# Title：

Improving Language Understanding by Generative Pre-Training

# Abstract：

在自然语言理解内有很多不一样的任务，虽然有很多大量的没有标号的文本文件，但是标好的数据是相对较少的，这使得（**PROBLEM：**）我们在标好的数据上训练出分辨模型比较难（数据太少）。

先在没有标号的数据上训练一个预训练模型，接下来在有标号的模型上训练一个微调模型（在CV中8、9年前成为主流的算法，但在NLP一直没有流行起来，因为NLP没有像ImageNet那样标好的大的训练集）。

GPT在微调的时候需要根据任务改变输入输出，这使得只需要很少的改变模型的架构就行。

# Introduction：

当时最流行的模型是词嵌入模型，在使用无标注的文本进行训练时，有两个困难出现了，第一是你不知道该如何解决目标函数该用哪个，第二是不知道怎样把学到的文本表示传递到下游的子任务上。

GPT这篇文章提出了一个半监督的方法（现在叫自监督学习）

GPT的模型是基于Transformer的架构，跟RNN相比，Transformer进行迁移学习的时候学习到的特征更加稳健，在迁移时使用任务相关的表示。

# Framework

## Unsupervised pre-training

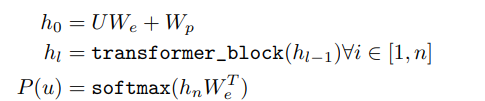
假设有个文本（无标号）表示成：U = {u1, . . . , un}，GPT使用标准语言模型的目标函数最大化以下似然函数



k为超参数

记录ui出现的概率，它把前面k个词（上下文窗口）拿过来，用后面的模型Ɵ，来预测ui的概率。把每一个词的预测的概率加起来（log为加，如果没有log就是相乘，使得出现的文本的联合概率和想出现的一样），就是目标函数L1。

使用的时Transformer的decoder：对于encoder来讲，它能看到所有的元素，对于decoder来讲，因为有掩码的存在，它只能看到当前元素和它之前的元素。



如果要预测u这个词的概率，那就把这个词和前面的k个词拿出来变成U，进行词嵌入的投影，乘We加上位置信息的编码Wp，得到第一层的输入。然后做L层Transformer块，拿到输出后在做一个投影，经过softmax得到概率分布。

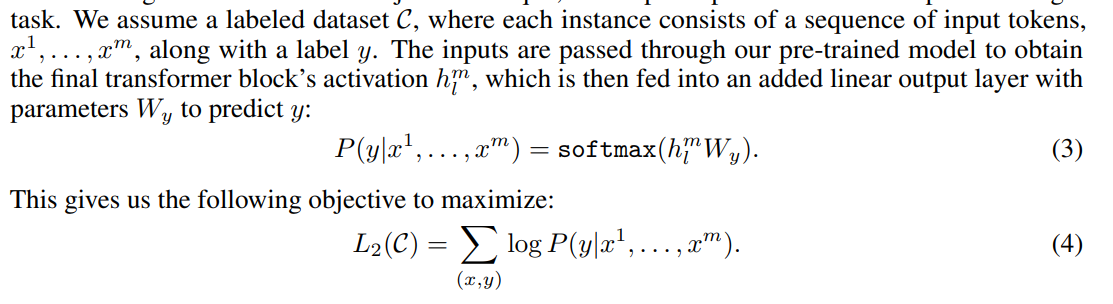
GPT和BERT的主要区别还不是encoder或者decoder的选取，是目标函数的选取。

GPT的目标函数使用的更难，是给定输入预测后面的词，BERT是“完形填空“。预测未来比完形填空要难，

这也是为什么GPT-3效果这么好的原因，更难训练，训练起来效果就更好

## Supervised fine-tuning

在微调内训练数据是有标号的

y是有标号的，（3）是一个分类的目标函数

再把之前的目标函数加进来也不错（给定完整的序列，预测序列对应的标号）



λ是超参数

如何把下游任务表示成一个序列+标号的形式呢？

