现代信息检索
Modern Information Retrieval

第15讲 Neural IR
基于深度神经网络的IR模型

提纲

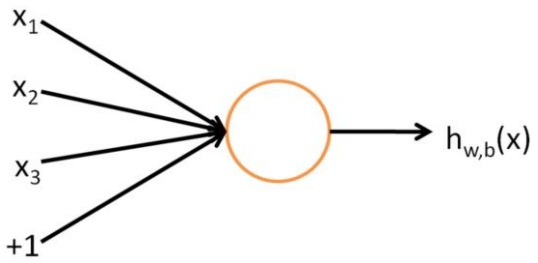
- 深度神经网络(DNN)基础
- Neural IR Model

提纲

- 深度神经网络(DNN)基础
- Neural IR Model

神经元

- 最简单的神经网络—神经元



对应的计算如下：

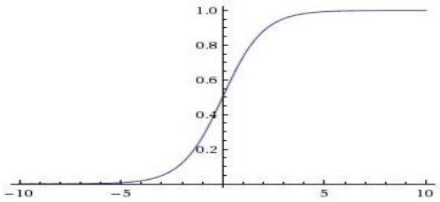
$$h_{W,b} = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$$

其中W和b为需要学习的网络参数，f为激活函数。

激活函数

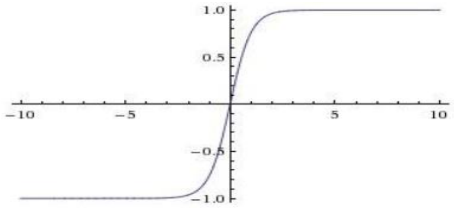
- 激活函数：主要作用是引入非线性，增强网络的表示能力。
- Sigmoid函数

$$f(z) = \frac{1}{1 + e^{-z}}, \qquad (0,1)$$



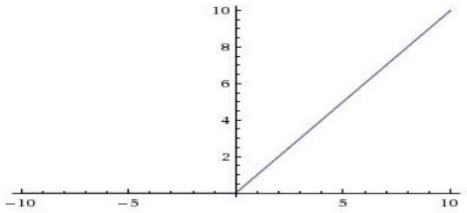
- Tanh函数

$$f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}, \qquad (-1,1)$$



- ReLU函数

$$f(z) = \begin{cases} z, z > 0 \\ 0, z \le 0 \end{cases}, \qquad [0, +\infty)$$

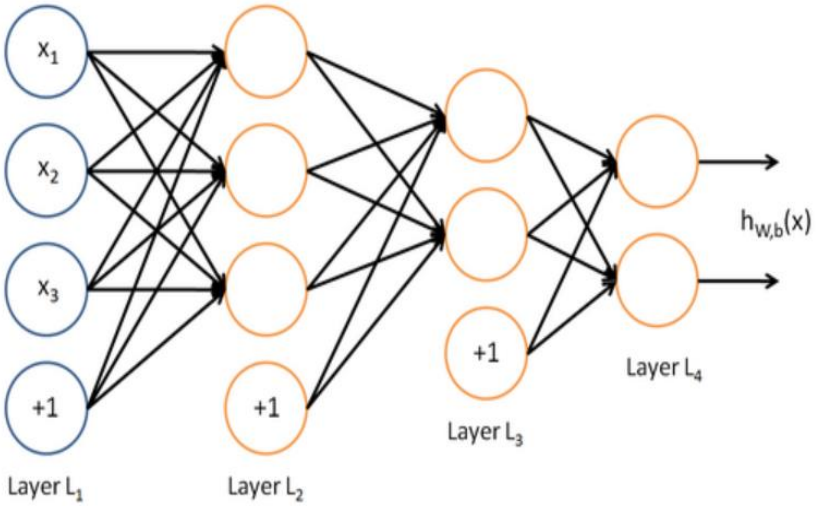


激活函数

- 上述激活函数特点
  - Sigmoid：两端饱和区梯度极小；输出不以0为中心；指数函数计算代价大。
  - Tanh：两端饱和区梯度极小；输出以0为中心；指数函数计算代价大。
  - ReLU：在激活值大于0时不存在梯度极小的情况；输出不以0为中心；计算代价小；收敛速度快。
- 除了上述三种激活函数，还有其它一些激活函数，如Maxout，Leaky ReLU，ELU等。
- 激活函数对参数的学习过程影响较大，需要根据情况适当选择。

神经元组合成为神经网络

- 最简单的多层神经网络—多层感知机 (Multi-Layer Perceptron, 简称MLP)



由多个神经元组成，一些神经元的输出作为另一些神经元的输入。

Softmax归一化

- Softmax归一化是在使用神经网络进行分类时常用的方法，对于分类问题，通常需要给出可能属于每一个类别的概率，即需要输出介于0和1之间，且加和为1，对于未归一化输出$(o_1, o_2, \dots, o_m)$，具体计算如下：

$$y_i = \frac{e^{o_i}}{\sum_{j=1}^{j=m} e^{o_j}}$$

- $(y_1, y_2, \dots, y_m)$即为归一化后的输出，满足值介于0和1之间且求和为1的要求。

参数的学习

- **损失函数**

为了衡量模型预测的效果，通常会定义一个关于模型预测y’与实际标签y的函数 L(y’, y)，注意到 y’ 是模型参数 θ 的一个表达式，通过最小化 L(y’, y) 可以得到模型参数 θ 的一组值使得模型的预测 y’ 能够足够接近实际标签 y 。

- **例：交叉熵损失**

交叉熵损失是应用最为广泛的一种损失函数，即用训练数据与模型间的交叉熵来衡量预测分布于实际分布的差距，它的形式如下：

$$L(\theta) = -E_{x,y\sim\tilde{p}_{data}} \log_2 p_{model}(y|x)$$

交叉熵损失与负对数似然是等价的；p_{model}取高斯分布就得到均方误差。

参数的学习

- **目标：** 学习一组网络参数，使得预测 y' 与实际标签y的误差 (损失)最小。
- **BP算法：** 即反向传播算法，是学习神经网络参数的一个重要方法，给定一个样本 (x, y)，包含如下两个过程：
  - 前向计算 (forward)：根据输入 x，计算网络的输出 y'；
  - 反向计算 (backward)：计算网络预测 y' 与标签 y 之间的误差 (损失) 关于网络各参数的梯度；主要应用求导的链式法则。
- **梯度下降算法：** BP算法只是得到了误差 (损失) 关于网络参数的梯度，而梯度下降算法定义了网络参数的更新方式，如SGD：

$$\theta = \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$$

  - 其它常见的参数更新方式：Momentum，Adam，Adagrad，RMSprop等
  - 在实际应用中，一般是同时使用一组样本 (一个batch) 来对网络参数进行更新。
  - 另外还有一些二阶的方法：牛顿法，共轭梯度，BFGS

正则化

- 为什么需要正则化？

一般的学习算法都是通过最小化训练集上损失函数来得到的，若训练数据的数据量较小或者分布不均，对于容量较大的模型而言，则学习到的模型会过度拟合训练数据分布而与真实分布有一定的差距，所以需要正则化来防止学习到的模型过度拟合训练数据分布，从而增强模型的泛化能力。

若想要进一步了解，请参考偏差-方差分解理论。

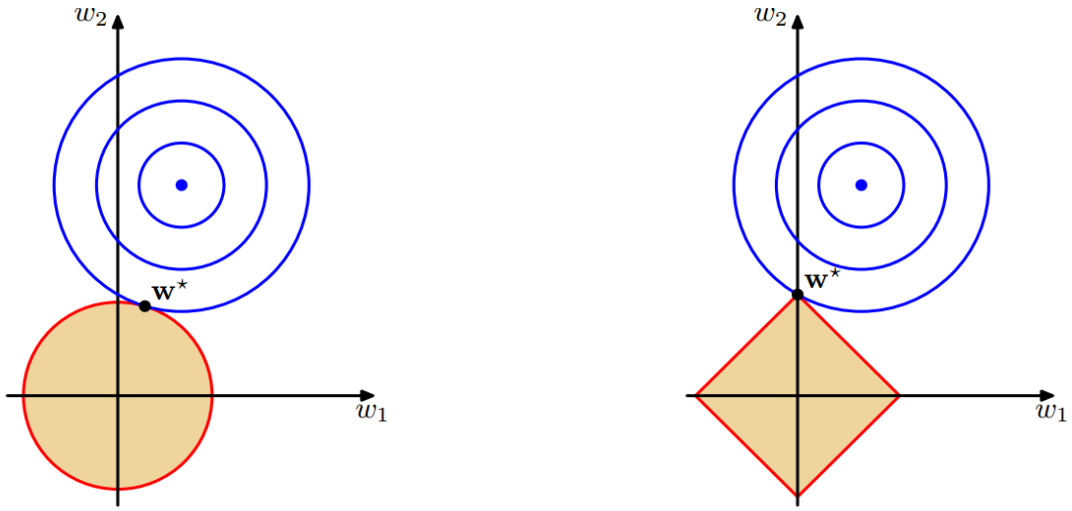
- L1与L2正则

机器学习中常用的正则方法，通过在损失函数中增加模型参数的1-范数或2范数项来约束模型参数的范围：

  - 一般认为L1正则会使得模型参数的某些维度变为0，因此具有特征选择的作用；

正则化

- L1与L2正则图解：L1正则（右），L2正则（左）



图中同一个蓝色环上的损失相同，中心点损失最小；红色环上模相等，原点处模最小，为0；黑色点为解，在黑色点处损失的减小与模的增加达到临界点，即损失的继续减小不能弥补模增加的部分，导致它们的和反而增加了。

正则化

- DNN中常用的正则化方法
  - 数据集增强：通过对已有的数据样本做特定的变换来构造新的样本。
  - 噪声鲁棒性：通过往输入、权重或者标签中注入噪声来达到正则化的效果。
  - 提前终止：通过引入验证集，训练到验证集上误差达到最小时，则停止训练。
  - 参数共享：通过使网络的不同部分共享参数达到正则化效果，参数共享减小了模型的假设空间。
  - Bagging集成方法：训练若干模型，然后由这些模型对输出进行表决，以此来减小泛化误差。
  - Dropout：通过对神经元以一定概率进行丢弃达到正则化效果，通常认为是Bagging的一种近似。

卷积神经网络 (CNN)

- 卷积神经网络 (CNN)：用来专门处理具有类似网络结构数据的神经网络；卷积网络至少有一层的结构使用了卷积运算。
- 卷积运算：

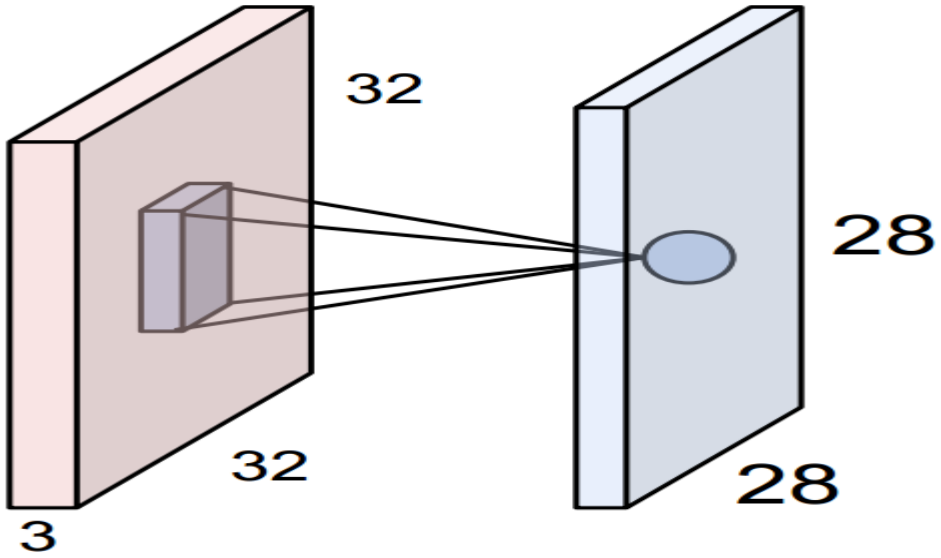
$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n) K(m,n)$$

其中 I 为二维输入， 对应的卷积核 K 也为二维；对于高维输入，计算类似。

- 卷积网络的特点：
  - 稀疏连接：卷积核的参数规模远小于输入的规模，可以用于探索一些小而有意义的特征；与全连接相比，稀疏连接减小了网络的参数规模。
  - 参数共享：不同输入共享相同的参数，进一步减小了网络的参数规模。

卷积神经网络 (CNN)

- 卷积图解



输入：32x32x3；卷积核：5x5x3，1个；输出：28x28x1，步长：1

卷积神经网络 (CNN)

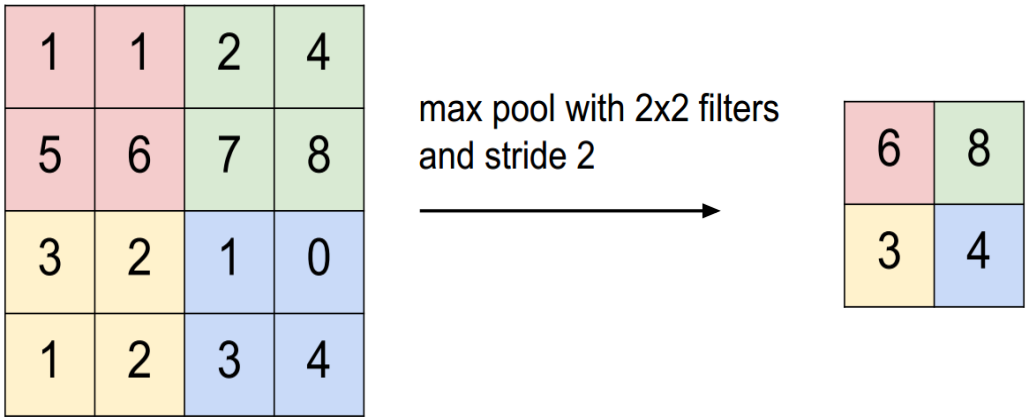
- 池化(Pooling)：池化的总体思想是使用某一位置的相邻输出的总体统计特征来代替网络在该位置的输出。

- 常见池化方式: max-pooling, min-pooling, average-pooling, sum-pooling。

- 以下用max-pooling举例

卷积神经网络 (CNN)

- Max-pooling图解



- 卷积层的三个过程：
  - 卷积：卷积核对输入的线性变换
  - 激活：激活函数对卷积核输出的非线性变换
  - 池化：对激活输出进行进一步调整
- 两个参数：filter的大小，stride：filter移动的步长

池化的特点

- 近似不变性：当输入做少量平移时，输出不会发生变化；
- 近似不变性使得网络更多地关注某些特征是否出现而不关心其具体的位置；
- 由于近似不变性，网络能够容忍一些微小的噪声或者扰动。

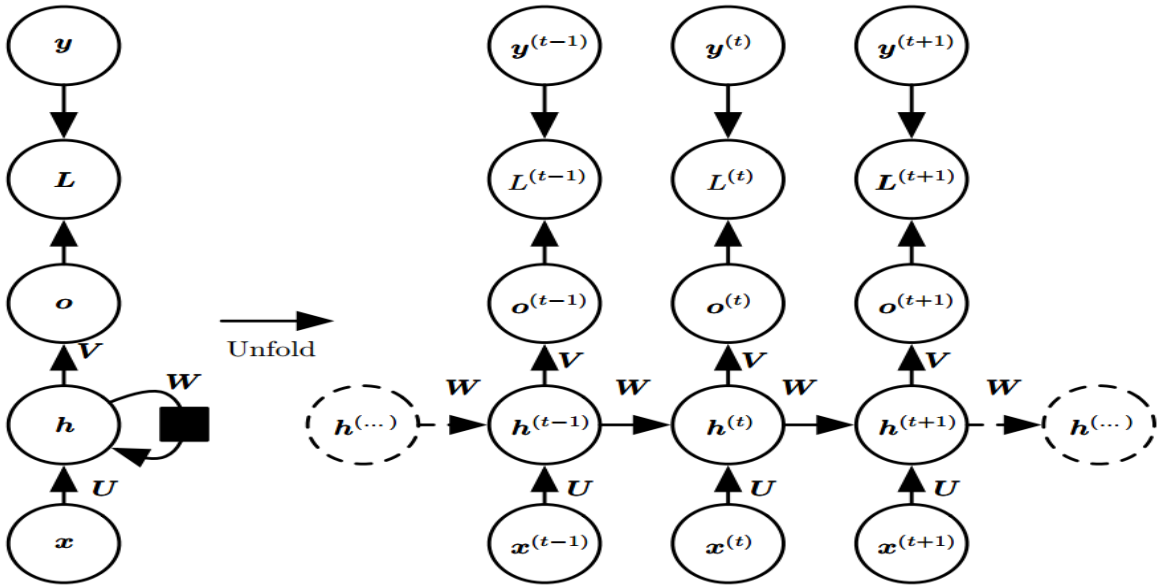
- 卷积和池化带来的好处主要有：减少参数，减少噪声

循环神经网络 (RNN)

- 循环神经网络 (RNN)：一种用来专门处理序列数据的神经网络。
- 一个序列当前的输出与前面的输出有关
  - 网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出
- 在实践中，为了降低复杂性往往假设当前的状态只与前面的几个状态相关

循环神经网络 (RNN)

- 示例：一个简单的多对多（many-to-many）结构，如语言模型。



对于语言模型而言，模型的任务是根据前 t-1 个单词（$x^0 \sim x^{t-1}$）预测第 t 个单词（y^{t-1}，亦即x^t），将预测看作一个分类任务，则o^{t-1}为一个|V| 维向量，表示取各个词的未归一化概率，则模型的参数可以采用最大似然的方法进行估计。

循环神经网络 (RNN)

- 假定输出采用$softmax$进行归一化，则网络对应的计算如下：

$$h^{(t)} = \tanh\left(b + Wh^{(t-1)} + Ux^{(t)}\right)$$

$$o^{(t)} = Vh^{(t)}$$

$$\hat{y}^{(t)} = softmax\left(o^{(t)}\right)$$

- 负对数似然函数（损失函数）如下：

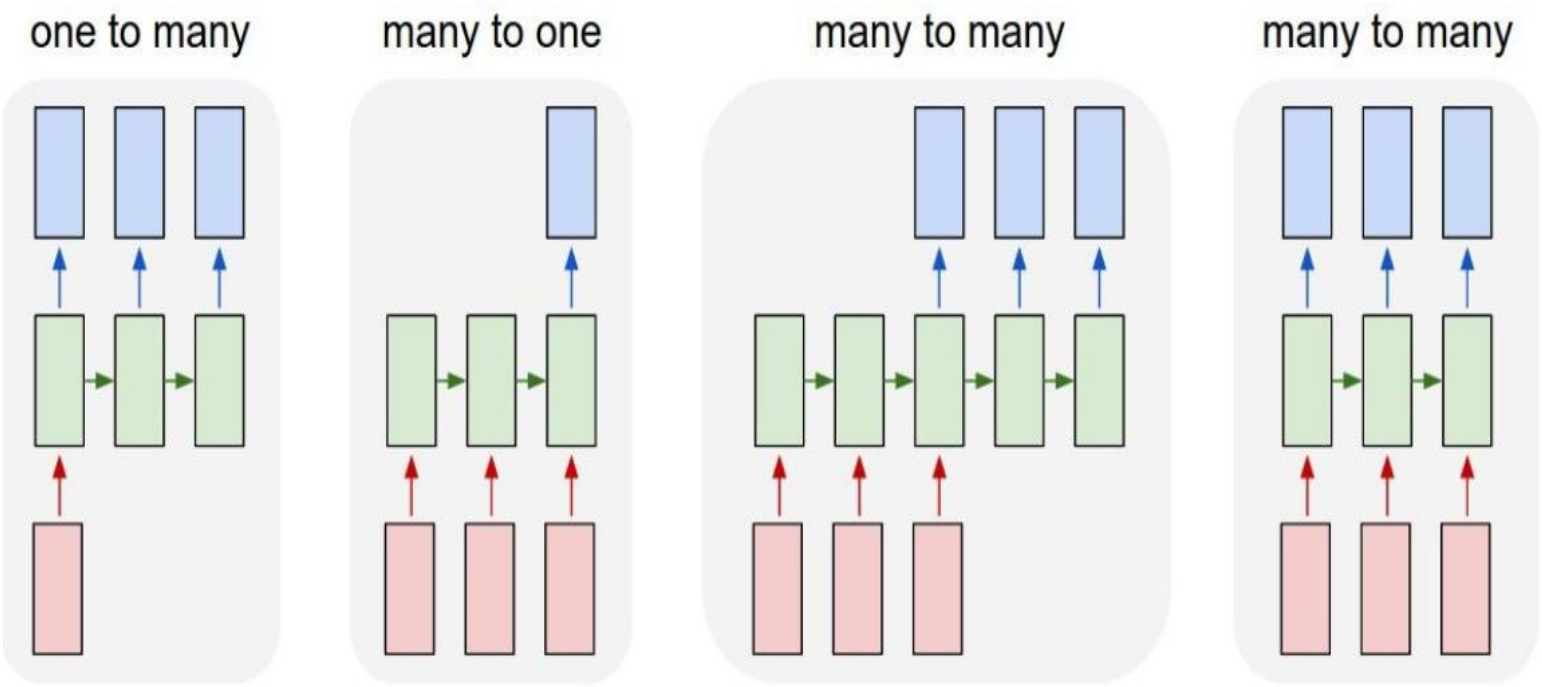
$$L\left(\{x^{(1)}, \dots, x^{(\tau)}\}, \{y^{(1)}, \dots, y^{(\tau)}\}\right)$$

$$= \sum_t L^{(t)} = -\sum_t log\ p_{model}\left(y^{(t)} | \{x^{(1)}, \dots, x^{(t)}\}\right)$$

- 网络的训练可以通过梯度下降法最小化上述损失函数，梯度的计算可以采用反向传播(BP)算法。
- 可以看到，展开的RNN事实上可以看作一般的非循环神经网络，只是它的参数是共享的。

循环神经网络 (RNN)

- 一些常见的RNN结构：



循环神经网络 (RNN)

- 梯度消失与爆炸：

  考虑一个不包含激活函数与输入的非常简单的RNN，它的计算如下：

$$h^{(t)} = W^T h^{(t-1)} = (W^t)^T h^{(0)}$$

  对W进行如下特征分解：

$$W = Q\Lambda Q^T$$

  $h^{(t)}$可以进一步简化为：

$$h^{(t)} = Q^T \Lambda^t Q h^{(0)}$$

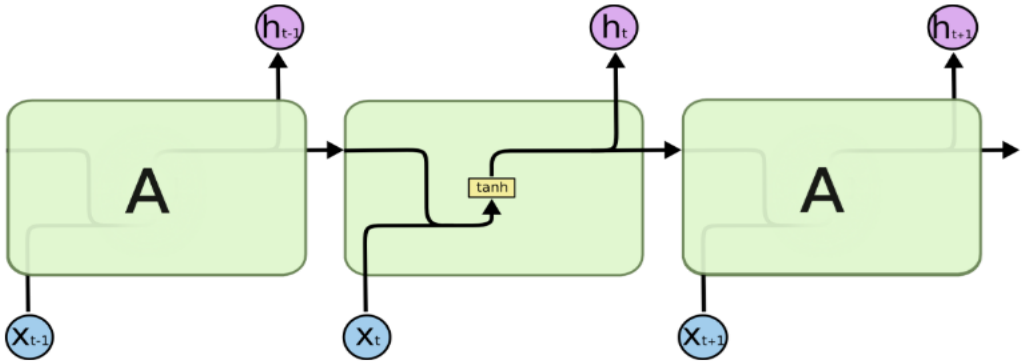
可以看到，当 t 较大时，绝对值大于1的特征值对应的部分将会迅速增大，而绝对值小于1的特征值对应的部分将会迅速减小，分别对应了梯度爆炸与梯度消失。

循环神经网络 (RNN-LSTM)

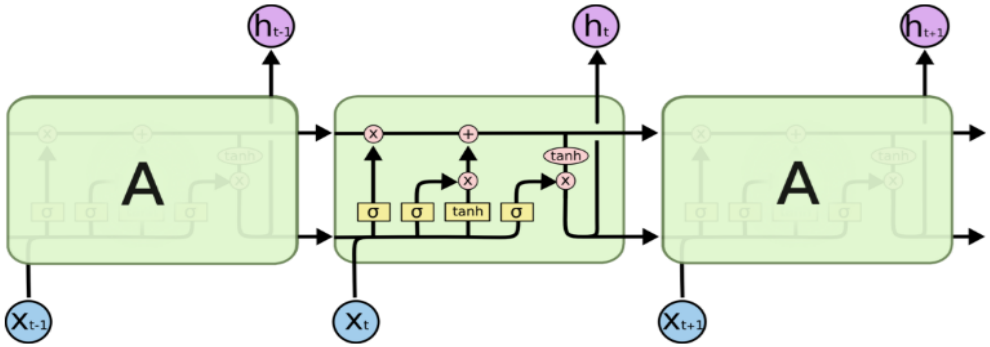
- 如前所述，RNN存在梯度爆炸与梯度消失的问题，这样网络无法处理长距离的依赖；
- ✓ 对于梯度爆炸，可以使用梯度裁剪的方式解决；
- ✓ 对于梯度消失的问题，LSTM维持一个记忆细胞， 保证信息在长期传输的过程中不会丢失；通过有选择地相加的方式将状态直接传到下一单元来减轻梯度消失的问题。

循环神经网络 (RNN-LSTM)

- LSTM与RNN的比较
- ✓ 最重要的是记忆细胞，与 RNN通过对旧的隐藏状态进行矩阵乘法的操作不同， LSTM计算的是记忆细胞的新旧状态的差异



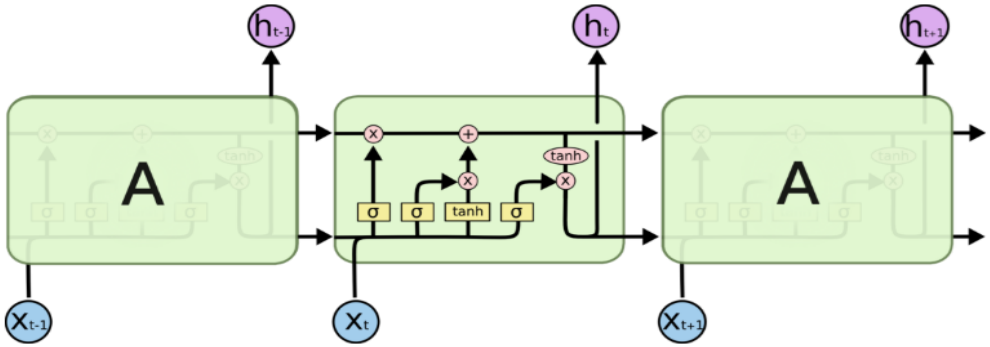
The repeating module in a standard RNN contains a single layer.



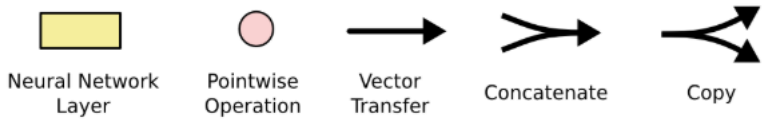
The repeating module in an LSTM contains four interacting layers.

循环神经网络 (RNN-LSTM)

- 网络结构



The repeating module in an LSTM contains four interacting layers.

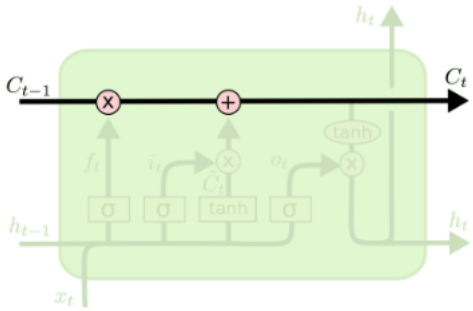


循环神经网络 (RNN-LSTM)

- 网络结构

✓ 给定上一时刻的隐藏状态$h^{(t-1)}$和本时刻的输入$x^{(t)}$

✓ 在不考虑控制门的情况下，LSTM的记忆细胞可简化为



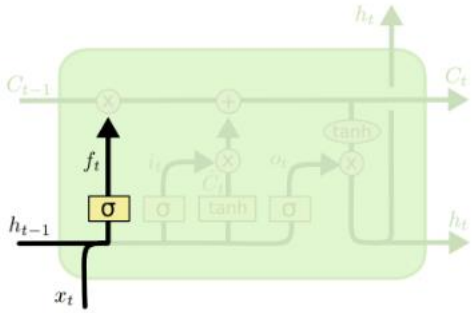
$$\tilde{c}^{(t)} = \tanh\left(W_c x^{(t)} + U_c h^{(t-1)}\right)$$

$$c^{(t)} = \tilde{c}^{(t)} + c^{(t-1)}$$

本质上就是新旧信息的叠加

循环神经网络 (RNN-LSTM)

- 网络结构
- ✓ 给定上一时刻的隐藏状态$h^{(t-1)}$和本时刻的输入$x^{(t)}$
- ✓ 添加遗忘门， 抛弃部分之前状态的信息



$$f^{(t)} = \sigma\left(W_f x^{(t)} + U_f h^{(t-1)}\right)$$

$$\tilde{c}^{(t)} = \tanh\left(W_c x^{(t)} + U_c h^{(t-1)}\right)$$

$$\mathrm{c}^{(t)} = \tilde{c}^{(t)} + c^{(t-1)} \odot f^{(t)}$$

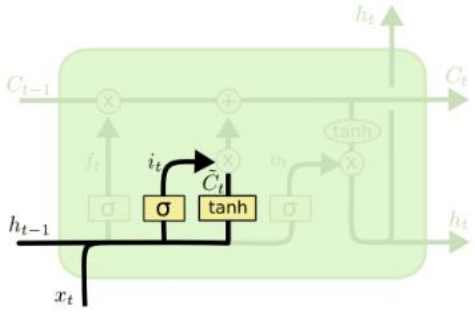
注：门控的激活函数是sigmoid是因为其输出值在0-1区间， 模拟了信息的取舍；其它采用tanh的因为其输出值有正负值，模拟了状态更新中的增减

循环神经网络 (RNN-LSTM)

- 网络结构

✓ 给定上一时刻的隐藏状态$h^{(t-1)}$和本时刻的输入$x^{(t)}$

✓ 添加输入门， 决定本状态保留的的信息

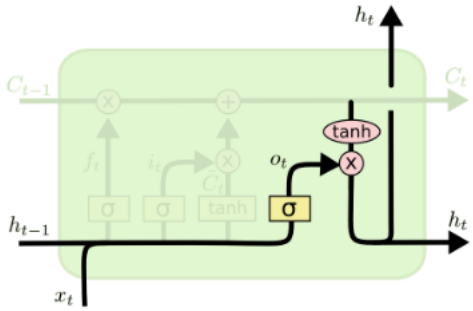


$$f^{(t)} = \sigma\left(W_f x^{(t)} + U_f h^{(t-1)}\right)$$
$$i^{(t)} = \sigma\left(W_i x^{(t)} + U_i h^{(t-1)}\right)$$

$$\tilde{c}^{(t)} = \tanh\left(W_c x^{(t)} + U_c h^{(t-1)}\right)$$
$$c^{(t)} = \tilde{c}^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)}$$

循环神经网络 (RNN-LSTM)

- 网络结构
- ✓ 给定上一时刻的隐藏状态$h^{(t-1)}$和本时刻的输入$x^{(t)}$
- ✓ 添加输出门，决定输出



$$f^{(t)} = \sigma\left(W_f x^{(t)} + U_f h^{(t-1)}\right)$$
$$i^{(t)} = \sigma\left(W_i x^{(t)} + U_i h^{(t-1)}\right)$$
$$o^{(t)} = \sigma\left(W_o x^{(t)} + U_o h^{(t-1)}\right)$$

$$\tilde{c}^{(t)} = \tanh\left(W_c x^{(t)} + U_c h^{(t-1)}\right)$$
$$c^{(t)} = \tilde{c}^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)}$$
$$h^{(t)} = \tanh(c^{(t)}) \odot o^{(t)}$$

DNN基础：总结

- DNN（深度神经网络）：一种多层的神经网络，采用一个或多个隐藏层学习数据暗含的特征，从而得到更好的数据表示
- 两种常见的DNN结构
  - CNN（卷积神经网络）：应用于类似网络结构数据，例如图像矩阵
    - 使用卷积和池化减少参数，减少噪声
  - RNN（循环神经网络）：应用于序列数据
    - 隐藏层之间的节点有连接
    - 梯度爆炸（特征值>1）与消失(特征值<1)：引入LSTM
- 后面介绍如何应用于信息检索

参考资料

- UFLDL教程：http://ufldl.stanford.edu/wiki/index.php/UFLDL%E6%95%99%E7%A8%8B
- Goodfellow I, Bengio Y, Courville A. Deep learning[M]. MIT press, 2016. http://www.deeplearningbook.org/
- cs231n slides: http://cs231n.stanford.edu/2016/syllabus
- Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures[C]//Proceedings of the 32nd International Conference on Machine Learning (ICML-15). 2015: 2342-2350.
- Lipton Z C, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning[J]. arXiv preprint arXiv:1506.00019, 2015.
- Bishop C. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer[M]// Stat Sci. 2006:140-155.
- Christopher Olah. Understanding LSTM Networks.

- 注：本小节所有图均来自上述材料，为了简洁未一一注明，特此说明。

提纲

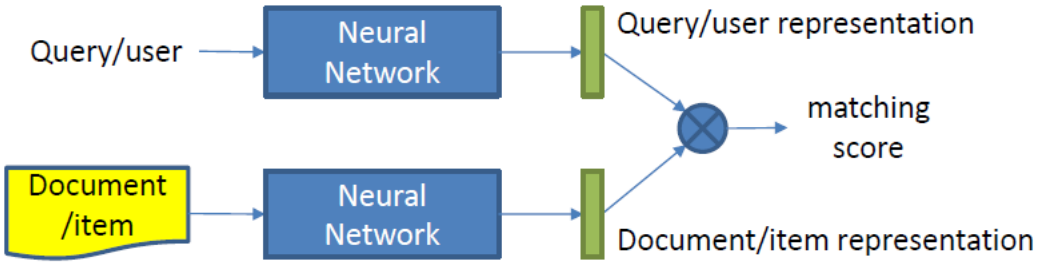
1. 上一讲回顾
2. 深度神经网络(DNN)基础
3. Neural IR Model

Neural IR模型分类

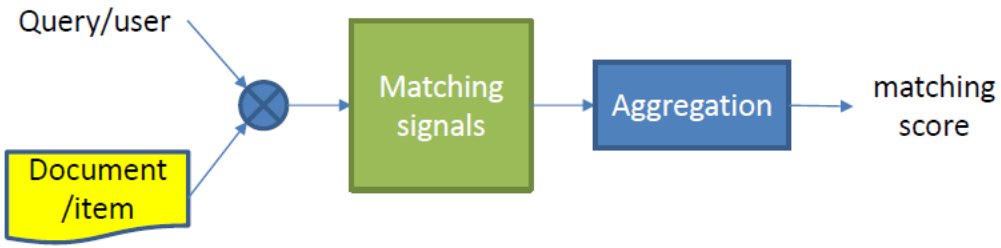
- Representation based：学习文本的分布式表示， 在高维空间匹配
  - 词表示：one hot → distributed
  - 句子表示：bag of words → distributed
  - 匹配能力取决于学习文本表示的算法能力
  - 代表模型：DSSM, CDSSM,
- Matching function：文本之间先进行交互匹配，再对匹配信号进行融合
  - 输入：比较底层的输入
  - 匹配函数：cosine, dot product → NN
  - 优点， 可以考虑更加丰富的匹配信号， 如软匹配 (soft matching)
  - 代表模型： MatchPyramid, DRMM, K-NRM, PACRR, NPRF
- Combination of both: 既考虑Representation又考虑Matching function
  - 代表模型：Duet

两类模型图示

- Representation方法



- Matching function 方法



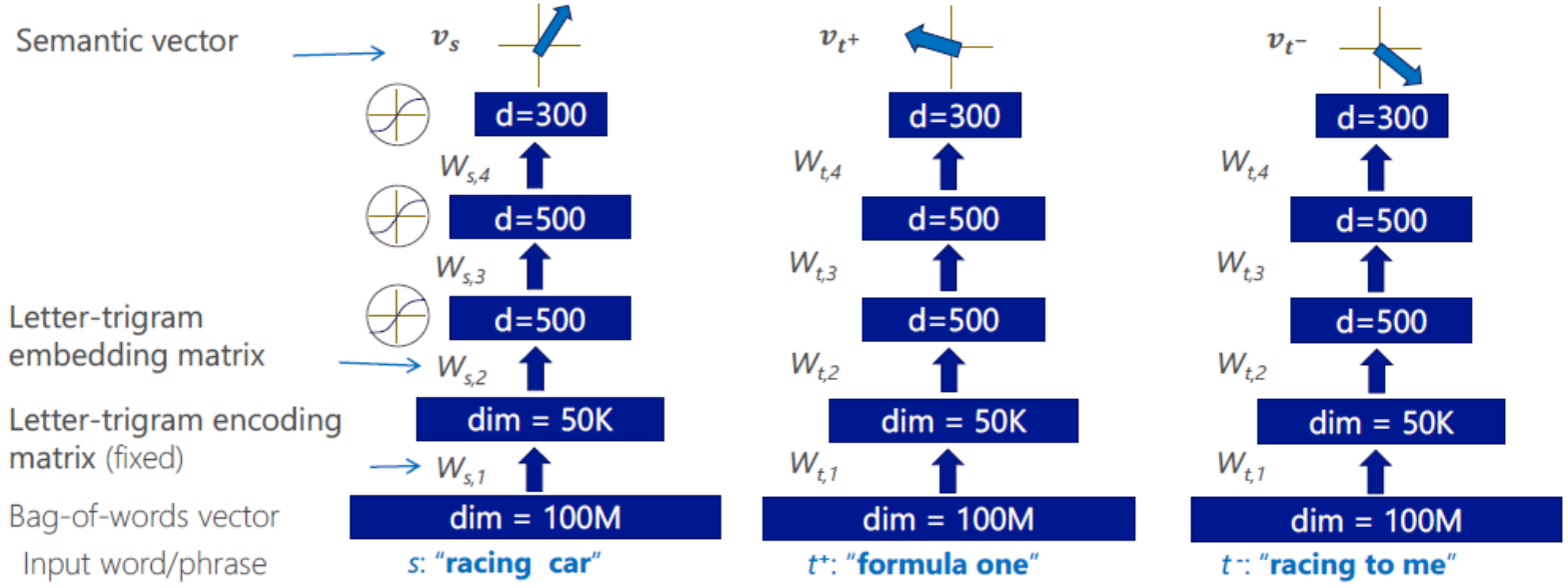
区别在于匹配过程中，query与document交互的方式

DSSM

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, Larry P. Heck. Learning Deep Structured Semantic Models for web search using clickthrough data. CIKM 2013: 2333-2338

- word hashing: Bag of letter-trigrams representation
  - “#candy# #store#” --> #ca can and ndy dy# #st sto tor ore re#
  - 优点：降维，未见词泛化， 对错误拼写的鲁棒性

- 模型：DNN学习查询，文本的语义表示， cosine相似度作为匹配评分



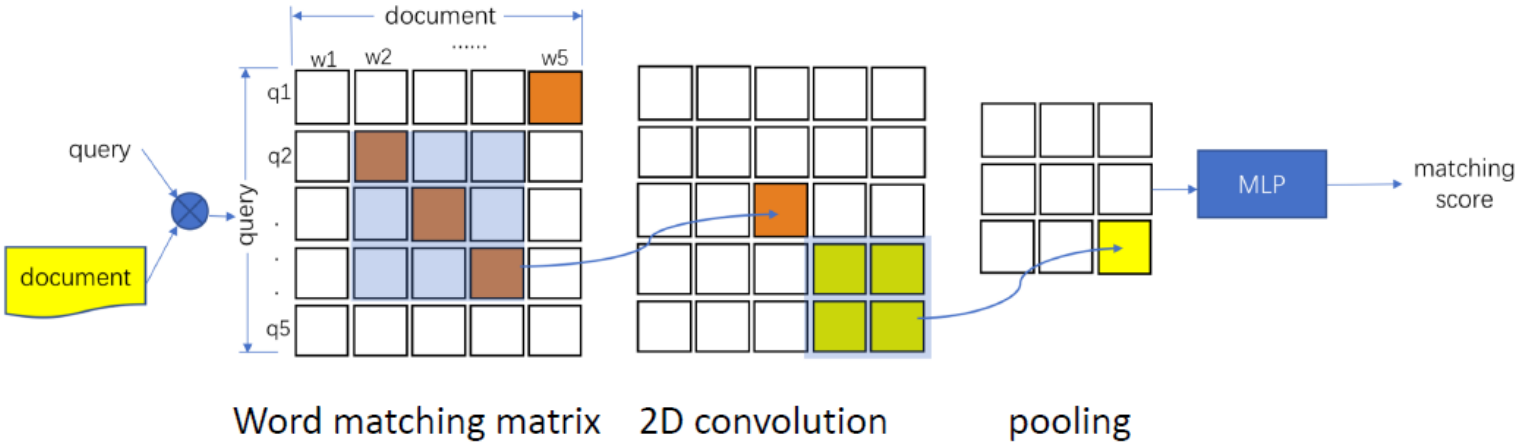
DSSM

- 一般而言， Query/Doc Representation 只学习文本的语义表示， 而在信息检索中，相关性/精确 匹配 (relevance/exact matching) 是最重要的匹配模式， 语义/软 匹配（semantic/soft matching）才是其次，故Representation的方法具有很大的局限性
  - Query: Chinese Cuisine
  - Doc 1: Chinese food
  - Doc 2: Japanese Cuisine

- 后面介绍的基于Matching function的方法考虑了精准匹配，软匹配 (soft match, 例如 boat - ship) 等匹配模式， 才使得基于DNN的NIR模型能超越传统检索模型如BM25, QL等。

MatchPyramid

- 动机
  - 考虑各种层次的匹配信号，包括单词层次、短语层次以及句子层次等等；
  - 在图像领域， 基于CNN特征提取的图像金字塔被证明是有效的
- 模型



Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Xueqi Cheng: A Study of MatchPyramid Models on Ad-hoc Retrieval. CoRR abs/1606.04648 (2016)

MatchPyramid (cont')

MatchPyramid 是一个比较原始的工作，后面出现了一些更加有效的模型，一般是从以下的方面入手：

- Word matching matrix是有效的匹配模型的基础
- 短查询 vs 长文档：Heterogeneous
- 不同查询词之间的重要性
- 是否考虑匹配位置信息
- 现在的一般做法
  - 计算单个/多个查询词对整个文档的匹配信号
  - 匹配信号 $\xrightarrow{\text{NIR模型}}$ 匹配的分布
  - 对所有查询词的匹配分布进行融合

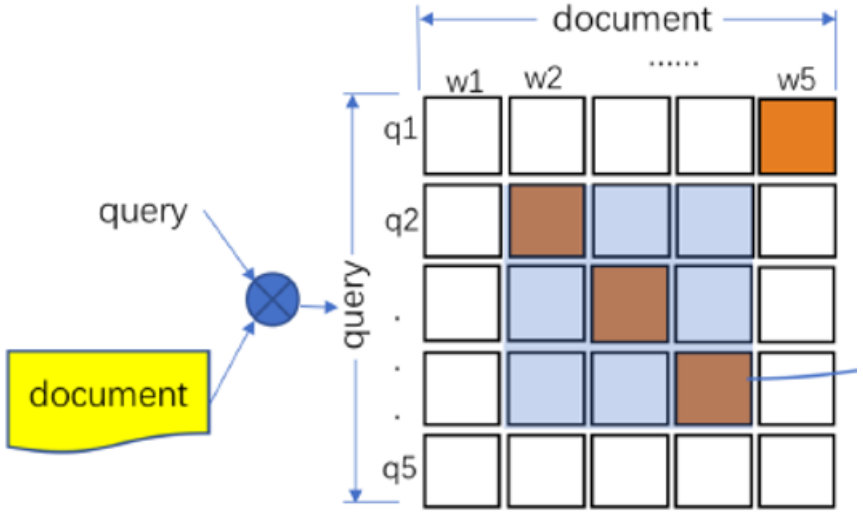
DRMM（Deep Relevance Matching Model)

- 背景与基本思想：
  - 现有的基于DNN的检索模型将检索任务视为两段文本的匹配任务，更多地关注语义匹配 (Semantic Matching)，即所谓软匹配
  - 相比普通的文本匹配任务，检索任务更需要关注相关性匹配 (Relevance Matching)
  - 通过显式地对精确匹配信号 (Exact Matching Signals)，查询词重要度 (Query Term Importance)，以及多样匹配要求 (Diverse Matching Requirement)进行建模，得到的模型更加适合于检索任务
  - 借鉴了图像领域的统计灰度直方图的方法

Jiafeng Guo, Yixing Fan, Qingyao Ai, W. Bruce Croft: A Deep Relevance Matching Model for Ad-hoc Retrieval. CIKM 2016: 55-64

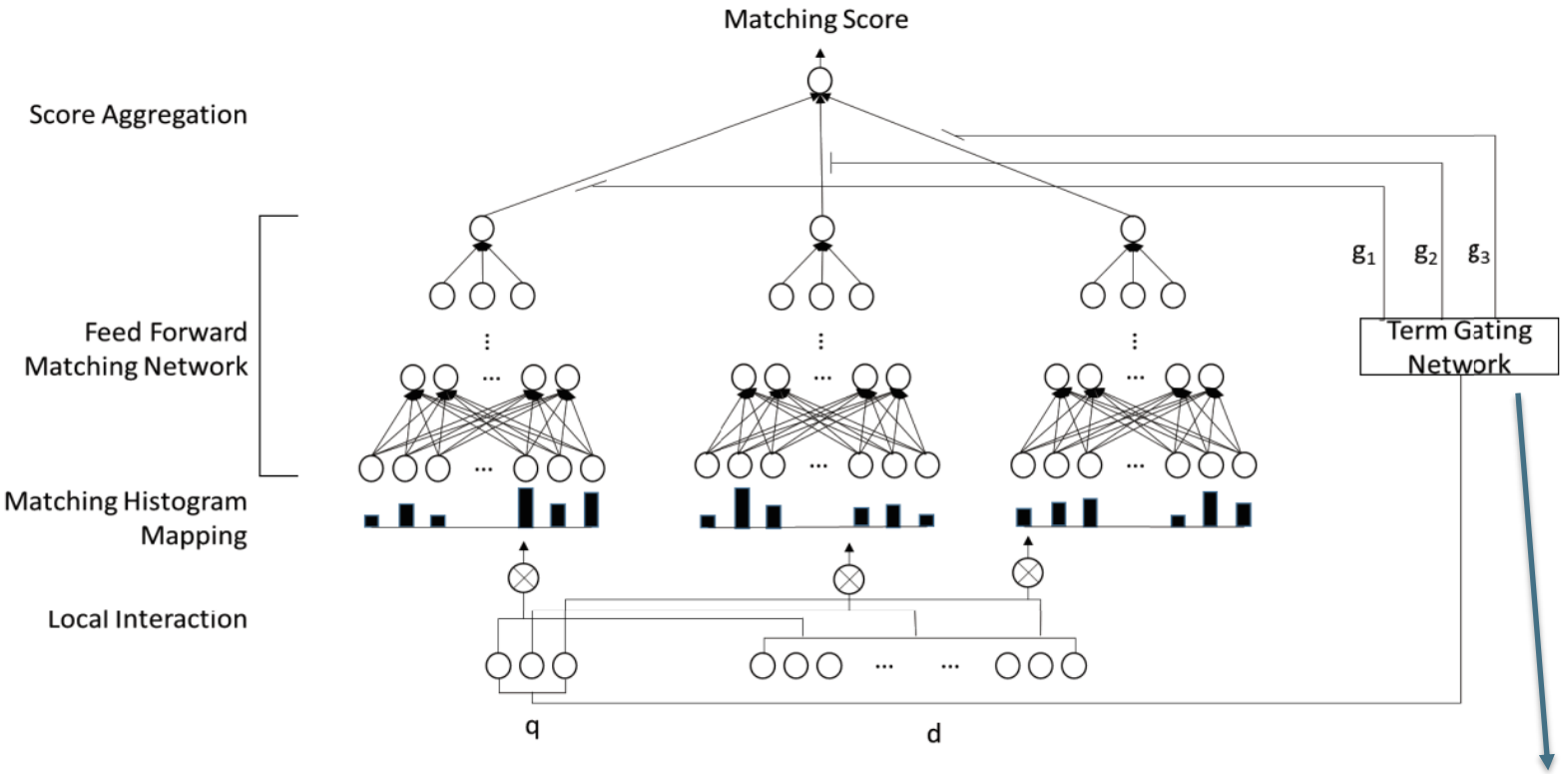
构建交互矩阵与分布计数直方图

- 构建查询(q) – 文档(d)相似度(交互)矩阵



- 对每一行（即每一个查询词）统计矩阵中相似度数值的分布区间计数（Count），一般是11个区间，即[0, 1]以0.1为单位

- Count取对数 (LCH)，然后输入到前馈网络
  - 每个查询词对应一个前馈网络
- Softmax(前馈网络输出评分 * IDF)，线性加和得到文档最终评分



DRMM模型结构

来源： Guo etc., CIKM 2016

Idf衡量查询词重要性

- 实验设置
  - 与baseline比较：QL，BM25，DSSM，CDSSM，ARC-I，ARC-II，MatchPyramid
  - 余弦相似度计数变换方式与查询词权重计算方式对模型的影响：直接使用计数(CH)，除以总数(NH)，取对数(LCH)；输入查询词向量(TV)，输入查询词逆文档频率(IDF)
- 实验结果

Robust-04 collection

Model Type	Model Name	Topic titles			Topic descriptions		
		MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
Traditional Retrieval Baselines	QL	0.253	0.415	0.369	0.246	0.391	0.334
	BM25	0.255	0.418	0.370	0.241	0.399	0.337
Representation-Focused Matching Baselines	$DSSM_D$	0.095^-	0.201^-	0.171^-	0.078^-	0.169^-	0.145^-
	$CDSSM_D$	0.067^-	0.146^-	0.125^-	0.050^-	0.113^-	0.093^-
	ARC-I	0.041^-	0.066^-	0.065^-	0.030^-	0.047^-	0.045^-
Interaction-Focused Matching Baselines	ARC-II	0.067^-	0.147^-	0.128^-	0.042^-	0.086^-	0.074^-
	MP_{IND}	0.169^-	0.319^-	0.281^-	0.067^-	0.142^-	0.118^-
	MP_{COS}	0.189^-	0.330^-	0.290^-	0.094^-	0.190^-	0.162^-
	MP_{DOT}	0.083^-	0.159^-	0.155^-	0.047^-	0.104^-	0.092^-
Our Approach	$DRMM_{CH\times TV}$	0.253	0.407	0.357	0.247	0.404	0.341
	$DRMM_{NH\times TV}$	0.160^-	0.293^-	0.258^-	0.132^-	0.217^-	0.186^-
	$DRMM_{LCH\times TV}$	0.268^+	0.423	0.381	0.265^+	0.423^+	0.360^+
	$DRMM_{CH\times IDF}$	0.259	0.412	0.362	0.255	0.410^+	0.344
	$DRMM_{NH\times IDF}$	0.187^-	0.312^-	0.282^-	0.145^-	0.243^-	0.199^-
	$DRMM_{LCH\times IDF}$	$\mathbf{0.279}^+$	$\mathbf{0.431}^+$	$\mathbf{0.382}^+$	$\mathbf{0.275}^+$	$\mathbf{0.437}^+$	$\mathbf{0.371}^+$

DRMM：总结

- DRMM是第一个在TREC数据集能够取得比传统检索模型更好效果的基于DNN模型
- DRMM的设计思路在一定程度上借鉴了传统的TF-IDF
  - 匹配count取Log，term gating取IDF
- 性能限制：直方图计算是不可微的， 不能在GPU中计算， 并且需要遍历整个矩阵，所以当数据规模大的时候， DRMM会比较慢， 限制了其应用场景

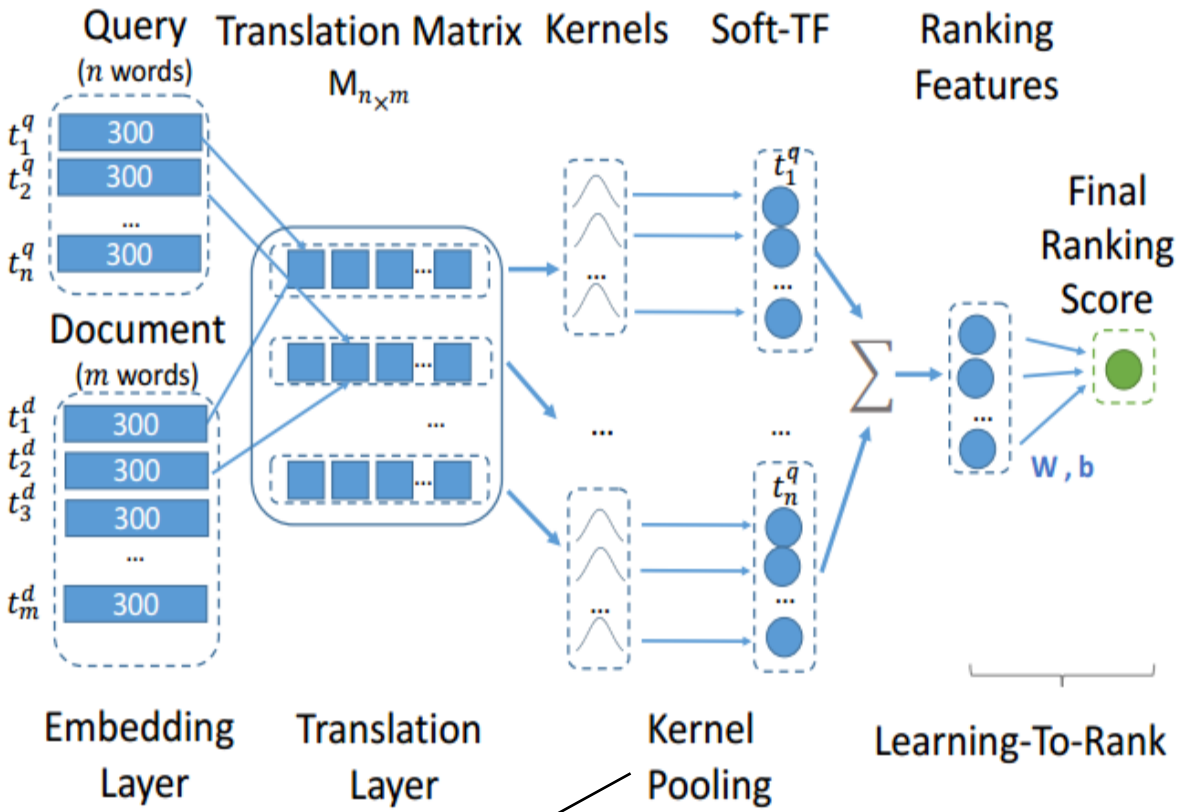
K-NRM (Kernel-based Neural Relevance Model)

- 可视为对DRMM的改进
- 使用kernel-pooling技术提取多层次的软匹配 (soft-match) 特征
- 软匹配特征输入learning-to-rank层获取最终排序评分。

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, Russell Power: End-to-End Neural Ad-hoc Ranking with Kernel Pooling. SIGIR 2017: 55-64

模型结构

Embedding Layer 将单词映射为其分布式表示；查询词与文档词之间的相似度构成Translation Matrix；将K个核作用于每个查询词对应的Translation Matrix的一行，得到每个查询词对应的K维软匹配特征，求和得到最终的Ranking Features；一个Learning-to-rank层作用于最终的特征得到最终的排序评分。



使用RBF 核函数将矩阵每一行转化为一个对数评分 Soft-TF

$$\phi(M) = \sum_{i=1}^{n} \log \vec{K}(M_i)$$

K-NRM

- 模型
  - 核心：Kernel pooling
    - 对word matching matrix的每一行，用多个高斯核提取特征。 每个高斯核的μ代表了其统计的单词-单词相似度的均值，如μ=1.0代表此高斯核统计的是精准匹配信号， 其它的则是软匹配的信号；每个高斯核的σ代表的是核的宽度，如μ=1.0的高斯核的σ很小，σ=0.00001，代表了其只考虑精准匹配。

$$K_k(M_i) = \sum_j \exp(-\frac{(M_{ij} - \mu_k)^2}{2\sigma_k^2})$$

- 讨论
  - K-NRM是对DRMM的改造：用RBF核代替了直方图计数， RBF核是可微的，因此K-NRM比DRMM的效率高很多
  - RBF核能提取到很好的软匹配特征

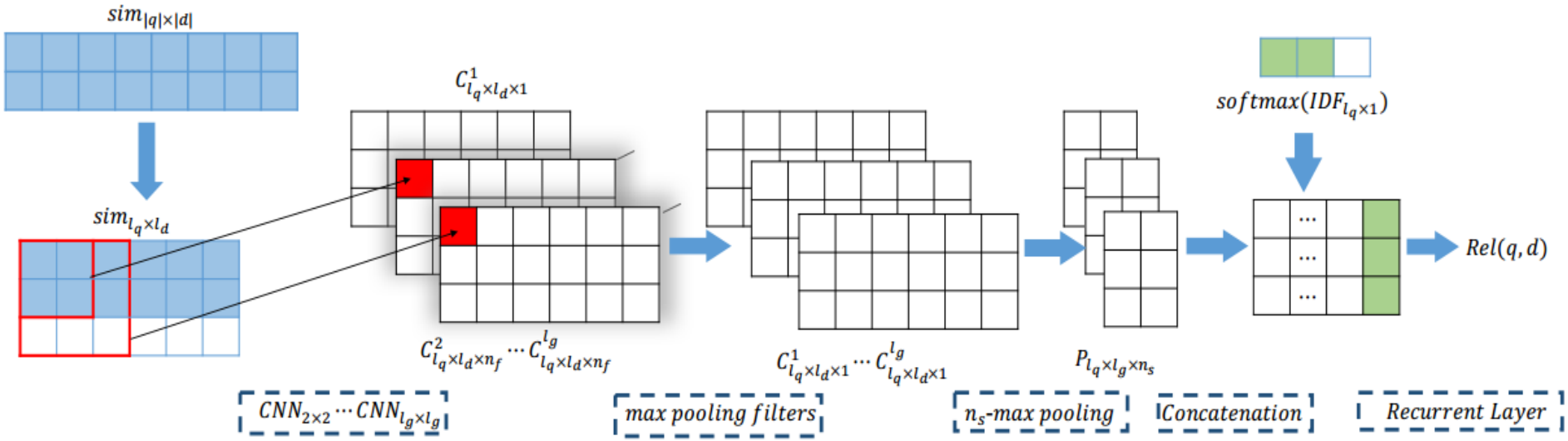
PACRR (Position Aware Convolutional Recurrent Relevance Model)

- [Hui etc., EMNLP 2017]
  - 背景与基本思想：
    - 现有基于DNN的检索模型主要基于unigram单词匹配，对于位置相关的匹配信息 (如term proximity和term dependencies) 的建模还没有充分的研究；
    - 本文通过将具有不同大小 (k=2,…,lg) 卷积核的卷积层作用于查询与文档间的单词-单词相似度矩阵，来对k-gram匹配信息进行建模。

Hui K, Yates A, Berberich K, et al. PACRR: A Position-Aware Neural IR Model for Relevance Matching. EMNLP 2017: 1060-1069.

模型结构

- 首先，计算查询与文档之间的单词-单词相似度矩阵$sim_{|q|x|d|}$，并通过裁剪或者补齐等方式得到固定大小的矩阵sim_{lqxld}；对于核大小为k x k (k=2,…,lg) 的卷积层，用n_f个卷积核作用于矩阵sim_{lqxld}并对卷积核维度进行max pooling，得到与sim_{lqxld}形状相同的矩阵；之后，对文档维度进行n_s-max pooling，并将不同的k值对应的结果以及查询词的IDF信息以查询词为基准连接起来；最后将查询词向量送入RNN得到最终评分。



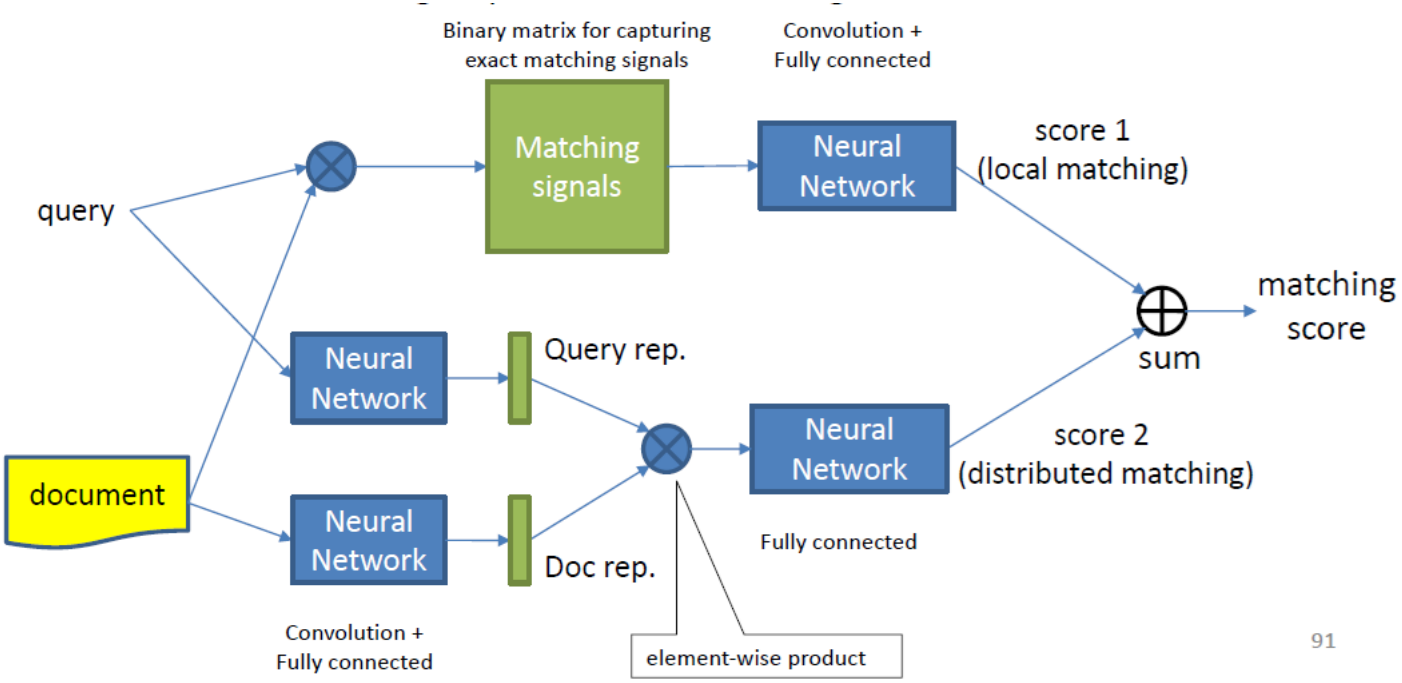
PACRR模型结构

来源：Hui etc., EMNLP 2017

作者后续研究表明使用前馈网络结果更好

DUET

- 动机
  - Representation 与 Matching function的方法是互补的
- 模型



Bhaskar Mitra, Fernando Diaz, Nick Craswell: Learning to Match using Local and Distributed Representations of Text for Web Search. WWW 2017: 1291-1299

实验结果

Method	NDCG@20	ERR@20
QL 传统查询似然模型	0.231	0.131
MatchPyramid (Pang et al. 2016)	0.278	0.176
DRMM (Guo et al. CIKM16)	0.300	0.193
K-NRM (Xiong et al. SIGIR17)	0.324	0.201
DUET (Mitra et al. WWW17)	0.267	0.179
PACRR-firstk (Hui et al. EMNLP17)	0.339	0.221

在Web Track 14 dataset 上的结果(Hui et al., EMNLP'17)

上述方法的问题：效果

- 缺少PRF机制
  - 比较实验：缺少PRF baseline
- 事实上，文档通常很长，查询通常都非常短，匹配信号通常较为稀疏，需要反馈文档补充查询相关的信息
- 已知PRF是一种非常有效的提升经典IR模型方法，如在Robust04数据集上

模型	NDCG@20
BM25 传统模型	0.4158
BM25+QE 传统模型+PRF	0.4353
QL+RM3 传统模型+PRF	0.4398
DRMM (Guo et al. CIKM16)	0.4297
K-NRM (Xiong et al. SIGIR17)	0.3989
PACRR (Hui et al. EMNLP18)	0.4082

上述方法的问题：效率

- 本质上是重排算法，依赖unsupervised 初始检索结果
  - 例如大多数工作都是重排BM25的前1000个结果
- 交互矩阵的实时构建带来效率问题

SNRM (Standalone Neural Ranking Model)

- 解决效率问题
  - 背景与基本思想：
    - 现有基于DNN的检索模型主要基于召回+重排的模式，若直接学习文本的稀疏向量表示，可以将问题简化成直接召回文档的模式
    - 对稀疏矩阵构建倒排索引，使得retrieval time的开支与传统BM25模型相似，解决效率问题

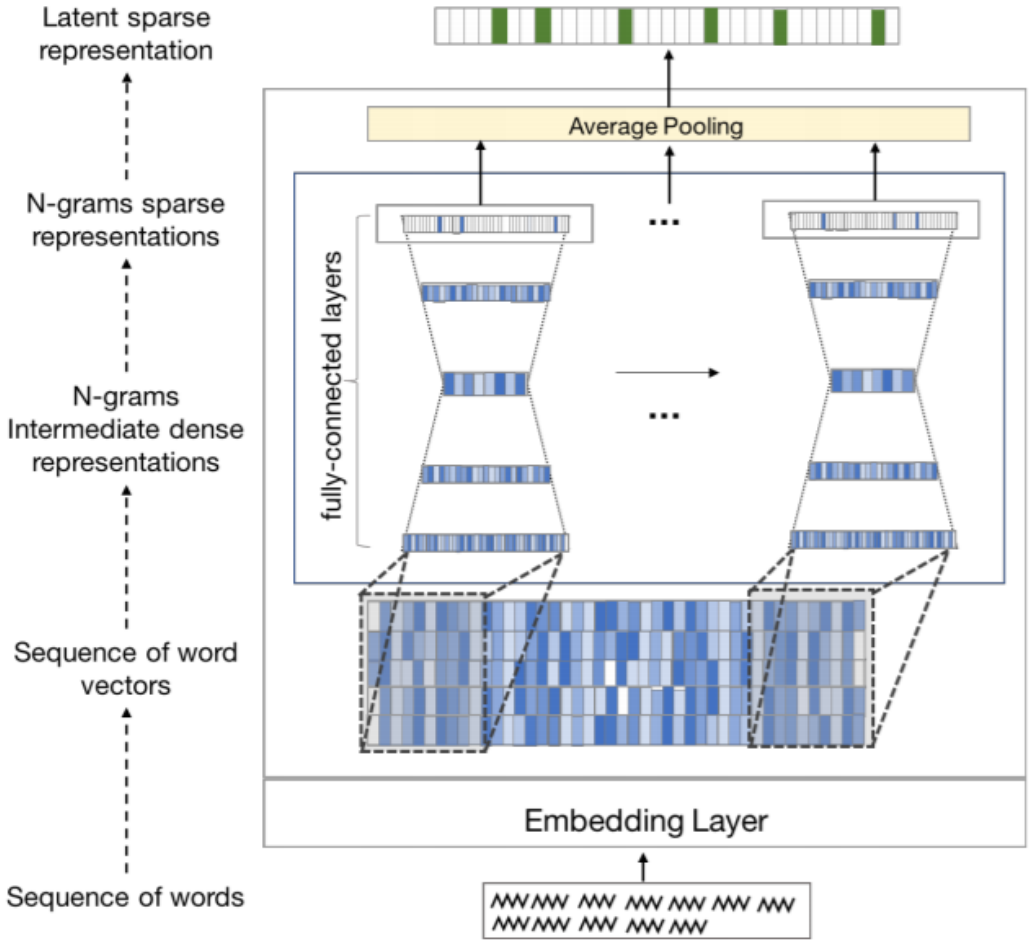
Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, et al.,: From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. CIKM 2018: 497-506

SNRM

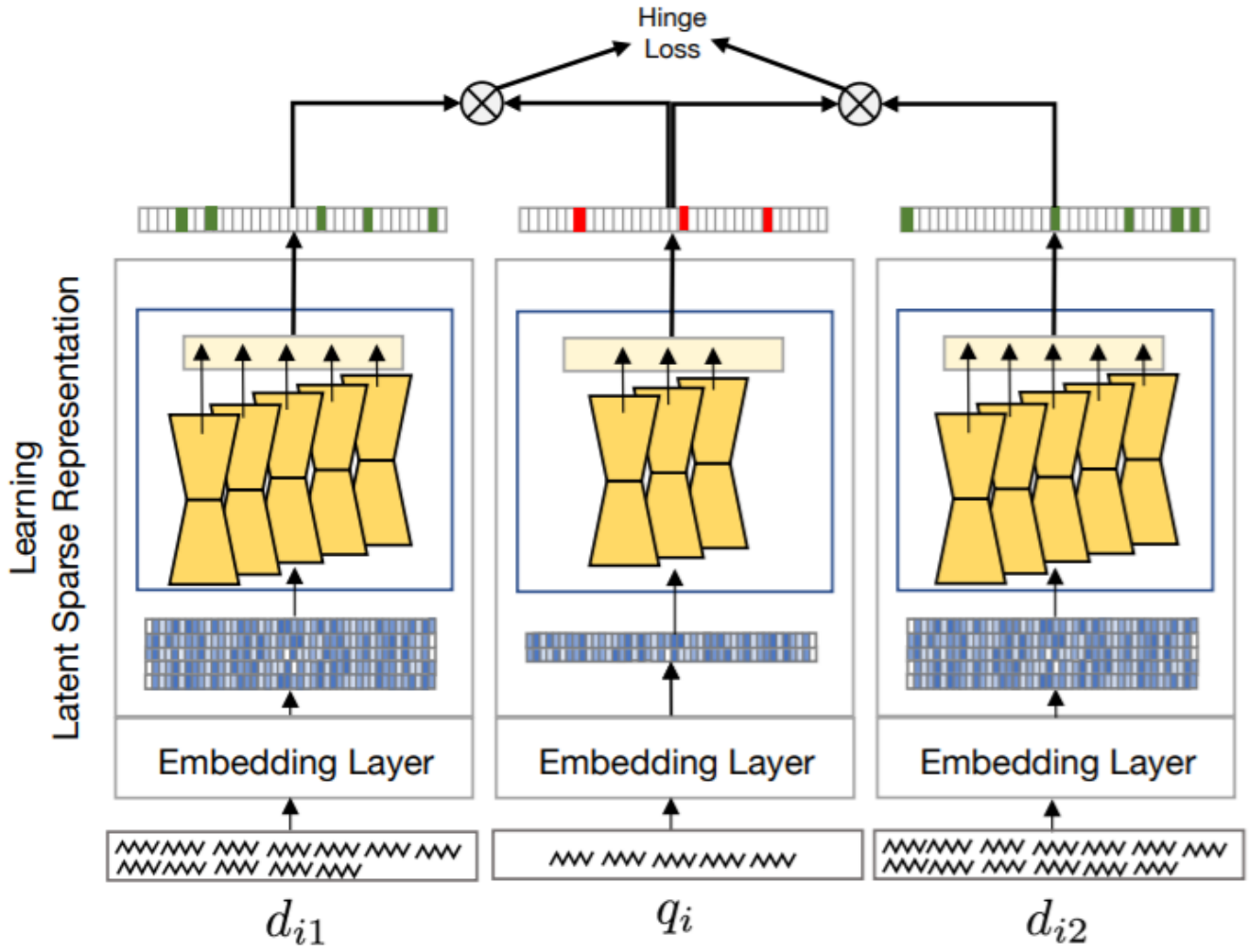
1. 类似于CNN中取window size为n，步长为1的核， 提取text snippet， 经由MLP学习ngram的表示

2. 采用的MLP是滴漏形状的，输出层的大小设置为20，000，保证稀疏性

3. 所有snippet的稀疏表示做平均池化作为文本的向量表示



SNRM： 监督学习得到文本稀疏表示



(a) Training time

SNRM： Inference

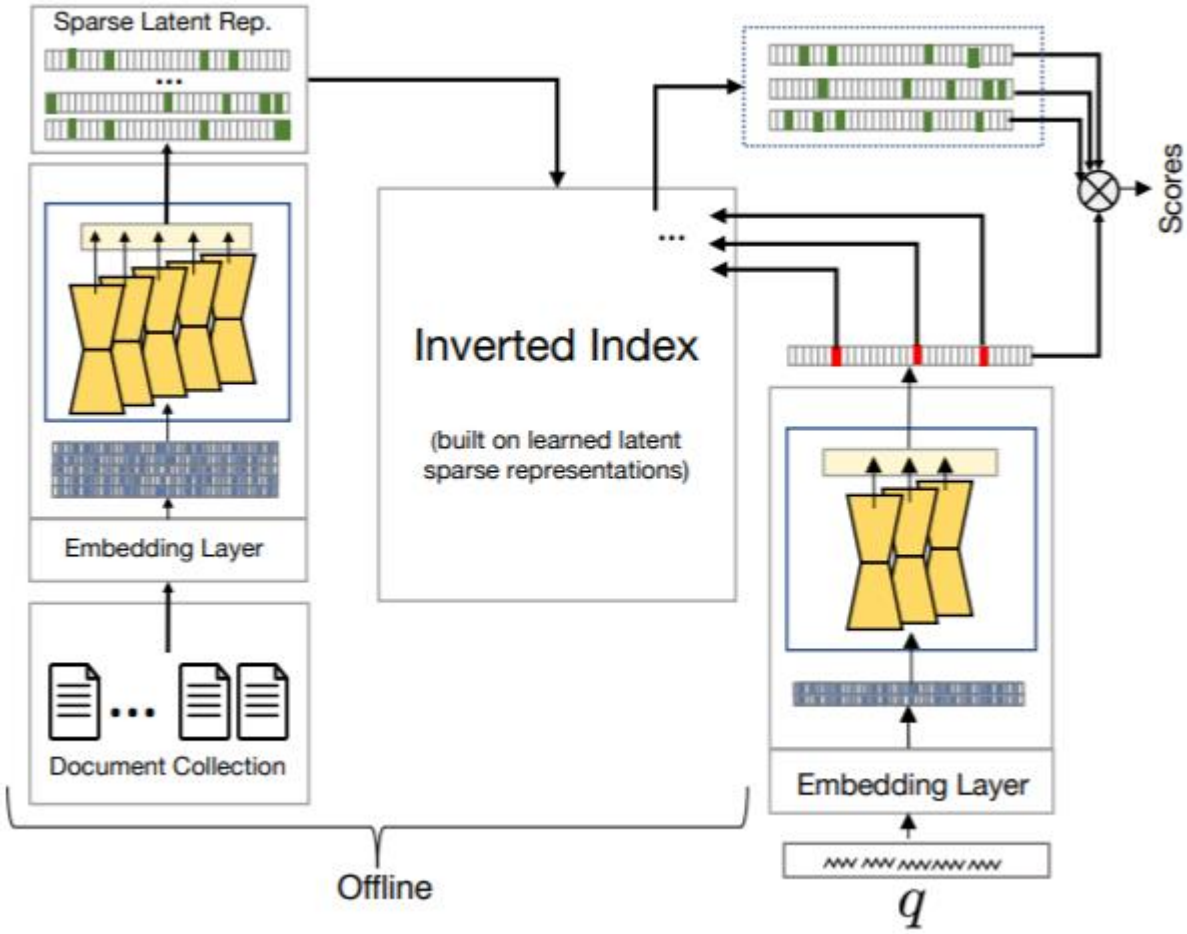
1. 为稀疏向量构建倒排索引

2. 文档得分：

$$\text{retrieval score}(q,d) = \sum_{\vec{q}_i|_{>0}} \vec{q}_i \vec{d}_i$$

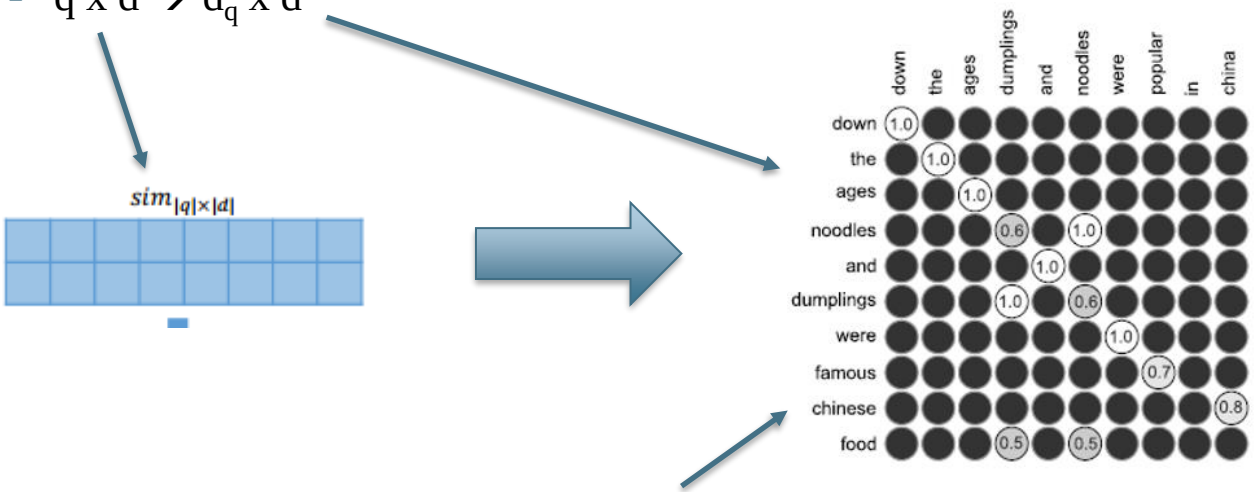
2. +PRF（伪相关反馈）：

$$\vec{q}^* = \vec{q} + \alpha \frac{1}{k} \sum_{i=1}^{k} \vec{d}_i$$



Neural PRF (NPRF), EMNLP18

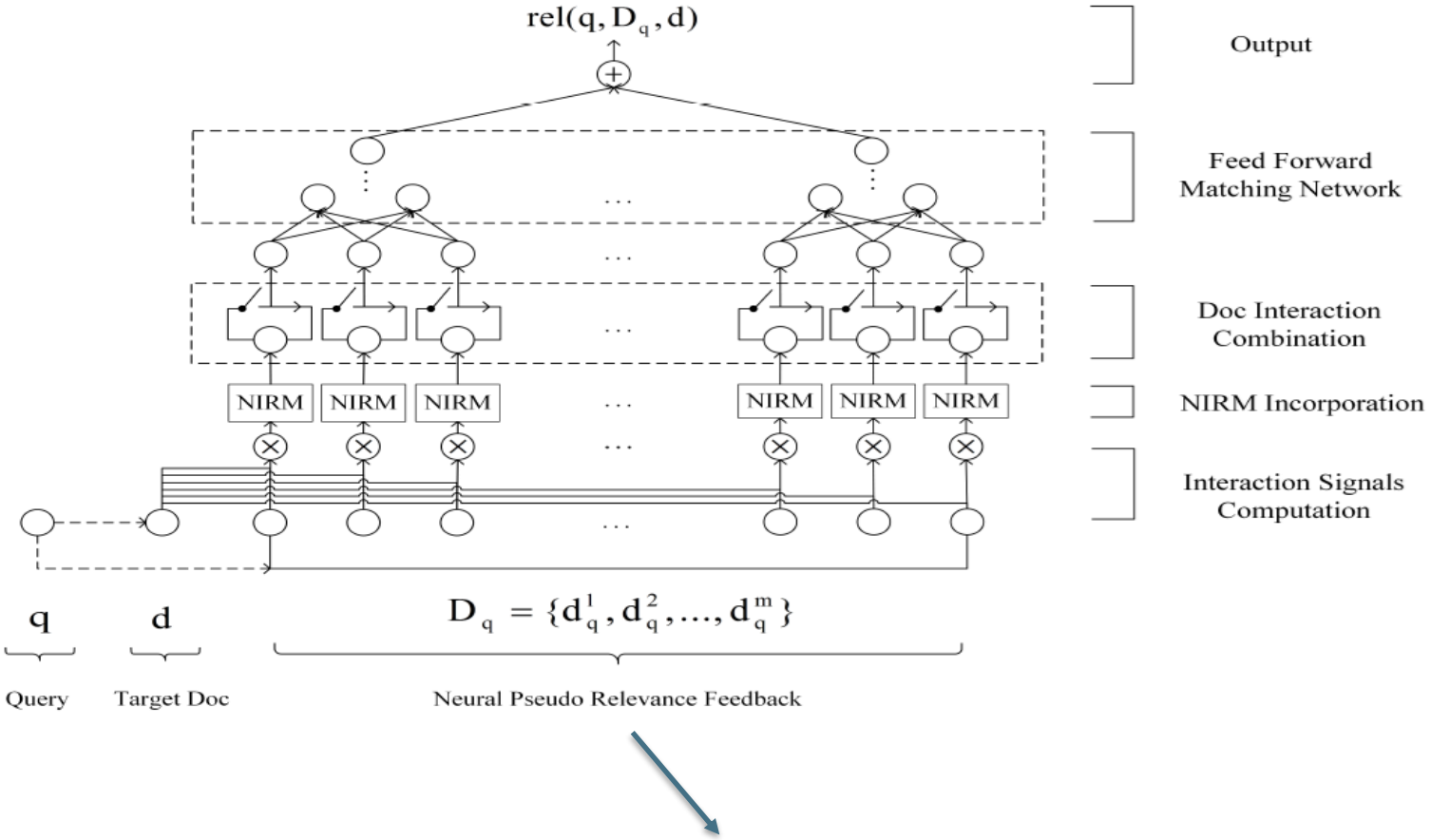
- 基本思想
  - 将反馈文档视为原始查询的扩充表示
  - 通过增强与查询相关的信息匹配信号获得更好的交互矩阵
    - q x d → d_q x d



  - 反馈文档 匹配 待评分文档

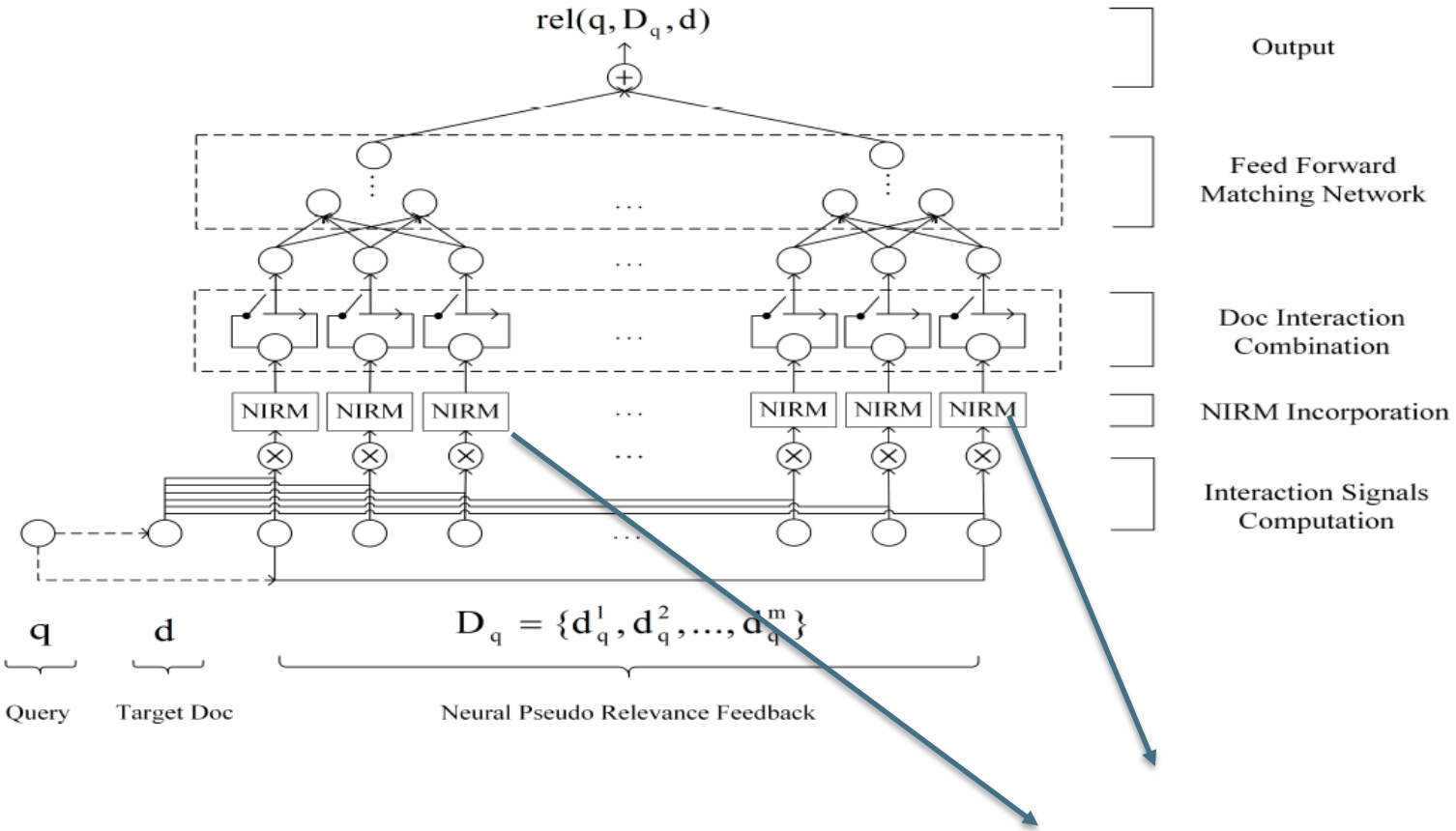
Canjia Li, Yingfei Sun, Ben He, Le Wang, Kai Hui, Andrew Yates, Le Sun, Jungang Xu: NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval. EMNLP 2018: 4482-4491

Neural PRF (NPRF), EMNLP18



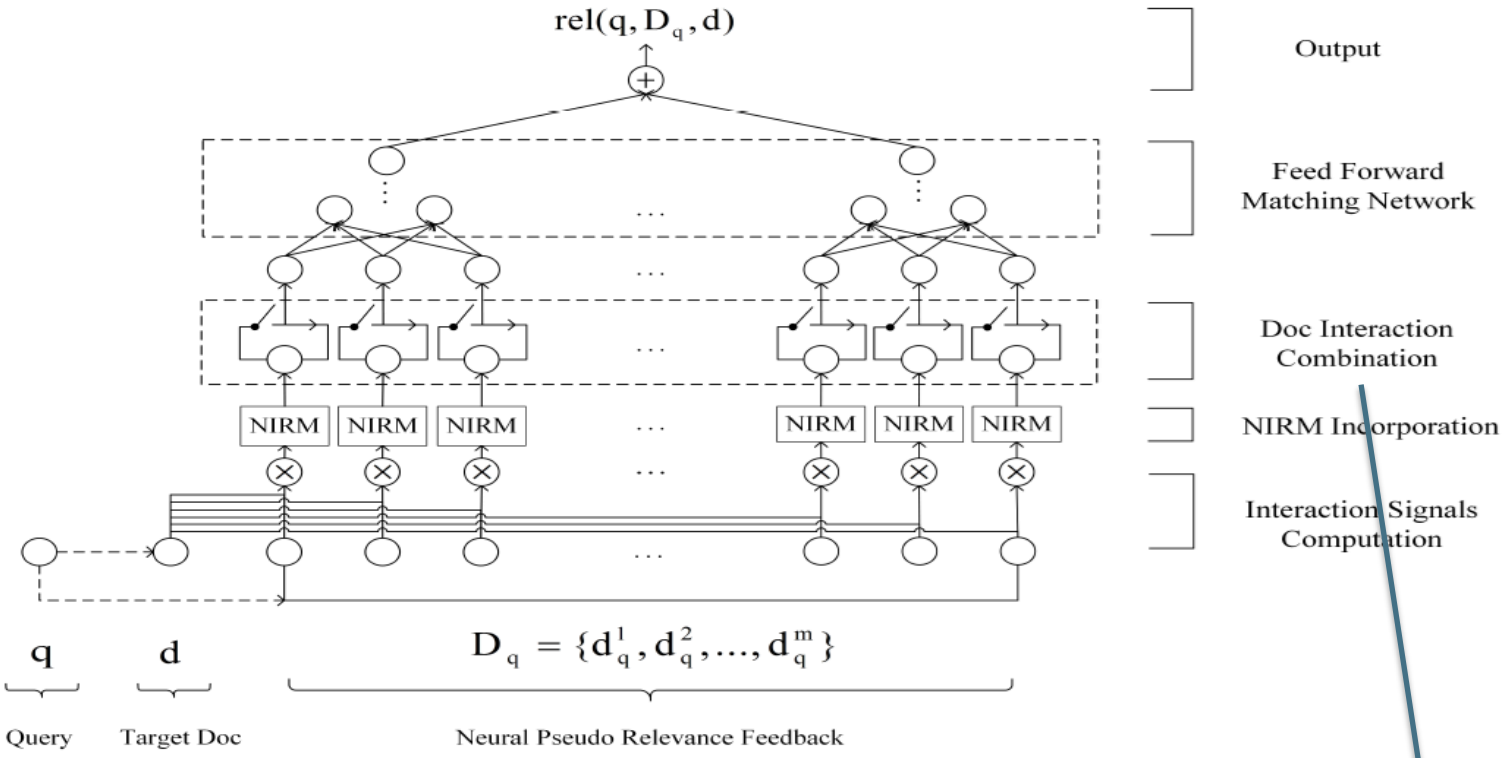
- 提取top m 个伪相关文档， 每个伪相关文档都当作是查询的一个扩充表示

Neural PRF (NPRF), EMNLP18



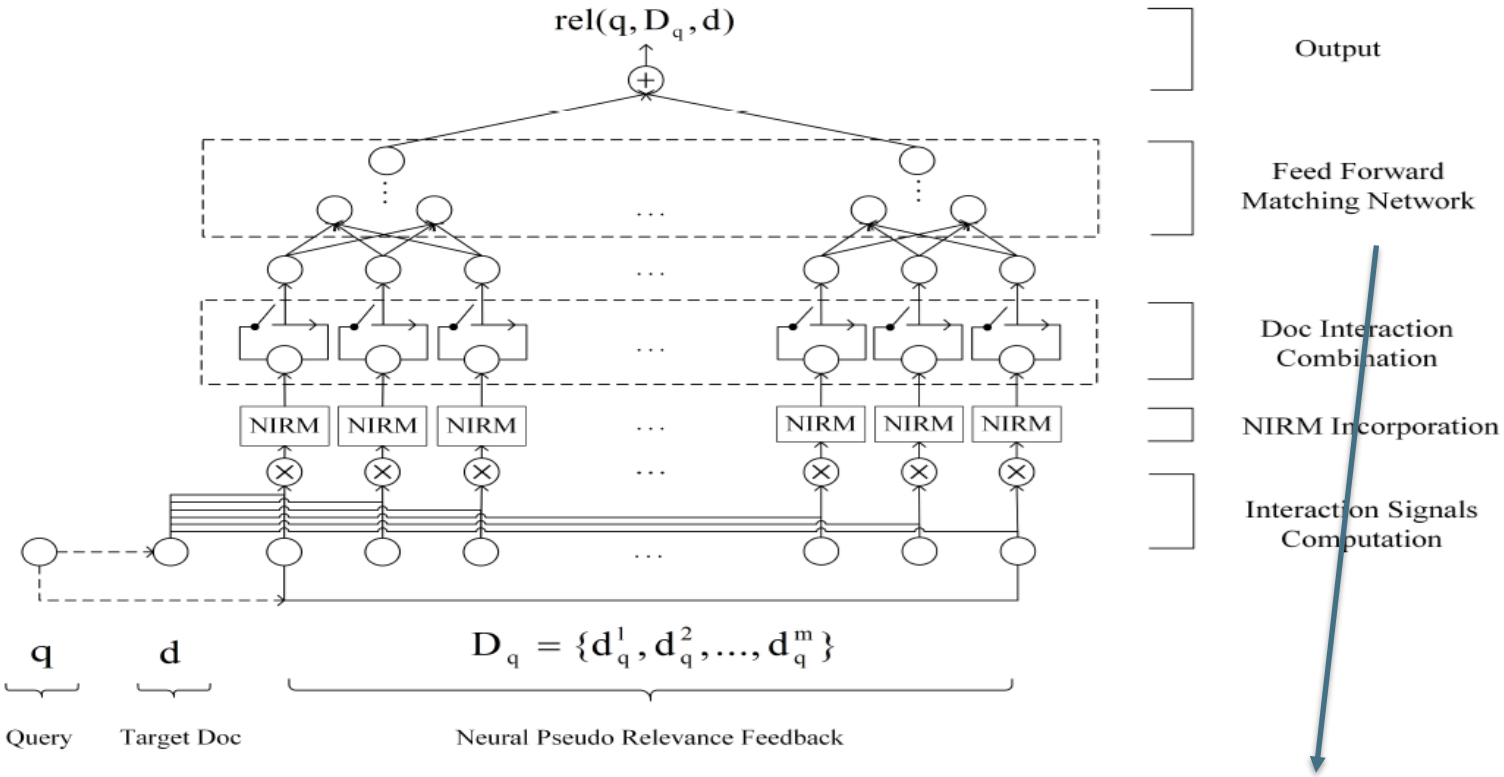
- 已有模型如DRMM、K-NRM作为基本评分功能模块
- 利用反馈文档得到更密集的交互信号矩阵
- 每篇文档的bag-of-words表示作为查询， 用DRMM/K-NRM对待评分文档进行打分

Neural PRF (NPRF), EMNLP18



- 乘以原始BM25 score区分不同反馈文档的质量
- 抑制低质量反馈文档的影响

Neural PRF (NPRF), EMNLP18



- 所有伪相关文档的分数进行融合

实验设置

- Standard TREC 1-3 and Robust04 datasets
- Loss function: hinge loss on 0/1/2 labels
- Report on common IR metrics
  - Mean Average Precision (MAP)
  - Normalized Discount Cumulative Gain (NDCG@20)
  - Precision (P@20)
- Cross validation
  - 5-fold cross validation
  - Model selection on validation set with best MAP

与BM25比较

Model	Title						Description					
	MAP		P@20		NDCG@20		MAP		P@20		NDCG@20	
BM25	0.2533	-	0.3612	-	0.4158	-	0.2479	-	0.3514	-	0.4110	-
NPRF$_{ff}$-DRMM	0.2823†	11.46%	0.3941†	9.11%	0.4350†	4.62%	0.2766†	11.58%	0.3908†	11.21%	0.4421†	7.56%
NPRF$_{ff'}$-DRMM	0.2837†	12.00%	0.3928†	8.74%	0.4377†	5.27%	0.2774†	11.90%	0.3984†	13.38%	0.4493†	9.32%
NPRF$_{ds}$-DRMM	**0.2904**†	14.66%	**0.4064**†	12.52%	**0.4502**†	8.28%	**0.2801**†	12.95%	**0.4026**†	14.57%	**0.4559**†	10.92%
NPRF$_{ff}$-KNRM	0.2809†	10.90%	0.3851†	6.62%	0.4287	3.11%	0.2720†	9.71%	0.3867†	10.06%	0.4356†	5.99%
NPRF$_{ff'}$-KNRM	0.2815†	11.13%	0.3882†	7.48%	0.4264	2.55%	0.2737†	10.39%	0.3892†	10.74%	0.4382†	6.61%
NPRF$_{ds}$-KNRM	0.2846†	12.36%	0.3926†	8.69%	0.4327	4.06%	0.2800†	12.95%	0.3972†	13.03%	0.4477†	8.94%

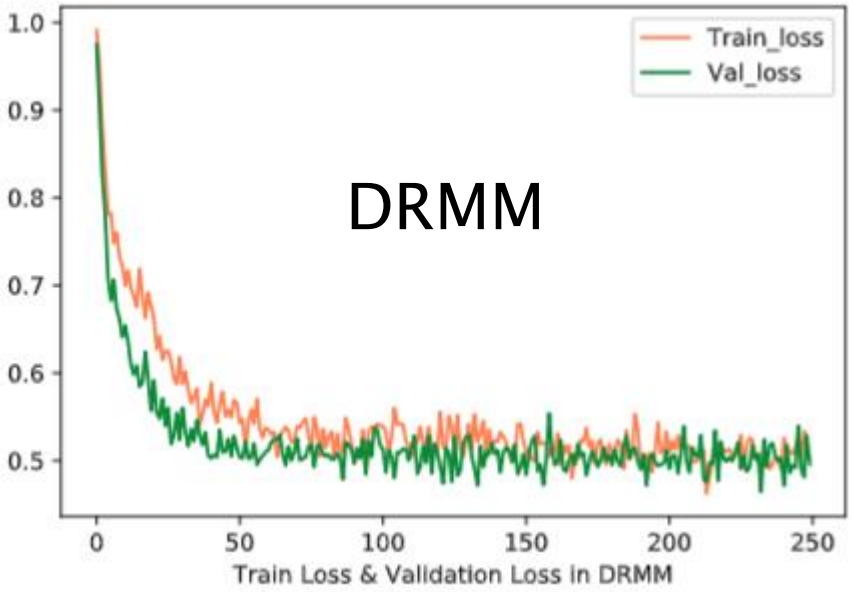
Table 2: Comparisons between NPRF and *BM25* on the *Robust04* dataset. Relative performances compared with *BM25* are in percentages. Significant improvements relative to the baselines are marked with †.

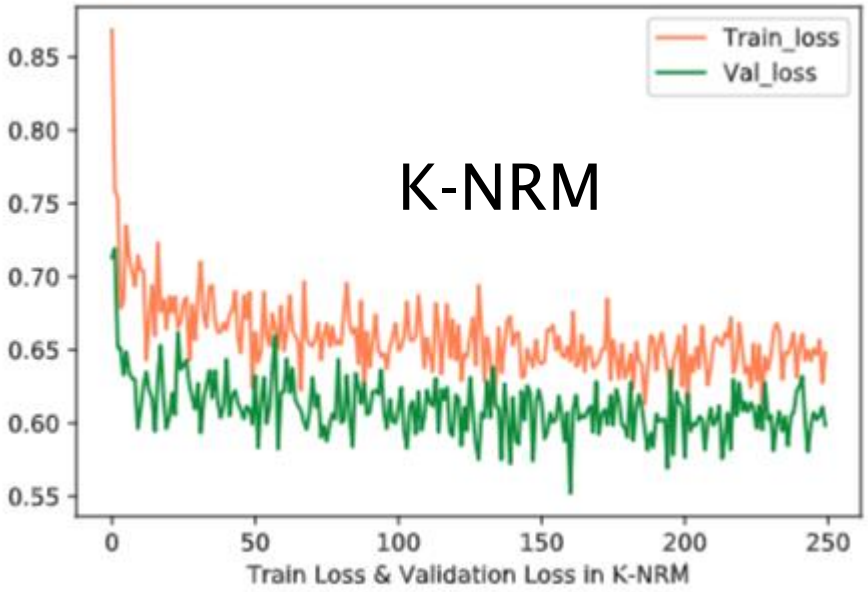
与Neural IR模型比较

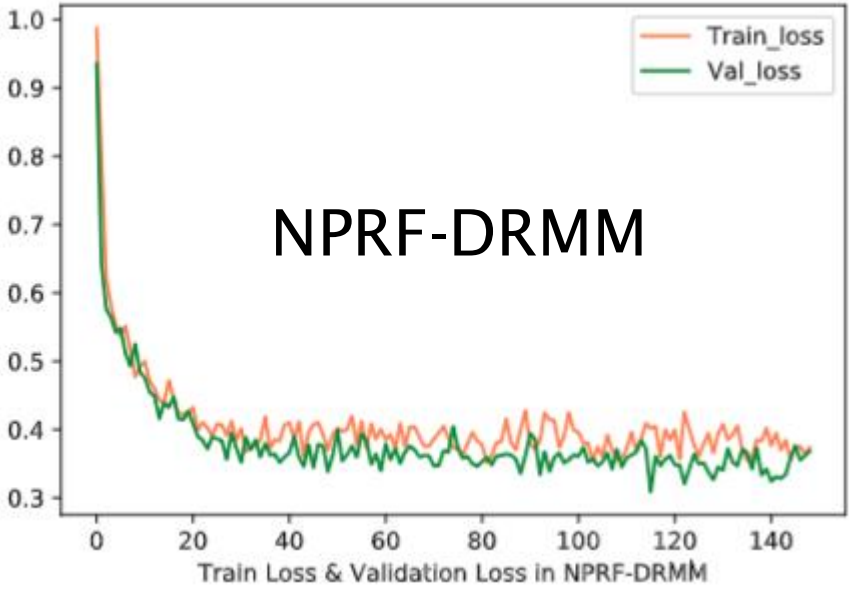
- DRMM: [Guo et al. CIKM16]
- K-NRM: [Xiong et al. SIGIR17]
- PACRR: [Hui et al. EMNLP17]

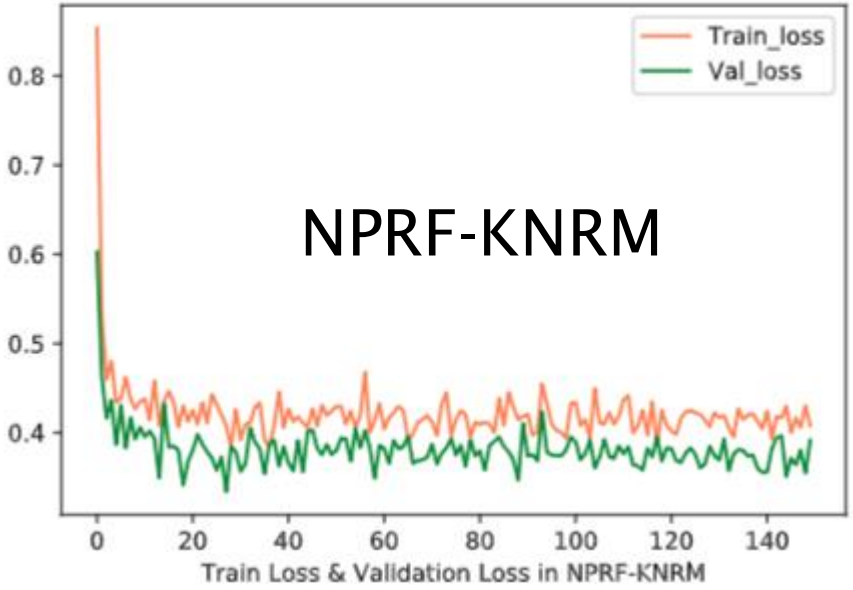
Model	Title						Description					
	MAP		P@20		NDCG@20		MAP		P@20		NDCG@20	
DRMM	0.2688	-	0.3713	-	0.4297	-	0.2630	-	0.3558	-	0.4135	-
K-NRM	0.2464	-	0.3510	-	0.3989	-	0.1687	-	0.2301	-	0.2641	-
PACRR-firstk	0.2540	-	0.3631	-	0.4082	-	0.2087	-	0.2962	-	0.3362	-
NPRF$_{ff}$-DRMM	0.2823	5.03%	0.3941†	6.14%	0.4350	1.24%	0.2766†	5.17%	0.3908†	9.84%	0.4421†	6.92%
NPRF$_{ff'}$-DRMM	0.2837†	5.55%	0.3928	5.78%	0.4377	1.87%	0.2774†	5.48%	0.3984†	11.97%	0.4493†	8.67%
NPRF$_{ds}$-DRMM	**0.2904**†	8.05%	**0.4064**†	9.46%	**0.4502**	4.78%	**0.2801**†	6.46%	**0.4026**†	13.15%	**0.4559**†	10.26%
NPRF$_{ff}$-KNRM	0.2809†	14.00%	0.3851†	9.72%	0.4287†	7.48%	0.2720†	61.22%	0.3867†	68.08%	0.4356†	64.96%
NPRF$_{ff'}$-KNRM	0.2815†	14.25%	0.3882†	10.60%	0.4264†	6.90%	0.2737†	62.21%	0.3892†	69.12%	0.4382†	65.93%
NPRF$_{ds}$-KNRM	0.2846†	15.50%	0.3926†	11.85%	0.4327†	8.47%	0.2800†	65.98%	0.3972†	72.62%	0.4477†	69.55%

Table 4: Comparisons between NPRF and neural IR models on *Robust04*. Relative performances of NPRF-DRMM(KNRM) compared with DRMM (K-NRM) are in percentages, and statistically significant improvements are marked with †.


DRMM
Train_loss
Val_loss
1.0
0.9
0.8
0.7
0.6
0.5
0
50
100
150
200
250
Train Loss & Validation Loss in DRMM


K-NRM
Train_loss
Val_loss
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0
50
100
150
200
250
Train Loss & Validation Loss in K-NRM


NPRF-DRMM
Train_loss
Val_loss
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
120
140
Train Loss & Validation Loss in NPRF-DRMM


NPRF-KNRM
Train_loss
Val_loss
0.8
0.7
0.6
0.5
0.4
0
20
40
60
80
100
120
140
Train Loss & Validation Loss in NPRF-KNRM

与传统PRF/QE方法比较

	TREC1-3						Robust04					
	Title			Description			Title			Description		
Model	MAP	P@20	NDCG@20	MAP	P@20	NDCG@20	MAP	P@20	NDCG@20	MAP	P@20	NDCG@20
BM25+QE	**0.2873**	0.5200	0.5330	**0.2601**	0.4973	0.5093	**0.2966**	0.3839	0.4353	**0.2926**	0.3817	0.4340
QL+RM3	0.2734	0.5093	0.5198	0.2421	0.4627	0.4801	0.2842	0.3878	0.4398	0.2686	0.3506	0.4150
DRMM (QE)	0.2741	0.5183	0.5345	0.2380	0.5077	0.5229	0.2876	0.4002	**0.4549**	0.2711	0.3822	0.4392
K-NRM (QE)	0.2633	0.5127	0.5235	0.2307	0.4877	0.5039	0.2521	0.3644	0.4062	0.2380	0.3304	0.3785
NPRF$_{ds}$-DRMM	0.2698	0.5187	0.5282	0.2527	**0.5283**	0.5444†	0.2904	**0.4064**	0.4502	0.2801	**0.4026**†	**0.4559**†
NPRF$_{ds}$-KNRM	0.2707	**0.5303**	**0.5406**	0.2505	0.5270	**0.5460**†	0.2846	0.3926	0.4327	0.2800	0.3972†	0.4477

Table 5: Comparisons between NPRF and *query expansion baselines* on *TREC1-3* and *Robust04*. Significant improvements over the best baseline is marked with †.

DRMM/K-NRM(QE): 传统查询扩展，输入DRMM/K-NRM评分

NPRF的优势主要体现在early precision(P@20, NDCG@20)

总结与展望

- 基于DNN的检索模型的研究虽然目前取得了一定的成果，但还有许多问题没有解决
  - 尚未得到明显优于传统模型（如BM25+QE）的结果
  - 很多论文回避了与传统PRF模型的比较
- CNN、统计直方图：有用；RNN：没有效果
- 长文本IR应用中往往DNN方法效果有限
- 但是在商品推荐、基于title的检索、microblog retrieval等短文本应用中效果不错
- 通过CNN等方法提取的特征 Vs 基于信息理论进行概率估计得到的特征
  - 是否有本质区别？
- 很多在NLP领域证明非常有效的方法，在IR领域尚未发挥威力

参考资料

[DSSM] Huang, Po-Sen, et al. Learning deep structured semantic models for web search using clickthrough data. CIKM 2013.

[DUET] Bhaskar Mitra, Fernando Diaz, Nick Craswell: Learning to Match using Local and Distributed Representations of Text for Web Search. WWW 2017: 1291-1299

[MatchPyramid] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Xueqi Cheng: A Study of MatchPyramid Models on Ad-hoc Retrieval. CoRR abs/1606.04648 (2016)

[DRMM] Jiafeng Guo, Yixing Fan, Qingyao Ai, W. Bruce Croft: A Deep Relevance Matching Model for Ad-hoc Retrieval. CIKM 2016: 55-64

[K-NRM] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, Russell Power:

End-to-End Neural Ad-hoc Ranking with Kernel Pooling. SIGIR 2017: 55-64

[PACRR] Hui K, Yates A, Berberich K, et al. PACRR: A Position-Aware Neural IR Model for Relevance Matching. EMNLP 2017: 1060-1069.

[NPRF]Canjia Li, Yingfei Sun, Ben He, et al. NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval. EMNLP 2018.

Xu, Jun, Xiangnan He, and Hang Li. Deep Learning for Matching in Search and Recommendation. SIGIR 2018.

Surveys

Jun Xu, Xiangnan He, and Hang Li. Deep Learning for Matching in Search and Recommendation. SIGIR 2018

Bhaskar Mitra, Nick Craswell: An Introduction to Neural Information Retrieval. Foundations and Trends in Information Retrieval 13(1): 1-126 (2018)

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, Xueqi Cheng: A Deep Look into Neural Ranking Models for Information Retrieval. CoRR abs/1903.06902 (2019)

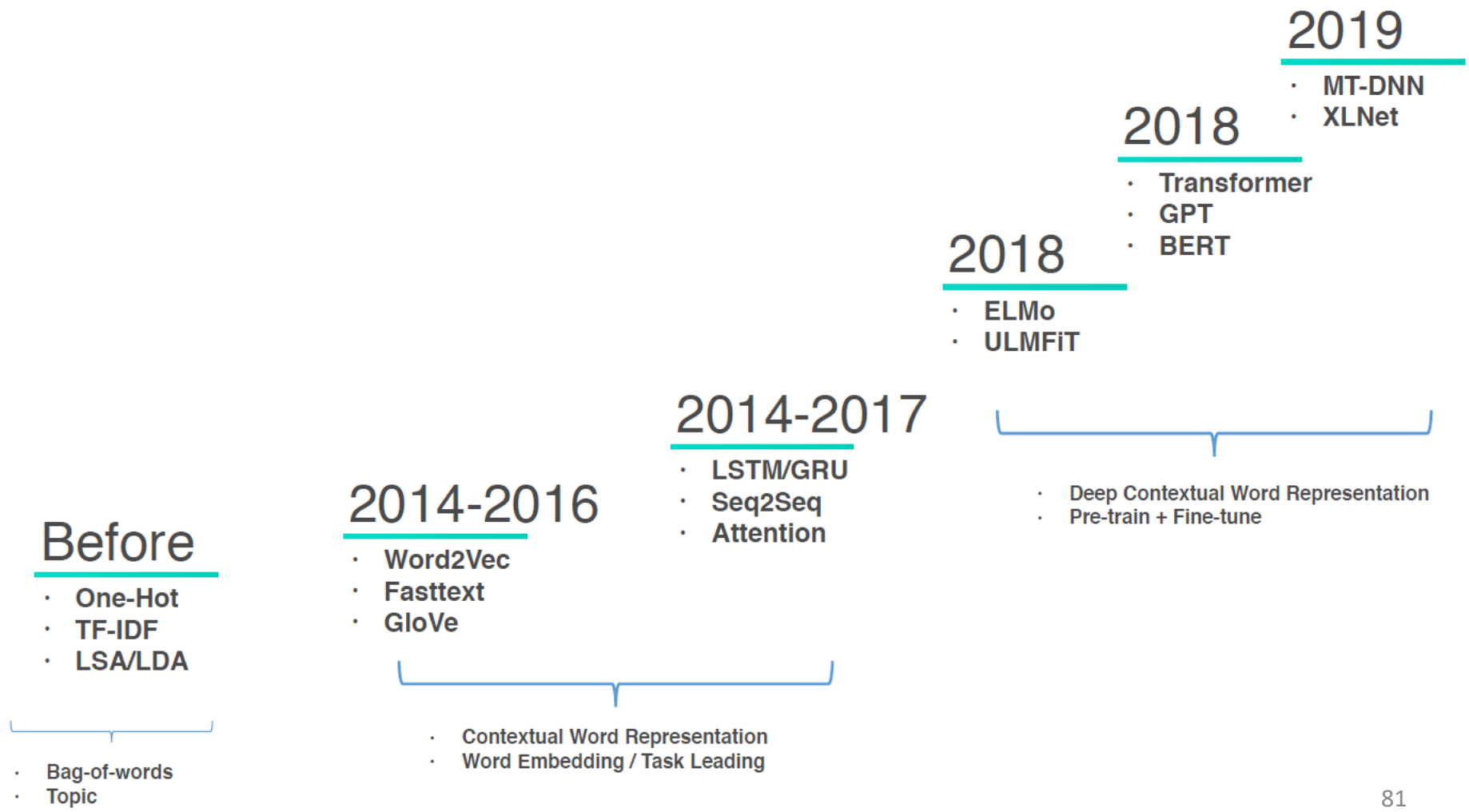
提纲

1. 上一讲回顾
2. 深度神经网络(DNN)基础
3. Neural IR Model
5. BERT

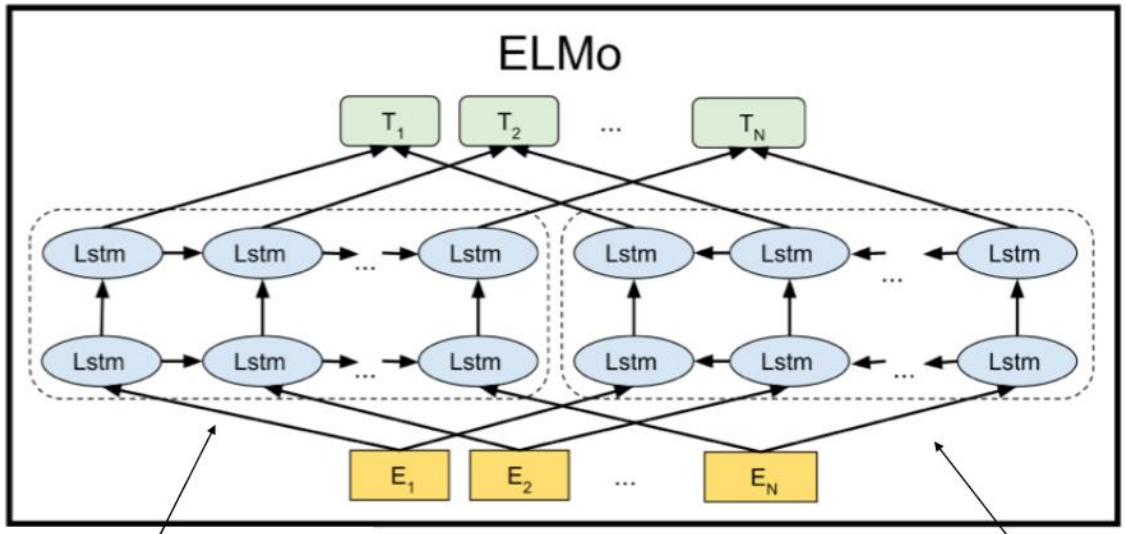
BERT简介

- 背景
  - Word2vec 学习到的是静态的embedding，而缺乏了对上下文建模的能力，同时缺乏语序的信息。
  - 以Bank为例，它表示的意思可以是“银行”，也可以是“河岸”。word2vec模型无法区分此类词汇在不同上下文场景下的区别
  - 之后出现了如ELMO这类的用LSTM双向语言模型，当获取一个词的embedding的时候，需要将整体的上下文作为输入，那么在不同的上下文的场景下同一个单词可以有不同的表达意思。
  - 最近，BERT是一种更强的基于上下文的文本特征抽取器…
- 语言模型的关键
  - 上下文信息提取能力
  - 保留单词位置信息能力
  - 更多的训练数据，更好的训练任务，…..

语言模型发展


2019
· MT-DNN
· XLNet
2018
· Transformer
· GPT
· BERT
2018
· ELMo
· ULMFiT
2014-2017
· LSTM/GRU
· Seq2Seq
· Attention
· Deep Contextual Word Representation
· Pre-train + Fine-tune
2014-2016
· Word2Vec
· Fasttext
· GloVe
· Contextual Word Representation
· Word Embedding / Task Leading
Before
· One-Hot
· TF-IDF
· LSA/LDA
· Bag-of-words
· Topic

ELMO



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

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

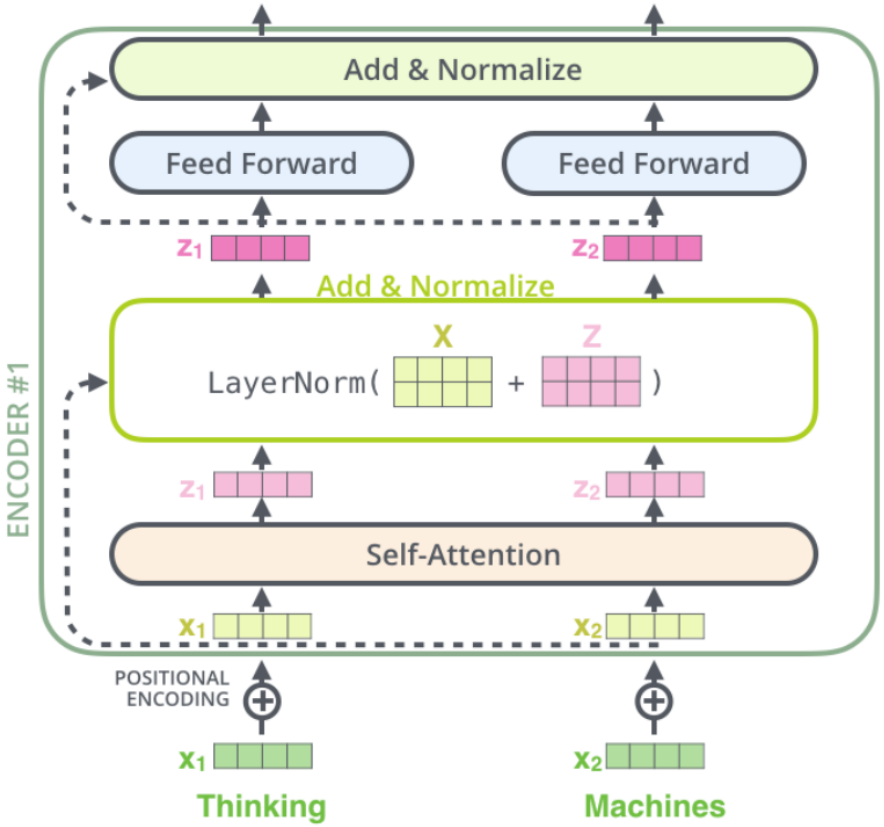
缺点：不够深，LSTM本身比较弱：无注意力机制，长序依赖消失

Transformer

自注意力机制：

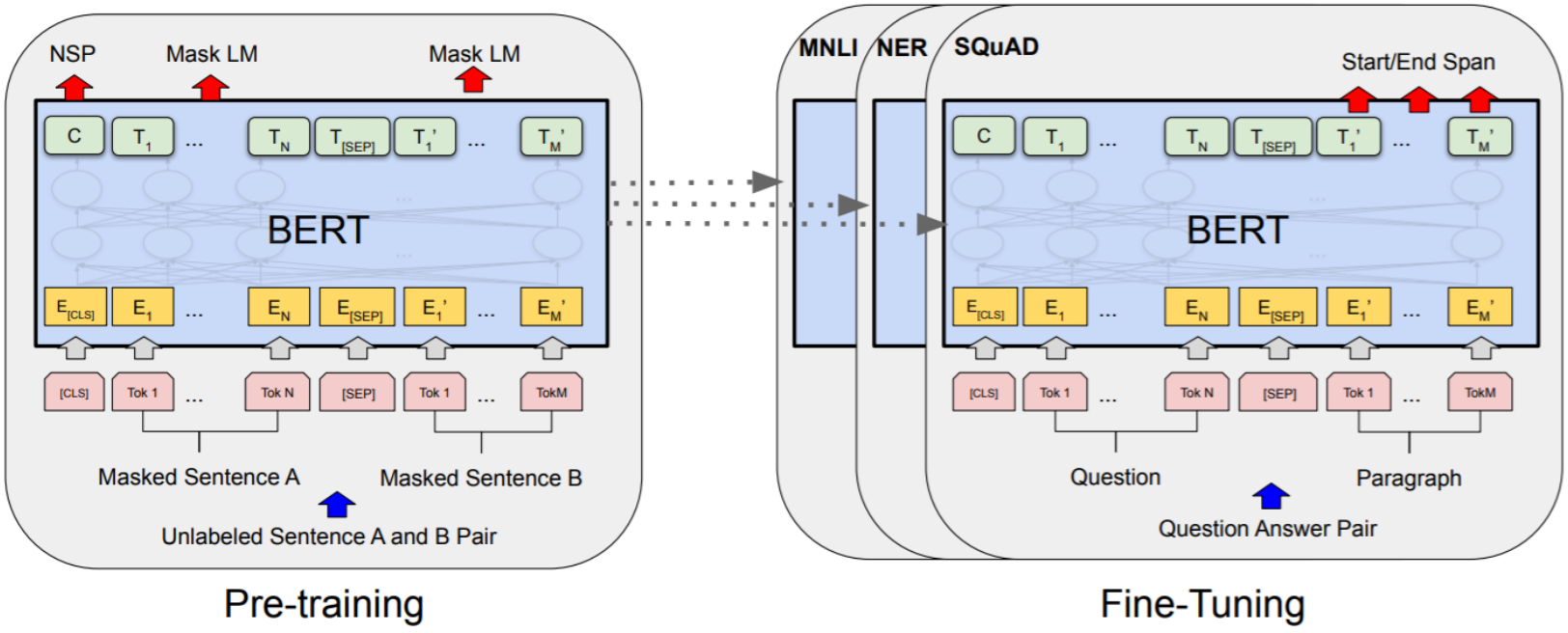
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

可总结为：
一层自注意力机制
一层前向神经网络
两层残差连接



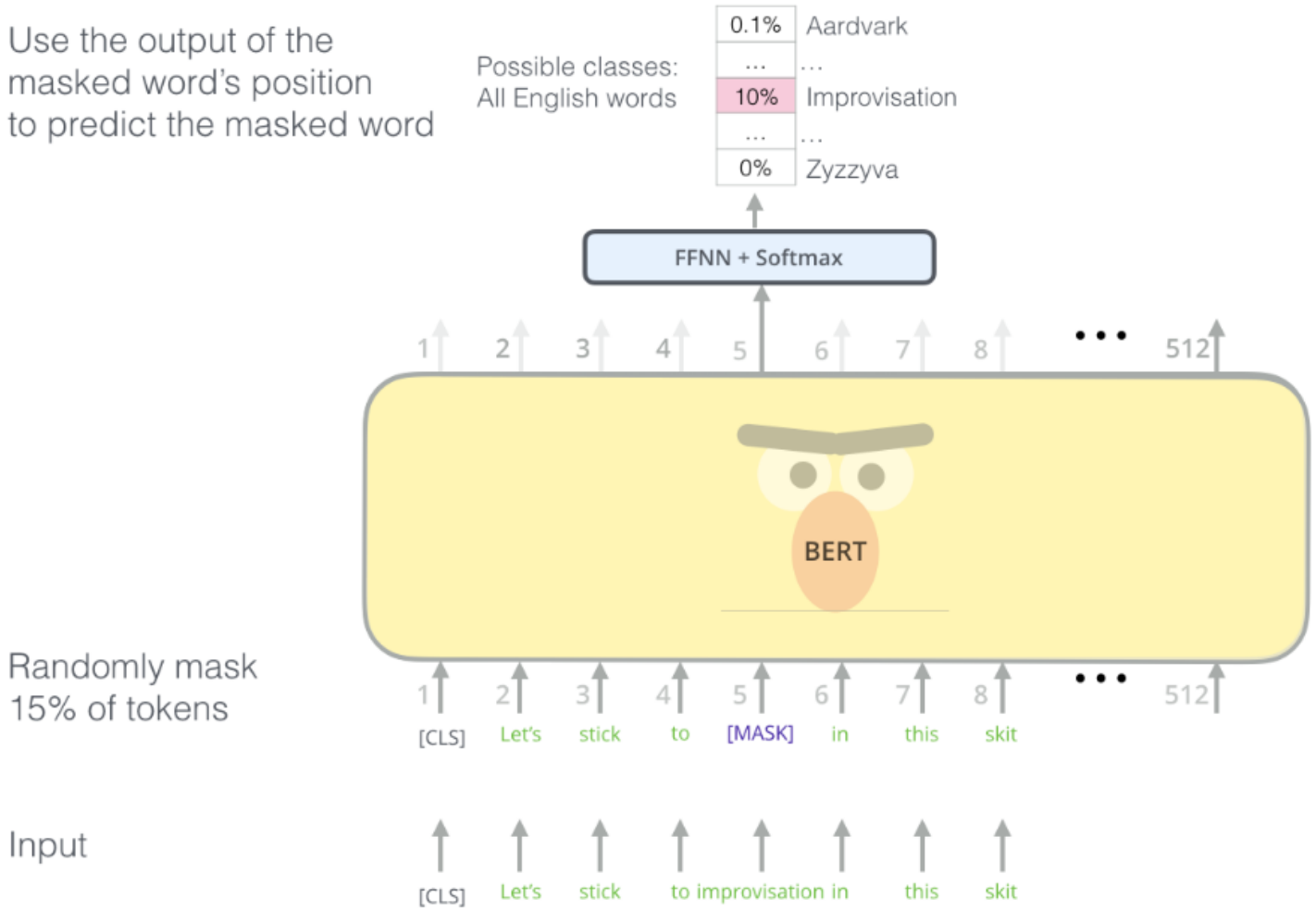
自注意力机制计算快，不存在RNN中的梯度消失问题；
特征提取能力强，每个token都注意到其它的token信息

BERT

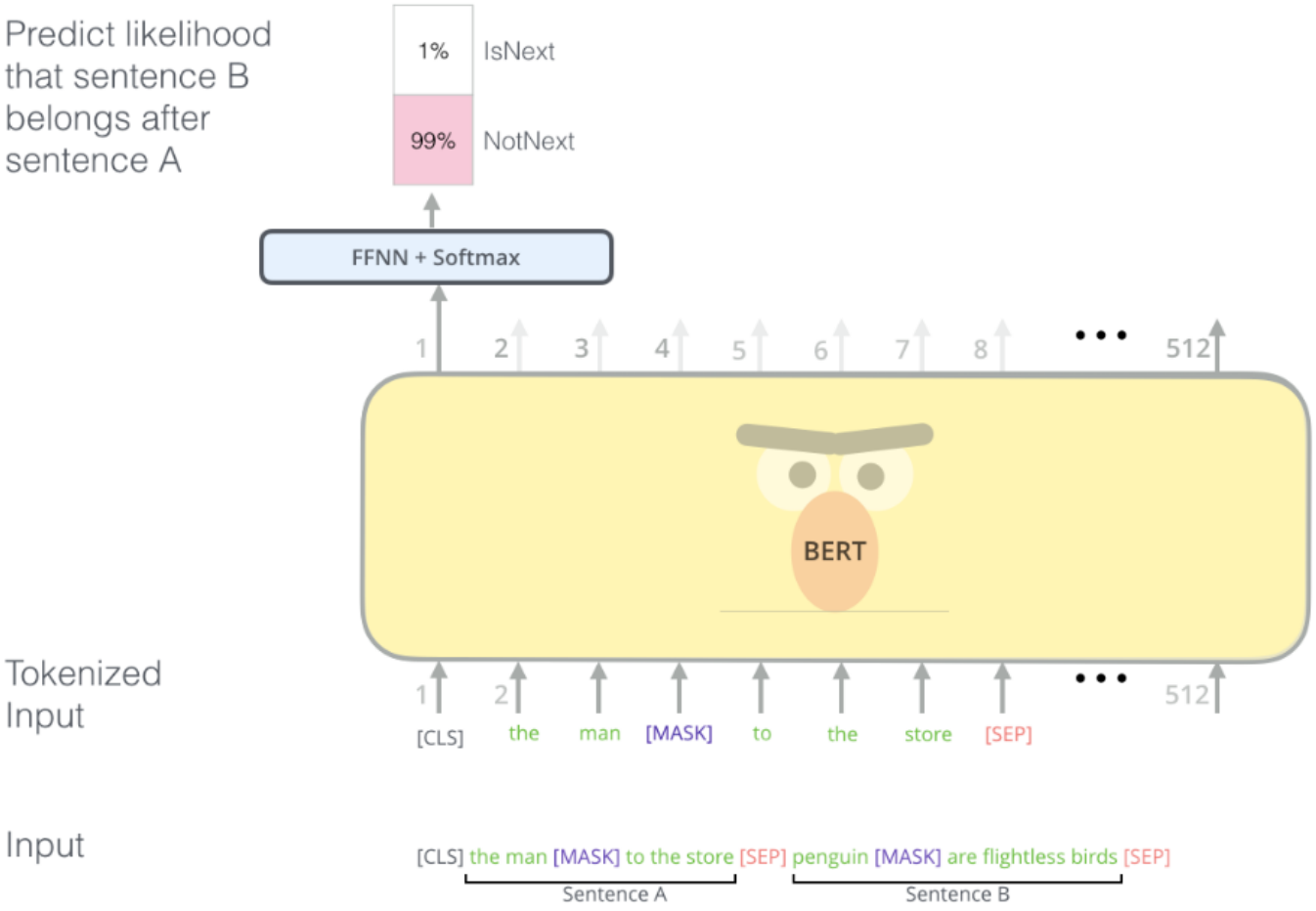


- 以Transformer作为底层特征提取器，堆叠Transformer
- 在BooksCorpus和Wikipedia大规模语料上面预训练
- 引导了NLP领域里面的pre-training-> Fine-tuning的范式

BERT-Mask Language Model



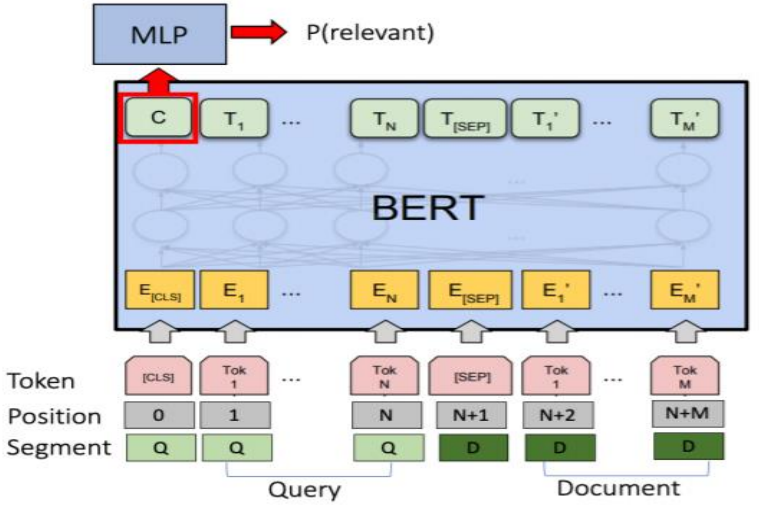
BERT-Next Sentence Prediction



基于BERT的检索模型

- 基于神经网络的检索模型最为人诟病的是模型通常都是跟弱的Baseline进行比较，始终无法超越历史最强的baseline, Bert的出现打破了这种僵局
- 如何做
  - “[cls] query [sep] document [sep]” 作为输入，最后一层的CLS做0/1分类
  - 使用0/1交叉熵损失

    对BERT进行finetune



基于BERT的检索模型

- 效果 [Nogueira et al. Arxiv]
  - MS MARCO (MRR Metric)

Method	DEV	TEST
BM25	16.7	16.5
KNRM	21.8	19.8
Conv-KNRM	29.0	27.1
IRNet	27.8	28.1
BERT Base	34.7	-
BERT Large	36.5	35.8

基于BERT的检索模型

- 剖析BERT的用法 [Qiao et al. Arxiv]
  - Bert (Rep): Representation-based ranker. It first obtains the representations of query and document using pre-trained Bert separately and then computes their cosine similarity as relevance score.
  - Bert (Last-Int): Interaction-based ranker. It feeds the text pair as input to Bert and uses the [CLS] embedding from the last layer for classification task.
  - Bert (Mult-Int): It applies [CLS] embedding of each layer and then sums up the individual learned scores.
  - Bert (Term-Trans): It builds up the similarity matrix as in neural IR models and then sums up the similarity scores.

基于BERT的检索模型

- 理解BERT [Padigela et al. SIGIR 2019]
  - H1: BM25 is more biased towards higher query term frequency compared to BERT.
  - H2: Bias towards higher query term frequency hurts the BM25 performance.
  - H3: BERT retrieves documents with more novel words.
  - H4: BERT's improvement over BM25 is higher for longer queries.

基于BERT的检索模型

- 长文本下的BERT [MacAvaney et al. SIGIR 2019]
  - Idea: Test Bert's embedding for neural ranking models
  - Type I: utilize Bert's embedding, not tune
  - Type II: utilize Bert's embedding, co-training
  - Type III: concatenate [CLS] embedding into neural ranking models' matching signals
    - For DRMM, concatenate with histogram features
    - For KNRM, concatenate with kernel features
    - For PACRR, concatenate with n-gram features produced by CNN

基于BERT的检索模型

- 长文本下的BERT [Dai et al. SIGIR 2019]
  - Idea: Split long documents into segments with overlap
  - BERT-FirstP: relevance score from the first passage
  - BERT-MaxP: max relevance score among the passages
  - BERT-SumP: sum up relevance scores from all passages

参考资料

- Rodrigo Nogueira, Kyunghyun Cho: Passage Re-ranking with BERT. CoRR abs/1901.04085 (2019)
- Yifan Qiao, Chenyan Xiong, Zheng-Hao Liu, Zhiyuan Liu: Understanding the Behaviors of BERT in Ranking. CoRR abs/1904.07531 (2019)
- Sean MacAvaney, Andrew Yates, Arman Cohan, Nazli Goharian: CEDR: Contextualized Embeddings for Document Ranking. CoRR abs/1904.07094 (2019)
- Harshith Padigela, Hamed Zamani, W. Bruce Croft: Investigating the Successes and Failures of BERT for Passage Re-Ranking. CoRR abs/1905.01758 (2019)
- Peng Xu, Xiaofei Ma, Ramesh Nallapati, Bing Xiang: Passage Ranking with Weak Supervsion. CoRR abs/1905.05910 (2019)
- Zhuyun Dai, Jamie Callan: Deeper Text Understanding for IR with Contextual Neural Language Modeling. CoRR abs/1905.09217 (2019)
- Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin：Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. EMNLP 2019