预训练语言模型

常宝宝
北京大学计算语言学研究所
chbb@pku.edu.cn

概要

- 预训练语言模型概要
- ELMo
- GPT
- BERT
- BART

迁移学习

- 利用深度学习建模NLP任务，需要标注数据
  - 句法树库
  - 语义角色标注语料库(命题库)
  - …
- 标注数据规模有限

- 人类能把以前解决老问题时所学到的知识或者经验用于解决新问题

- 基于深度学习模型实现知识迁移
  - 在不同的任务之间实现知识迁移
  - 实现知识从源任务到目标任务的迁移
  - 在富资源任务上训练模型，将知识迁移至贫资源任务

迁移学习

- 迁移学习可以分成两个阶段

  (1) 预训练阶段

  基于源任务数据训练源任务模型

  (2) 迁移阶段

  迁移源任务习得的知识并基于目标任务数据训练目标任务模型
- 迁移学习可以缓解特定任务对标注数据规模的需求
- 设想，
  - 源任务：词类标注 词类标注数据集
  - 目标任务：句法分析 句法树库
  - 基于词类标注数据集预训练，利用词类标注模型中的参数初始化句法分析模型(部分)参数，基于句法树库训练句法分析模型

自指导学习

- 有指导预训练：利用标注数据进行预训练
  - 需要有大规模源任务标注数据
  - 选择机器翻译作为源任务，利用大规模平行语料库训练模型
- 自指导预训练：利用无标注数据进行训练
  - 存在海量无标注数据(未加标注的语料库)
  - 从无标注数据中提取指导信号，构造**自指导任务**作为源任务
- 自指导预训练是NLP中主流预训练技术
- 预训练技术显著提升了众多NLP任务的性能

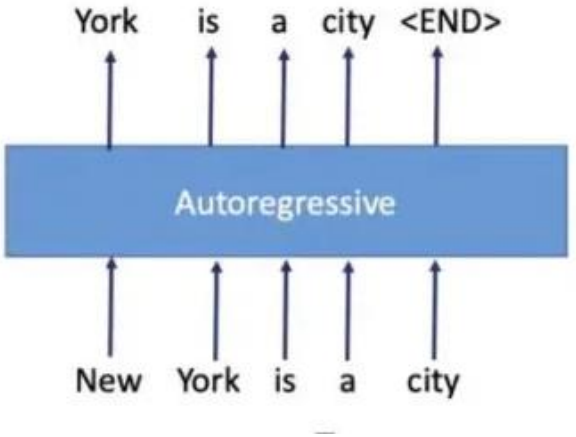
预训练

- 自指导预训练任务
  - 自回归语言重构
  - 降噪自编码任务

- 自回归语言生成任务

$$p(x_1 x_2 \cdots x_T) = \prod_{t=1}^{N} p(x_t | \boldsymbol{x}_{<t})$$

以自回归方式重构训练语料
构建自回归语言模型

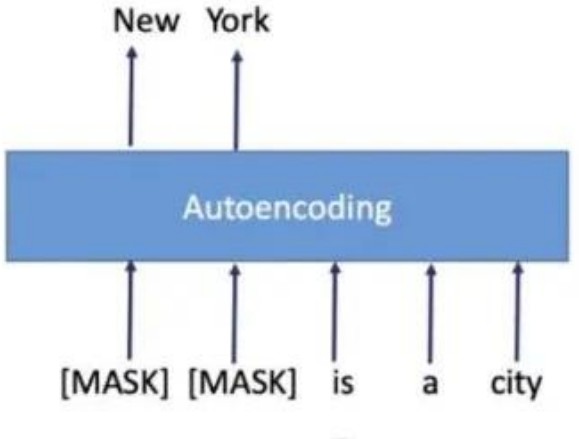


$$\log p_\theta(\boldsymbol{x}) = \sum_{t=1}^{T} \log p_\theta(x_t | \boldsymbol{x}_{<t})$$

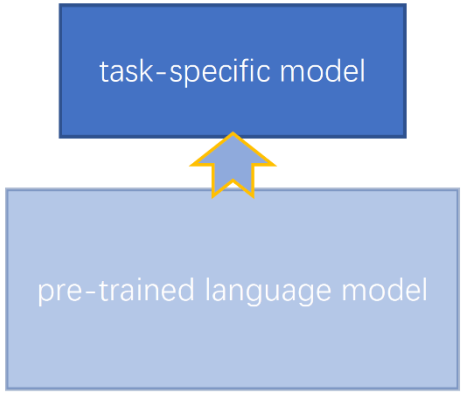
预训练

- 降噪自编码任务
  - 按照一定的策略向文本中注入噪音
    - 例如，按照一定策略遮蔽原始文本中的词例
  - 构建模型，利用模型去除噪音，重构原始文本
    - 例如，利用模型重构被遮蔽的词例

$$\log p_\theta(\overline{\boldsymbol{x}}|\hat{\boldsymbol{x}}) = \sum_{t=1}^{T} m_t \log p_\theta(x_t|\hat{\boldsymbol{x}})$$



知识迁移



- 迁移架构
  - 预训练语言模型 + 目标任务模型结构
  - 模型参数、分布式表示
- 迁移策略
  - 特征提取(feature extraction)
    - 在目标任务训练过程中，预训练模型参数冻结
  - 参数精调(fine-tuning)
    - 在目标任务训练过程中，预训练模型参数同时更新
- 预训练模型与目标任务架构逐渐同质化
  - 只在预训练模型架构添加较小的任务相关组件

NLP领域中的预训练

- 词向量预训练(word embedding)
  - 2008, C&W model
  - 2013, word2vec model
  - 2014, GloVe model
  - 2016, fastText model
- 语境敏感向量(contextualized word embedding)
  - 2018, ELMo、GPT、BERT
  - 2019, GPT-2、RoBERTa、XLNet
  - 2020, GPT-3、spanBERT, BART , T5
  - 2021, DeBERTa
  - 2022, PaLM

预训练语言模型

- 模型网络结构
  - 预训练语言模型采用了何种网络结构
  - BiLSTM (e.g. ELMo)
  - Transformer-E (e.g. BERT)
  - Transformer-D (e.g. GPT)
  - Transformer (e.g. BART)
- 预训练任务
  - 自回归语言模型 (e.g. ELMo, GPT)
  - 降噪自编码语言模型(e.g. BERT)
- 预训练模型应用(迁移)
  - 目标任务模型组件

概要

- 预训练语言模型概要
- ELMo
- GPT
- BERT
- BART

ELMo

- ELMo预训练语言模型提出的动机
  - Word2Vec、GloVe词向量，针对词型，无法反映词在当前语境中的意义
  - 不能有效应对一词多义现象，例如：bank, virus
  - 同一个词在不同的语境中，应该生成不同的词向量
- 语境敏感的词向量(contextualized word vector)
- 语境敏感的词向量应该反映两侧的语境信息
  - 基于BiLSTM捕获两侧的语境信息

ELMo

- ELMo通过建立自回归语言模型学习词向量
- 前向语言模型

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

- 后向语言模型

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

- 将二者组合，建立双向语言模型

ELMo

- 基于两个LSTM模型建立两个单向语言模型
  - 共享输入层词向量Θ_x
  - LSTM层参数不共享$\overrightarrow{\Theta}_{LSTM}$、$\overleftarrow{\Theta}_{LSTM}$
  - 输出层参数共享Θ_s

- 组合两个方向的似然函数，基于最大似然原则训练模型

$$\sum_{k=1}^{N}\Big(\log p(t_k|t_1,t_2,\dots,t_{k-1};\Theta_x,\overrightarrow{\Theta}_{LSTM},\Theta_s) + \log p(t_k|t_{k+1},t_{k+2},\dots,t_N;\Theta_x,\overleftarrow{\Theta}_{LSTM};\Theta_s)\Big)$$

ELMo

- 双向语言模型多层堆叠，设堆叠L层
- 对词例t_k而言，$2L+1$个词向量

$$R_k = \left\{ x_k^{LM}, \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} | j = 1,2,\dots,L \right\}$$

令：

$$h_{k,0}^{LM} = x_k^{LM}$$

$$h_{k,j}^{LM} = \left[\overrightarrow{h}_{k,j}^{LM}; \overleftarrow{h}_{k,j}^{LM} \right], j = 1,2,\dots,L$$

即：

$$R_k = \left\{ h_{k,j}^{LM} | j = 0,1,\dots,L \right\}$$

ELMo

- ELMo在目标任务中的应用
  - 通常作为Feature Extractor
  - 在目标任务中，将ELMo各层表示加权组合作为目标任务的输入特征，与目标任务中词向量拼接作为输入$[x_k; ELMo_k^{task}]$

$$ELMo_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}$$

- s_j^{task}是各层表示的权重参数，γ^{task}用于调整ELMo在任务模型中的比重，通过目标任务训练确定
- 在目标任务中，双向语言模型参数冻结，通常不再改变

ELMo

- small (total parameter=13.6M, 1024/128)
- medium (total parameter=28.0M, 2048/256)
- original (total parameter=93.6M, 4096/512)
- original(5.5B) (total parameter=93.6M, 4096/512)

概要

- 预训练语言模型概要
- ELMo
- GPT
- BERT
- BART

GPT

- GPT是自回归语言模型
- 构建单向语言模型

$$p(t_1, t_2, \dots, t_N) = \prod_{i=1}^{N} p(t_i | t_{i-k}, \dots, t_{i-1})$$

- 优化目标 对数似然函数

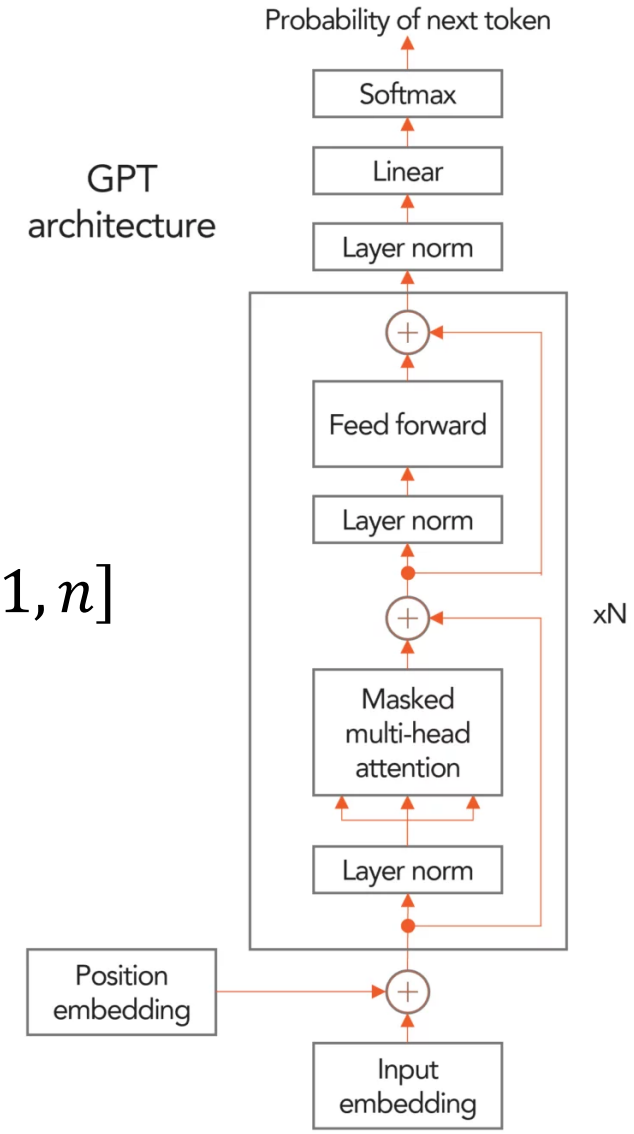
$$L_1(T) = \sum_i \log P(t_i | t_{i-k}, \cdots, t_{i-1}; \Theta)$$

GPT

- 基于Transformer解码器
- 去掉了交叉注意力子层
- Layer Normalization改用pre-LN

$$h_0 = UW_e + W_p$$
$$h_l = \text{transformer\_block}(h_{l-1}) \ \forall l \in [1, n]$$
$$P(u) = \text{softmax}(h_n W_e^{\top})$$

UW_e是词向量矩阵， W_p是位置向量矩阵



GPT

- GPT、GPT-2、GPT-3 原理相同
- GPT在目标任务中的应用
  - pre-training + fine tuning 范式
  - zero-/one-/few-shot learning

- 分类模型

$$P(y|x^1 \cdots x^m) = softmax(h_l^m W_y)$$

- 目标函数

$$L_2(C) = \sum_{(x,y)} \log P(y|x^1 \cdots x^m)$$

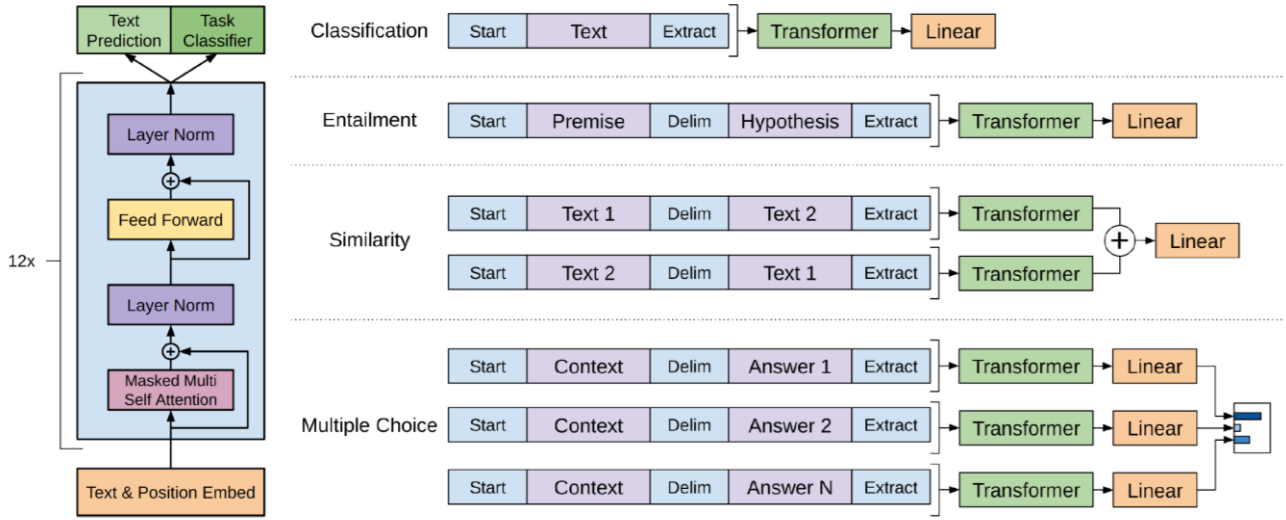
增加辅助任务-语言模型$L_1(C)$

$$L_3(C) = L_2(C) + \lambda \cdot L_1(C)$$

- 基于精调的序列分类模型
  - 序列分类问题$x^1 \cdots x^m \rightarrow y$
  - 将序列$x^1 \cdots x^m$输入预训练模型
  - 将最后一个token的输出作为序列表示
  - 基于序列表示进行线性分类
  - 预训练模型参数随着目标任务训练同步更新(精调)

GPT

- 目标任务(输入)形式多样，GPT的输入是序列形式
- 结构化输入转换成序列形式：通过添加特殊的token将输入转换成序列形式
  - 序列开始标记、序列分隔标记、序列结尾标记



两个序列可以互换生成两个序列

将题干和选项分别拼接形成多个序列

GPT

- GPT
  - L=12, H=768, A=12， total parameter=117M
- GPT-2
  - SMALL (L=12, H=768, total parameter=117M)
  - MEDIUM (L=24, H=1024, total parameter=345M)
  - LARGE (L=36, H=1280, total parameter=762M)
  - EXTRA-LARGE (L=48, H=1600, total parameter=1542M)
- GPT-3
  - 原文中说明有8个不同规模的模型
  - 最大的模型：L=96, H=12888, A=96, total parameters = 175B

概要

- 预训练语言模型概要
- ELMo
- GPT
- BERT
- BART

BERT

- GPT是单向语言模型
- BERT期望编码双向语境信息
- 除单序列处理任务之外，BERT也期望支持输入是一对序列的任务。
  - 单序列任务
    - 情感分析
    - …
  - 序列对任务
    - 文本蕴含判断
    - …

BERT

- BERT是降噪自编码语言模型
- 预训练任务1：训练 遮蔽语言模型(MLM)
  - 类似完形填空(cloze)
  - 随机选择15%词例进行遮蔽，训练模型基于左右语境复原
    - 在文本中植入噪音
    - 80%置换为[mask]，10%替换为其他词例，10%维持不变

训练语料 $D = \{x_0, x_1, \cdots, x_n, x_{n+1}\}$，遮蔽处理后记作$\widetilde{D}$

MLM训练目标

$$L(D) = \sum_{i=1}^{m} \log P([mask]_i = y_i | \widetilde{D}; \Theta)$$

BERT

- 预训练任务1：句子接续关系判断(NSP)
  - 输入一对句子，判断这两个句子在原始文本中是否相邻
  - 训练数据中，50%的句子对具有接续关系，50%的句子没有接续关系
  - 二分类任务(IsNext / NotNext)
- NSP任务的动机是适应以句子对作为输入的目标任务
  - 蕴含关系判断、自然语言推理(NLI)、问答(QA)等

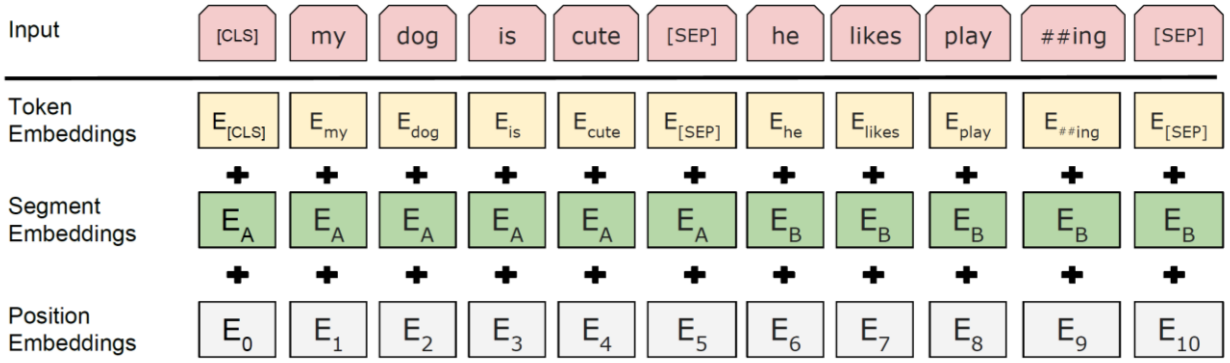
- MLM和NSP两个任务联合训练，最小化两个任务的组合损失

BERT

- BERT基于transformer编码器
- 序列输入格式：

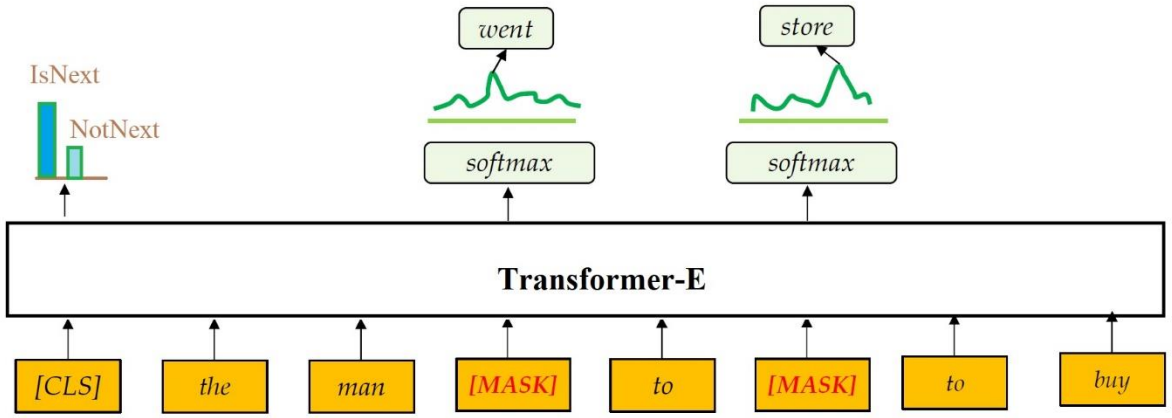
  [CLS]+segment A+[SEP]+segment B+[SEP]
  - 特殊token： [CLS]、[SEP]

- 输入向量组成
  - 词向量
  - segment编号
  - 位置向量



BERT

- 模型预训练架构

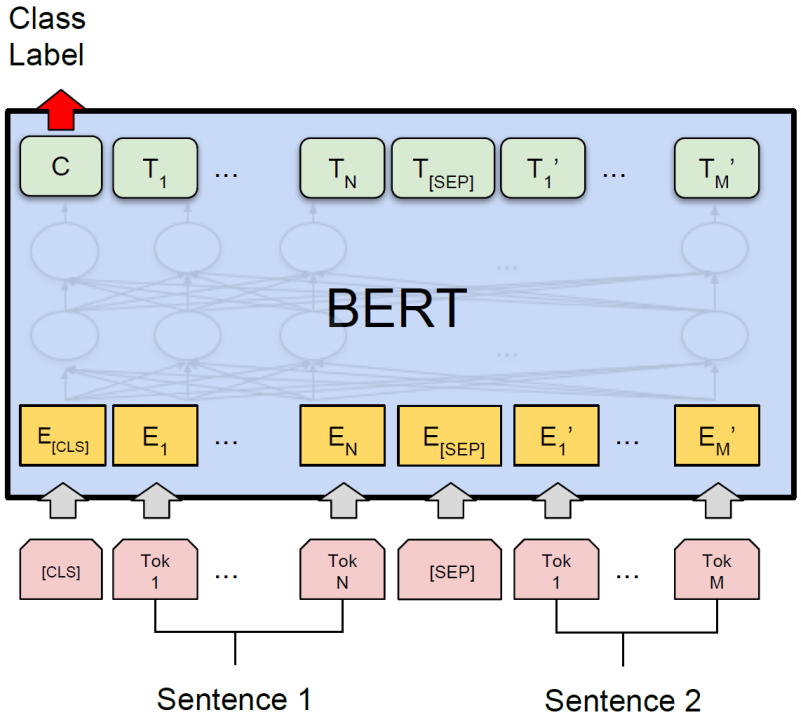


- 模型输出
  - $C \in \mathbb{R}^H$，词例[CLS]的表示向量
  - $T_i \in \mathbb{R}^H$，其他词例的表示向量

BERT

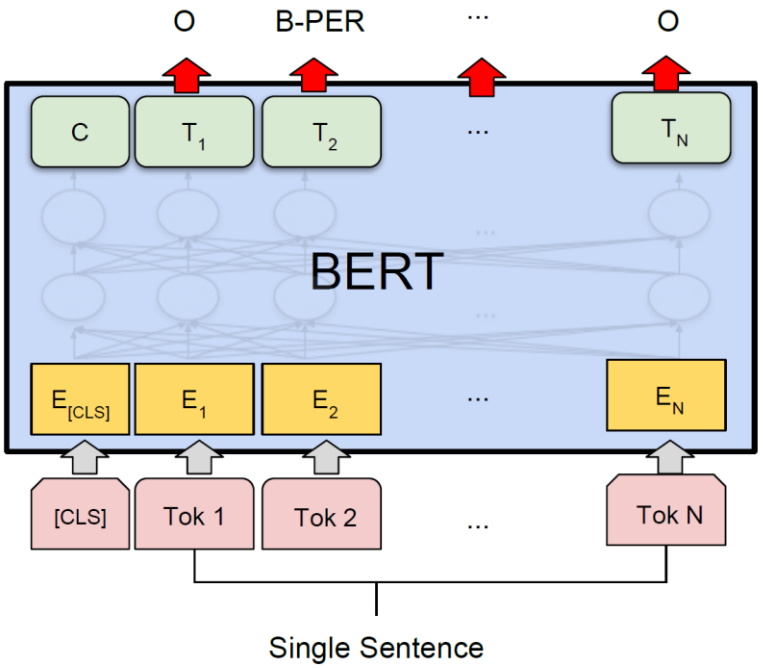
- BERT模型在目标任务中的应用
  - 遵循 pre-training + fine tuning的范式
- 只在BERT表示基础上，添加额外任务输出层
- 序列分类任务
  - 将[CLS]的向量表示C视作序列表示向量
  - 基于C添加分类层，输出类别分布向量 $softmax(CW^T)$
  - 序列对分类任务

    两个序列分别对应segment A和B



BERT

- 序列标注任务
  - 直接基于词例表示向量T_i预测标签分布

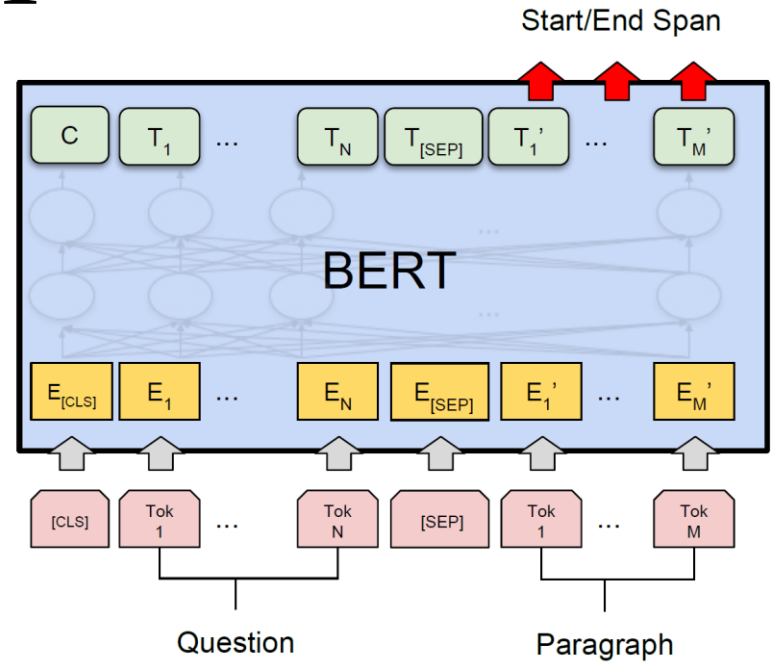


BERT

- (提取式)阅读理解任务

  给定Question
  给定阅读材料Paragraph
  在阅读材料中标记作为
  答案的span

- 基于词例表示T_i，计算该词例作为span开始和结尾的可能性



$$P_{i\ as\ start} = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$$

- 选择合法的得分高的span作为答案

$$\hat{s}_{i,j} = \max_{j>i}\left(S \cdot T_i + E \cdot T_j\right)$$

BERT

- 多项选择类任务
  - 抽象为序列化题干+若干序列化选项
  - 将题干和每个选项拼接成句子对，得到若干句子对
  - 将每个句子对输入BERT得到[CLS]表示
  - 基于CLS表示计算作为正确选择的分值
  - 选择得分最高的选项作为最终选项

$$softmax(S \cdot C_i)$$

On stage, a woman takes a seat at the piano. She

a) sits on a bench as her sister plays with the doll.
b) smiles with someone as the music plays.
c) is in the crowd, watching the dancers.
d) nervously sets her fingers on the keys.

BERT

- Base模型
  - 12层编码器
  - hidden size: 768
  - attention head: 12
  - 参数规模： 110M
- Large模型
  - 24层编码器
  - hidden size: 1024
  - attention head: 16
  - 参数规模：340M

BERT

- 应用较多的预训练语言模型
- 有很多针对性改进工作
  - 去掉NSP预训练任务
  - 采用动态遮蔽策略
  - 使用更长的输入序列
  - 采用更大batch size
  - 遮蔽连续词例span
  - …

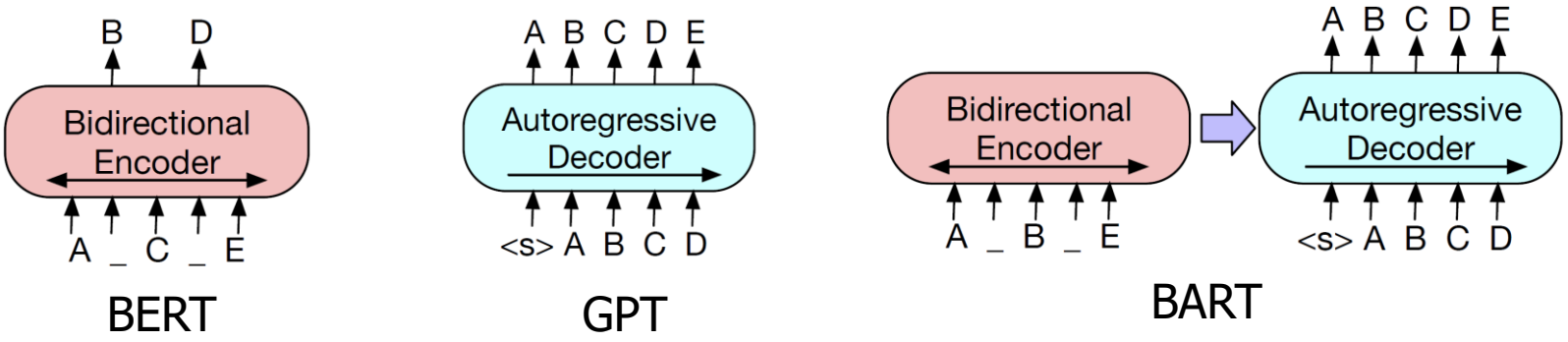
- RoBERTa、SpanBERT、ALBERT、DistilBERT、DeBERTa、……

概要

- 预训练语言模型概要
- ELMo
- GPT
- BERT
- BART

BART

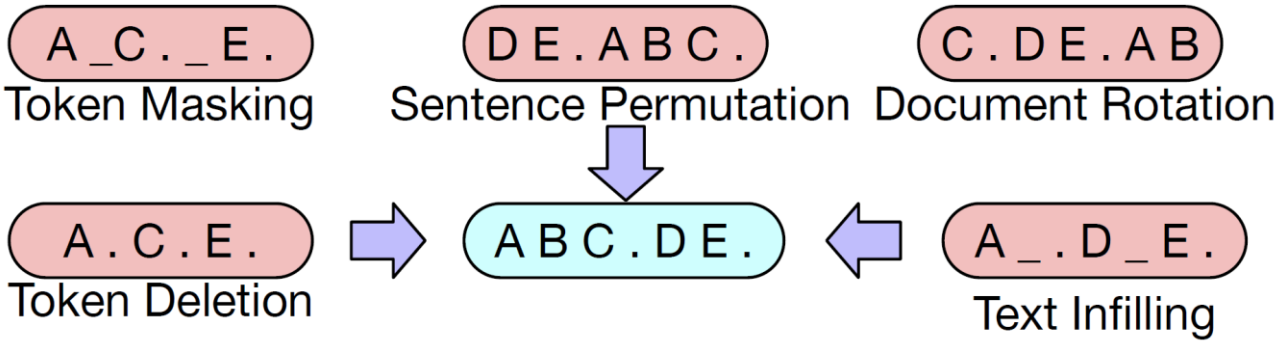
- BERT具有双向编码能力，长于自然语言理解(NLU)类任务
- GPT是自回归语言模型，长于自然语言生成(NLG)类任务



- BART兼具二者，综合降噪自编码与自回归生成两个任务
  - 基于编码器-解码器架构
  - 在编码器端对输入文本注入噪音(编码器具有双向编码能力)
  - 在解码器端以自回归生成方式重构文本

BART

- BART定义了5种预训练任务
  - 在编码器端以不同方式注入噪音，在解码器端重构输入



  - 优化目标函数：negative log likelihood of the original text

BART

- 预训练任务1：词例遮蔽(token masking)
  - 随机选择词例进行遮蔽，替换为[mask]

    A B C . D E . $\Longrightarrow$ A _ C . _ E .
- 预训练任务2：词例删除(token deletion)
  - 随机选择词例进行删除

    A B C . D E . $\Longrightarrow$ A C . E .
  - 重构时，需要确定删除词例的位置
- 预训练任务3：文本填充(text infilling)
  - 随机选择长度为$0 \leq l \leq L$的片段，替换为单个[mask]

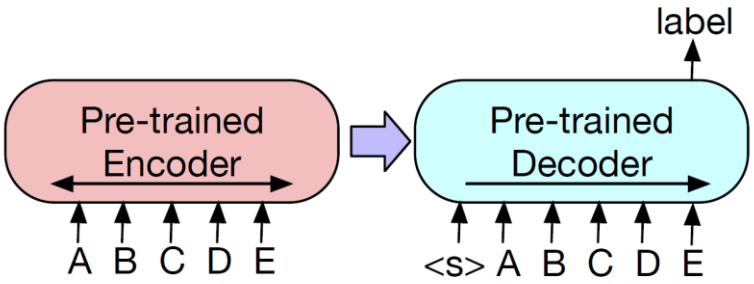
    A B C . D E . $\Longrightarrow$ A _ . D _ E .
  - 基于泊松分布确定span长度
  - 重构时，模型需要确定词例的数量

BART

- 预训练任务4：句子随机重排(sentence permutation)
  - 随机打乱句子的顺序
  - A B C . D E . ⟹ D E . A B C .
- 预训练任务5：文档旋转(document rotation)
  - 随机选择词例，旋转文档使该词例成为序列第一个词例
  - A B C . D E . ⟹ C . D E . A B
  - 重构时，模型需要判定哪个词例是文档第一个词例
- 基于完整的Transformer架构(encoder + decoder)
- 实验中效果较好的预训练任务
  - Text Infilling, Token Masking, Token Deletion
  - 最终使用的预训练任务 Text Infilling + Sentence Permutation

BART

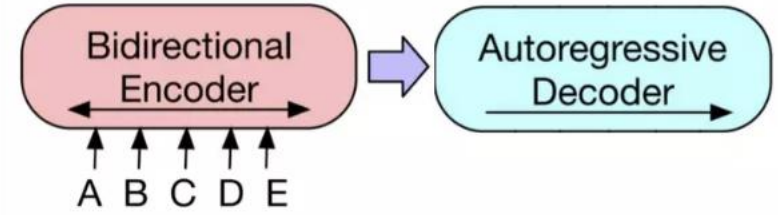
- BART在目标任务中的应用
  - 遵循 pre-training + fine tuning的范式
- 在BART表示基础上，添加额外任务输出层
- 序列分类任务
  - 将文本同时输入编码器和解码器
  - 解码器最后一个词例对应的表示向量作为序列表示向量
  - 增加分类层，基于序列表示向量进行分类



BART

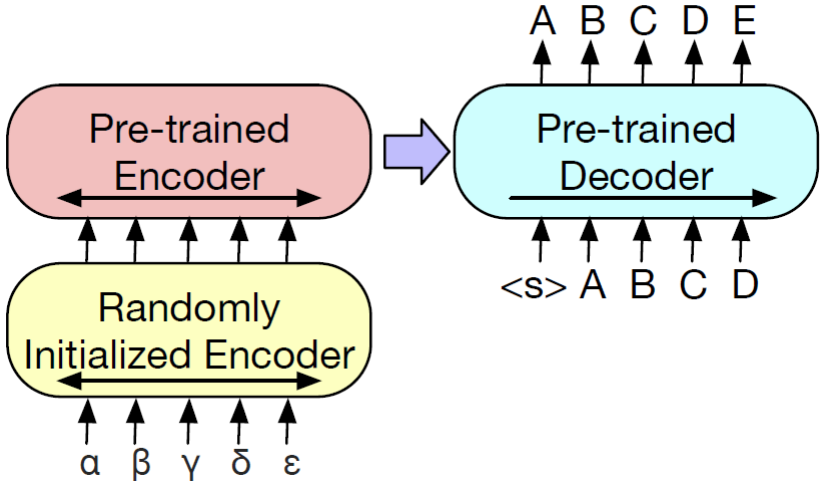
- 序列标注任务
  - 将文本同时输入编码器和解码器
  - 将解码器的输出作为每个词例的表示向量
  - 基于词例表示向量预测词例标签

- 序列生成任务
  - 将文本输入编码器(例如：原始文本)
  - 解码器以自回归方式生成输出文本(例如：摘要文本)



BART

- 机器翻译
  - 需要处理不同语言问题
  - 增加一个额外编码器，输入为源语言文本
  - 用这个额外编码器替换BART模型的输入层
  - fine-tuning分成两个阶段
  - 第一阶段冻结大部分BART参数，只更新新引入的编码器参数和BART模型的位置向量及第一层自注意力输入变换矩阵
  - 第二阶段更新全部参数



- 实验显示，在NLU任务上，BART模型表现与BERT相当，但在NLG任务上BART模型具有优势

BART

- Base模型
  - 6层encoder+6层decoder
  - hidden size: 768
  - attention head: 12
  - 参数规模：140M

- Large模型
  - 12层encoder+12层decoder
  - hidden size: 1024
  - attention head: 16
  - 参数规模: 400M