ERNIE Tutorial（论文笔记 + 实践指南）

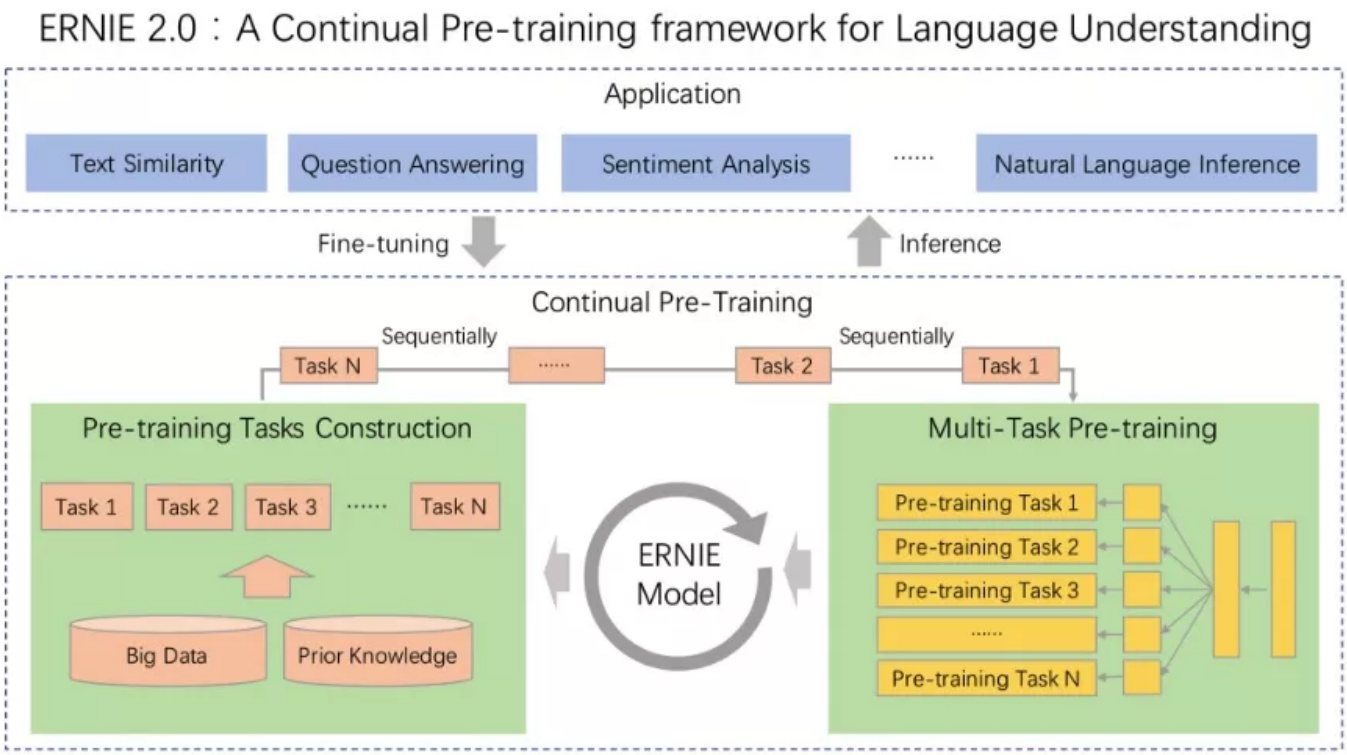
原创：太子長琴 AINLP 8月28日

作者：太子長琴（NLP算法工程师）

ERNIE 2.0 发布了，刷新了 SOTA，而且中文上做了不少优化，这种大杀器作为一个 NLP 工程师我觉得有必要深入了解了解，最好能想办法用到工作中。

> **ERNIE 2.0** 是基于持续学习的语义理解预训练框架，使用多任务学习增量式构建预训练任务。**ERNIE 2.0** 中，新构建的预训练任务类型可以无缝的加入训练框架，持续的进行语义理解学习。通过新增的实体预测、句子因果关系判断、文章句子结构重建等语义任务，**ERNIE 2.0** 语义理解预训练模型从训练数据中获取了词法、句法、语义等多个维度的自然语言信息，极大地增强了通用语义表示能力。

这是官方的描述，我觉得那张配图可能更加直观：



老习惯还是先理论再实践，所以看 paper 先，我们只看关键信息。

论文

ERNIE 核心思想：更多任务的 Bert。

Abstract

ERNIE 2.0 通过多任务学习获取语料中的更多信息，包括：词、句共现，词法，句法，语义等。

Introduction

出发点依然是摘要中提到的，举了几个例子挺不错：

- 命名实体可能包含一些观念上的信息（不应拆分或作为普通词处理）
- 句子顺序和邻近句子能让模型学到结构（句法）表示
- 文档语义相似度或句子间的话语关系能让模型学到语义表示

ERNIE 使用多任务学习，它们共享 encoding 网络（即词法、句法和语义编码信息可以在每项任务中使用），而且很容易定制训练任务（上图右下的 Pre-training Task 可以随意调整）。

Related Work

Unsupervised Transfer Learning for Language Representation

- context-independent
  - Word2Vec, GloVe: 通过词共现
- context-dependent
  - ELMo: Language model
  - OpenAI GPT: Transformer
  - BERT: Mask language model with a next sentence prediction task
  - XLM: Two methods to learn cross-lingual language model
  - MT-DNN: Learning several supervised tasks in GLUE based on pre-trained model
  - XLNet: Transformer-XL, learns bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order.

Continual Learning

在多个任务序列上训练模型，模型在学习新任务时能记起之前学过的任务。灵感来源于人类的学习过程。

The ERNIE 2.0 Framework

ERNIE 2.0 可以通过不断引入各种各样的预训练任务帮助模型高效学习词、句法和语义的表征。通过多任务学习不断更新预训练模型。在微调时，首先用预训练模型的参数初始化，然后根据特定任务的数据进行微调。

Continual Pre-training

包含两个步骤：

- 使用大规模数据和先验知识持续构建无监督预训练任务
  - 构建不同类型的任务：单词感知任务、结构感知任务、语义感知任务
  - 所有任务均为 self-supervised 或 weak-supervised，无需人工标注
- 通过多任务逐步更新 ERNIE 模型
  - 首先 train 一个简单的初始模型
  - 每增加一个任务，使用上一个模型的参数初始化，新模型用先前的模型训练

架构如下：

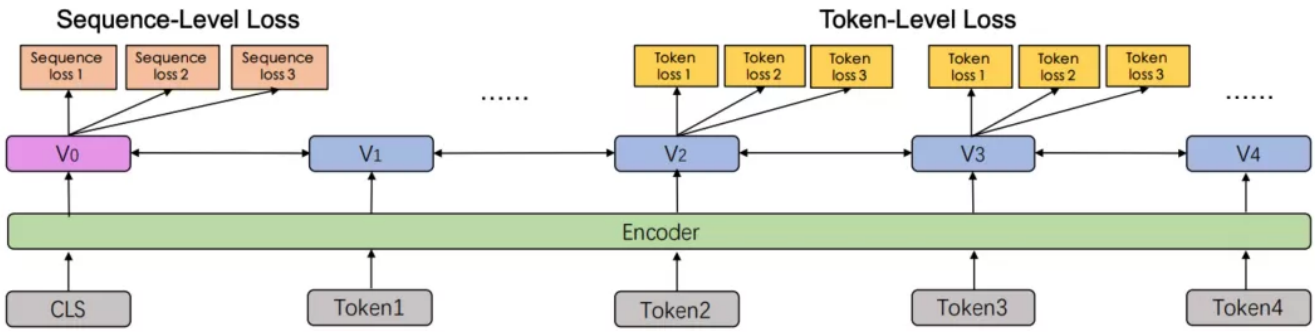


Figure 2: The architecture of multi-task pre-training in the ERNIE 2.0 framework, in which the encoder can be recurrent neural networks or a deep transformer.

- 共享的 encoder
- 两种 loss functions

Fine-tuning for Application Tasks

可以适应各种自然语言理解任务，比如：问答、推理、语义相似度等。

ERNIE 2.0 Model

Model Structure

- Transformer Encoder
  - 多层 Transformer
  - 使用 self-attention 捕捉每个 token 的上下文信息
  - Sequence 开始位置添加 CLS（分类任务）标记，多任务之间用 SEP 标记分割
- Task Embedding

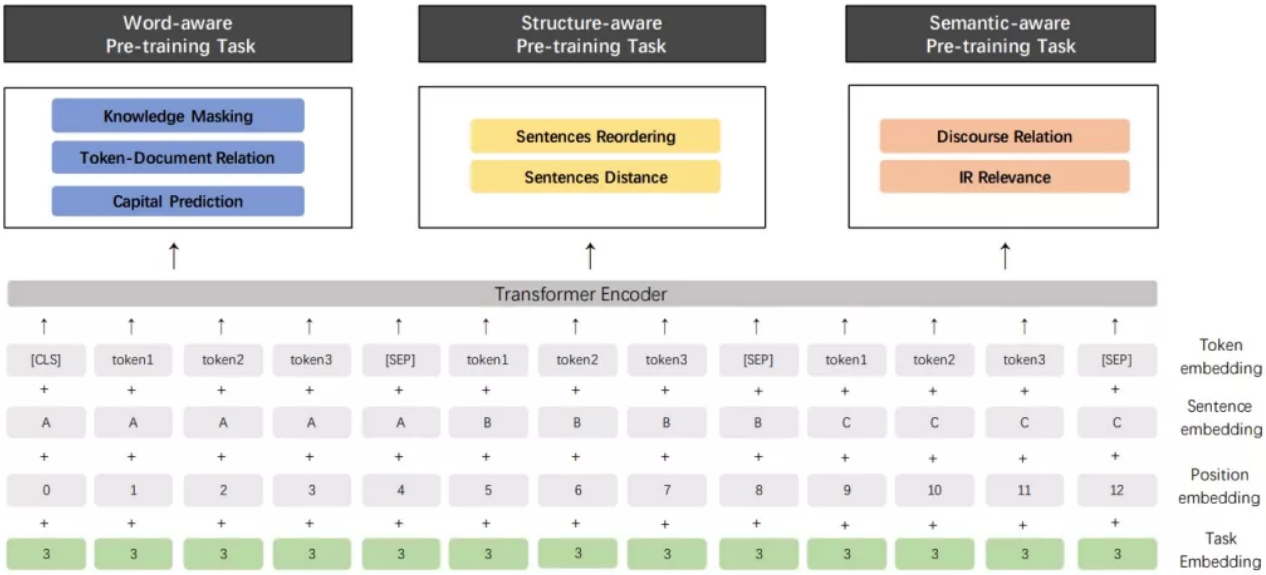


Figure 3: The structure of the ERNIE 2.0 model. The input embedding contains the token embedding, the sentence embedding, the position embedding and the task embedding. Seven pre-training tasks belonging to different kinds are constructed in the ERNIE 2.0 model.

Pre-training Tasks

- Word-aware Pre-training Tasks
  - Knowledge Masking Task: 短语和命名实体 mask，初始化模型。
  - Capitalization Prediction Task: 判断一个英文单词是否大写。
  - Token-Document Relation Prediction Task: 预测某段中的 token 是否出现在原文的其他部分。根据经验，出现在文档许多部分中的单词通常是常用单词或与文档主题相关。因此通过识别段中的文档关键词可以在一定程度上捕获文档的关键词。
- Structure-aware Pre-training Tasks
  - Sentence Reordering Task: 学习句子间关系。一个段落分成 m 小段后 random shuffle，然后让模型识别打乱的顺序（分类任务）。
  - Sentence Distance Task: 使用文档级别信息学习句子的距离。一个三分类任务：0 表示两个句子在一个文档中且相邻；1 表示在一个文档不相邻；2 表示不在一个文档。
- Semantic-aware Pre-training Tasks
  - Discourse Relation Task: 预测两个句子的语义或修辞关系。
  - IR Relevance Task: 学习 IR 中短文本相关性，一个三分类任务预测 query 和 title 之间的关系：0 表示强相关，用户输入 query 后点击了 title；1 表示弱相关，用户输入 query，title 在搜索结果中但用户未点击；2 表示不相关。

Experiments

主要与 Bert 和 XLNet 对比。

Pre-training and Implementation

- Pre-training Data

Corpus Type	English(#tokens)	Chinese(#tokens)
Encyclopedia	2021M	7378M
BookCorpus	805M	-
News	-	1478M
Dialog	4908M	522M
IR Relevance Data	-	4500M
Discourse Relation Data	171M	1110M

Table 1: The size of pre-training datasets.

- Pre-training Settings
  - base model: 12 layers, 12 self-attention heads and 768-dimensional of hidden size
  - large model: 24 layers, 16 self-attention heads and 1024-dimensional of hidden size
  - Adam optimizer: β1=0.9, β2=0.98
  - Batch size: 393216 tokens
  - Learning rate: 5e-5 (English model), 1.28e-4 (Chinese model)
  - Scheduled by decay scheme noam with warmup over the first 4000 steps for every task

Fine-tuning Tasks

- English

Corpus	Task	#Train	#Dev	#Test	#Label	Metrics
CoLA	Acceptability	8.5k	1k	1k	2	Matthews corr
SST-2	Sentiment	67k	872	1.8k	2	Accuracy
MNLI	NLI	393k	20k	20k	3	Accuracy
RTE	NLI	2.5k	276	3k	2	Accuracy
WNLI	NLI	634	71	146	2	Accuracy
QQP	Paraphrase	364k	40k	391k	2	Accuracy/F1
MRPC	Paraphrase	3.7k	408	1.7k	2	Accuracy/F1
STS-B	Similarity	7k	1.5k	1.4k	1	Pearson/Spearman corr
QNLI	QA/NLI	108k	5.7k	5.7k	2	Accuracy
AX	NLI	-	-	1.1k	3	Matthews corr

Table 2: The details of GLUE benchmark. The #Train, #Dev and #Test denote the size of the training set, development set and test set of corresponding corpus respectively. The #label denotes the size of the label set of the corresponding corpus.

- CoLA: 用于判断句子是否符合语法规范
- SST-2: 情感分析
- MNLI: 文本推理
- RTE: 自然语言推理
- WNLI: 段落间相互信息
- QQP: 判断问答对是否重复
- MRPC: 句子对语义相关性
- STS-B: 数据集，包括插图说明、新闻标题和论坛文字
- QNLI: 判断给定的文本对是否是 问题-答案对
- AX: 对模型进行语言分析

- Chinese Tasks

Corpus	Task	#Train	#Dev	#Test	#Label	Metrics
CMRC 2018	MRC	10K	3.2K	-	-	EM/F1
DRCD	MRC	27K	3.5K	3.5K	-	EM/F1
DuReader	MRC	271.5K	10K	-	-	EM/F1
MSRA-NER	NER	21K	2.3k	4.6K	7	F1
XNLI	NLI	392K	2.5K	2.5K	3	Accuracy
ChnSentiCorp	SA	9.6K	1.2K	1.2K	2	Accuracy
LCQMC	SS	240K	8.8K	12.5K	2	Accuracy
BQ Corpus	SS	100K	10K	10K	2	Accuracy
NLPCC-DBQA	QA	182K	41K	82K	2	mrr/F1

Table 3: The details of Chinese NLP datasets. The #Train, #Dev and #Test denote the size of the training set, development set and test set of corresponding corpus respectively. The #label denotes the size of the label set of the corresponding corpus.

- 阅读理解: CMRC 2018, DRCD, DuReader
- 命名实体识别: MSRA-NER (SIGHAN 2006)
- 自然语言推理: XNLI
- 情感分析: ChnSentiCorp
- 语义相似度: LCQMC, BQ Corpus
- 问答: NLPCC-DBQA

Implementation Details

Task	*BASE*			*LARGE*		
	Epoch	Learning Rate	Batch Size	Epoch	Learning Rate	Batch Size
CoLA	3	3e-5	64	5	3e-5	32
SST-2	4	2e-5	256	4	2e-5	64
MRPC	4	3e-5	32	4	3e-5	16
STS-B	3	5e-5	128	3	5e-5	128
QQP	3	3e-5	256	3	5e-5	256
MNLI	3	3e-5	512	3	3e-5	256
QNLI	4	2e-5	256	4	2e-5	256
RTE	4	2e-5	4	5	3e-5	16
WNLI	4	2e-5	8	4	2e-5	8

Table 4: The Experiment settings for GLUE dataset

实践

官方文档还是比较清晰的，我觉得对于大多数公司和个人来说，还是不要自己去训练了，实在太费钱了。不过如果是领域性很强的任务，还是可以重新训练一下的，要不然模型太 general。

Fine-tuning

现在来到 “使用” 部分，这个依赖百度的 Paddle，可以通过官方文档先安装好。目前 2.0 的中文模型还没有发布，不过估计用不了多久就有了。其实我们需要做的就是 Fine-tuning，也就是利用官方的预训练模型 + 我们自己特定任务的数据得到 “符合” 我们自己特定任务的模型。不同类型的任务参数不一样，可以根据自己的任务类型选择特定的参数，参数在上面提到的官方文档中可以找到。

官方给了三个常用的例子：

- 分类任务
- 序列标注任务
- 阅读理解任务

我们只需设置好 data 和 model 路径（可以用 autoenv，比较方便），然后按文档说明运行 scripts 下的脚本即可。友好提醒下，如果你电脑没有 GPU 的话就不要运行了，几乎肯定会死机（我已经试过了，成功运行了，然后只能强制重启了……）。当然配置比较渣（Mac Pro 2015）：

- 2.7 GHz Intel Core i5
- 8GB 1867 MHz DDR3
- Intel Iris Graphics 6100 1536 MB

Fine-tuning 方面没啥特别的，只要你的输入和给的数据集一样即可。另外，官方文档还给出了获取特定任务句子和词 Embedding 的代码，可参考 FAQ1。

模型训练好后需要进行预测，官方也给出了 `predict_classifier.py`，直接使用即可，使用方式可以参见文档的 FAQ2，感觉稍微弄弄就可以直接 deploy 的节奏。

Pre-training

刚刚说了，如果领域性比较强的话还是建议 pre-train 一下的，这个也很方便，我们只需要把数据处理成 ERNIE 要的格式即可（每行训练数据包括 5 个分号分割的字段）：

- token_ids: 输入句子对 token 的 id
- sentence_type_ids: 输入句子对标签，0 和 1
- position_ids: 输入句子对 token 的位置
- seg_labels: 分词边界信息，0 表示词首、1 表示非词首、-1 为占位符
- next_sentence_label: 输入的句子对是否存在上下句关系，0 表示无 1 表示有

然后将 scripts 下脚本中的数据集换成自己的即可开始训练。

小结

目前官方给出的代码其实是 v1.0 的版本，所以 pre-train 的也是 1.0 版本，1.0 只有两个任务，对应：lm_loss 和 next_sent_loss，2.0 中文版会有更多的 loss。源代码也大概看了下，涉及很多不太熟悉的 Paddle，重点看了输入和 loss，感觉比较重要的就是 mask 和 negtive samples 的代码，分别在 `batching.py` 和 `pretraining.py` 中。当然，实际应用角度来看，其实我们只要搞定推理部分即可，即使要自己训练，也可以直接用官方给出的参数，调参还是交给有钱的大公司吧 :D

再次感慨下 Pre-training 在 NLP 领域的如火如荼，有钱的巨头们不断砸出更好的 model，我等吃瓜群众默默拿来使用，顺便给他们点个赞。最后说句题外话，看了下 ERNIE 的 Issue，我觉得官方应该给一个 How-To-Ask-Questions-The-Smart-Way：）

参考

- ERNIE/README.zh.md at develop · PaddlePaddle/ERNIE
- [1907.12412v1] ERNIE 2.0: A Continual Pre-training Framework for Language Understanding

原文链接：

https://yam.gift/2019/08/02/Paper/2019-08-02-Baidu-ERNIE-Tutorial/