21.9.14快手面试×

一、自我介绍

二、研究方向：

1．主要做了哪些研究

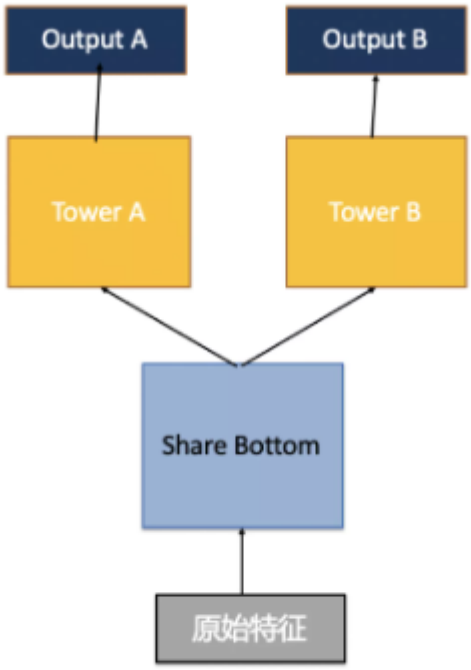
三、实习：

1．业务介绍

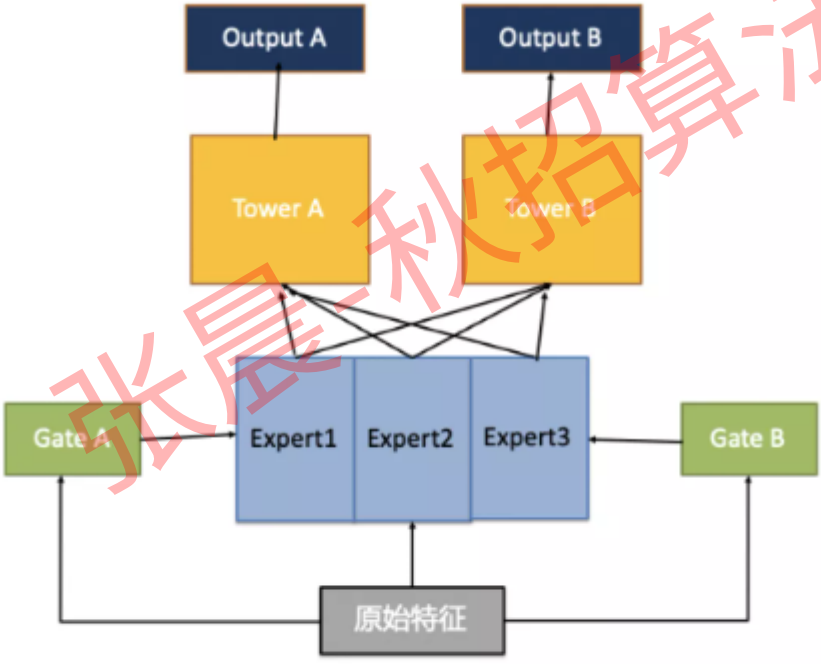
2．规则生成文案具体的生成方法

3．MMOE原理及网络结构

mmoe的思想就是，在底部的embedding层上设立多个expert网络，不同的task可以根据需求选择expert，通过expert输出的概率加权平均获得每个task的输入。每个task是单独的mlp tower。mmoe这样做可以解决什么问题呢？如果底部是完全共享的网络，对于相关性比较高的任务，效果会好一些。但是对于相关性比较差的任务，这种share-bottom的效果就不是很好。



share bottom结构



MMoE结构

4．为啥会想到用MMOE，有哪几个任务

常用的排序模型比如deepFM，DCN等均为单任务模型，即一个模型完成一个任务。如果需要完成多个任务的话，则可以针对每个任务单独训练一个模型。这样存在一个明显的问题就是对多个子任务建模很容易忽略任务之间的关联和约束等等，从而导致多个任务的整体效果无法达到最优。在多任务学习

中，往往会将多个相关的任务放在一起来学习。例如在推荐系统中，排序模型同时预估候选的点击率和浏览时间。相对于单任务学习，多任务学习有以下优势：

- 多个任务共享一个模型，占用内存量减少；
- 多个任务一次前向计算得出结果，推理速度增加；
- 关联任务通过共享信息，相互补充，可以提升彼此的表现。

四、技能：

1．介绍一下word2vec

word2vec，即词向量，就是一个词用一个向量来表示。是2013年Google提出的。word2vec工具主要包含两个模型：skip-gram和CBOW，以及两种高效训练的方法：负采样（negative sampling）和层序softmax（hierarchical softmax）。word2vec词向量可以较好地表达不同词之间的相似和类比关系。word2vec是一个NLP工具，它可以将所有的词向量化，这样词与词之间就可以定量的去度量他们之间的关系，挖掘词之间的联系。

2．word2vec的优化方法（https://zhuanlan.zhihu.com/p/86680049 <https://zhuanlan.zhihu.com/p/86680049> ）

在CBOW和skip-gram讲解完成后，我们会发现Word2Vec模型是一个超级大的神经网络（权重矩阵规模非常大）。

举个例子，我们拥有10000个单词的词汇表，我们如果想嵌入300维的词向量，那么我们的输入-隐层权重矩阵和隐层-输出层的权重矩阵都会有 10000 x 300 = 300万个权重，在如此庞大的神经网络中进行梯度下降是相当慢的。更糟糕的是，你需要大量的训练数据来调整这些权重并且避免过拟合。百万数量级的权重矩阵和亿万数量级的训练样本意味着训练这个模型将会是个灾难。

- 根据单词出现频率构建好的huffman树，沿着路径从根节点到对应的叶子节点，一层一层的利用sigmoid函数做二分类，判断向左还是向右走，规定沿着左子树走，那么就是负类(霍夫曼树编码1)，沿着右子树走，那么就是正类(霍夫曼树编码0)。一路上的概率连乘，最终得到某个单词的输出概率。经过分层softmax优化之后，权重更新的复杂度从O（V）降至O（logV），极大提高了效率。当然，如果中心词是一个很生僻的词，还是需要在霍夫曼树中向下走很久，这也是它的一个不可避免的缺点。

- 分层softmax在每次循环迭代过程中依然要处理大量节点上的更新运算，而负采样技术只需更新“输出向量”的一部分。负抽样的目的是为了最终输出的上下文单词（正样本）[基于训练样本的半监督学习]在采样过程中应该保留下来并更新，同时也需要采集部分负样本（非上下文单词）。通过负采样，在更新隐层到输出层的权重时，只需更负采样的单词，而不用更新词汇表所有单词，从而节省巨大计算量。

3．word2vec与glove、FastText的不同点

word2vec VS FastText: 都可以无监督学习词向量， fastText训练词向量时会考虑subword；fastText还可以进行有监督学习进行文本分类，其主要特点：

- 结构与CBOW类似，但学习目标是人工标注的分类结果；
- 采用hierarchical softmax对输出的分类标签建立哈夫曼树，样本中标签多的类别被分配短的搜寻路径；
- 引入N-gram，考虑词序特征；
- 引入subword来处理长词，处理未登陆词问题；

word2vec VS glove:

- word2vec是局部语料库训练的，其特征提取是基于滑窗的；而glove的滑窗是为了构建co-occurance matrix，是基于全局语料的，可见glove需要事先统计共现概率；因此，word2vec可以进行在线学习，glove则需要统计固定语料信息。
- word2vec是无监督学习，同样由于不需要人工标注；glove通常被认为是无监督学习，但实际上glove还是有label的，即共现次数。
- word2vec损失函数实质上是带权重的交叉熵，权重固定；glove的损失函数是最小平方损失函数，权重可以做映射变换。
- 总体来看，glove可以被看作是更换了目标函数和权重函数的全局word2vec。

4．word2vec两种模型的应用场景

CBOW比Skip-gram 训练速度快，但是应用场景基本一样，不过实践角度而言，skip-gram训练出来的词向量会稍好一点，因为模型的训练机制会对向量的预测能力要求更高（1 predict n）

5．对于生僻字word2vec的哪种模型更好，原因是什么

在skip-gram当中，每个词都要收到周围的词的影响，每个词在作为中心词的时候，都要进行K次的预测、调整。因此， 当数据量较少，或者词为生僻词出现次数较少时， 这种多次的调整会使得词向量相对的更加准确。因为尽管cbow从另外一个角度来说，某个词也是会受到多次周围词的影响（多次将其包含在内的窗口移动），进行词向量的跳帧，但是他的调整是跟周围的词一起调整的，grad的值会平均分到该词上， 相当于该生僻词没有收到专门的训练，它只是沾了周围词的光而已。在skip-gram里面，每个词在作为中心词的时候，实际上是 1个学生 VS K个老师，K个老师（周围词）都会对学生（中心词）进行“专业”的训练，这样学生（中心词）的“能力”（向量结果）相对就会扎实（准确）一些，但是这样肯定会使用更长的时间；cbow是 1个老师VS K个学生，K个学生（周围词）都会从老师（中心词）那里学习知识，但是老师（中心词）是一视同仁的，教给大家的一样的知识。至于你学到了多少，还要看下一轮（假如还在窗口内），或者以后的某一轮，你还有机会加入老师的课堂当中（再次出现作为周围词），跟着大家一起学习，然后进步一点。因此相对skip-gram，你的业务能力肯定没有人家强，但是对于整个训练营（训练过程）来说，这样肯定效率高，速度更快。

6．BERT原理

BERT模型的全称是Bidirectional Encoder Representations from Transformers，它是一种新型的语言模型。之所以说是一种新型的语言模型，是因为它通过联合调节所有层中的双向Transformer <https://www.cnblogs.com/huangyc/p/9813907.html> 来训练预训练深度双向表示。

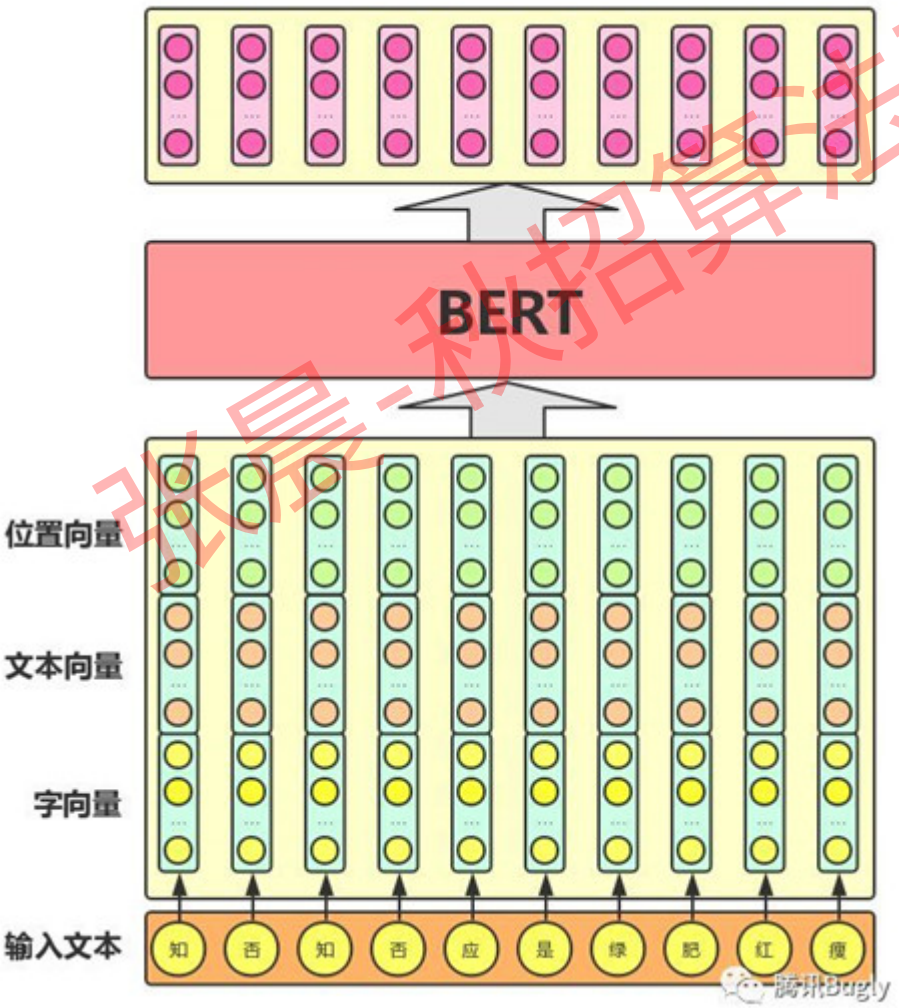
7．BERT任务是什么

- 为了训练深度双向Transformer表示，采用了一种简单的方法：随机掩盖部分输入词，然后对那些被掩盖的词进行预测，此方法被称为“Masked LM”(MLM)。预训练的目标是构建语言模型，BERT模型采用的是bidirectional Transformer。那么为什么采用“bidirectional”的方式呢？因为在预训练语言模型来处理下游任务时，我们需要的不仅仅是某个词左侧的语言信息，还需要右侧的语言信息。文章作者在一句话中随机选择15%的词汇用于预测。对于在原句中被抹去的词汇，80%情况下采用一个特殊符号[MASK]替换，10%情况下采用一个任意词替换，剩余10%情况下保持原词汇不变。这么做的主要原因是：在后续微

调任务中语句中并不会出现[MASK]标记，而且这么做的另一个好处是：预测一个词汇时，模型并不知道输入对应位置的词汇是否为正确的词汇（10%概率），这就迫使模型更多地依赖于上下文信息去预测词汇，并且赋予了模型一定的纠错能力。

- 很多句子级别的任务如自动问答（QA）和自然语言推理（NLI）都需要理解两个句子之间的关系，譬如上述Masked LM任务中，经过第一步的处理，15%的词汇被遮盖。那么在这一任务中我们需要随机将数据划分为等大小的两部分，一部分数据中的两个语句对是上下文连续的，另一部分数据中的两个语句对是上下文不连续的。然后让Transformer模型来识别这些语句对中，哪些语句对是连续的，哪些对子不连续。

8．BERT的输入输出是什么



BERT模型通过查询字向量表将文本中的每个字转换为一维向量，作为模型输入；模型输出则是输入各字对应的融合全文语义信息后的向量表示。此外，模型输入除了字向量，还包含另外两个部分：

1. 文本向量：该向量的取值在模型训练过程中自动学习，用于刻画文本的全局语义信息，并与单字/词的语义信息相融合

2. 位置向量：由于出现在文本不同位置的字/词所携带的语义信息有差异（比如：“我爱你”和“你爱我”），因此，BERT模型对不同位置的字/词分别附加一个不同的向量以作区分

最后，BERT模型将字向量、文本向量和位置向量的加和作为模型输入。特别地，在目前的BERT模型中，文章作者还将英文词汇作进一步切割，划分为更细粒度的语义单位（WordPiece），例如：将playing分割为play和##ing；此外，对于中文，目前作者尚未对输入文本进行分词，而是直接将单字作为构成文本的基本单位。

9．BERT损失函数是什么

在第一部分的损失函数中，如果被 mask 的词集合为 M，因为它是一个词典大小 |V| 上的多分类问题，所用的损失函数叫做负对数似然函数（且是最小化，等价于最大化对数似然函数），那么具体说来有：

$$L_1(\theta,\theta_1) = -\sum_{i=1}^{M} \log p(m = m_i|\theta,\theta_1), m_i \in [1,2,\ldots,|V|]$$

关于NLP那些你不知道的事

在第二部分的损失函数中，在句子预测任务中，也是一个分类问题的损失函数：

$$L_2(\theta,\theta_2) = -\sum_{j=1}^{N} \log p(n = n_i|\theta,\theta_2), n_i \in [\mathrm{IsNext}, \mathrm{NotNext}]$$

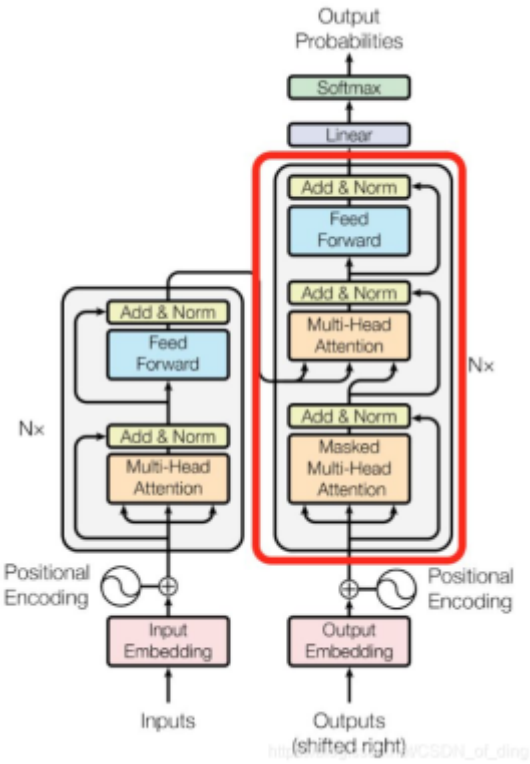
关于NLP那些你不知道的事

两个任务联合学习的损失函数是：

$$L(\theta,\theta_1,\theta_2) = -\sum_{i=1}^{M} \log p(m = m_i|\theta,\theta_1) - \sum_{j=1}^{N} \log p(n = n_i|\theta,\theta_2)$$

关于NLP那些你不知道的事

10．transformer的encoder部分介绍一下



注意力机制，其实可以理解为就是在计算相关性，很自然的想法就是去更多地关注那些相关更大的东西。这里首先要引入Query，Key和Value的概念，Query就是查询的意思，Key就是键用来和你要查询的Query做比较，比较得到一个分数（相关性或者相似度）再乘以Value这个值得到最终的结果。那么这个Q，K，V从哪里来呢，这里采用的是self-attention的方式，也就是**从输入自己** $X_{embedding}$ **来产生**，即做**线性映射**产生Q，K，V：

$$
\begin{aligned}
Q &= X_{embedding} * W_Q \\
K &= X_{embedding} * W_K \\
V &= X_{embedding} * W_V
\end{aligned}
$$

接下来说说注意力机制的计算，假设Q，K，V为切分完后的矩阵（其中一个头），根据两个向量的点积越大越相似，我们通过 QK^T 求出注意力矩阵，再根据注意力矩阵来给Value进行加权，即

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

其中 $\sqrt{d_k}$ 是为了把注意力矩阵变成标准正态分布，softmax进行归一化，使每个字与其他字的注意力权重之和为1。

层归一化与批归一化(batch normalization)的区别。两者思路相近，只是归一化的对象不同。批归一化会对一个小批的输入在相同神经元上的值进行归一化，即对一批数据中每一个特征进行归一化。层归一化是对同一输入在该层不同神经元上的值进行归一化，与批没有关系，即对于每个样本做归一化。

11．自注意力机制的相似度计算、softmax及加权求和的时间复杂度分别是多少

自注意力包括三个步骤：相似度计算、softmax和加权求和。相似度计算的时间复杂度是O(n^2d)，因为可以看成(n,d)和(d,n)两个矩阵的相乘。softmax的时间复杂度是O（n^2）。加权求和的时间复杂度是O(n^2d)，同样可以看成(n,n)和(n,d)的矩阵相乘。

五、算法：

1．最小的k个数（排序及优化）