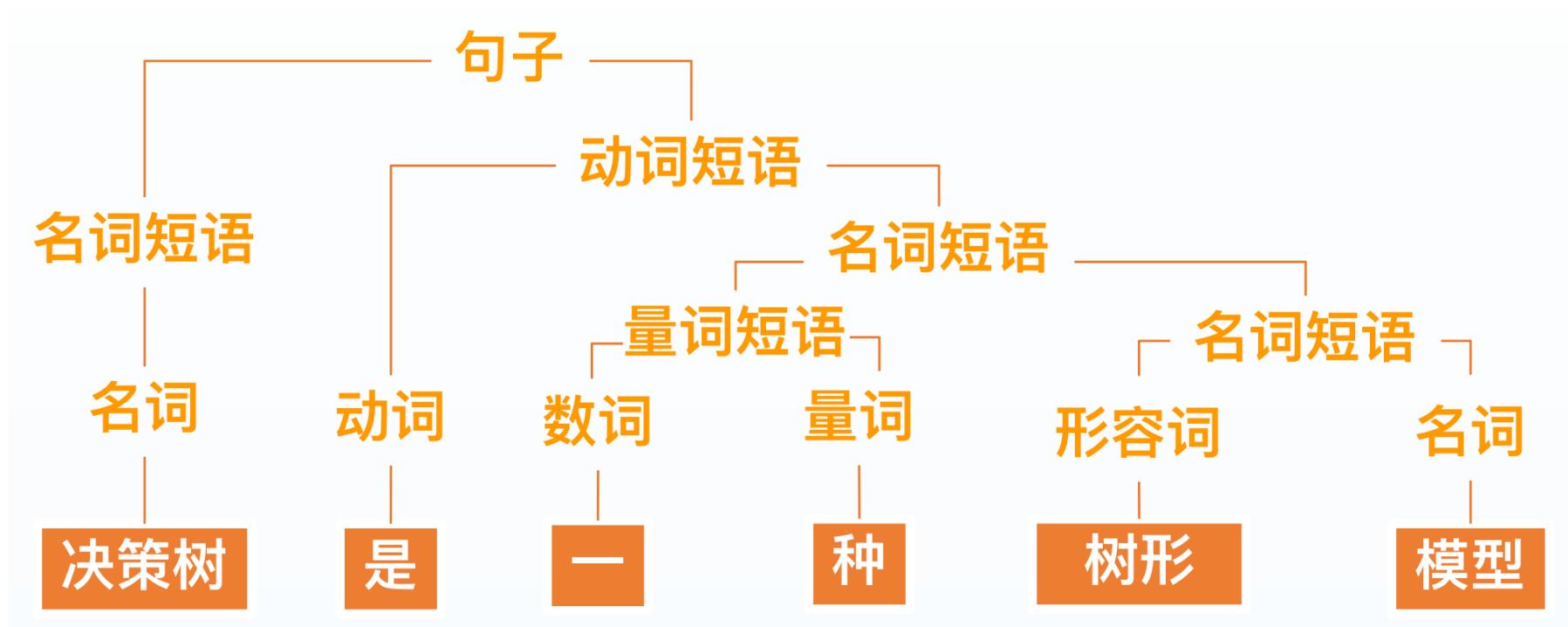
Number

Preposition

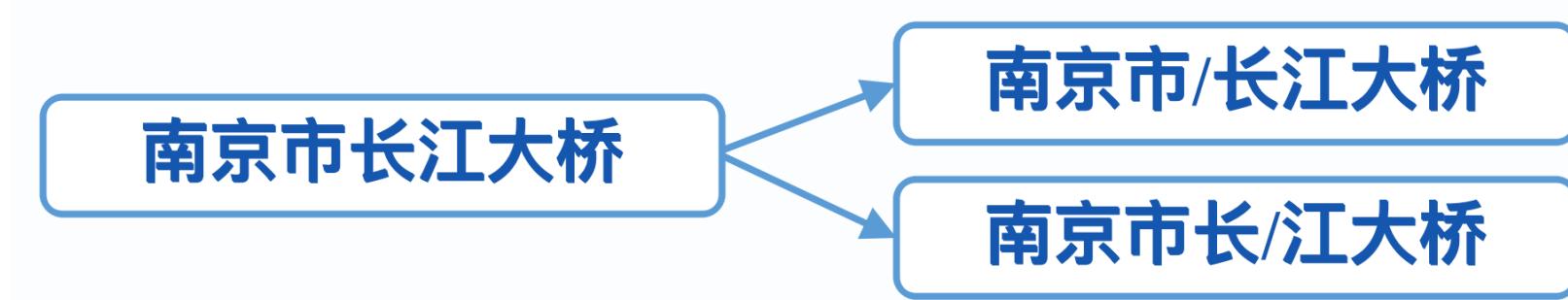
Pronoun

Verb

Task 2: Parsing



Task 3: Tokenization



Task 4: Machine Translation



Task 5: Named Entity Recognition

Obama is the president of the United States

Jim bought 300 shares of Acme Corp. in 2006

Task 6: Text Classification

- ✓ 垃圾邮件过滤
- ✓ 情感识别
- ✓ 新闻分类
- ✓ 色情文档识别

Task 7: Question Answering

Task 8: Sentiment Analysis

- ✓ 《复联3》是一部史无前例的电影
- ✓ 让人有些绝望的电影结局
- ✓ MIUI8系统还算流畅，功能多，人性化，但是广告不能完全关闭

Task 9: Coreference Resolution

- ✓ 甲队打败了乙队，他们更强
- ✓ 虽然甲队打败了乙队，但他们都很强

Contents

- Introduction to NLP
- Word Embedding and RNNs
- Attention and Transformers
- Self-supervised Learning
- Pre-trained Language Models

Why word embedding?

One-hot Encoding

<i>apple</i>	= [1 0 0]
<i>banana</i>	= [0 1 0]
<i>pineapple</i>	= [0 0 1]

- Problems
 - Embedding size
 - transductive
 - No meaning in it

```
motel = [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
```

```
hotel = [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
```

Idea

- “You shall know a word by the company it keeps.”

*...government debt problems turning into **banking** crises as happened in 2009...*

*...saying that Europe needs unified **banking** regulation to replace the hodgepodge...*

*...India has just given its **banking** system a shot in the arm...*

Word Embedding

- Distributed embedding
 - Dense vector
 - Word vector
 - Word representation
 - Distributed representation
-
- Limited dimension / word meaning included

$$\text{banking} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Word2Vec

- “You shall know a word by the company it keeps”
 - CBOW
 - Skip-gram

CBOW

- Continuous bag of words

1. Generate one-hot word vectors for the input context of size m:

$$(x^{c-m}, \dots, x^{c-1}, x^{c+1}, \dots, x^{c+m} \in R^{|V|}).$$

2. Get embedded word vectors for the context.

$$v_{c-m} = x^{c-m} \cdot \mathbf{W}, \dots, v_{c+m} = x^{c+m} \cdot \mathbf{W} \in R^N$$

3. Average context word vectors get \hat{v} .

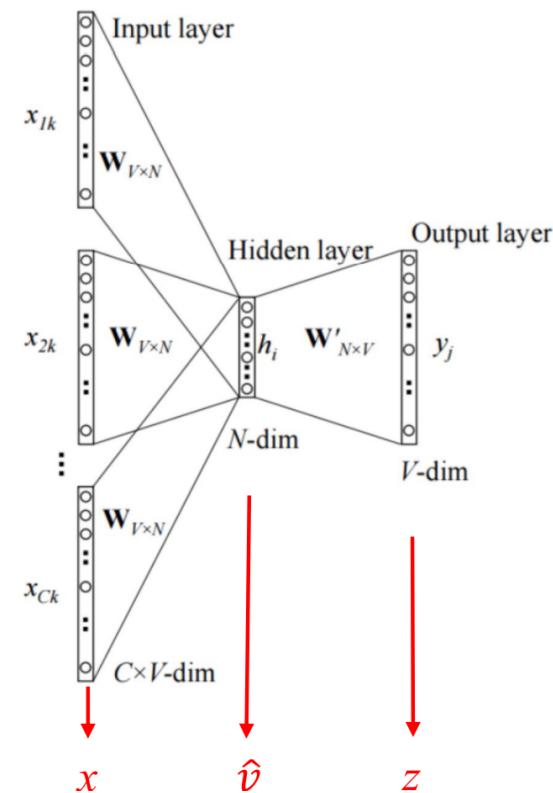
$$\hat{v} = \frac{v_{c-m} + \dots + v_{c+m}}{2m} \in R^N$$

4. Generate a score vector z.

$$z = \hat{v} \cdot \mathbf{W}' \in R^{|V|}$$

5. Turn the score vector into probabilities \hat{y} .

$$\hat{y} = softmax(z)$$



Skip-gram

1. Generate one-hot input vector $x \in R^{|V|}$ of the center word.

2. Get embedded word vectors for the center word.

$$v_c = x \cdot W \in R^N$$

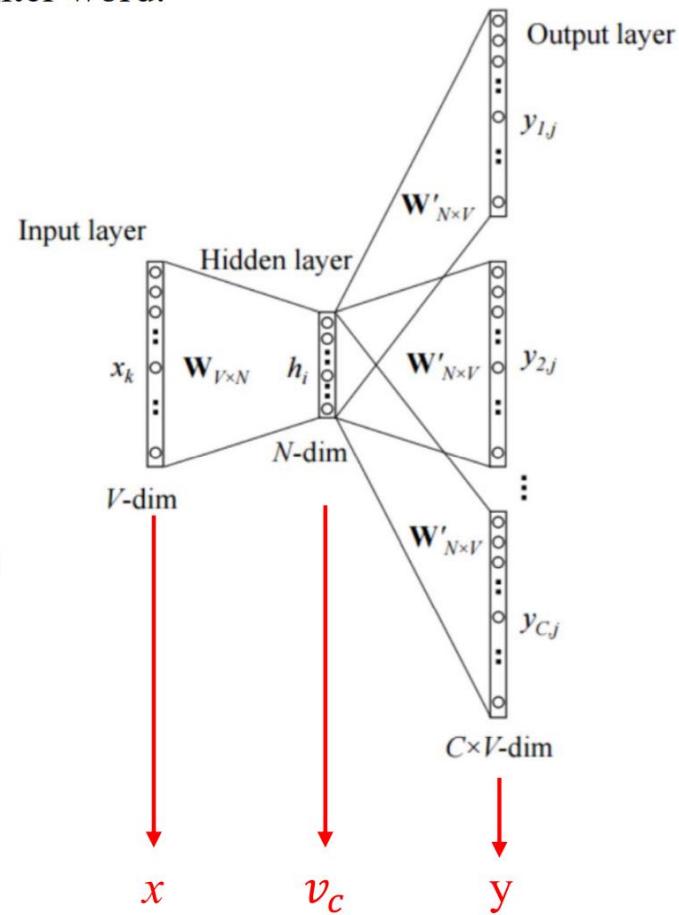
3. Generate a score vector z.

$$z = v_c \cdot W'$$

4. Turn the score vector into probabilities \hat{y} .

$$\hat{y} = softmax(z)$$

5. Note that $\hat{y}_{c-m}, \dots, \hat{y}_{c-1}, \hat{y}_{c+1}, \dots, \hat{y}_{c+m}$ are the probabilities of observing context word.



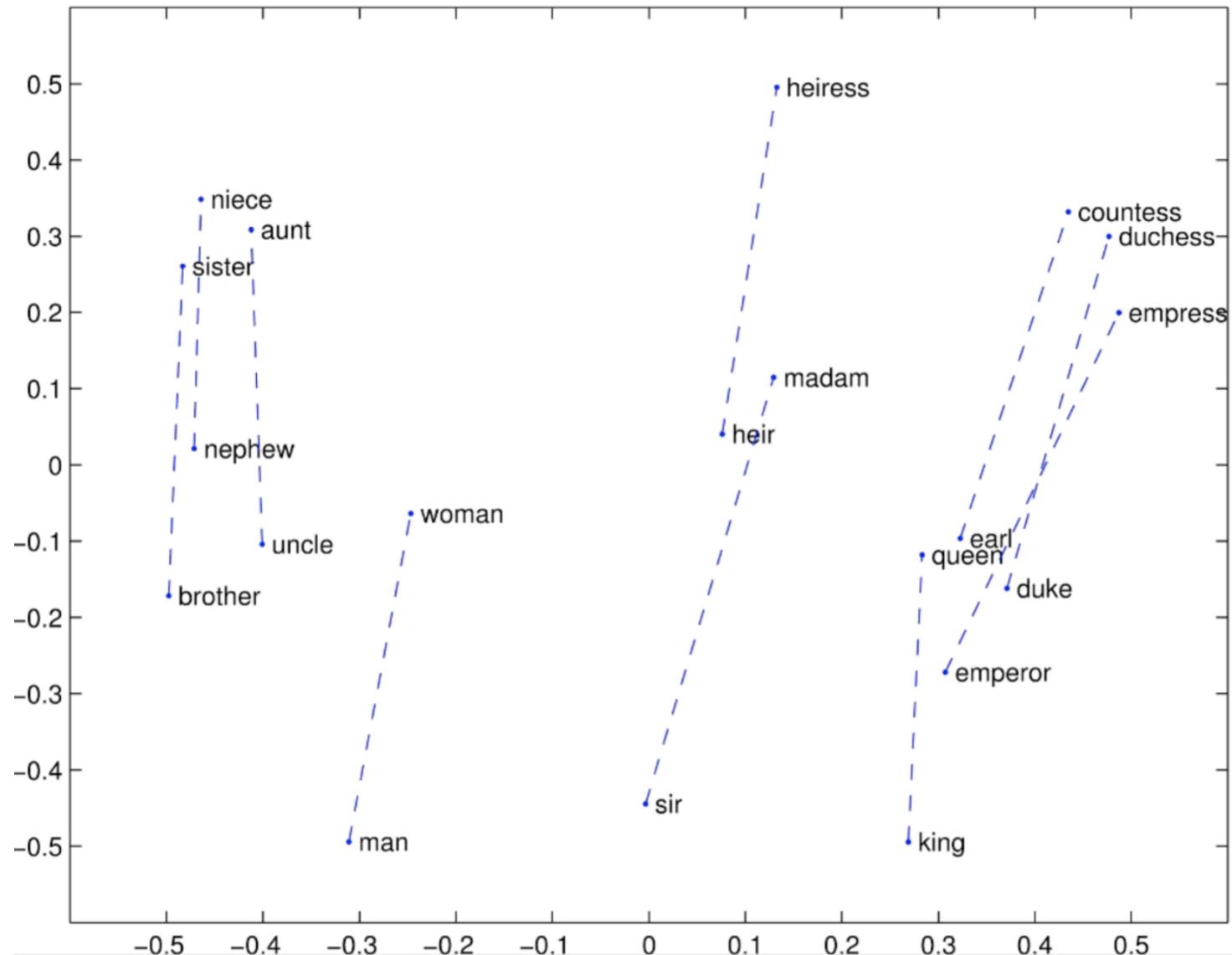
Summary of Word2Vec

- What's the idea?
- What's the difference between CBOW and Skip-gram?
- Problem of vocabulary size
 - Negative sampling & hierarchical softmax

How do we evaluate word embeddings?

- Intrinsic evaluation
- Extrinsic evaluation

Intrinsic Evaluation



Extrinsic Evaluation

- Use word vectors on downstream tasks

Glove

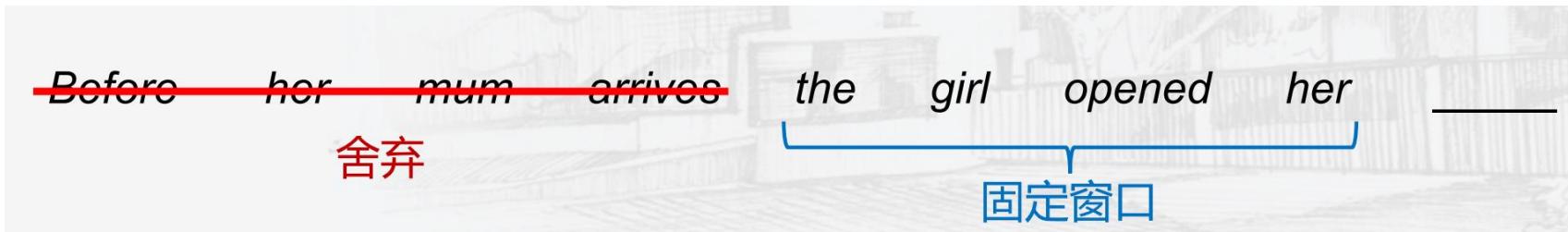
- Word embeddings by Christopher Manning @ Stanford
- Global Vectors for Word Representations
- <https://nlp.stanford.edu/projects/glove/>
- Download
 - Code
 - Trained word embeddings

Language Model (LM)

- The girl opened her _____
 - Laptop
 - Books
 - ...
- A language model tries to **predict the next word (token)** given the previous token sequence.

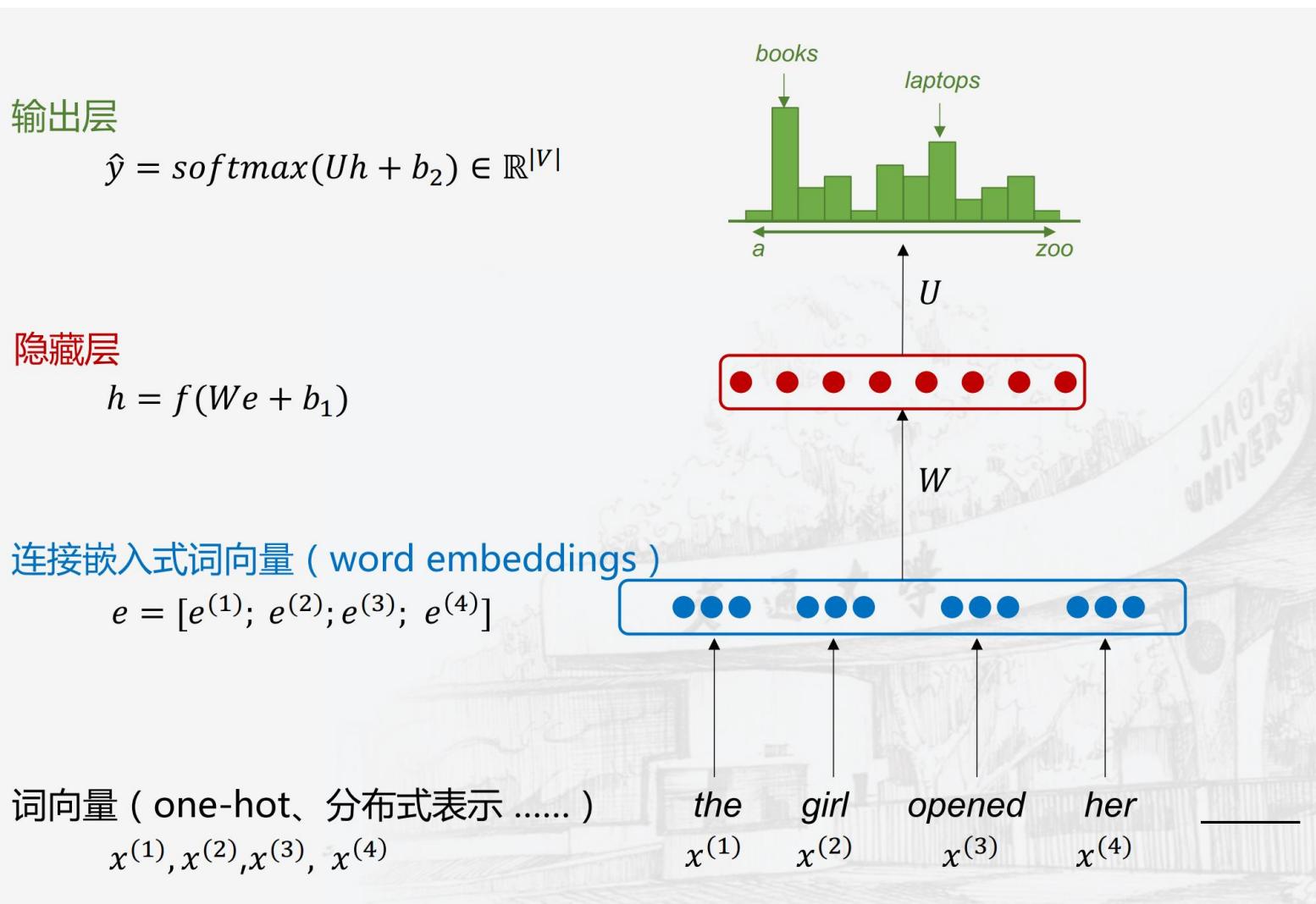
$$P(x^{(t+1)} | x^{(t)}, \dots, x^{(1)})$$

Context window



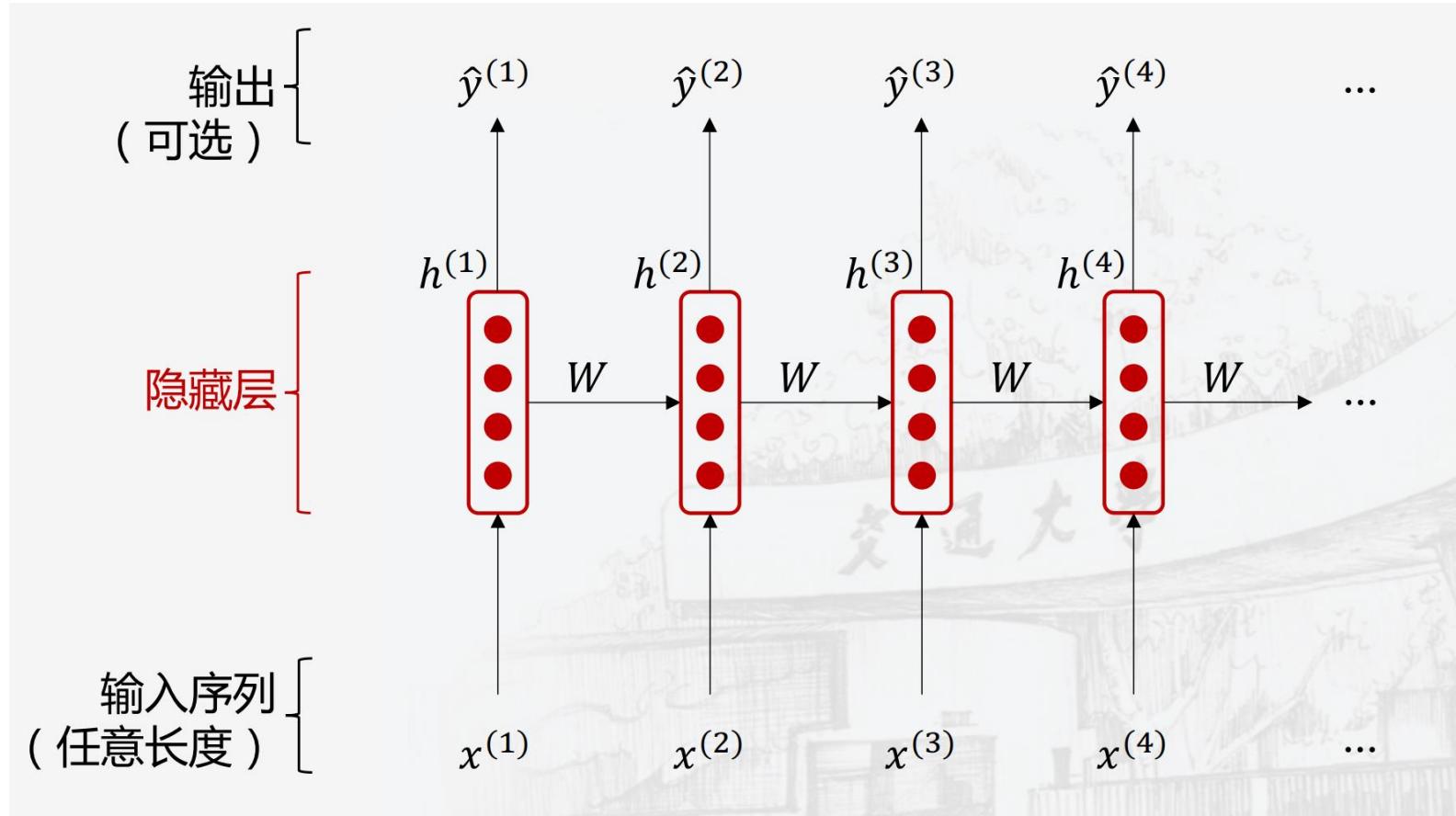
Linear Layer LM

- Problem
 - Window size
 - W parameter size
 - Never enough W



Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs)



RNN LM

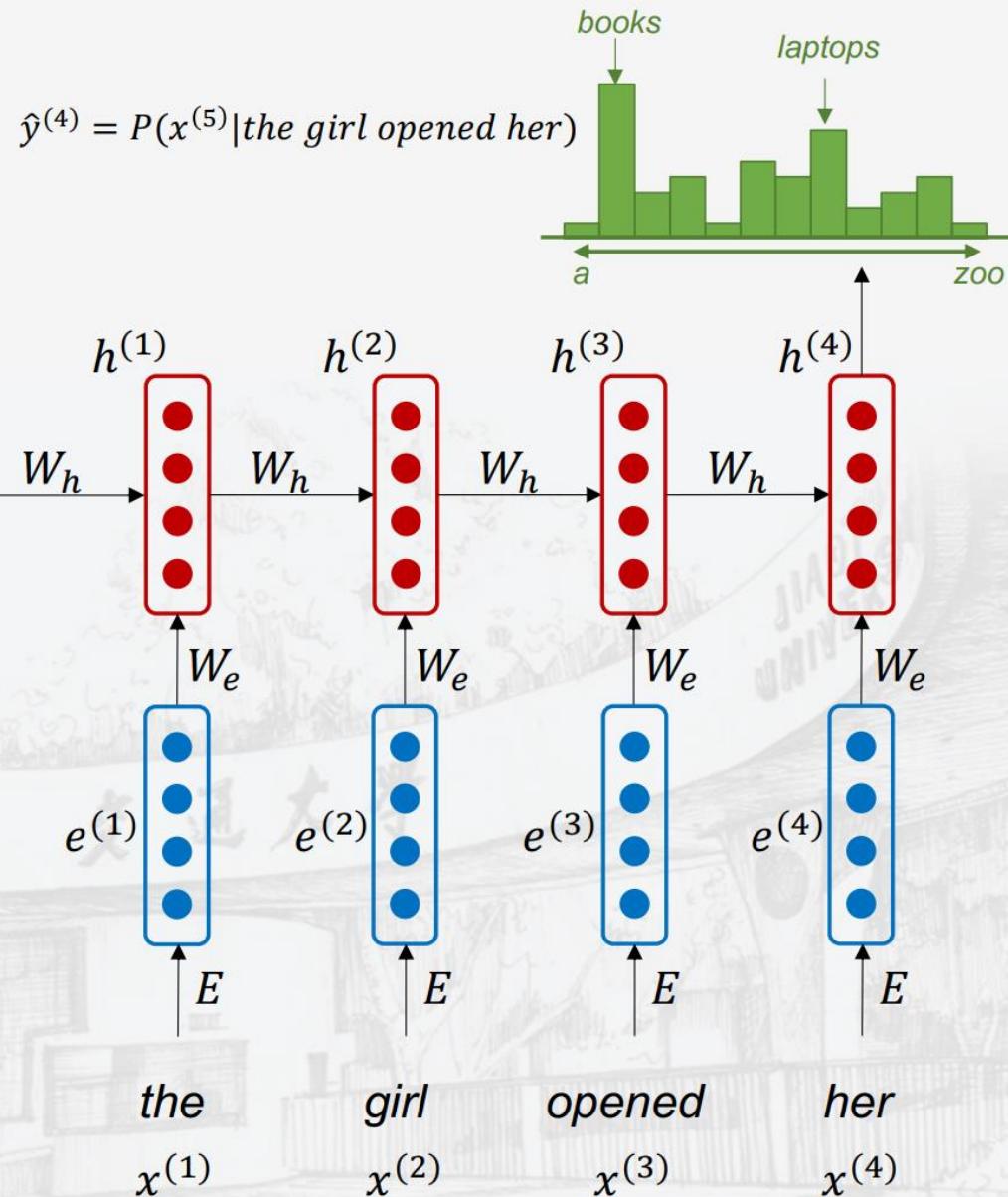
基于递归神经网络的LM

优点

- 可处理任意长度句子；
- 第 t 步的计算（理论上）使用了前面多步的信息；
- 模型体量不随着输入变长而增加；
- 每一步使用同一个 W ，降低计算量。

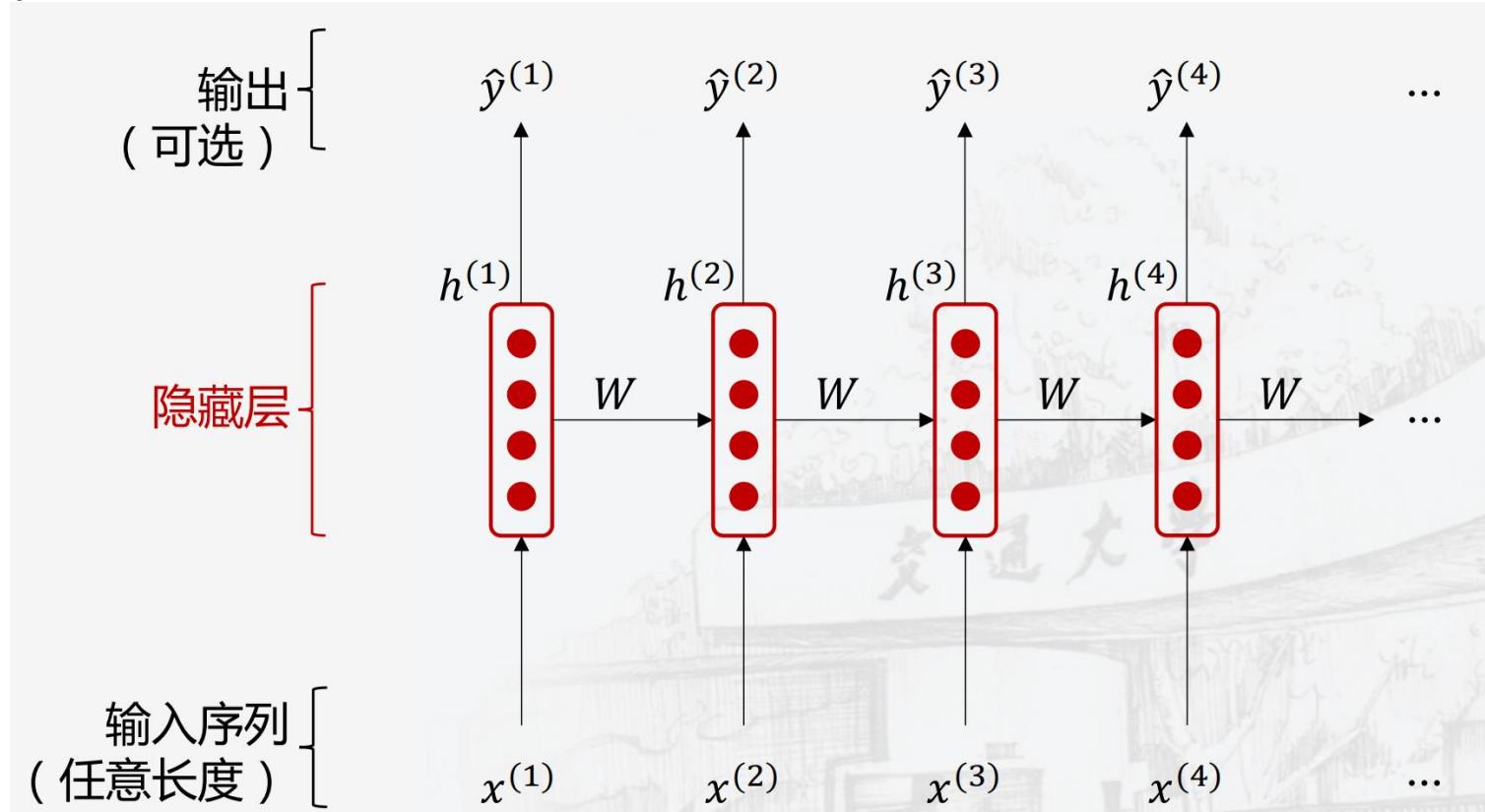
缺点

- 递归计算缓慢；
- 实际上，将前面很多步的信息完整传递是困难的。



Important Issues

- When do we initialize W ?
- When do we initialize $h(0)$?



Text Classification with RNN

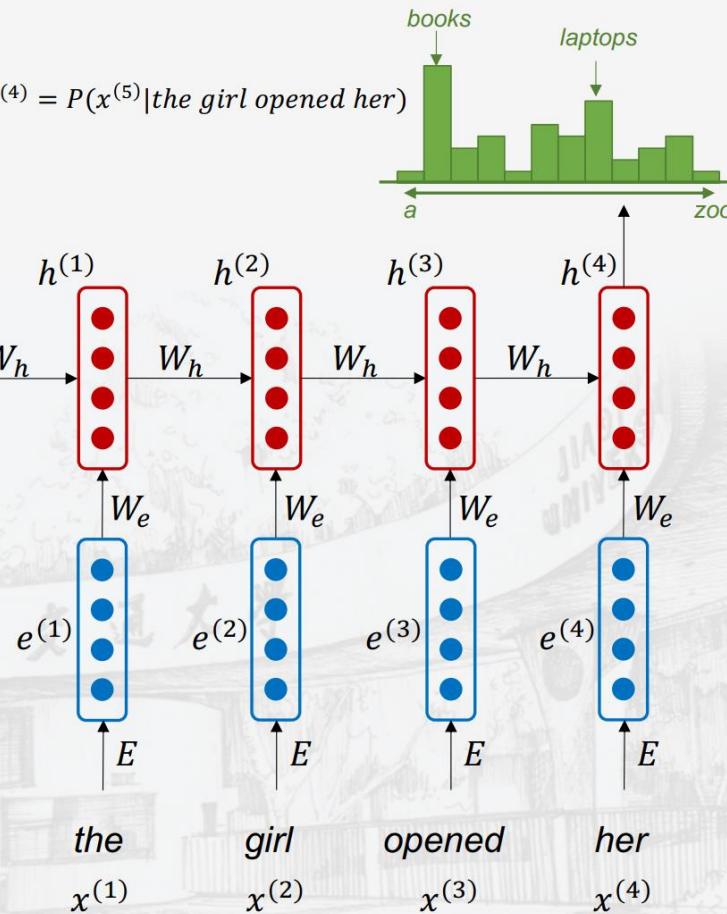
基于递归神经网络的LM

优点

- 可处理任意长度句子；
- 第 t 步的计算（理论上）使用了前面多步的信息；
- 模型体量不随着输入变长而增加；
- 每一步使用同一个 W ，降低计算量。

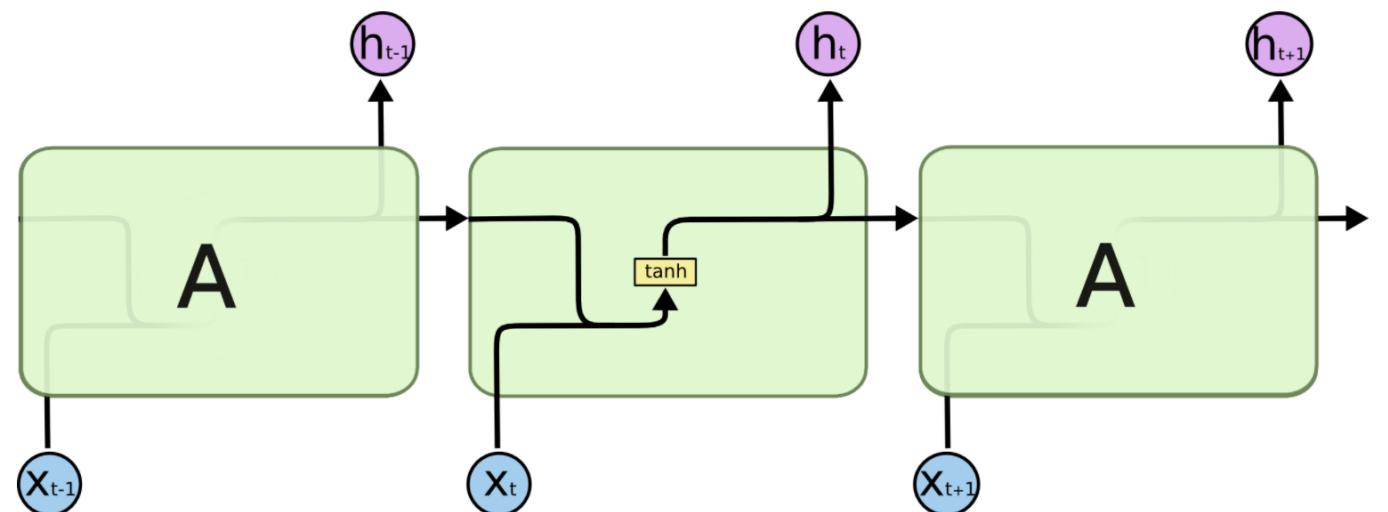
缺点

- 递归计算缓慢；
- 实际上，将前面很多步的信息完整传递是困难的。



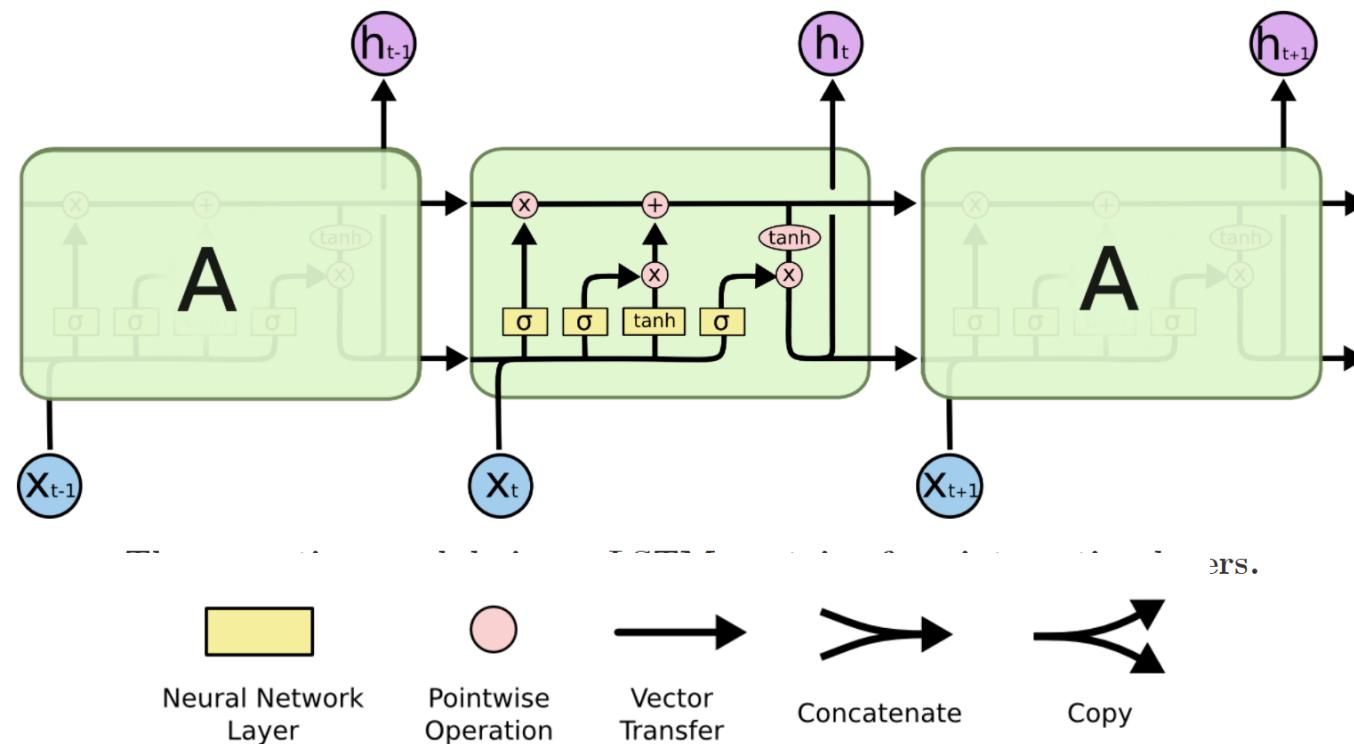
RNN Problems

- Long-term dependencies
- Vanishing gradient problem



LSTM comes to the rescue

- Long short-term memory



Summary

- NLP tasks
- “You shall know a word by the company it keeps”
- Word2Vec
 - CBOW
 - Skip-gram
- Language Model
- RNN and LSTM

Take a break!

Contents

- Introduction to NLP
- Word Embedding and RNNs
- Attention and Transformers
- Self-supervised Learning
- Pre-trained Language Models

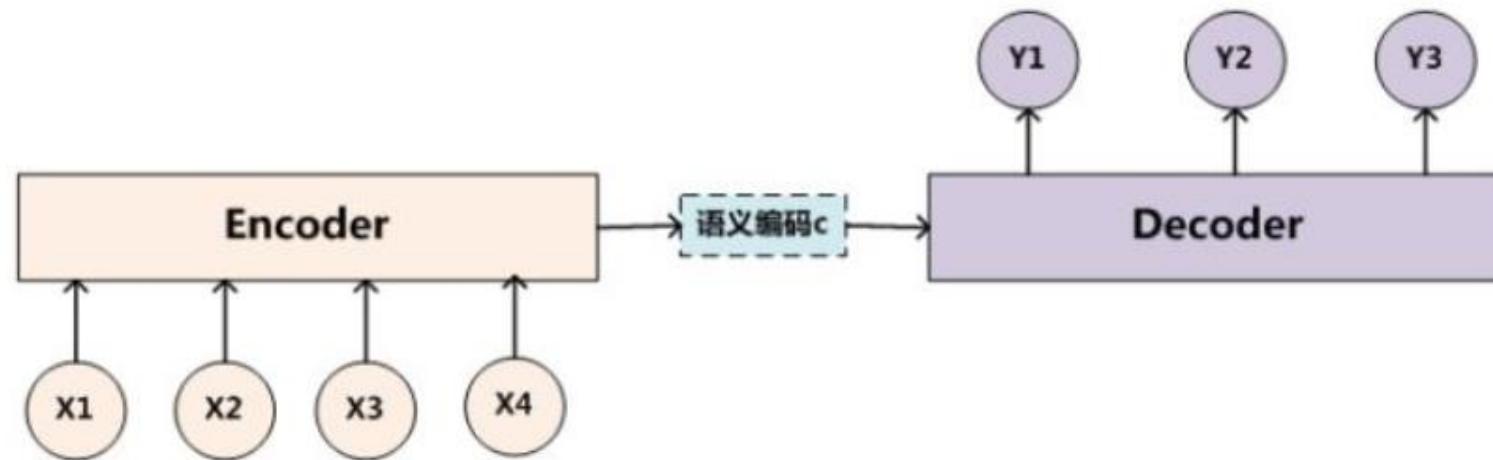
Attention



If you are not satisfied with the baby leakage guarantee, we will get your money back. Read more about our leakfree guarantee at www.baby.com

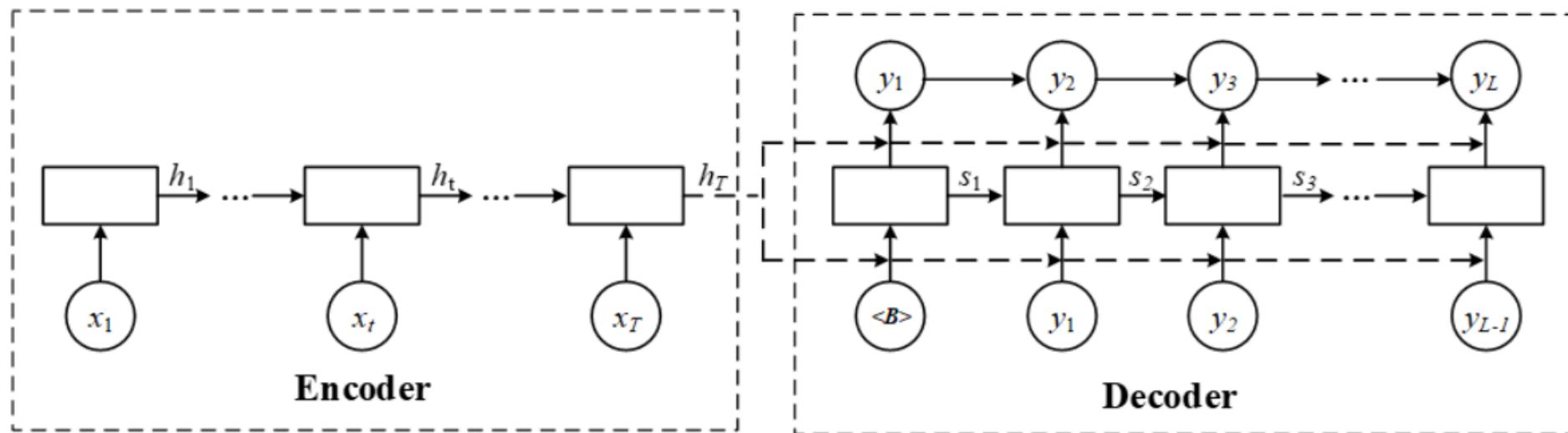
Encoder-Decoder Architecture

- Sequence-to-sequence



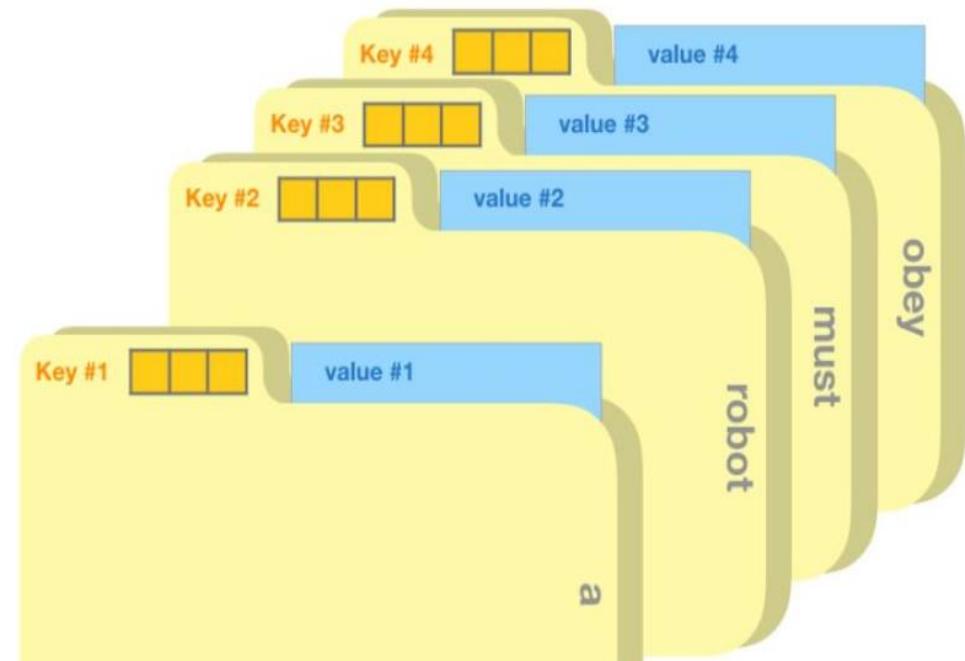
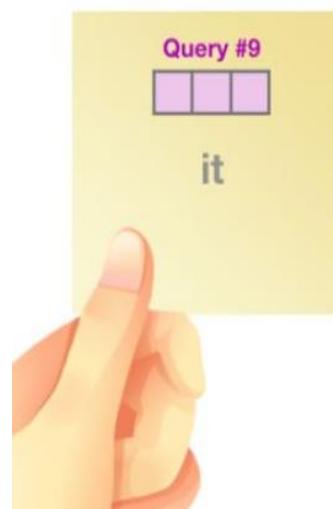
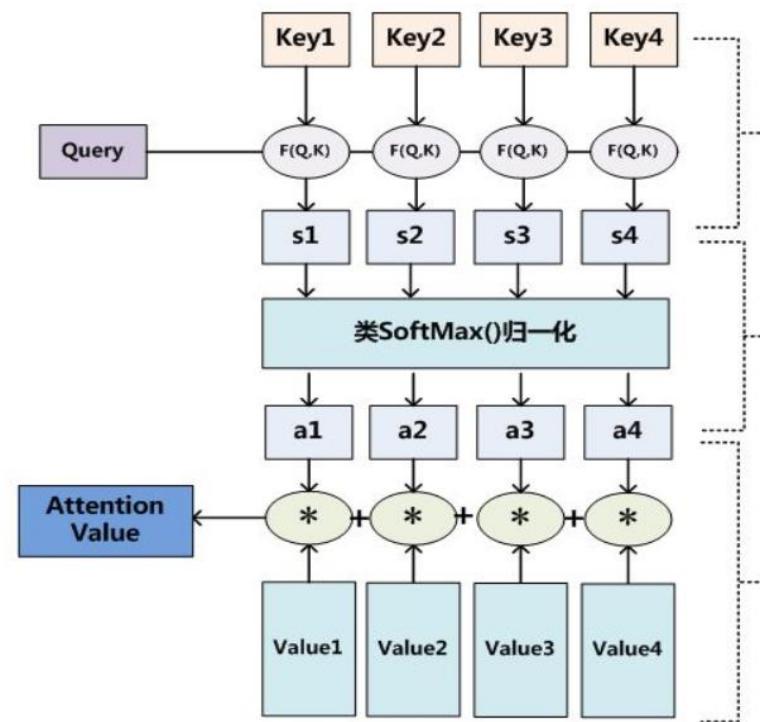
Machine Translation

- Encoder: RNN Decoder: RNN
 - What is the input of the decoder?
 - When does the output sequence stop?



QKV Attention

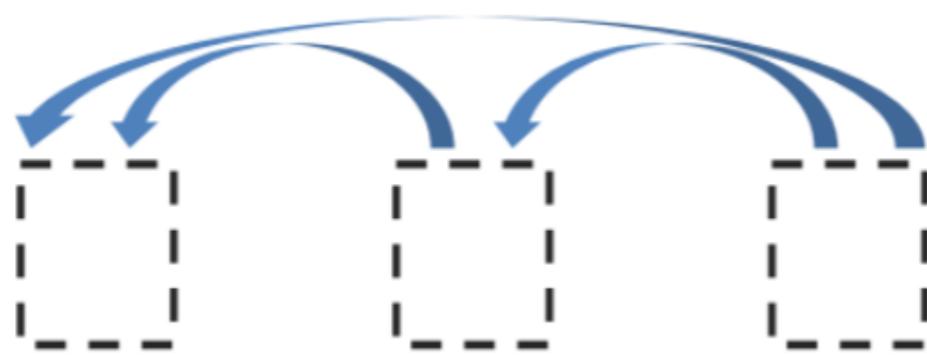
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Self-attention in NLP

- $Q=K=V$

Masked Self-attention



Multi-head attention

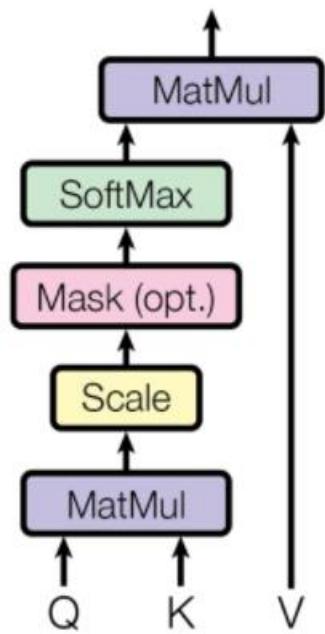
$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) \mathbf{W}^o$$

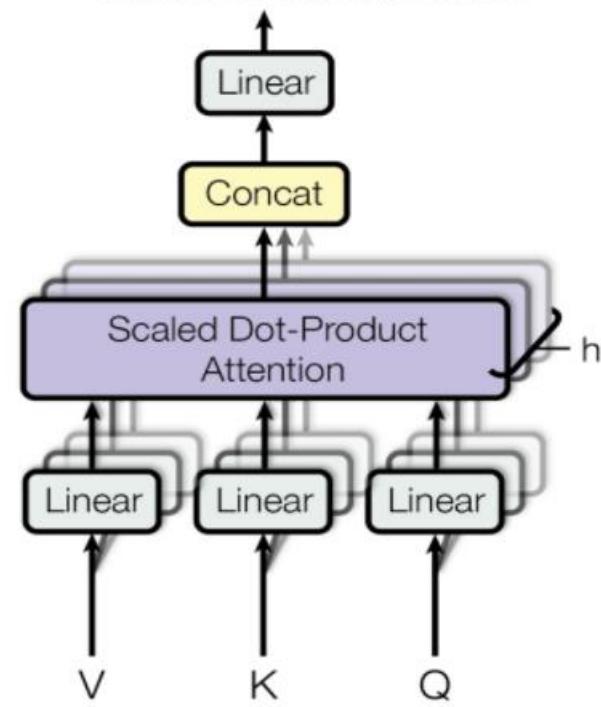
- Project onto more subspaces
- Dimension reduction

Figures

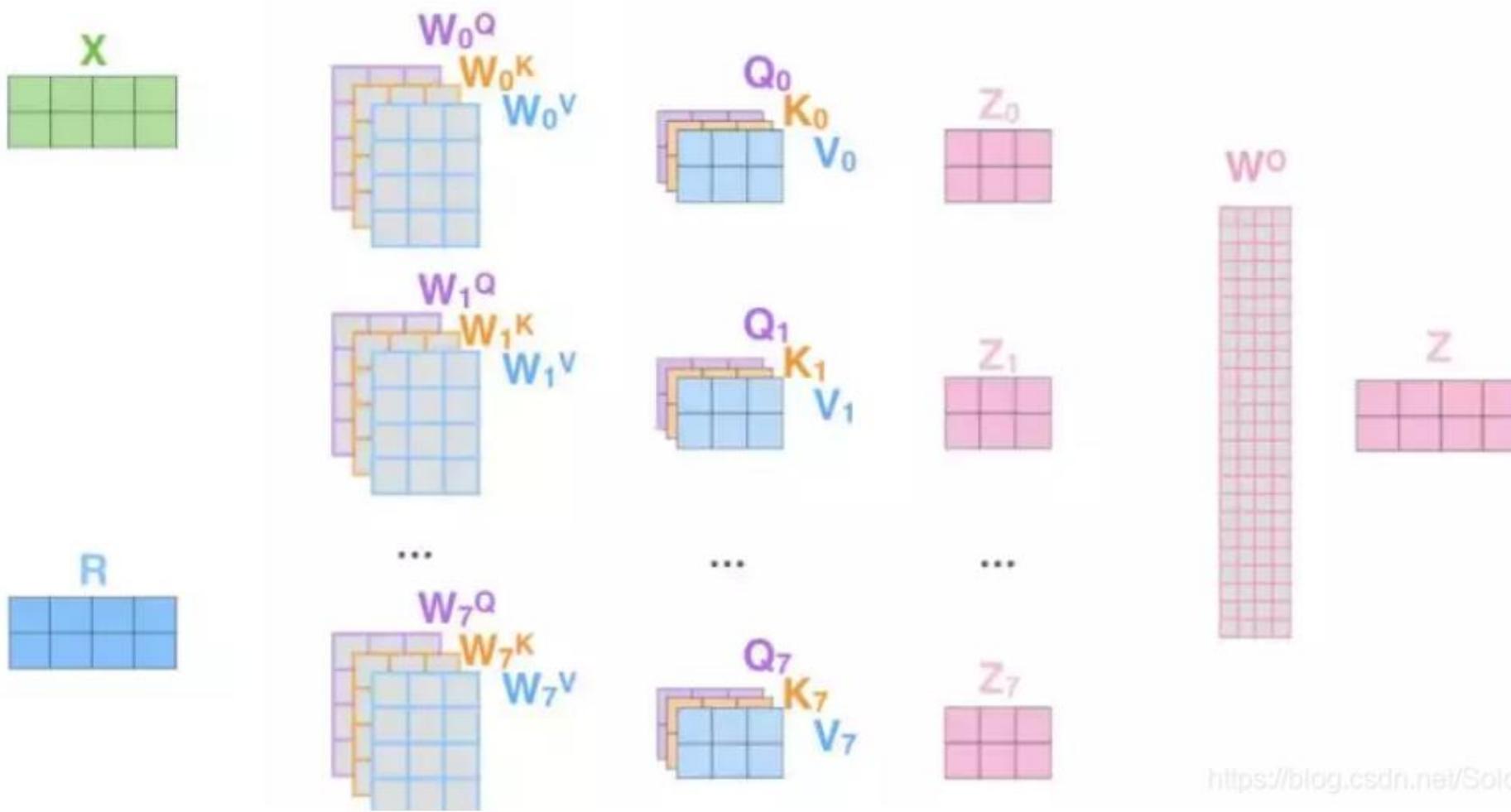
Scaled Dot-Product Attention



Multi-Head Attention



Figure



Multi-head Self-attention

- Where Q=K=V for this:

$$\text{head}_i = \text{Attention}(\text{QW}_i^Q, \text{KW}_i^K, \text{VW}_i^V)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^o$$

Figure

Attention Visualizations

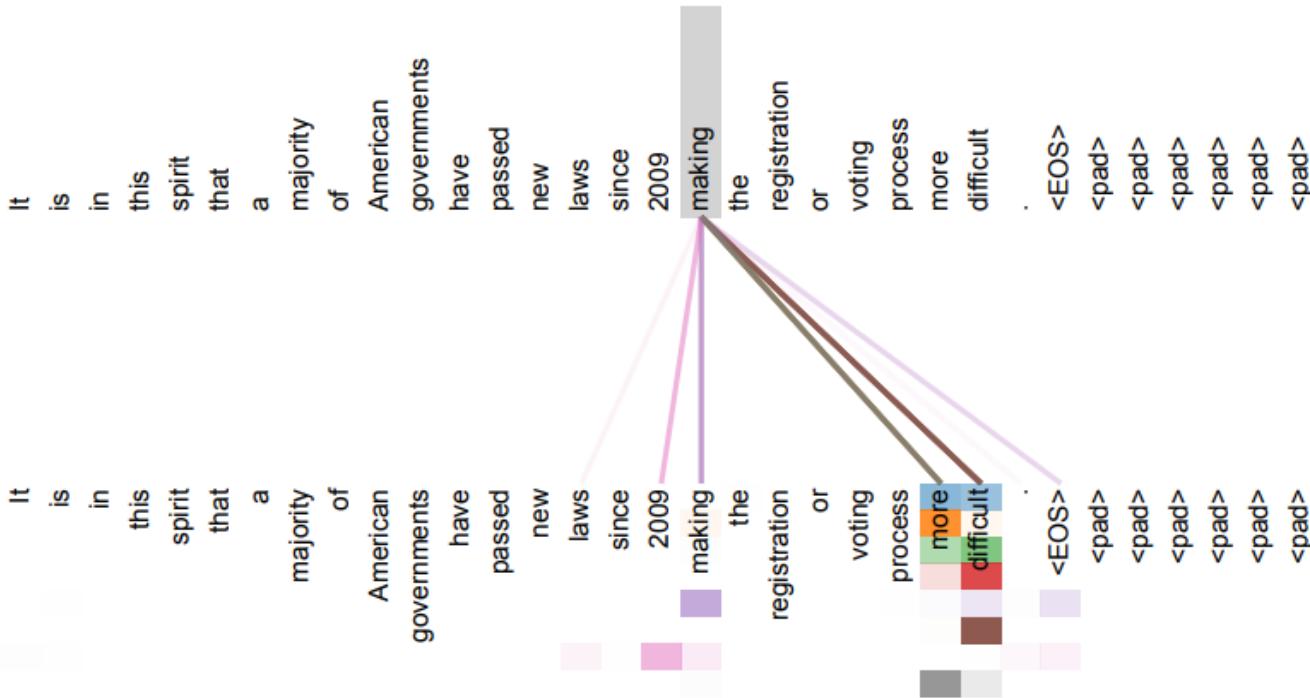
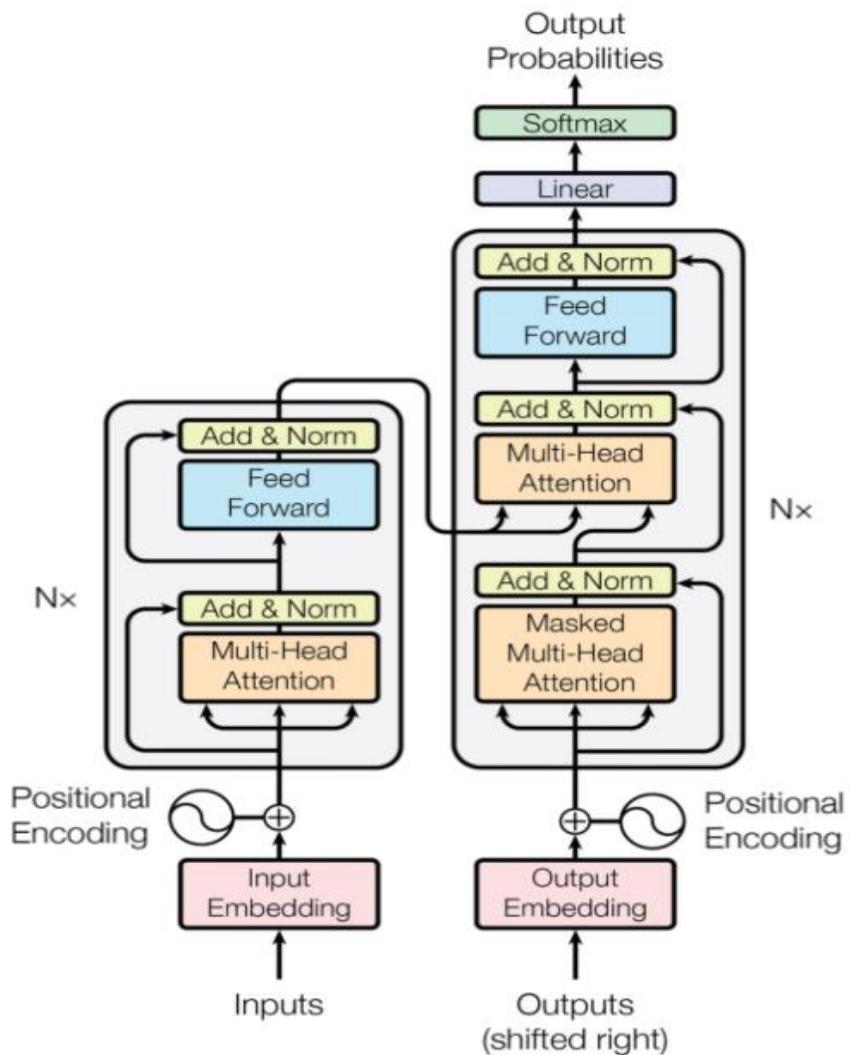


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb ‘making’, completing the phrase ‘making...more difficult’. Attentions here shown only for the word ‘making’. Different colors represent different heads. Best viewed in color.

Transformers

- “Attention is All You Need”
 - Encoder and decoder
 - Residual connection
 - Positional encoding



Summary

- QKV attention
- Self-attention and in NLP
- Multi-head attention
- transformers

Take a break!

Contents

- Introduction to NLP
- Word Embedding and RNNs
- Attention and Transformers
- **Self-supervised Learning**
- Pre-trained Language Models

Revisit: Language Model

- A language model tries to **predict the next word(token)** given the previous token sequence.

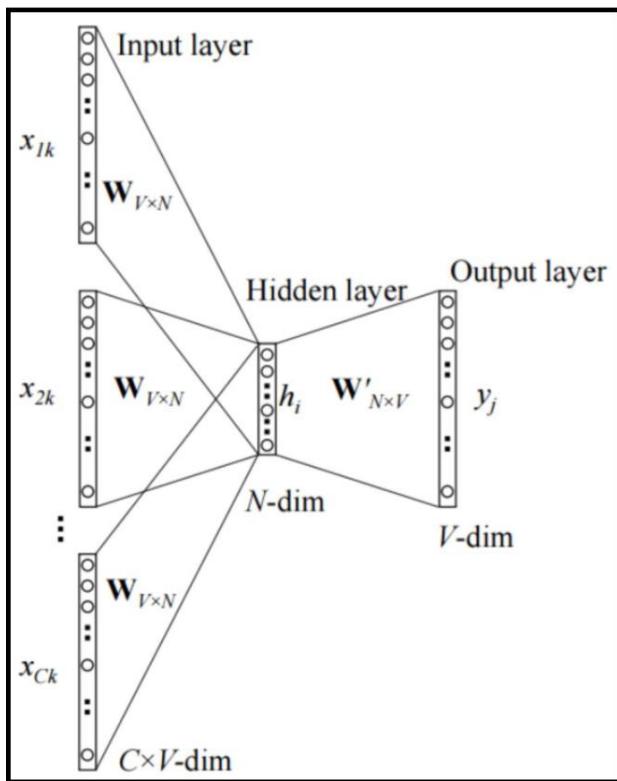
$$P(x^{(t+1)} | x^{(t)}, \dots, x^{(1)})$$

- RNN Language Model
- How to train an RNN language model?

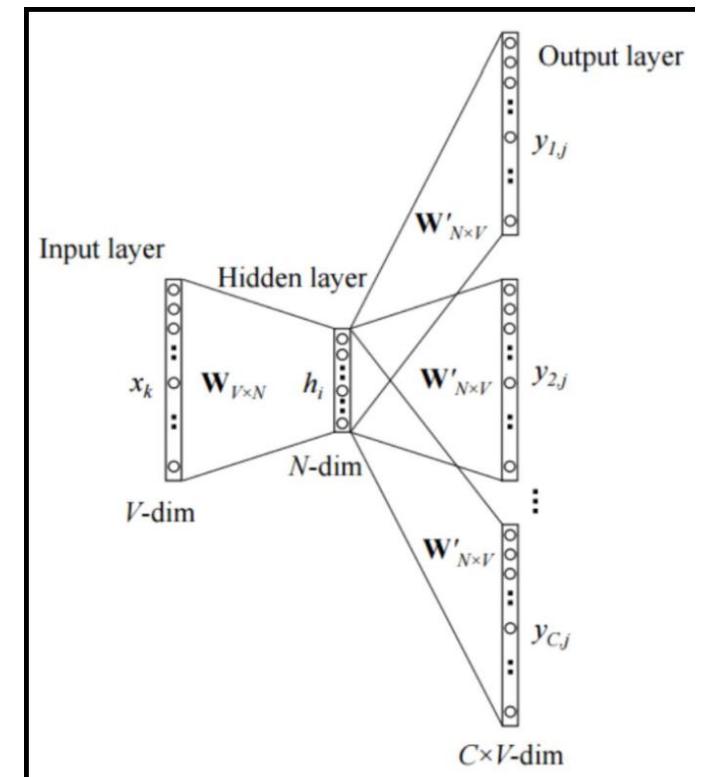
Revisit: Word2Vec

- “you shall know a word by the company it keeps”

- CBOW

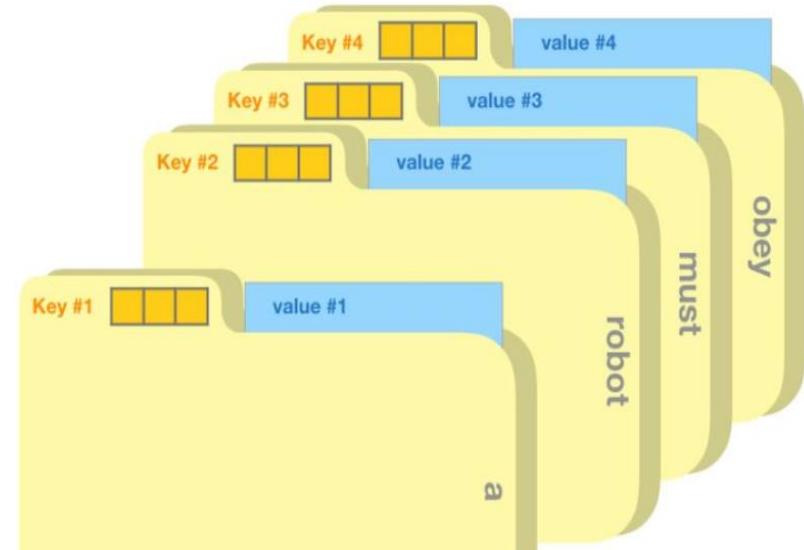
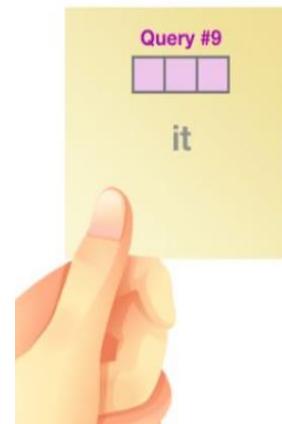


Skip-Gram



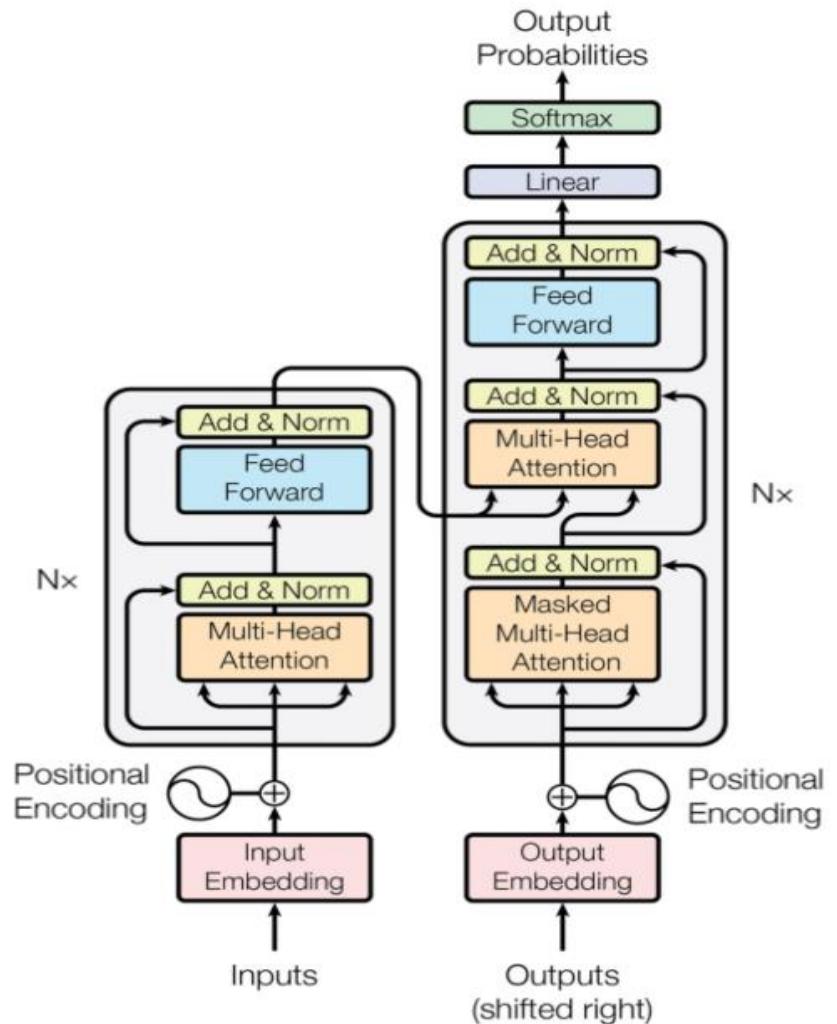
Revisit: Attention Mechanism

- Query-Key-Value Framework
- Self-attention
 - $Q=K=V$
- Masked self-attention
 - Only K V before the current token
- Multi-head attention
 - Just like multiple convolution kernels



Revisit: Transformer

- Positional encoding
 - Why and how
- Residual connection
- Input & output
- Why is it faster than RNN?
- Multi-layer transformer encoder?



Motivation

- Data annotation has always been the soft spot...
- Few-shot learning
- Unsupervised learning
- Semi-supervised learning
- Self-supervised learning

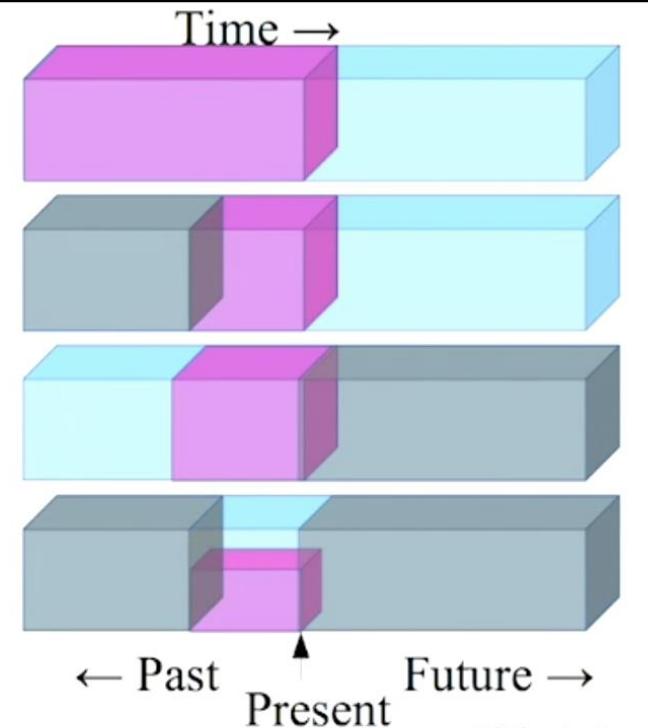
Definition

- Framing a supervised learning task in a special form where **predict only a subset of information using the rest**
- In which way, both inputs and labels are provided by the dataset

But

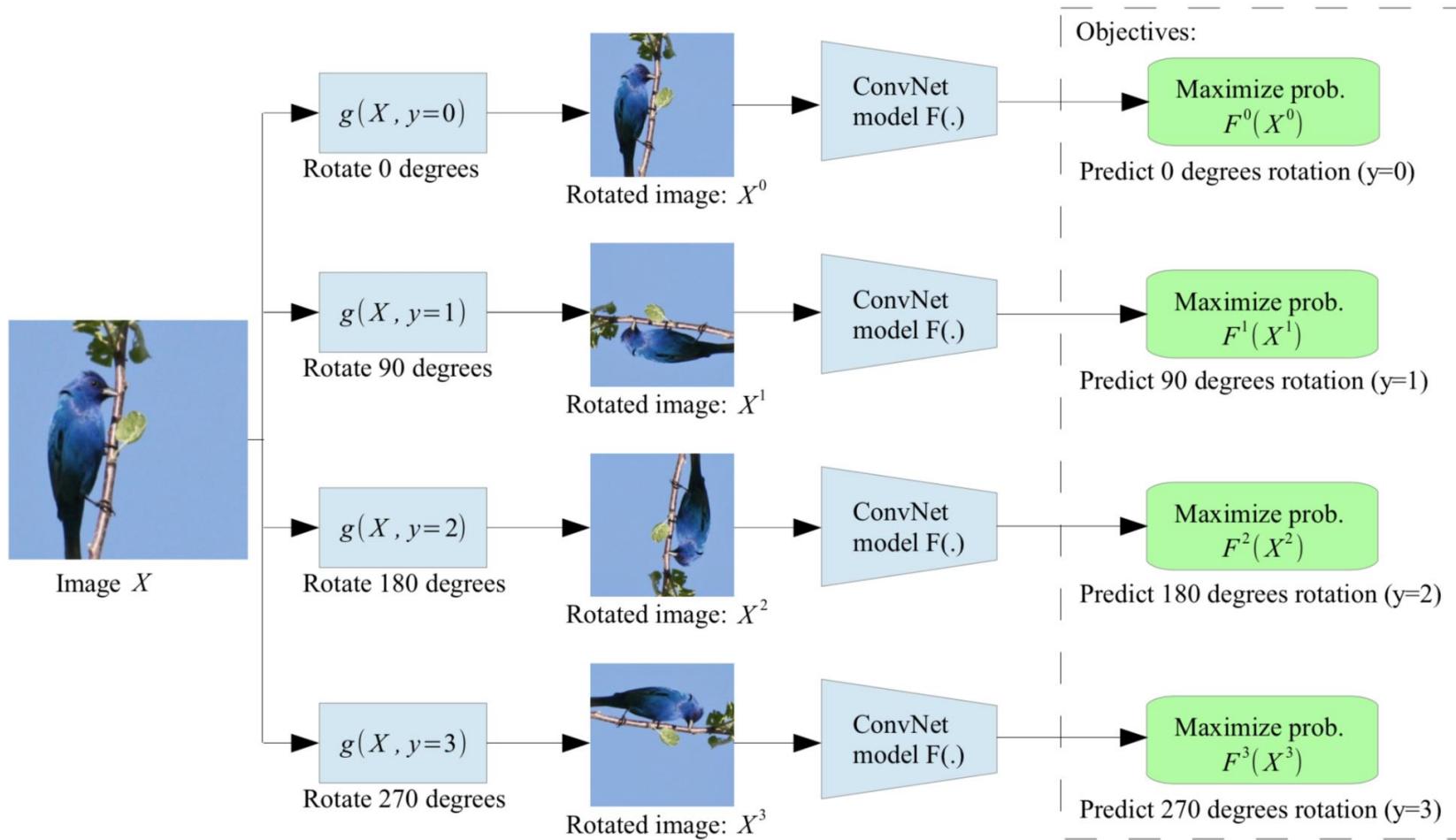
- How do we decide which to predict?

- ▶ Predict any part of the input from any other part.
- ▶ Predict the **future** from the **past**.
- ▶ Predict the **future** from the **recent past**.
- ▶ Predict the **past** from the **present**.
- ▶ Predict the **top** from the **bottom**.
- ▶ Predict the **occluded** from the **visible**
- ▶ **Pretend there is a part of the input you don't know and predict that.**



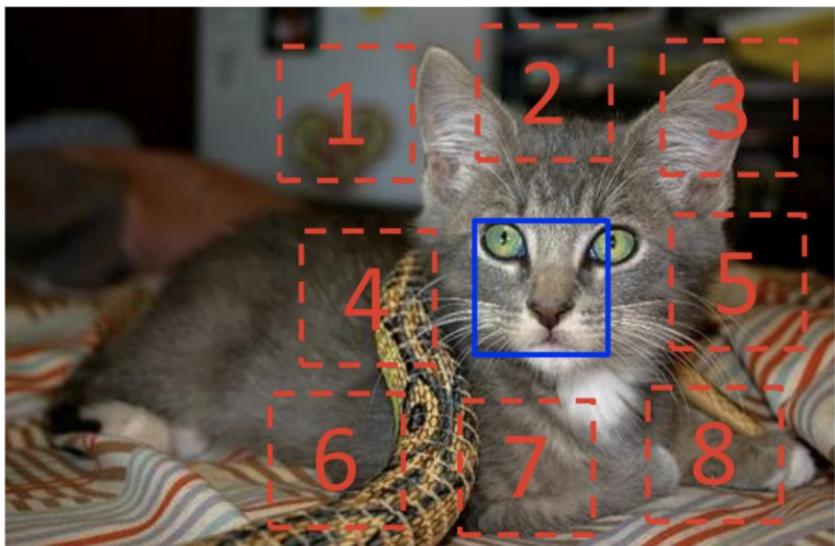
Slide: LeCun

Image: Rotation



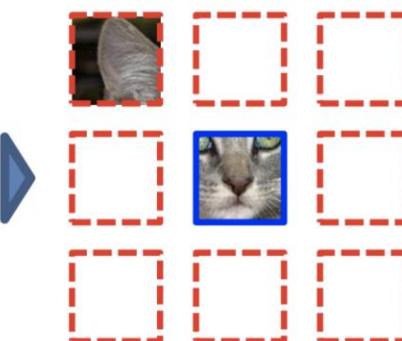
- Learn image semantics in the process

Image: Patch



$$X = (\text{cat eye}, \text{ear}) ; Y = 3$$

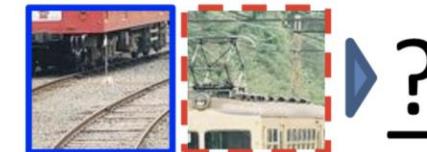
Example:



Question 1:

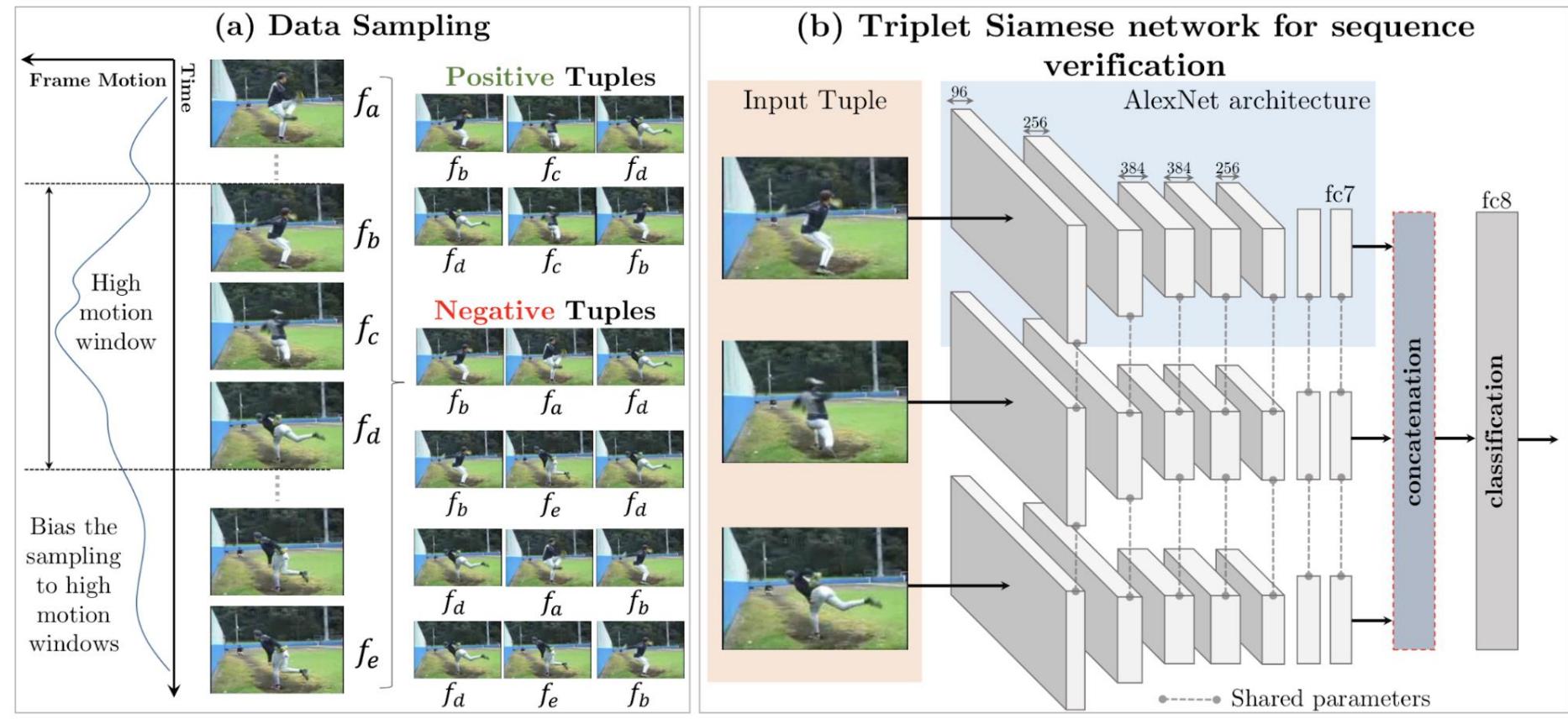


Question 2:



- Gap between patches is crucial...

Video: Frame Sequence



NLP

- Predict next word of a sentence
 - Language Model
- Predict words within the context window
 - word2vec

Why self-supervision works

- In the process of solving auxiliary tasks, the network learns **representations** of inputs
- that could be utilized to solve downstream tasks
- **Representation Learning**
 - In comparison to **feature engineering**
 - A system automatically discover features (e.g. word embedding)
- ICLR

However

- The self-supervised task is not my objective!
- Self-supervised applications serve as a pre-training method, that captures semantic/context/correlation, learns **generic knowledge**
- Fine-tuning for the downstream tasks
- Which brings to transfer learning, specifically pre-train and fine-tune

Motivation

- Train a network from scratch?
 - Limited data, especially supervised
 - Limited computational resources (maybe u don't feel it now lol)
- Observation: tasks that are similar in context
 - image classification, HUGE dataset and lots of efforts
 - Image style transfer (e.g. horse 2 zebra)
 - Is there any supervised data at all?
 - Same context: understanding images



Transfer Learning

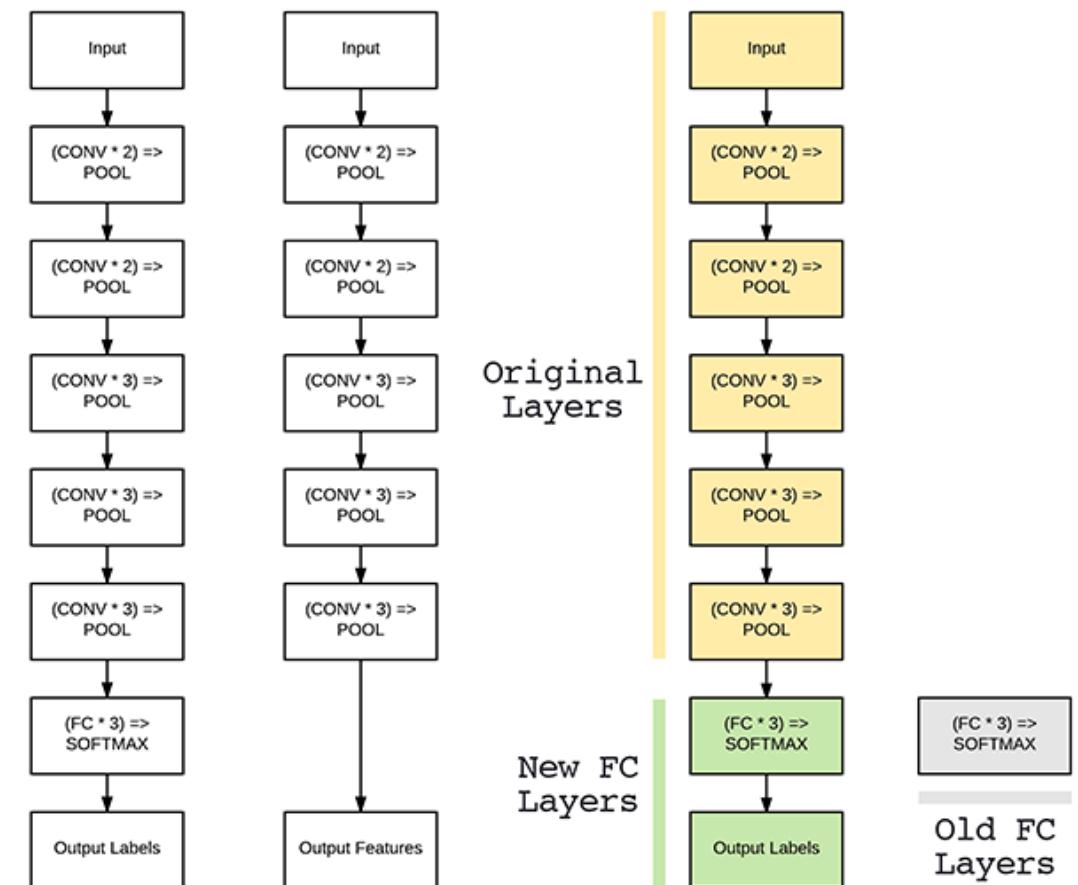
- Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.
- Often,
 - The first task is rich in data(supervised in CV & self-supervised in NLP)
 - The first model is computationally expensive
 - The second task lacks data or accurate annotation
 - Both tasks are similar in underlying generic knowledge

TL: Feature Extraction Approach

- Extract the data representation from the first approach as the input for the second model, keeps the first model param unchanged
- For instance
 - Word2vec in NLP
- Question: Are features discovered in the first task exactly what we want in solving the second task?

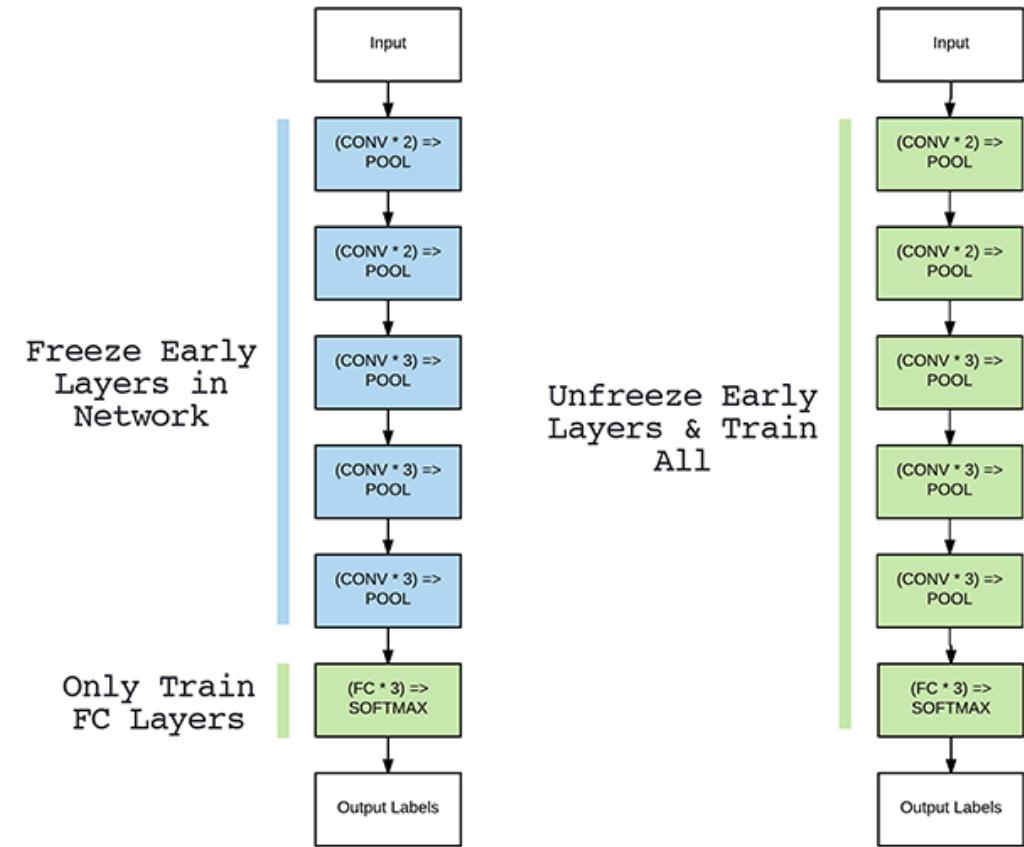
TL: Fine-Tuning Approach

- #1 Architecture Change
- MOST COMMON APPROACH
 - Truncate the last softmax layer
 - and replace it with a new one
- Retain original layer's params



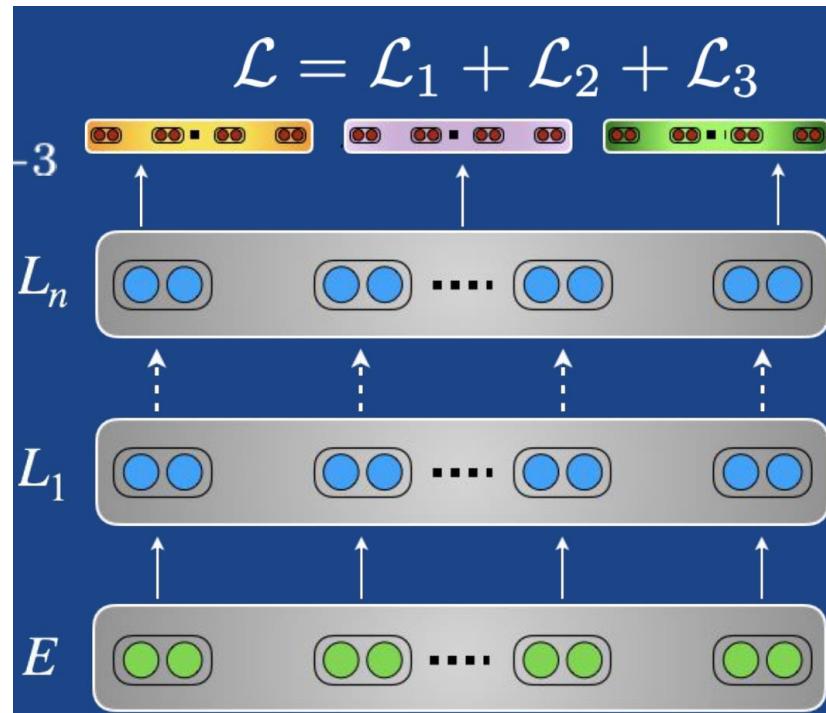
TL: Fine-Tuning Approach

- #2 Optimization Scheme: Shallow vs Deep FT
- Tricks in deep FT
 - Reduced learning rate
 - Gradual unfreezing (direction?)
 - Tougher regularization
 - warmup

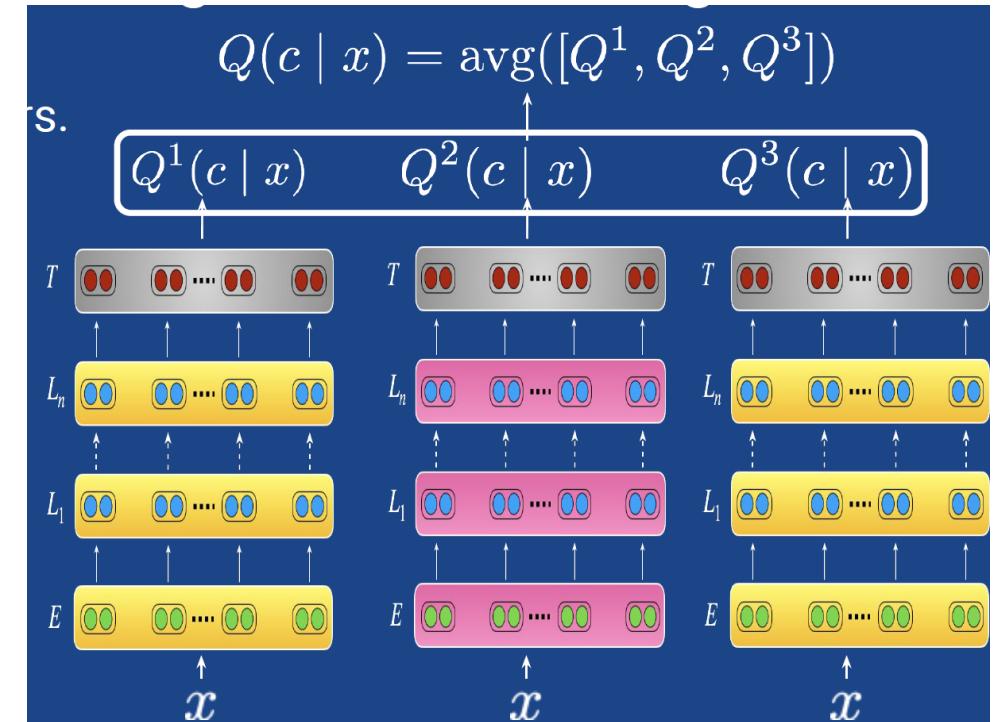


TL: Fine-Tuning Approach

- #3 Getting more signal
- Multi-task fine-tuning



Ensembling



Takeaway till now

- Self-supervised learning
- Transfer learning
- Pre-training
- Fine-tuning

Contents

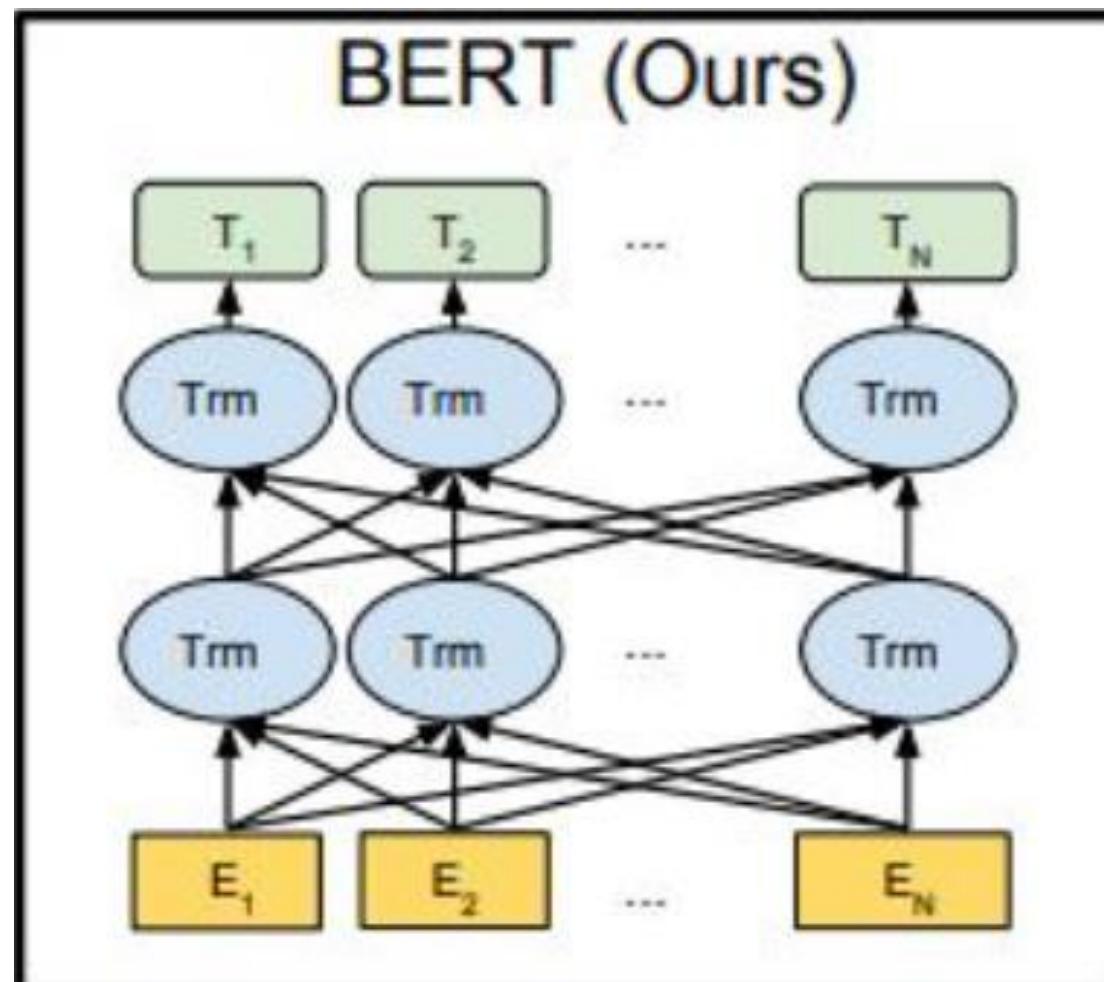
- Introduction to NLP
- Word Embedding and RNNs
- Attention and Transformers
- Self-supervised Learning
- Pre-trained Language Models

BERT

- Bidirectional Encoder Representation from Transformers
- After transformers, another shocking breakthrough from Google to the NLP community
- A self-supervised pre-training Language Model
 - Architecture?
 - Self-supervised pre-training tasks?
 - How do I use it in fine-tuning downstream tasks?

BERT Architecture

- Only encoder of transformer
- Bidirectional? Undirectional
 - Read the entire sequence at once
 - The ‘context window’ is total



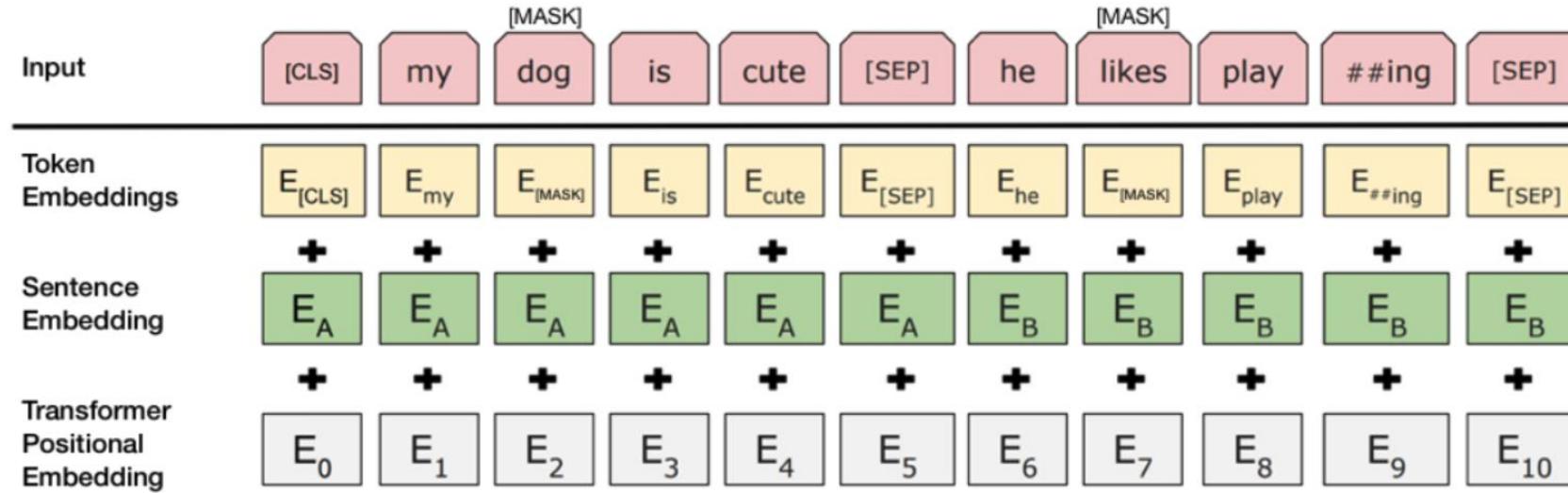
BERT Pre-training Tasks

- #1 Masked Language Model
- Pick 15% tokens and replace with
 - 80% [MASK], 10% random token, 10% the original one
 - Why not all [MASK]? [MASK] is not in fine-tuning process

$$P(w_i | w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_n)$$

- #2 Next Sentence Prediction
- Each sequence is a concatenation of two sentences A & B
 - 50% B follows A
 - 50% B is a random sentence in the corpus
 - Binary classification

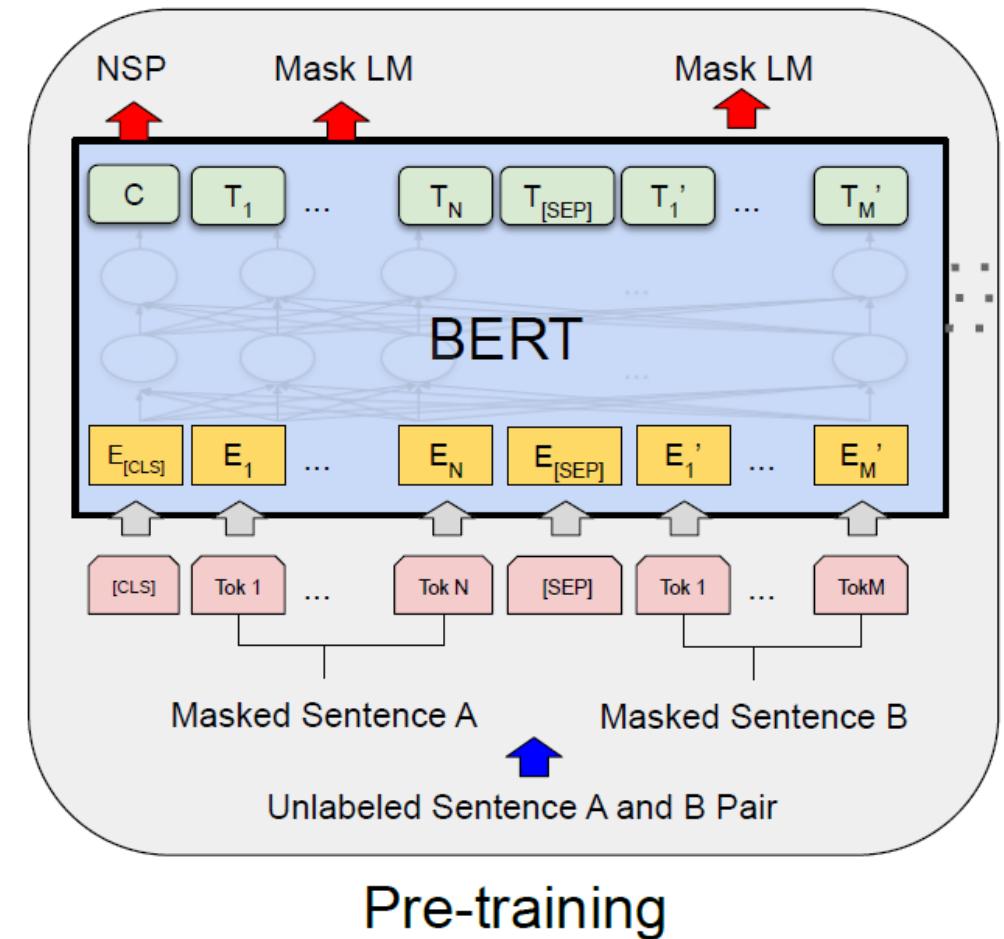
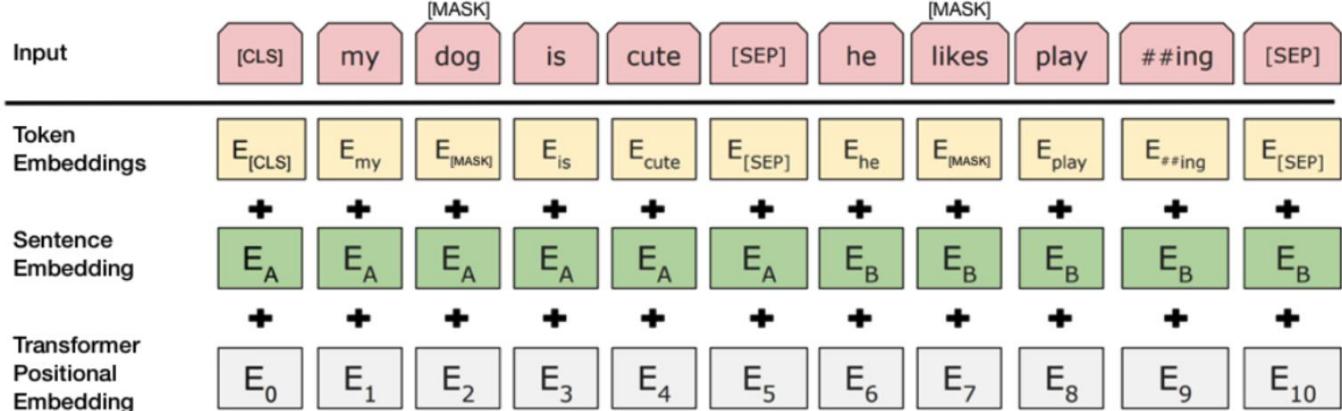
Joint Pre-training Input



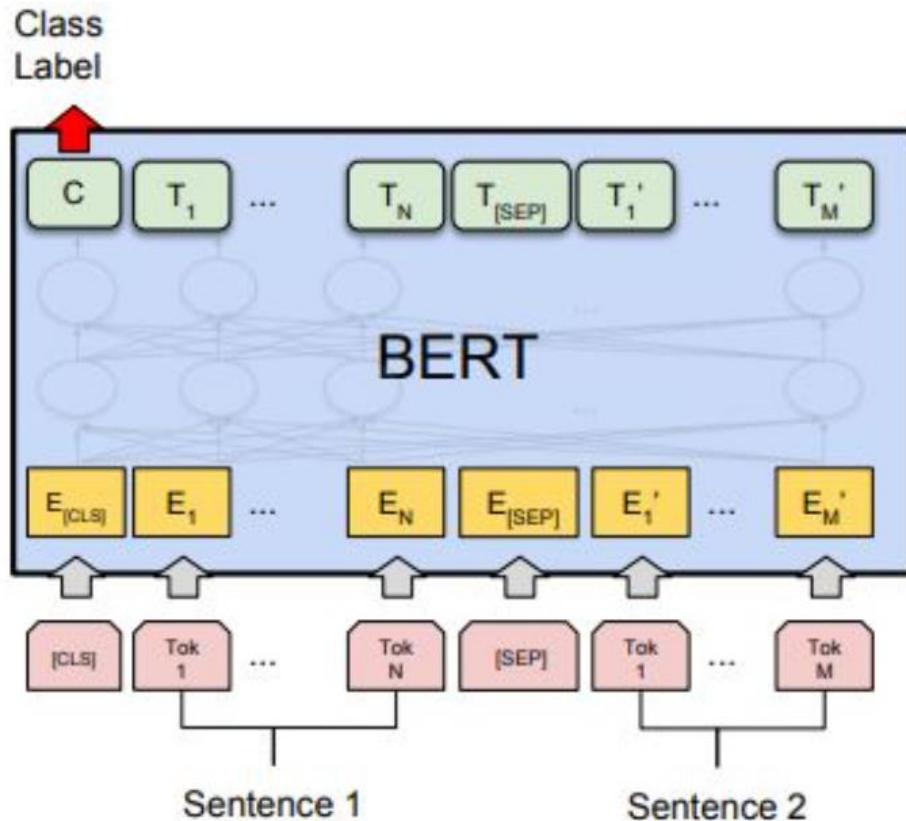
- [CLS]: prompter for next sentence prediction task
- [SEP]: sentence separator
- [MASK]: this word is masked due to Masked LM task

Joint Pre-training

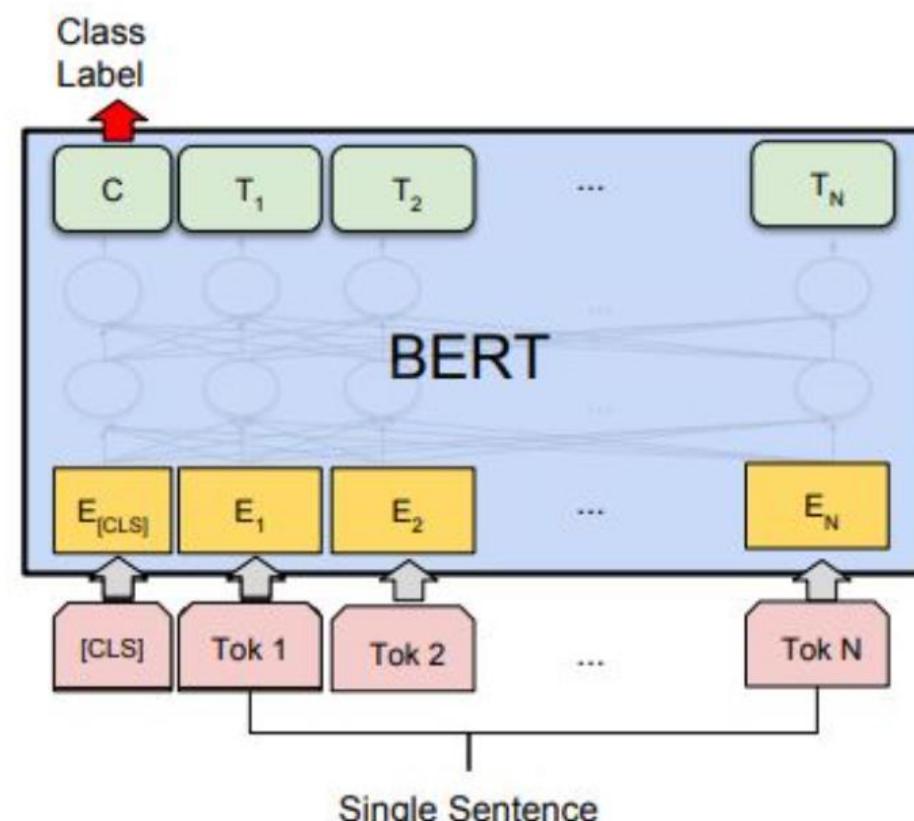
- Next sentence prediction
 - Softmax @ [CLS] token
- Masked Language Model
 - Softmax @ any [MASK] token



Fine-tuning on Downstream NLP Tasks

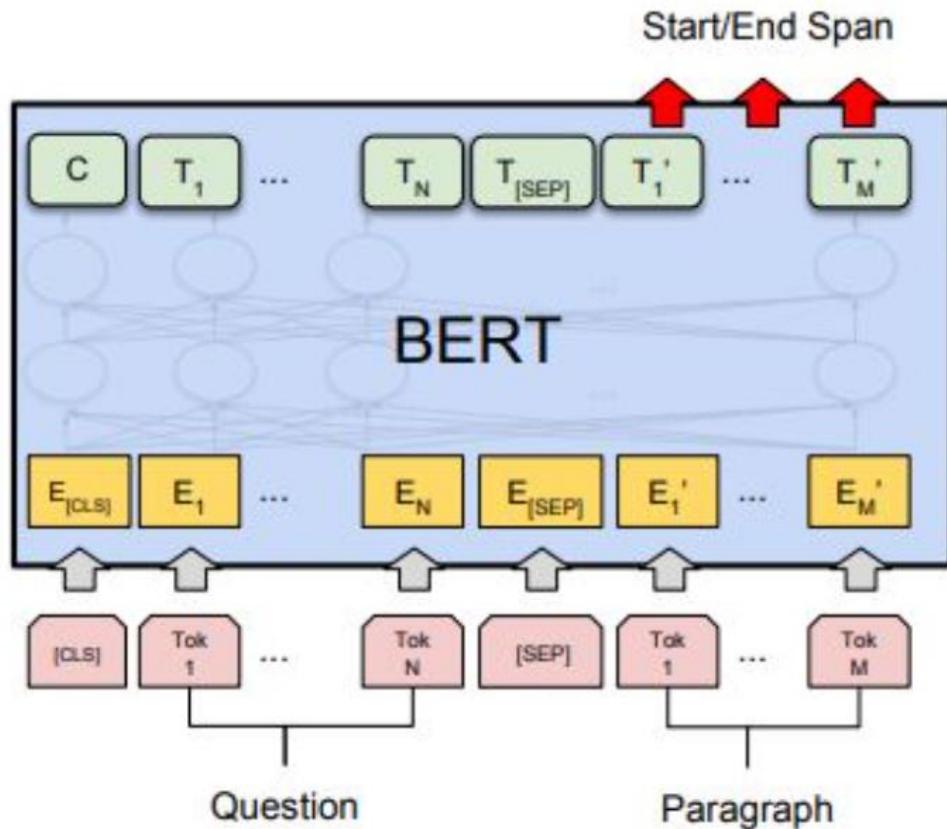


(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

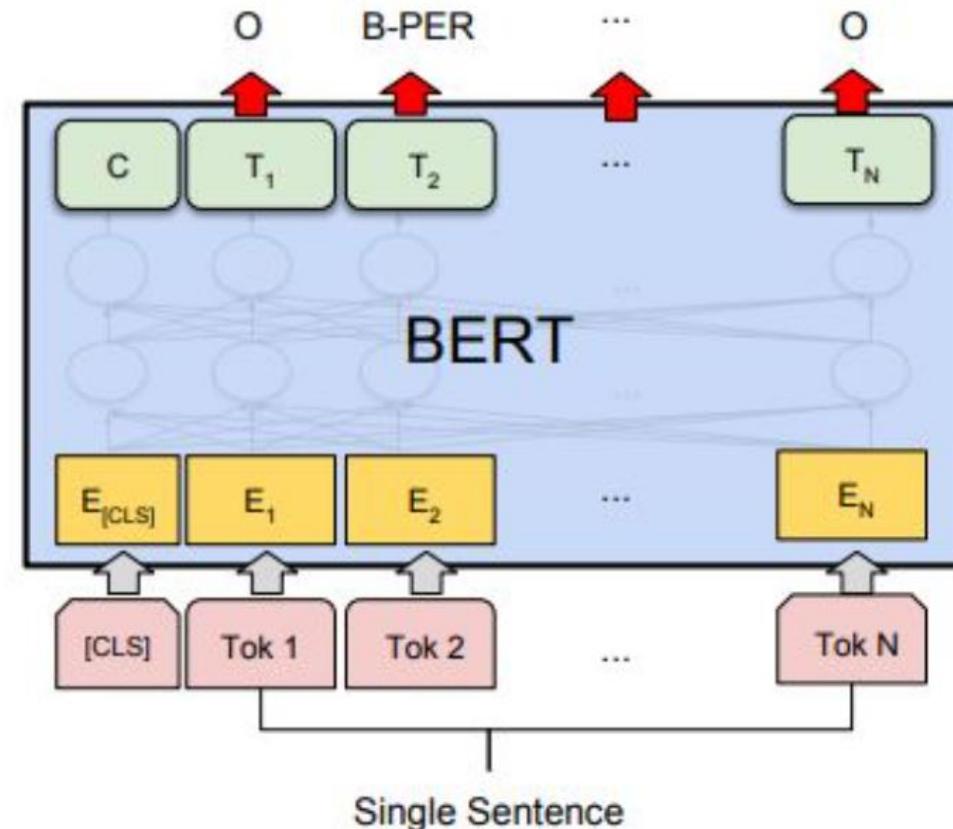


(b) Single Sentence Classification Tasks:
SST-2, CoLA

Fine-tuning on Downstream NLP Tasks



(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

BERT Contributions

- #1 A deep model could also succeed in NLP
- #2 Representation Learning will be ubiquitous in NLP
 - Word2vec: one word, one embedding
 - BERT & more: context-aware embedding
- #3 Pre-training & Fine-tuning schema is substantiated
 - In fact, use pre-trained models whenever u can!

RoBERTa

- Robustly Optimized BERT Pre-training Approach
 - Shorter pre-train time, better downstream performance
- Change #1: Remove Next Sentence Prediction
 - Better performance on downstream tasks
- Change #2: Bigger batch sizes & longer sequences
 - Augment parallelization & end-task acc
- Change #3: Dynamically change [MASK] strategy
 - Re[MASK] every 4 epochs, 10 different [MASK]ed seqs in 40 epochs

GPT-3

- “Language Models are Few-shot Learners” by OpenAI
- 175 billion parameters, 12million \$ for pre-training once
- Outperforms on Cloze, Q&A, Translation, Coreference Resolution, Reasoning, Reading Comprehension, Standard Test,...
- Few-shot, one-shot & zero-shot nature further substantiates the efficacy of pre-training to capture generic knowledge in text

Summary

- Self-supervised Learning
 - Learn to predict part of itself using the rest
- Pre-training & Fine-tuning
 - Pre-train captures generic knowledge, fine-tune for downstream
 - Architecture, Optimization tricks & More Signal
- BERT
 - Architecture, Pre-Training Tasks, how to pre-train and fine-tune
- RoBERTa & GPT-3

Project

- Reproduce results of the paper
 - “BotRGCN: Twitter Bot Detection with Relational Graph Convolutional Networks”
- Huggingface transformers github, pipeline, for NLP part
- Torch geometric for GNN part
- The template.py framework
- Please seek <https://github.com/BunsenFeng/BotRGCN> for reference.

Thx for Attention

Shangbin Feng, Xi'an Jiaotong University

wind_binteng@stu.xjtu.edu.cn

October 7, 2021