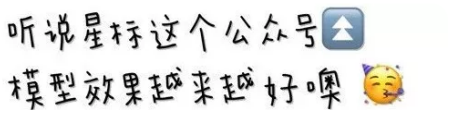
# 预训练模型超全知识点梳理与面试必备高频FAQ

JayLou娄杰 NewBeeNLP 2020-10-10




**作者** | JayLou娄杰
**原文** | 文末『阅读原文』处

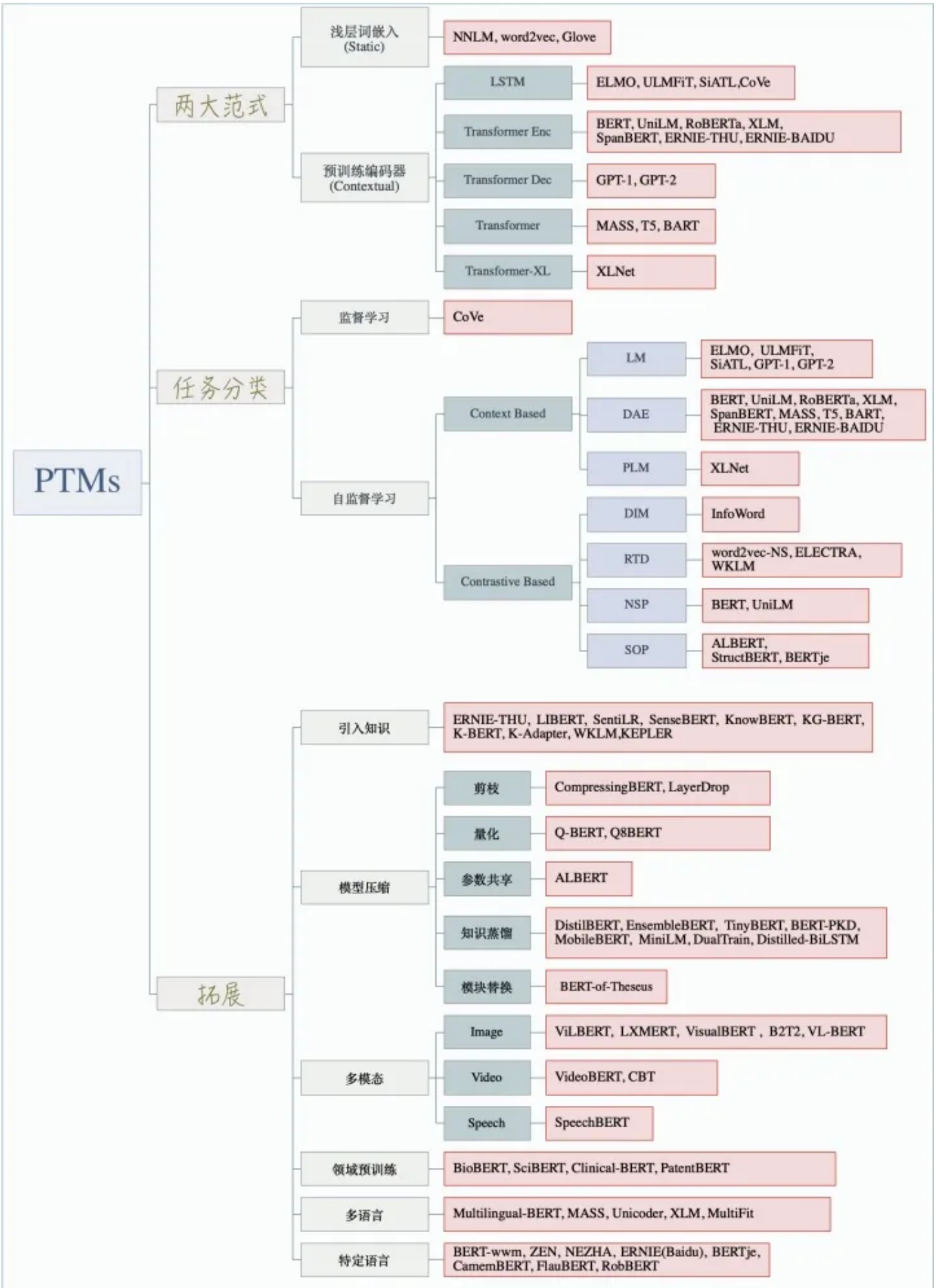
**预训练模型**(Pre-trained Models,PTMs)的出现将NLP带入了一个全新时代。2020年3月18日，邱锡鹏老师发表了关于NLP预训练模型的综述《Pre-trained Models for Natural Language Processing: A Survey》，这是一篇全面的综述，系统地对PTMs进行了归纳分类。

本文以此篇综述论文为主要参考，通过借鉴不同的归纳方法进行总结，同时也整合了专栏之前已经介绍过的《nlp中的词向量对比》和《nlp中的预训练语言模型总结》两篇文章，以QA形式对PTMs进行全面总结归纳。

获取**总结图片下载**以及**单模型精读**请到下面的github地址，希望为大家的学习工作提供一些帮助。

```
https://github.com/loujie0822/Pre-trained-Models
```

> 笔者注：本文总结与原综述论文也有一些不同之处（详见文末），如有错误或不当之处请指正。很多总结归纳的点不太好拿捏，大家多给意见~




# PTMs： NLP预训练模型

知乎专栏：高能NLP之路@JayLou

Ref：《Pre-trained Models for Natural Language Processing: A Survey》

## 为什么要进行预训练？

深度学习时代，为了充分训练深层模型参数并防止过拟合，通常需要更多标注数据喂养。在NLP领域，标注数据更是一个昂贵资源。PTMs从大量无标注数据中进行预训练使许多NLP任务获得显著的性能提升。总的来看，预训练模型PTMs的优势包括：

1. 在庞大的无标注数据上进行预训练可以获取更通用的语言表示，并有利于下游任务；
2. 为模型提供了一个更好的初始化参数，在目标任务上具备更好的泛化性能、并加速收敛；
3. 是一种有效的正则化手段，避免在小数据集上过拟合（一个随机初始化的深层模型容易对小数据集过拟合）；

## 词嵌入和分布式表示

词嵌入是自然语言处理（NLP）中语言模型与表征学习技术的统称。概念上而言，它是指把一个维数为所有词的数量的高维空间嵌入到一个维数低得多的连续向量空间中，每个单词或词组被映射为实数域上的向量，这也是分布式表示：向量的每一维度都没有实际意义，而整体代表一个具体概念。

分布式表示相较于传统的独热编码（one-hot）表示具备更强的表示能力，而独热编码存在维度灾难和语义鸿沟（不能进行相似度计算）等问题。传统的分布式表示方法，如矩阵分解（SVD/LSA）、LDA等均是根据全局语料进行训练，是机器学习时代的产物。

PTMs也属于分布式表示的范畴，本文的PTMs主要介绍深度学习时代、自NNLM[2]以来的 “modern” 词嵌入。

## PTMs两大范式

PTMs的发展经历从浅层的词嵌入到深层编码两个阶段，按照这两个主要的发展阶段，我们归纳出PTMs两大范式：「浅层词嵌入」和「预训练编码器」。

## 浅层词嵌入

浅层词嵌入，这一类PTMs范式是我们通常所说的“词向量”，其主要特点是学习到的是上下文独立的静态词嵌入，其主要代表为NNLM[2]、word2vec（CBOW[3]、Skip-Gram[3]）、Glove等。这一类词嵌入通常采取浅层网络进行训练，而应用于下游任务时，整个模型的其余部分仍需要从头开始学习。因此，对于这一范式的PTMs没有必要采取深层神经网络进行训练，采取浅层网络加速训练也可以产生好的词嵌入。

浅层词嵌入的主要缺陷为：

- 词嵌入与上下文无关，每个单词的嵌入向量始终是相同，因此不能解决一词多义的问题。
- 通常会出现OOV问题，为了解决这个问题，相关文献提出了字符级表示或sub-word表示，如CharCNN[5]、FastText[6]和 Byte-Pair Encoding[7]。

| 词嵌入 | 训练目标 | 全局/局部语料 | 特点 |
| --- | --- | --- | --- |
| NNLM | 语言模型 | 局部语料 | 基于语言模型进行训练的，词嵌入只不过是NNLM的一个产物而已； |
| word2vec | 非语言模型（窗口上下文） | 局部语料 | 1)为加速训练舍弃NNLM中的隐藏层、词嵌入直接sum;<br>2) 采用分层SoftMax(带权路径最小的哈夫曼树) 和负采样 进行运算优化;<br>3)损失函数：带权重的交叉熵，权重固定； |
| Glove | 非语言模型（词共现矩阵） | 全局语料 | 1) 基于全局语料构建词共现矩阵然后进行[illegible]<br>2)损失函数：最小平方损失函数，权重可以做映射变换。 |



图1给出了三种常见的浅层词嵌入之间的对比，Glove可以被看作是更换了目标函数和权重函数的全局word2vec。此外，相关文献也提出了句子和文档级别的嵌入方式，如 Skip-thought[8]、Context2Vec[9]等。

## 预训练编码器

第二类PTMs范式为预训练编码器，主要目的是通过一个预训练的编码器能够输出上下文相关的词向量，解决一词多义的问题。这一类预训练编码器输出的向量称之为「上下文相关的词嵌入」。

| 编码器 | PTMs代表 | 计算方式 | 特点 |
| --- | --- | --- | --- |
| MLP | NNLM/word2vec | 前馈+并行 | 不考虑序列（位置）信息，不能处理变长序列； |
| CNNs | | 前馈+并行 | 考虑序列（位置）信息，不能处理长距离依赖，聚焦于n-gram的局部上下文编码，pooling操作会导致序列（位置）信息丢失； |
| RNNs | ELMO | 循环+串行 | 天然适合处理序列（位置）信息，但仍不能处理长距离依赖（由于BPTT导致的梯度消失等问题），故又称之为“较长的短期记忆单元(LSTM)” |
| Transformer | GPT（Decoder）<br>BERT（Encoder） | 前馈+并行 | 1）self-attention解决长距离依赖，无位置偏差；<br>2）self-attention可看作是权重动态调整的全连接网络； |
| Transformer-XL | XLNet | 循环+串行 | 基于Transformer引入循环机制+相对位置编码，增强长距离建模能力； |
| 长距离依赖建模能力 | | | Transformer-XL > Transformer > RNNs > CNNs |



图2给出了NLP各种编码器间的对比。PTMs中预训练编码器通常采用LSTM和Transformer（Transformer-XL），其中Transformer又依据其attention-mask方式分为Transformer-Encoder和Transformer-Decoder两部分。此外，Transformer也可看作是一种图神经网络GNN[10]。

这一类「预训练编码器」范式的PTMs主要代表有**ELMO**[11]、**GPT-1**[12]、**BERT**[13]、**XLNet**[14]等。

## PTMs按照任务类型分类

PTMs按照任务类型可分为2大类：监督学习 和 无监督学习/自监督学习。

监督学习在NLP-PTMs中的主要代表就是**CoVe**[15]，CoVe作为机器翻译的encoder部分可以应用于多种NLP下游任务。除了CoVe外，NLP中的绝大多数PTMs属于自监督学习。

自监督学习是无监督学习的一种方法[16]，自监督学习[17]主要是利用辅助任务从大规模的无监督数据中挖掘自身的监督信息，通过这种构造的监督信息对网络进行训练，从而可以学习到对下游任务有价值的表征。因此，从“构造监督信息”这个角度来看，自监督也可看作是监督学习和无监督学习的一种融合[1]。严格地讲，从是否由人工标注来看，自监督学习属于无监督学习的范畴。

综合各种自监督学习的分类方式，笔者将NLP-PTMs在自监督学习中分为两种类型[17]：基于上下文（Context Based）和基于对比（Contrastive Based）。

### 基于上下文（Context Based）

基于上下文的PTMs，主要基于数据本身的上下文信息构造辅助任务，在NLP中我们通常引入语言模型作为训练目标。PTMs中的语言模型主要分为三大类：

| 语言模型 | | 优点 | 缺点 |
| --- | --- | --- | --- |
| LM | 自回归语言模型 | 语言模型联合概率的无偏估计，考虑被预测单词之间的相关性，适合生成任务； | 按照文本序列顺序拆解（从左至右分解），无法获取双向上下文信息表征； |
| DAE | 自编码语言模型 | 本质为降噪自编码(DAE)特征表示，通过引入噪声[MASK]构建MLM获取双向上下文信息表征 | 1）引入独立性假设，为语言模型联合概率的有偏估计，没有考虑预测token之间的相关性；<br>2）预训练时的「MASK」噪声在finetune阶段不会出现，造成两阶段不匹配问题； |
| PLM | 排列语言模型 | 综合了LM和DAE-LM两者的优点 | 收敛速度较慢，XL对于长序列文本建模采样，并仅预测了排列后序列中的最后几个token； |



**第一类：自回归语言模型（LM）**

$$p(x_{1:T}) = \prod_{t=1}^{T} p(x_t|x_{0:t-1})$$

- 优点：
  - **语言模型（language model，LM）联合概率的无偏估计**，即为传统的语言模型，考虑被预测单词之间的相关性，天然适合处理自然生成任务；
- 缺点：
  - 联合概率按照文本序列**顺序拆解**（从左至右分解），无法获取双向上下文信息表征；
- 代表模型：ELMO、GPT-1、GPT-2[18]、ULMFiT[19]、SiATL[20]；

**第二类：自编码语言模型（DAE）**

$$p(x_{1:T}) \approx \sum_{t=1}^{T} m_t log p(x_t|\tilde{x})$$

- 优点：
  - 本质为**降噪自编码(DAE)特征表示，通过引入噪声[MASK]构建MLM**(Masked language model)，获取双向上下文信息表征（本文将自编码语言模型统一称为**DAE**，旨在采用部分损坏的输入，旨在恢复原始的未失真输入）；如果当前token被预测，则 $m_t = 1$ 否则 $m_t = 0$ ， $\tilde{x}$ 为原始文本被替换后的输入。
- 缺点：
  - 引入独立性假设，为**语言模型联合概率的有偏估计**，没有考虑预测token之间的相关性；

  - 预训练时的「MASK」噪声在finetune阶段不会出现，造成两阶段不匹配问题；为解决这一问题，在15%被预测的token中，80%被替换为「MASK」，10%被随机替换，10%被替换为原词。

- 代表模型：BERT、MASS [21]、T5[22]、RoBERTa[23]、UniLM[24]、XLM[25]、SpanBERT[26]、ERNIE-Baidu[27][28]、E-BERT[29]、ERNIE-THU[30]、BART[31]。

**BERT**[13]是自编码语言模型的一个典型代表，但其采用的MLM策略和Transformer-Encoder结构，导致其不适合直接处理生成任务。为了解决这一问题，也可采用基于**Seq2Seq MLM**方法：encoder部分采取masked策略，而decoder部分以自回归的方式预测encoder部分被mask的token。此外，还有很多基于自编码语言模型的PTMs提出了不同的MLM增强策略，称之为 Enhanced Masked Language Modeling (**E-MLM**) [1]。

上述DAE具体的PTMs方法见图4。

**第三类：排列语言模型（PLM）**

排列语言模型综合了LM和DAE-LM两者的优点。严格来讲，PLM和LM是标准的自回归语言模型（注：PLM是一种广义的自回归方法[14]），而MLM不是一个标准的语言模型，其引入独立性假设，**隐式地学习**预测token（mask部分本身的强相关性）之间的关系。如果衡量序列中被建模的依赖关系的数量，标准的自回归语言模型可以达到上界，不依赖于任何独立假设。LM和PLM能够通过自回归方式来**显式地学习**预测token之间的关系。然而，LM无法对双向上下文进行表征，借鉴 NADE[32]的思想，PLM将这种传统的自回归语言模型（LM）进行推广，将顺序拆解变为**随机拆解**（从左至右分解），产生上下文相关的双向特征表示。

PLM最为典型的代表就是**XLNet**[14]，这是对标准语言模型的一个复兴[33]：提出一个框架来连接标准语言模型建模方法和预训练方法。

一个关键问题：为什么PLM可以实现双向上下文的建模？**PLM的本质就是语言模型联合概率的多种分解机制的体现，其将LM的顺序拆解推广到随机拆解**。PLM没有改变原始文本序列的自然位置，只是定义了token预测的顺序。PLM只是针对语言模型建模不同排列下的因式分解排列，并不是词的位置信息的重新排列。

最后，我们对基于上述三类语言模型的PTMs进行总结：

| 语言模型 | 模型 | 编码器 | 主要亮点 |
| --- | --- | --- | --- |
| LM | ELMO | LSTM | 2个单向语言模型（前向和后向）的拼接 |
| LM | ULMFiT | LSTM | 引入逐层解冻解决finetune中的灾难性问题； |
| LM | SiATL | LSTM | 引入逐层解冻+辅助LM解决finetune中的遗忘问题； |
| LM | GPT-1 | Transformer-Decoder | 首次将Transformer应用于预训练语言模型； |
| LM | GPT-2 | Transformer-Decoder | 没有特定模型的精调流程，生成任务取得很好效果； |
| DAE：MLM | BERT | Transformer-Encoder | MLM获取上下文相关的双向特征表示； |
| DAE：Seq2SeqMLM | MASS / T5 | Transformer | 改进BERT生成任务：统一为类似Seq2Seq的预训练框架； |
| DAE：E-MLM | UNILM | Transformer-Encoder | 改进BERT生成任务：3个mask矩阵：LM/MLM/Seq2Seq LM； |
| DAE：E-MLM | RoBERTa | Transformer-Encoder | 预训练过程中采取动态mask，不像BERT在预处理做静态mask； |
| DAE：E-MLM | XLM | Transformer-Encoder | 在翻译语言模型的平行语料上执行MLM |
| DAE：E-MLM | SpanBERT | Transformer-Encoder | 采取random span mask和span boundary objective 2个预训练目标； |
| DAE：E-MLM | ENRIE-BAIDU | Transformer-Encoder | mask实体和短语，2.0引入多任务进行增量学习； |
| DAE：E-MLM | ENRIE-THU/E-BERT | Transformer-Encoder | 引入知识：将实体向量与文本表示融合； |
| DAE | BART | Transformer | 采取Seq2Seq框架和5种DA方法 |
| PLM | XLNet | Transformer-XL | 双向上下文表征+双注意力流 |



## 基于对比（Contrastive Based）

基于对比（Contrastive Based），不同于**Context Based**主要基于数据本身的上下文信息构造辅助任利用，Contrastive Based主要利用样本间的约束信息构造辅助任务，这类方法也是 Contrastive learning[34]（CTL）。CTL假设观察到的文本对（正样本）在语义上比随机采样的文本（负样本）更相似。CTL 背后的原理是「在对比中学习」。相较于语言建模，CTL 的计算复杂度更低，因而在预训练中是理想的替代训练标准。

CTL通过构建正样本（positive）和负样本（negative），然后度量正负样本的距离来实现自监督学习[17]:可以使用点积的方式构造距离函数，然后构造一个 softmax 分类器，以正确分类正样本和负样本。鼓励相似性度量函数将较大的值分配给正例，将较小的值分配给负例：

$$\mathcal{L}_N = -\mathbb{E}_{x,y+,y-}\left[\log \frac{\exp(s\left(x, y^{+}\right))}{\exp(s(x, y^{+})) + \sum_{j=1}^{N-1} \exp\left(s(x, y_j^{-})\right)}\right]$$

相似性度量函数通常可采取两种方式：

$$s(x,y) = f_{enc(x)}^{T} f_{enc(x)}$$

或

$$s(x,y) = f_{enc}(x \oplus y)$$

**第一类：Deep InfoMax (DIM)**

DIM方法来源于CV领域，对于全局的特征（编码器最终的输出）和局部特征（编码器中间层的特征），DIM需要判断全局特征和局部特征是否来自同一图像[17]。

**InfoWord** [35]将DIM引入到NLP中，用Mutual Information的一个下界InfoNCE来重新解释BERT和XLNET的objective，并提出一个新的DIM objective以最大化一个句子的global representation和其中一个ngram的local representation之间的Mutual Information。

**第二类：Replaced Token Detection (RTD)**

噪声对比估计（Noise-Contrastive Estimation，NCE）[36]通过训练一个二元分类器来区分真实样本和假样本，可以很好的训练词嵌入。RTD于与 NCE 相同，根据上下文语境来预测token是否替换 。

- **word2vec**[3]中的negative sampling可看作是RTD，负样本从词表中进行带权采样。
- **ELECTRA**[37]提出了一种新的预训练任务框架，构建生成器-判别器，生成器通过MLM任务对被mask的token进行预测，迭代器判断原始句子中的每个token是否被replace过。生成器相当于对输入进行了筛选，使判别器的任务更难，从而学习到更好的表示。生成器-判别器共享embedding，生成器部分采用small-bert，判别器部分对每一个token采用sigmoid计算loss。finetune阶段只采用判别器部分。RTD也被看作解决MLM中「MASK」在预训练和finetune间差异的一种手段。
- **WKLM**[38]在实体level进行替换，替换为具有相同实体类型的实体名称。

**第三类：Next Sentence Prediction (NSP)**

NSP 区分两个输入句子是否为训练语料库中的连续片段，第二个句子50%为第一句子实际的连续片段，50%从其他语料随机选择。NSP可以引导模型理解两个输入句子之间的关系，从而使对此信息敏感的下游任务受益，如QA任务。而RoBERTa[23]表明：NSP在对单个文档中的文本块进行训练时，去除NSP任务或在下游任务上可以稍微提高性能。

**第四类：Sentence Order Prediction (SOP)**

SOP 使用同一文档中的两个连续片段作为正样本，而相同的两个连续片段互换顺序作为负样本。NSP融合了主题预测和相关性预测，主题预测更容易，这使得模型进行预测时仅依赖于主题学习。与NSP不同，SOP使用同一文档中的两个连续段作为正样本，但顺序互换为负样本。采取SOP任务的PTMs有ALBERT[39]、StructBERT[40]、BERTje[41]。

图5对上述基于对比（Contrastive Based）的四类PTMs进行了总结：

| Contrastive Based方法 | 特点 | PTMs |
| --- | --- | --- |
| DIM：Deep InfoMax | 最大化全局特征和局部特征间的互信息 | InfoWord |
| RTD：Replaced Token Detection | 根据上下文语境来预测token是否替换 | word2vec-ns/ELECTRA/WKLM |
| NSP：Next Sentence Prediction | 区分两个输入句子是否为语料库中的连续片段 | BERT/ERNIE/SpanBERT |
| SOP：Sentence Order Prediction | 相关性预测，将两个连续片段互换顺序 | ALBERT/StructBERT/BERTje |



# PTMs有哪些拓展

## 引入知识

PTMs通常从通用大型文本语料库中学习通用语言表示，但是缺少特定领域的知识。PTMs中设计一些辅助的预训练任务，将外部知识库中的领域知识整合到PTMs中被证明是有效的[1]。

- **ERNIE-THU**[30]将在知识图谱中预先训练的实体嵌入与文本中相应的实体提及相结合，以增强文本表示。由于语言表征的预训练过程和知识表征过程有很大的不同，会产生两个独立的向量空间。为解决上述问题，在有实体输入的位置，将实体向量和文本表示通过非线性变换进行融合，以融合词汇、句法和知识信息。
- **LIBERT**[42]（语言知识的BERT）通过附加的语言约束任务整合了语言知识。
- **SentiLR**[43]集成了每个单词的情感极性，以将MLM扩展到标签感知MLM（LA-MLM），ABSA任务上都达到SOTA。
- **SenseBERT**[44] 不仅能够预测被mask的token，还能预测它们在给定语境下的实际含义。使用英语词汇数据库 WordNet 作为标注参照系统，预测单词在语境中的实际含义，显著提升词汇消歧能力。
- **KnowBERT**[45] 与实体链接模型以端到端的方式合并实体表示。
- **KG-BERT**[46]显示输入三元组形式，采取两种方式进行预测：构建三元组识别和关系分类，共同优化知识嵌入和语言建模目标。这些工作通过实体嵌入注入知识图的结构信息。
- **K-BERT**[47]将从KG提取的相关三元组显式地注入句子中，以获得BERT的扩展树形输入。
- **K-Adapter**[48]通过针对不同的预训练任务独立地训练不同的适配器来注入多种知识，从而可以不断地注入知识，以解决注入多种知识时可能会出现灾难性遗忘问题。

- 此外，这类PTMs还有WKLM[38]、KEPLER[49]和[50]等。

## 模型压缩

由于预训练的语言模型通常包含至少数亿个参数，因此很难将它们部署在现实应用程序中的在线服务和资源受限的设备上。模型压缩是减小模型尺寸并提高计算效率的有效方法。

5种PTMs的压缩方法为：

- **pruning（剪枝）**：将模型中影响较小的部分舍弃。
  - 如Compressing BERT[51]，还有结构化剪枝 LayerDrop [52]，其在训练时进行Dropout，预测时再剪掉Layer，不像知识蒸馏需要提前固定student模型的尺寸大小。
- **quantization（量化）**：将高精度模型用低精度来表示；
  - 如Q-BERT[53]和Q8BERT[54]，量化通常需要兼容的硬件。
- **parameter sharing （参数共享）**：相似模型单元间的参数共享；
  - ALBERT[39]主要是通过矩阵分解和跨层参数共享来做到对参数量的减少。
- **module replacing（模块替换）**：
  - BERT-of-Theseus[55]根据伯努利分布进行采样，决定使用原始的大模型模块还是小模型，只使用task loss。
- **knowledge distillation （知识蒸馏）**：通过一些优化目标从大型、知识丰富、fixed的teacher模型学习一个小型的student模型。蒸馏机制主要分为3种类型：
  - 从软标签蒸馏：DistilBERT [56]、EnsembleBERT[57]
  - 从其他知识蒸馏：TinyBERT[58]、BERT-PKD、MobileBERT[59] 、 MiniLM[60] 、DualTrain[61]
  - 蒸馏到其他结构：Distilled-BiLSTM[62]

| 知识蒸馏PTMs | 主要方法 |
| --- | --- |
| DistilBERT | 软标签蒸馏，KL散度作为loss |
| TinyBERT | 层与层蒸馏：embedding/hidden state/self-attention distributions |
| BERT-PKD | 层与层蒸馏：hidden state |
| MobileBERT | 软标签蒸馏+层与层蒸馏：hidden state/self-attention distributions |
| MiniLM | Self-attention distributions /self- attention value relation. |
| DualTrain | Dual Projection |
| Distilled-BiLSTM | 软标签蒸馏，将Transformer蒸馏到LSTM |
| EnsembleBERT | 取多个Ensemble模型的软标签进行蒸馏 |



## 多模态

随着PTMs在NLP领域的成功，许多研究者开始关注多模态领域的PTMs，主要为通用的视觉和语言特征编码表示而设计。多模态的PTMs在一些庞大的跨模式数据语料库（带有文字的语音、视频、图像）上进行了预训练，如带有文字的语音、视频、图像等，主要有 VideoBERT[63]、CBT[64]、UniViLM[65]、ViL-BERT[66]、LXMERT[67]、VisualBERT [68]、B2T2[69]、Unicoder-VL[70]、UNITER [71]、VL-BERT[72]、SpeechBERT[73]。

## 领域预训练

大多数PTM都在诸如Wikipedia的通用语料中训练，而在领域化的特定场景会收到限制。如基于生物医学文本的BioBERT[74]，基于科学文本的SciBERT[75]，基于临床文本的Clinical-BERT[76]。一些工作还尝试将PTMs适应目标领域的应用，如医疗实体标准化[77]、专利分类PatentBERT [78]、情感分析SentiLR[79]关键词提取[80]。

## 多语言和特定语言

学习跨语言共享的多语言文本表示形式对于许多跨语言的NLP任务起着重要的作用。

- **Multilingual-BERT**[81]在104种 Wikipedia文本上进行MLM训练（共享词表），每个训练样本都是单语言文档，没有专门设计的跨语言目标，也没有任何跨语言数据，M-BERT也可以很好的执行跨语言任务。
- **XLM** [25]通过融合跨语言任务（翻译语言模型）改进了M-BERT，该任务通过拼接平行语料句子对进行MLM训练。

- **Unicoder**[82]提出了3种跨语言预训练任务：1)cross-lingual word recovery；2) cross-lingual paraphrase classification;3) cross-lingual masked language model.

虽然多语言的PTMs在跨语言上任务表现良好，但用单一语言训练的PTMs明显好于多语言的PTMs。此外一些单语言的PTMs被提出：BERT[83]，ZEN[84],NEZHA[85], ERNIE-Baidu[27][28], BERTje[86], CamemBERT[87], FlauBERT[88], RobBERT [89]。

# 对PTMs进行迁移学习

PTMs从大型语料库中获取通用语言知识，如何有效地将其知识适应下游任务是一个关键问题。迁移学习的方式主要有归纳迁移（顺序迁移学习、多任务学习）、领域自适应（转导迁移）、跨语言学习等。NLP中PTMs的迁移方式是顺序迁移学习。

## 如何迁移?

1）选择合适的**预训练任务**：语言模型是PTM是最为流行的预训练任务；同的预训练任务有其自身的偏置，并且对不同的任务会产生不同的效果。例如，NSP任务可以使诸如问答（QA）和自然语言推论（NLI）之类的下游任务受益。

2）选择合适的**模型架构**：例如BERT采用的MLM策略和Transformer-Encoder结构，导致其不适合直接处理生成任务。

3）选择合适的**数据**：下游任务的数据应该近似于PTMs的预训练任务，现在已有有很多现成的PTMs可以方便地用于各种特定领域或特定语言的下游任务。

4）选择合适的**layers**进行transfer：主要包括Embedding迁移、top layer迁移和all layer迁移。如word2vec和Glove可采用Embedding迁移，BERT可采用top layer迁移，Elmo可采用all layer迁移。

5）**特征集成**还是**fine-tune**？对于特征集成预训练参数是freeze的，而fine-tune是unfreeze的。特征集成方式却需要特定任务的体系结构，fine-tune方法通常比特征提取方法更为通用和方便。

## fine-tune策略：

通过更好的微调策略进一步激发PTMs性能

- 两阶段fine-tune策略：如第一阶段对中间任务或语料进行finetune，第二阶段再对目标任务fine-tune。第一阶段通常可根据特定任务的数据继续进行fine-tune预训练。
- 多任务fine-tune：MTDNN[90]在多任务学习框架下对BERT进行了fine-tune，这表明多任务学习和预训练是互补的技术。
- 采取额外的适配器：fine-tune的主要缺点是其参数效率低，每个下游任务都有自己的fine-tune参数。因此，更好的解决方案是在固定原始参数的同时，将一些可fine-tune的适配器注入PTMs。
- 逐层阶段：逐渐冻结而不是同时对所有层进行fine-tune，也是一种有效的fine-tune策略。

## PTMs还有哪些问题需要解决?

（本部分来自[91]，有删减和修正）

虽然 PTMs已经在很多 NLP 任务中显示出了他们强大的能力，然而由于语言的复杂性，仍存在诸多挑战。综述论文给出了五个未来 PTMs发展方向的建议。

### PTMs的上限

目前，PTMs并没有达到其上限。大多数的PTMs可通过使用更长训练步长和更大数据集来提升其性能。目前NLP中的SOTA也可通过加深模型层数来更进一步提升。这将导致更加高昂的训练成本。因此，一个更加务实的方向是在现有的软硬件基础上，设计出更高效的模型结构、自监督预训练任务、优化器和训练技巧等。例如，ELECTRA [37]就是此方向上很好的一个解决方案。

### 面向任务的预训练和模型压缩

在实践中，不同的目标任务需要 PTMs拥有不同功能。而 PTMs与下游目标任务间的差异通常在于两方面：模型架构与数据分布。尽管较大的PTMs通常情况下会带来更好的性能表现，但在低计算资源下如何使用是一个实际问题。例如，对于 NLP 的 PTM 来说，对于模型压缩的研究只是个开始，Transformer 的全连接架构也使得模型压缩具有挑战性。

### PTMs的架构设计

对于PTMs，Transformer 已经被证实是一个高效的架构。然而 Transformer 最大的局限在于其计算复杂度（输入序列长度的平方倍）。受限于 GPU 显存大小，目前大多数 PTM 无法处理超过 512 个

token 的序列长度。打破这一限制需要改进 Transformer 的结构设计，例如 Transformer-XL[92]。

## finetune中的知识迁移

finetune是目前将 PTM 的知识转移至下游任务的主要方法，但效率却很低，每个下游任务都需要有特定的finetune参数。一个可以改进的解决方案是固定PTMs的原始参数，并为特定任务添加小型的finetune适配器，这样就可以使用共享的PTMs 服务于多个下游任务。

## PTMs 的解释性与可靠性

PTMs 的可解释性与可靠性仍然需要从各个方面去探索，它能够帮助我们理解 PTM 的工作机制，为更好的使用及性能改进提供指引。

> 写在最后：本文总结与原综述论文[1]的一些不同之处：

1. 本文定义了PTMs两大范式：浅层词嵌入和预训练编码器。不同于原文，XLNet在原综述论文中被归为Transformer-Encoder，本文认为将其归为Transformer-XL更合适。

2. 本文PTMs按照自监督学习的分类不同于原文。本文按照 基于上下文（Context Based）和基于对比（Contrastive Based）两种方式归类；将原文的LM、MLM、DAE、PLM归为Context Based；

3. 本文将原文MLM和DAE统一为DAE；

4. 其他：1）在3.1.2的E-MLM段落中，可以将StructBERT拿出来，只放在SOP；2）3.1.5对ELECTRA的描述，应采取ELECTRA原文中的主要方法（参数共享），两阶段的方法只是一种实验尝试；3）在puring部分可以补充LayerDrop；4）应将UniLM归为MLM；

## 一起交流

**重磅推荐！NewBeeNLP目前已经建立了多个不同方向交流群（机器学习 / 深度学习 / 自然语言处理 / 面试交流 / 大厂内推 等），赶紧添加下方微信加入一起讨论学习吧！**