文本表征学习

6. 大粒度文本单元表征

EE1513

宋彦

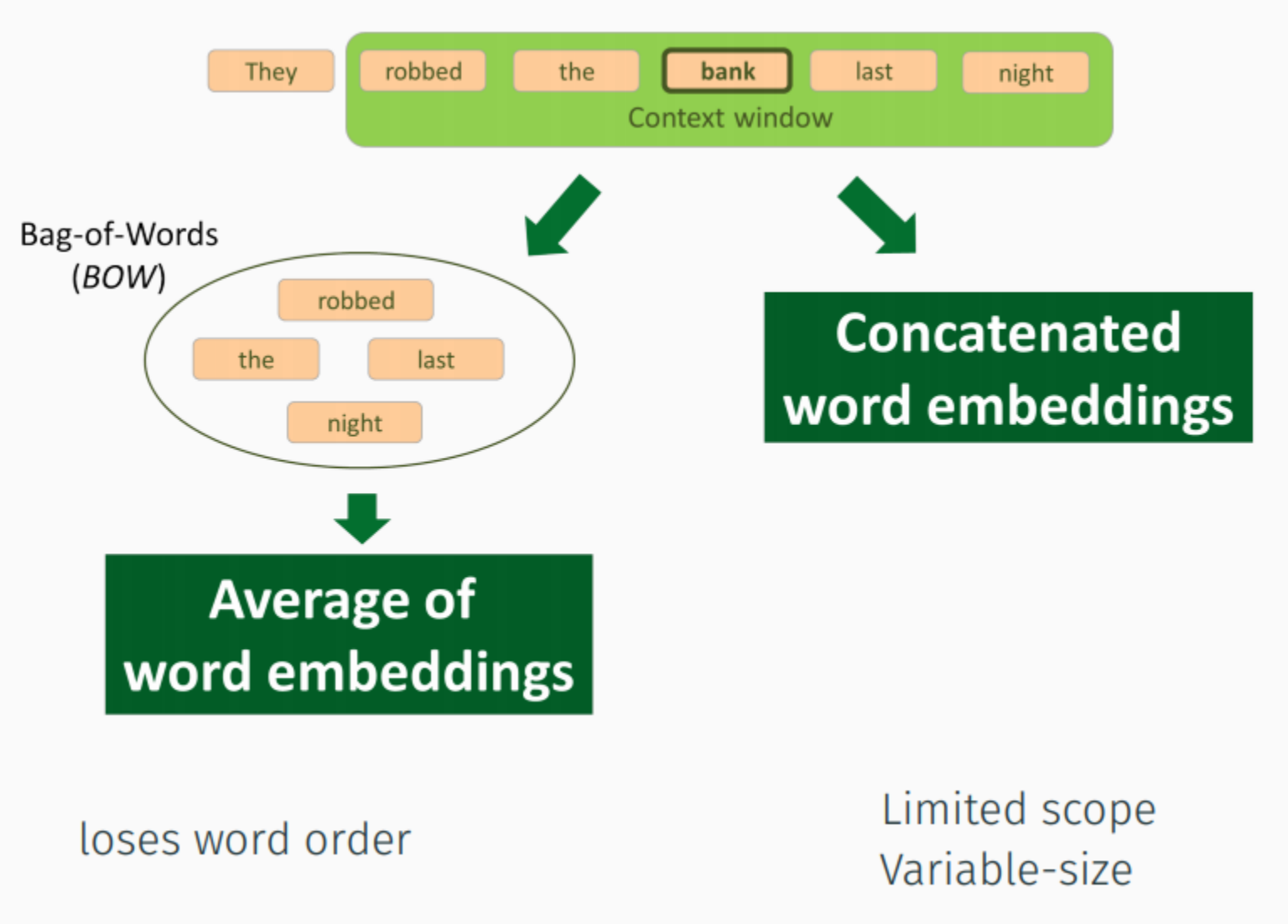
大纲

- 短语及句子的表征学习算法
  - Doc2vec
  - Context2vec
- 句子级语言模型
  - Skip-thought
- 从有监督的任务中学习文本表征
  - 自然语言推理（NLI）

为什么我们需要短语及句子级别的表征

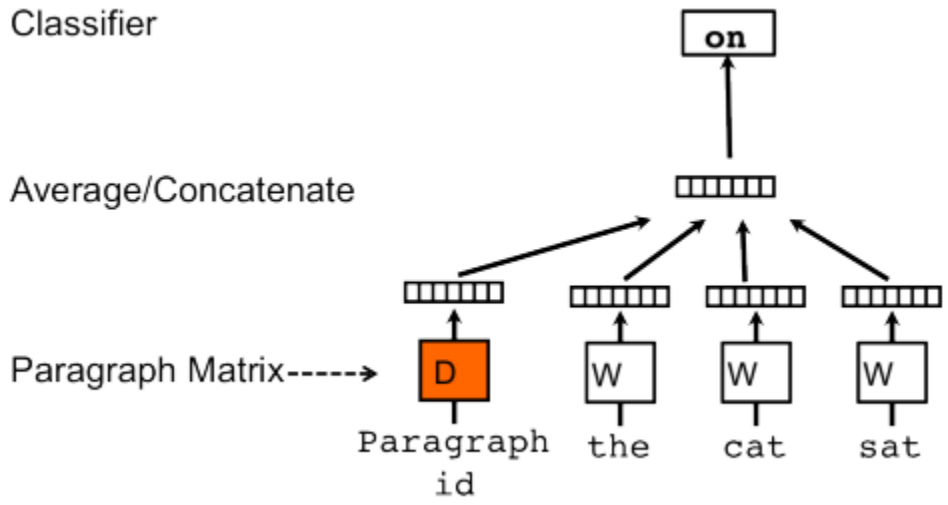
- 有很多句子级别的NLP任务
  - 情感分析
  - 文本分类
  - 摘要
  - ……
- 针对这些任务的机器学习算法需要把句子表征为一个向量
- 句子向量并非词向量的简单求和，而是包含了字所没有的信息
- 所以需要句子级别表征的学习算法

Doc2vec - 背景

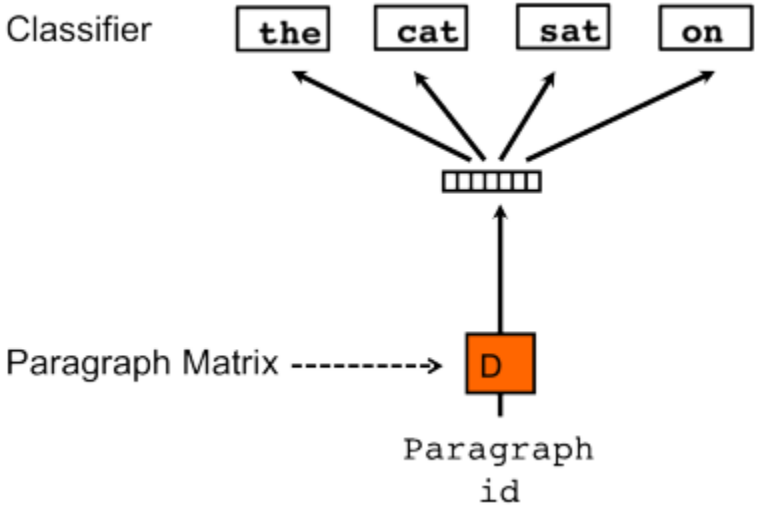

They
robbed
the
bank
last
night
Context window
Bag-of-Words
(BOW)
robbed
the
last
night
Concatenated
word embeddings
Average of
word embeddings
loses word order
Limited scope
Variable-size

Doc2vec – 概述

- Le and Mikolov (2014)
  - 受到word2vec算法的启发，引入一个特殊的词表示段落
  - 提出两种学习段落表征的方法
    - 段落表征向量的分布式记忆模型(PV-DM)
    - 段落表征向量的分布式词袋模型(PV-DBOW)



PV-DM

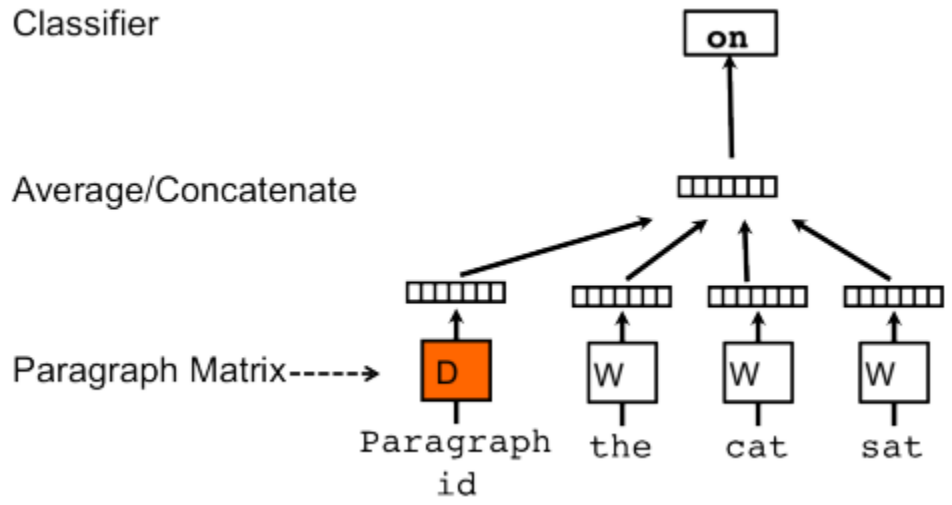


PV-DBOW

Le and Mikolov, Distributed Representations of Sentences and Documents, ICML-2014

Doc2vec - PV-DM

- PV-DM 训练
  - 使用滑动窗口选择上下文词
  - 特殊词和上下文词被用于预测下一个词
  - 模型存储所有学习的参数（包括词向量和分类器参数）



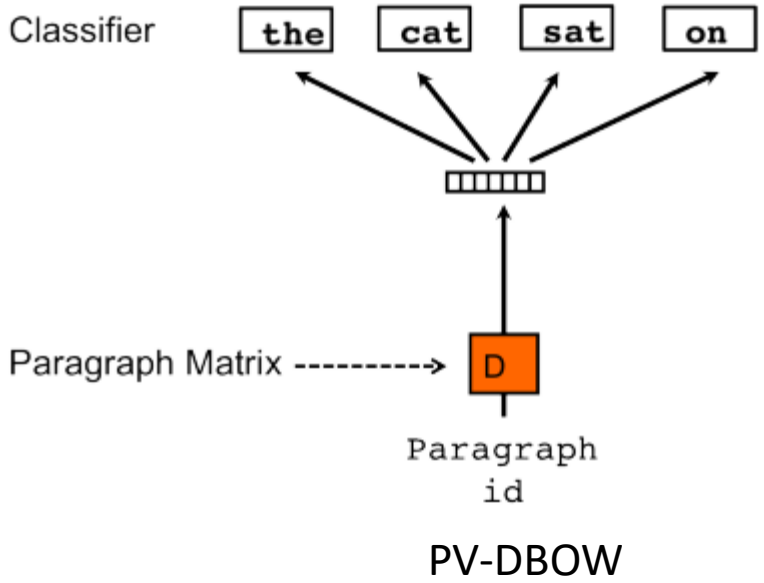
PV-DM

- PV-DM 预测
  - 随机初始化句子表征，固定所有模型参数，使用SGD更新句子表征直到收敛，得到测试句子的表征

Le and Mikolov, Distributed Representations of Sentences and Documents, ICML-2014

Doc2vec - PV-DBOW

- PV-DBOW 训练
  - 随机采样句子中的词，使用特殊词预测这些采样的词
  - 模型存储分类器中的参数



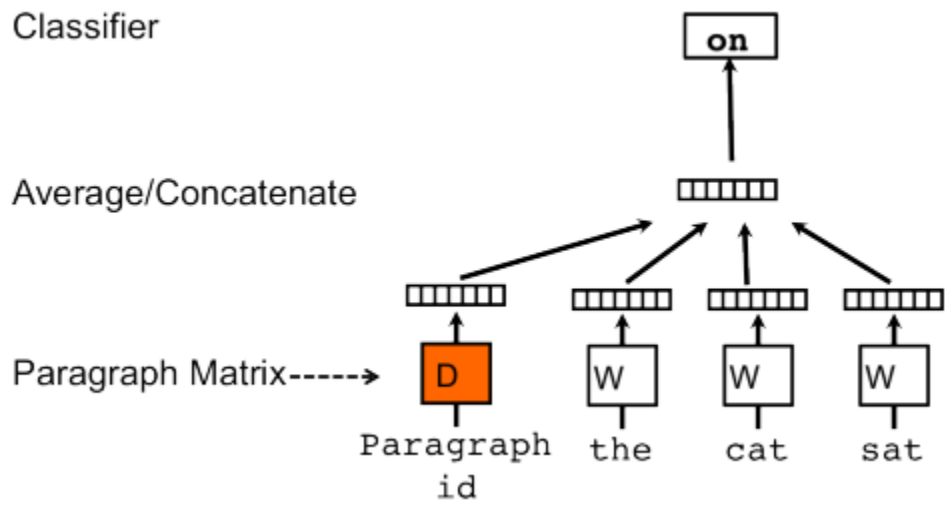
PV-DBOW

- PV-DBOW 预测
  - 随机初始化句子表征，固定所有模型参数，使用SGD更新句子表征直到收敛，得到测试句子的表征

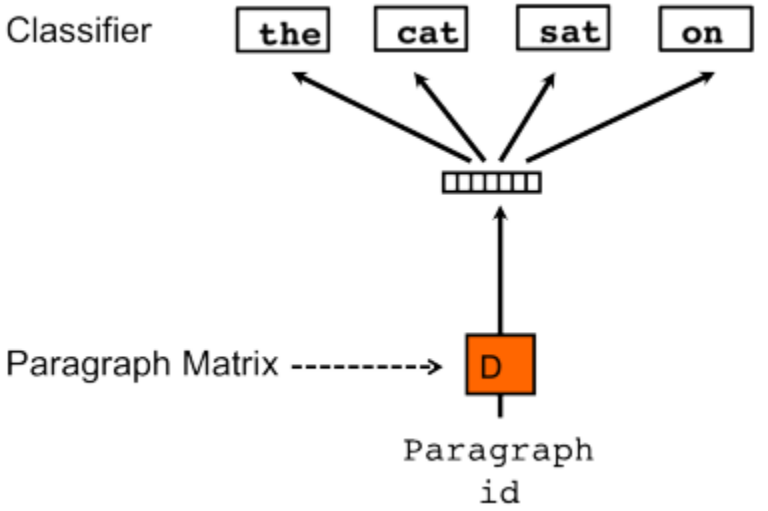
Le and Mikolov, Distributed Representations of Sentences and Documents, ICML-2014

Doc2vec – 问题

- 词序信息被忽视
- 需要SGD获取句子表征
- 因此需要更好的学习句子表征的方法



PV-DM



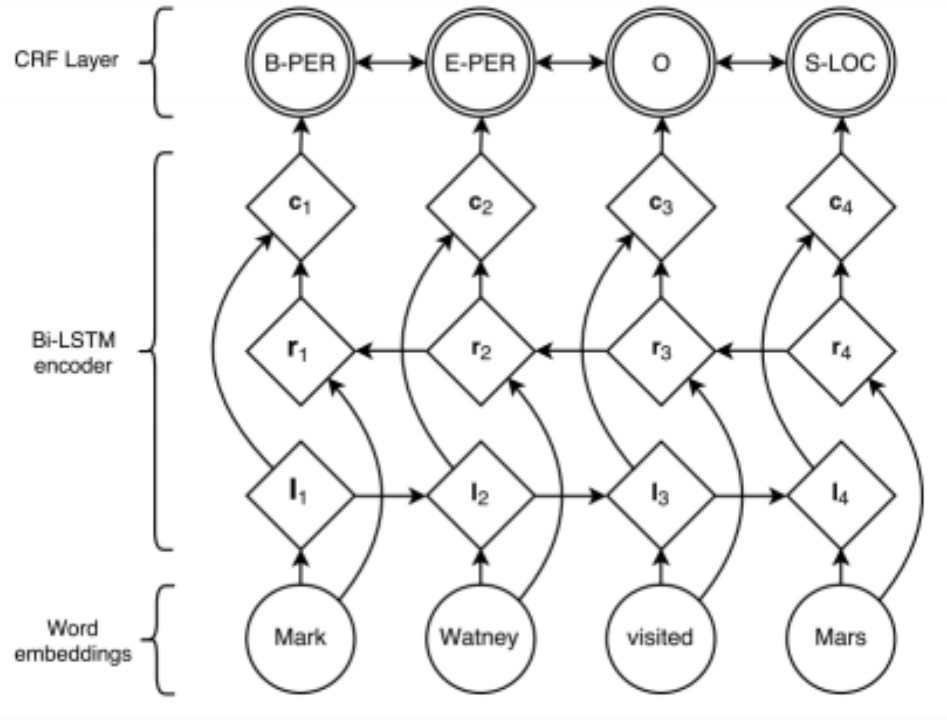
PV-DBOW

Le and Mikolov, Distributed Representations of Sentences and Documents, ICML-2014

Context2vec - 背景

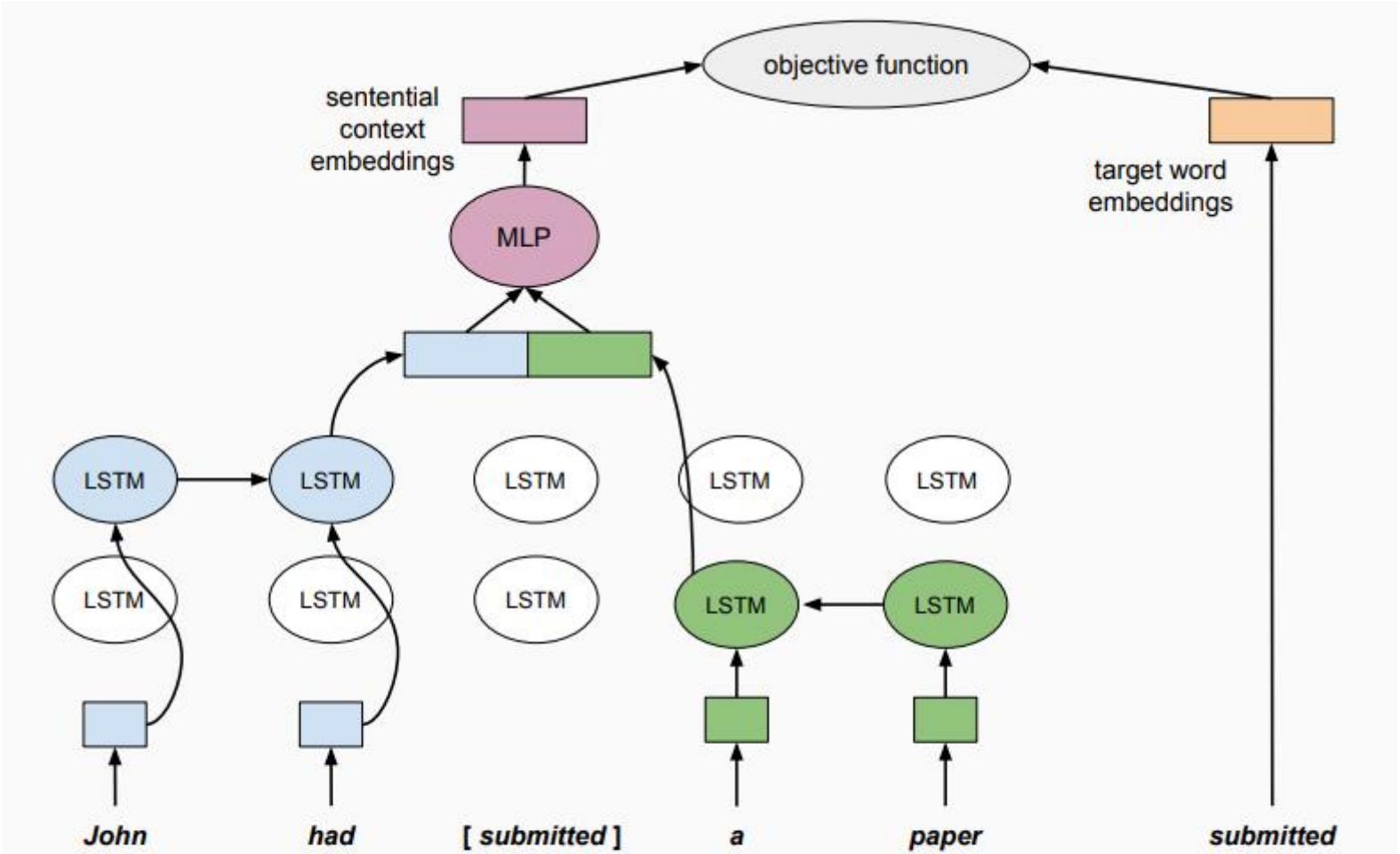
输入定长，用LSTM考虑语序关系

- 另一种学习句子表征的方法：
  - 使用BiLSTM编码词序
  - 与任务相关的训练
  - 使用词向量来记录从大规模语料中学习的信息



Context2vec

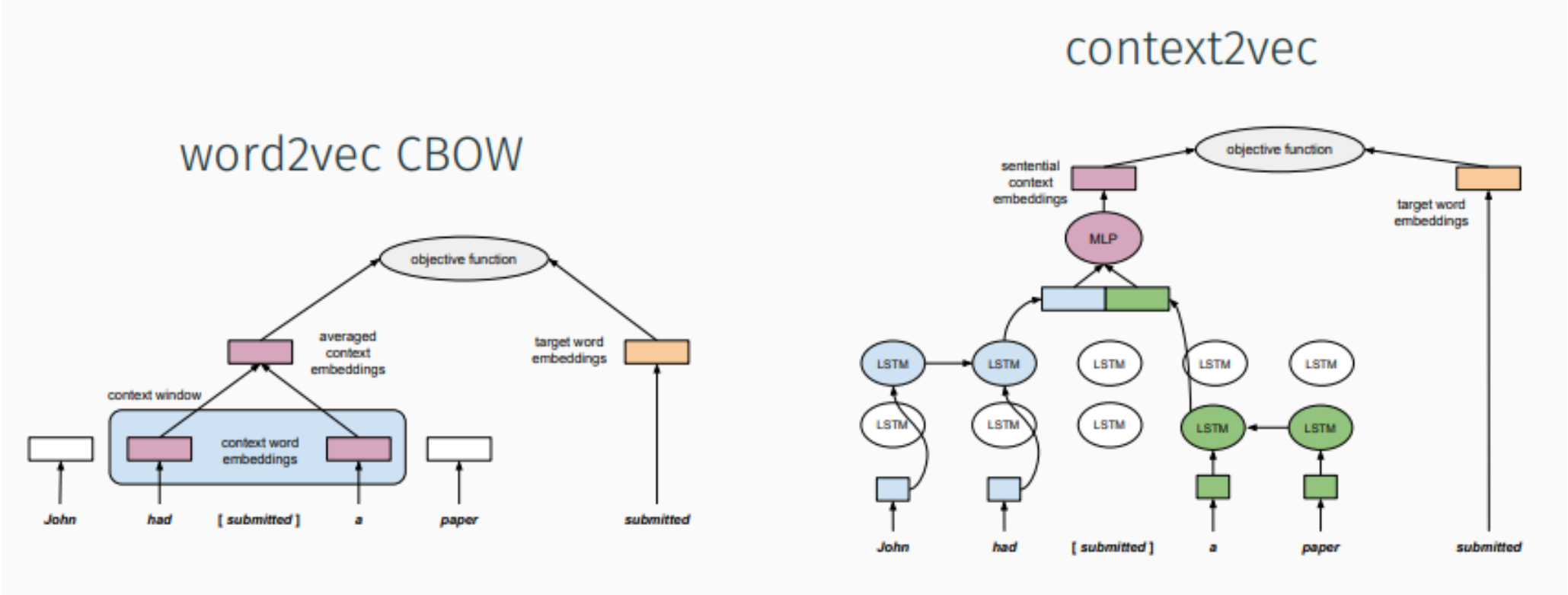
- Context2vec
  - 结构化上下文
  - 结构化样本



$$S = \sum_{(t,c)\in PAIRS} \left(\log \sigma(\vec{c}_{c2v} \cdot \vec{t}) + \sum_{t' \in NEGS_{(t,c)}} \log \sigma(-\vec{c}_{c2v} \cdot \vec{t'}) \right)$$

Context2vec

- 比较：context2vec = word2vec - CBOW + BiLSTM



Melamud et al., context2vec: Learning Generic Context Embedding with Bidirectional LSTM, CoNLL-2016

Context2vec - 评测

- 句子补全：使用一个词补全句子
- 词汇替换：给定词汇列表，选择列表中的一个词替换句子中的某个词
- 词义消歧：选择当前词在当前语境下的意义

TEST

This adds a wider perspective.

TRAIN

- They add (s2) a touch of humor.
- The minister added (s4) : the process remains fragile.

- Implementation: Shortest context-context cosine distance (kNN)

Context2vec – 评测

	c2v	Avg*
Sentence completion	**65.1**	49.7
LexSub (sample)	**56.0**	42.5
LexSub (all-words)	**47.9**	38.9
WSD	**72.8**	61.4

* Avg baseline:

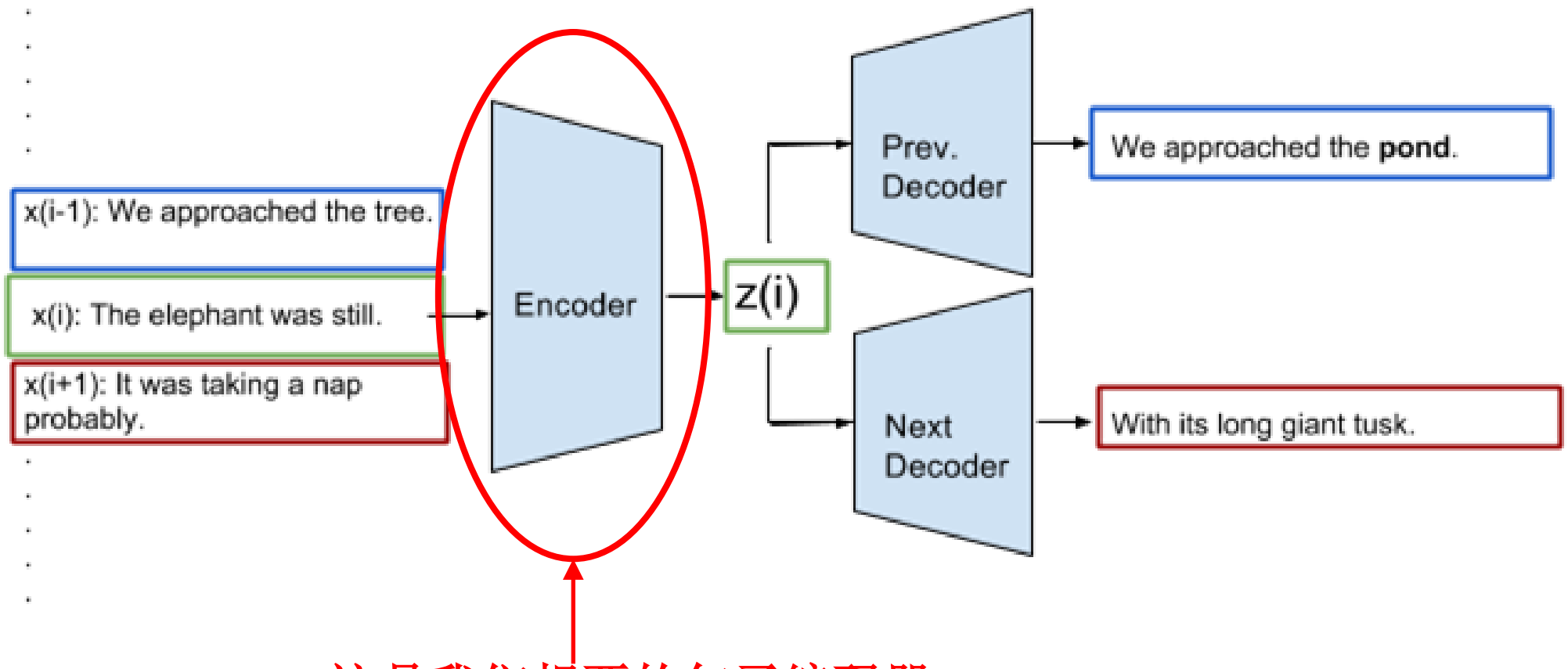
- Based on standard Skip-gram word embeddings
- Hyperparameters optimized: context size, word weights.

句子级语言模型

句子级语言模型

- 我们不能直接通过平均词向量的方式得到句子向量，因为句子中词的顺序对句义有很大影响
- Skip-Thought (Kiros, et al, 2015)：在词语言模型中，词义由其周围词决定。受其启发，句子的表征由其周围的句子决定
- Skip-Thought包括三个部分
  - 编码器 (RNN)
    - 输入：稀疏的句子表征
    - 输出：句子向量表征
  - 前句解码器(RNN)
    - 输入：句子向量表征
    - 输出：前一个句子的预测
  - 后句解码器(RNN)
    - 输入:句子向量表征
    - 输出：后一个句子的预测

Skip-Thought

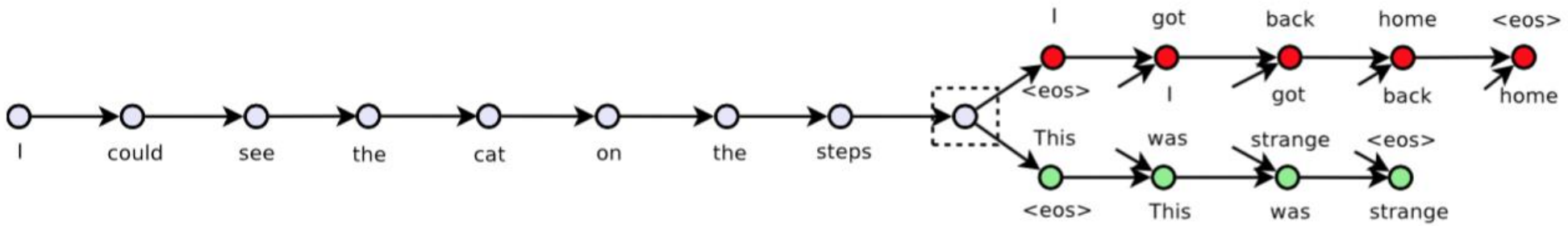


这是我们想要的句子编码器

Kiros et al., Skip-Thought Vectors, ICML-2015

Skip-Thought

- 给定包括三个连续句子的元组(s_{i-1}，s_i，s_{i+1})
  - s_i是第i个句子；s_{i-1} 是前一个句子；s_{i+1} 是后一个句子
- 使用下面的架构编码s_i和重建s_{i-1} 与 s_{i+1}



- 红色和绿色分别代表前一个句子和后一个句子的解码器

Skip-Thought

- 编码器
  - 设w^1，w^2，…，w^N为句子中的单词，N为该句子中的单词数。
  - 在每个时间步，编码器都会产生一个隐状态 h^t，它是序列 w^1，w^2，…，w^t 的表示。
  - 因此隐状态 h^N 代表完整的句子。
  - 然后我们迭代以下方程序列来编码句子。

$$
\begin{aligned}
\mathbf{r}^t &= \sigma(\mathbf{W}_r\mathbf{x}^t + \mathbf{U}_r\mathbf{h}^{t-1}) \\
\mathbf{z}^t &= \sigma(\mathbf{W}_z\mathbf{x}^t + \mathbf{U}_z\mathbf{h}^{t-1}) \\
\bar{\mathbf{h}}^t &= \tanh(\mathbf{W}\mathbf{x}^t + \mathbf{U}(\mathbf{r}^t \odot \mathbf{h}^{t-1})) \\
\mathbf{h}^t &= (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t
\end{aligned}
$$

$\bar{h}^t$ 是时刻t时，提出的更新向量

z^t 是更新门

r^t 是重置门

$\odot$ 是逐个成分的成绩

Skip-Thought

- 解码器
  - 解码器是一个神经网络语言模型，其给定条件是编码器输出
  - 解码器的计算与编码器类似，只是引入矩阵 Cz、Cr、C，它们用于通过更新门、重置门和隐状态计算。
  - 每个解码器使用单独的参数
  - 连接解码器隐藏状态的权重矩阵共用一套参数，用于计算单词的分布
  - 解码通过下面的方程序列迭代

$$
\begin{aligned}
\mathbf{r}^t &= \sigma(\mathbf{W}_r^d \mathbf{x}^{t-1} + \mathbf{U}_r^d \mathbf{h}^{t-1} + \mathbf{C}_r \mathbf{h}_i) \\
\mathbf{z}^t &= \sigma(\mathbf{W}_z^d \mathbf{x}^{t-1} + \mathbf{U}_z^d \mathbf{h}^{t-1} + \mathbf{C}_z \mathbf{h}_i) \\
\bar{\mathbf{h}}^t &= \tanh(\mathbf{W}^d \mathbf{x}^{t-1} + \mathbf{U}^d (\mathbf{r}^t \odot \mathbf{h}^{t-1}) + \mathbf{C}\mathbf{h}_i) \\
\mathbf{h}_{i+1}^t &= (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t
\end{aligned}
$$

h^t_{i+1} 是 t 时刻的隐向量

i 表示当前句子

i+1 表示下一个句子

Skip-Thought

- 目标函数
  - 针对每个样本，损失是前一句子与后一句子损失的和
  - 总损失是所有训练样本损失之和

$$\sum_t \log P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) + \sum_t \log P(w_{i-1}^t | w_{i-1}^{<t}, \mathbf{h}_i)$$

Skip-Thought 实验

- 语义相关性
  - 在 SemEval shared task 上训练
  - 使用简单的逻辑回归
- 结果与有监督的方法类似，超越了除去Dependency Tree LSTM 之外的所有方法

Method	r	ρ	MSE
Illinois-LH [18]	0.7993	0.7538	0.3692
UNAL-NLP [19]	0.8070	0.7489	0.3550
Meaning Factory [20]	0.8268	0.7721	0.3224
ECNU [21]	0.8414	–	–
Mean vectors [22]	0.7577	0.6738	0.4557
DT-RNN [23]	0.7923	0.7319	0.3822
SDT-RNN [23]	0.7900	0.7304	0.3848
LSTM [22]	0.8528	0.7911	0.2831
Bidirectional LSTM [22]	0.8567	0.7966	0.2736
Dependency Tree-LSTM [22]	**0.8676**	**0.8083**	**0.2532**
bow	0.7823	0.7235	0.3975
uni-skip	0.8477	0.7780	0.2872
bi-skip	0.8405	0.7696	0.2995
combine-skip	0.8584	0.7916	0.2687
combine-skip+COCO	0.8655	0.7995	0.2561

Skip-Thought 的特点及用处

- 特点
  - 非监督方法训练
  - 通用性强
  - 非常直接的句子级语言模型
  - 对噪音有很强的容忍性
    - 无需深度清洗训练数据
  - 可以创造的表征直接有用
- 用处
  - 它被设计用于普适性应用
  - 在很多任务上不一定可以超越最好的模型
  - 在文本处理流程中可能会有用处
    - 可以继续进一步预训练
    - 对噪音有一定的容忍性
  - 未来研究
    - 探究方法局限的解决方法
    - 使用更大的上下文、网络结构和表征空间

前面不已经是监督了吗？
哦，上下文为样本，不是标签，所以是无监督

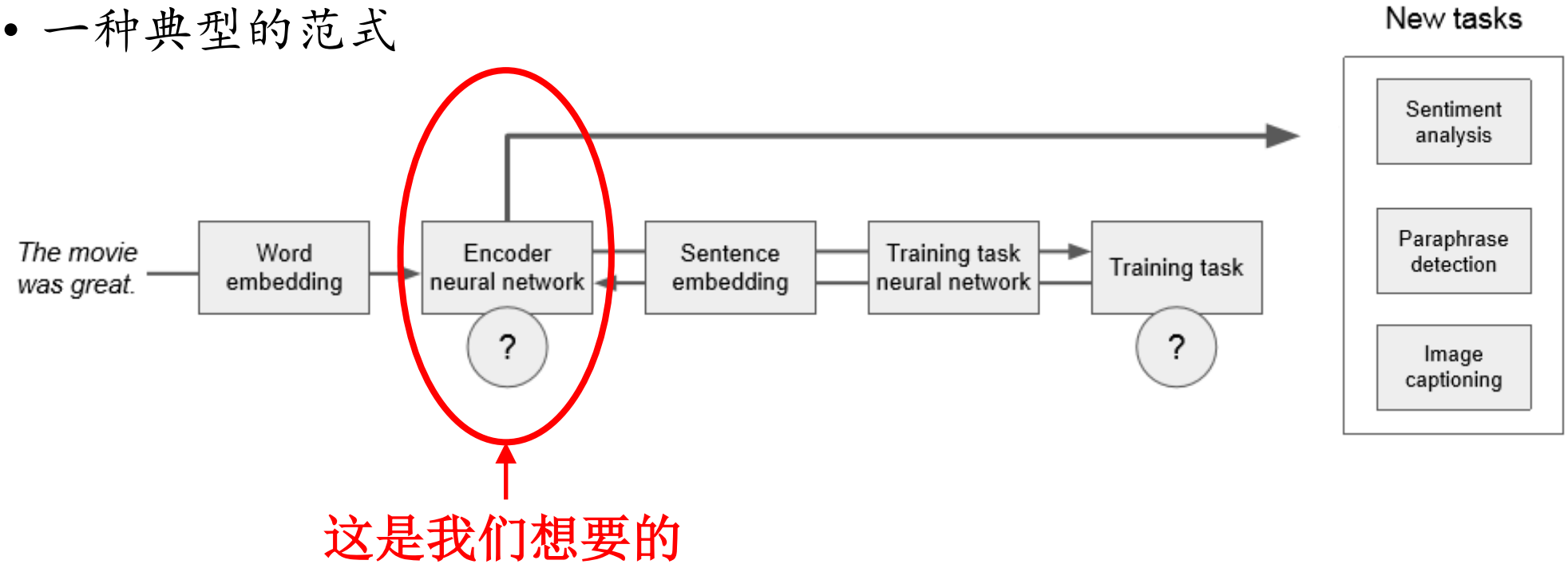
从有监督的任务中学习

从有监督的任务中学习

- 我们已经展示了无监督方法有助于学习短语和句子的表征，那么我们能否利用有监督的方法学习表征？
- 有监督方法的好处
  - 人工标注数据
  - 易于设计目标函数
  - 对下游任务更加友好
- 有监督方法的问题：
  - 需要人工标注
  - 固定的目标函数，缺乏灵活性
  - 对大多数有帮助，但是可能会损害其他任务

从有监督的任务中学习

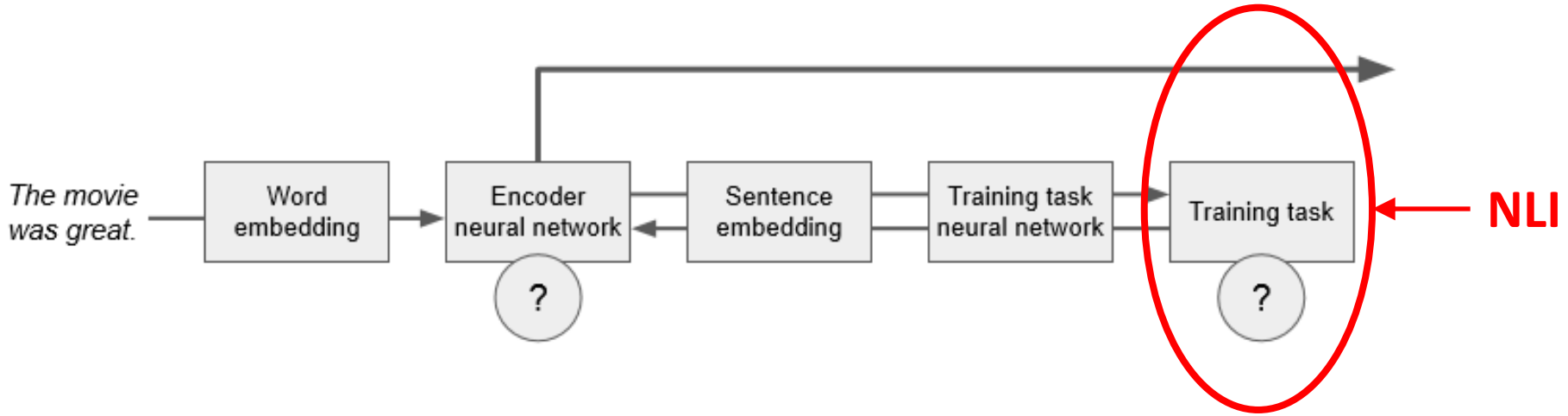
- 选择合适的任务很重要
- 选择合适的模型结构很重要
  - 一种典型的范式



Conneau et al., Supervised Learning of Universal Sentence Representations from Natural Language Inference Data, EMNLP-2017

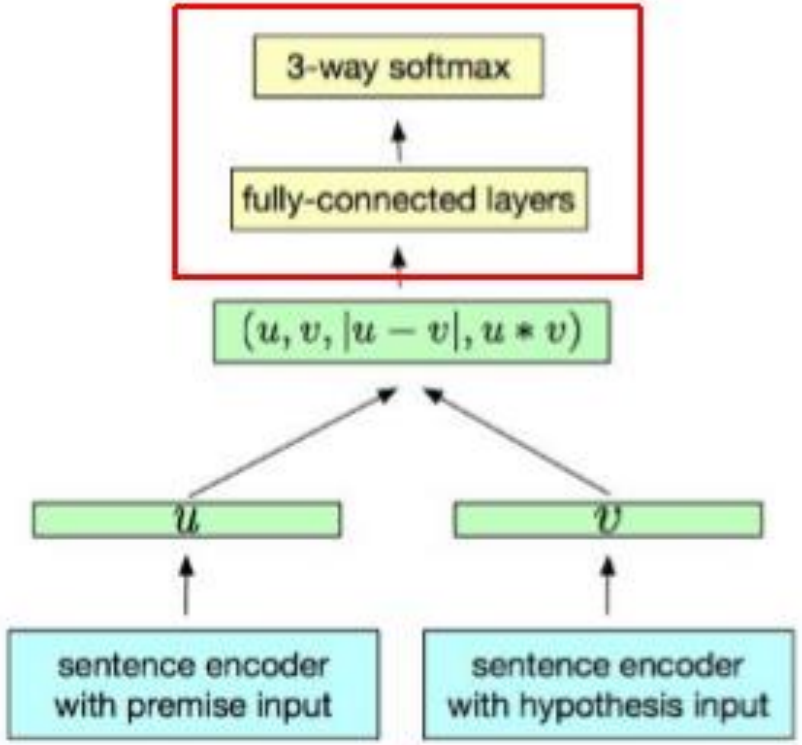
训练任务 - 自然语言推理

- 为什么选择自然语言推理（NLI）
  - NLI是一个很重要的理解任务
  - 是一个典型的分类任务
  - 易于数据处理
  - 需要句子之间的语义推理
  - 训练的过程可以学习语义信息



训练任务 – 自然语言推理

- 网络架构
  - 前提和假设通过同一个编码器然后得到表征 u 和 v
  - 结合 u 和 v：$(u, v, |u-v|, u*v)$
  - 使用全连接层分类



网络架构的影响

- 什么样的架构最适合把句子编码为一个向量?
  - CNN? RNN? LSTM? 双向? …
- 一些重要的特征
  - 考虑序列信息?
  - 更多参数?
  - 层次结构?
  - 更多的与词相关的信息?

评测

Task type	Task
Binary and multi-class classification	Sentiment analysis for movie reviews (MR and SST)
	Sentiment analysis for product reviews (CR)
	Subjectivity / objectivity (SUBJ)
	Opinion polarity (MPQA)
	Question answering (TREC)
Entailment and semantic relatedness	Entailment (SICK-E) and relatedness (SICK-R)
Semantic Textual Similarity	SemEval 2014 (STS14)
Paraphrase detection	Microsoft paraphrase classification (MRPC)
Caption image retrieval	Caption - Image retrieval (COCO)

评测 - 架构

- BiLSTM-Max 取得最好的分数
- NLI上表现好并不意味着更好的迁移能力
- 一些架构在迁移方面表现不好

Model	dim	NLI		Transfer	
		dev	test	micro	macro
LSTM	2048	81.9	80.7	79.5	78.6
GRU	4096	82.4	81.8	81.7	80.9
BiGRU-last	4096	81.3	80.9	82.9	81.7
BiLSTM-Mean	4096	79.0	78.2	83.1	81.7
Inner-attention	4096	82.3	82.5	82.1	81.0
HConvNet	4096	83.7	83.4	82.0	80.9
BiLSTM-Max	4096	**85.0**	**84.5**	**85.2**	**83.7**

评测 - 句子表征的维度

- 对于Inner-Attention, HConvNet,BiLSTM-Max,模型性能随着维度升高而变好

- 对于LSTM, GRU, BiGRU-last 和 BiLSTM-Max, 性能受到维度的影响不大