[预训练语言模型专题] ENRIE(Tsinghua)：知识图谱与BERT相结合，为语言模型赋能助力

原创 管扬 朴素人工智能

来自专辑

预训练语言模型

本文为预训练语言模型专题的第**18**篇。

快速传送门

1-4:[萌芽时代]、[风起云涌]、[文本分类通用技巧] 、 [GPT家族]

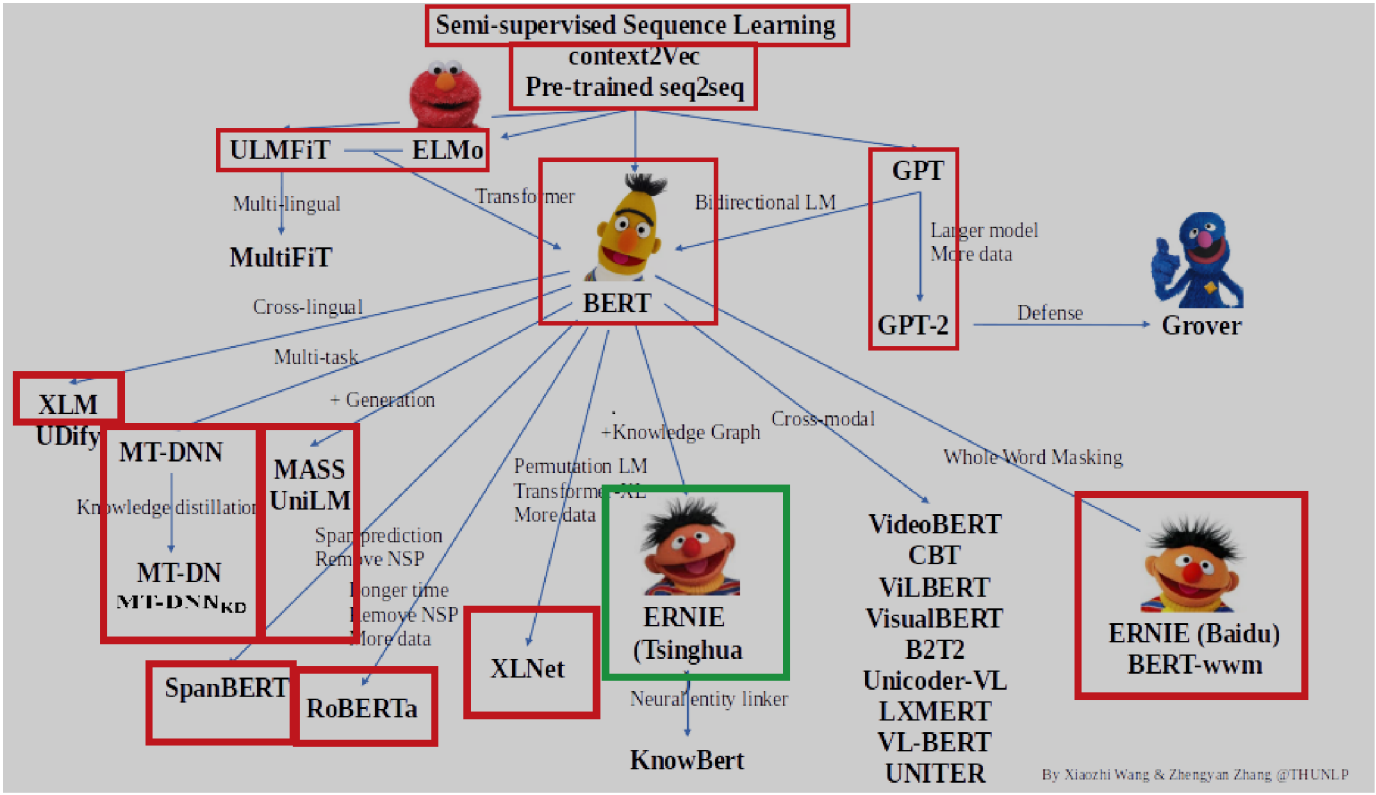
5-8:[BERT来临]、[浅析BERT代码]、[ERNIE合集]、[MT-DNN(KD)]

9-12:[Transformer]、[Transformer-XL]、[UniLM]、[Mass-Bart]

13-16：[跨语种模型]、[XLNet]，[RoBERTa]、[SpanBERT]

17：[跨模态语言模型]

感谢清华大学自然语言处理实验室对**预训练语言模型**架构的梳理，我们将沿此脉络前行，探索预训练语言模型的前沿技术，红框中为已介绍的文章，绿框中为本期介绍的模型，欢迎大家留言讨论交流。



ERNIE: Enhanced Language Representation with Informative Entities（2019）

在之前的一期推送中，我们给大家介绍过百度的ERNIE。其实清华大学NLP实验室，比百度更早一点，也发表了名为ERNIE的模型，即Enhanced Language Representation with Informative Entities。

他们认为现存的预训练语言模型很少会考虑与知识图谱（Knowledge Graph: KG）相结合，但其实知识图谱可以提供非常丰富的结构化知识和常识以供更好的语言理解。他们觉得这其实是很有意义的，可以通过外部的知识来强化语言模型的表示能力。在这篇文章中，他们使用大规模语料的语言模型预训练与知识图谱相结合，更好地利用语义，句法，知识等各方面的信息，推出了Enhanced language representation model（ERNIE），在许多知识驱动的任务上获得了巨大提升，而且更适用于广泛通用的NLP任务。

作者提出，要将知识嵌入到自然语言模型表达中去，有两个关键的挑战：

1. 知识的结构化编码

   对于一个给定的文本，如何从知识图谱中，高效地将和文本相关的常识或知识抽取出来并编码是一个重要问题。

2. 异构信息融合

   语言模型表示的形式和知识图谱的表达形式是大不相同的，是两个独立的向量空间。怎么样去设计一个独特的训练任务来将，语义，句法，知识等信息融合起来是另一个挑战。

针对这些挑战， 清华NLP实验室提出方案是 **E**nhanced
Language **R**epresentatio**N** with **I**nformative **E**ntities (ERNIE)

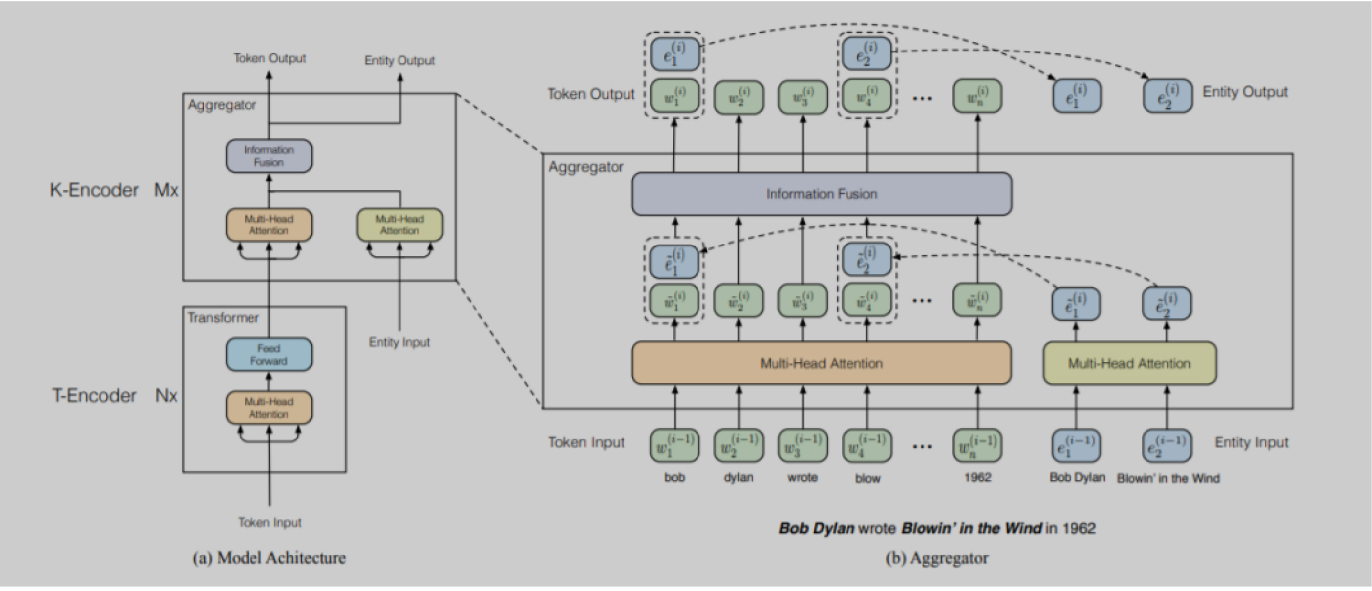
- 首先，通过识别文本中的命名实体，然后将其链指到知识图谱中的相应实体上，进行知识的抽取和编码。相比于直接使用知识图谱中基于图结构的信息，作者通过TranE这样的知识嵌入算法，对知识图谱的图结构实体进行编码，然后将这富有信息的实体表示作为ERNIE的输入，这样就可以把知识模块中的实体的信息表示，引入到模型下层的语义表示中去。
- 其次，和BERT类似，采用了MLM和NSP的预训练目标。除此以外，为了更好地融合文本信息和知识信息，设计了一个新的预训练目标，通过随机地mask一些命名实体，同时要求模型去知识图谱中寻找合适的实体，来填充被mask掉的部分。这个训练目标这样做就可以要求语言模型，同时利用文本信息和知识图谱来对token-entity进行预测，从而成为一个富有知识的语言表达模型。

本文在两个知识驱动的NLP任务entity typing 和 relation classification进行了实验，ENRIE在这两个任务上的效果大幅超越BERT，因为其充分利用了语义，句法和知识信息。在其他的NLP任务上，ENRIE的效果也很不错。

定义

首先，定义我们的文本token序列为{w1, . . . , wn}，n为token序列的长度。同时，输入的token可以在KG中对应entity。所对应entity的序列为{e1, . . . , em}， m是序列中entity的数量。因为不一定每一个token都对应得到KG中的一个entity，所以在大多数情况下m不等于n。所有token的集合也就是字典为V，在KG中所有entity的列表为E。如果，某个在V中的token $w \in V$ 在KG中有对应的entity $e \in E$。那么这个对应关系定义为f(w) = e

我们可以看下方的模型结构图，大概包括两个模块。



1. 下层的文本编码器（T-Encoder），负责捕捉基本的词法和句法的信息，其与BERT的encoder实现是相同的，都是多层的Transformer，层数为N。

2. 上方的知识编码器（K-Encoder），负责将跟entity相关的知识信息融入到下方层传来的文本编码信息中，两者可以在统一的特征空间中去表示。T-Encoder的输出是{w1, . . . , wn}，实体输入通过TranE得到的知识嵌入为{e1, . . . , em}。两者通过K-Encoder计算出对应的特征以实现特定任务。

$$\{\boldsymbol{w}_1^o, \ldots, \boldsymbol{w}_n^o\}, \{\boldsymbol{e}_1^o, \ldots, \boldsymbol{e}_n^o\} = \text{K-Encoder}(\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}, \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m\}).$$

K-Encoder的结构和BERT略微不同，包含M个stacked aggregators。首先对token的输出和entity的embedding通过两个多头自注意力进行self attention。

$$\{\tilde{\boldsymbol{w}}_1^{(i)}, \ldots, \tilde{\boldsymbol{w}}_n^{(i)}\} = \text{MH-ATT}(\{\boldsymbol{w}_1^{(i-1)}, \ldots, \boldsymbol{w}_n^{(i-1)}\}),$$
$$\{\tilde{\boldsymbol{e}}_1^{(i)}, \ldots, \tilde{\boldsymbol{e}}_m^{(i)}\} = \text{MH-ATT}(\{\boldsymbol{e}_1^{(i-1)}, \ldots, \boldsymbol{e}_m^{(i-1)}\}).$$

接着，通过以下的式子进行两者的结合。Wt和We分别是token和Embedding的attention权重矩阵。

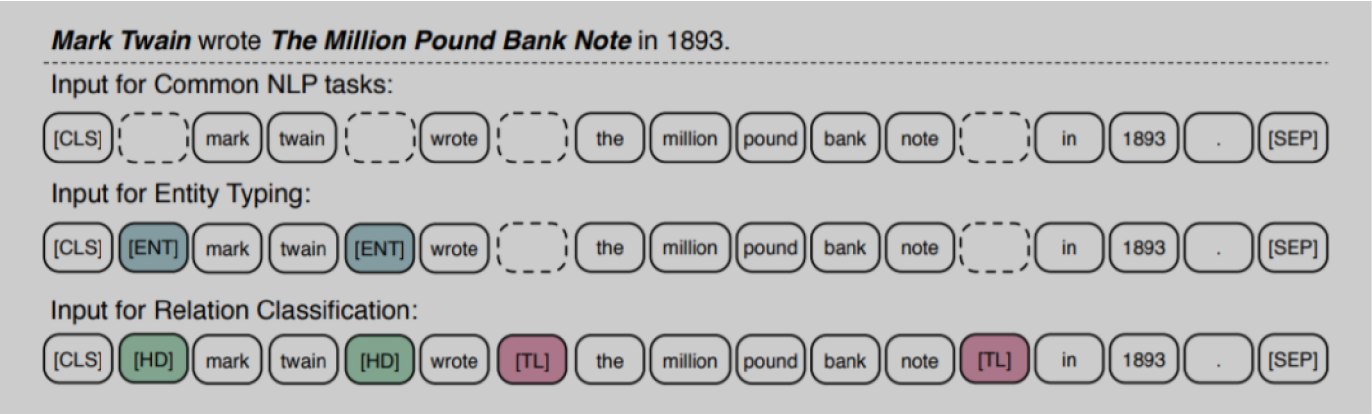
$$\boldsymbol{h}_j = \sigma(\tilde{\boldsymbol{W}}_t^{(i)} \tilde{\boldsymbol{w}}_j^{(i)} + \tilde{\boldsymbol{W}}_e^{(i)} \tilde{\boldsymbol{e}}_k^{(i)} + \tilde{\boldsymbol{b}}^{(i)}),$$
$$\boldsymbol{w}_j^{(i)} = \sigma(\boldsymbol{W}_t^{(i)} \boldsymbol{h}_j + \boldsymbol{b}_t^{(i)}),$$
$$\boldsymbol{e}_k^{(i)} = \sigma(\boldsymbol{W}_e^{(i)} \boldsymbol{h}_j + \boldsymbol{b}_e^{(i)}).$$

Pre-training for Injecting Knowledge

除了结构的改变以外，文章提出了特殊的预训练语言模型训练目标。通过随机地mask一些entity然后要求模型通过知识图谱中实体来进行选择预测，起名为denoising entity auto-encoder（dEA）。由于知识图谱中entity的数量规模相对softmax层太大了，会首先在KG中进行筛选找到相关的entity。有时候token和entity可能没有正确的对应，就需要采取一些措施。

- 5%的情况下，会将token对应的entity替换成一个随机的entity，这是让模型能够在align错的时候，能够纠正过来。
- 15%的情况下，会将entity mask掉，纠正没有把所有存在的entity抽取出来和entity进行对应的问题。
- 其余的情况下，保持token-entity alignments 不变，来将entity的表示融合进token的表示，以获得更好的语言理解能力。

Fine-tuning for Specific Tasks



对于大量普通的NLP任务来说，ERNIE可以采取和BERT相似的finetune策略，将[CLS]的输出作为输入文本序列的表示。对于一些知识驱动的任务，我们设计了特殊的finetune流程。

对于关系分类任务，任务要求模型根据上下文，对给定的entity对的关系进行分类。本文设计了特殊的方法，通过加入两种mark token来高亮实体。[HD] 表示head entity， [TL]表示tail entity。

对于实体类别分类任务，finetune的方式是关系分类的简化版，通过[ENT]标示出entity的位置，指引模型同时结合上下文和实体的信息来进行判断。

模型细节

从头开始训ENRIE的代价太大了，所以模型用了BERT的参数初始化。利用英文WIKI作为语料，和WiKidata进行对应，语料中包含大约4500M个subwords，和140M个entities，将句中小于三个实体的样本丢弃。通过TranE算法在WiKidata上训练entity的embedding。使用了部分WiKidata，其中包含5040986个实体和24267796个三元组。

模型尺度上来说，T-encoder的层数N为6，K-encoder层数M为6。隐藏层维度两个网络分别Hw = 768, He = 100。Attention的头数分别 Aw = 12, Ae = 4。总参数量大约114M。

ERNIE仅在语料上训练了一轮，最大文本长度由于速度原因设为256，batch-size为512。除了学习率为5e-5，其他参数和BERT几乎一样。

实验效果

直接放图吧，比当时的state-of-the-art ： BERT在很多任务上都提升了不少。

Model	Acc.	Macro	Micro
NFGEC (Attentive)	54.53	74.76	71.58
NFGEC (LSTM)	55.60	75.15	71.73
BERT	52.04	75.16	71.63
ERNIE	**57.19**	**76.51**	**73.39**

Table 2: Results of various models on FIGER (%).

Model	FewRel			TACRED		
	P	R	F1	P	R	F1
CNN	69.51	69.64	69.35	70.30	54.20	61.20
PA-LSTM	-	-	-	65.70	64.50	65.10
C-GCN	-	-	-	69.90	63.30	66.40
BERT	85.05	85.11	84.89	67.23	64.81	66.00
ERNIE	88.49	88.44	**88.32**	69.97	66.08	**67.97**

Table 5: Results of various models on FewRel and TACRED (%).

这里作者认识到，有了知识图谱的介入，可以用更少的数据达到更好的效果。

Model	MNLI-(m/mm) 392k	QQP 363k	QNLI 104k	SST-2 67k
$BERT_{BASE}$	84.6/83.4	71.2	-	93.5
ERNIE	84.0/83.2	71.2	91.3	93.5
Model	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k
$BERT_{BASE}$	52.1	85.8	88.9	66.4
ERNIE	52.3	83.2	88.2	68.8

Table 6: Results of BERT and ERNIE on different tasks of GLUE (%).

结论

在文中提出了一种方法名为ERNIE，来将知识的信息融入到语言表达的模型中。具体地，提出了knowledgeable aggregator 和预训练任务dEA来更好地结合文本和知识图谱两个异构的信息源。实验表明，ENRIE能更好地在有限的数据上进行训练和泛化。

未来还有几个重要的方向值得研究

1. 将知识嵌入到基于特征的预训练语言模型如ELMo。
2. 引入更多不同的结构化知识进入到语言表达模型中去，比如ConceptNet，这和WiKidata是完全不同的方式。
3. 进行真实世界更广泛的语料收集，可以进行更通用和有效的预训练

未完待续

本期的论文就给大家分享到这里，感谢大家的阅读和支持，下期我们会给大家带来其他预训练语言模型的介绍，敬请大家期待！

欢迎关注朴素人工智能，这里有很多最新最热的论文阅读分享，有问题或建议可以在公众号下留言，也可以点击下方**原文链接**，去我们的知乎专栏哦！

往期推荐

- [预训练语言模型专题] RoBERTa: 捍卫BERT的尊严
- [预训练语言模型专题] Transformer-XL 超长上下文注意力模型
- [预训练语言模型专题] XLNet：公平一战！多项任务效果超越BERT
- [预训练语言模型专题] Huggingface简介及BERT代码浅析
- [预训练语言模型专题] SpanBERT: 抽取式问答的利器

阅读原文