

高等机器学习

自然语言处理



张卫强



清华大学
Tsinghua University

What is Natural Language Processing (NLP)

- The process of computer analysis of input provided in a human language (natural language), and conversion of this input into a useful form of representation.
- The field of NLP is primarily concerned with getting computers to perform useful and/or interesting tasks with human languages.
- The field of NLP is secondarily concerned with helping us come to a better understanding of human language.

Forms of natural language

- The input/output of an NLP system can be:
 - written **text** (the focus of this class)
 - **speech**
- To process written text, we need:
 - **lexical, syntactic, semantic** knowledge about the language
 - **discourse** information, real world **knowledge**
- To process spoken language, we need everything required to process written text, plus the challenges of **speech recognition** and **speech synthesis**.

Components of NLP

- Natural Language Understanding (NLU)
 - Mapping the given input in the natural language into a useful representation.
 - Different level of analysis required:
 - morphological* (形态) *analysis*,
 - syntactic* (句法) *analysis*,
 - semantic* (语义) *analysis*,
 - discourse* (篇章) *analysis*, ...
- Natural Language Generation (NLG)
 - Producing output in the natural language (e.g., from some internal representation)
 - Different level of synthesis required:
 - deep planning* (what to say),
 - syntactic generation*

Why NL understanding is hard?

- Natural language is extremely rich in form and structure, and very ambiguous.
 - How to represent meaning,
 - Which structures map to which meaning structures.
- One input can mean many different things. Ambiguity can be at different levels.
 - Lexical (word level) ambiguity -- different meanings of words
 - Syntactic ambiguity -- different ways to parse the sentence
 - Interpreting partial information -- how to interpret pronouns
 - Contextual information -- context of the sentence may affect the meaning of that sentence.
- Many inputs can mean the same thing.

Examples

- 南京市长江大桥
 - 南京市长/江大桥
 - 南京市/长江大桥
- 新学生宿舍
 - 新/学生宿舍
 - 新学生/宿舍
- 他想起来了
 - 他/想起来了
 - 他想/起来了
- 这个人谁也没问过
 - 这个人/谁也没问过 (他)
 - 这个人 (他) /谁也没问过
- 这份报告, 我写不好
 - 这份报告, 我写/不好
 - 这份报告, 我/写不好
- I made her duck.
 - I cooked duck for her.
 - I cooked duck belonging to her.

Knowledge of language

- Phonology (音系) – concerns how words are related to the sounds that realize them.
- Morphology (词法) – concerns how words are constructed from more basic meaning units called morphemes (词素). A morpheme is the primitive unit of meaning in a language.
- Syntax (句法) – concerns how can be put together to form correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of other phrases.
- Semantics (语义) – concerns what words mean and how these meaning combine in sentences to form sentence meaning.

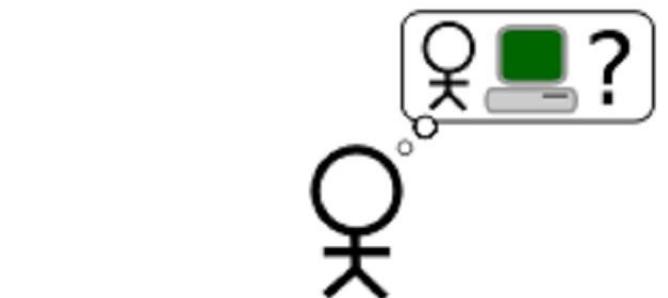
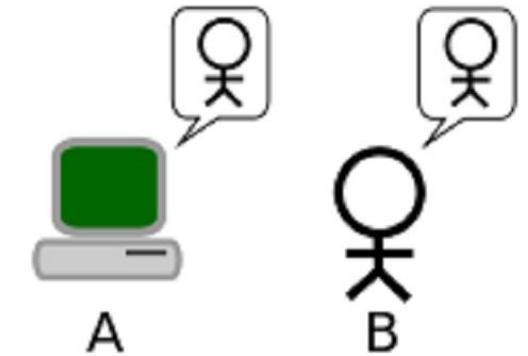
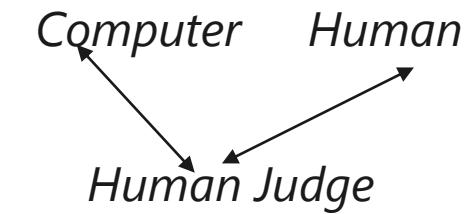
Knowledge of language (cont.)

- Pragmatics (语用学) – concerns how sentences are used in different situations and how use affects the interpretation of the sentence.
- Discourse (语篇) – concerns how the immediately preceding sentences affect the interpretation of the next sentence. For example, interpreting pronouns and interpreting the temporal aspects of the information.
- World Knowledge – includes general knowledge about the world. What each language user must know about the other's beliefs and goals.

Turing Test

Language and intelligence

- Human Judge asks tele-typed questions to Computer and Human.
- Computer's job is to act like a human
- Human's job is to convince Judge that he is not machine
- Computer is judged "intelligent" if it can fool the judge
- Judgment of intelligence is linked to appropriate answers to questions from the system.



NLP - an inter-disciplinary Field

- NLP borrows techniques and insights from several disciplines.
- Linguistics: How do words form phrases and sentences? What constraints the possible meaning for a sentence?
- Computational Linguistics: How is the structure of sentences identified? How can knowledge and reasoning be modeled?
- Computer Science: Algorithms for automatons, parsers.
- Psychology: What linguistic constructions are easy or difficult for people to learn to use?
- Philosophy: What is the meaning, and how do words and sentences acquire it?

Some buzz-words

- NLP – Natural Language Processing
- CL – Computational Linguistics
- SP – Speech Processing
- HLT – Human Language Technology
- NLE – Natural Language Engineering
- SNLP – Statistical Natural Language Processing
- Other Areas:
 - Speech Generation, Text Generation, Speech Understanding, Information Retrieval,
 - Dialogue Processing, Inference, Spelling Correction, Grammar Correction,
 - Text Summarization, Text Categorization,

Top Conference:
ACL
NAACL
EMNLP

Brief history of NLP

- 1940s –1950s: Foundations
 - Development of formal language theory (Chomsky, Backus, Naur, Kleene)
 - Probabilities and information theory (Shannon)
- 1957 – 1970s:
 - Use of formal grammars as basis for natural language processing (Chomsky, Kaplan)
 - Use of logic and logic based programming (Minsky, Winograd, Colmerauer, Kay)
- 1970s – 1983:
 - Probabilistic methods for early speech recognition (Jelinek, Mercer)
 - Discourse modeling (Grosz, Sidner, Hobbs)
- 1983 – 1993:
 - Finite state models (morphology) (Kaplan, Kay)
- 1993 – present:
 - Strong integration of different techniques, different areas.

A photograph of a man and a woman taking a selfie at night. They are standing outdoors, with trees and string lights visible in the background. The man is holding a black smartphone up to take the picture. Both individuals are smiling.

Focus of this class: Deep Learning for
Natural Language Processing

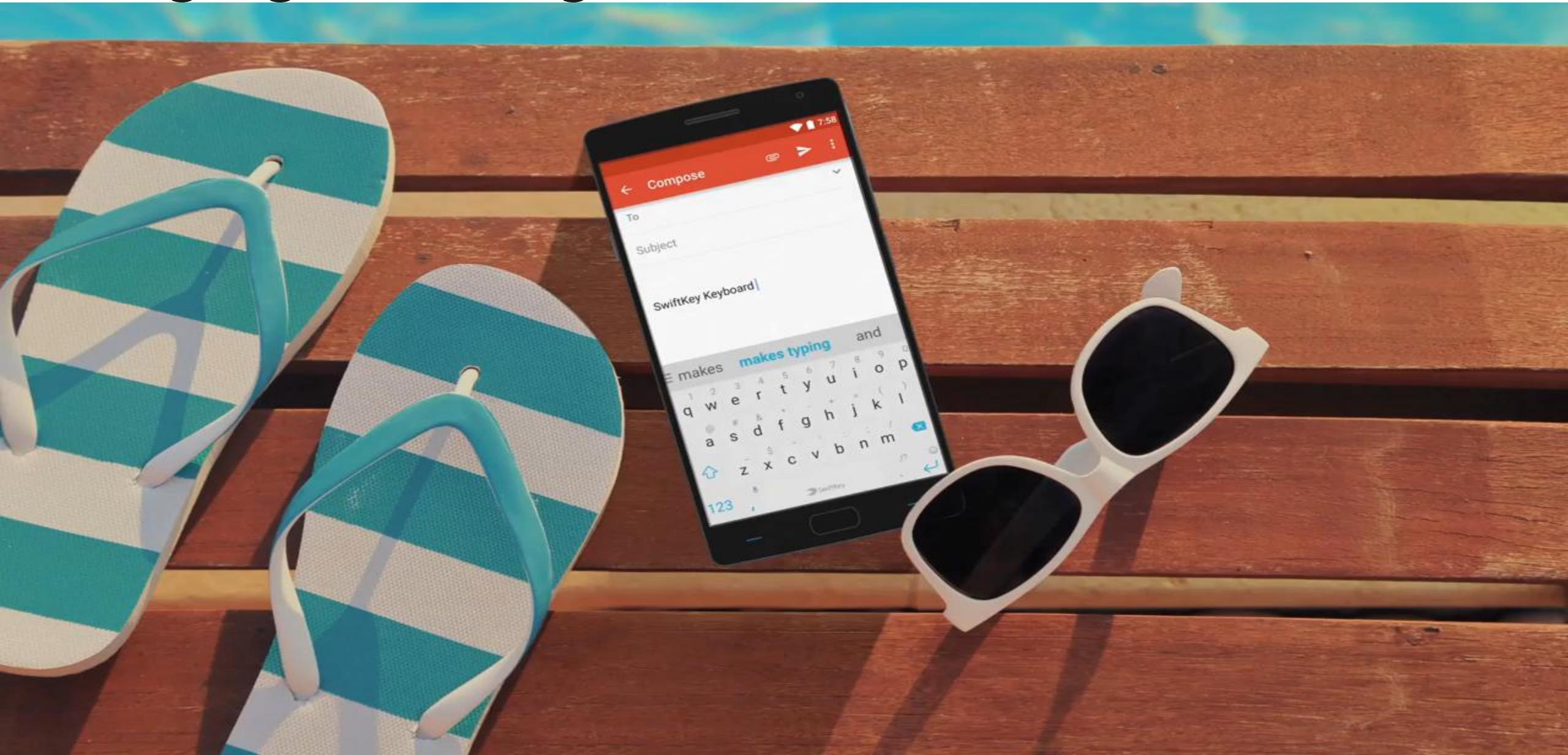
Outline

- What can NLP do today
- Word embedding
 - Word2vec, FastText
 - Probabilistic embedding, corner embedding, ...
- Model architectures (Review)
- Sequence learning
 - Sequence to sequence learning
 - Sequence to label modeling
- Pretraining
 - CoVe, ELMo, GPT, BERT, ChatGPT



Part 1: What can NLP do today?

Language modeling



Text generation

Couplet



Riddle



Poetry



Modern Poetry



Song Lyrics



Music (Lyrics & Melody)



2006

2014

2014

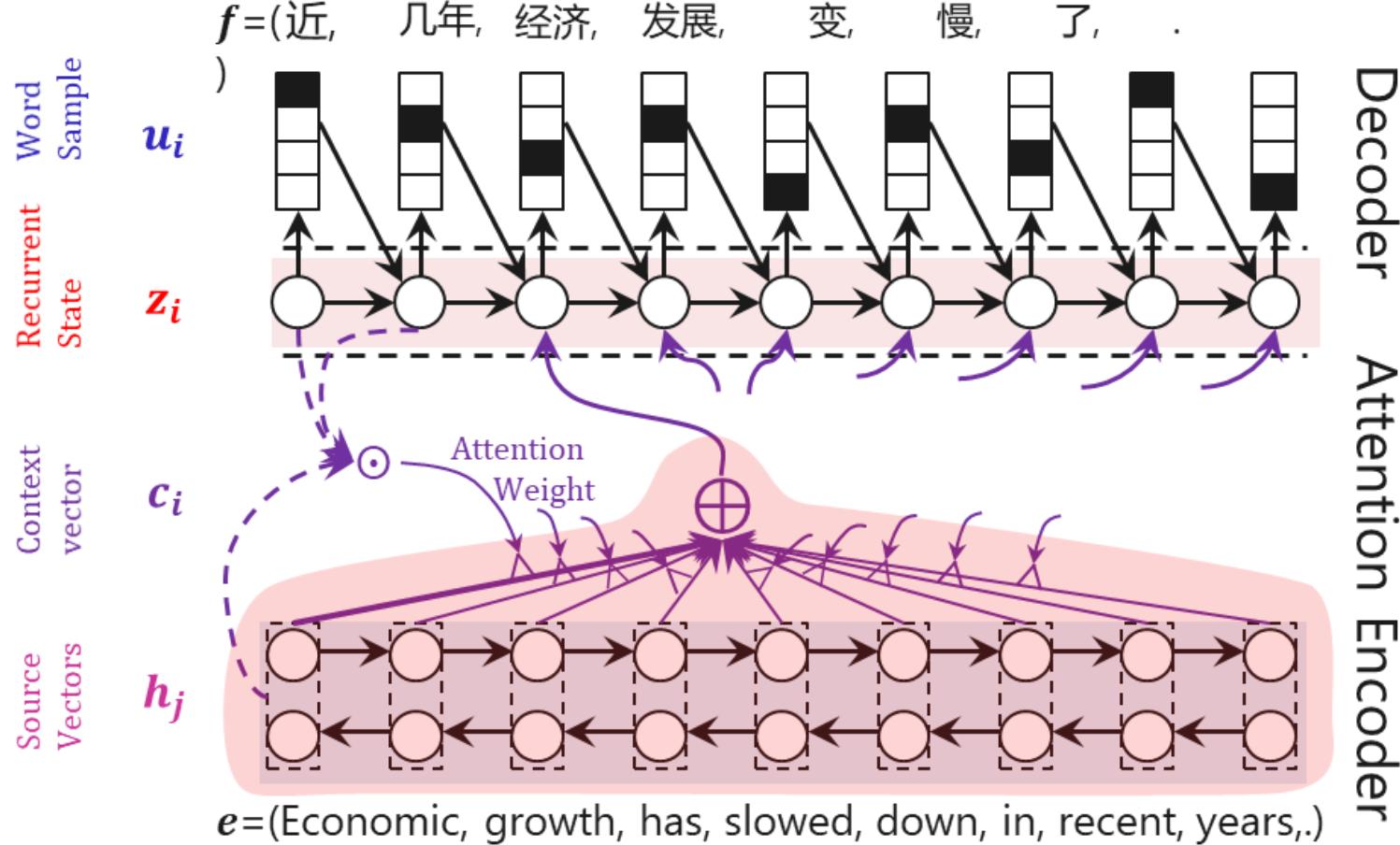
2016/2017

2016/2017

2016/2017



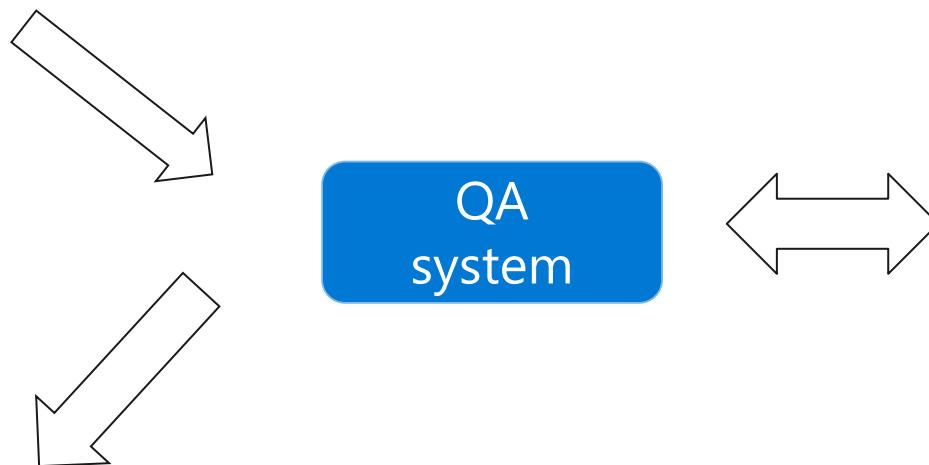
Machine translation



Question answering

Question: who is the daughter of Donald Trump?

Answer: Ivanka Trump,
Tiffany Trump



SQuAD

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called “showers”.

What causes precipitation to fall?

gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

graupel

Where do water droplets collide with ice crystals to form precipitation?

within a cloud

Passage:

In a small village in England about 150 years ago, a mail coach was standing on the street. It didn't come to that village often. People had to pay a lot to get a letter. The person who sent the letter didn't have to pay the postage, while the receiver had to. "Here's a letter for Miss Alice Brown," said the mailman.

"I'm Alice Brown," a girl of about 18 said in a low voice.

Alice looked at the envelope for a minute, and then handed it back to the mailman.

"I'm sorry I can't take it, I don't have enough money to pay it", she said.

A gentleman standing around were very sorry for her. Then he came up and paid the postage for her.

When the gentleman gave the letter to her, she said with a smile, "Thank you very much, This letter is from Tom. I'm going to marry him. He went to London to look for work. I've waited a long time for this letter, but now I don't need it, there is nothing in it."

"Really? How do you know that?" the gentleman said in surprise.

"He told me that he would put some signs on the envelope. Look, sir, this cross in the corner means that he is well and this circle means he has found work. That's good news."

The gentleman was Sir Rowland Hill. He didn't forgot Alice and her letter.

"The postage to be paid by the receiver has to be changed," he said to himself and had a good plan.

"The postage has to be much lower, what about a penny? And the person who sends the letter pays the postage. He has to buy a stamp and put it on the envelope." he said . The government accepted his plan. Then the first stamp was put out in 1840. It was called the "Penny Black". It had a picture of the Queen on it.

Questions:

1): The first postage stamp was made _

- A. in England
- B. in America
- C. by Alice
- D. in 1910

2): The girl handed the letter back to the mailman because _

- A. she didn't know whose letter it was
- B. she had no money to pay the postage
- C. she received the letter but she didn't want to open it
- D. she had already known what was written in the letter

3): We can know from Alice's words that _

- A. Tom had told her what the signs meant before leaving
- B. Alice was clever and could guess the meaning of the signs
- C. Alice had put the signs on the envelope herself
- D. Tom had put the signs as Alice had told him to

4): The idea of using stamps was thought of by _ .

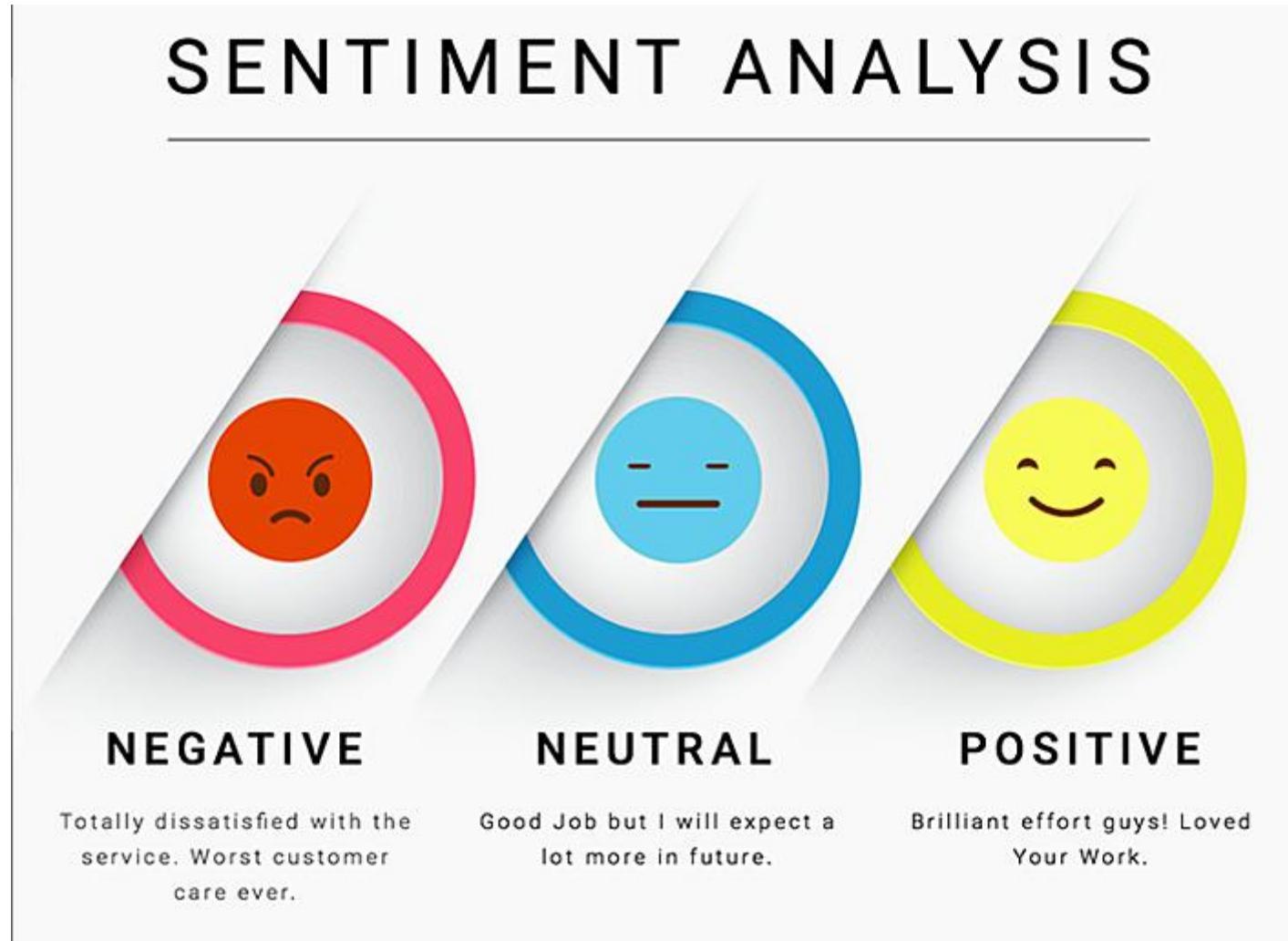
- A. the government
- B. Sir Rowland Hill
- C. Alice Brown
- D. Tom

5): From the passage we know the high postage made _ .

- A. people never send each other letters
- B. lovers almost lose every touch with each other
- C. people try their best to avoid paying it
- D. receivers refuse to pay the coming letters

Answer: ADABC

Sentiment analysis



ChatGPT

· 文献调研

H hi, good afternoon

G Good afternoon! How can I help you today? ↑ ↓

H can you help me find some paper about spiking neural network in the last 1 years ↑ ↓

G Sure! Here are a few recent papers on spiking neural networks that you may find helpful:
1. "Spiking Neural Networks with Structured Plasticity" (2021)
2. "Spiking Neural Networks for Energy-Efficient Computing" (2021)
3. "A Survey of Spiking Neural Networks: Models, Simulation and Applications" (2021)
4. "Exploring Spiking Neural Networks for Sequence Processing" (2021)
5. "Real-time Spiking Neural Network Acceleration on FPGAs" (2022)

These papers can be found in major machine learning and neural networks conference proceedings such as the International Conference on Learning Representations (ICLR) and the Conference on Neural Information Processing Systems (NeurIPS).

You can also use search engines such as Google Scholar and arXiv to find more recent papers on this topic.

· 论文写作润色助理

YM I want you to act as an academic journal editor. Please rephrase the paragraph from an academic angle based on the writing style of the Nature journal:
Various on-workstation neural-spike-based brain machine interface (BMI) systems have reached the point of in-human trials, but on-node and on-implant BMI systems are still under exploration. Such systems are constrained by the area and battery. Researchers should consider the algorithm complexity, available resources, power budgets, CMOS technologies, and the choice of platforms when designing BMI systems. However, the effect of these factors is currently still unclear.

G The advancements in on-workstation neural-spike-based brain machine interface (BMI) systems have facilitated their evaluation in human trials. However, the exploration of on-node and on-implant BMI systems is ongoing, facing limitations in regards to their area and battery. The design of such systems necessitates careful consideration of various factors, including algorithm complexity, available resources, power budgets, CMOS technologies, and platform selection. Despite these efforts, the impact of these factors on the BMI system remains uncertain. ↑ ↓

ChatGPT

· 给审稿人发邮件催促审稿

I am a graduate student. I want to write an email to the reviewer of Nature journal to ask him to accelerate the review process of my submitted paper.

Subject: Request for Expedited Review of Submitted Paper

Dear [Reviewer's Name],

I hope this email finds you well. I am writing to inquire about the status of my submitted paper, [Paper Title], which I submitted to Nature journal for publication.

As a graduate student, this publication holds great significance for my research and career. I am eagerly awaiting the outcome of the review process, but I was wondering if it would be possible to request an expedited review.

I understand that the review process is comprehensive and thorough, but I was hoping that you could prioritize my paper due to the time-sensitive nature of my research. I would greatly appreciate any efforts you can make to ensure a prompt review and decision on my paper.

Thank you for your time and consideration. I look forward to hearing from you soon.

Best regards,

[Your Name]

· 跟导师请假

敬爱的导师：

您好！

我希望通过这封信向您请假，因为我将无法参加明天的组会。很抱歉给您带来了不便。

请允许我解释一下我的具体情况。由于我家中突然发生了紧急情况，我的亲戚需要紧急住院，我需要立即返回家中照顾他们并处理相关手续。因此，我将无法按时参加明天的组会。在这段时间里，我会尽可能安排好我的时间，以便在处理家庭紧急事务的同时，尽量不影响我的学术工作和团队进度。

为了弥补我的缺席，我会与组内其他成员保持紧密联系，了解会议内容并尽快补充我所错过的信息。同时，我将尽自己所能，在请假期间完成相关的工作和任务。

再次为给您和团队带来的不便表示歉意，并感谢您对我的理解和支持。请您考虑批准我的请假请求，我将非常感激。

祝好！

Regenerate response

DeepSeek

· 研究生如何高效学习机器学习类课程？

已深度思考 (用时 27 秒) ^

嗯，用户问的是研究生如何高效学习机器学习类课程。这个问题看起来挺常见的，但得仔细分析一下，用户可能的具体需求是什么。首先，研究生阶段的学习和本科不同，更强调自主学习和深入研究。所以用户可能希望不仅仅是上课，而是如何系统地掌握机器学习，包括理论和应用。

接下来，用户可能的情况：可能是刚开始接触机器学习的研究生，或者已经有一定基础但感觉学习效率不高的人。他们可能需要一个全面的学习策略，涵盖课程选择、理论学习、编程实践、项目经验、论文阅读等方面。

然后，得考虑用户可能没有明确提到的需求。比如，时间管理、资源推荐、如何应对数学基础薄弱的问题，或者如何将机器学习应用到自己的研究领域。这些可能都是潜在的需求点。

需要分步骤来思考。首先，机器学习的基础理论，数学基础很重要，比如线性代数、概率统计、优化方法。研究生可能需要巩固这些，但时间有限，如何高效复习？可能需要推荐重点内容或资源。

然后是课程学习，如何选择合适的课程，比如经典的网课如Andrew Ng的课程，或者更高级的课程如CS229。还要结合学校的课程，合理分配时间。

编程实践是关键，但研究生可能已经有编程基础，需要更深入的项目或比赛经验。比如Kaggle竞赛，或者参与实验室的项目。同时，复现论文代码，这可能帮助他们理解算法细节。

论文阅读也是研究生必备的技能，如何高效阅读和跟踪前沿研究？比如使用Google Scholar关注顶会，学习如何快速抓住论文重点，做笔记的方法等。

论文阅读也是研究生必备的技能，如何高效阅读和跟踪前沿研究？比如使用Google Scholar关注顶会，学习如何快速抓住论文重点，做笔记的方法等。

另外，交流与讨论，加入学习小组，参加学术活动，这些都能促进学习。可能用户在这方面缺乏经验，需要建议如何寻找这样的机会。

还有时间管理，研究生可能需要平衡课程、研究和生活，如何制定计划，避免拖延。可能需要建议使用时间管理工具，比如番茄工作法，或者任务分解。

资源推荐方面，用户可能需要具体的书籍、在线课程、博客、社区等。比如《机器学习》（周志华）、《深度学习》（花书），或者优秀的博客如Towards Data Science。

最后，保持动力和调整心态，研究生阶段压力大，如何保持长期的学习热情，避免焦虑。可能需要建议设定小目标，定期总结，保持健康等。

需要确保回答结构清晰，涵盖各个关键点，同时给出具体可行的建议，避免泛泛而谈。可能还要提醒用户结合自身研究方向，有针对性地学习，比如如果研究方向是NLP，就需要加强自然语言处理相关的知识和实践。

可能还需要注意避免一些常见误区，比如只重视理论忽视实践，或者相反；或者盲目追求新算法而不理解基础。这些点也可以在建议中提到，帮助用户避开陷阱。

总结来说，回答需要全面覆盖高效学习的各个方面，给出具体策略和资源，同时结合研究生阶段的特点，强调自主学习和研究能力的培养。

DeepSeek

· 研究生如何高效学习机器学习类课程？

研究生阶段高效学习机器学习类课程，需要结合系统性学习、实践应用和科研思维的培养。以下分阶段策略和具体建议供参考：

一、基础构建阶段（1-2个月）

1. 数学基础速补

- 线性代数：重点掌握矩阵分解（SVD、特征分解）、张量运算（如PyTorch/TensorFlow中的广播机制）
- 概率统计：深入理解贝叶斯推断、KL散度、MCMC采样（推荐《Pattern Recognition and Machine Learning》第2章）
- 优化方法：掌握梯度下降变体（AdamW、LAMB优化器）、二阶优化（拟牛顿法）的工程实现

2. 核心算法手推实现

- 使用Jupyter Notebook复现经典算法：

```
python
```

复制

```
# 手写线性回归+动量梯度下降
class LinearRegression:
    def __init__(self, alpha=0.01, beta=0.9):
        self.alpha = alpha
        self.beta = beta
        self.v = 0 # 动量项
```

二、进阶实践阶段（3-4个月）

3. 工业级项目开发

- 构建可部署的ML Pipeline：

```
python
```

复制

```
# 使用MLflow实现实验跟踪
import mlflow
from hyperopt import fmin, tpe, hp

def train_model(params):
    with mlflow.start_run():
        model = XGBClassifier(**params)
        model.fit(X_train, y_train)
        score = roc_auc_score(y_val, model.predict_proba(X_val)[:,1])
        mlflow.log_metric("auc", score)
        mlflow.xgboost.log_model(model, "model")
    return -score # 最小化目标

space = {
    'max_depth': hp.quniform('max_depth', 3, 18, 1),
    'gamma': hp.uniform('gamma', 0, 20)
}
best = fmin(fn=train_model, space=space, algo=tpe.suggest, max_evals=100)
```

4. 论文复现技巧

DeepSeek

· 研究生如何高效学习机器学习类课程？

三、科研攻坚阶段（持续）

5. 高效文献管理

- 建立双链笔记系统（Obsidian/Logseq）：

```
[[Transformer]] 在[[ViT]]中的应用表明...
```

复制

- 使用Zotero配合自定义JavaScript脚本自动抓取arXiv最新论文

6. 创新点挖掘方法

- 使用因果推理框架检验模型鲁棒性：

```
python
```

```
# 使用DoWhy库进行因果分析
from dowhy import CausalModel
model = CausalModel(
    data=df,
    treatment='treatment_var',
    outcome='outcome_var',
    graph="digraph { treatment_var -> outcome_var; confounder -> treatment_var; confounder -> outcome_var }"
)
```

复制

四、效率工具链

- 开发环境：配置VS Code Remote SSH + Docker开发环境
- 实验监控：使用Weight & Biases实时跟踪GPU利用率
- 知识管理：编写Python脚本自动将Markdown笔记转换为Anki卡片

五、避坑指南

- 避免“Github Star陷阱”：当某个实现star数 > 5000时，需警惕代码复杂度与论文的匹配度
- 慎用过早优化：在TPU集群上跑简单实验前，先用CPU模式验证算法可行性
- 警惕过时理论：当发现教材中仍在使用MNIST数据集时，应补充现代数据集（如ImageNet-1K）的比较实验

六、前沿方向速通

- 每周精读1篇ICLR/NeurIPS高分论文（OpenReview评分>8）
- 关注新兴领域：
 - 几何深度学习（图神经网络3.0架构）
 - 量子机器学习（PennyLane库实践）
 - 神经微分方程（TorchDiffEq库）

通过以上方法，配合每日4小时深度工作（建议采用90分钟×3段的番茄工作法），可在6个月内完成从入门到科研创新的跨越。关键是在每个阶段设立明确产出目标（如第一阶段产出数学推导笔记，第二阶段完成Kaggle竞赛前10%，第三阶段产出可投稿的论文草稿）。



Part 2: Word Embeddings

How to represent words

- Classical approach: one-hot vectors
 - “hotel” vs. “motel”, “dog” vs. “cat”
 - Super high dimension
 - Semantic meaning is missing

$$w^{aardvark} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^a = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{at} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots w^{zebra} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$(w^{hotel})^T w^{motel} = (w^{hotel})^T w^{cat} = 0$$

Representing words by their context

- Distributional semantics: A word's meaning is given by the words that frequently appear close-by
 - "You shall know a word by the company it keeps" (J. R. Firth 1957: 11)
 - One of the most successful ideas of modern statistical NLP!
- When a word w appears in a text, its context is the set of words that appear nearby (within a fixed-size window).
- Use the many contexts of w to build up a representation of w

...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...

Count based representations

- Principle: capture word co-occurrences
- 2 options: windows vs. full document
- Window: use window around each word
 - Captures both syntactic (POS) and semantic information
- Word-document co-occurrence matrix
 - Give general topics (all sports terms will have similar entries) leading to "Latent Semantic Analysis"

Latent semantic analysis

Term-document matrix x

$$\mathbf{t}_i^T \rightarrow \begin{bmatrix} \mathbf{x}_{1,1} & \dots & \mathbf{x}_{1,j} & \dots & \mathbf{x}_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{x}_{i,1} & \dots & \mathbf{x}_{i,j} & \dots & \mathbf{x}_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{x}_{m,1} & \dots & \mathbf{x}_{m,j} & \dots & \mathbf{x}_{m,n} \end{bmatrix}$$

Document vector d_j

$$\mathbf{d}_j = \begin{bmatrix} \mathbf{x}_{1,j} \\ \vdots \\ \mathbf{x}_{i,j} \\ \vdots \\ \mathbf{x}_{m,j} \end{bmatrix}$$

Term vector t_i

$$\mathbf{t}_i^T = [\mathbf{x}_{i,1} \ \dots \ \mathbf{x}_{i,j} \ \dots \ \mathbf{x}_{i,n}]$$

Question: how to compute the similarity between two terms/documents?

$$\begin{array}{c}
X \\
(\mathbf{d}_j) \\
\downarrow \\
\left[\begin{array}{cccccc} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n} \end{array} \right] = (\hat{\mathbf{t}}_i^T) \rightarrow \left[\begin{array}{c} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_l \end{array} \right] \dots \left[\begin{array}{c} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_l \end{array} \right] \cdot \left[\begin{array}{ccc} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{array} \right] \cdot \left[\begin{array}{ccc} \mathbf{v}_1 & \dots & \mathbf{v}_l \\ \vdots & \ddots & \vdots \\ \mathbf{v}_1 & \dots & \mathbf{v}_l \end{array} \right]
\end{array}$$

$$\begin{aligned}
XX^T &= (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V^{T^T}\Sigma^T U^T) = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T \\
X^T X &= (U\Sigma V^T)^T (U\Sigma V^T) = (V^{T^T}\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T \Sigma V^T
\end{aligned}$$

GloVe: Global Vectors for Word Representation

- To construct an approximate relationship between Word Vector and Co-occurrence Matrix, the authors propose the following formula to approximate the relationship between them:
- $w_i^T \tilde{w}_j + b_i + \tilde{b}_j = \log(x_{ij})$
- where w_i and \tilde{w}_j are word vectors, b_i and \tilde{b}_j are biases.
- By designing a suitable objective function, word vectors can be obtained.

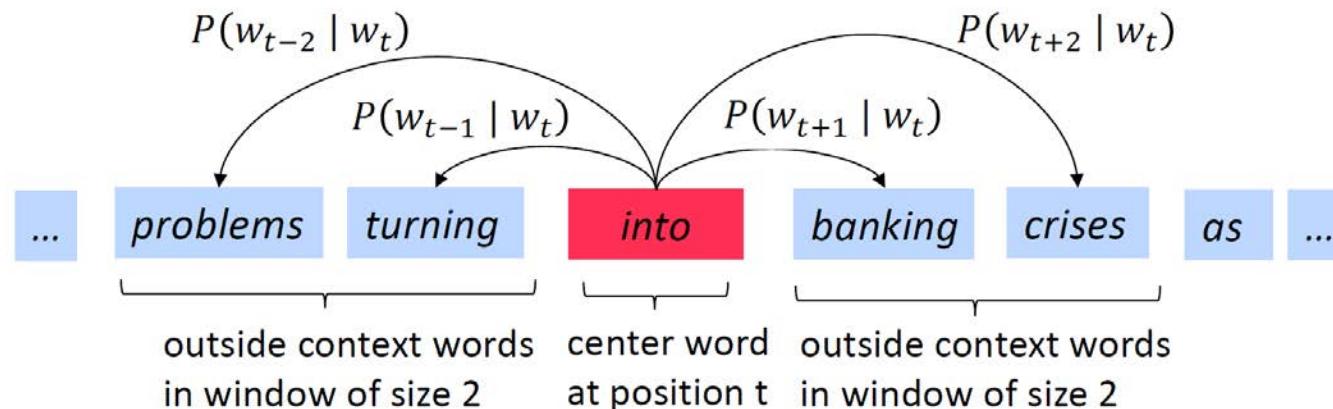
How to represent words

- Classical approach: one-hot vectors
 - "hotel" vs. "motel", "dog" vs. "cat"
 - Super high dimension
 - Semantic meaning is missing
- Dense vectors
 - Use a low-dimensional dense vector
 - SVD based methods
 - NN based approach: Word2vec
 - Skip-gram
 - Continuous bag-of-words (CBOW)

$$w^{aardvark} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^a = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{at} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots w^{zebra} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$
$$(w^{hotel})^T w^{motel} = (w^{hotel})^T w^{cat} = 0$$

Word2vec: skip-gram

- Predict the surrounding words given the center word



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

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

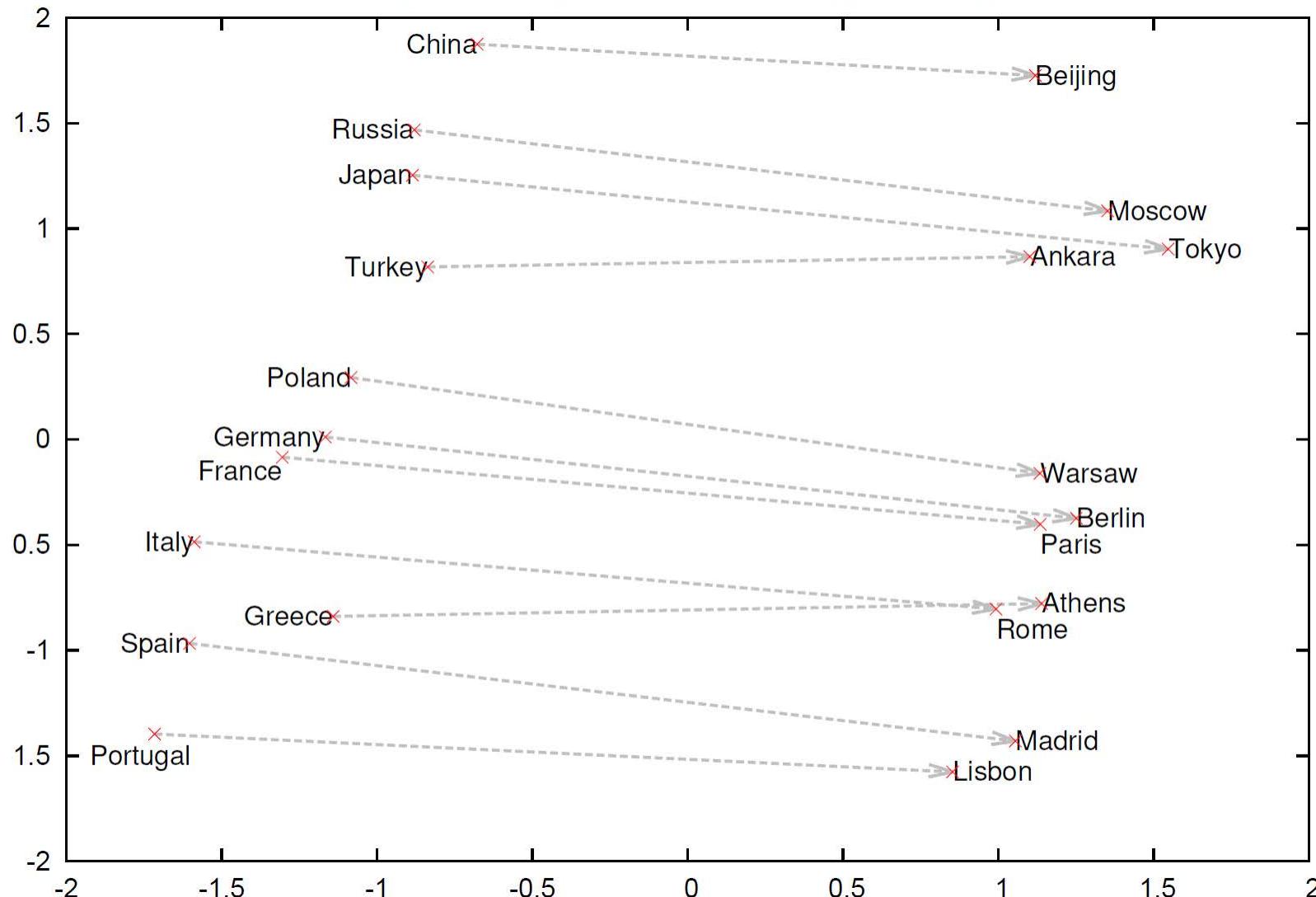
Implementation details

- The normalization factor is costly to compute
 - $O(|V|)$
- Hierarchical softmax: $O(\log_2 |V|)$
 - Use a binary tree with $|V|$ leaf nodes to represent all the words
 - Each word can be reached by a path from the root of the tree
- Negative sampling
 - Sample k negative words from a noise distribution $P(w)$
 - $k \in [5,20]$ for small datasets and $k \in [2,5]$ for large datasets
 - $P(w) \approx U(w)^{\frac{3}{4}}$
- Subsampling frequent words

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

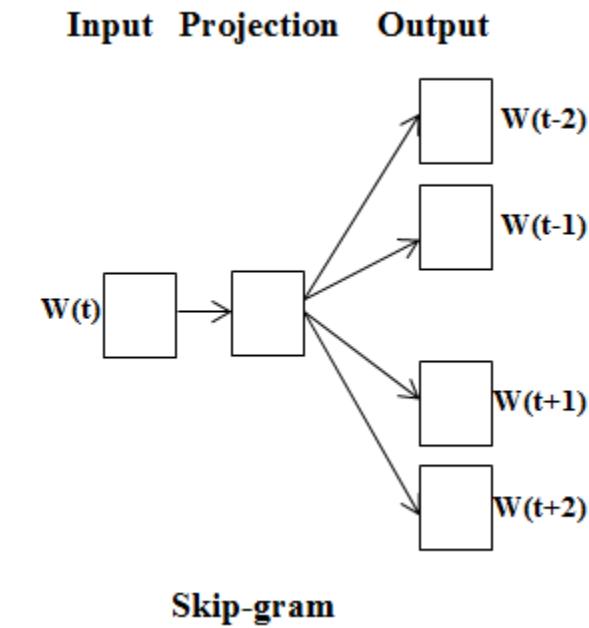
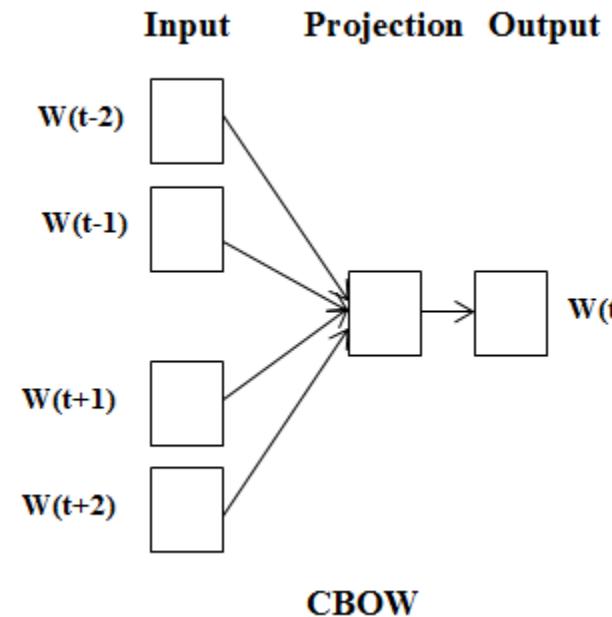
$$\log \sigma(u_o^T v_c) + \sum_{i=1}^k \mathbb{E}_{j \sim P(w)} [\log \sigma(-u_j^T v_c)]$$

Country and capital vectors projected by PCA



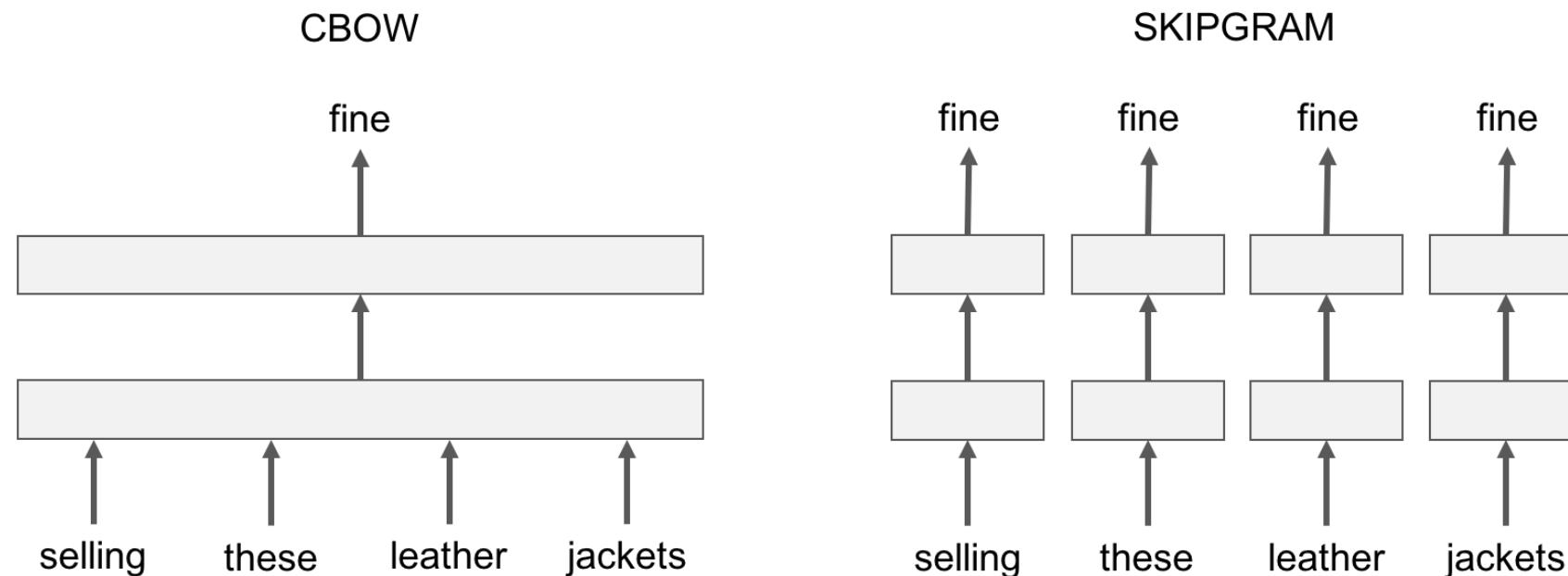
CBOW: continuous bag of words

- Predict a center word from the surrounding context
- Word order within the window is ignored
- Shared weight matrix between input and projection layer for all word positions
- Projection Layer + Softmax Layer



Skip-gram vs. CBOW

- Complexity: CBOW < Skip-gram



I am selling these fine leather jackets

FastText:

- Enrich word vectors with subword information
 - A word is decomposed into a set of n-grams
 - E.g., the set of 3-grams for “where”: <wh, whe, her, ere, re>
- Can be applied to both Skip-gram and CBOW

$$v_w = \sum_{g \in G_w} v_g$$

Importance of n-grams

Out-of-vocabulary word

Query word? gearshift
gearing 0.790762
flywheels 0.779804
flywheel 0.777859
gears 0.776133
driveshafts 0.756345
driveshaft 0.755679
daisywheel 0.749998
wheelsets 0.748578
epicycles 0.744268
gearboxes 0.73986

Word level

Query word? accomodation
sunnhordland 0.775057
accomodations 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

Subword level

accomodations 0.96342
accommodation 0.942124
accommodations 0.915427
accommodative 0.847751
accommodating 0.794353
accomodated 0.740381
amenities 0.729746
catering 0.725975
accomodate 0.703177
hospitality 0.701426

Byte-pair encoding (BPE)

- Handling the out-of-vocabulary problem
- $\text{Dict}=\{\text{'low':5, 'lower':2, 'newest': 6, 'widest':3}\}$
- $\text{Dict}[\text{'lowest'}] = \text{None}$, but 'lowest' is highly related to Dict
- Seems there share some common part, like 'low', 'est'
- Can we leverage some common part of the words in Dict ?

Byte-pair encoding (BPE)

- $Dict = \{ \text{'low':} 5, \text{'lower':} 2, \text{'newest':} 6, \text{'widest':} 3 \}$
- Step-1: split the words into characters
 - $\{ \text{'l o w } </w>: 5, \text{'l o w e r } </w>: 2, \text{'n e w e s t } </w>: 6, \text{'w i d e s t } </w>: 3 \}$
- Step-2: merge the (subword[i], subword[i+1]) whose freq is the highest
 - $\{ \text{'l o w } </w>: 5, \text{'l o w e r } </w>: 2, \text{'n e w e s t } </w>: 6, \text{'w i d e s t } </w>: 3 \}$
 - $\text{Freq(es)} = 9$
 - $\{ \text{'l o w } </w>: 5, \text{'l o w e r } </w>: 2, \text{'n e w e s t } </w>: 6, \text{'w i d e s t } </w>: 3 \}$
 - $\text{Freq(es t)} = 9$
 - $\{ \text{'l o w } </w>: 5, \text{'l o w e r } </w>: 2, \text{'n e w e s t } </w>: 6, \text{'w i d e s t } </w>: 3 \}$
 - $\text{Freq(est } </w>)=9$

Byte-pair encoding

```
# print output
# ('e', 's')
# ('es', 't')
# ('est', '</w>')
# ('l', 'o')
# ('Lo', 'w')
# ('n', 'e')
# ('ne', 'w')
# ('new', 'est</w>')
# ('Low', '</w>')
# ('w', 'i')
# ('wi', 'd')
# ('wid', 'est</w>')
# ('Low', 'e')
# ('Lowe', 'r')
# ('Lower', '</w>')
```

lowest

{l o w e s t </w>}
{l o w es t </w>}
{l o w est </w>}
{l o w est</w>}
{lo w est</w>}
{low est</w>}

So lowest = low@@ est</w>



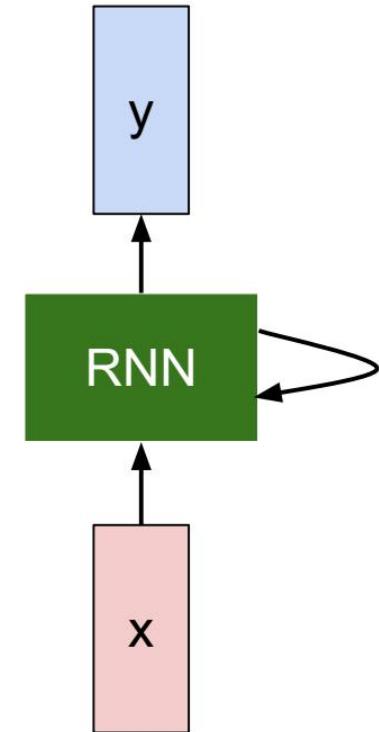
Part 3: Model Architectures (Review)

Recurrent function

We can process a sequence of vectors \mathbf{x} by applying a recurrence formula at every time step:

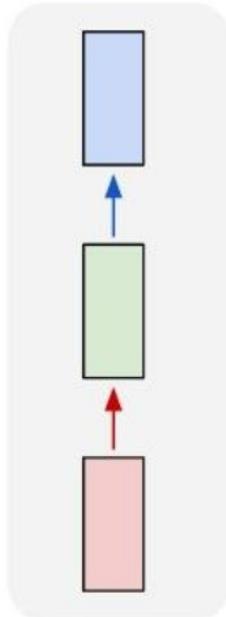
$$h_t = f_W(h_{t-1}, x_t)$$

new state / old state input vector at
some function | some time step
with parameters W

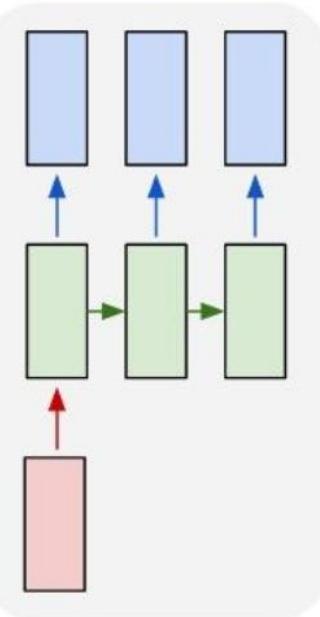


RNNs: flexibilities and possibilities

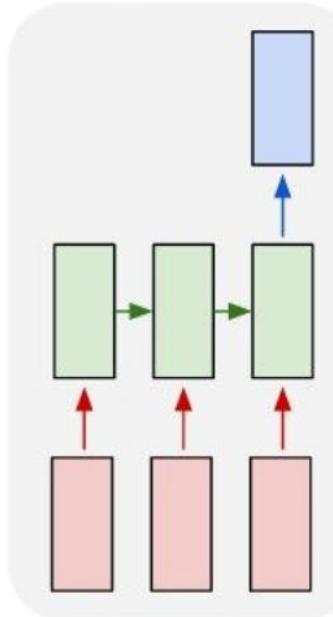
one to one



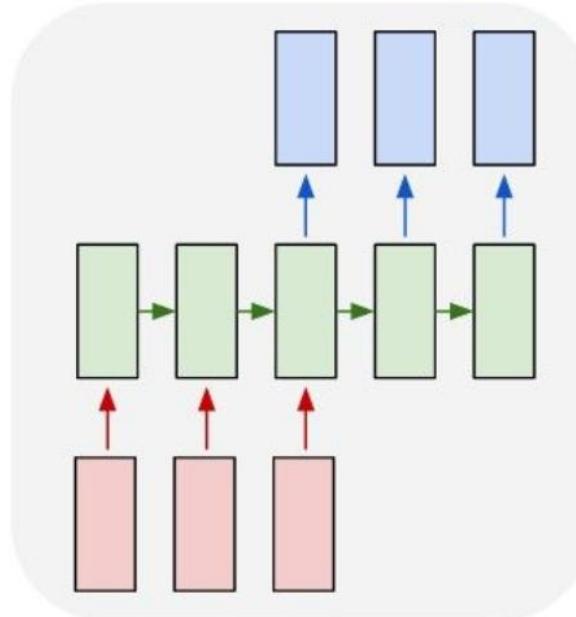
one to many



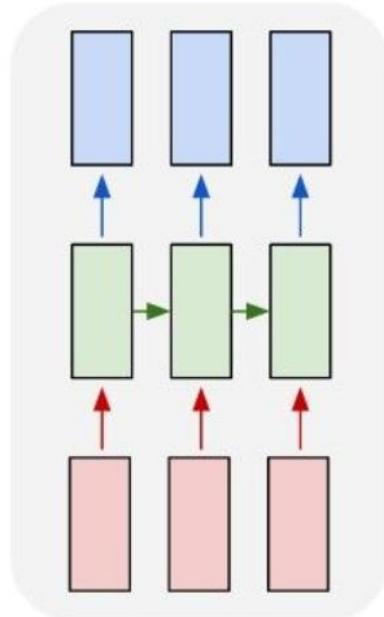
many to one



many to many

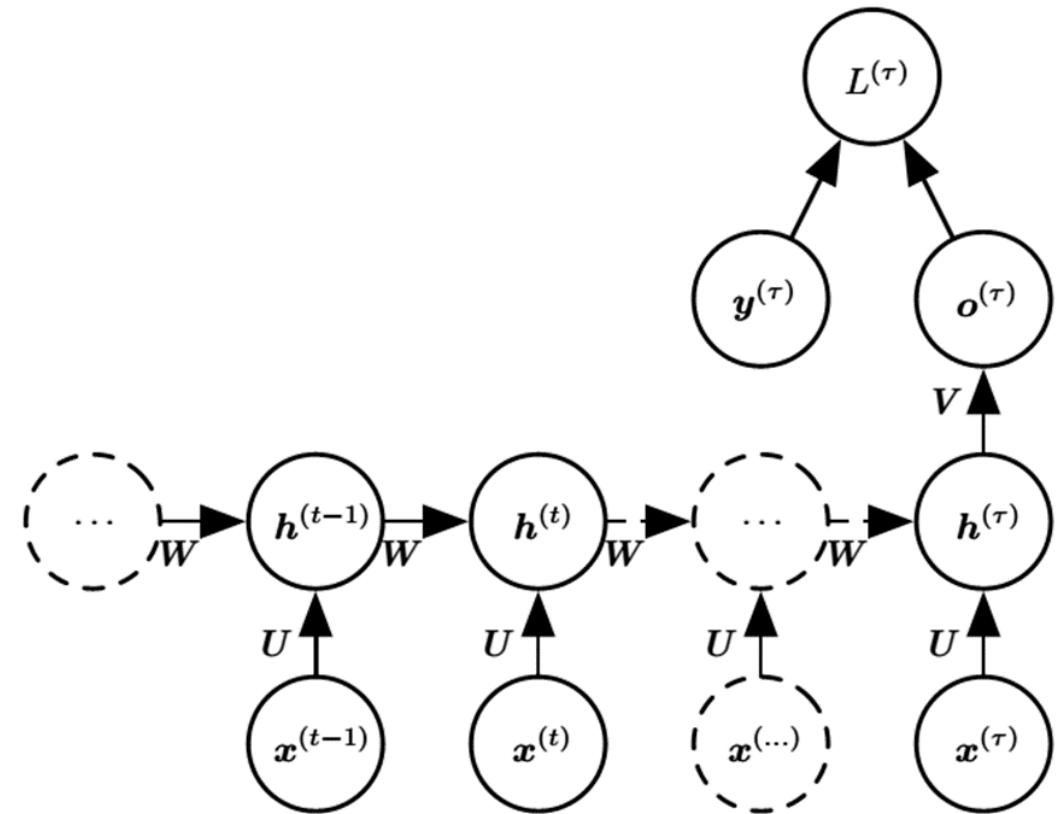


many to many

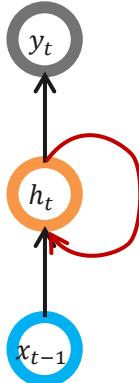


Sentence classification

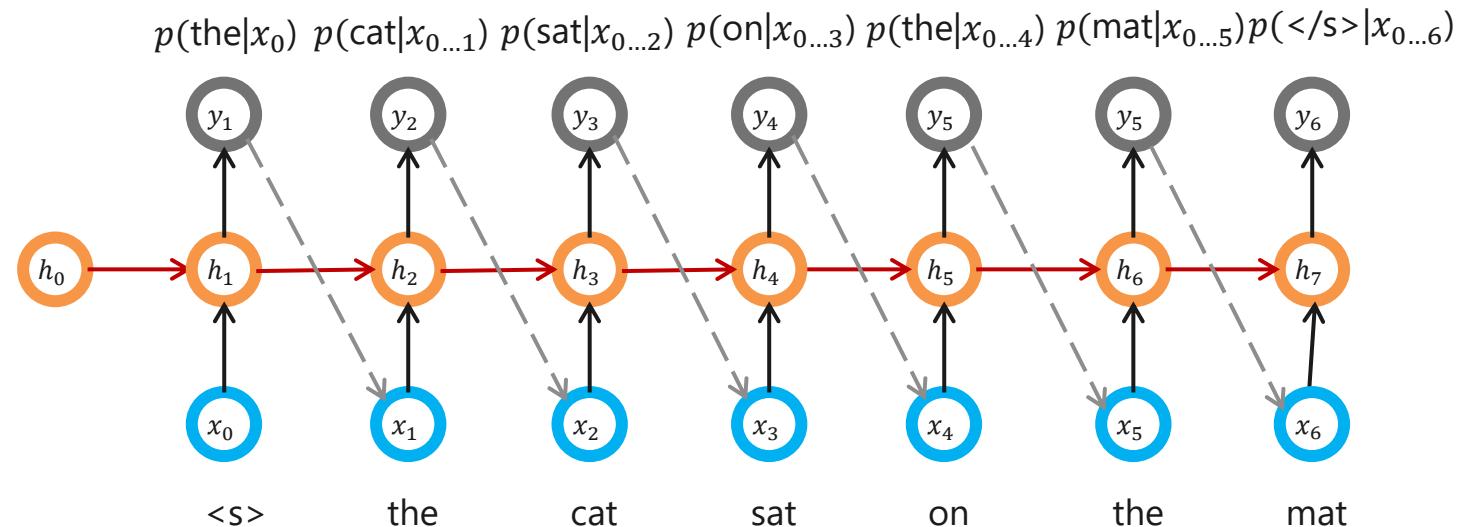
- Encoder RNN
- Example: sentiment analysis
 - Input: text sequence
 - Output: 1/-1, negative/positive



Language modeling

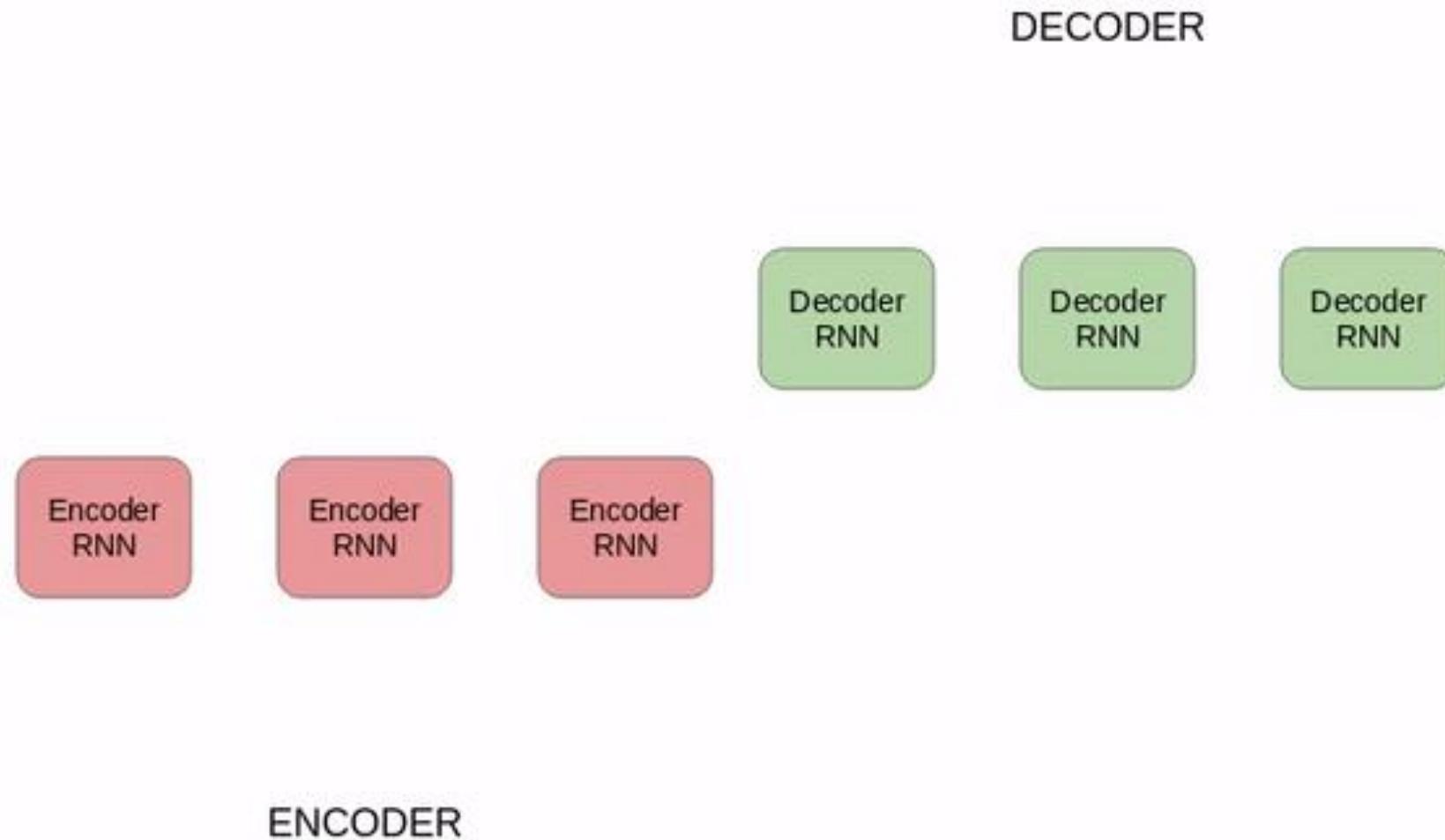


RNN

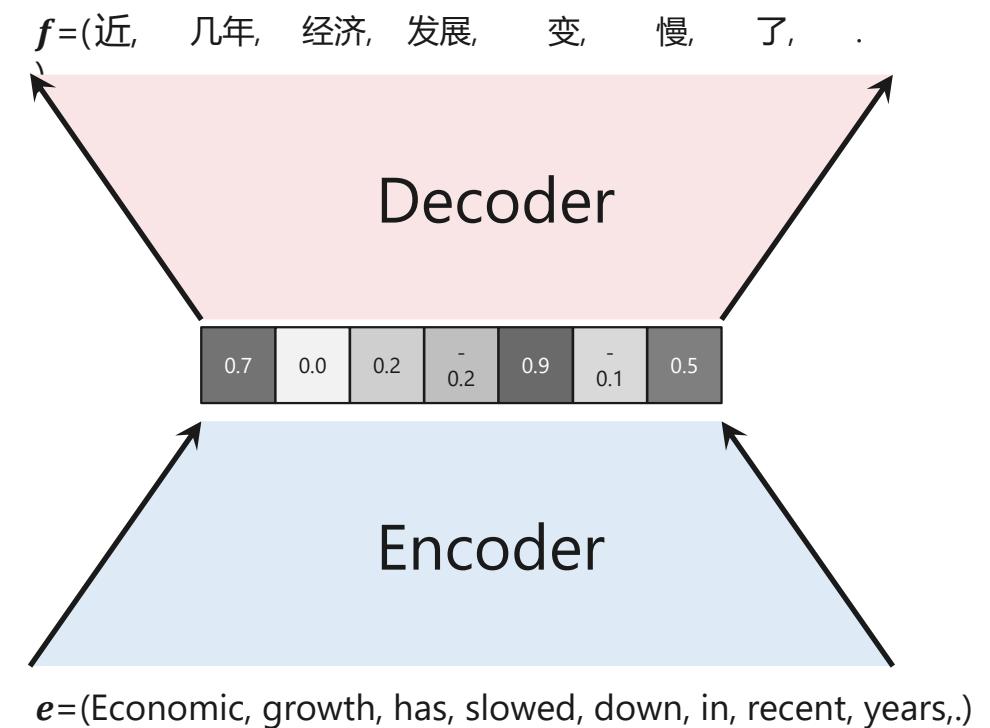
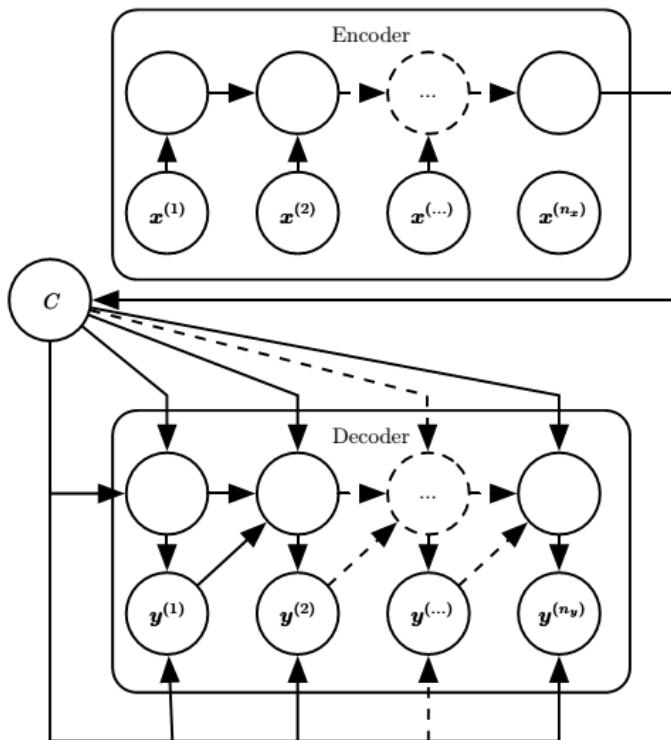


Unfolded RNN for Language Modeling

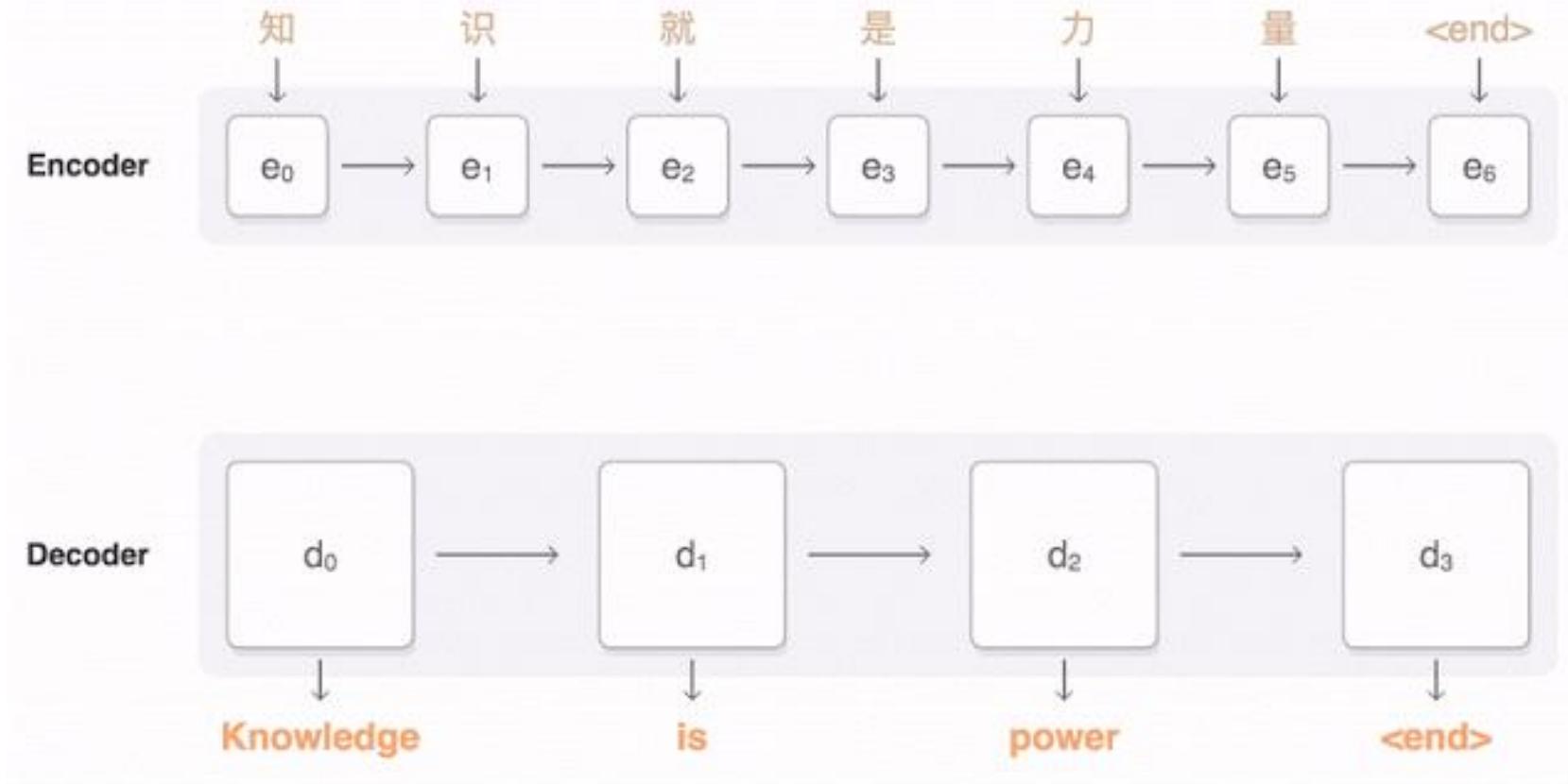
Question answering



Machine translation



Machine translation

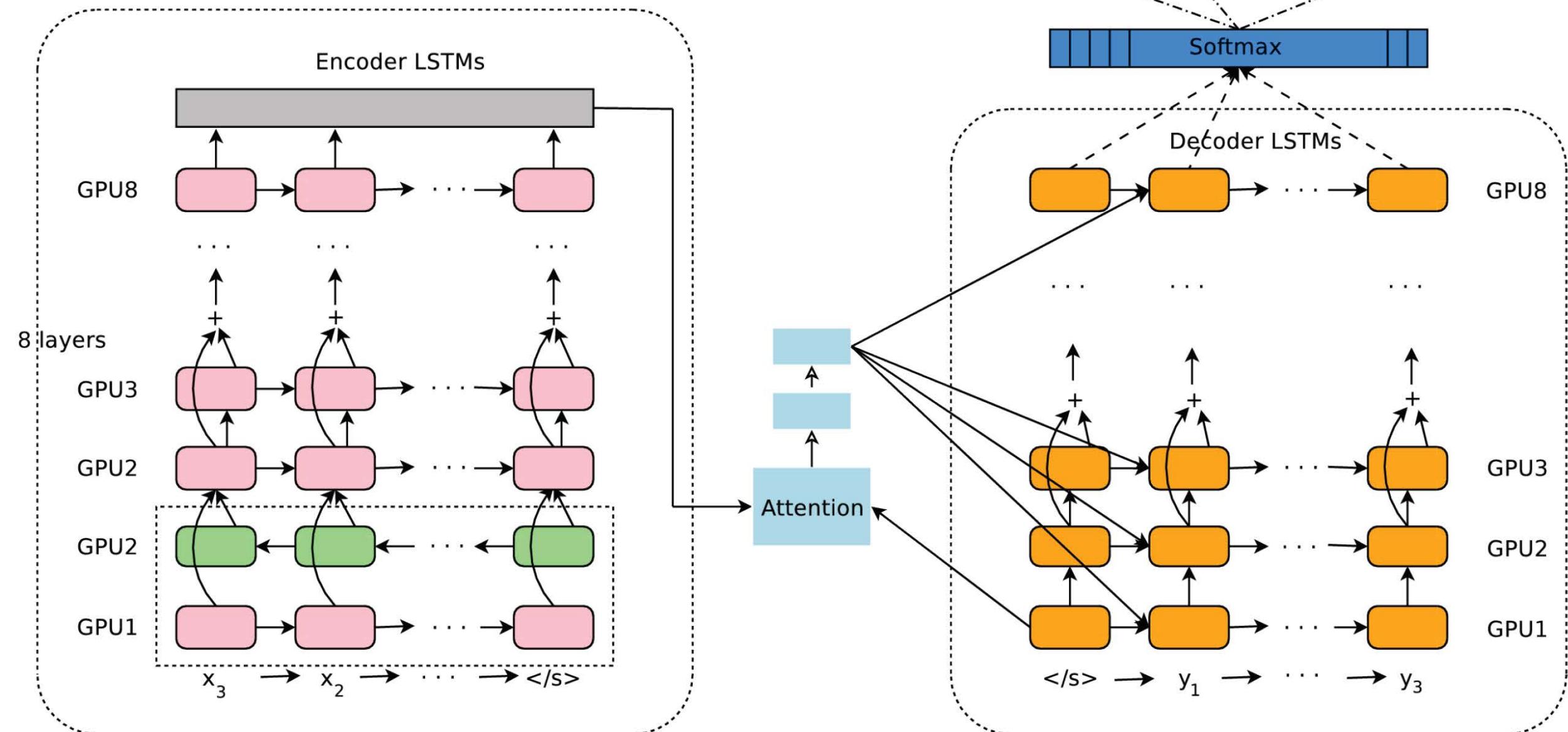




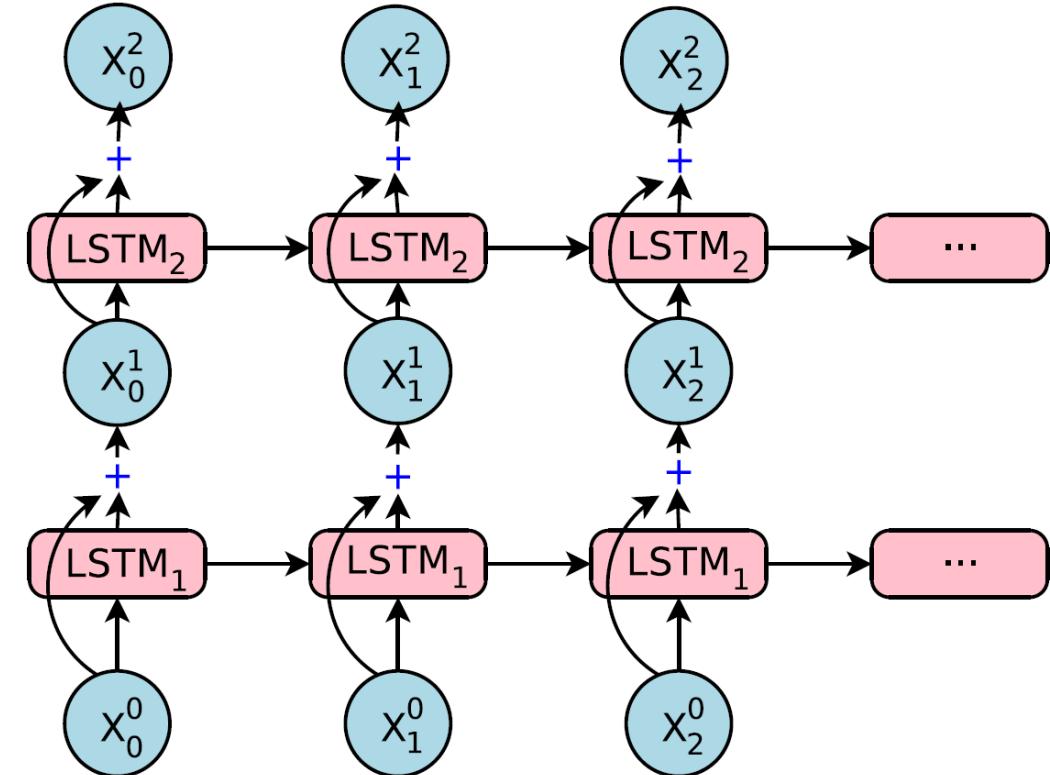
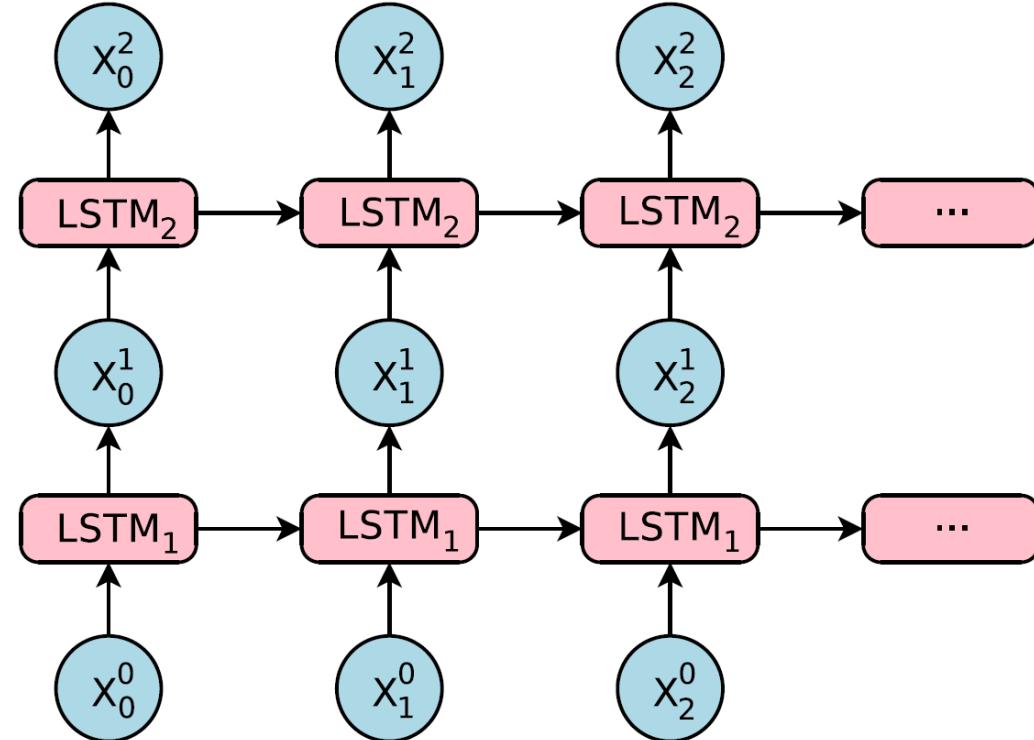
Case Study

Google's Neural Machine Translation System

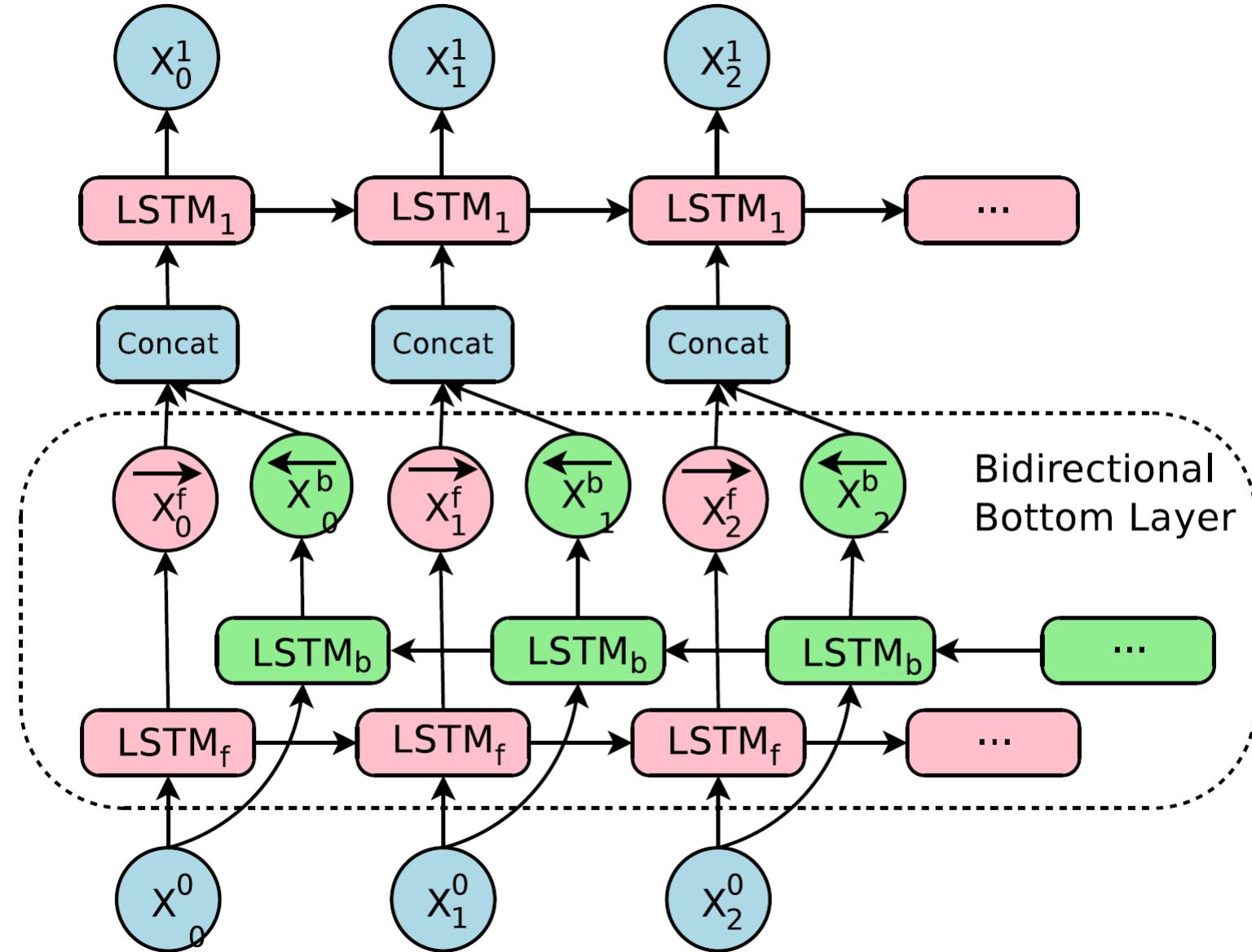
8-layer LSTM



Make it deep through residual connections



Bidirectional bottom layer



Loss functions

$$\mathcal{O}_{\text{ML}}(\boldsymbol{\theta}) = \sum_{i=1}^N \log P_{\theta}(Y^{*(i)} \mid X^{(i)}) .$$

$$\mathcal{O}_{\text{RL}}(\boldsymbol{\theta}) = \sum_{i=1}^N \sum_{Y \in \mathcal{Y}} P_{\theta}(Y \mid X^{(i)}) \ r(Y, Y^{*(i)}).$$

$$\mathcal{O}_{\text{Mixed}}(\boldsymbol{\theta}) = \alpha * \mathcal{O}_{\text{ML}}(\boldsymbol{\theta}) + \mathcal{O}_{\text{RL}}(\boldsymbol{\theta})$$

Decoding: refined beam search

- Max-probability based beam search with a coverage penalty and length normalization

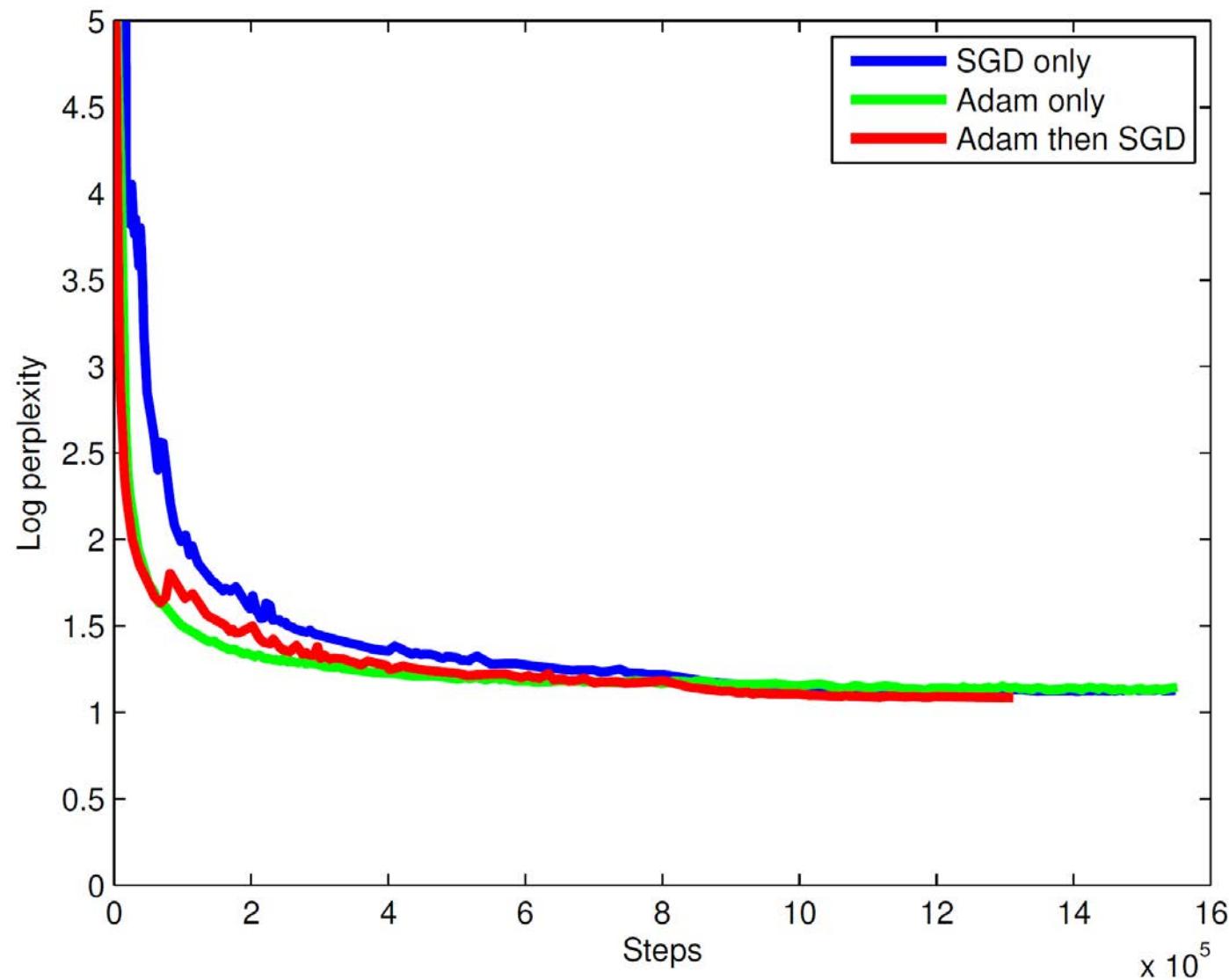
$$s(Y, X) = \log(P(Y|X))/lp(Y) + cp(X; Y)$$

$$lp(Y) = \frac{(5 + |Y|)^\alpha}{(5 + 1)^\alpha}$$

$$cp(X; Y) = \beta * \sum_{i=1}^{|X|} \log(\min(\sum_{j=1}^{|Y|} p_{i,j}, 1.0)),$$

BLEU		α					
		0.0	0.2	0.4	0.6	0.8	1.0
β	0.0	30.3	30.7	30.9	31.1	31.2	31.1
	0.2	31.4	31.4	31.4	31.3	30.8	30.3
	0.4	31.4	31.4	31.4	31.1	30.5	29.6
	0.6	31.4	31.4	31.3	30.9	30.1	28.9
	0.8	31.4	31.4	31.2	30.8	29.8	28.1
	1.0	31.4	31.3	31.2	30.6	29.4	27.2

Optimizer



Single model results

WMT En->Fr (newstest2014)

Model	BLEU
Word	37.90
Character	38.01
WPM-8K	38.27
WPM-16K	37.60
WPM-32K	38.95
Mixed Word/Character	38.39
PBMT [15]	37.0
LSTM (6 layers) [31]	31.5
LSTM (6 layers + PosUnk) [31]	33.1
Deep-Att [45]	37.7
Deep-Att + PosUnk [45]	39.2

WMT En->De (newstest2014)

Model	BLEU
Word	23.12
Character (512 nodes)	22.62
WPM-8K	23.50
WPM-16K	24.36
WPM-32K	24.61
Mixed Word/Character	24.17
PBMT [6]	20.7
RNNSearch [37]	16.5
RNNSearch-LV [37]	16.9
RNNSearch-LV [37]	16.9
Deep-Att [45]	20.6

Ensemble results

WMT En->Fr (newstest2014)

	Model	BLEU
	WPM-32K (8 models)	40.35
RL-refined	WPM-32K (8 models)	41.16
	LSTM (6 layers) [31]	35.6
	LSTM (6 layers + PosUnk) [31]	37.5
	Deep-Att + PosUnk (8 models) [45]	40.4

WMT En->De (newstest2014)

	Model	BLEU
	WPM-32K (8 models)	26.20
RL-refined	WPM-32K (8 models)	26.30

	Model	BLEU	Side-by-side averaged score
	PBMT [15]	37.0	3.87
	NMT before RL	40.35	4.46
	NMT after RL	41.16	4.44
	Human		4.82

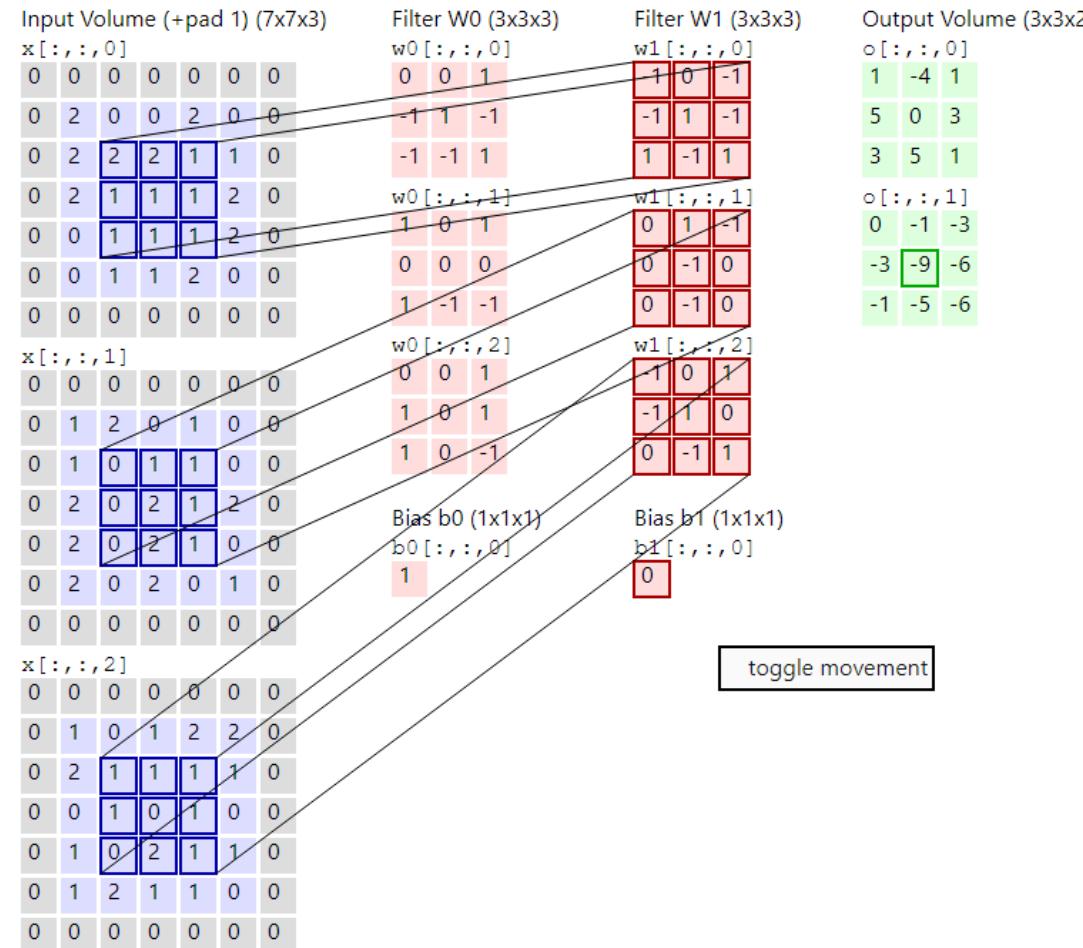


Convolutional Neural Network

CNN also works

Convolutional Neural Networks (CNNs)

- Convolution + Pooling



Convolutional Neural Networks (CNNs)

- Convolution operations over the words
- Multiple convolutional filters

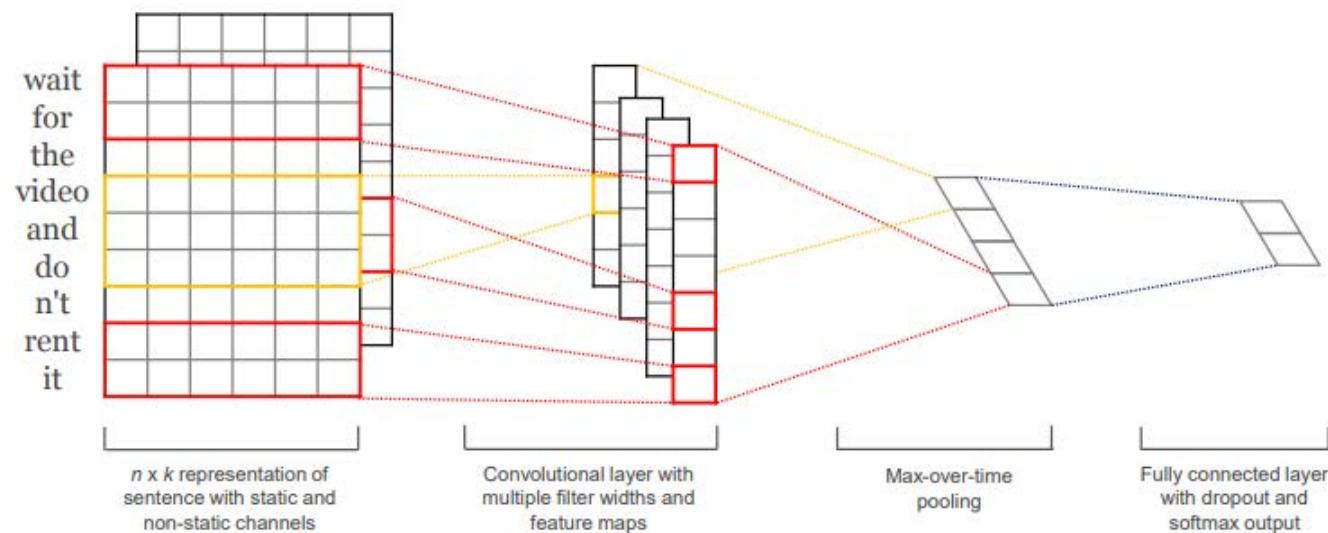


Figure 1: Model architecture with two channels for an example sentence.

Convolutional Sequence-to-Sequence

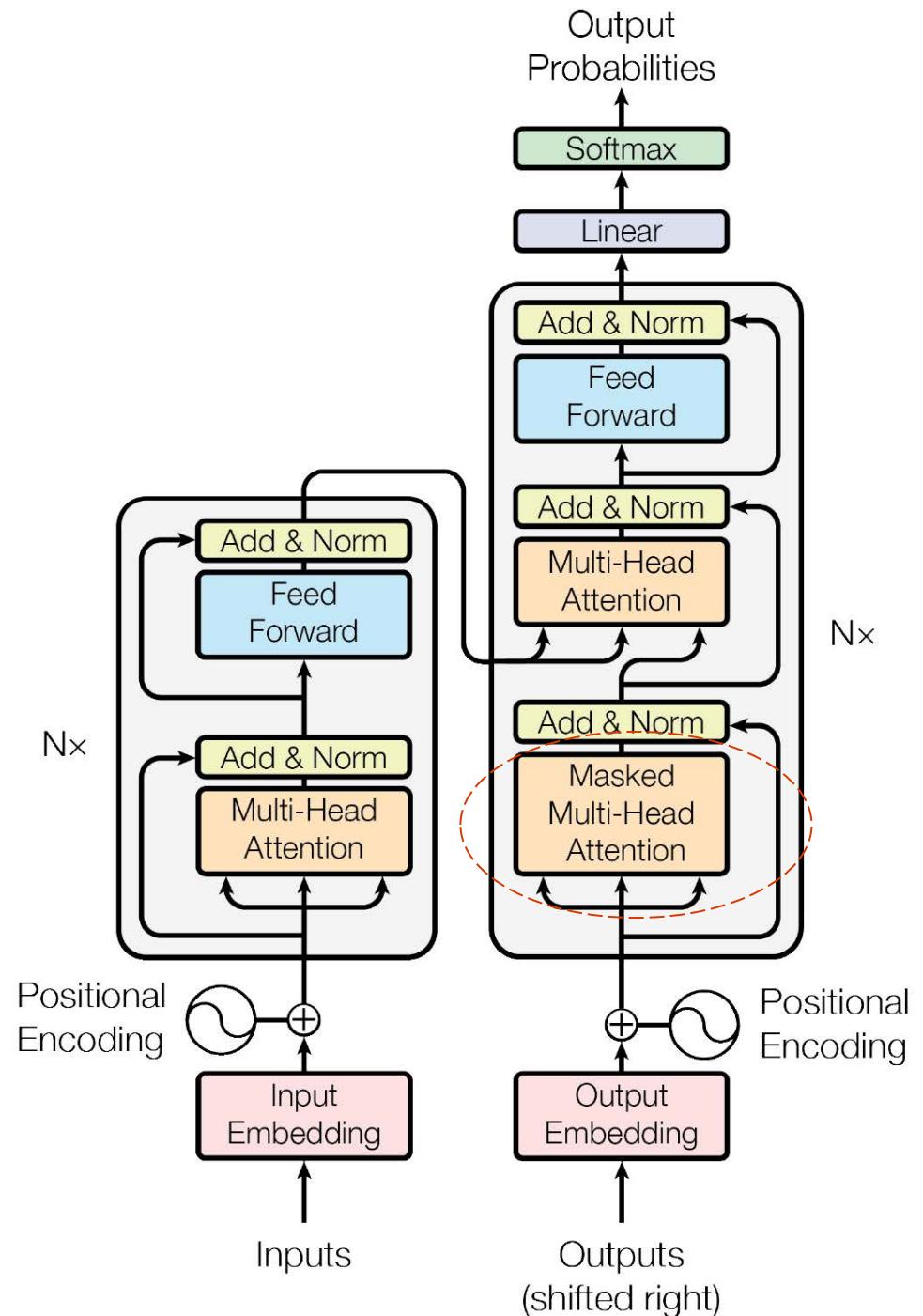
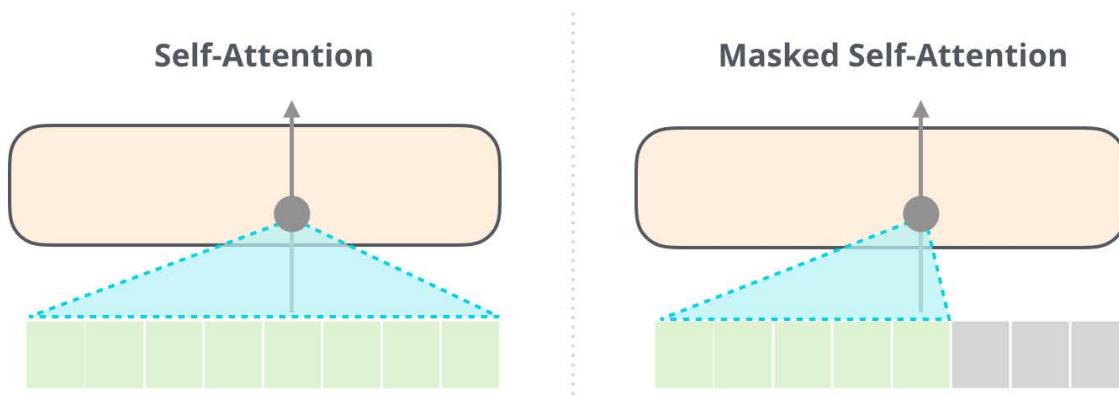
. la maison de Léa <end> .

Transformer

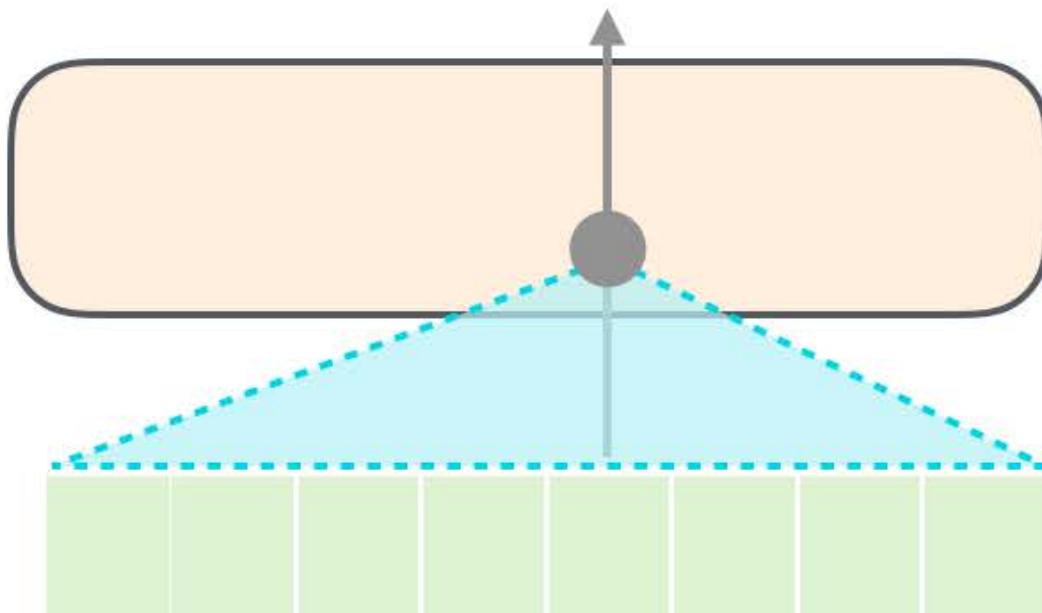
- Attention is all you need

Transformer architecture

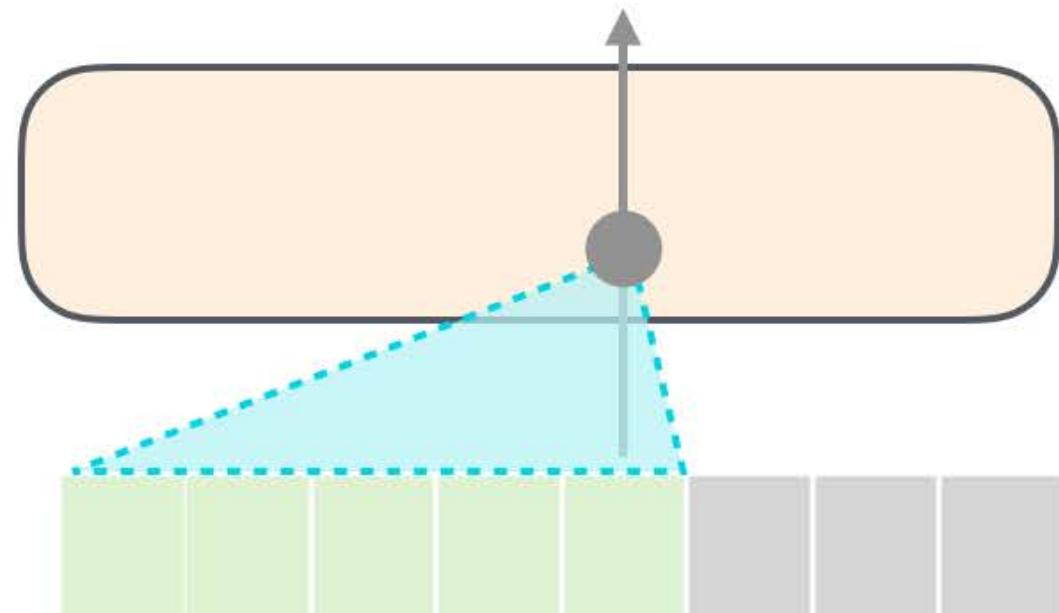
- Encoder-decoder architecture
 - Attention layer
 - Feed-forward layer
- Attentions in the model
 - Self-attention layers in the encoder
 - Masked self-attention layers in the decoder
 - Encoder-decoder attention layers in decoder



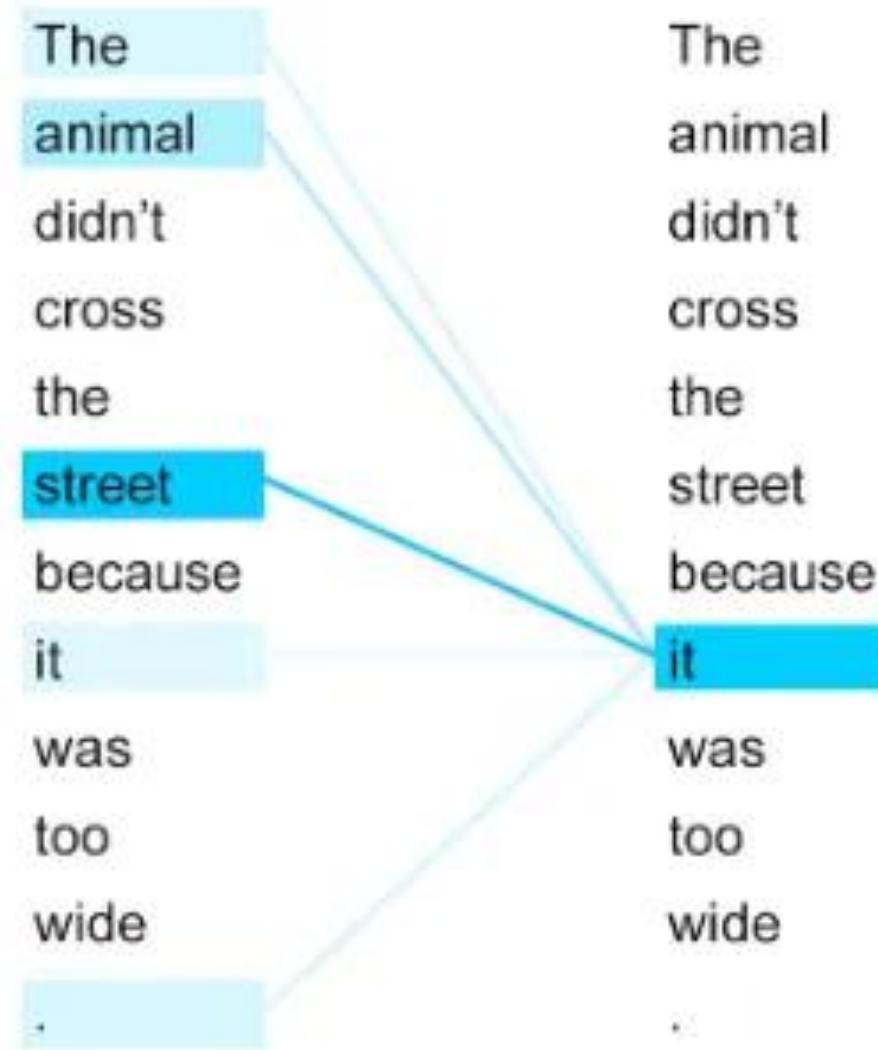
Self-Attention



Masked Self-Attention



Why self-attention?



Calculating self-attention

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector
I	q_1	k_1	v_1
Love		k_2	v_2
China		k_3	v_3

Calculating self-attention

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$

Calculating self-attention

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score	Softmax
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}

Calculating self-attention

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}	$x_{11} v_1$
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}	$x_{12} v_2$
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}	$x_{13} v_3$

Calculating self-attention

$$z_i = \sum_i x_{ij} v_j$$

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}	$x_{11} v_1$	z_1
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}	$x_{12} v_2$	
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}	$x_{13} v_3$	

Calculating self-attention

$$z_i = \sum_i x_{ij} v_j$$

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}	$x_{11} v_1$	z_1
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}	$x_{12} v_2$	
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}	$x_{13} v_3$	

Word	Q vector	K vector	V vector
I	q_1	k_1	v_1
Love	q_2	k_2	v_2
China		k_3	v_3

Calculating self-attention

$$z_i = \sum_i x_{ij} v_j$$

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}	$x_{11}v_1$	z_1
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}	$x_{12}v_2$	
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}	$x_{13}v_3$	

Word	Q vector	K vector	V vector	Score
I	q_1	k_1	v_1	$q_2 \cdot k_1 / \sqrt{d_k}$
Love	q_2	k_2	v_2	$q_2 \cdot k_2 / \sqrt{d_k}$
China		k_3	v_3	$q_2 \cdot k_3 / \sqrt{d_k}$

Calculating self-attention

$$z_i = \sum_i x_{ij} v_j$$

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}	$x_{11} v_1$	z_1
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}	$x_{12} v_2$	
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}	$x_{13} v_3$	

Word	Q vector	K vector	V vector	Score	Softmax
I	q_1	k_1	v_1	$q_2 \cdot k_1 / \sqrt{d_k}$	x_{21}
Love	q_2	k_2	v_2	$q_2 \cdot k_2 / \sqrt{d_k}$	x_{22}
China		k_3	v_3	$q_2 \cdot k_3 / \sqrt{d_k}$	x_{23}

Calculating self-attention

$$z_i = \sum_i x_{ij} v_j$$

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}	$x_{11}v_1$	z_1
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}	$x_{12}v_2$	
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}	$x_{13}v_3$	

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v
I	q_1	k_1	v_1	$q_2 \cdot k_1 / \sqrt{d_k}$	x_{21}	$x_{21}v_1$
Love	q_2	k_2	v_2	$q_2 \cdot k_2 / \sqrt{d_k}$	x_{22}	$x_{22}v_2$
China		k_3	v_3	$q_2 \cdot k_3 / \sqrt{d_k}$	x_{23}	$x_{23}v_3$

Calculating self-attention

$$z_i = \sum_i x_{ij} v_j$$

- 3 vectors for each word: query vector, key vector, value vector

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}	$x_{11} v_1$	z_1
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}	$x_{12} v_2$	
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}	$x_{13} v_3$	

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_2 \cdot k_1 / \sqrt{d_k}$	x_{21}	$x_{21} v_1$	z_1
Love	q_2	k_2	v_2	$q_2 \cdot k_2 / \sqrt{d_k}$	x_{22}	$x_{22} v_2$	z_2
China		k_3	v_3	$q_2 \cdot k_3 / \sqrt{d_k}$	x_{23}	$x_{23} v_3$	

Calculating self-attention

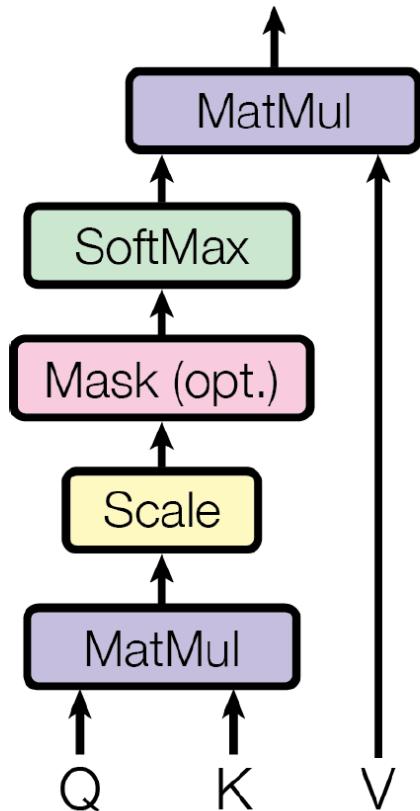
$$z_i = \sum_i x_{ij} v_j$$

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_1 \cdot k_1 / \sqrt{d_k}$	x_{11}	$x_{11} v_1$	z_1
Love		k_2	v_2	$q_1 \cdot k_2 / \sqrt{d_k}$	x_{12}	$x_{12} v_2$	
China		k_3	v_3	$q_1 \cdot k_3 / \sqrt{d_k}$	x_{13}	$x_{13} v_3$	

Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_2 \cdot k_1 / \sqrt{d_k}$	x_{21}	$x_{21} v_1$	z_1
Love	q_2	k_2	v_2	$q_2 \cdot k_2 / \sqrt{d_k}$	x_{22}	$x_{22} v_2$	z_2
China		k_3	v_3	$q_2 \cdot k_3 / \sqrt{d_k}$	x_{23}	$x_{23} v_3$	

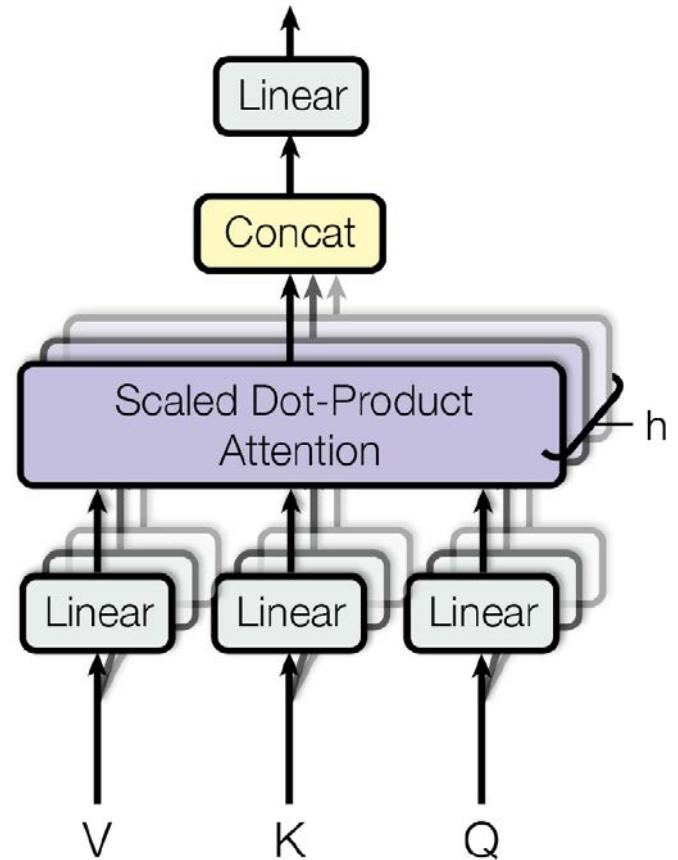
Word	Q vector	K vector	V vector	Score	Softmax	Softmax*v	Sum
I	q_1	k_1	v_1	$q_3 \cdot k_1 / \sqrt{d_k}$	x_{31}	$x_{31} v_1$	z_1
Love	q_2	k_2	v_2	$q_3 \cdot k_2 / \sqrt{d_k}$	x_{32}	$x_{32} v_2$	z_2
China	q_3	k_3	v_3	$q_3 \cdot k_3 / \sqrt{d_k}$	x_{33}	$x_{33} v_3$	z_3

Scaled Dot-Product Attention



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

Multi-Head Attention



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

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

WMT14

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [17]		23.75		
Deep-Att + PosUnk [37]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [36]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [31]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [37]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [36]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1		$3.3 \cdot 10^{18}$
Transformer (big)	28.4	41.0		$2.3 \cdot 10^{19}$



Part 4: Sequence learning

Sequence-to-sequence learning

- Learn a mapping from \mathcal{X} to \mathcal{Y} , where
 - \mathcal{X} and \mathcal{Y} are collection of sequences.

	\mathcal{X}	\mathcal{Y}
Machine translation*	Source language	Target language
Q & A	Question	Answer
Dialogue	Context	response
Speech synthesis	Text	Speech
Text summarization	Long document	A summary
Image captioning	image	text

*Sequence to Sequence Learning with Neural Networks, NeurIPS'2014

Differences & Challenge

- Dependency of tokens in target sequence
 - Training and evaluation
- Large target space
 - $|\text{vocab size}|^{\text{seq len}}$
- Various kinds of input/output
 - Long input and long output
 - Multi-modality input
 -

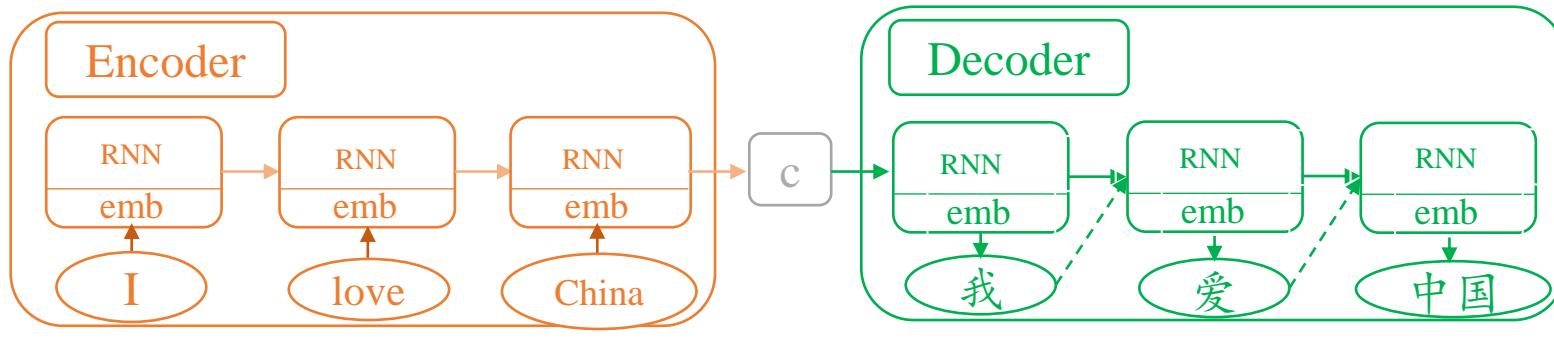
Machine translation



Neural Machine Translation (NMT)

- A classical seq2seq task
- Various of settings

Encoder-decoder framework for NMT



$$P(y_t | y_{<t}, c)$$

More details:

1. embedding: $\mathcal{V} \rightarrow \mathbb{R}^d$
2. $h_t = \text{RNN}(h_{t-1}, x_t)$ (LSTM, GRU)
3. $y_t = \text{softmax}(\dots)$ over $\mathcal{V}_{\text{target-language}}$



Q: Dependency of tokens in target sequence
A: Training solved by the encoder-decoder arch

NMT (Training objective)

- $\min \mathcal{J}(\theta) := -\sum_{(x,y) \in \mathcal{B}} \log P(y|x; \theta)$
- For any sentence pair (x, y) ,

$$\log P(y|x; \theta) = \sum_{t=1}^{T_y} \log P(y_t|y_{t-1}, x; \theta)$$

- Use the ground truth as input
 - Teacher forcing
 - Speed & stability

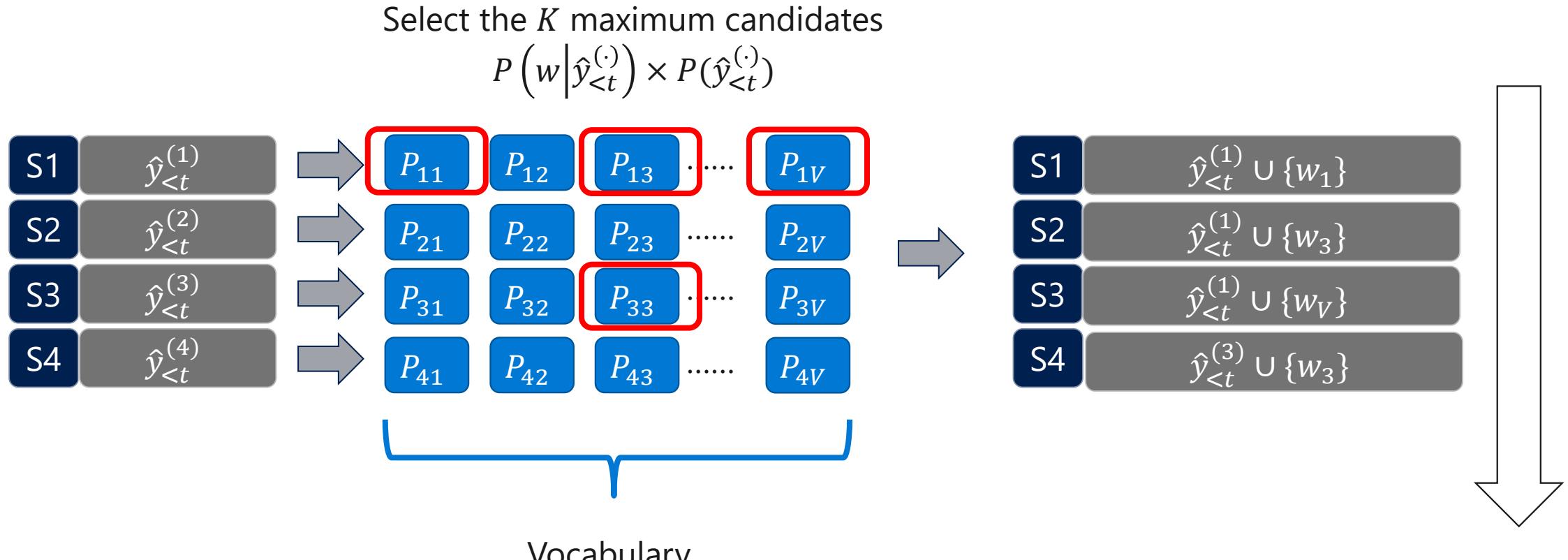


Q: Large target space
A: Teacher forcing

NMT (Inference)

- Assume we need to decode a sentence with length T
- \mathcal{M}_1 : Brute force: $\operatorname{argmax} P(\hat{y}|x; \theta)$, where $\hat{y} \in \mathcal{V}_{\text{target-language}}^T$
 - $V^T \rightarrow 1, 30000^{30} \approx 2^{446} \approx 10^{134}$
- \mathcal{M}_2 : Greedy search: at each step, $\hat{y}_t = \operatorname{argmax} P(\cdot | \hat{y}_{<t}, x; \theta)$
 - $\max_{\hat{y} \in \mathcal{Y}} P(\hat{y}|x) = \max \prod_{t=1}^T P(\hat{y}_t | \hat{y}_{<t}, x) \geq \prod_{t=1}^T \max P(\hat{y}_t | \hat{y}_{<t}, x)$
- \mathcal{M}_3 : Beam search: greedily maintain the top- K candidates

NMT (Beam Search)

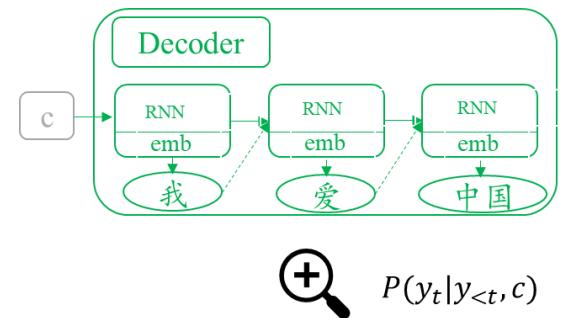


Evaluation

$$\text{BLEU} = \text{BP} \cdot \exp \left(\sum_{n=1}^4 \frac{1}{4} \log p_n \right)$$

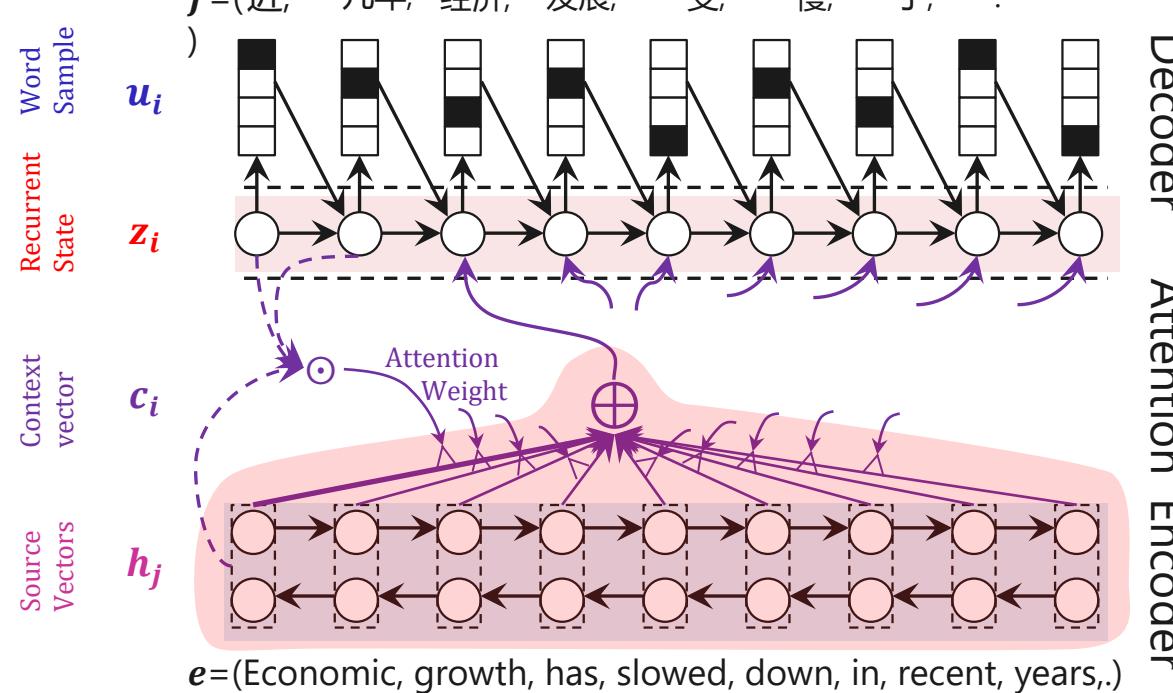
Question: Is it good to use $P(y_t|y_{<t}, c)$ only?

- “..... that the use of a fixed-length vector is a **bottleneck** in improving the performance of this basic encoder–decoder architecture.....” [ref1]
- How about using an adaptive representation the source context, depending on current y_t ?



[1] $P(y_t|y_{t-1}; c)$ can be implemented in different ways; the c could appear in $P(y_1|y_0 ; c)$ only and disappear for $t>1$
[ref1] NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE, ICLR'15

Attention mechanism



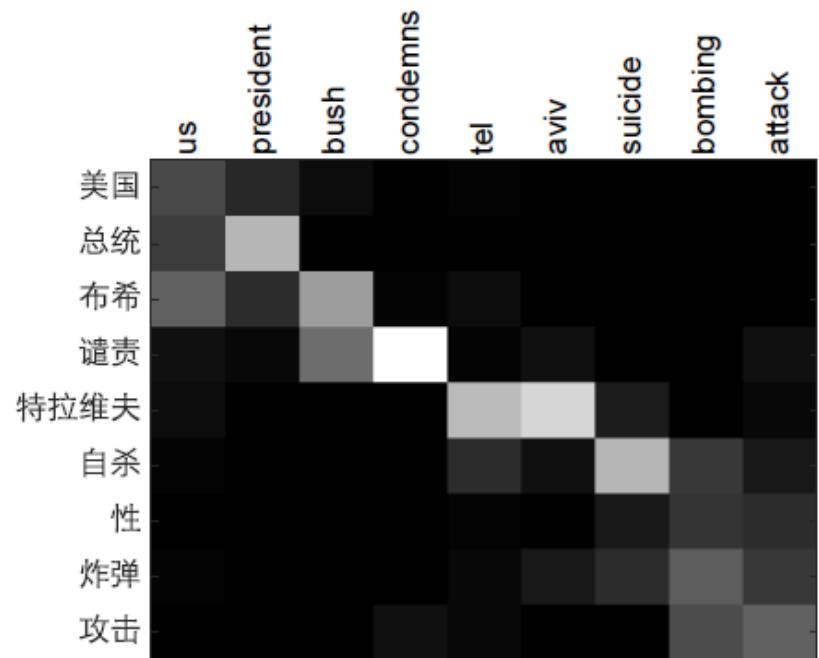
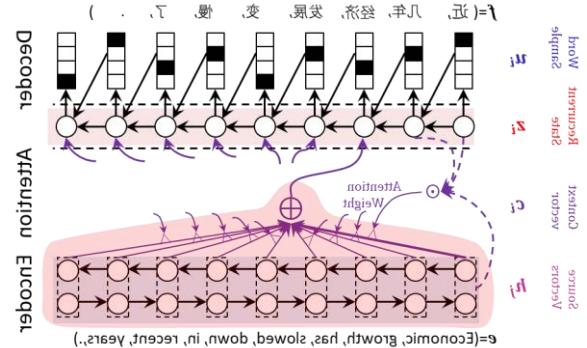
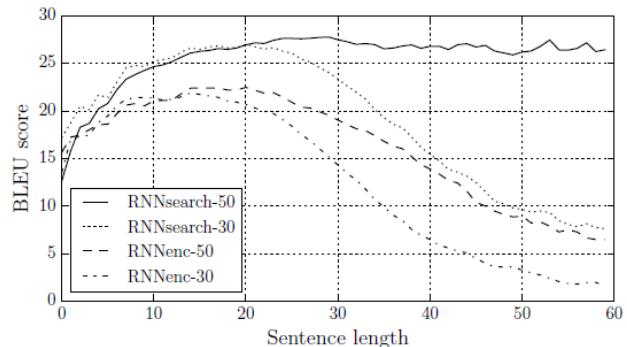
Attention mechanism

Previous Decoder: $y_t = \text{LSTM}(y_{t-1}, z_{t-1}, c)$

Decoder + attention: LSTM($y_{t-1}, z_{t-1}, \textcolor{teal}{c}_t$)

$$\begin{aligned} c_t &= \sum_{j=1}^T \alpha_j h_j \\ \alpha_j &= \frac{\exp(W_z z_{t-1} + W_h h_j)}{\sum_j^T \exp(W_z z_{t-1} + W_h h_j)} \end{aligned}$$

Model	All
RNNencdec-30	13.93
RNNsearch-30	21.50
RNNencdec-50	17.82
RNNsearch-50	26.75

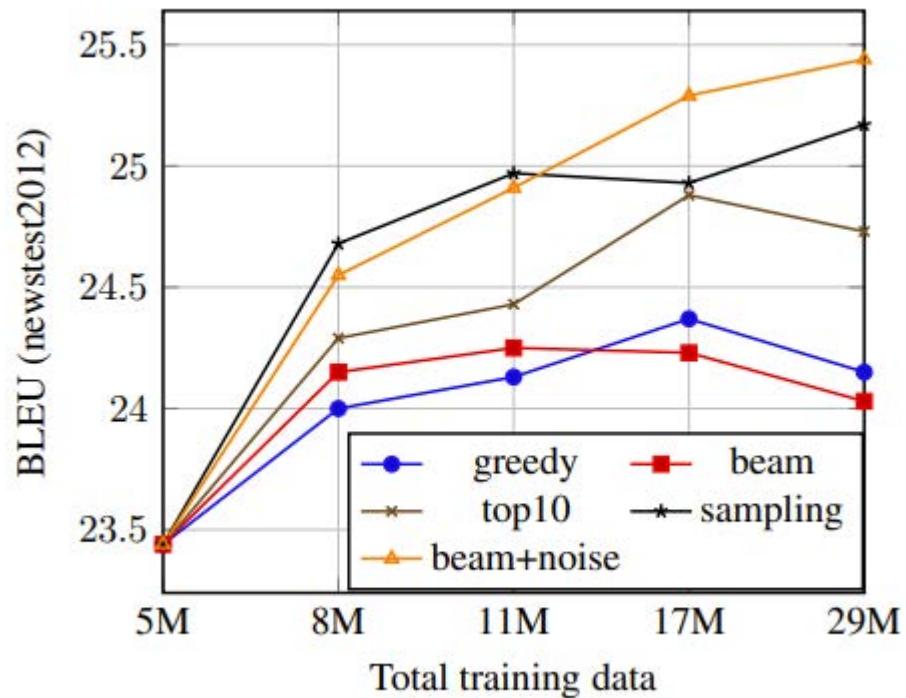


Back translation



- A simple yet effective method to use unsupervised data
- Suppose we want to work on English-to-Chinese translation.
 - We have some aligned (English, Chinese) data
 - We have a large amount of unlabeled Chinese data.
- Step-1: Train a Chinese-to-English translation data
- Step-2: Translate the unlabeled Chinese to English'
- Step-3: Train the English-to-Chinese on (English', Chinese) \cup (English, Chinese)

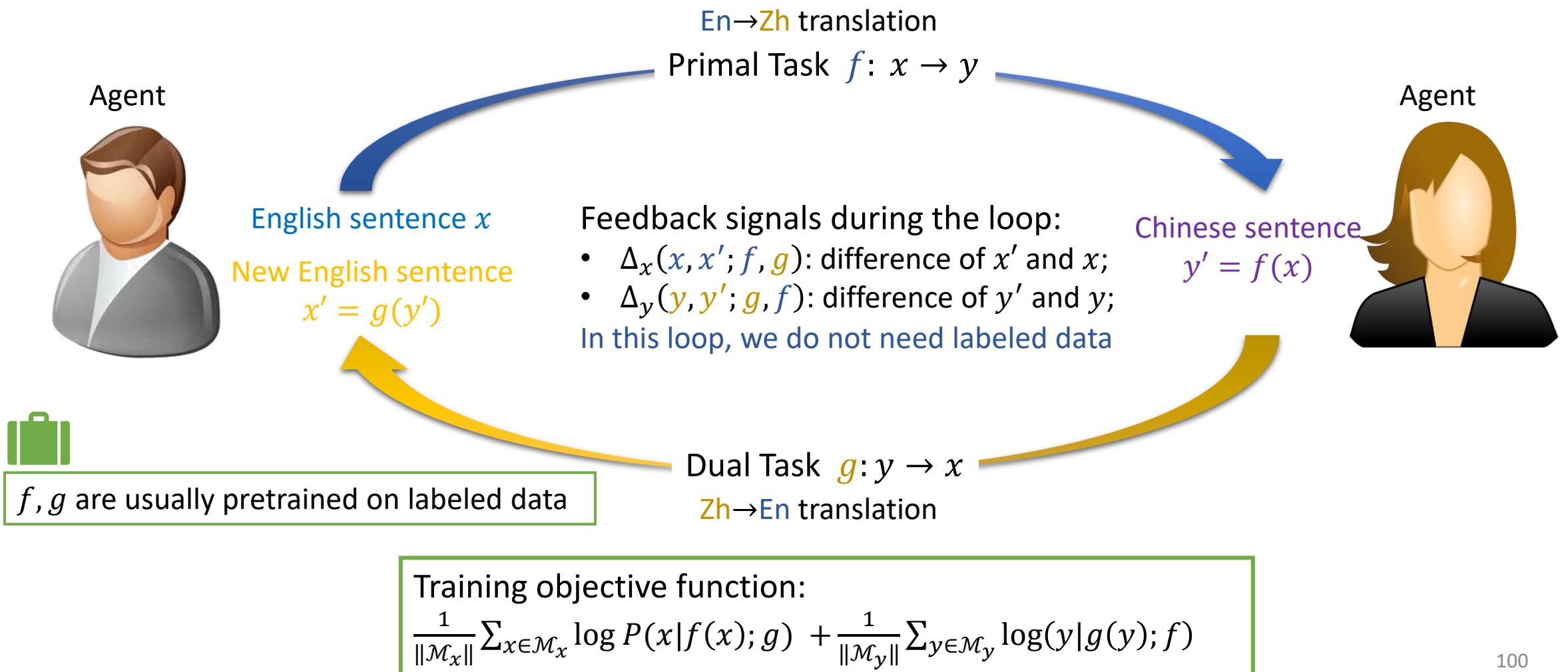
Results



When
#unlabeled=226M,

	En-De	En-Fr
a. Gehring et al. (2017)	25.2	40.5
b. Vaswani et al. (2017)	28.4	41.0
c. Ahmed et al. (2017)	28.9	41.4
d. Shaw et al. (2018)	29.2	41.5
DeepL	33.3	45.9
Our result	35.0	45.6
<i>detok. sacreBLEU³</i>	33.8	43.8

Dual learning for machine translation

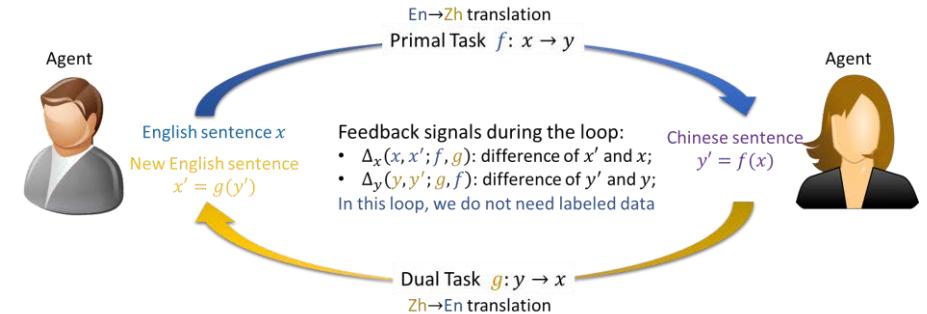


Dual learning for unsupervised NMT

- Primal task: $f: \mathcal{X} \rightarrow \mathcal{Y}$
- Dual task: $g: \mathcal{Y} \rightarrow \mathcal{X}$
- Duality loss

$$\bullet L^{\text{dual}} = \mathbb{E}_{x \sim \mathcal{X}}[-\log P(x|f(x); g)] + \mathbb{E}_{y \sim \mathcal{Y}}[-\log P(y|g(y); f)]$$

- Auto-encoder loss
 - $L^{\text{LM}} = \mathbb{E}_{x \sim \mathcal{X}}[-\log P_{x \rightarrow x}(x|\pi(x))] + \mathbb{E}_{y \sim \mathcal{Y}}[-\log P_{y \rightarrow y}(y|\pi(y))]$
- Is that all?
 - The models are initialized by some pre-trained models in an unsupervised way (e.g., cross-lingual embedding, XLM)
 - Revise the model architecture to some *shared* architecture



[1] Artetxe, M., Labaka, G., Agirre, E., & Cho, K. Unsupervised neural machine translation. *ICLR'18*

[2] Lample, G., Ott, M., Conneau, A., Denoyer, L., & Ranzato, M. A. Phrase-based & neural unsupervised machine translation. *EMNLP'18*

[3] Lample, G., & Conneau, A. (2019). Cross-lingual language model pretraining. *NeurIPS'19*



Sequence to label learning

Mainly discuss the models with pre-training

Sequence to label

- One sequence to label
 - Sentiment classification (SST-2)
 - CoLA (judge the grammatical acceptability of a sentence)
- Multiple sequences to label
 - Similarity and paraphrase tasks (STS-B, MRPC, QQP; whether semantically equivalent)
 - Natural language inference (MNLI, RTE)
 - QA (SQuAD)
 - Multiple choices (RACE)
- Sequence labeling
 - Named entity recognition / Relation extraction

One sequence to label

SST-2, binary classification, movie

eg 1: it 's a charming and often affecting journey . [positive]

eg 2: it 's slow -- very , very slow
[negative]

CoLA

eg 1: The angrier Mary got, the more she looked at pictures. [acceptable]

eg 2: Mickey looked up it. [not acceptable]

Multiple sentences

STS-B (whether two seq semantically match)

Eg 1. A plane is taking off.

An air plane is taking off.

score: 5.000

Eg 2. A boy skateboards down the metal handrail at night.

A brown dog runs on the sand, holding a stick.

score: 0.00

RTE (sentence 1 => sentence 2)

Eg 1 U.S. forces have been engaged in intense fighting after insurgents launched simultaneous attacks in several Iraqi cities, including Fallujah and Baqubah.

Fallujah and Baqubah are Iraqi cities.

Entailment

Eg 2 The harvest of sea-weeds is not allowed in the Puget Sound because of marine vegetation's vital role in providing habitat to important species.

Marine vegetation is harvested.

not-entailment

SQuAD

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

RACE Reading Comprehension Dataset

Passage:

In a small village in England about 150 years ago, a mail coach was standing on the street. It didn't come to that village often. People had to pay a lot to get a letter. The person who sent the letter didn't have to pay the postage, while the receiver had to. "Here's a letter for Miss Alice Brown," said the mailman.
"I'm Alice Brown," a girl of about 18 said in a low voice.
Alice looked at the envelope for a minute, and then handed it back to the mailman.
"I'm sorry I can't take it, I don't have enough money to pay it", she said.
A gentleman standing around were very sorry for her. Then he came up and paid the postage for her.
When the gentleman gave the letter to her, she said with a smile, "Thank you very much, This letter is from Tom. I'm going to marry him. He went to London to look for work. I've waited a long time for this letter, but now I don't need it, there is nothing in it."
"Really? How do you know that?" the gentleman said in surprise.
"He told me that he would put some signs on the envelope. Look, sir, this cross in the corner means that he is well and this circle means he has found work. That's good news."
The gentleman was Sir Rowland Hill. He didn't forget Alice and her letter.
"The postage to be paid by the receiver has to be changed," he said to himself and had a good plan.
"The postage has to be much lower, what about a penny? And the person who sends the letter pays the postage. He has to buy a stamp and put it on the envelope." he said . The government accepted his plan. Then the first stamp was put out in 1840. It was called the "Penny Black". It had a picture of the Queen on it.

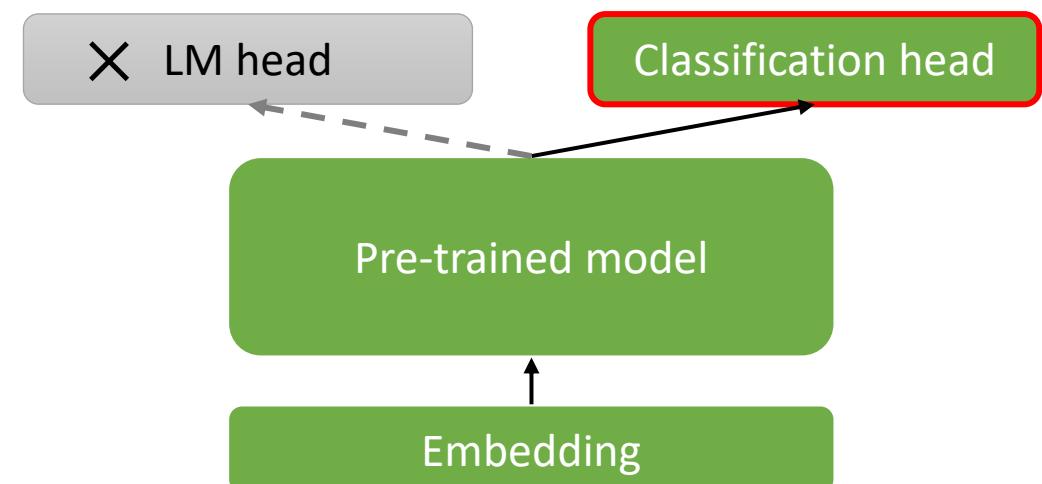
Questions:

- 1): The first postage stamp was made ..
A. in England B. in America C. by Alice D. in 1910
- 2): The girl handed the letter back to the mailman because ..
A. she didn't know whose letter it was
B. she had no money to pay the postage
C. she received the letter but she didn't want to open it
D. she had already known what was written in the letter
- 3): We can know from Alice's words that ..
A. Tom had told her what the signs meant before leaving
B. Alice was clever and could guess the meaning of the signs
C. Alice had put the signs on the envelope herself
D. Tom had put the signs as Alice had told him to
- 4): The idea of using stamps was thought of by ..
A. the government
B. Sir Rowland Hill
C. Alice Brown
D. Tom
- 5): From the passage we know the high postage made ..
A. people never send each other letters
B. lovers almost lose every touch with each other
C. people try their best to avoid paying it
D. receivers refuse to pay the coming letters

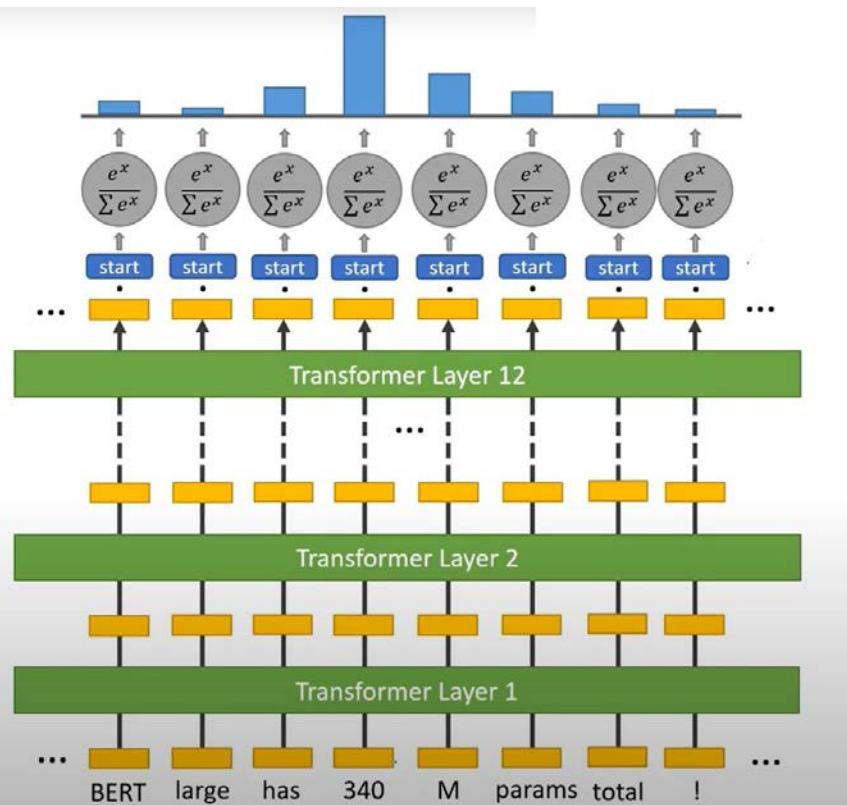
Answer: ADABC

Solutions -- classification

- We share the pre-training based methods.
- Methods are built upon BERT
- For classification / regression tasks,
 - Prepare a pretrained model
 - Try to re-use the parameters of the pretrained model and training on the downstream tasks
 - If >1 inputs, concatenate them
 - [cls] seq 1 [sep] seq 2 [sep]
 - Use smaller learning rate



QA



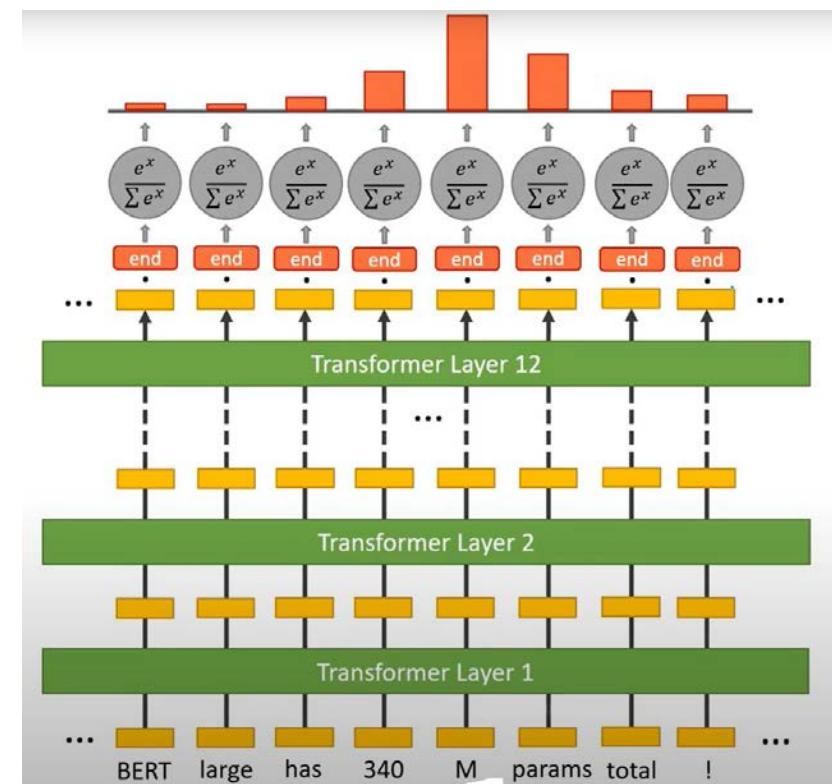
The answer is $\text{argmax } ST_i + ET_j$ where $j \geq i$

For null answer, then $S[\text{cls}] + E[\text{cls}]$ is the max value (with tolerance τ)

Question: How many parameters does BERT-large have?

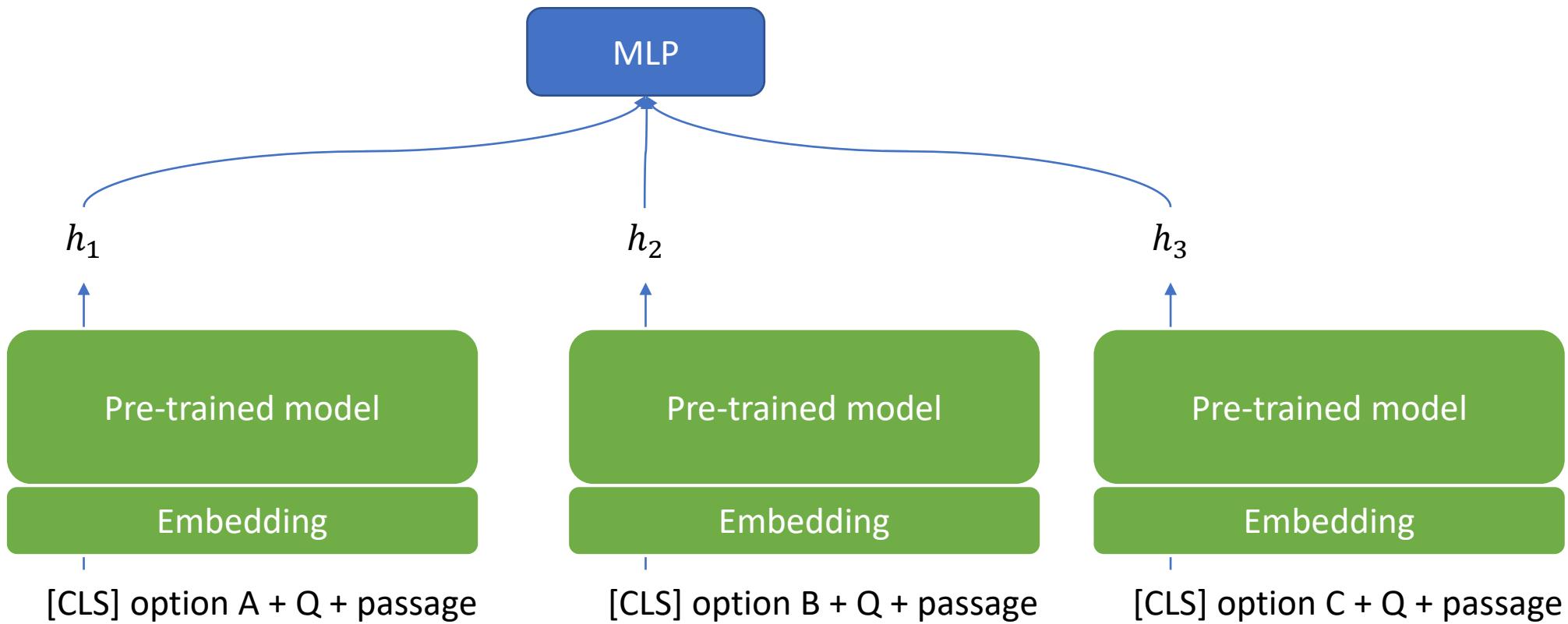
Reference Text:

BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.



*Images are extracted from
<https://www.youtube.com/watch?v=I8ZYCvgGu0o>*

Multiple choice

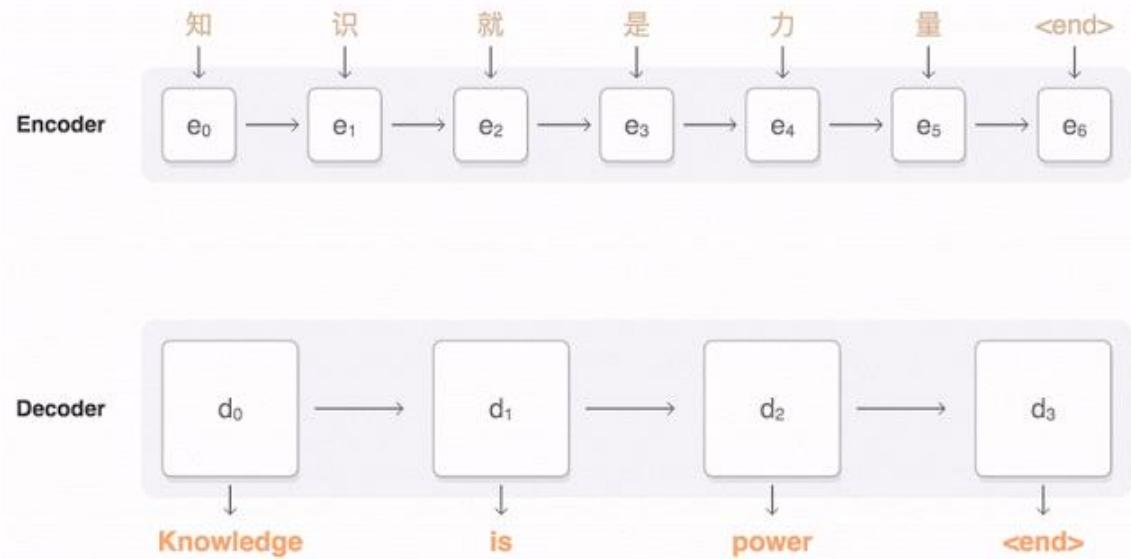




Part 4: Pretraining

Why pretraining

- Many NLP tasks → needs to design a model for each task
- Small training data → difficult to train a big DL model
- So can we leverage unlabeled data?



In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called “showers”.

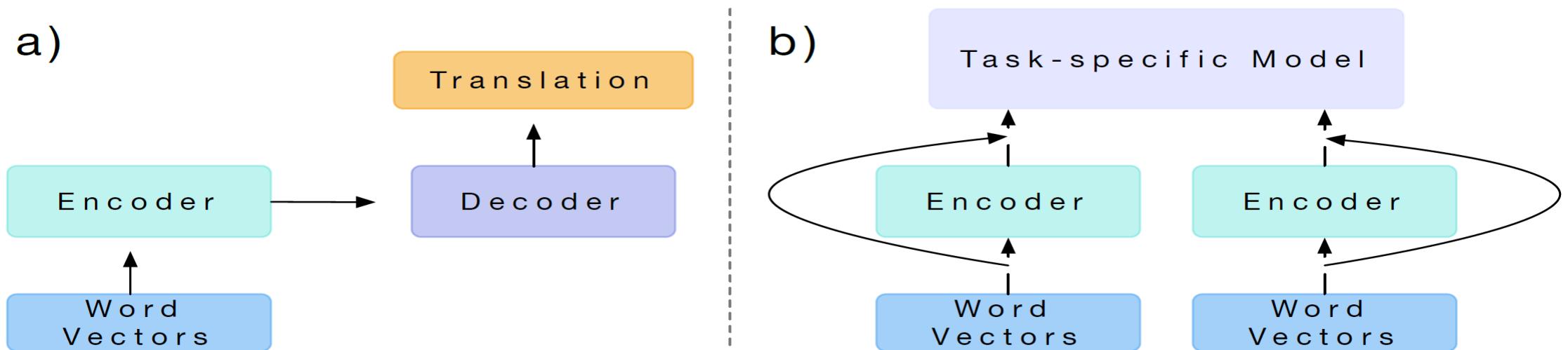
What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

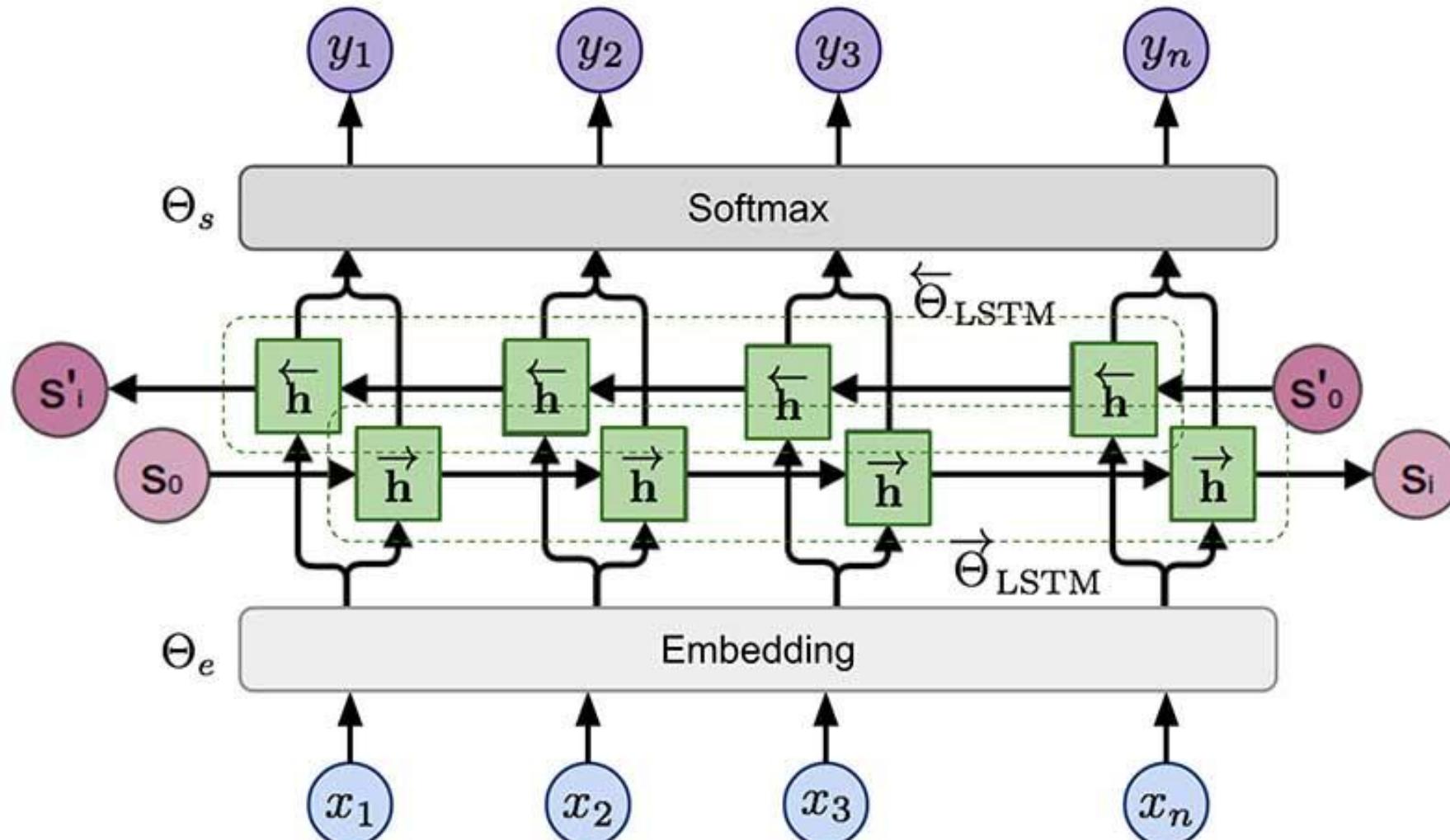
Where do water droplets collide with ice crystals to form precipitation?
within a cloud

CoVe: pretraining from machine translation

- Pretraining: deep LSTM based MT $\text{CoVe}(x) = \text{biLSTM}(\text{GloVe}(x))$
- Context vector applied to downstream tasks $v = [\text{GloVe}(x); \text{CoVe}(x)]$
- Sentiment analysis, question classification, entailment, question answering



ELMo: Embeddings from Language Model



Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. Deep contextualized word representations. NAACL 2018.

ELMo

- Pretraining objective
- ELMo representations

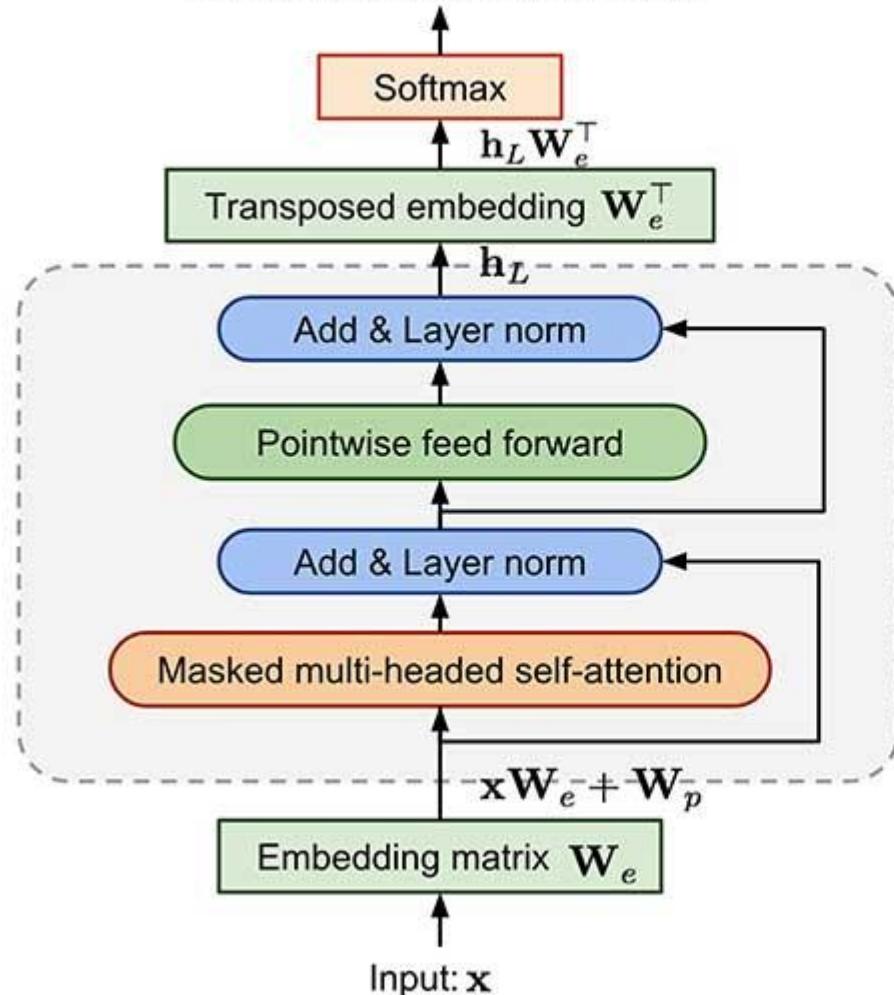
$$\mathcal{L} = - \sum_{i=1}^n \left(\log p(x_i \mid x_1, \dots, x_{i-1}; \Theta_e, \overrightarrow{\Theta}_{\text{LSTM}}, \Theta_s) + \log p(x_i \mid x_{i+1}, \dots, x_n; \Theta_e, \overleftarrow{\Theta}_{\text{LSTM}}, \Theta_s) \right)$$

$$v_i = f(R_i; \Theta^{\text{task}}) = \gamma^{\text{task}} \sum_{\ell=0}^L s_i^{\text{task}} \mathbf{h}_{i,\ell}$$

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE
SQuAD	Liu et al. (2017)	84.4	81.1	85.8
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17
SRL	He et al. (2017)	81.7	81.4	84.6
Coref	Lee et al. (2017)	67.2	67.2	70.4
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5

GPT: Generative Pre-training Transformer

Output: Probabilities over tokens



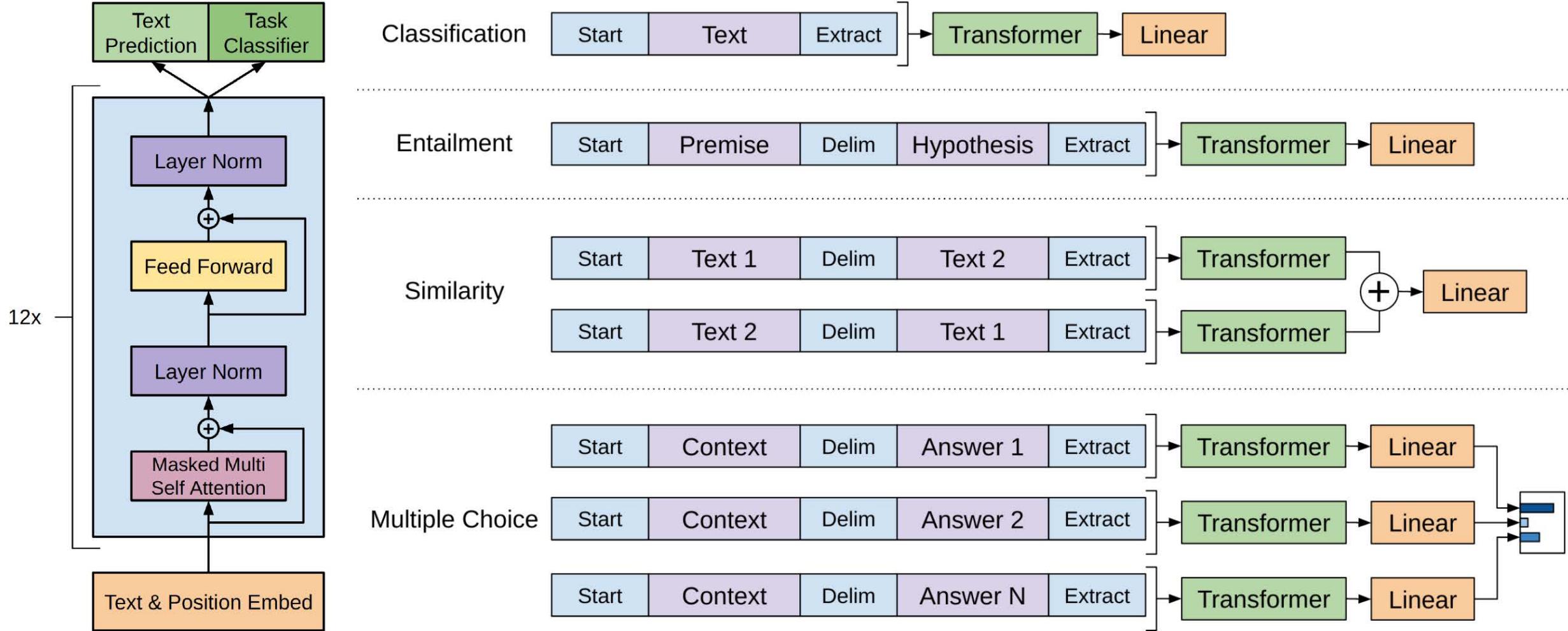
- Pretrain a deep Transformer decoder

$$\mathcal{L}_{\text{LM}} = - \sum_i \log p(x_i \mid x_{i-k}, \dots, x_{i-1})$$

Transformer Block
Repeat $\times L=12$

$$\begin{aligned} \mathbf{h}_\ell &= \text{transformer_block}(\mathbf{h}_{\ell-1}) \\ \ell &= 1, \dots, L \end{aligned}$$

Supervised fine-tuning



GPT family

- GPT-1
 - Improving Language Understanding by Generative Pre-Training
- GPT-2
 - Language Models are Unsupervised Multitask Learners
- GPT-3
 - Language Models are Few-Shot Learners
- GPT-4
 - Multimodal model (accepting image and text inputs, emitting text outputs)
- They are Transformer-based language models
 - + extremely large #parameters
 - + extremely large corpus

GPT-4o

May 13, 2024

Hello GPT-4o

We're announcing GPT-4o, our new flagship model that can reason across audio, vision, and text in real time.

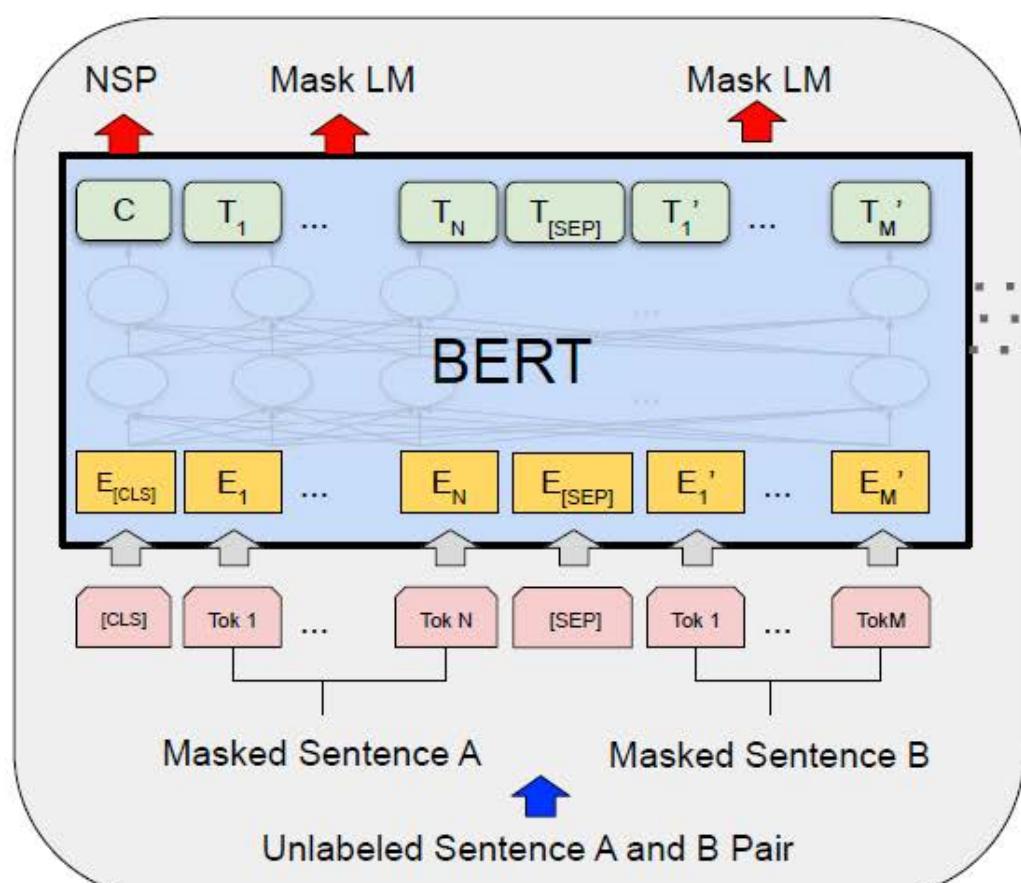
Contributions > Try on ChatGPT ↗ Try in Playground ↗ Rewatch live demos >



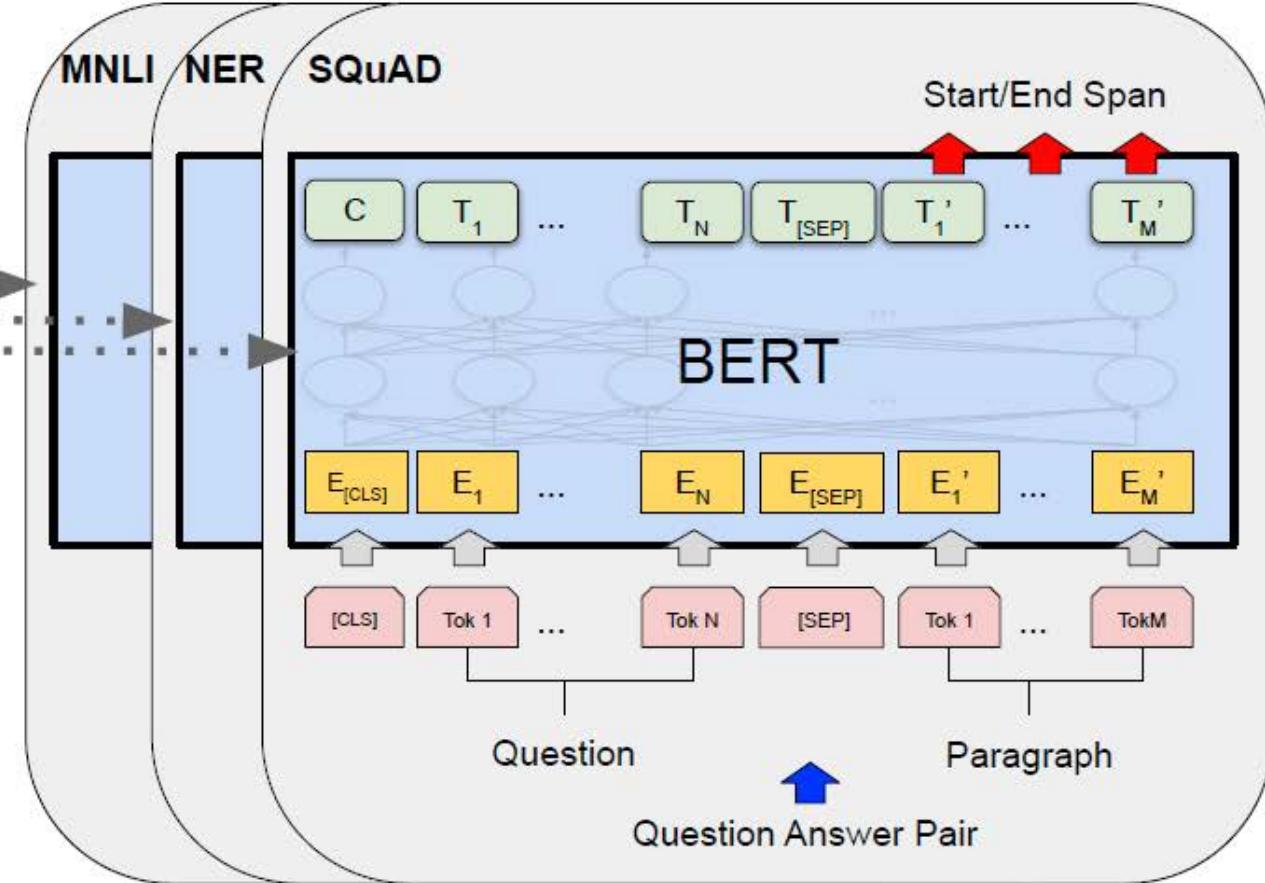
BERT: Bidirectional Encoder Representations from Transformers

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL 2019.

BERT pretraining and fine-tuning



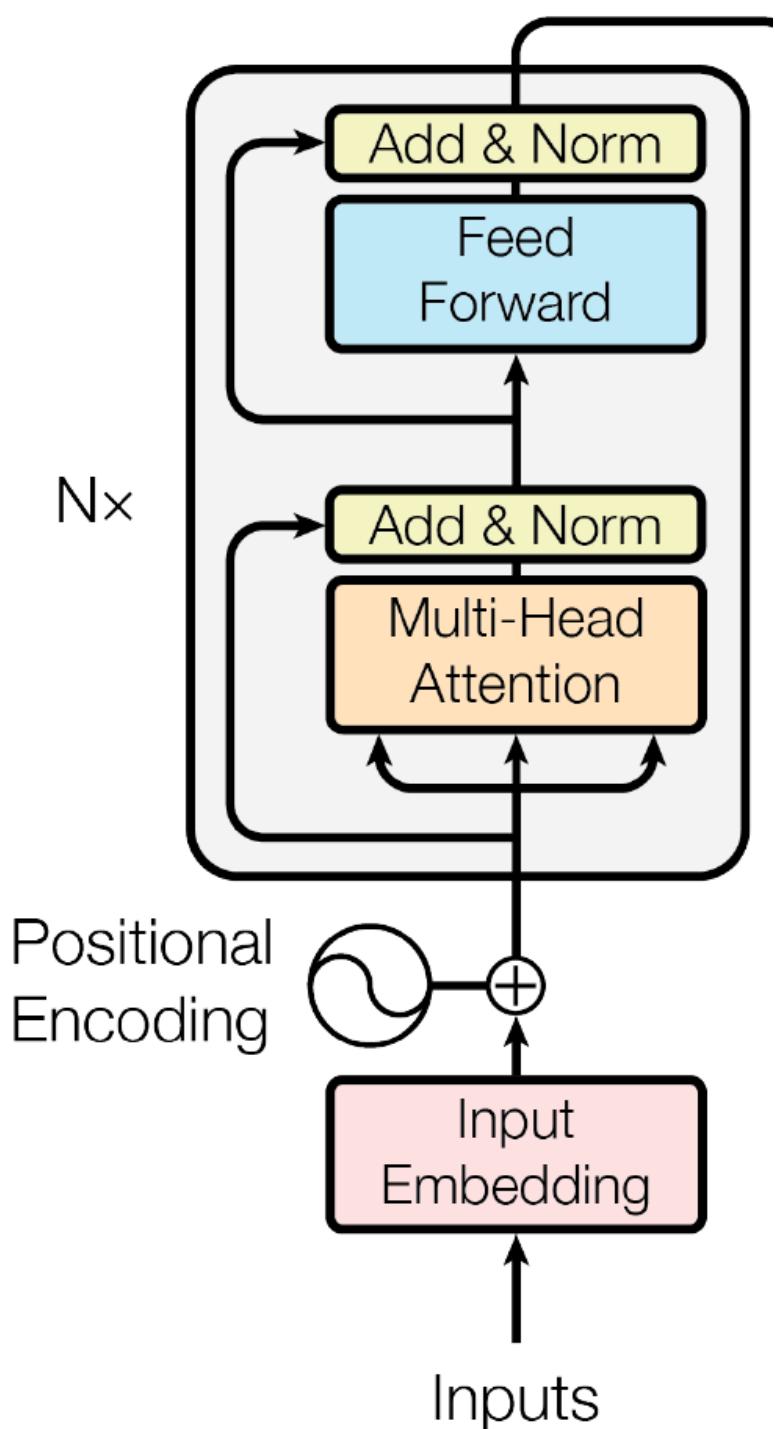
Pre-training



Fine-Tuning

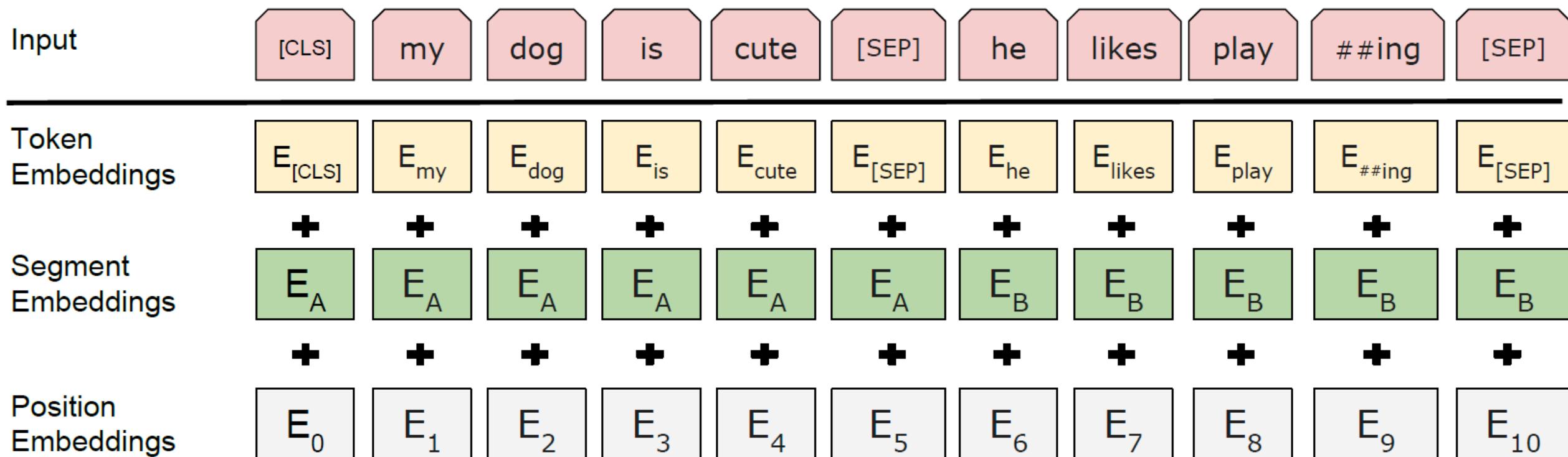
Pretraining tasks

- Mask language model (MLM)
 - Randomly mask 15% of tokens in each sequence
 - (a) with 80% probability, replace the chosen words with [MASK];
 - (b) with 10% probability, replace with a random word;
 - (c) with 10% probability, keep it the same.
 - The model only predicts the missing words
- Next sentence prediction
 - Sample sentence pairs (A, B) so that:
 - 50% of the time, B follows A;
 - 50% of the time, B does not follow A.
 - The model processes both sentences and output a binary label indicating whether B is the next sentence of A.

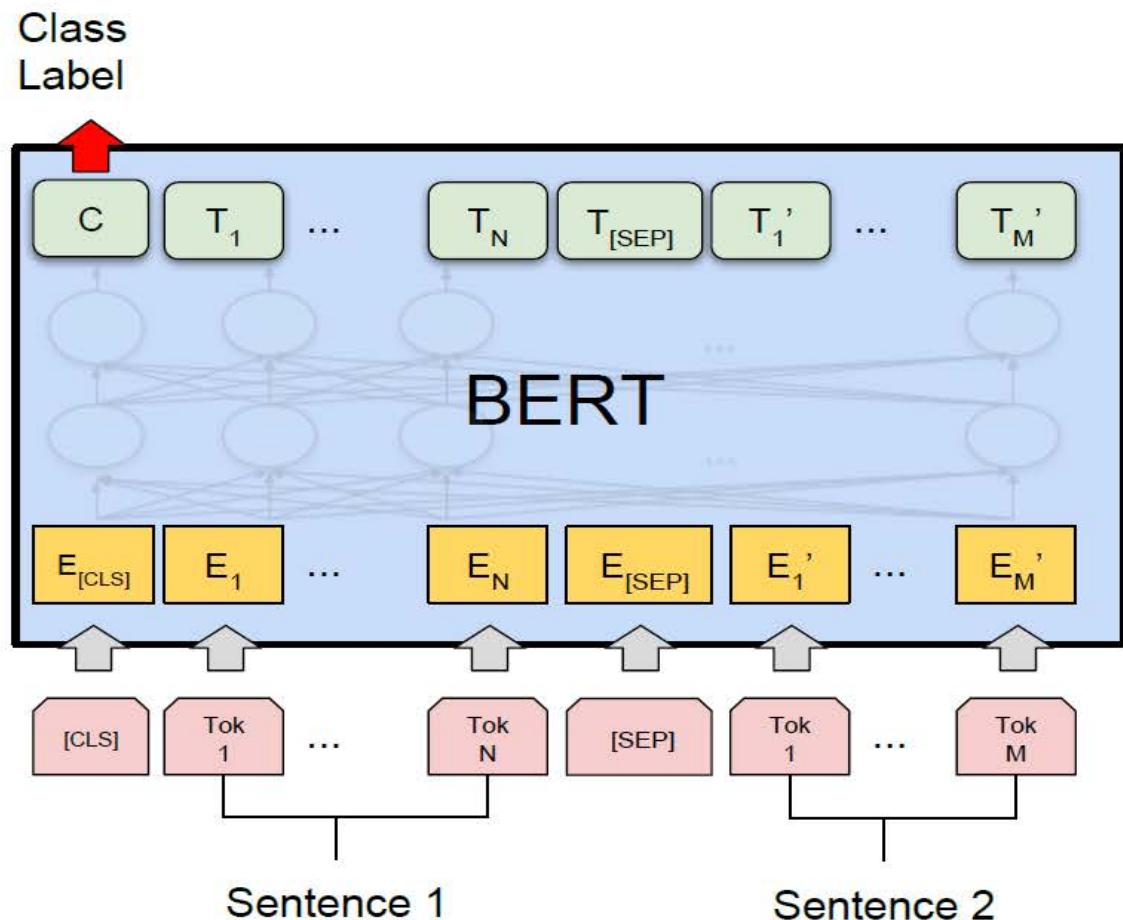


BERT input representations

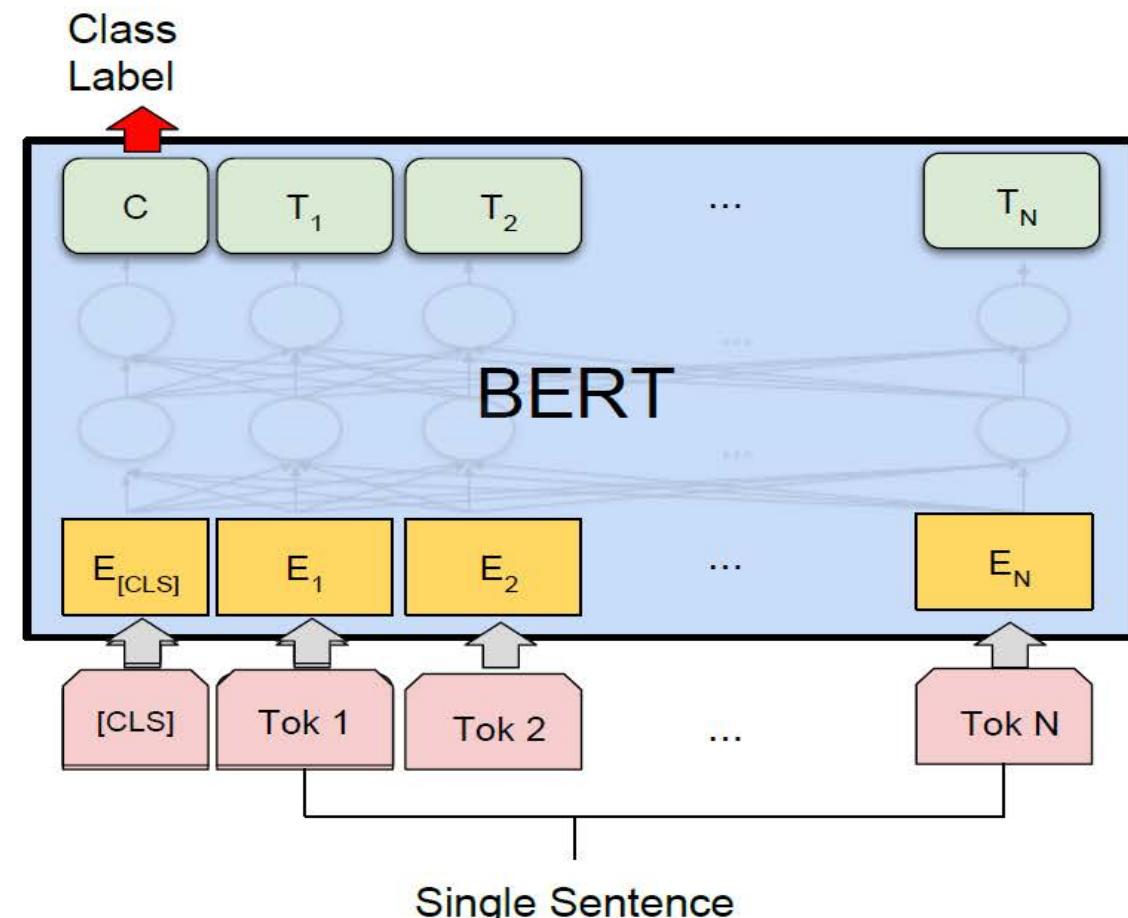
- Word pieces
- Segment embeddings
- Position embeddings



Fine-tuning

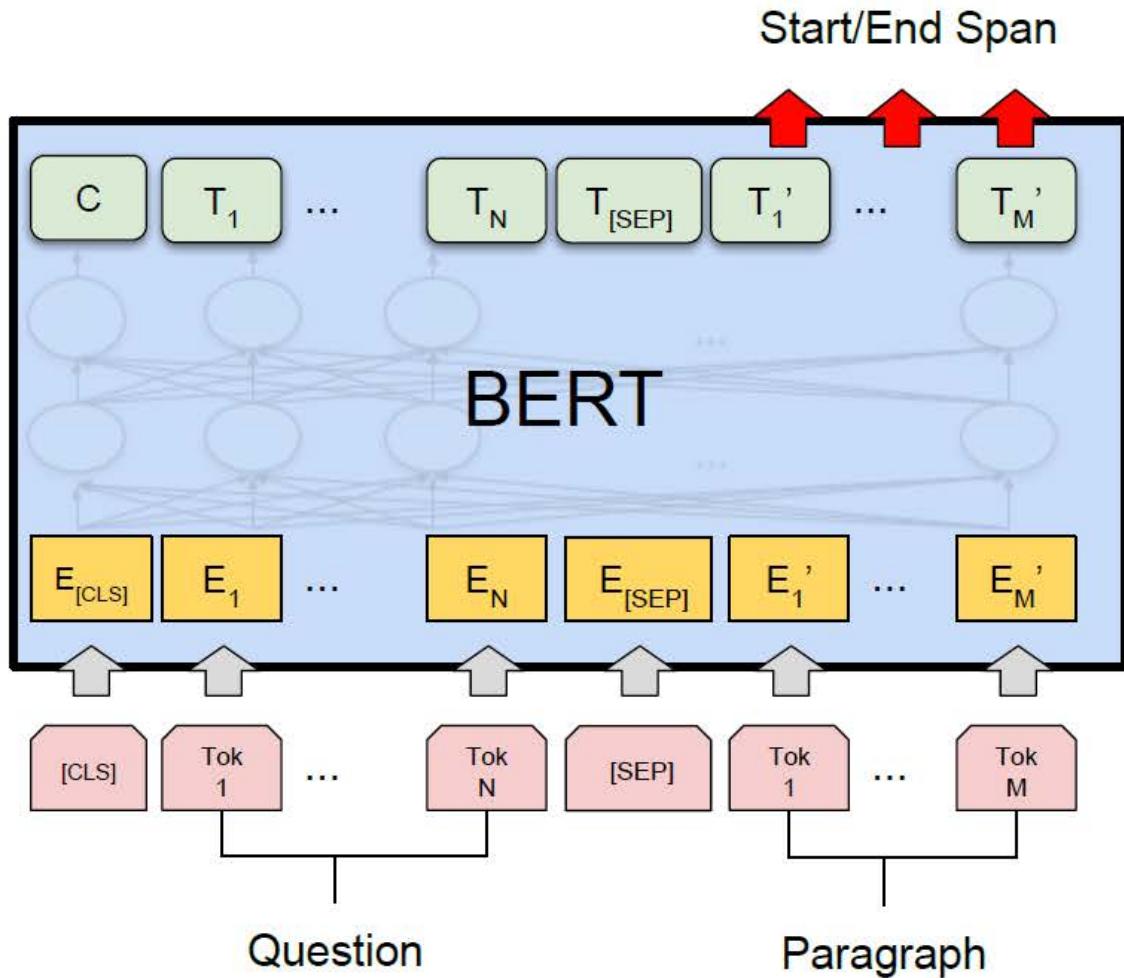


(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

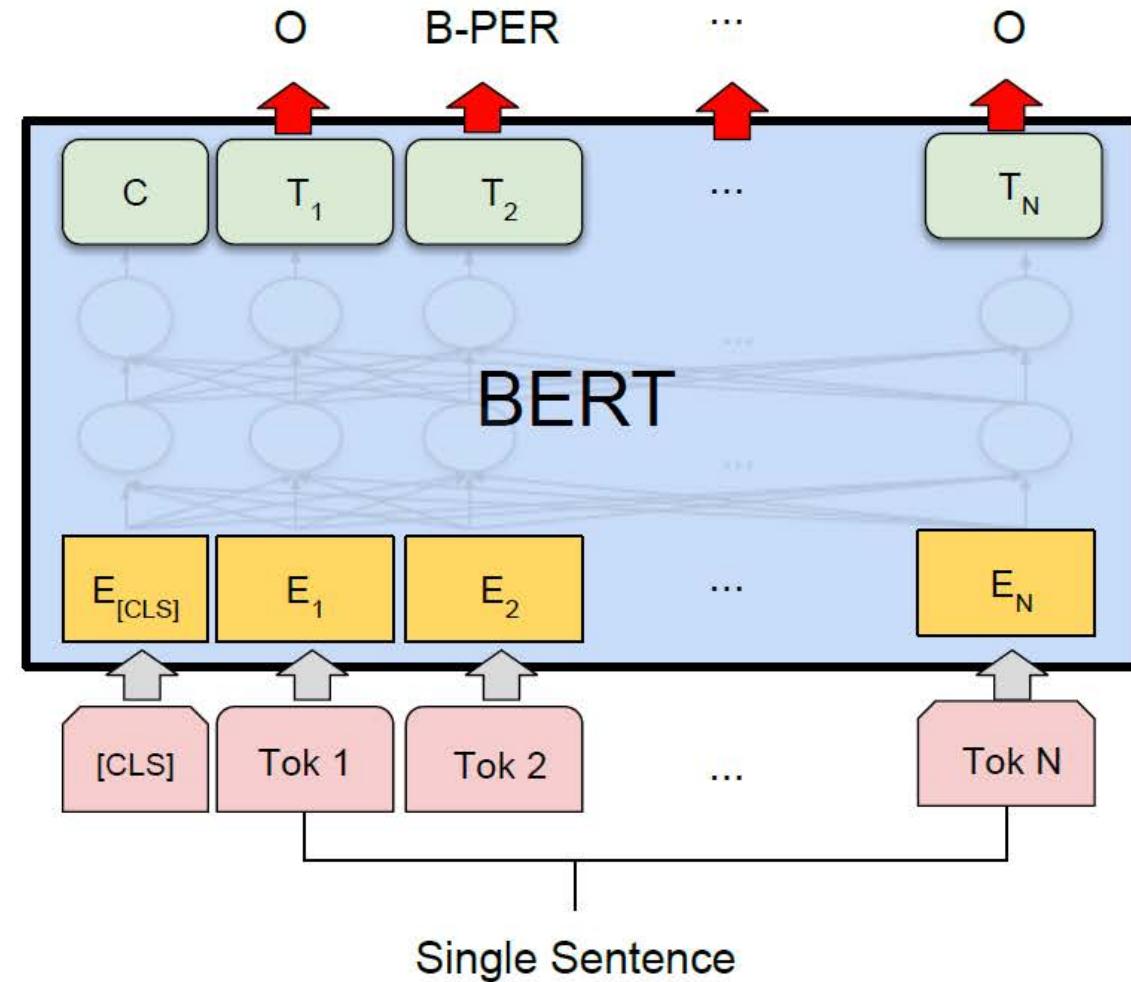


(b) Single Sentence Classification Tasks:
SST-2, CoLA

Fine-tuning



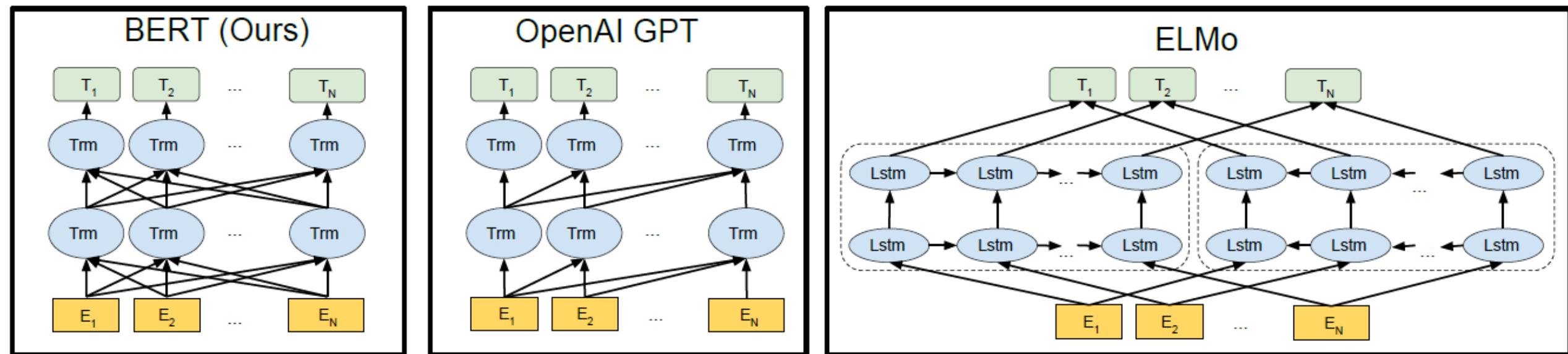
(c) Question Answering Tasks:
SQuAD v1.1



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

BERT vs. GPT/ELMo

- Train a large language model on free text and then fine-tune on specific tasks without customized network architectures.
- Make training bi-directional



GLUE results

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT _{BASE}	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

SQuAD results

SQuAD 1.1

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
BERT _{BASE} (Single)	80.8	88.5	-	-
BERT _{LARGE} (Single)	84.1	90.9	-	-
BERT _{LARGE} (Ensemble)	85.8	91.8	-	-
BERT _{LARGE} (Sgl.+TriviaQA)	84.2	91.1	85.1	91.8
BERT _{LARGE} (Ens.+TriviaQA)	86.2	92.2	87.4	93.2

SQuAD 2.0

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-	-	71.4	74.4
Ours				
BERT _{LARGE} (Single)	78.7	81.9	80.0	83.1

Ablation study

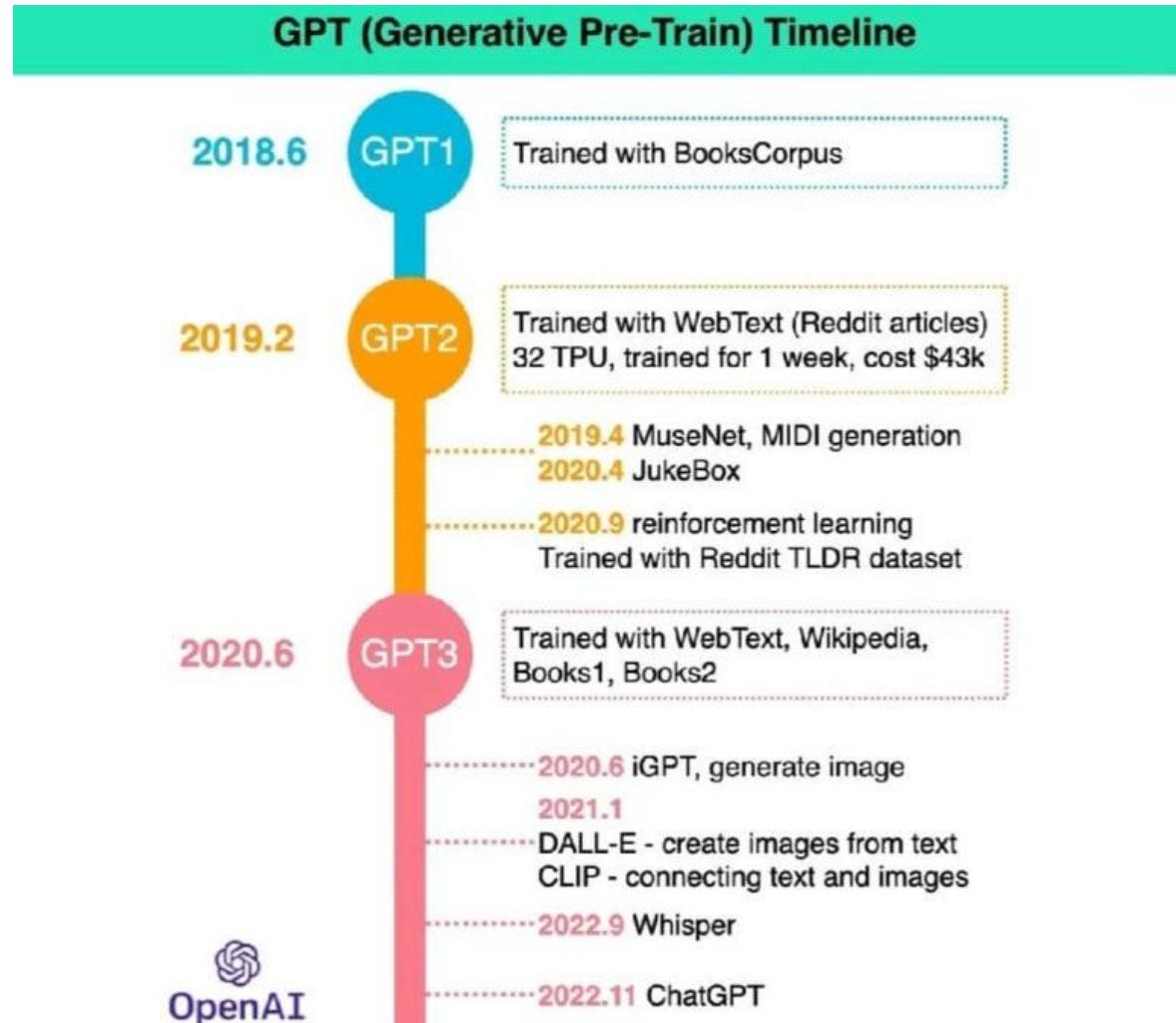
- “No NSP” is trained without the next sentence prediction task.
- “LTR & No NSP” is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT.
- “+ BiLSTM” adds a randomly initialized BiLSTM on top of the “LTR + No NSP” model during fine-tuning.

Tasks	Dev Set				
	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
BERT _{BASE}	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

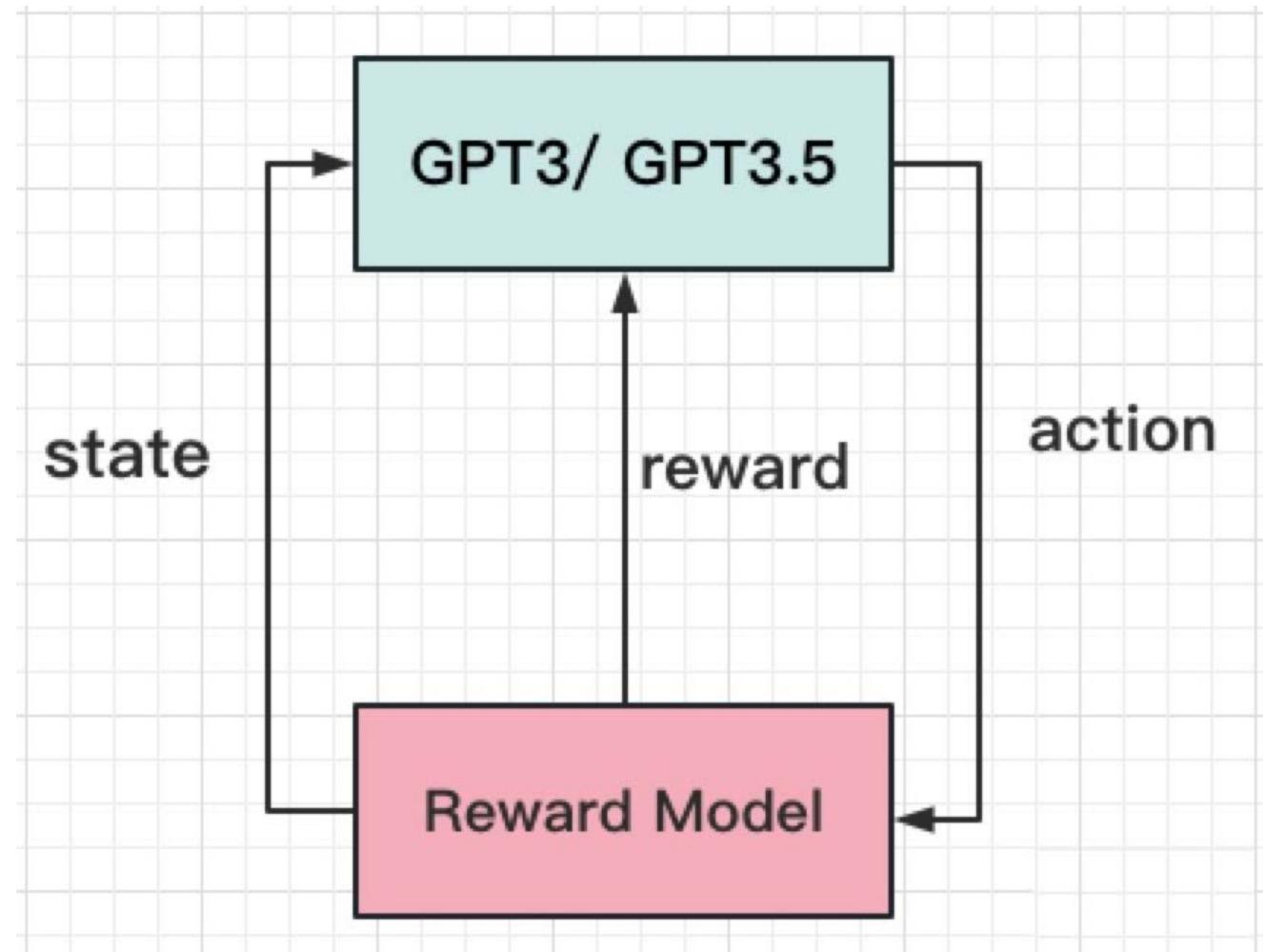
ChatGPT

Introducing ChatGPT

- We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.



RLHF (Reinforcement Learning from Human Feedback)

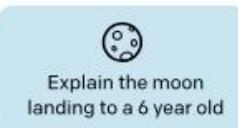


InstructGPT/ChatGPT

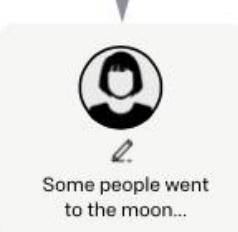
Step 1

Collect demonstration data, and train a supervised policy.

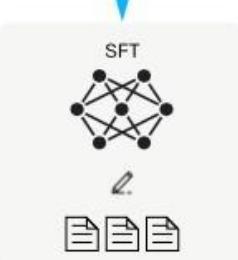
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



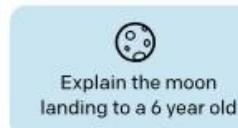
This data is used to fine-tune GPT-3 with supervised learning.



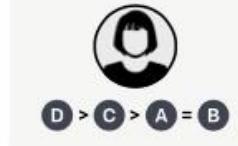
Step 2

Collect comparison data, and train a reward model.

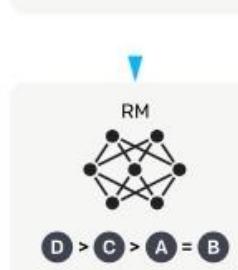
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



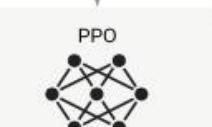
Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.



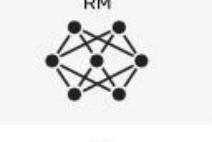
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.

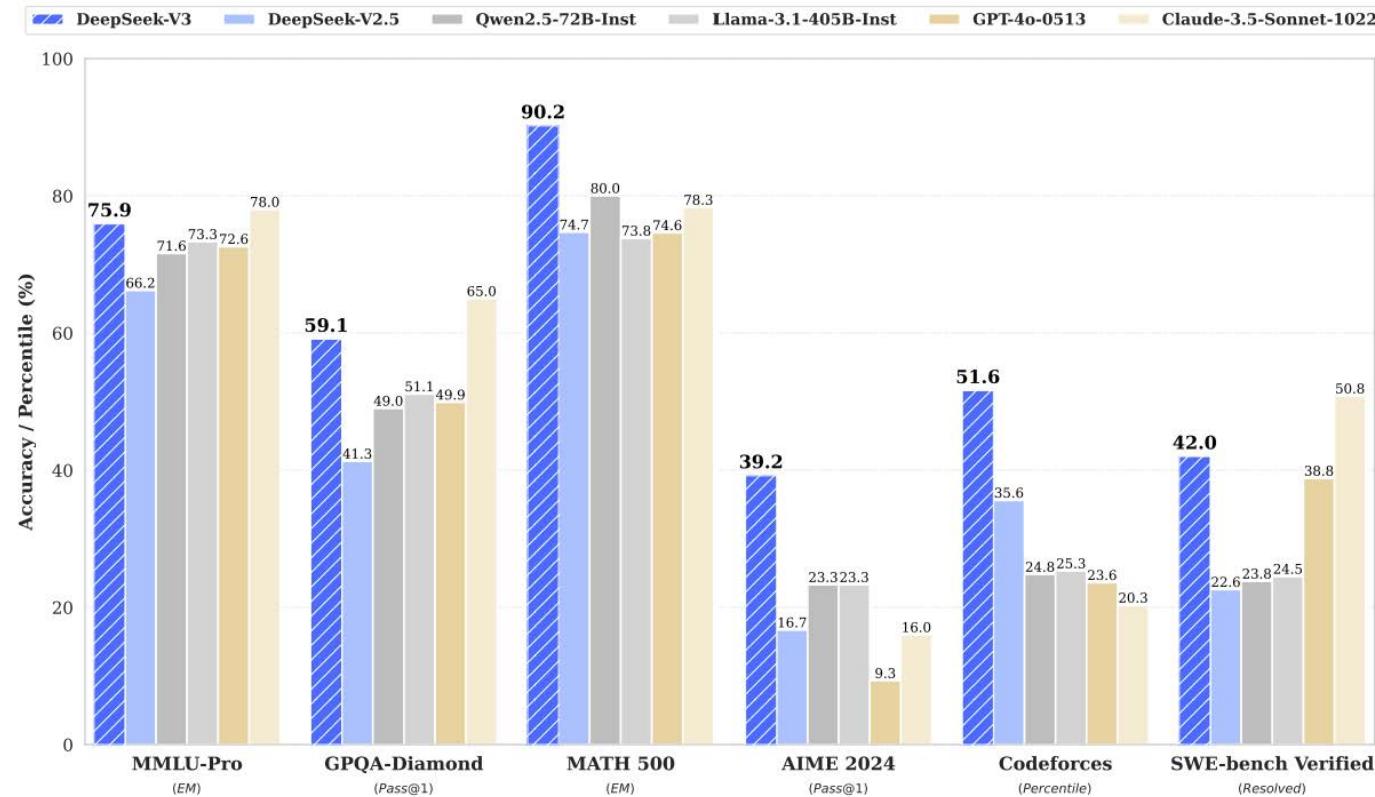


r_k

DeepSeek

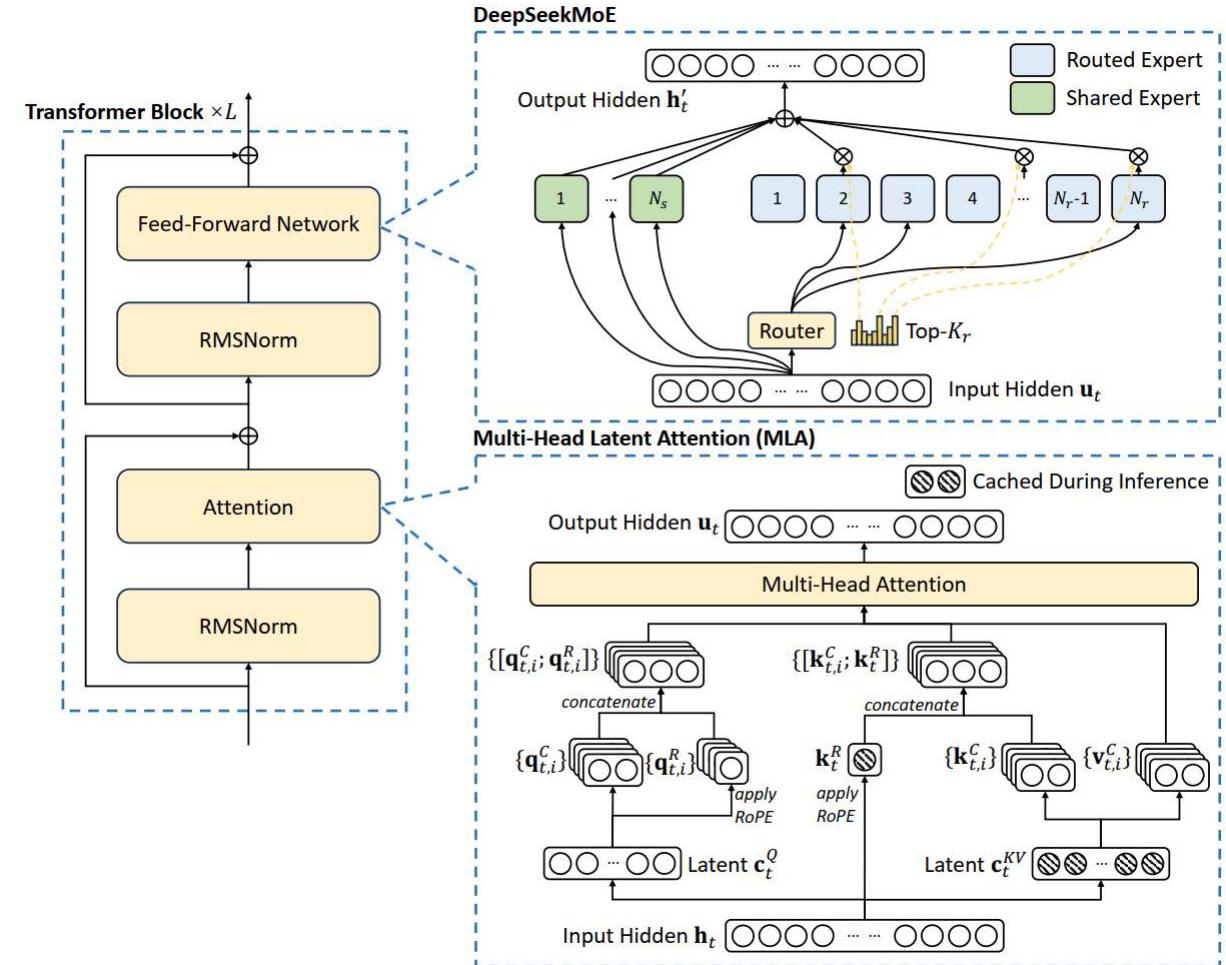
DeepSeek-V3

- DeepSeek-V3 is a strong Mixture-of-Experts (MoE) language model with 671B total parameters with 37B activated for each token.



Basic Architecture

- To achieve efficient inference and cost-effective training, DeepSeek-V3 adopts Multi-head Latent Attention (MLA) and DeepSeekMoE architectures, which were thoroughly validated in DeepSeek-V2.



Multi-Head Latent Attention

- The core of MLA is the low-rank joint compression for attention keys and values to reduce Key-Value (KV) cache during inference:

A diagram illustrating the decomposition of a compressed latent vector. A box labeled "compressed latent vector for keys and values" has a line pointing to the left side of the equation $\mathbf{c}_t^{KV} = W^{DKV} \mathbf{h}_t$. Below this, another equation shows the vector split into two parts: $[\mathbf{k}_{t,1}^C; \mathbf{k}_{t,2}^C; \dots; \mathbf{k}_{t,n_h}^C] = \mathbf{k}_t^C = W^{UK} \mathbf{c}_t^{KV}$, where \mathbf{k}_t^C is highlighted with a blue box. Further down, the vector is shown as $\mathbf{k}_t^R = \text{RoPE}(W^{KR} \mathbf{h}_t)$, where \mathbf{k}_t^R is also highlighted with a blue box. Finally, the full vector is given as $\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R]$, and below it, the equation $[\mathbf{v}_{t,1}^C; \mathbf{v}_{t,2}^C; \dots; \mathbf{v}_{t,n_h}^C] = \mathbf{v}_t^C = W^{UV} \mathbf{c}_t^{KV}$.

$$\boxed{\text{compressed latent vector for keys and values}} \rightarrow \boxed{\mathbf{c}_t^{KV}} = W^{DKV} \mathbf{h}_t,$$
$$[\mathbf{k}_{t,1}^C; \mathbf{k}_{t,2}^C; \dots; \mathbf{k}_{t,n_h}^C] = \mathbf{k}_t^C = W^{UK} \mathbf{c}_t^{KV},$$
$$\boxed{\mathbf{k}_t^R} = \text{RoPE}(W^{KR} \mathbf{h}_t),$$
$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R],$$
$$[\mathbf{v}_{t,1}^C; \mathbf{v}_{t,2}^C; \dots; \mathbf{v}_{t,n_h}^C] = \mathbf{v}_t^C = W^{UV} \mathbf{c}_t^{KV},$$

DeepSeekMoE

- DeepSeekMoE uses finer-grained experts and isolates some experts as shared ones.

$$\mathbf{h}'_t = \mathbf{u}_t + \sum_{i=1}^{N_s} \text{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t} \text{FFN}_i^{(r)}(\mathbf{u}_t),$$

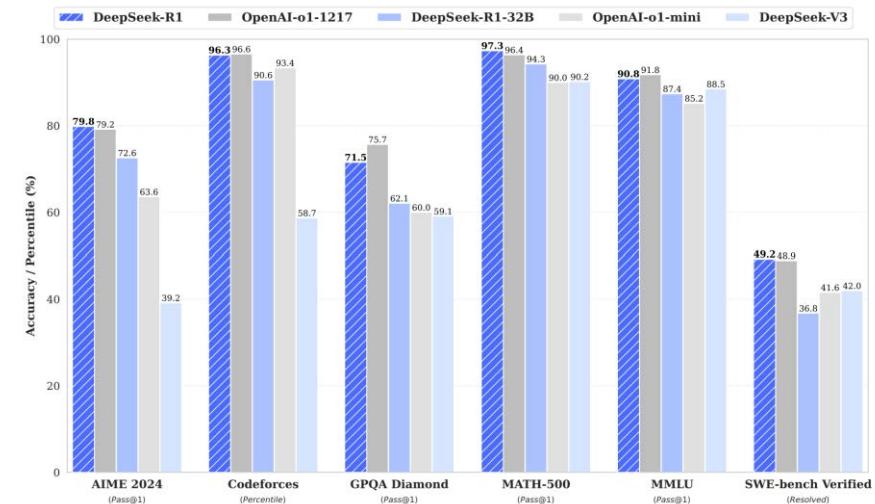
$$g_{i,t} = \frac{g'_{i,t}}{\sum_{j=1}^{N_r} g'_{j,t}},$$

$$g'_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{Topk}(\{s_{j,t} | 1 \leq j \leq N_r\}, K_r), \\ 0, & \text{otherwise,} \end{cases}$$

$$s_{i,t} = \text{Sigmoid}(\mathbf{u}_t^T \mathbf{e}_i),$$

DeepSeek-R1

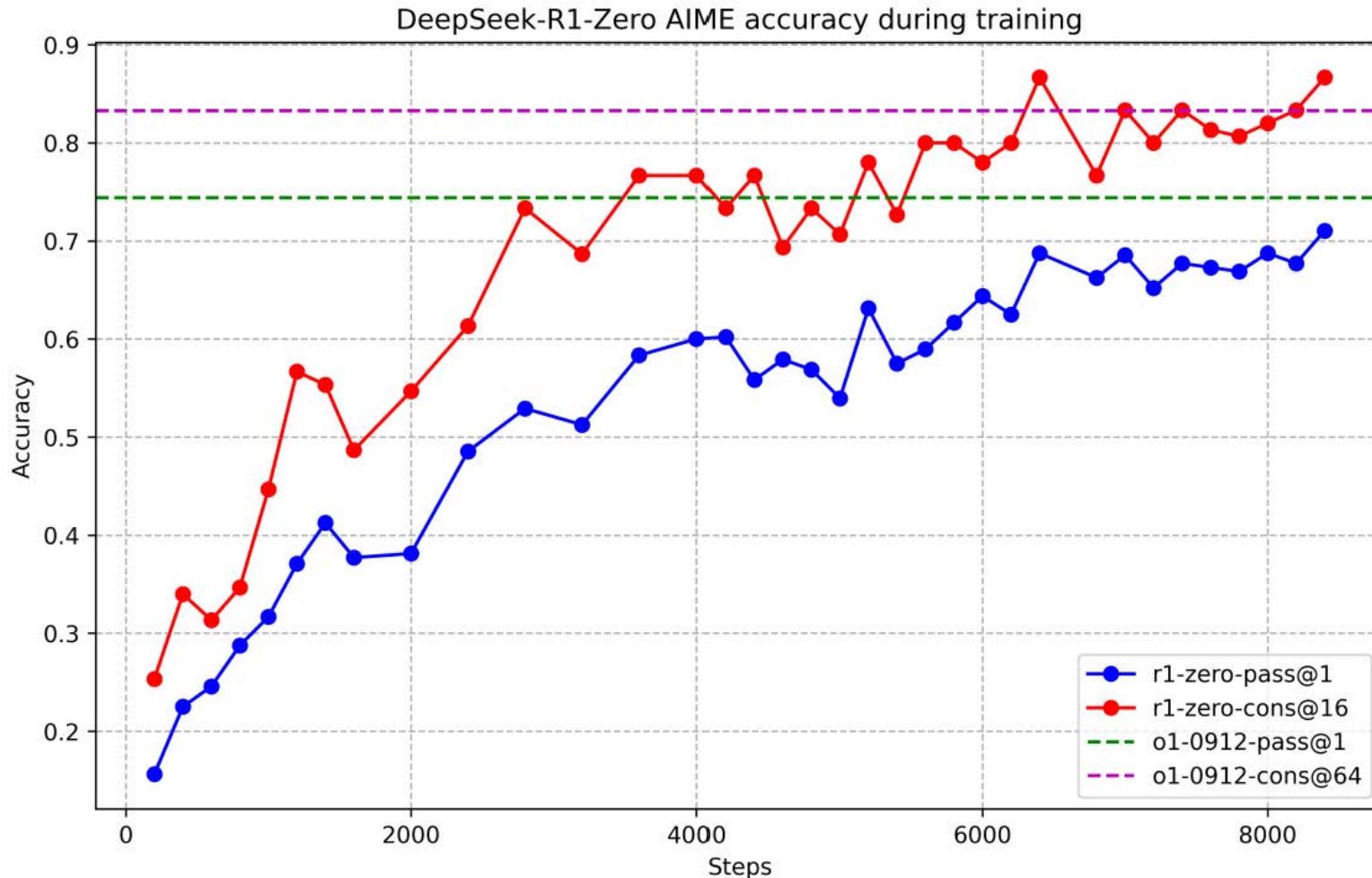
- DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities.
- DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeekR1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks.



Rule-Based Reward in RL

- **Accuracy rewards:** The accuracy reward model evaluates whether the response is correct. For example, in the case of math problems with deterministic results, the model is required to provide the final answer in a specified format (e.g., within a box), enabling reliable rule-based verification of correctness. Similarly, for LeetCode problems, a compiler can be used to generate feedback based on predefined test cases.
- **Format rewards:** In addition to the accuracy reward model, we employ a format reward model that enforces the model to put its thinking process between '<think>' and '</think>' tags.

Self-evolution Process



Takeaways

- Word embeddings
 - Word2vec
- Models for NLP
 - Transformer
- Pretraining
 - GPT
 - BERT
 - ChatGPT
 - DeepSeek

Key references

- Mikolov, T., Chen, K., Corrado, G., & Dean, J. Efficient estimation of word representations in vector space. arXiv:1301.3781.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. Distributed representations of words and phrases and their compositionality. NIPS 2013.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. Enriching Word Vectors with Subword Information. TACL 2017.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press. Chapter 10.
<http://www.deeplearningbook.org/contents/rnn.html>
- Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, 2016
- Bahdanau, D., Cho, K., & Bengio, Y. Neural machine translation by jointly learning to align and translate. ICLR 2015.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. Attention is all you need. NIPS 2017.
- McCann, B., Bradbury, J., Xiong, C., & Socher, R. Learned in Translation: Contextualized Word Vectors. NIPS 2017.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. Deep contextualized word representations. NAACL 2018.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL 2019.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. Language models are unsupervised multitask learners. OpenAI Blog, 2019.
- Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. Y. Mass: Masked sequence to sequence pre-training for language generation. ICML 2019.

Thanks!

- <http://web.ee.tsinghua.edu.cn/wqzhang>