ChatGPT的工作原理

— 0 —

前言

ChatGPT 能够自动生成一些读起来表面上甚至像人写的文字的东西，这非常了不起，而且出乎意料。但它是如何做到的？为什么它能发挥作用？我在这里的目的是大致介绍一下 ChatGPT 内部的情况，然后探讨一下为什么它能很好地生成我们认为是有意义的文本。

还没使用过ChatGPT的伙伴可以点击下面链接直接使用（不需要科学上网工具，后台对接的是OpenAI和微软的官方接口）：
https://chatgpt.zntjxt.cn/

我首先要说明一下，我将把重点放在正在发生的事情的大的方向上，虽然我会提到一些工程细节，但我不会深入研究它们。（我所说的实质内容也同样适用于目前其他的“大型语言模型”LLM 和 ChatGPT）。

首先要解释的是，ChatGPT 从根本上说总是试图对它目前得到的任何文本进行“合理的延续”，这里的“合理”是指“在看到人们在数十亿个网页上所写的东西之后，人们可能会期望某人写出什么”。

因此，假设我们已经得到了 “人工智能最好的是它能去做 ……” 的文本（“The best thing about AI is its ability to”）。想象一下，扫描数十亿页的人类书写的文本（例如在网络上和数字化书籍中），并找到这个文本的所有实例 —— 然后看到什么词在接下来的时间里出现了多少。

ChatGPT 有效地做了类似的事情，除了（正如我将解释的）它不看字面文本；它寻找在某种意义上 “意义匹配” 的东西。但最终的结果是，它产生了一个可能出现在后面的词的排序列表，以及 “概率”。

The best thing about AI is its ability to

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

值得注意的是，当 ChatGPT 做一些事情，比如写一篇文章时，它所做的基本上只是反复询问 “鉴于到目前为止的文本，下一个词应该是什么？” —— 而且每次都增加一个词。（更准确地说，正如我将解释的那样，它在添加一个 “标记”，这可能只是一个词的一部分，这就是为什么它有时可以 “编造新词”）。

在每一步，它得到一个带有概率的单词列表。但是，它究竟应该选择哪一个来添加到它正在写的文章（或其他什么）中呢？人们可能认为它应该是 “排名最高” 的词（即被分配到最高 “概率” 的那个）。

但是，这时就会有一点巫术开始悄悄出现。因为出于某种原因 —— 也许有一天我们会有一个科学式的理解 —— 如果我们总是挑选排名最高的词，我们通常会得到一篇非常“平淡”的文章，似乎从来没有“显示出任何创造力”（甚至有时一字不差地重复）。但是，如果有时（随机的）我们挑选排名较低的词，我们会得到一篇“更有趣”的文章。

这里有随机性的事实意味着，假如我们多次使用同一个提示，我们也很可能每次都得到不同的文章。而且，为了与巫术的想法保持一致，有一个特定的所谓“温度”参数（temperature parameter），它决定了以什么样的频率使用排名较低的词，而对于论文的生成，事实证明，0.8 的“温度”似乎是最好的。（值得强调的是，这里没有使用任何“理论”；这只是一个在实践中被发现可行的问题）。例如，“温度”的概念之所以存在，是因为恰好使用了统计物理学中熟悉的指数分布，但没有“物理”联系 —— 至少到目前为止我们如此认为。）

在我们继续之前，我应该解释一下，为了论述的目的，我大多不会使用 ChatGPT 中的完整系统；相反，我通常会使用更简单的 GPT-2 系统，它有一个很好的特点，即它足够小，可以在标准的台式电脑上运行。

因此，对于我展示的所有内容，包括明确的沃尔弗拉姆语言（Wolfram Language）代码，你可以立即在你的计算机上运行。

例如，这里是如何获得上述概率表的。首先，我们必须检索底层的“语言模型”神经网：

```
In[ ]:= model =
          NetModel[{"GPT2 Transformer Trained on WebText Data",
            "Task" → "LanguageModeling"}]

Out[ ]= NetChain[ Input port:  string
                  Output port: class ]
```

稍后，我们将看看这个神经网的内部，并谈谈它是如何工作的。但现在我们可以把这个“网络模型”作为一个黑匣子应用于我们迄今为止的文本，并要求按概率计算出该模型认为应该选择的前五个词：

```
In[ ]:= model["The best thing about AI is its ability to", {"TopProbabilities", 5}]

Out[ ]= {do → 0.0288508, understand → 0.0307805,
         make → 0.0319072, predict → 0.0349748, learn → 0.0445305}
```

这就把这个结果变成了一个明确的格式化的“数据集”：

```
In[ ]:= Dataset[ReverseSort[Association[%]],
          ItemDisplayFunction → (PercentForm[#, 2] &)]
```

Out[ ]=

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

如果重复“应用模型”——在每一步中加入概率最高的词（在此代码中被指定为模型的“决定”），会发生什么：

```
In[•]:= NestList[StringJoin[#, model[#, "Decision"]] &,
          "The best thing about AI is its ability to", 7]

Out[•]= {The best thing about AI is its ability to,
         The best thing about AI is its ability to learn,
         The best thing about AI is its ability to learn from,
         The best thing about AI is its ability to learn from experience,
         The best thing about AI is its ability to learn from experience.,
         The best thing about AI is its ability to learn from experience. It,
         The best thing about AI is its ability to learn from experience. It's,
         The best thing about AI is its ability to learn from experience. It's not}
```

如果再继续下去会发生什么？在这种情况下（“零温度”），很快就会出现相当混乱和重复的情况：

```
The best thing about AI is its ability to learn from experience. It's not just a matter
of learning from experience, it's learning from the world around you. The AI is a very
good example of this. It's a very good example of how to use AI to improve your life.
It's a very good example of how to use AI to improve your life. The AI is a very good
example of how to use AI to improve your life. It's a very good example of how to use AI to
```

但是，如果不总是挑选“顶级”词，而是有时随机挑选“非顶级”词（“随机性”对应“温度”为 0.8）呢？人们又可以建立起文本：

{The best thing about AI is its ability to,
The best thing about AI is its ability to create,
The best thing about AI is its ability to create worlds,
The best thing about AI is its ability to create worlds that,
The best thing about AI is its ability to create worlds that are,
The best thing about AI is its ability to create worlds that are both,
The best thing about AI is its ability to create worlds that are both exciting,
The best thing about AI is its ability to create worlds that are both exciting,}

而每次这样做，都会有不同的随机选择，文本也会不同 —— 如这 5 个例子：

The best thing about AI is its ability to learn. I've always liked the

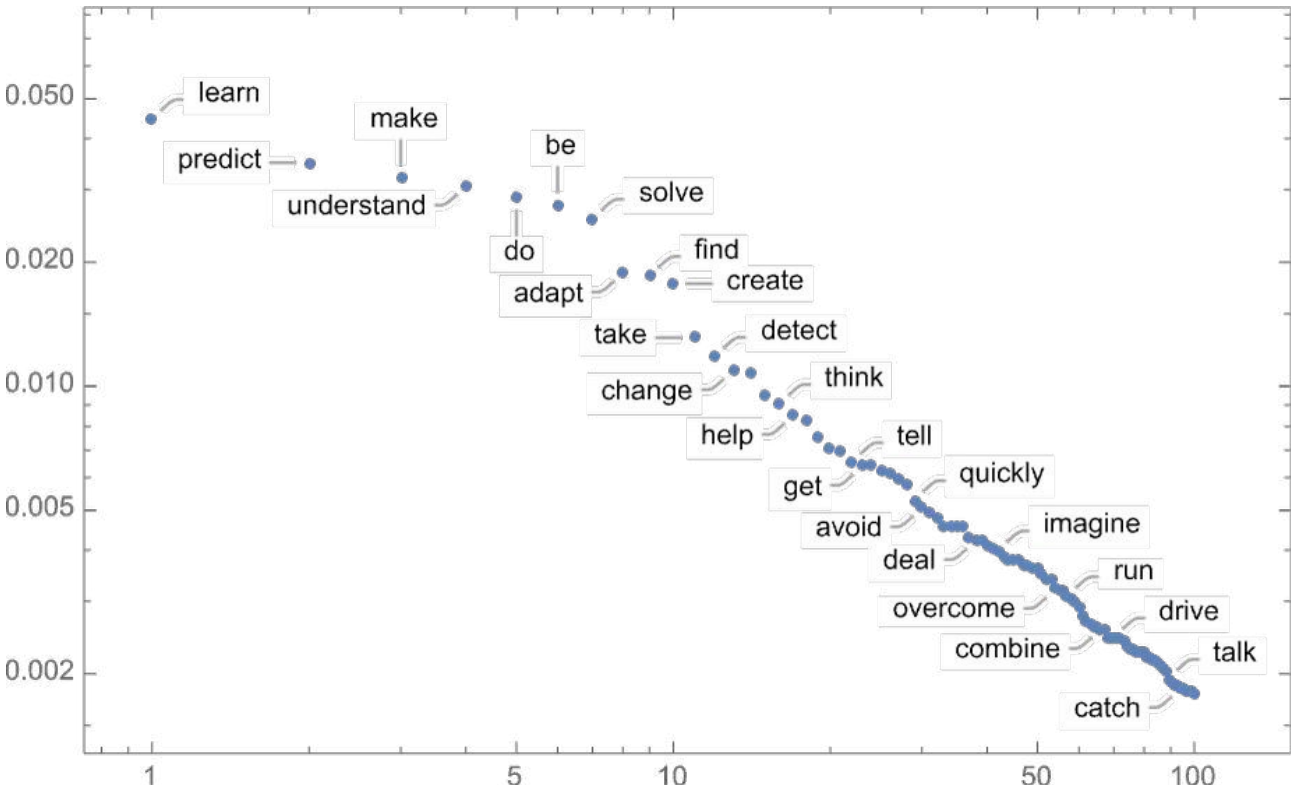
The best thing about AI is its ability to really come into your world and just

The best thing about AI is its ability to examine human behavior and the way it

The best thing about AI is its ability to do a great job of teaching us

The best thing about AI is its ability to create real tasks, but you can

值得指出的是，即使在第一步，也有很多可能的 “下一个词” 可供选择（温度为 0.8），尽管它们的概率下降得很快（是的，这个对数图上的直线对应于 n-1 的 “幂律” 衰减，这是语言的一般统计的特点）：



那么，如果继续下去会发生什么？这里有一个随机的例子。它比顶层词（零温度）的情况要好，但顶多还是有点奇怪：

The best thing about AI is its ability to learn from experience. It's not just a matter of learning from experience, it's learning from the world around you. The AI is a very good example of this. It's a very good example of how to use AI to improve your life. It's a very good example of how to use AI to improve your life. The AI is a very good example of how to use AI to improve your life. It's a very good example of how to use AI to

这是用最简单的 GPT-2 模型（来自 2019 年）做的。用较新和较大的 GPT-3 模型，结果更好。这里是用同样的 “提示” 产生的顶部文字（零温度），但用最大的 GPT-3 模型：

The best thing about AI is its ability to automate processes and make decisions quickly and accurately. AI can be used to automate mundane tasks, such as data entry, and can also be used to make complex decisions, such as predicting customer behavior or analyzing large datasets. AI can also be used to improve customer service, as it can quickly and accurately respond to customer inquiries. AI can also be used to improve the accuracy of medical diagnoses and to automate the process of drug discovery.

这是“温度为 0.8”时的一个随机例子：

The best thing about AI is its ability to learn and develop over time, allowing it to continually improve its performance and be more efficient at tasks. AI can also be used to automate mundane tasks, allowing humans to focus on more important tasks. AI can also be used to make decisions and provide insights that would otherwise be impossible for humans to figure out.

— 1 —

概率从何而来?

好吧，ChatGPT 总是根据概率来选择下一个词。但是这些概率从何而来？

让我们从一个更简单的问题开始。让我们考虑一次生成一个字母（而不是单词）的英语文本。我们怎样才能算出每个字母的概率呢？

我们可以做的一个非常简单的事情就是取一个英语文本的样本，然后计算不同字母在其中出现的频率。因此，举例来说，这是计算维基百科上关于“猫”（cat） 的文章中的字母：

```
In[ ]:= LetterCounts[WikipediaData["cats"]]

Out[ ]= <| e → 4279, a → 3442, t → 3397, i → 2739, s → 2615, n → 2464, o → 2426,
  r → 2147, h → 1613, l → 1552, c → 1405, d → 1331, m → 989, u → 916,
  f → 760, g → 745, p → 651, y → 591, b → 511, w → 509, v → 395, k → 212,
  T → 114, x → 85, A → 81, C → 81, I → 68, S → 55, F → 42, z → 38, E → 36
```

而这对“狗”（dog） 也有同样的作用：

```
In[ ]:= LetterCounts[WikipediaData["dogs"]]

Out[ ]= <| e → 3911, a → 2741, o → 2608, i → 2562, t → 2528, s → 2406,
  n → 2340, r → 1866, d → 1584, h → 1463, l → 1355, c → 1083, g → 929,
  m → 859, u → 782, f → 662, p → 636, y → 500, b → 462, w → 409,
  v → 406, k → 151, T → 90, C → 85, I → 80, A → 74, x → 71, S → 65,
```

结果相似，但不一样（“o” 在 “dogs” 文章中无疑更常见，因为毕竟它出现在 “dog” 这个词本身）。尽管如此，如果我们采取足够大的英语文本样本，我们可以期待最终得到至少是相当一致的结果。

```
In[ ]:= English LANGUAGE [ character frequencies ]

Out[ ]= {e → 12.7%, t → 9.06%, a → 8.17%, o → 7.51%, i → 6.97%, n → 6.75%,
  s → 6.33%, h → 6.09%, r → 5.99%, d → 4.25%, l → 4.03%, c → 2.78%, u → 2.76%,
  m → 2.41%, w → 2.36%, f → 2.23%, g → 2.02%, y → 1.97%, p → 1.93%, b → 1.49%,
  v → 0.978%, k → 0.772%, j → 0.153%, x → 0.150%, q → 0.0950%, z → 0.0740%}
```

下面是我们得到的一个样本，如果我们用这些概率生成一个字母序列：

```
rronoitadatcaeaesaotdoysaroiyiinnbantoioestlhddeocneooewceseciselnodrtrdgriscsatsepesdcnio
    uhoetsedeyhedslernevstothindtbmnaohngotannbthrdthtonsipieldn
```

我们可以通过添加空格将其分解为 “单词”，就像它们是具有一定概率的字母一样：

sd n oeiaim satnwhoo eer rtr ofiianordrenapwokom del oaas ill e h f
rellptohltvoettseodtrncilntehtotrkthrslo hdaol n sriaefr hthehtn ld gpod a h y oi

我们可以通过强迫 “字长” 的分布与英语中的分布相一致，在制造 “单词” 方面做得稍微好一点：

ni hilwhuei kjtn isjd erogofnr n rwhwfao rcuw lis fahte uss cpnc
nlu oe nusaetat llfo oeme rrhrtn xdses ohm oa tne ebedcon oarvthv ist

我们在这里没有碰巧得到任何 “实际的词”，但结果看起来稍好一些。不过，要想更进一步，我们需要做的不仅仅是随机地分别挑选每个字母。例如，我们知道，如果我们有一个 “q”，下一个字母基本上必须是 “u”：

这里有一个字母本身的概率图：