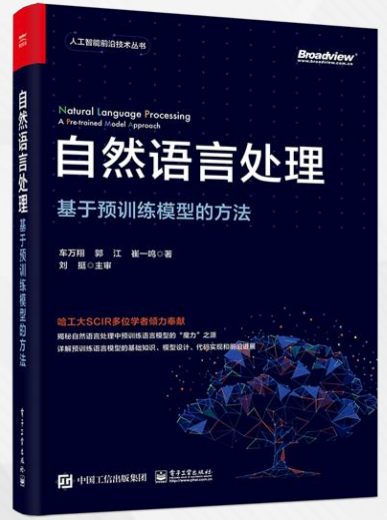


动态词向量预训练模型

车万翔、郭江、崔一鸣

社会计算与信息检索研究中心
哈尔滨工业大学

目 录

CONTENTS

1 词向量——从静态到动态

2 基于语言模型的动态词向量预训练

目 录

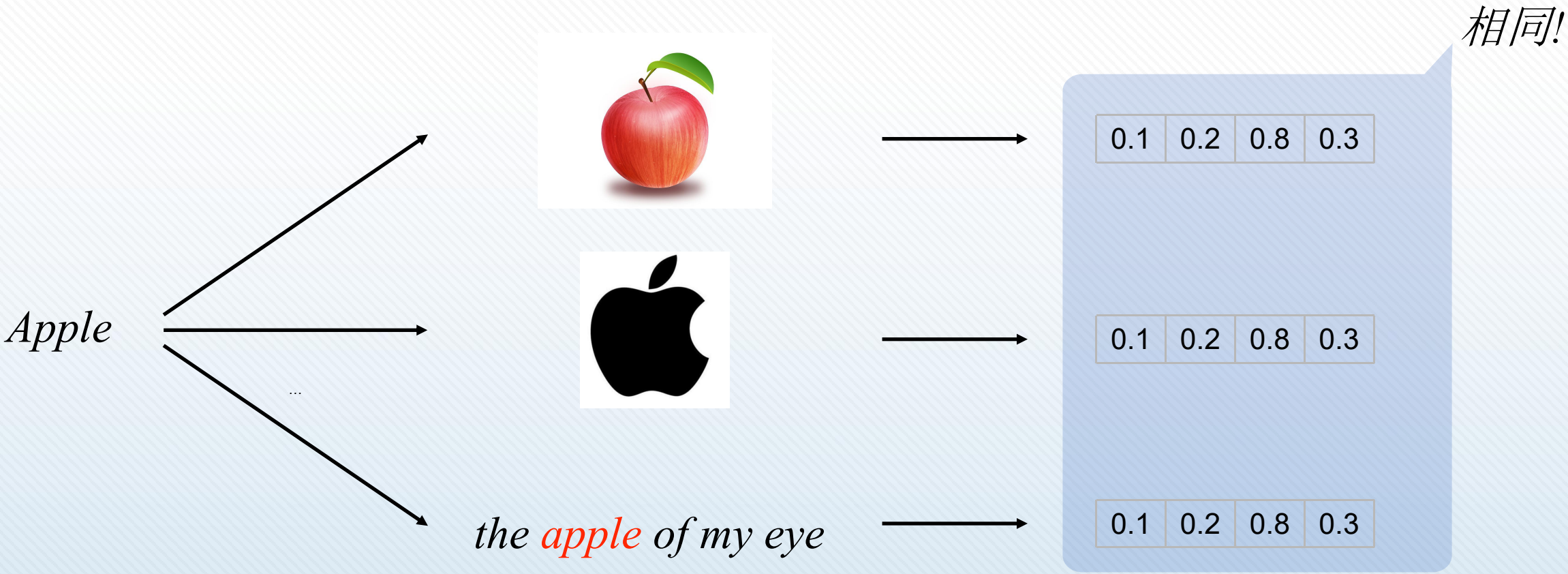
CONTENTS

1 词向量——从静态到动态

2 基于语言模型的动态词向量预训练

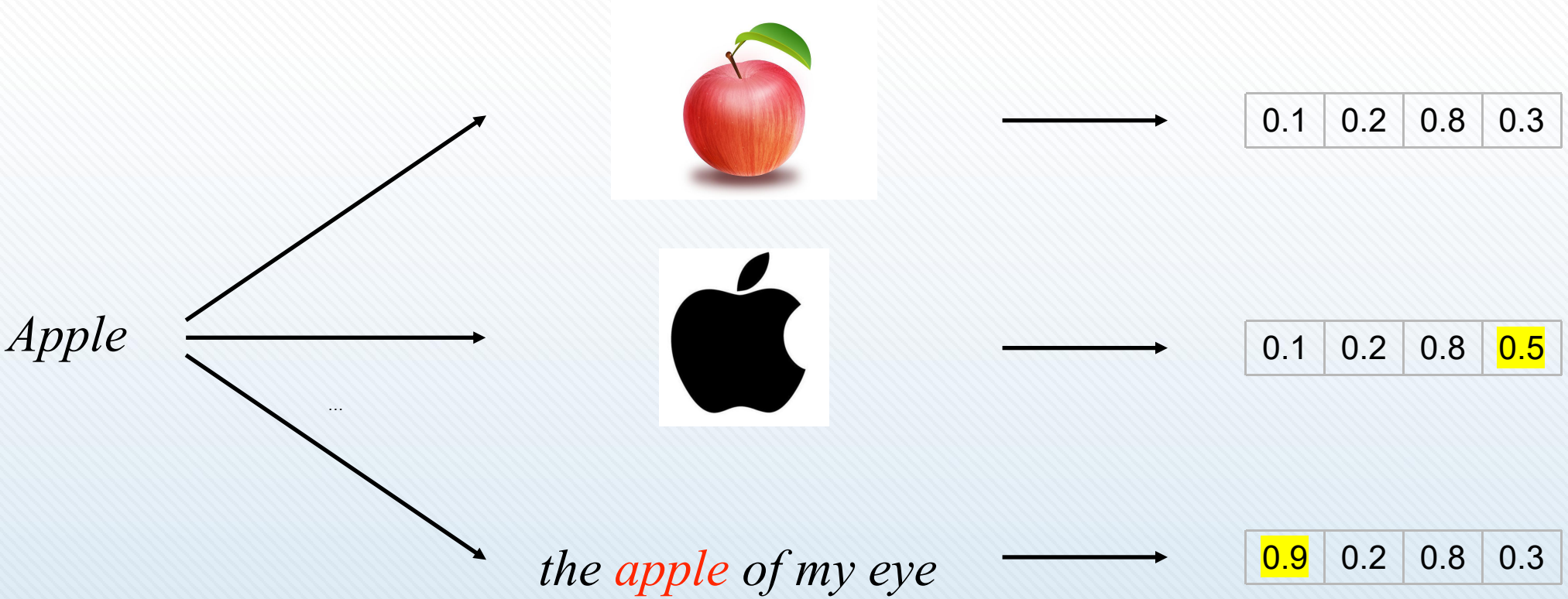
词向量——从静态到动态

- 静态词向量的问题
  - 很多词包含多种语义信息，静态词向量无法解决“一词多义”的表示问题



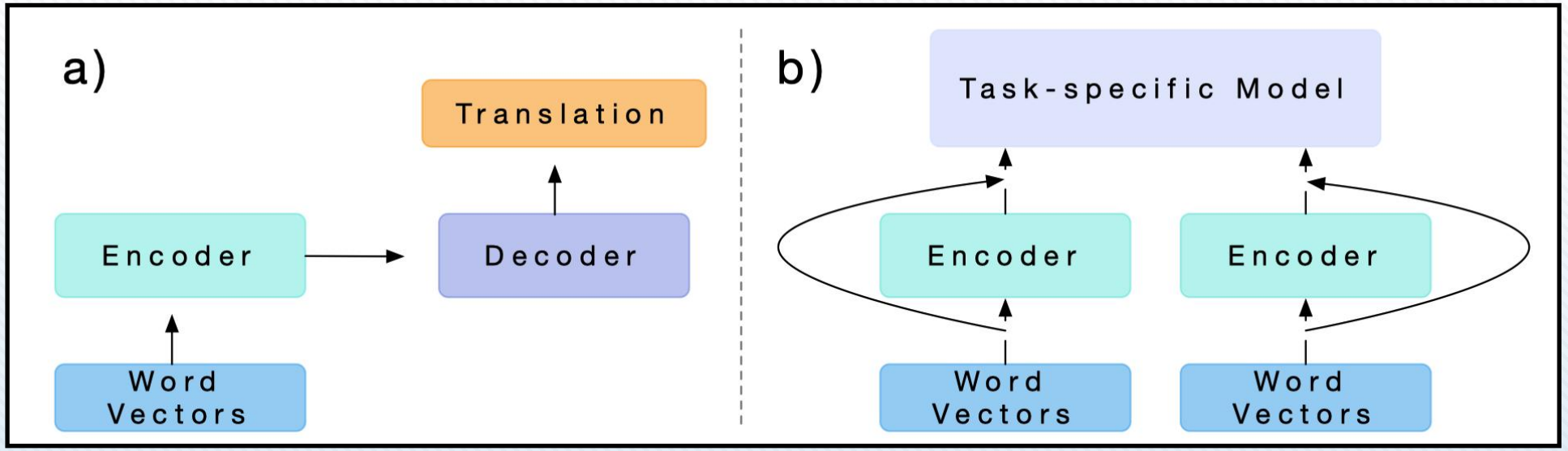
词向量——从静态到动态

- 静态词向量的问题
  - 词向量应根据其所处的上下文的不同而发生改变



词向量——从静态到动态

- CoVe (Contextualized Word Vectors)
  - 首次提出使用上下文相关的文本表示，即每个token的向量表示不唯一
  - 主要思想：将神经机器翻译（NMT）的表示迁移到通用NLP任务上



词向量——从静态到动态

- CoVe (Contextualized Word Vectors)
  - 训练阶段：训练一个神经机器翻译模型（NMT）
    - 给定一个源语言句子w^x和目标语言句子w^z

$$h = \text{MT-LSTM}(\text{GloVe}(w^x))$$

    - 基于注意力机制的解码器

$$\alpha_t = \text{softmax}\left(H(W_1 h_t^{\text{dec}} + b_1)\right)$$

$$\tilde{h}_t = \left[\tanh\left(W_2 H^\top \alpha_t + b_2; h_t^{\text{dec}}\right)\right]$$

$$h_t^{\text{dec}} = \text{LSTM}\left([z_{t-1}; \tilde{h}_{t-1}], h_{t-1}^{\text{dec}}\right)$$

    - 输出层

$$p(\hat{w}_t^z | X, w_1^z, \ldots, w_{t-1}^z) = \text{softmax}\left(W_{\text{out}} \tilde{h}_t + b_{\text{out}}\right)$$

词向量——从静态到动态

- CoVe (Contextualized Word Vectors)
  - 如何应用在下游任务
    - 对于给定的一个句子w，通过如下方式获得CoVe表示

$$\text{CoVe}(w) = \text{MT-LSTM}(\text{GloVe}(w))$$

    - 可与传统的GloVe等词向量拼接，增强语义表示能力

$$\tilde{w} = [\text{GloVe}(w); \text{CoVe}(w)]$$

目 录

CONTENTS

1 词向量——从静态到动态

2 基于语言模型的动态词向量预训练

CoVe存在的问题

训练依赖于双语平行语料

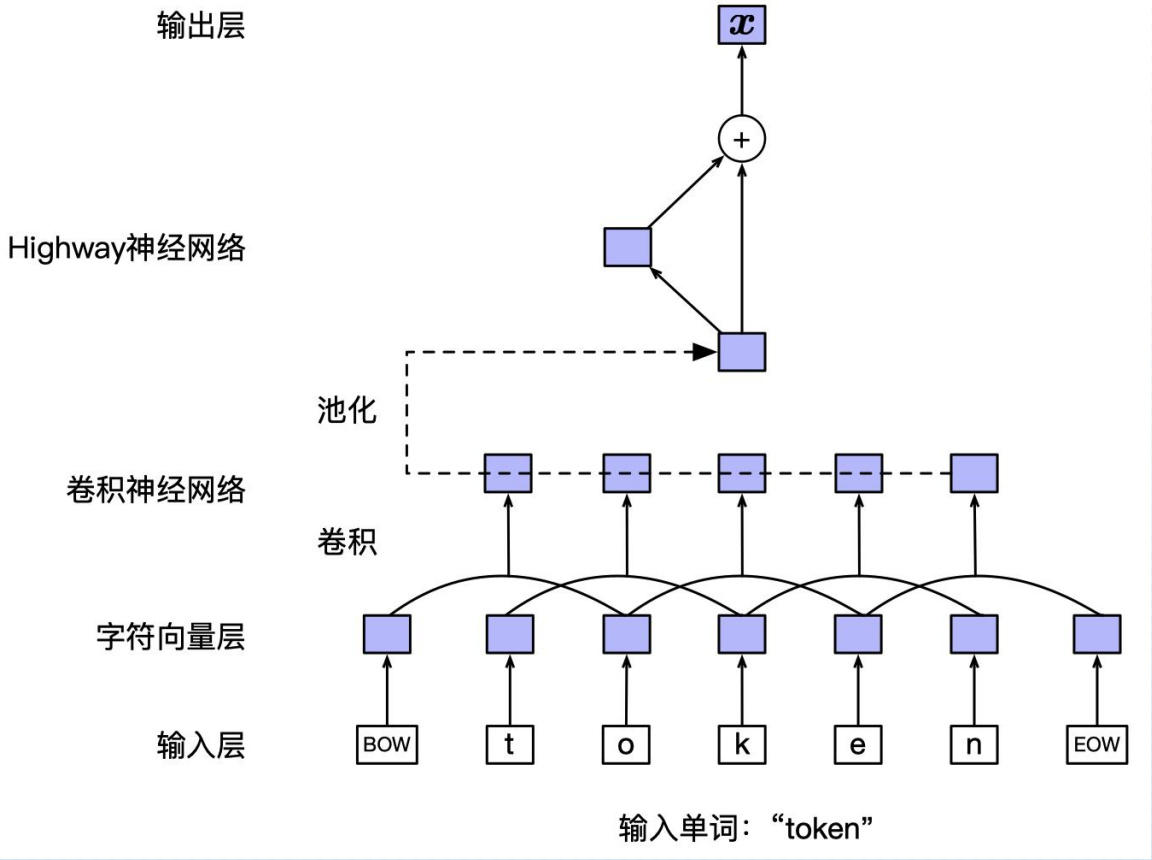
- 训练神经机器翻译模型需要双语平行语料，获取难度较高
- 相比单语语料，覆盖的领域也相对优先，通用性一般

单独使用效果一般，性价比不高

- 实验结果表明单独使用CoVe的效果一般
- 需要搭配传统静态词向量才能获得较为显著的性能提升

基于语言模型的动态词向量预训练

- 双向语言模型BiLM
  - 双向语言模型从前向（从左到右）和后向（从右到左）两个方向同时建立语言模型



基于语言模型的动态词向量预训练

- 双向语言模型BiLM
  - 输入表示层

$$\boldsymbol{v}_{c_i} = \boldsymbol{E}^{\text{char}} \boldsymbol{e}_{c_i}$$

$$\boldsymbol{x}_t = \boldsymbol{g} \odot \boldsymbol{f}_t + (\boldsymbol{1} - \boldsymbol{g}) \odot \text{ReLU}(\boldsymbol{W}\boldsymbol{f}_t + \boldsymbol{b})$$

$$\boldsymbol{g} = \sigma(\boldsymbol{W}^{\text{g}}\boldsymbol{f}_t + \boldsymbol{b}^{\text{g}})$$

  - 前向语言模型

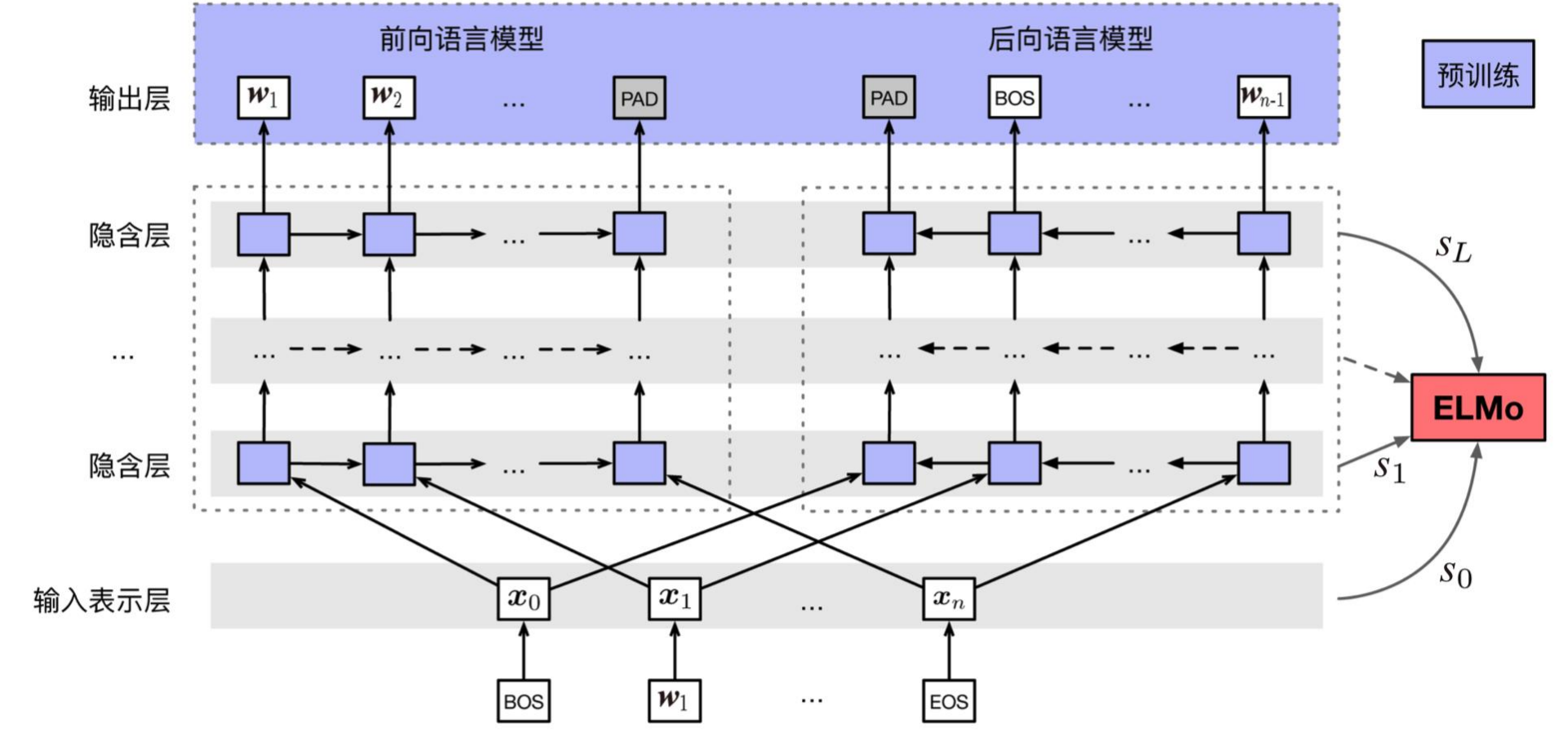
$$P(w_1 w_2 \cdots w_n) = \prod_{t=1}^{n} P(w_t | \boldsymbol{x}_{1:t-1}; \overrightarrow{\boldsymbol{\theta}}^{\text{lstm}}, \boldsymbol{\theta}^{\text{out}})$$

  - 后向语言模型

$$P(w_1 w_2 \cdots w_n) = \prod_{t=1}^{n} P(w_t | \boldsymbol{x}_{t+1:n}; \overleftarrow{\boldsymbol{\theta}}^{\text{lstm}}, \boldsymbol{\theta}^{\text{out}})$$

基于语言模型的动态词向量预训练

ELMo词向量



基于语言模型的动态词向量预训练

- ELMo词向量
  - ELMo采取对不同层次的向量表示进行加权平均的机制，为不同的下游任务提供更多的组合自由度

$$\mathbb{R}_t = \{\boldsymbol{x}_t, \boldsymbol{h}_{t,j} | j = 1, \cdots, L\}$$

$$\mathrm{ELMo}_t = f(\mathbb{R}_t, \Psi) = \gamma^{\mathrm{task}} \sum_{j=0}^{L} s_j^{\mathrm{task}} \boldsymbol{h}_{t,j}$$

  - ELMo特点
    - **动态（上下文相关）**：词的ELMo向量表示由其当前上下文决定；
    - **鲁棒（Robust）**：ELMo向量表示使用字符级输入，对于未登录词具有强鲁棒性；
    - **层次**：ELMo词向量由深度预训练模型中各个层次的向量表示进行组合，为下游任务提供了较大的使用自由度。

基于语言模型的动态词向量预训练

- 模型实现
  - 数据准备
    - 使用清洗后并经过分词等预处理的语料
    - 需要同时构建词级别与字符级别的训练语料，并建立相应的词表
  - 双向语言模型
    - ELMo模型的核心是双向语言模型
    - 编码器部分主要包括基于字符的输入表示层以及前向、后向 LSTM 层
  - 训练
    - 在数据、模型组件构建完成后，使用实际数据对模型进行训练
    - 训练过程将输出每一次迭代后的前向语言模型的困惑度值
    - 训练完成后，便可以利用双向语言模型的编码器编码输入文本并获取动态词向量

基于语言模型的动态词向量预训练

模型实现

使用AllenNLP调用ELMo

```
>>> from allennlp.modules.elmo import Elmo, batch_to_ids
>>> options_file = "https://allennlp.s3.amazonaws.com/models/elmo/2
    x4096_512_2048cnn_2xhighway/elmo_2x4096_512_2048cnn_2xhighway_options.
    json"
>>> weights_file = "https://allennlp.s3.amazonaws.com/models/elmo/2
    x4096_512_2048cnn_2xhighway/elmo_2x4096_512_2048cnn_2xhighway_weights.
    hdf5"
>>> elmo = Elmo(options_file, weight_file, num_output_representations=1,
    dropout=0)
```

```
>>> sentences = [['I', 'love', 'Elmo'], ['Hello', 'Elmo']]
>>> character_ids = batch_to_ids(sentences)
    # 输出大小为2*3*50(字符向量维度)的张量
>>> embeddings = elmo(character_ids)
>>> print(embeddings)
```

基于语言模型的动态词向量预训练

- 应用与评价
  - 作为下游任务特征
    - 即插即用，可以与静态词向量进行拼接

$$[\boldsymbol{x}_k; \mathrm{ELMo}_k]$$

    - 也可以与隐层输出进行拼接

$$[\boldsymbol{h}_k; \mathrm{ELMo}_k]$$

    - 右侧给出了利用ELMo实现文本分类的示例代码

```
class ELMoMLP(nn.Module):
    def __init__(self, elmo, hidden_dim, num_class):
        super(ELMoMLP, self).__init__()
        # ELMo预训练编码器，可使用AllenNLP预训练ELMo模型
        self.elmo = elmo
        # 隐含层
        self.fc1 = nn.Linear(self.elmo.get_output_dim(), hidden_dim)
        # 输出层
        self.fc2 = nn.Linear(hidden_dim, num_class)
        self.activate = F.relu

    def forward(self, inputs, lengths):
        elmo_output = self.elmo(inputs)
        embeds = elmo_output['elmo_representations'][0]
        mask = elmo_output['mask']

        # 将每个序列中词的ELMo向量均值作为该序列的向量表示，作为MLP的输入
        embeds = torch.sum(embeds * mask.unsqueeze(2), dim=1) / lengths.
    unsqueeze(1)
        hidden = self.activate(self.fc1(embeds))
        output = self.fc2(hidden)
        log_probs = F.log_softmax(output, dim=1)
        return log_probs
```

基于语言模型的动态词向量预训练

- 应用与评价
  - 上下文相关的词义相似性检索
    - ELMo相比GloVe（静态词向量）在词义消歧和近邻分析任务上都有比较好的表现

模型	词	近邻
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
ELMo	Chico Ruiz made a spectacular play on Alusik's grounder ···	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play
	Olivia De Havilland signed to do a Broadway play for Garson ···	··· they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement

谢 谢!

理解语言，认知社会

以中文技术，助民族复兴

长按二维码，关注哈工大SCIR

微信号：HIT_SCIR