深度剖析知识增强语义表示模型——ERNIE

原创 张贵发 AINLP 2019-12-14

收录于话题
#AINLP@我爱自然语言处理

98个

作者：张贵发
研究方向：自然语言处理

文章目录

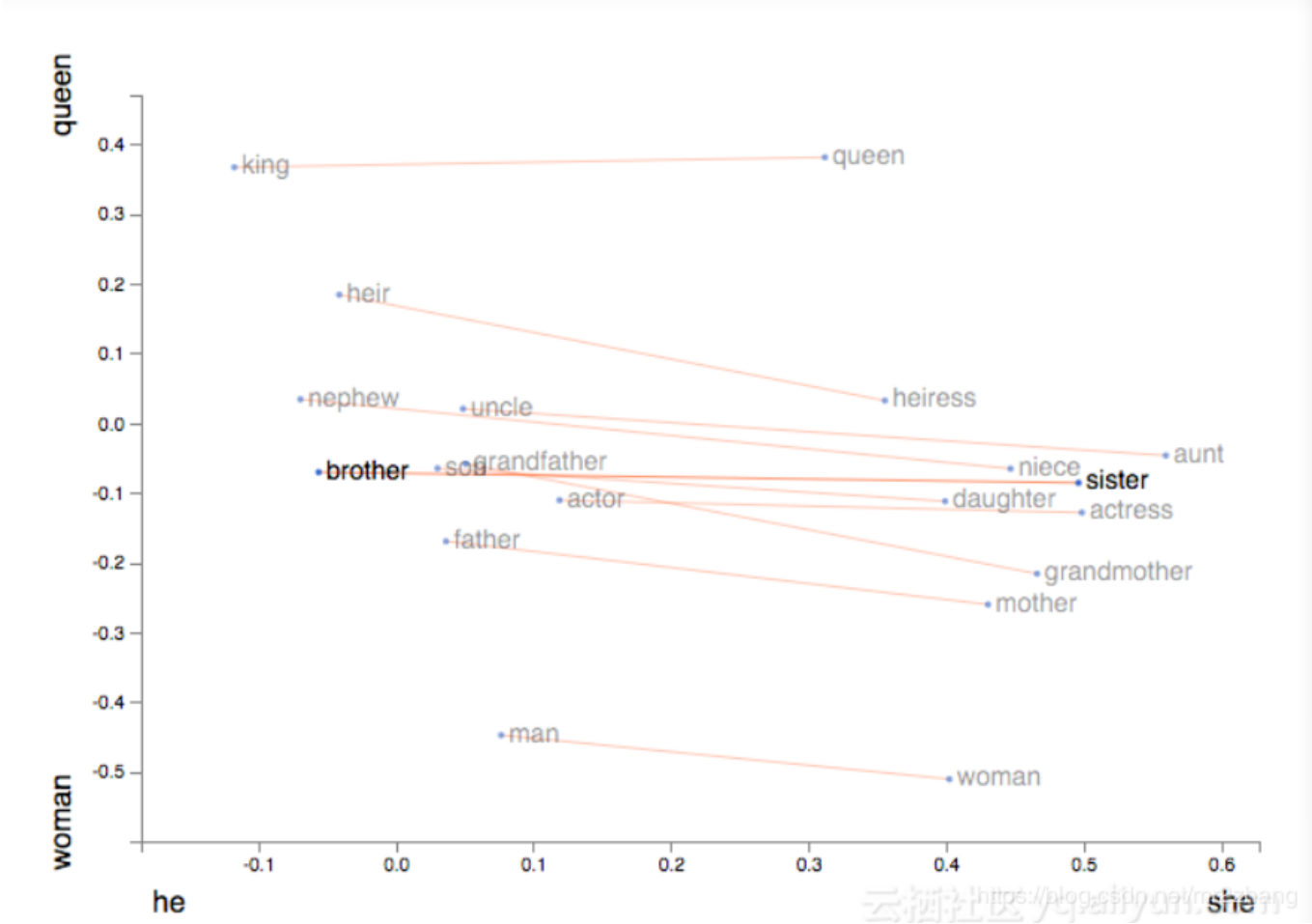
无监督文本的深度神经网络的出现，nlp领域又火了起来，深度神经网络大大提升了nlp任务的效果。虽然早期的网络也是基于上下文进行的向量建模，但是由于单向信息流的弊端，效果上始终难以大幅度提升。Transformer中的多层self-attention的出现，推进了深度网络的发展。Google提出的BERT模型，通过掩盖的term，利用多层的self-attention的双向建模能力，横扫了NLP比赛的各大排行榜。

前文介绍了bert,想详细了解Bert请参见：一步步理解bert

什么是语义表示

ERNIE是一个语言理解模型，最大的优点就是特别好的理解人类的语言。文字其实是概念背后的符号，更重要的其实是概念的本身。词语是具有语义的，怎么正确表示语义呢？语义的特点是什么？语义上比较近的词语真正的距离也是比较接近的，怎么对这部分进行表达，就是词向量，词向量每个词背后对应的是一个高维的向量，所以他们之间的距离是可以度量的。

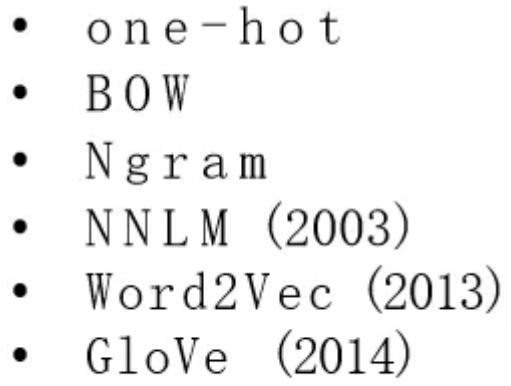
静态词向量



如图中所示:将文本信息映射到数字空间，变成数字表示的向量，在这种表示上，保留了词语间的距离信息。在句子的空间结构上我们期望获取更底层的之间的关系比如：

- $V_{King} - V_{Queen} = V_{Man} - V_{Women}$
- $V_{Paris} - V_{France} = V_{Berlin} - V_{German}$

- king和queen之间的关系相比与man与woman的关系大体应该相同的，那么他们通过矩阵运算，维持住这种关系
- Paris 和France之间的关系相比与Berlin与German的关系大体应该相同的，那么他们通过矩阵运算，维持住这种关系

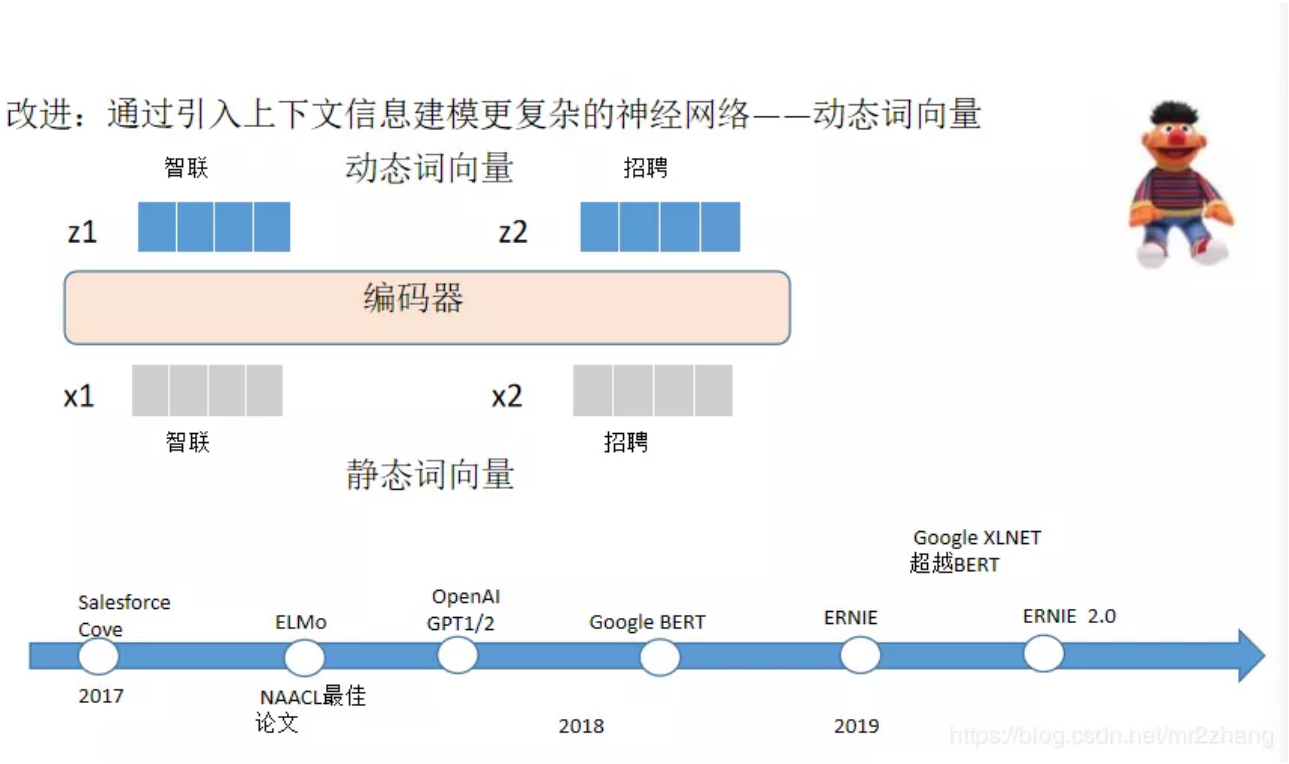


这种向量表示不仅仅可以进行距离的计算，还可以进行推断，怎么把这种词向量给学好，出现了各种技术，最好的就是 深度学习的神经网络 ， NLM、skip-gram、cbow、这些都是学习词向量比较有名的网络，为静态词向量，但是很难建模人类语言的复杂性，（一词多义、多词一意）带有上下文，语义根据上下文会变化的。如：
词向量的问题：无法解决歧义问题

- He deposited his money in this bank .
- His soldiers were arrayed along the river bank .

word embeding 有个问题就是我们的词通常有很多语义的，比如bank是银行还是河岸，具体的意思要取决与上下文，如果我们强行用一个向量来表示语义的话，只能把这两种语义都编码在这个向量里，但实际一个句子中，一个词只有一个语义，那么这种编码是有问题的。为了解决这种问题，出现了新的技术，动态词向量，或者上下文词向量，这个词向量就建模了上下文。

动态词向量（上下文词向量）



在这里插入图片描述

如图中展示，在进行映射的同时，编码器还建模了上下文，保证词向量的上下文信息。这样的词向量中不仅保留了词信息，还保留了上下文信息，基于上下文可以很好的预测。同时深度网络模型也不断的发展，不断取得大的进步。2017年Cove出现。2018年ELMo获得了NAACL最佳论文，后来的BERT、ERNIE、XLNET等。
在这些网络的发展中，最终要的是注意力机制的应用，都是不断的改进注意力机制，改变训练任务。

- RNN/LSTM/GRU
- Seq2Seq
- Attention
  - Luong Attention
  - Bahdanau Attention

随着序列任务的发展，到Seq2Seq，接下来出现注意力机制，主要流行的注意力机制有两种，是Luong Attention 和Bahdanau Attention。具体的注意力机制怎么起作用的呢？可以参考https://zhuanlan.zhihu.com/p/40920384

(1) $h_t = RNN_{enc}(x_t, h_{t-1})$，Encoder方面接受的是每一个单词word embedding，和上一个时间点的hidden state。输出的是这个时间点的hidden state。

(2) $s_t = RNN_{dec}(y_{t-1}, s_{t-1})$， Decoder方面接受的是目标句子单词的word embedding，和上一个时间点的hidden state。

(3) $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$，context vector是一个对 J encoder输出的hidden states的一个加权平均。

(4) $\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})}$，每一个encoder的hidden states对应的权重。

(5) $e_{ij} = score(s_i, h_j)$，通过decoder的hidden states加上encoder的hidden states来计算一个分数，用于计算权重(4)

(6) $\tilde{s}_t = tanh(W_c[c_t; s_t])$，将context vector 和 decoder的hidden states 串起来。

(7) $p(y_t|y_{<t}, x) = softmax(W_s \tilde{s}_t)$，计算最后的输出概率。

如图中是Luong Attention 和Bahdanau Attention两种注意力机制的主要区别点，Luong Attention是计算当前时间节点也就是i时刻，而Bahdanau Attention则是计算上一时间点，也就是i-1时刻。实际应用中效果区别不大，只是Luong Attention 更容易理解。

位置编码

后来的出现了掩盖词语的训练任务，出现了位置的编码信息，对于针对位置的编码，主要有以下几种：

- 用正弦位置编码（Sinusoidal Position Encoding）
- 学习位置向量（类似词向量）
- 相对位置表达（Relative Position Representations）

正玄位置编码：

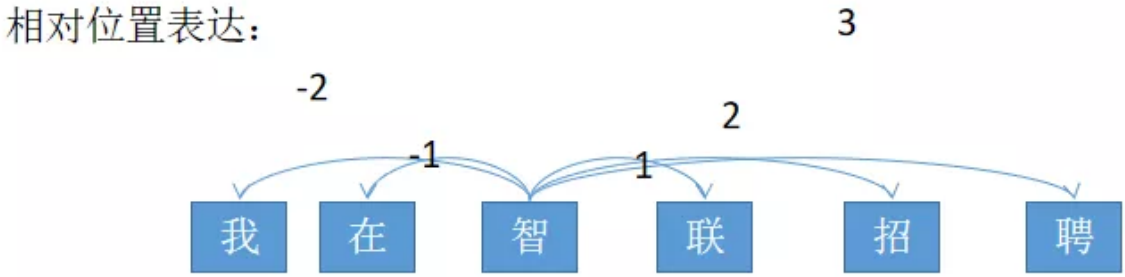
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

如：我在智联招聘。智的映射为：

$$[\sin(3/10000^{0/128}), \cos(3/10000^{1/128}), \sin(3/10000^{2/128}), \cos(3/10000^{3/128}), \ldots]$$

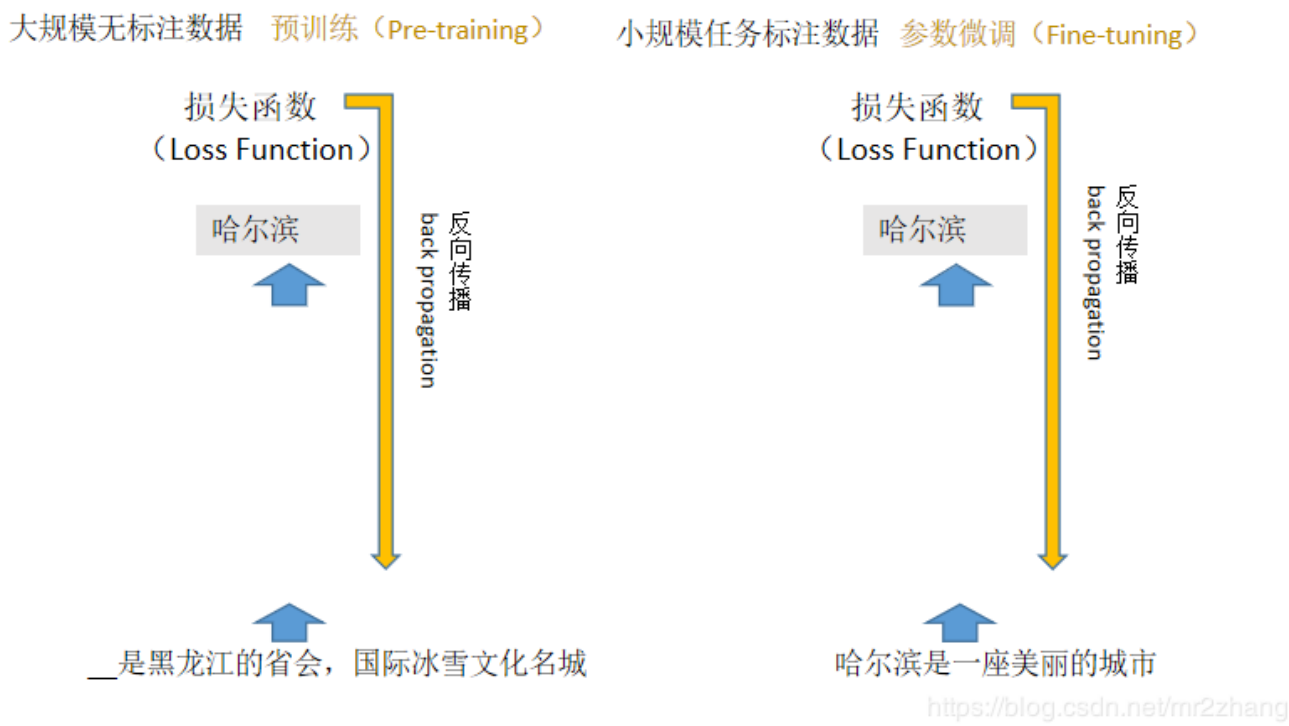
学习位置向量：把智所在的位置学习成为一个向量。



transformer-xl、清华与华为的ERNIE:绝对位置编码---》》相对位置编码

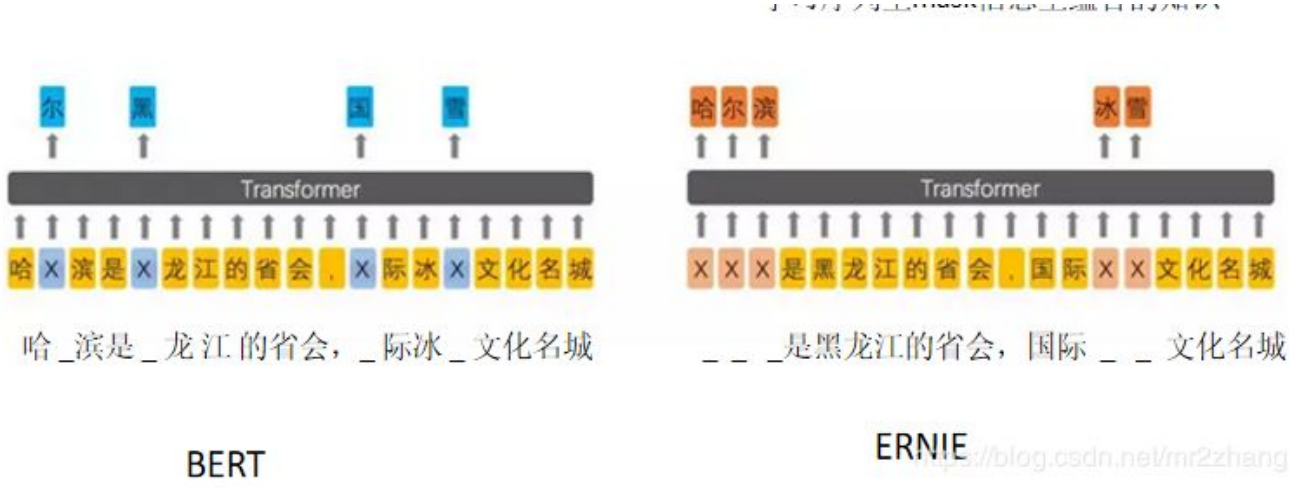
相对位置的表达，则是针对当前的位置信息的相对位置，如当前智为0，左右两边term距离当前距离的数值，正负号表示方向。绝对位置可以保留更多的信息，但是如bert中长度是512，如果一个句子是120，那么后面的位置信息全是0，过多的无用位置编码无意义。transformer-xl对位置编码改进为相对位置，引入了segment，不采用整个句子的，而是采用左右两端固定segment个数的term作为上下文。

上述的各个模型都特别好，怎么应用呢，进入了自然语言深度学习特别经典的预训练和下游任务的微调的机制，我们可以通过大量的无监督语料去学习一些简单的深度学习任务，就能得到一个很好的模型，比如bert 、ernie等等，这时候神经网络结构已经有了，它的神经网络的参数权重也有了，我们对下游的任务，比如说文本情感分析，哈尔滨是一个美丽的城市。判断它是一个积极的情绪还是消极的情绪，小规模的标注数据，把网络结构、参数照搬，热启动，在做一个反向传播的梯度更新，在这个任务里就能得到很好很好的效果，预训练和微调的机制。如下图：



ERNIE的原理介绍

上面简单回顾了静态词向量与动态词向量，以及其中重要的点，那么ERNIE是怎么工作，以及任务是怎么训练的呢？其实，ERNIE实在BERT的基础上进行的改进，那么将对比着BERT进行一步步剖析。



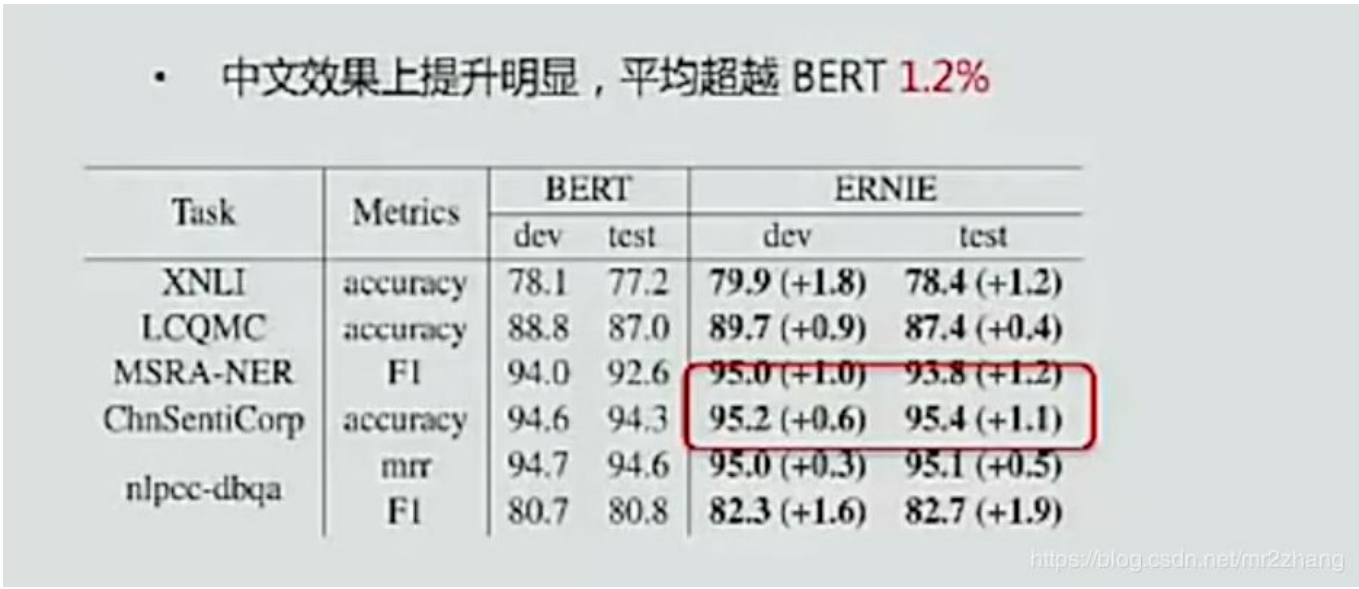
跟bert的缺陷进行对比，BERT训练任务是随机掩盖住15%的词进行预测，中文就是字，英文就是词，盖住之后去预测盖住的部分是什么样子的，训练任务叫做隐码语言模型，这个使它得到很好的效果，这个任务也存在缺陷，强行掩盖把词与词，字与字之间的关系给拆散了，比如说哈尔

滨，再去训练的时候把尔给盖住了，ERNIE的隐码语言模型盖住的不是一个字而是词或短语、命名实体，这样去预测整体。两者的主要区别如下：

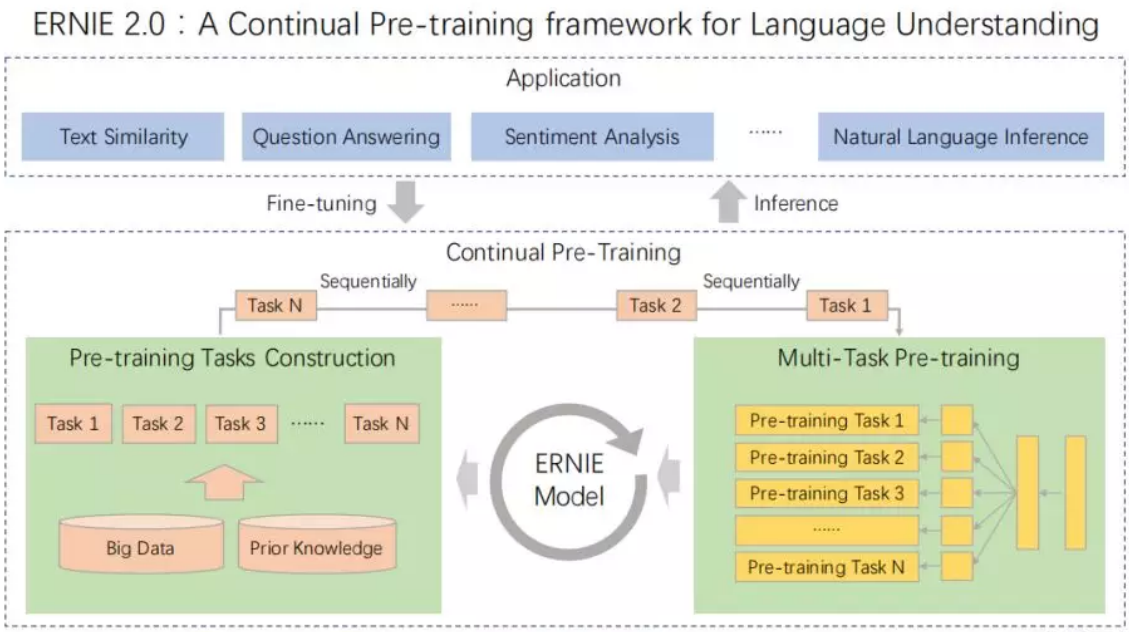
- BERT mask(sub-word) lm任务存在的问题
  - Word哈尔滨：sub-word 哈##尔##滨
  - Sub-word :预测可以通过word的 局部信息完成
  - 模型缺乏全局建模信息的能力
- 针对BERT mask sub-word 任务存在的问题，百度提出基于知识增强的语义理解模型
  - ERNIE mask word & entity
  - 强迫模型通过全局信息去预测mask掉的内容，学习序列里mask信息里蕴含的知识

ERNIE学到全局的信息，使它能学习到非常先验的结果，已经进行了大量的自然语言处理的分词模型，短语拼接的模型，命名实体识别的模型，能够提前把这些词或者短语给标注出来，标注出来之后再去学，这样看它是基于已有的策略再去进行海量的数据上训练出一个很好的模型。

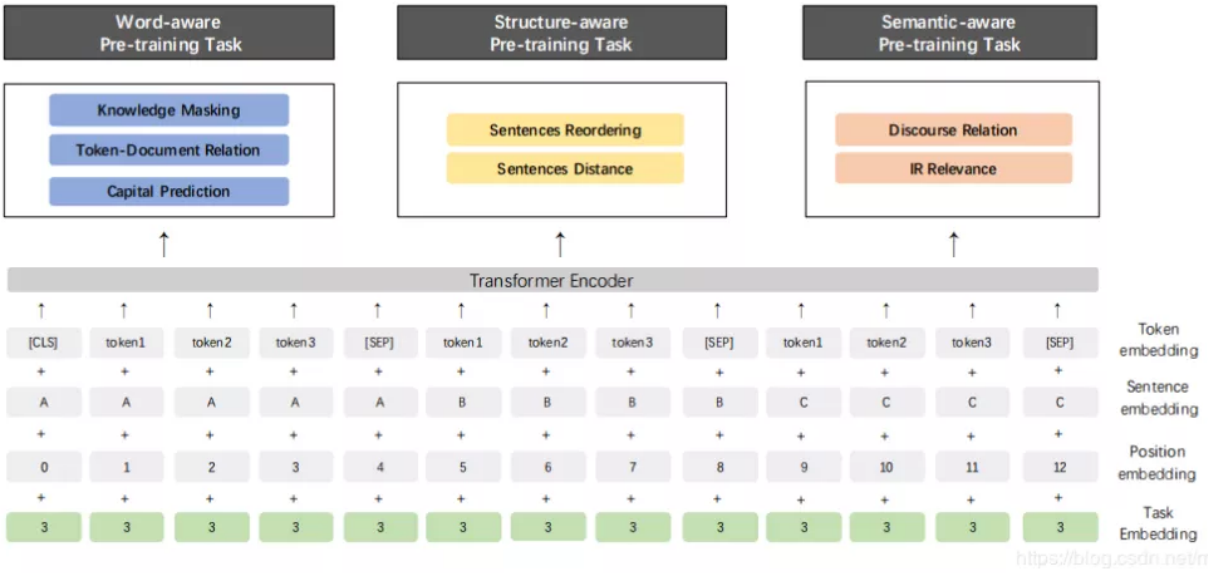
ERNIE 1.0的效果图：



针对bert的变种不断更新迭代，以及XLNET的产生，都不再局限于某个词，也是可以掩盖住短语、实体等进行任务训练，ERNIE也在不断的更新迭代，出现了ERNIE 2.0。



2.0加入更多的任务，经过左边各种各样的自然语言处理的任务去设计一些自然语言处理的辅助任务，基于这些辅助任务让ernie不断的去学习，持续的学习各种各样的多任务，效果不断的提升，然后再去处理下游的任务，比如文本相似度、只能问答，文本匹配效果得到非常明显的提升。



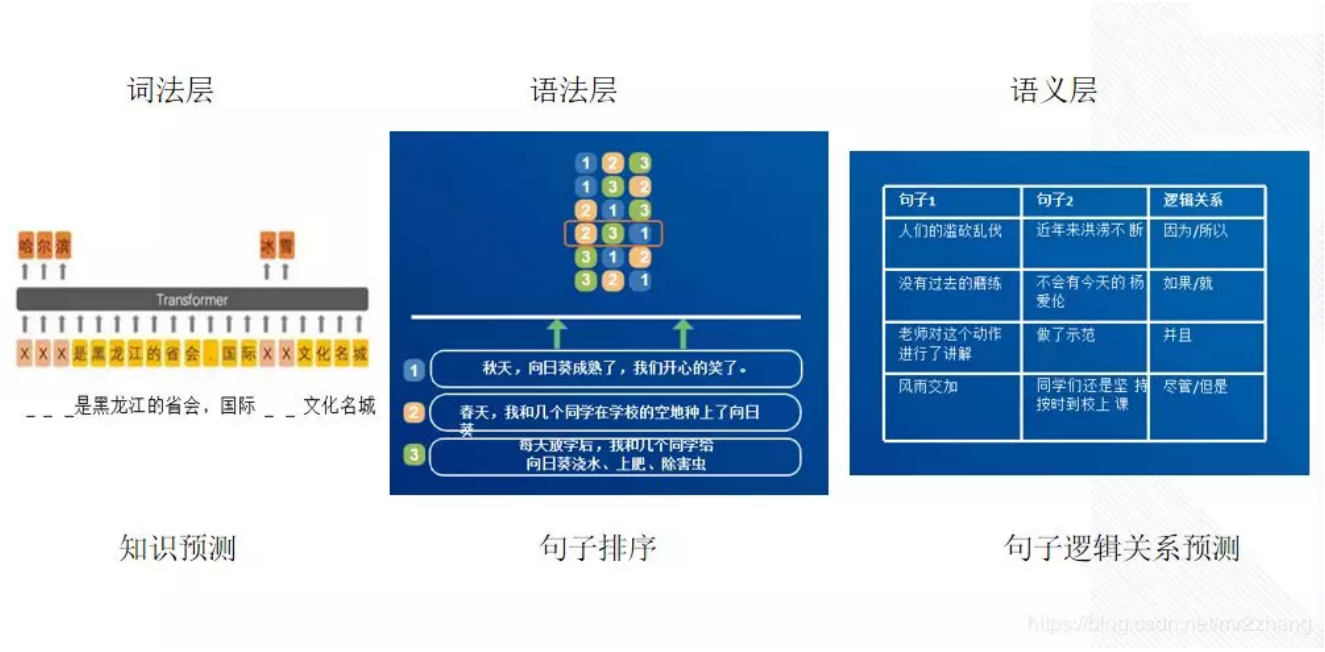
神经网络上的改造

上述所说各种各样的任务，怎么让ernie区分，特别加入了最下面这行的taskid，叫做taskid embedding,对于上面的输入是典型的ernie 1.0的输入，每个词的embedding,每个词所在句子的embedding，以及词在语料中位置的embedding,第四个呢就加上了任务的id embedding.让每个任务有单独的识别。

辅助任务

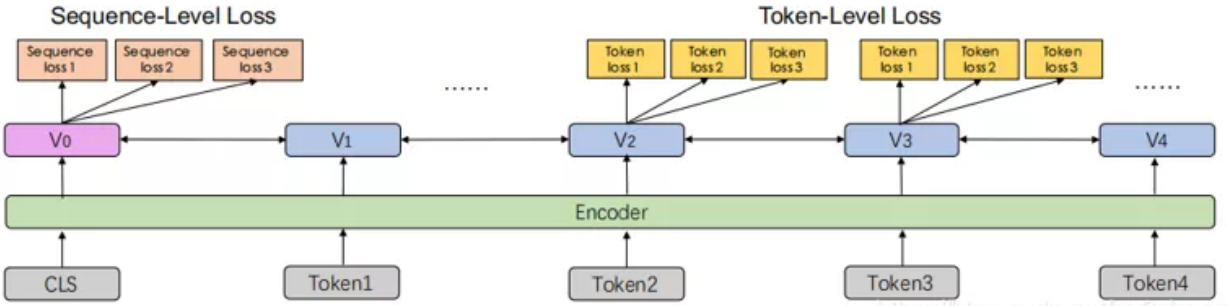
总共整理了7个任务，7个任务分位3类，一类是词法层的任务，一类是语法类的任务，一类是语义类的任务，词法层的任务 1.0的掩盖短语、实体。典型的词法任务就是句子排序，我们把三句话打乱顺序的给摆正，3句话的所有组合的阶乘，遍历组合都遍历一遍，让ernie去判断是这么多可能的哪一个，就把排序问题变成了一个多分类的问题，这样就很好的排序。语义层的关系：乱砍、乱伐，出现自然灾害，因为所以的关系，还有如果就的关系。尽管 但是 转折关系，或者并且这种并列关系，等等是语义的逻辑关系，大量的语料去学习这些关系，使得ernie能很好的提升自己的效果，

如下图所示：



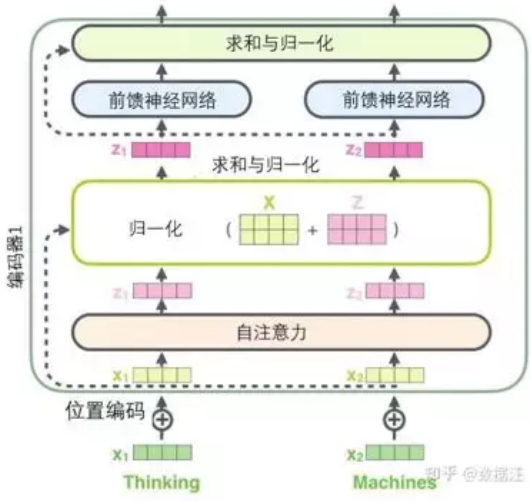
学习过程

有这么多的输入，那么对于输出怎么学习呢？这些任务让它很好的学习，关键看他的输出，大致分为两类，token_evel就是词汇层次的任务，一类是sentence_level就是句子层次的任务，可以理解为填空题，判断这个词到底是哪一类 哪个词 是敏感词 还是非敏感词，是关键词还是非关键词，另一部分呢就是句子输出cls是a类还是b类是顺序1还是顺序2，来学习到句子级别的任务。那么这部分神经网络是什么样的呢？



在这里插入图片描述

其实上述提到过，ERNIE是基于BERT的改进，内部网络的设计其实还是基于双向的self-attention的encoder。

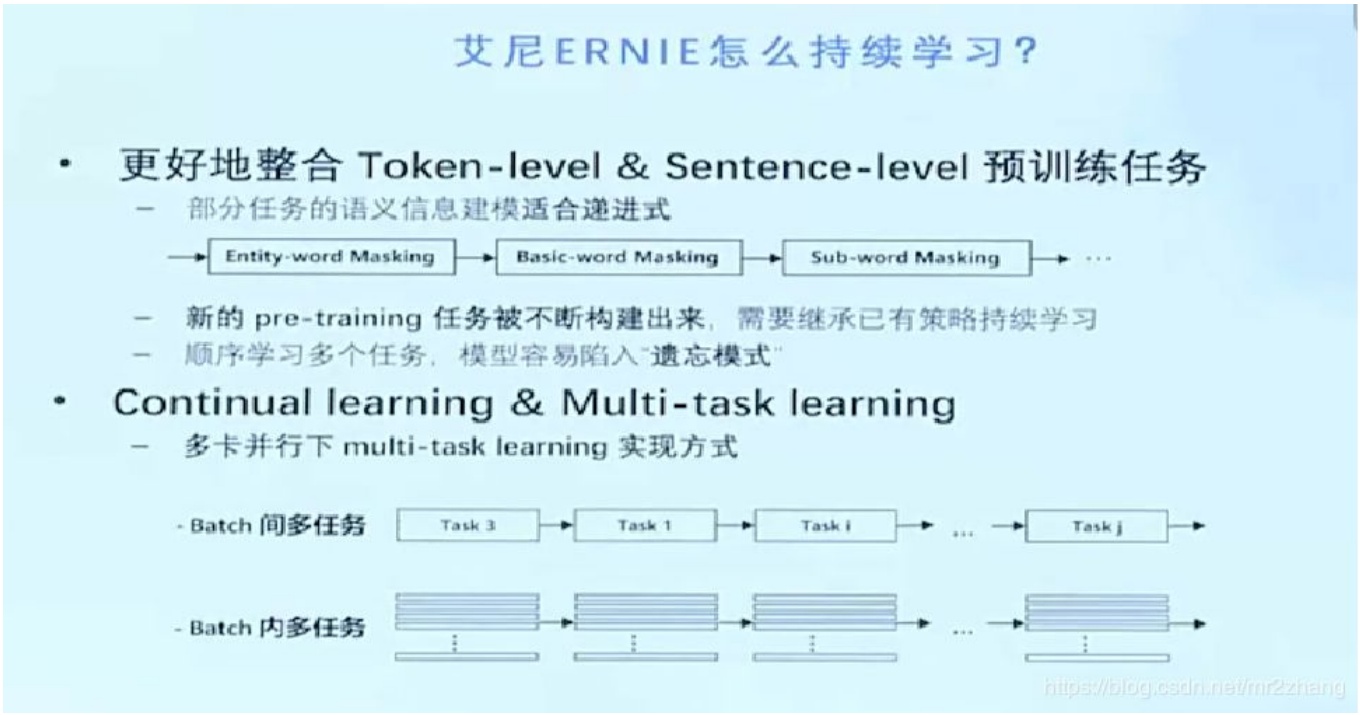


transformer编码器
《Attention Is All You Need》

transformer自注意力机制发现搭配关系

来自与all attention you need 这篇论文，tranfromer,总共有12层，然后每一层又包括自注意力机制、归一化、又包括求和与归一化、以及一些前馈神经网络的一些机制，最终构成了一个相对复杂的神经网络结构，这里面最核心的是自注意力机制，能学会每个词汇和词之间的关系比如说：词汇与词汇之间的搭配，keep up we have 他们之间的关系线的粗细表示关系的强弱。在一步步理解bert中已经描述过，这里不再过多的赘述。

我们训练任何深度学习模型基本上包括三大部分：第一部分定义任务，一部分网络结构，第三部分呢设计你的优化算法，前面大概介绍了任务和网络结构，接下来看一下优化算法，它本身是一个多任务的学习，并不是一口气学多个任务，都学好，很可能越学越差，相互之间不是促进而是促退，
那么ERNIE是怎么进行学习的呢?



ernie这里采用了两种机制，顺序学习，一次学习一个任务，参数是不断的递进更新的，这样的好处是比较容易但另一方面会有很大的问题，容易陷入到遗忘状态，这种比较适合任务关系比较紧密的任务，同样是隐码语义模型，第一次学被盖住的字，第二次学习被盖住测词，第三次学习被盖住的句子是怎样的，本质还是完型填空，但是填写的内容不一样，这样是可以进行顺序学习的。但是对于其他的任务可能就得同时的学习，比如每次迭代是一个batch,batch与batch之间的任务可以是不一样的，而同一个batch里面也可以学习不同的任务，batch内有多任务机制，batch内有多任务的机制，使得它能在不同的任务之间不断的来回切换，使它记得学到不同任务的特点，使它不会陷入到之前的遗忘模式。
ERNIE的官方对比效果图：

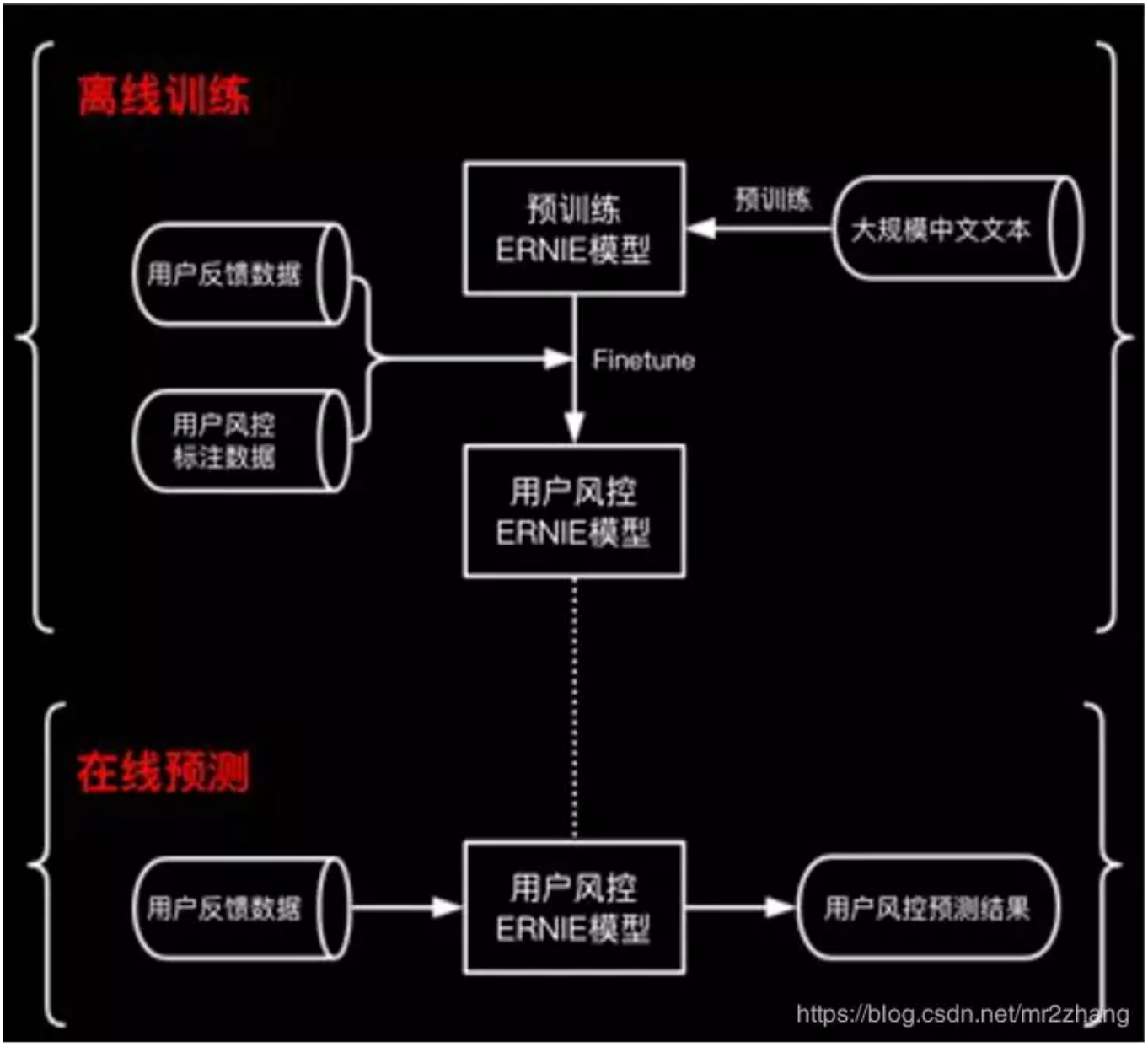
	数据集	ERNIE-Large效果	ERNIE-Base效果	BERT-Base效果
自动问答	NLPCC-DBQA	85.8% (+5.0%)	85.3%(+4.5%)	80.8%
自然语言推断	XNLI	81.0% (+3.8%)	79.7%(+2.5%)	77.2%
情感分析	ChnSentiCorp	95.8% (+1.5%)	95.5%(+1.2%)	94.3%
文本语义相似度	LCQMC	87.9% (+0.9%)	87.9%(+0.9%)	87.0%
命名实体识别	MSRA-NER	95.0% (+2.4%)	93.8%(+1.2%)	92.6%
机器阅读理解	Dureader	64.2% (+4.7%)	61.3%(+1.8%)	59.5%
机器阅读理解	CMRC2018	71.5% (+5.2%)	69.1%(+2.8%)	66.3%
机器阅读理解	DRCD	89.0% (+4.1%)	88.0%(+3.1%)	84.9%

这里XLNET没有中文，只对比了bert的效果。

ERNIE的应用案例

ERNIE效果那么好，怎么应用到我们自己的场景中呢?

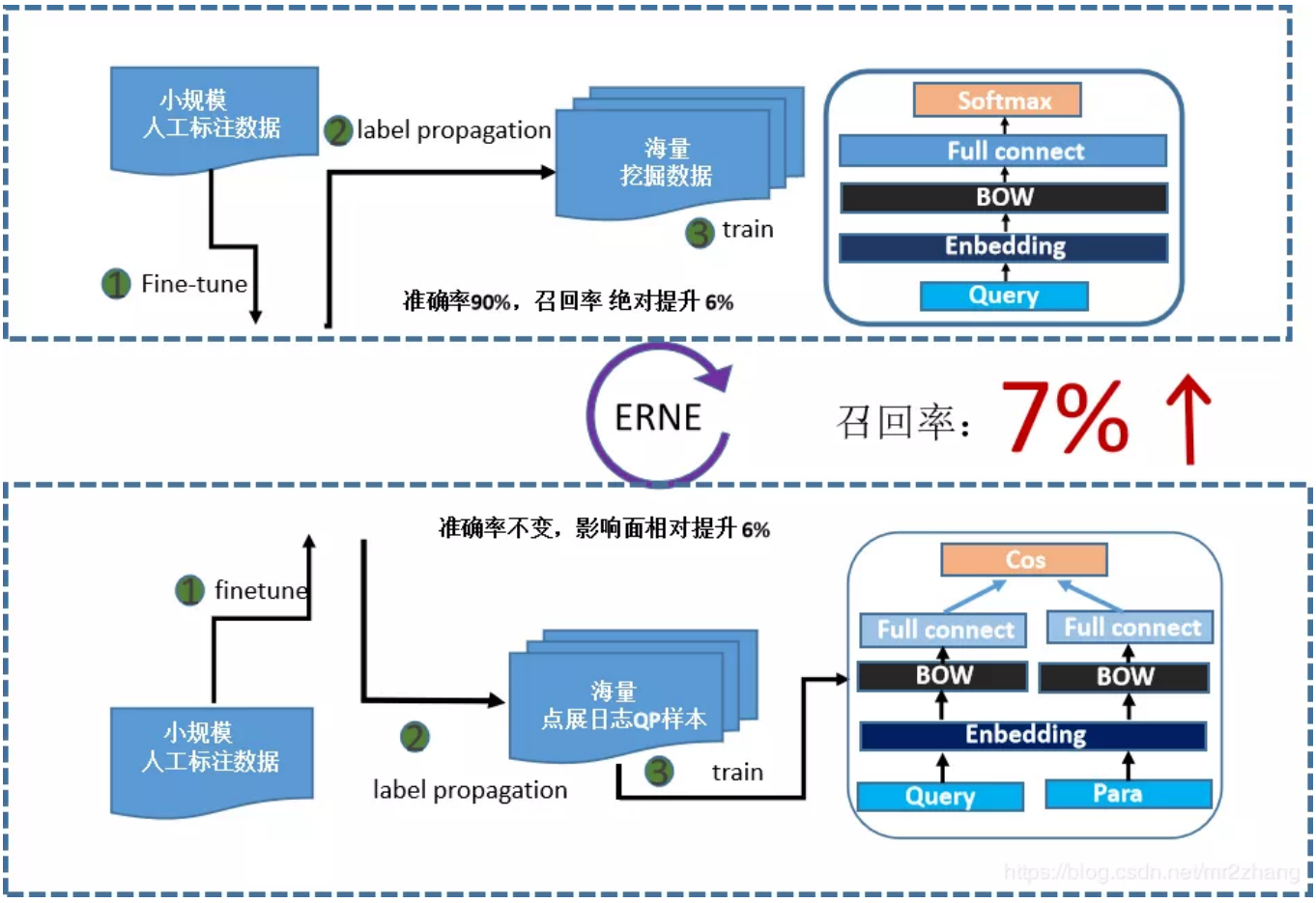
性能不敏感的场景：直接使用



度小满的风控识别上：训练完的ernie上直接进行微调，直接预测有没有风险对应的结果，传统的缺点：需要海量的数据，而这些数据也很难抓取到的，抓取这些特征之后呢还要进行复杂的文本特征提取，比如说挖掘短信中银行的催收信息，对数据要求的量很高，对数据人工的特征的挖掘也很高。这两项呢造成了大量的成本，如今只需ernie微调一下，当时直接在召回的排序上得到25%的优化。这种场景的特点是什么？对于用户的实时性的需求不是很强，不需要用户输入一个字段就返回结果。只要一天把所有数据得到，跑完，得到结果就可以了，统一的分析就可以了，适合少数据的分析场景

- 传统方法
  - 传统风控模型依赖于特征挖掘
  - 有标注的用户数据稀缺
- 对比效果提升
  - ERNIE用户风控模型，AUC提升显著
  - 21.5%的用户排序得到优化
  - ERNIE用户风控模型应用于度小满场景

ERNIE 的模型蒸馏案例：搜索问答Query识别和QP匹配



另外的一个场景需要非常高的性能优势的，采用的解决方案就是模型蒸馏，是搜索问答query识别和qp匹配，输入一个问题，得到答案，本质是文本匹配，实际是输入问题，把数据库中大量的候选答案进行匹配计算得分，把得分最高的返回。但是百度每天很多用户，很快的响应速度，数据量大，要求响应速度还快，这时候要求不仅模型特别准，而且还要特别快，怎么解决就是模型蒸馏，

把这个分成两段，第一段判断是不是真的问题可能有答案，过滤完是可能有答案的，再与数据库中进行匹配，因为大部分输入框的不一定是个问题，这样过滤掉一部分，排除掉一部分后，在做匹配就能得到很大的提升，提升还是不够
第一部分其实是文本分类，通过量的小规模的标注特征数据进行微调，得到一个好的模型，同时日志上是有很多没有标注的数据，用ernie对这些数据进行很好的标注，用一个更好的模型去标注数据，用这些标注数据训练相对简单的模型，就实现了蒸馏，ernie处理速度慢，但是可以用题海战术的方式训练简单的模型如simnet。

输入的是问题，接着词向量，然后是求和运算，然后是全连接，最后进行一个分类，这个模型比较简单。其实就是将ernie模型的效果蒸馏到了简单模型上，
同理，下面问题匹配也是，右边也是query和答案，embedding，加权求和，全连接，最后计算他们之间的预选相似度，就是余玄。时召回提升7%
搜索场景，用户的输入是多变的，没法穷尽所有的输入，提前计算好。

离线推荐

推荐场景：是可以提前计算好，保存好的，可变的比较少，视频本身就是存好的，变化量不会很大，更新也不会特别频繁，离线把相似度计算好，保存起来就可以，两两计算之间的相似度计算量是非常大的，
那么怎么减少计算量呢？使用了一个技术叫离线向量化，用ERNIE直接过一遍视频库，simnet的倒数第二层保留向量的输出。这样减少实际ERNIE直接对两两视频进行计算

原文链接：

https://blog.csdn.net/mr2zhang/article/details/103469195

推荐阅读

知识图谱存储与查询：自然语言记忆模块（NLM）

BERT论文笔记

XLNet 论文笔记

当BERT遇上知识图谱