[预训练语言模型专题] 银色独角兽GPT家族

原创 管扬 朴素人工智能

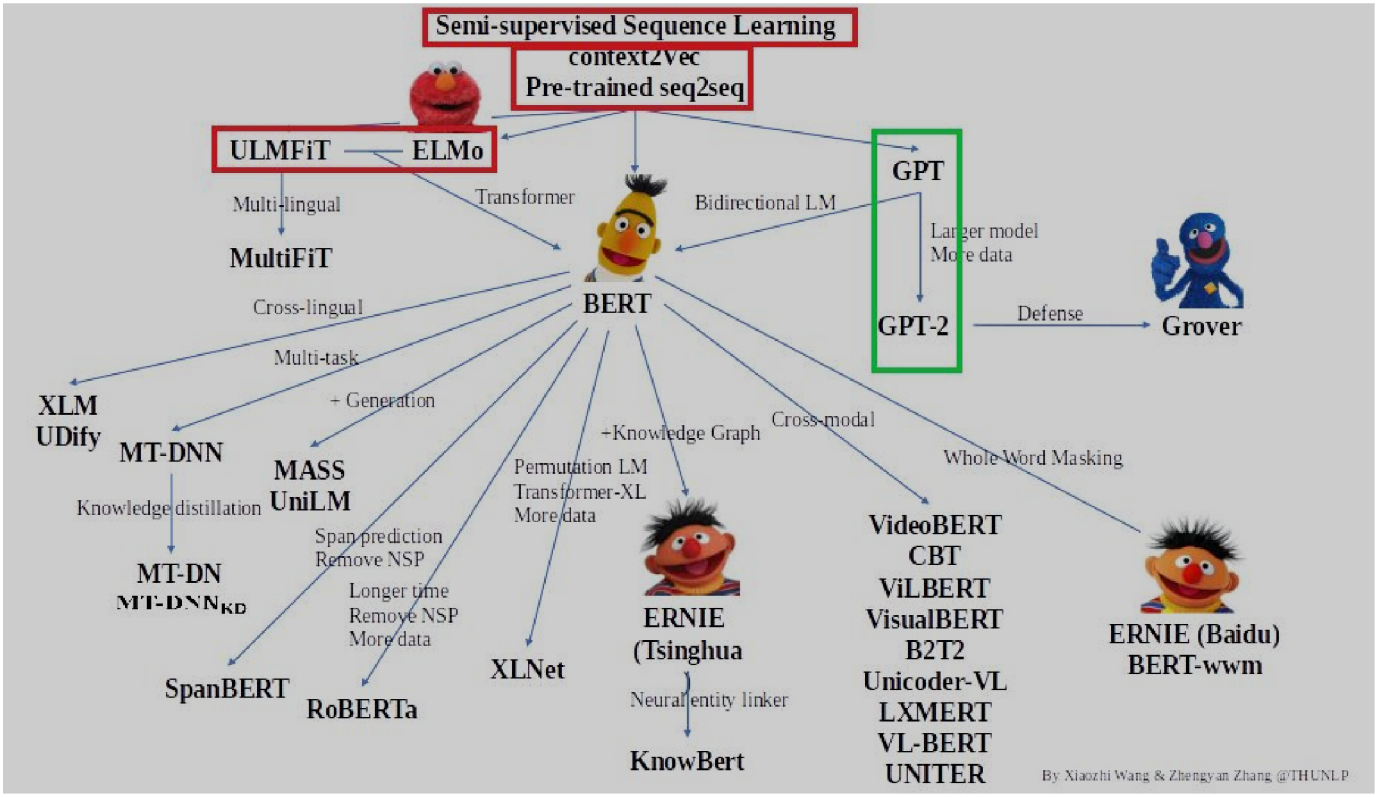
来自专辑

预训练语言模型

本文为预训练语言模型专题系列第四篇

前期回顾：[萌芽时代]、[风起云涌]、[文本分类通用训练技巧]

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，红色框为前期脚印，绿色框为本期介绍，欢迎大家留言讨论交流！



SHOCKING

在具体看论文之前，我们先来看看GPT-2，也是GPT家族巨人的惊人表现做一个开胃菜。在一个生成式任务上，OpenAI给出了这么一个开头：

SYSTEM PROMPT (HUMAN-WRITTEN)	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*

翻成中文是：在一项惊人的研究中，科学家们发现了一群**独角兽**，它们生活在安第斯山脉一个偏远的还没被开发山谷中，更令人惊讶的是这些独角兽会说一口完美的英语。

就这么一句话，GPT-2能接着它写出以下这么一大段故事：

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez. Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns. While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic." Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

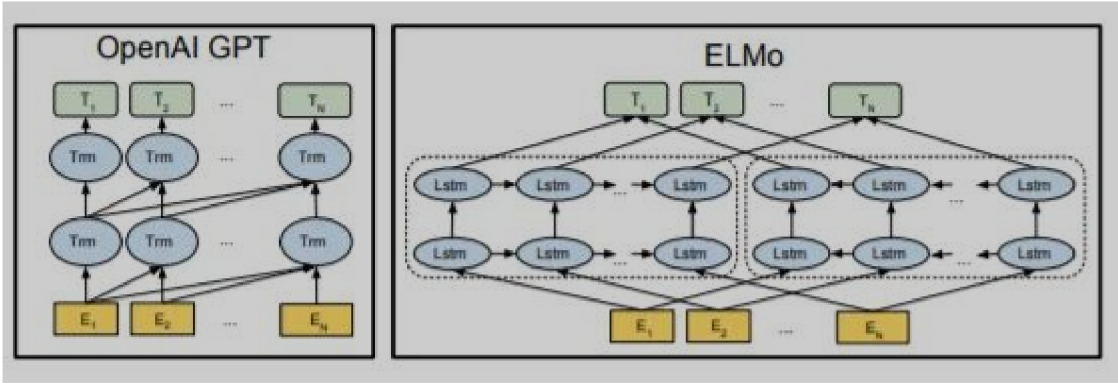
因为它们独特的角，科学家们把他们命名为Ovid's Unicorn，这些长着四个银色角的独角兽是原来科学家所不曾见过的。

.....中间还描写了，这些独角兽如何被发现，以及权威人士们评论这些生物是怎么出现的，最后还认为要验明这群独角兽是否是外星的种族，唯一方法就是通过DNA了。

这一本正经的胡说八道，让编者也是自愧不如啊。GPT-2这么强劲不是没有道理的，接下来让我们回归学术，查查GPT家族的户口本吧！

Improving Language Understanding by Generative Pre-Training (2018)

GPT是 “Generative Pre-Training” 的缩写，由OpenAI的四位作者发表于2018年。我想基于文章谈谈GPT模型的特点以及和之前模型的相似点。下图为GPT及ELMo的结构图。

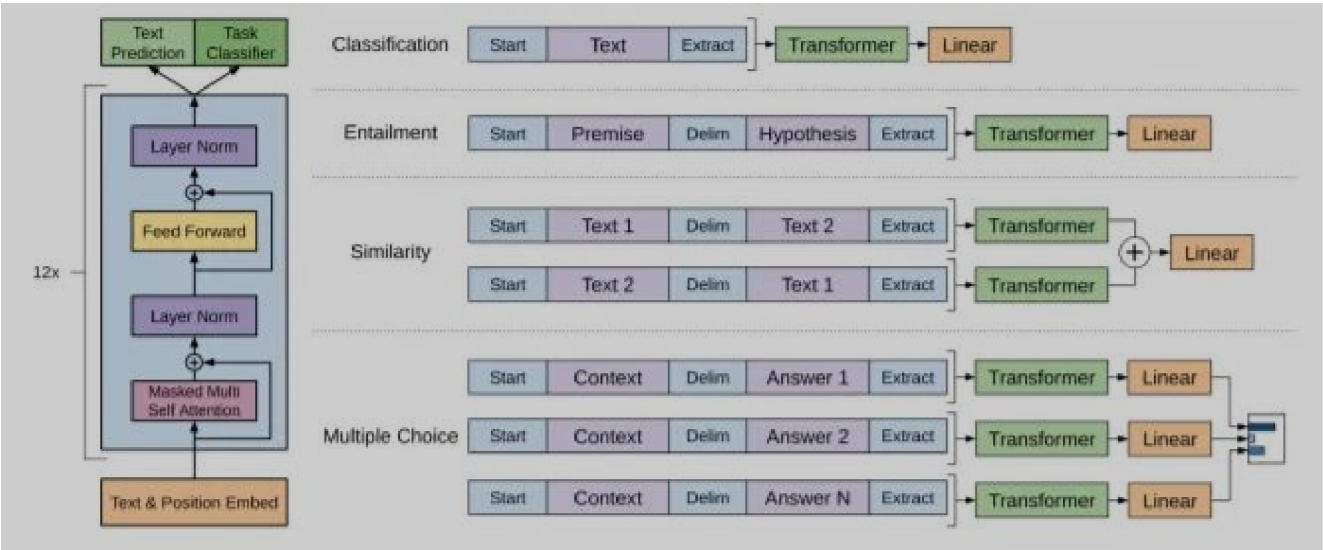


GPT模型不同于之前模型的特点：

1. GPT的模型核心组件选择了**Transformer**，区别于之前的其他一些模型包括ELMo的LSTM。这确实给模型带来了极大的性能和速度提升。关于Transformer原理，我受限于篇幅，不在这里赘述，但是强烈推荐大家对Transformer进行深入了解，可以查阅文后的参考阅读，网上的介绍也很多，如果大家希望我们出一期Transformer的推送，也可以在后台留言。

2. 为了方便将语言模型的能力transfer到下游的各种任务上去，GPT对模型的输入进行了规范，称为 **traversal-style input transformations**。把各种任务的结构化输入统一转化为有序的序列。避免在应对不同任务时，要对模型进行大的改动。其实做法很简单，我们看下面这个图就知道，使用特定的符号，来界定文本序列的开始，间隔和结束，并在预训练

时做相同的处理即可，BERT也使用了这种方法。



3. GPT对词典使用了 **bytepair encoding (BPE)** 。这样做的目的是为了缩小词典，就英文来说词是表示意义的基本单位，但是常用词有30000-50000个，全量的可达130000。字符只有256个，但是并不能很好地展现语义信息。所以应该可以找到一个更好的subword units来作为基本单元，即不能将句子长度被增加太多而降低模型性能，也能有效减少词典的大小以减少模型参数量，这种做法也确实对模型性能的提高有很大帮助。BPE的实现在下方模型细节会提到。

GPT模型和之前模型的共同点：

1. GPT模型使用经典的**two stage training**。第一个阶段，将一个大容量的语言模型在很大的无监督语料上进行预训练。第二个阶段，在特定任务的监督数据上进行finetune。

2. GPT模型沿袭了我们第一期 [萌芽时代] 中所讲文章 Semi-supervised Sequence Learning 的方法。将语言模型预训练后，把模型的参数作为**监督模型的起始点**的做法。而有区别于第一期所讲另一篇context2vec，或第二期 [风起云涌] 所讲的ELMo这样，通过预训练模型生成带上下文信息的向量表示，然后作为监督模型的特征。

3. GPT使用了**标准的语言模型目标**，优化某个词在其前面k个词出现情况下的条件概率。

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

4. 与之前的一些文章一样，GPT在特定任务的训练时，会把语言模型目标的误差一起加在总误差中**联合训练**，以提升模型的泛化能力，缓解灾难性遗忘

模型细节

GPT使用了 **multi-layer Transformer decoder** 作为语言模型的层结构。GPT模型一共有12层，每层是一个 transformer的decoder，有768维hidden state和12个attention头。position-wise feed-forward network 使用了3072维内部状态。输入层的tokens 经过word embedding 和 position embedding ，最后通过softmax得到输出的概率分布。语言模型的训练使用了2000轮的warmup，最大学习率为2.5e-4，接着通过cosine schedule下降到0，dropout使用0.1，输入的最大长度为512。在直接finetune任务目标的时候，模型训练了三轮。

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

同时GPT使用了 BPE vocabulary with 40,000 merges。BPE算法在论文Neural Machine Translation of Rare Words with Subword Units 中也有具体的代码，简单的说，算法会把相邻字符的bigram计算出现的次数，每次都把出现次数最多的bigram合并。原来的词典是256个unicode，最后经过num_merges次合并以后，词典中就多了num_merges个在整个文章中出现较多的“词根”， 比如lower中的low，widest中的est都会被合并起来。编者自己运行了一下下方的代码，最后得到的vocab是 {'low': 5, 'low e r ': 2, 'newest': 6, 'wi d est': 3}。出现最多的字符对都被merge起来，他们一般会倾向于具有独立的含义，而没有被merge的相邻字符对显然是很少联合表达某个含义，所以他们被分开进行embedding是合理的。

Algorithm 1 Learn BPE operations

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

Language Models are Unsupervised Multitask Learners（2019）

GPT家族的杰出后代GPT-2同样是由OpenAI发表，GPT-2希望能够使模型不经过任何改变就可以在下游任务上获得比较好的性能，也就是**zero-shot**。

要达成这个目标，模型就要是通用的，不能只在某些任务上进行训练。它的语言模型目标，跟GPT相似，但是因为模型要在多任务上都能有较好的表现，所以模型的预测不仅要基于前几个词作为条件，同时也要将任务考虑在内。

$$p(output|input, task)$$

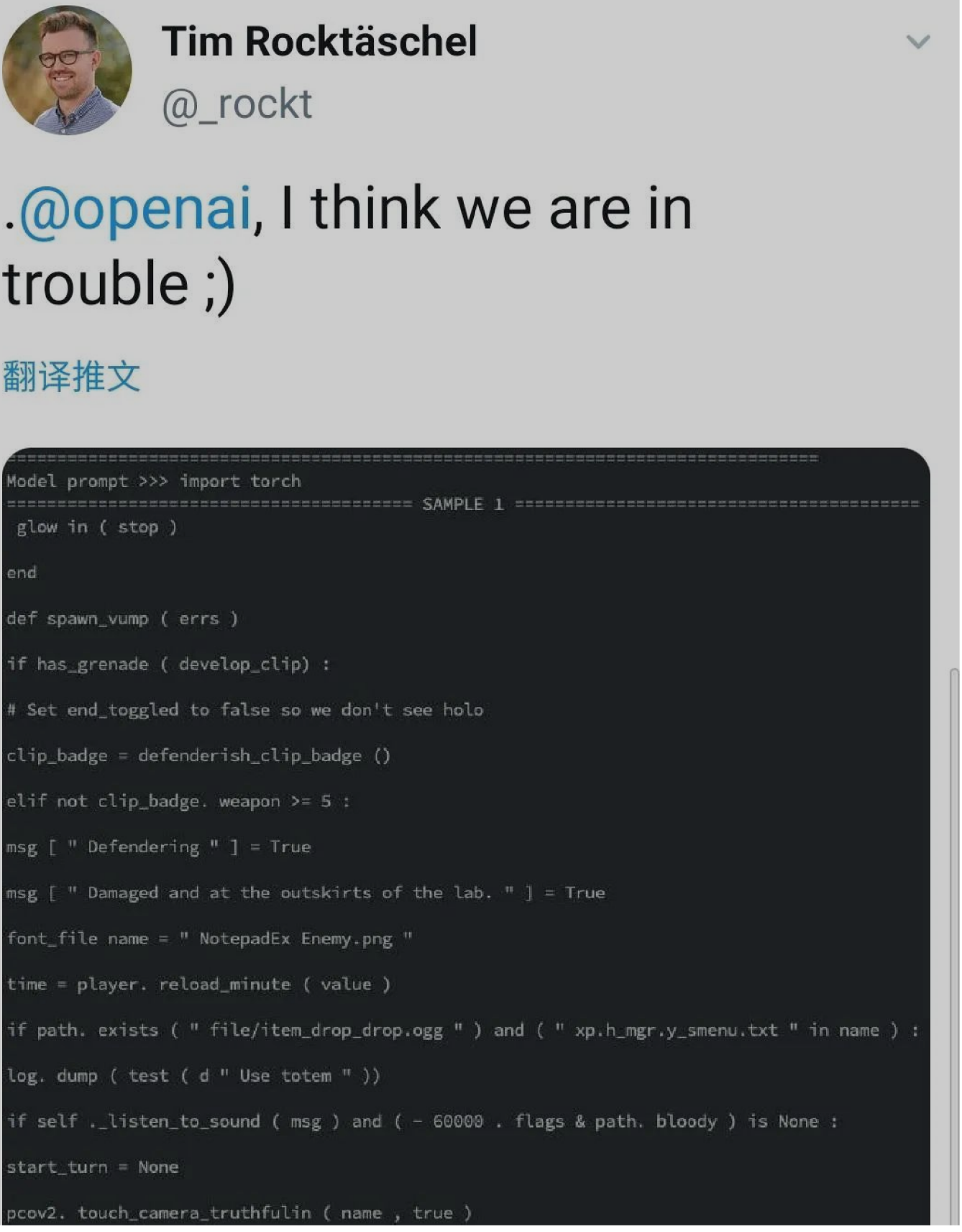
GPT2相比于GPT的几个变化：

1. **数据集变大了**，GPT2建立了一个包含了45M个链接的数据集，经过了去重和基于经验的清洗后还有8M个文章总共40GB的文字数据。

2. BPE现在会**避免不同类型的字符进行merge**，上面介绍过了BPE的原理，作者们发现在merge后会出现dog. dog? 等组合，这使字典的构建引入了噪声和不必要的变化。
3. **模型上的微小变化**， Layer normalization被移到了transformer层的输入位置，在最后一个self attention block 后添加了额外的 Layer normalization。
4. **超参变化**，字典扩充了10000个左右，一句句子最长长度从512提升到了1024，训练时 batch_size 提升到了512

它最小的模型结构等同于GPT模型，其次的等同于BERT-LARGE，最大的才是真正被称为GPT2的模型有1.5B个参数。模型如此庞大，比之前**最先进的预训练语言模型大了12倍，数据集大了15倍**。作者们认为一个巨大的模型，在一个巨大的数据集上经过了高效的训练，就能在不同的任务上取得优异的效果。事实也是这样，GPT2在七个任务上不需任何标注数据，就能超过原来的state-of-art。简直惊呆了！

更令人惊讶的是，GPT2竟然还会**写代码！！！**



难怪Facebook AI研究院的 Tim Rocktäschel 看了GPT2写的代码都表示：**我们要有大麻烦了！**

GPT2如此强大，以至于OpenAI 的研究人员们当时决定暂时不发布训练模型所用的数据、也不发布最大的真正GPT2的参数，因为他们认为模型有遭到恶意滥用的风险==

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预训练语言模型其他的论文阅读分享，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。

参考资料

1. Attention Is All You Need (2017)

   https://arxiv.org/abs/1706.03762

2. [NLP] Transformer详解

   https://zhuanlan.zhihu.com/p/44121378

3. GPT2

   http://m.elecfans.com/article/880042.html

4. 一分钟搞懂BPE

   https://blog.csdn.net/qq_27590277/article/details/88343988

推荐阅读

- 十分钟了解文本分类通用技巧
- 一次实体识别和实体消岐的积极尝试
- Kaggle TensorFlow 2.0 Question Answering 16名复盘
- [预训练语言模型的前世今生] 风起云涌
- LaserTagger: 文本生成任务的序列标注解决方案