

论文解读:BERT模型及fine-tuning

 **习翔宇** 
北京大学 软件工程博士在读

关注他

43 人赞了该文章

在上周BERT这篇论文[5]放出来引起了NLP领域很大的反响，很多人认为是改变了游戏规则的工作，该模型采用BERT + fine-tuning的方法，在11项NLP tasks中取得了state-of-the-art的结果，包括NER、问答等领域的任务。本文对该论文进行介绍。

1. 现有的Language Model Embedding

语言模型来辅助NLP任务已经得到了学术界较为广泛的探讨，通常有两种方式：

- 1. feature-based
- 2. fine-tuning

1.1 Feature-based方法

Feature-based指利用语言模型的中间结果也就是LM embedding, 将其作为额外的特征，引入到原任务的模型中，例如在TagLM[1]中，采用了两个单向RNN构成的语言模型，将语言模型的中间结果

引入到序列标注模型中，如下图1所示，其中左边部分为序列标注模型，也就是task-specific model，每个任务可能不同，右边是前向LM(Left-to-right)和后向LM(Right-To-Left)，两个LM的结果进行了合并，并将LM embedding与词向量、第一层RNN输出、第二层RNN输出进行了concat操作。

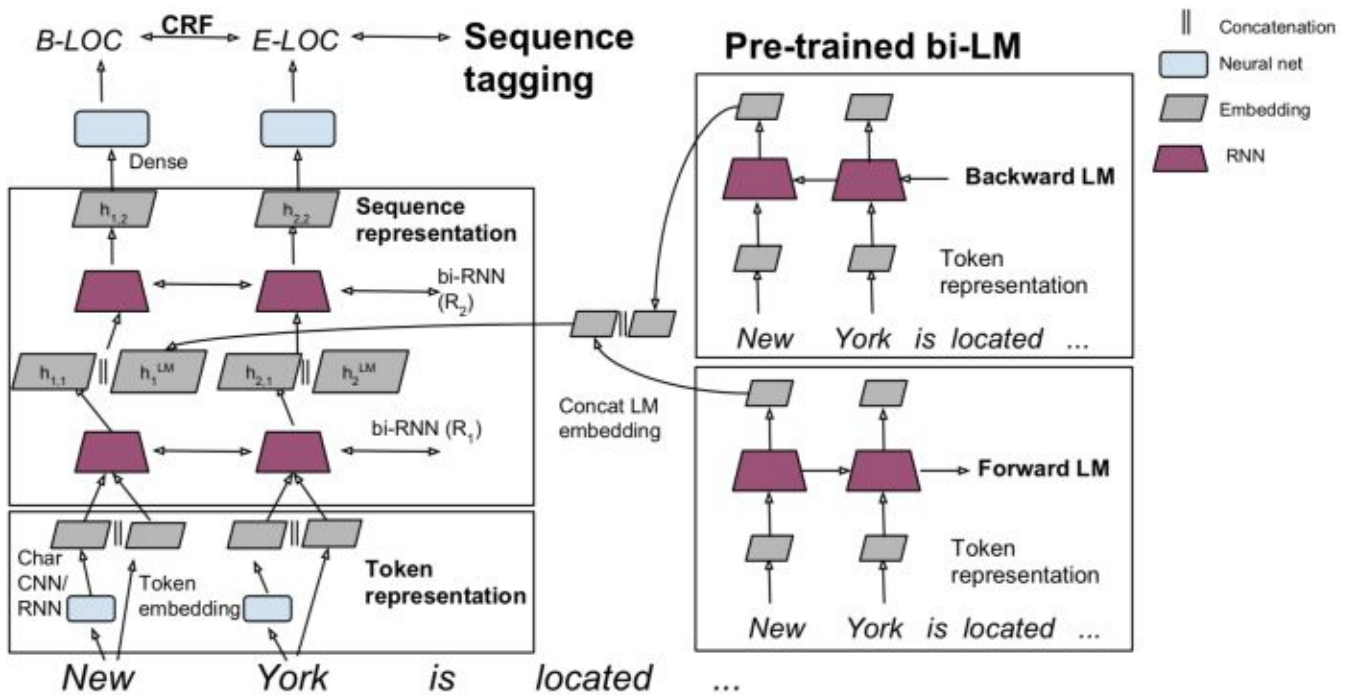


图1 TagLM模型示意图

通常feature-based方法包括两步：

1. 首先在大的语料A上无监督地训练语言模型，训练完毕得到语言模型
2. 然后构造task-specific model例如序列标注模型，采用有标记的语料B来有监督地训练task-specific model，将语言模型的参数固定，语料B的训练数据经过语言模型得到LM embedding，作为task-specific model的额外特征

ELMo是这方面的典型工作，请参考[2]

1.2 Fine-tuning方法

Fine-tuning方式是指在已经训练好的语言模型的基础上，加入少量的task-specific parameters，例如对于分类问题在语言模型基础上加一层softmax网络，然后在新的语料上重新训练来进行fine-tune。

例如OpenAI GPT [3] 中采用了这样的方法，模型如下所示

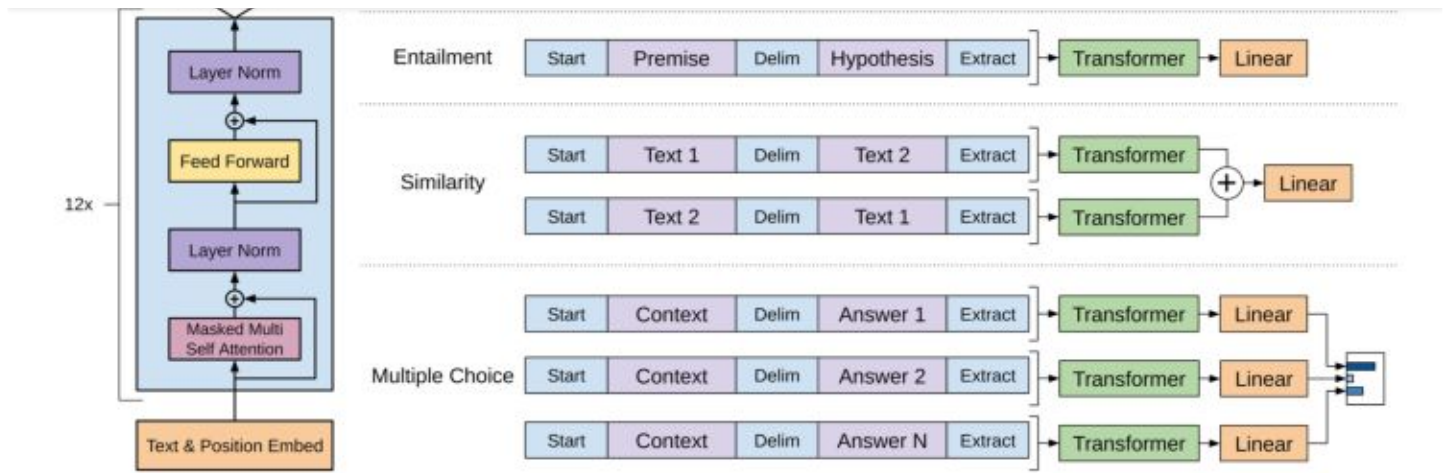


图2 Transformer LM + fine-tuning模型示意图

首先语言模型采用了Transformer Decoder的方法来进行训练，采用文本预测作为语言模型训练任务，训练完毕之后，加一层Linear Project来完成分类/相似度计算等NLP任务。因此总结来说，LM + Fine-Tuning的方法工作包括两步：

- 1. 构造语言模型，采用大的语料A来训练语言模型
- 2. 在语言模型基础上增加少量神经网络层来完成specific task例如序列标注、分类等，然后采用有标记的语料B来有监督地训练模型，这个过程中语言模型的参数并不固定，依然是trainable variables.

而BERT论文采用了LM + fine-tuning的方法，同时也讨论了BERT + task-specific model的方法。

2. BERT模型介绍

BERT采用了Transformer Encoder的模型来作为语言模型，Transformer模型来自于论文[4], 完全抛弃了RNN/CNN等结构，而完全采用Attention机制来进行input-output之间关系的计算，如下图中左半边部分所示，其中模型包括两个sublayer：

- 1. Multi-Head Attention 来做模型对输入的Self-Attention
- 2. Feed Forward 部分来对attention计算后的输入进行变换



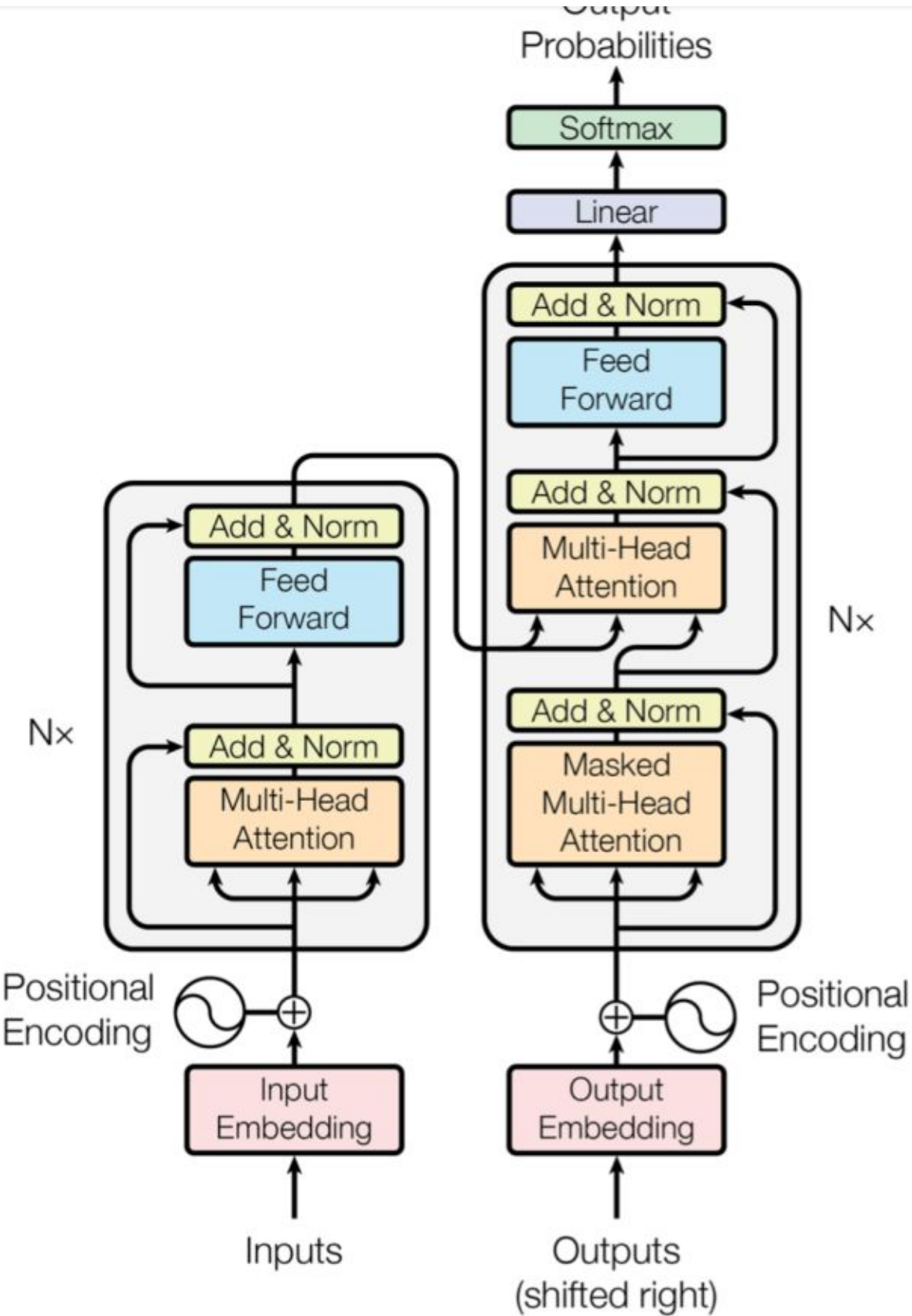
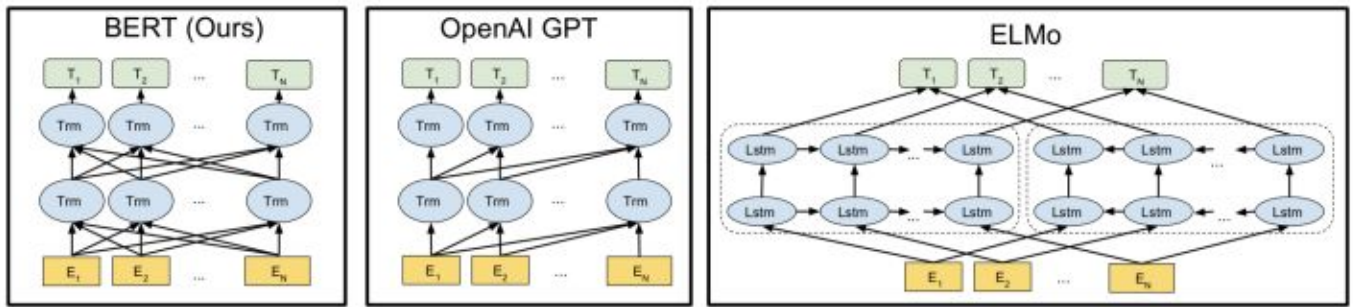


Figure 1: The Transformer - model architecture.

BERT模型如下图中左边第一个所示，它与OpenAI GPT的区别就在于采用了Transformer Encoder，也就是每个时刻的Attention计算都能够得到全部时刻的输入，而OpenAI GPT采用



下面我们介绍BERT的Pre-training tasks, 这里为了能够有利于token-level tasks例如序列标注, 同时有利于sentence-level tasks例如问答, 采用了两个预训练任务分别是

1. Masked Language Model
2. Next Sentence Prediction

2.1 Masked Language Model

现有的语言模型的问题在于, 没有同时利用到Bidirectional信息, 现有的语言模型例如ELMo号称是双向LM(BiLM), 但是实际上是两个单向RNN构成的语言模型的拼接, 如下图所示

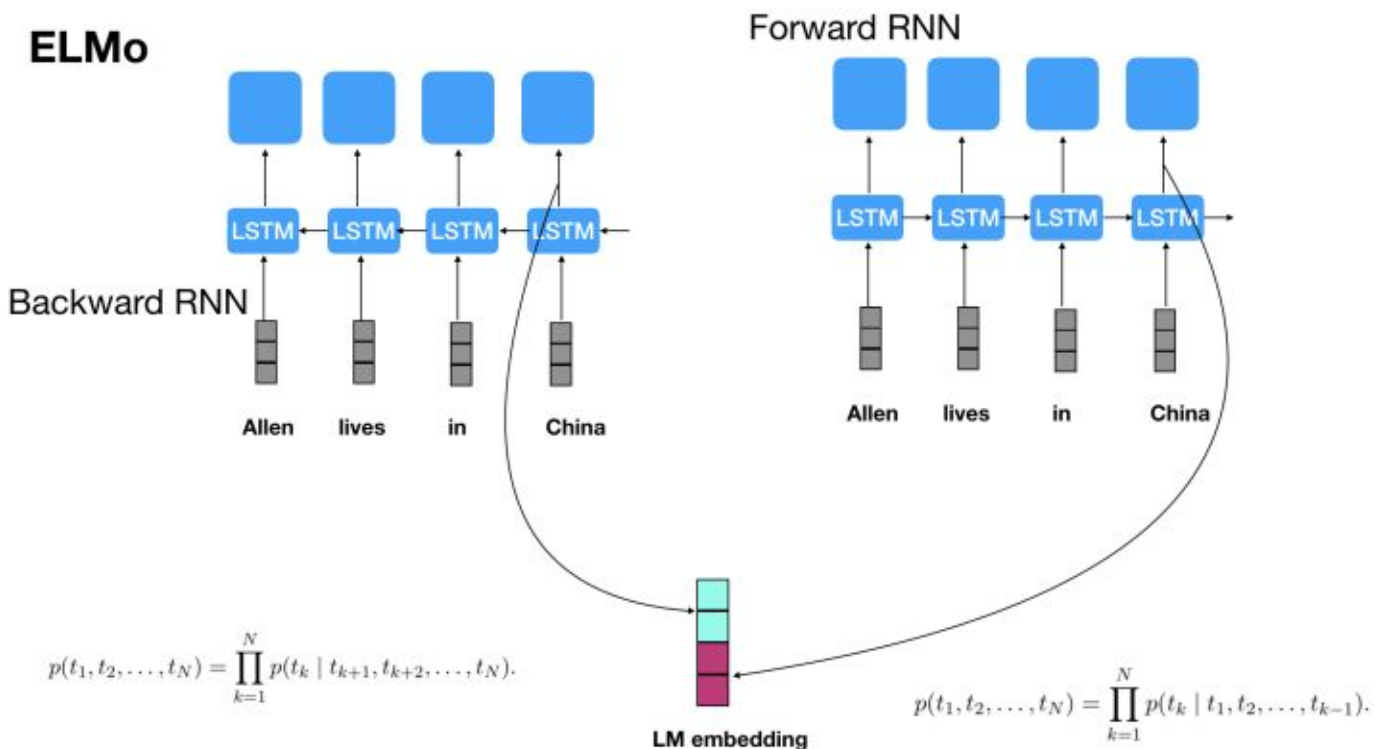


图3 ELMo模型示意图

因为语言模型本身的定义是计算句子的概率：

知乎

首发于
机器学习

$$- \prod_{i=1}^m p(w_i | w_1, w_2, \dots, w_{i-1})$$

前向RNN构成的语言模型计算的是：

$$p(w_1, w_2, w_3, \dots, w_m) = \prod_{i=1}^m p(w_i | w_1, w_2, \dots, w_{i-1}) \quad (3)$$

也就是当前词的概率只依赖前面出现词的概率。

而后向RNN构成的语言模型计算的是：

$$p(w_1, w_2, w_3, \dots, w_m) = \prod_{i=1}^m p(w_i | w_{i+1}, w_{i+2}, \dots, w_m) \quad (4)$$

也就是当前词的概率只依赖后面出现的词的概率。

那么如何才能同时利用好前面词和后面词的概率呢？BERT提出了Masked Language Model，也就是随机去掉句子中的部分token，然后模型来预测被去掉的token是什么。这样实际上已经不是传统的神经网络语言模型(类似于生成模型)了，而是单纯作为分类问题，根据这个时刻的hidden state来预测这个时刻的token应该是什么，而不是预测下一个时刻的词的概率分布了。

这里的操作是随机mask语料中15%的token，然后预测masked token，那么masked token 位置输出的final hidden vectors喂给softmax网络即可得到masked token的预测结果。

这样操作存在一个问题，fine-tuning的时候没有[MASK] token，因此存在pre-training和fine-tuning之间的mismatch，为了解决这个问题，采用了下面的策略：

- 80%的时间中：将选中的词用[MASK]token来代替，例如

my dog is hairy → my dog is [MASK]

- 10%的时间中：将选中的词用任意的词来进行代替，例如

my dog is hairy → my dog is apple

- 10%的时间中：选中的词不发生变化，例如

my dog is hairy → my dog is hairy

这样存在另一个问题在于在训练过程中只有15%的token被预测，正常的语言模型实际上是预测每个token的，因此Masked LM相比正常LM会收敛地慢一些，后面的实验也的确证实了这一点



知乎

首发于
机器学习

很多需要解决的NLP tasks依赖于句子间的关系，例如问答任务等，这个关系语言模型是获取不到的，因此将下一句话预测作为了第二个预训练任务。该任务的训练语料是两句话，来预测第二句话是否是第一句话的下一句话，如下所示

Next Sentence Prediction样例

而最终该任务得到了97%-98%的准确度。

2.3 模型输入

介绍了两个pre-training tasks之后，我们介绍该模型如何构造输入。如下图所示，输入包括三个embedding的求和，分别是：

1. Token embedding 表示当前词的embedding
2. Segment Embedding 表示当前词所在句子的index embedding
3. Position Embedding 表示当前词所在位置的index embedding



知乎

首发于
机器学习

1. 为了能够同时表示单词和句子对，多词子(例如QA中的Q/A)需要进行拼接作为单词，用segment embedding和[SEG]来进行区分
2. 句子第一个token总是有特殊含义，例如分类问题中是类别，如果不是分类问题那么就忽略
3. 三个embedding进行sum得到输入的向量

2.4. 模型训练

本文提出了两个大小的模型，分别是

1. BERT-Base: $L = 12$, $H = 768$, $A = 12$, Total parameters = 110M
2. BERT-Large: $L = 24$, $H = 1024$, $A = 16$, Total parameters = 340M

其中L表示Transformer层数，H表示Transformer内部维度，A表示Heads的数量

训练过程也是很花费计算资源和时间的，总之表示膜拜，普通人即便有idea没有算力也只能跪着。

2.5 fine-tuning

这里fine-tuning之前对模型的修改非常简单，例如针对sequence-level classification problem(例如情感分析)，取第一个token的输出表示，喂给一个softmax层得到分类结果输出；对于token-level classification(例如NER)，取所有token的最后层transformer输出，喂给softmax层做分类。

总之不同类型的任务需要对模型做不同的修改，但是修改都是非常简单的，最多加一层神经网络即可。如下图所示



知乎

首发于
机器学习

4. 实验及其分析

这里的实验可以说是NLP领域论文实验结果最残暴的一篇论文了，作者对11个NLP任务进行了fine-tuning，都取得了state-of-the-art的性能。我们介绍下NER任务的结果，如下所示





4.1 pre-train model的影响

对于Masked LM、NSP的选择是否会影响模型性能，这里做了测试，分别采用了四种模型设置进行比较，性能如下所示，显然BERTBase的效果最好的

4.2 training steps的影响

这里主要讨论Masked LM和普通LM的训练时间问题，可以看到

1. BERT的确需要训练很长steps
2. MLM的确收敛比LTR慢，但是很早就效果好于LTR了



知乎

首发于
机器学习

4.3 BERT+feature-based

由于并非所有的NLP任务都可以很容易地用Transformer encoder结构来表示，因此还是需要一个task-specific model结构。同时如果需要fine-tuning的话，transformer encoder模型很大，需要重新训练的话，需要的计算资源比feature-based方法更多，因此如果可以直接用BERT的Transformer的结果的话，就很方便使用了。因此本文做了一个BERT + task-specific model的实验。表明这种方式也是可以有很好的效果的。





5. 总结

1. BERT采用Masked LM + Next Sentence Prediction作为pre-training tasks, 完成了真正的Bidirectional LM
 2. BERT模型能够很容易地Fine-tune, 并且效果很好, 并且BERT as additional feature效果也很好
 3. 模型足够泛化, 覆盖了足够多的NLP tasks
-

Reference

[1]Peters, Matthew, et al. "Semi-supervised sequence tagging with bidirectional language models." *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2017.

[2]Peters M, Neumann M, Iyyer M, et al. Deep Contextualized Word Representations[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018, 1: 2227-2237.

[3]Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. URL [s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf](https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf), 2018.



[5]Pre-training of Deep Bidirectional Transformers for Language Understanding

编辑于 2018-10-17

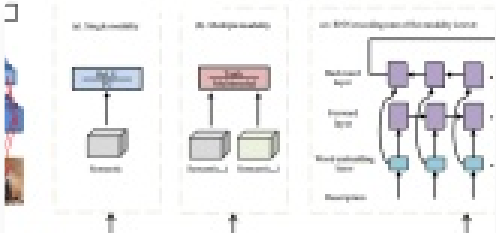
深度学习 (Deep Learning) 自然语言处理 迁移学习 (Transfer Learning)

文章被以下专栏收录

 **机器学习**
本专栏专注于介绍深度学习、传统机器学习、自然语言处理等内容，包括算法、工程...

关注专栏

推荐阅读



学习深度嵌入模型以解决Zero-shot Learning问题 (CVPR...

小栗子

快速找到论文数据的19个方法

在这个用数据说话的时代，能够打动人往往是用数据说话的理性分析，无论是对于混迹职场的小年轻，还是需要数据进行分析和研究的同学，能够找到合适的数据源都是非常重要的。特别是想要对一...

燕辞君

自
>|
今
命
的
型
关
CF
zh

24 条评论


⇌ 切换为时间排序

写下你的评论...



 斯稻朴

24 天前

 赞



 赞

 谷歌神教

24 天前

110M是指1.1亿？

 赞

 习翔宇 (作者) 回复 谷歌神教

24 天前

对

 赞

 KTV唱歌呢吗

23 天前

gothere 和 小国寡民 在微信上的聊天记录如下，请查收。

————— 2018-10-04 —————

小国寡民 10:17

你觉得同传的主要问题在哪？数据不够还是算法不够还是两方面都有？

小国寡民 10:19

按理腾讯科大讯飞这些都不是问题啊

小国寡民 10:20

包括谷歌，实力那么强，怎么就做不到同传呢？

gothere 10:21

机器翻译

gothere 10:21

1966有个alpac报告

gothere 10:21

语义不对等



知乎



首发于
机器学习

翻译太难

小国寡民 10:22

专业领域翻译呢？

小国寡民 10:22

会不会要好点？

小国寡民 10:23

我们初中那会老师教我们的，学习效率要提上去，要学会不求甚解。

小国寡民 10:23

机器翻译大概要先做到这步

gothere 10:24

简单的都做过了

小国寡民 10:26

可能大智能算法的基础还要把强化学习再上一个台阶

gothere 10:27

语义表示是根本

小国寡民 10:28

符号逻辑的智能化？

小国寡民 10:29

或者智能化的符号逻辑？

小国寡民 10:30

有点像缺一个数学上的朗兰兹纲领



知乎

首发于
机器学习

实际上所有的数学语义都要符合这个纲领

小国寡民 10:31

也就是说，它约束了数学的所有算法

小国寡民 10:32

从而因此保证了真实世界的不求甚解算法效率是最高的。

小国寡民 10:34

我们现在的聊天，就是个例子，至少这种交流方式就是普遍的现实。

小国寡民 10:35

机器翻译能做到像我们这样交流不就可以了嘛？

小国寡民 10:38

股票市场的根本也是盘面语义的解读，跟翻译是一回事，只要能读懂盘面语言，就能赚钱了

gothere 10:38

不是一回事

小国寡民 10:39

我实际就是这样做的，也赚到钱了啊

小国寡民 10:40

股票市场也是所谓不完美信息博弈，跟自然语言就是一回事

小国寡民 10:41

翻译的问题，也是不完美信息怎么处理的问题

小国寡民 10:43



知乎

首发于
机器学习

小国寡民 10:45

把不完美信息按理解难度逐步分解，然后再启发式理解，翻译也就完成了。

小国寡民 10:47

找到一个有代表性的简单模型，做一个规范重整，这是难度分解，规范化表示再反向去拟合，就是启发式理解。

小国寡民 10:50

上次你给我的词向量加lstm，其中词向量做的是分解，lstm做的是拟合。

小国寡民 10:52

只是都还不够规范，比如现在又有句向量，段向量。对应的反向拟合如残差，强化。

小国寡民 10:53

这些都需要一个整体的规范理解

小国寡民 10:55

skipgram，glovevec这些弥补规范化偏差的方法也都是很实用的。

小国寡民 10:56

这些都要能有机的整合到一起，才能让整个算法有效。

小国寡民 10:59

除了这块，不能想象还有其它什么困难让他们大公司在同传这块迟迟搞不定

小国寡民 11:06

前天看*的纪录片，*团队那帮人，看形象真是low到爆，就跟*那帮领导一样，一看就是老弱病残，跟人家谷歌，跟人家图灵，简直是两个极端，有时候真的看人就能看出端倪。

小国寡民 11:07



知乎

首发于
机器学习

小国寡民 11:11

做出来的神经网络有没有精气神，首先就要看做的人有没有。

小国寡民 11:18

翻译的信达雅其实就是翻译家的精气神，你说是吧？

gothere 11:53

老外做得很多

赞



习翔宇 (作者) 回复 KTV唱歌呢吗

23 天前

????????

赞



KTV唱歌呢吗 回复 习翔宇 (作者)

22 天前

神经网络解决人脸识别、围棋、翻译这些问题的范式是什么？

赞

展开其他 2 条回复



没钱吃白菜

22 天前

想问一下 segmentation embedding 指的是 sentence embedding 还是 词所在句子的 index embedding？论文中没找见解释

赞



习翔宇 (作者) 回复 没钱吃白菜

22 天前

index embedding，有解释吧，不然Sentence embedding从何而来

赞



PathricLee 回复 没钱吃白菜

12 天前

就是index embedding, 区分句子来源第一个词还是第二个。这个在问答等非对称句子中是用区别的。

赞



bright

12 天前

我也看了。尝试实现和应用bert的核心思想，包括使用预训练再fine-tuning以及其中的 masked language model。 github.com/brightmart/b...



知乎

首发于
机器学习

张啊啊

4 天前

请问大佬 这个的输出是怎么建立的？比如my dog is [MASK]，要预测这个 [MASK]的真实单词，它怎么从输入4个长的序列变成单个词的序列呢？还是我理解的不对

赞



习翔宇 (作者) 回复 张啊啊

4 天前

他做的是序列标注，每个词都预测，每个词对应的输出是字典大小的向量

赞



张啊啊 回复 习翔宇 (作者)

4 天前

原文写的是 “In all of our experiments , we mask 15% of all WordPiece tokens in each sequence at random. In contrast to denoising auto-encoders (Vincent et al., 2008), we only predict the masked words rather than reconstructing the entire input.” 这句话不是说的只预测masked的词吗？

赞

展开其他 3 条回复



XYXWT

3 天前

大佬好，新手想请教几个问题啊，文中说“语言模型采用了Transformer Decoder的方法来进行训练” 这里的Transformer Decoder如何理解呀？十分感激

赞



习翔宇 (作者) 回复 XYXWT

3 天前

你要去看Google 的transformer模型 zhuanlan.zhihu.com/p/46...

赞



习翔宇 (作者) 回复 XYXWT

3 天前

这两差别很大的

赞

展开其他 1 条回复



XYXWT

3 天前

还有您说的这两种方式1.feature-based 2.fine-tuning的区别，感觉还是没有很理解，菜鸟求教

