深度之眼
deepshare.net

本期论文主题:Elmo

导师: Yamada

《Deep contextualized word representations》

基于深度上下文的词表征

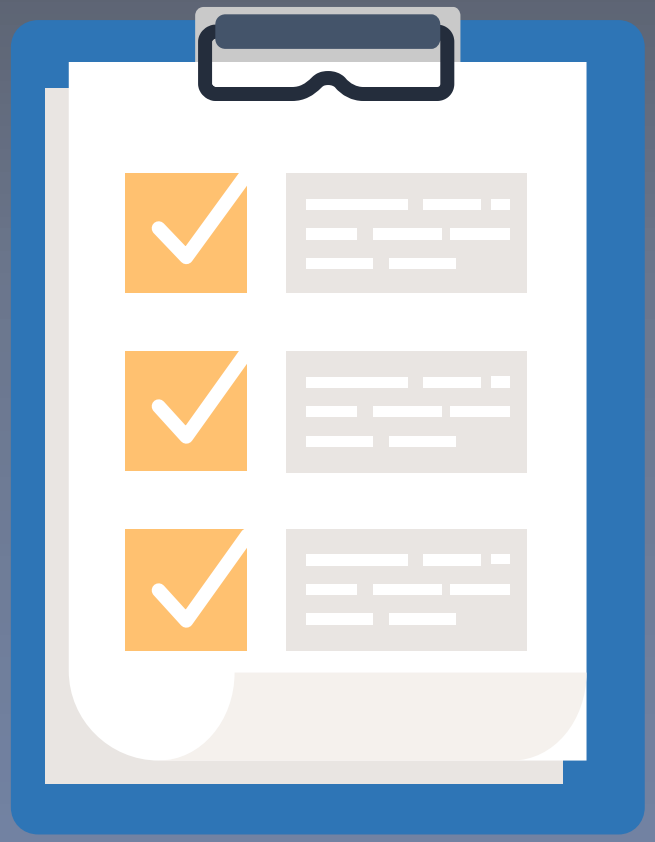
作者：Matthew E. Peters

单位：Allen Institute for Artificial Intelligence

发表会议及时间：NAACL，2018

上节回顾

Review in the lesson



深度之眼
deepshare.net

01 研究背景及成果意义

学习了nlp下游任务以及概念feature-based和fine-tuning、了解了论文的实验结果。

02 论文总览

论文总共包含6个部分，论文主要介绍elmo的结构。

03 回顾Word2vec以及Char CNN

回顾了Word2vec的流程以及学习了Char CNN的结构。

深度之眼
deepshare.net

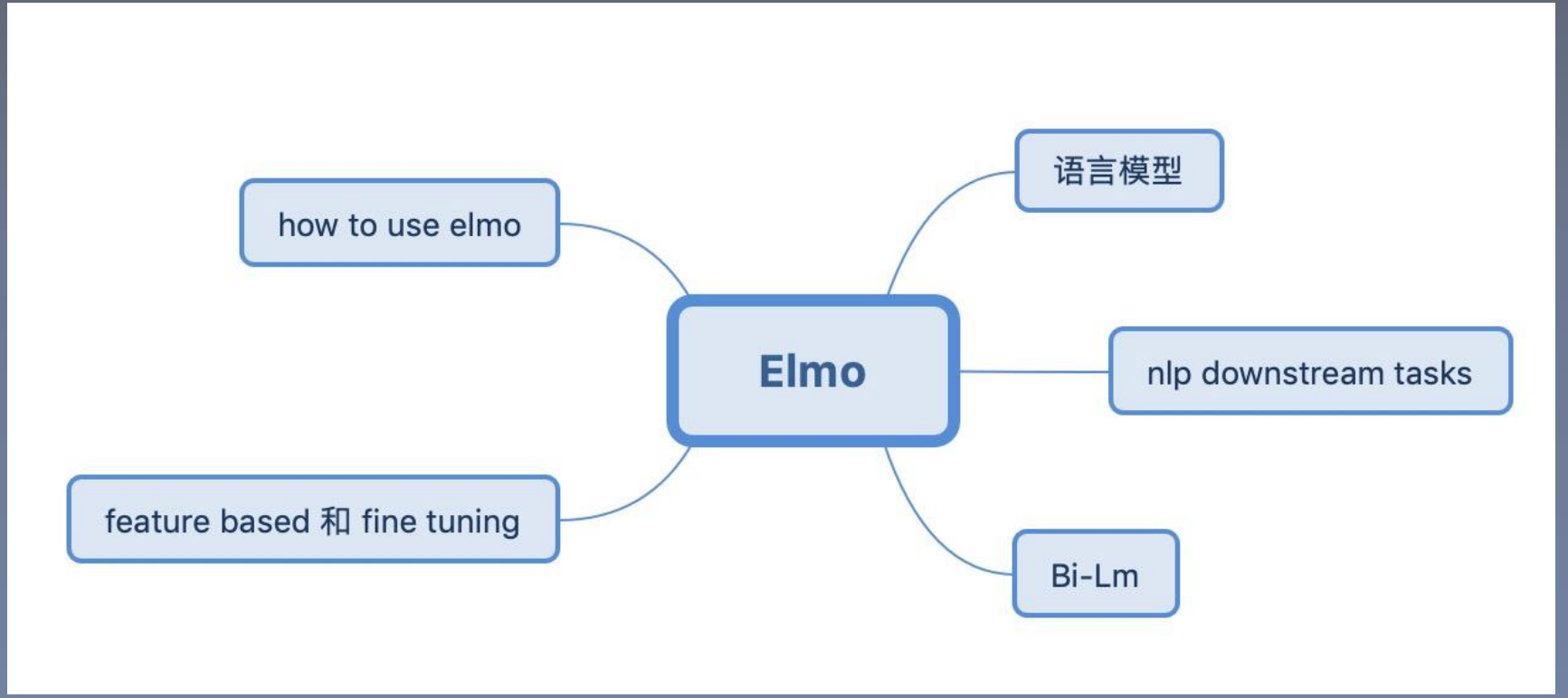
第二课：论文精读

The second lesson: the paper in detail

深度之眼
deepshare.net

目 录

知识树



Bidirectional language models

Bidirectional language models

给定输入句子：$(t_1, t_2, ...t_N)$

forward language model通过history token预测当前位置的token：

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

backward language model通过history token预测当前位置的token：

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

Bidirectional language models

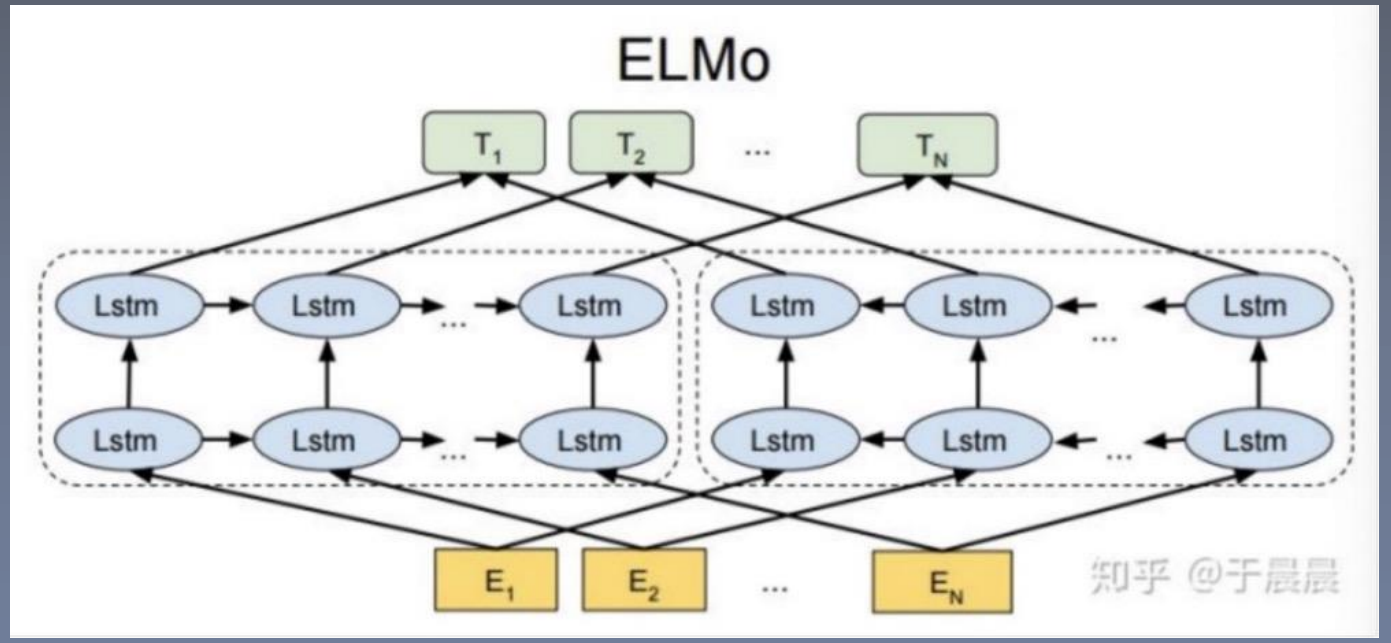
Bidirectional language models的损失函数定义为：

$$\sum_{k=1}^{N}(logp(t_k|t_1,...,t_{k-1};\Theta_x,\overrightarrow{\Theta}_{LSTM},\Theta_s) + logp(t_k|t_1,...,t_{k-1};\Theta_x,\overleftarrow{\Theta}_{LSTM},\Theta_s))$$

其中：Θ_x代表的是字符集的embedding，Θ_s代表的是softmax layer

Θ_{LSTM}代表的是lstm的参数

Bidirectional language models



E_1 E_2 ... E_N代表的是输入token 的embedding，为字向量和词向量的叠加

输入：【我 ，今天，去，天津，上学】

Bidirectional language models

输入：【我 ，今天，去，天津，上学】

词向量+字向量==========>[128,21,200]+[128,21,100]=>[128,21,300]

bilstm_layer1==========>[hidden_size=64]=>[128,21,128]

bilstm_layer2==========>[hidden_size=64]=>[128,21,128]

深度之眼
deepshare.net

Elmo

Elmo

$$R_k = \left\{ x_k^{LM}, \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} | j = 1, ..., L \right\}$$
$$= \left\{ h_{k,j}^{LM} | j = 0, ..., L \right\}$$

j: 表示的是第几层layer

x_k^{LM}:表示的第1层的embedding

$h_{k,j}^{LM}$:表示的language model embedding

LM：表示language model

Elmo

$$ELMo_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}$$

Θ^{task}:代表的是具体的下游任务

s_j^{task}:代表的是每层的weight系数 （下游任务中学习到的）

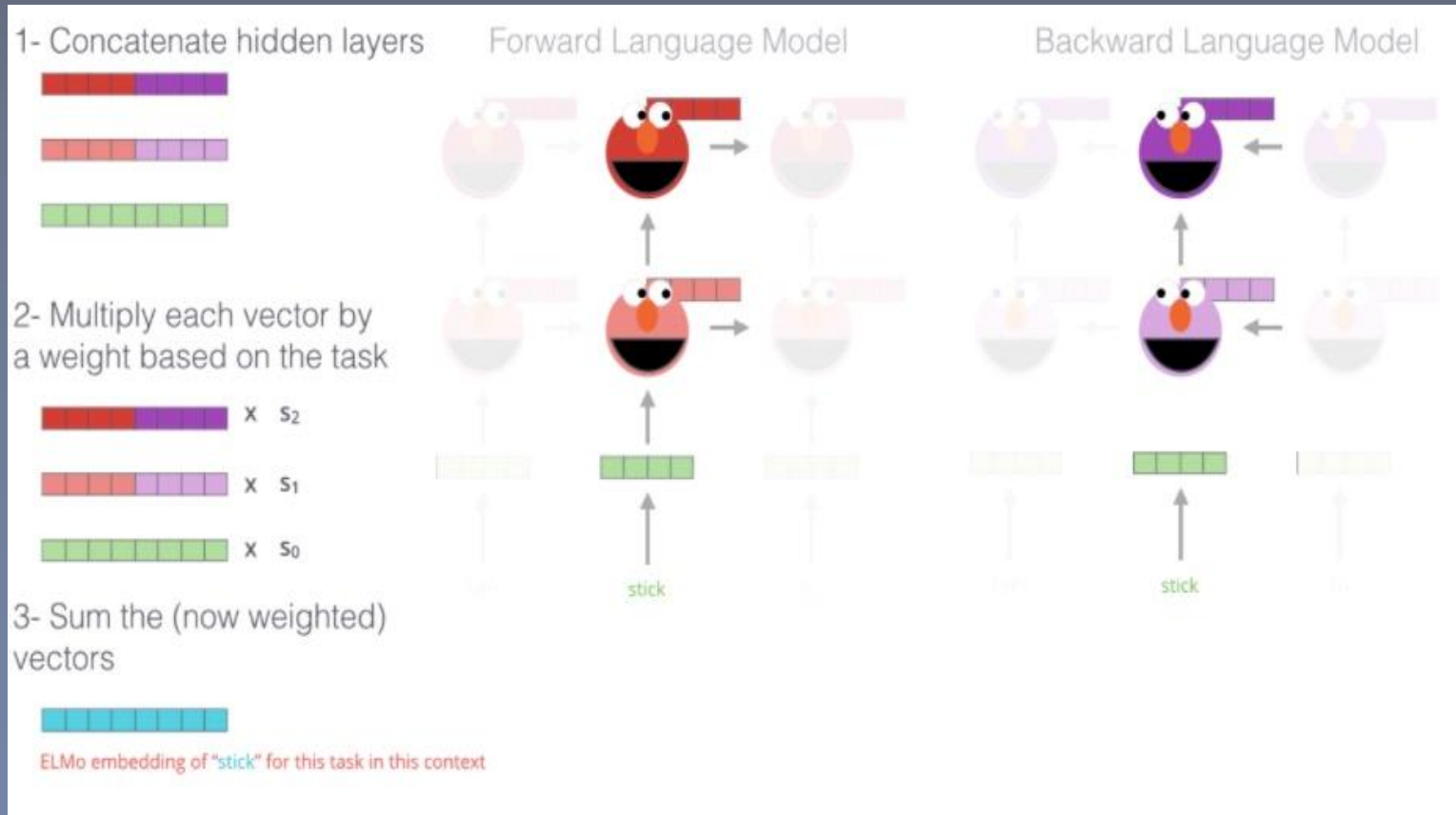
γ^{task}: 代表的是加权之前的layer normalization

Elmo

(1) 将每层的forward和backward的hideen layer进行concat [batch_size,max_length,hidden_units]

(2) 将concat后的vector乘以学习到的weight

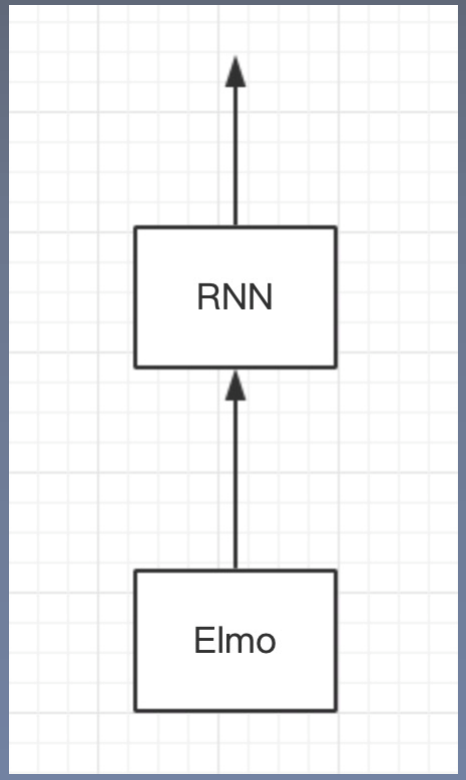
(3) 将weight后的向量进行求和得到最终的结果向量



深度之眼
deepshare.net

Elmo for NLP tasks

Elmo for NLP tasks



(1) 获取_x0008_pre-trained的word embedding和char-based embedding。模型就拥有 context-sensitive representations。

（2）freeze第一步的weights,进行下游任务。

$$[x_k; ELMo_k^{task}]$$

$$[h_k; ELMo_k^{task}]$$

Elmo for NLP tasks

elmo在nlp任务中的两种使用形式

(1) 在big data中训练elmo模型，获取想要的embedding结果
缺点:耗时耗力。不推荐（https://github.com/allenai/bilm-tf）
（https://github.com/HIT-SCIR/ELMoForManyLangs）

(2) 直接加载训练好的预训练模型和文件，在小数据集上训练获得最终的结果。
推荐
（https://github.com/allenai/bilm-tf）tensorflow实现
（https://github.com/strongio/keras-elmo/blob/master/Elmo%20Keras.ipynb）
keras实现
（https://github.com/allenai/allennlp/blob/9a6962f00d2b0d30b81900b4e9764ddc3433f400/tutorials/how_to/elmo.md）pytorch实现

深度之眼
deepshare.net

实验设置和结果分析

Experiment results

实验结果及分析

Results and Discussion

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

elmo在6项下游任务中表现不错。

论文总结

论文总结

Summary of the paper

深度之眼
deepshare.net

A

关键点

- Bidirectional language model原理
- Elmo在nlp任务中使用

B

小细节

- elmo中的weight是在下游任务中学习到的
- elmo的缺点

论文总结

Summary of the paper

C

启发点

- 模型创新的时候可以使用elmo生成词向量
- 可以使用char-cnn获取字向量然后和词向量拼接

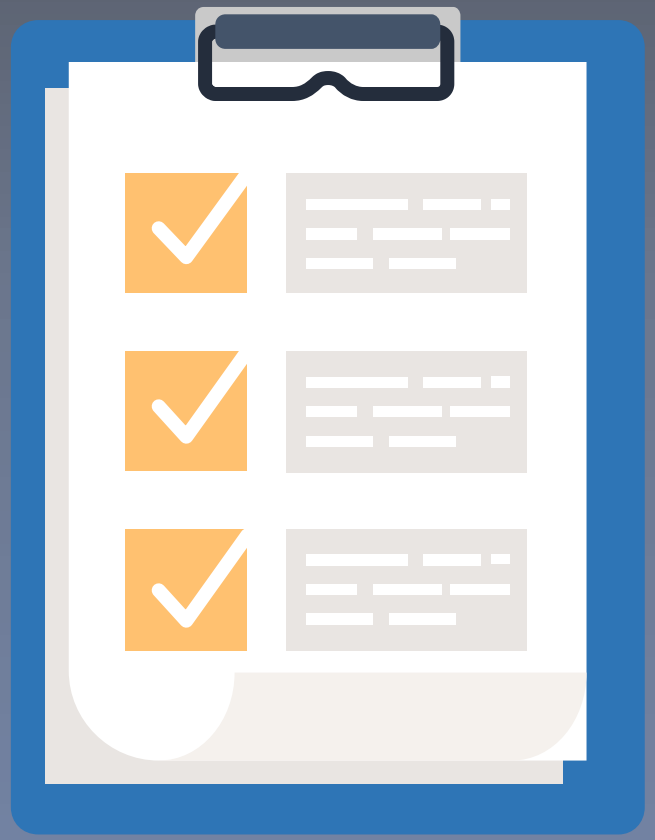
深度之眼
deepshare.net

本课回顾及下节预告

Review in the lesson and Preview of next lesson

本课回顾

Review in the lesson



深度之眼
deepshare.net

01 Bidirectional Language Model

讲解 Bidirectional Language Model构成，是由双向language model拼接而成。

02 elmo使用

elmo在nlp下游任务中有两种形式，一种为预训练、然后在下游任务中加载，一种为直接在big data中训练。

03 实验设置及结果分析

网络超参数设置，学习率，batchsize等
实验结果分析对比

04 论文总结

总结论文中创新点、关键点及启发点

下节预告

Preview of next lesson

01 搭建Transformer网络代码介绍

02 介绍Self Attention实现

03 基于翻译数据集训练Transformer模型

下节课前准备

Preview of next lesson

- 再次阅读Elmo论文
- 熟悉Elmo模型结构及数据预处理方式
- 配置PyTorch开发环境
- 下载Elmo代码
- https://github.com/yongyuwen/PyTorch-Elmo-BiLSTMCRF

结 语

循循而进，欲速则不达也。

深度之眼
deepshare.net

联系我们：
电话：18001992849
邮箱：service@deepshare.net
ＱＱ：2677693114

公众号

客服微信