格局打开，带你解锁 prompt 的花式用法

原创 Severus 夕小瑶的卖萌屋 2021-09-14 12:05

文 | Severus

> 就如同《倚天屠龙记》中的主角张无忌，语言模型修炼了深厚的内功，但是遇到他的乾坤大挪移之前，他空有一身本领却不会用。但学会之后，于所有武功又都融会贯通。光明顶上血战六大派，他可以打出比崆峒派威力更大的七伤拳，比少林寺更加正宗的龙爪手。武当山上，他可以最快学会太极拳和太极剑法。prompt 是否又是语言模型的乾坤大挪移呢?

大家好，我是 Severus，一个在某厂做中文文本理解的程序员。今天我想要分享的是在工业实践中使用 prompt 的一些实践和心得体会。话不多说，我们直接开始。

初次关注到 prompt 是在去年GPT-3发布之后，我读到了一篇论文，*It' s Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners*，了解我的小伙伴都会知道，虽然我是一个预训练语言模型的使用者，甚至生产者，但对于超大规模的语言模型，我一直持相对否定的态度，所以这篇文章的标题就相当吸引我，并且读下来之后，隐隐感觉，**将文本理解任务转换为预训练的任务形式，再使用预训练语言模型**，去做这个任务，这个思路简直太无懈可击了！利用它，我们可以更轻松地完成很多工作，又不必去面对例如样本类别均衡之类的数据分布上的困扰。

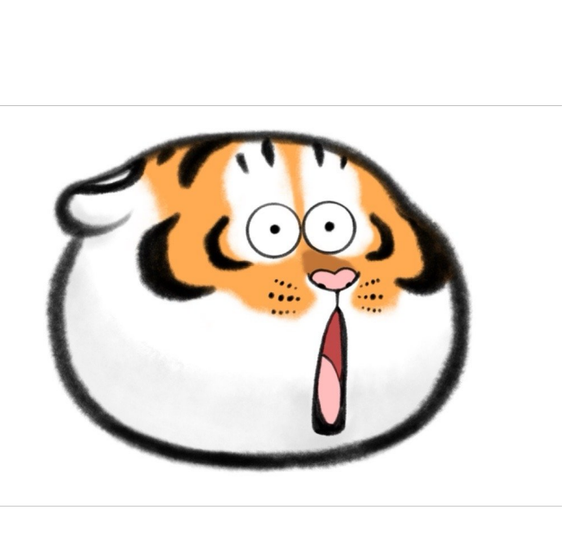
但当时却没有勇气直接应用起来。

到了今年，prompt 成为了一个相当火热的方向，在那篇 prompt 综述[1]出来了之后，我们知道，prompt 已经成气候了，它已经被很多工作验证是有用的了，也终于下定了决心，在我们的项目中尝试使用它，并看一下它到底有多么神奇。用过之后，不得不说，真香！而且除了小样本之外，我们也解锁了 prompt 的一些花式用法，今天我竹筒倒豆子，分享出来。

浅谈我对prompt的理解

本节是我基于我近几年的分析经验，尝试去理解一下 prompt 为什么这么香，以及怎么要使用它，它才这么香，当然这一节绝大多数是我的胡乱猜想，没有什么实验数据支撑，一家之言，不一定对。

初见 prompt，我的第一反应就是，在**下游任务上的训练方式和预训练的训练方式直接同源了**，那必然是可以充分利用到预训练语言模型的“遗产”了啊，而相比之下，**fine-tuning 的思路则是强行将任务转换成了另一种形式，对于预训练语言模型的知识是没有充分利用到的**，这么一看，简直是越看越靠谱了。



当然，上面那些想法现在有了更加明白的解释，那就是**将语言模型看做是知识库**（Language Models as Knowledge Bases）。实际上，prompt 可以看做是对**预训练语言模型中已经记忆的知识的一种检索方式**，由于 prompt 任务形式就是预训练任务，所以相比于 fine-tuning，当使用 prompt 形式向模型中输入样本时，**预测时所需使用到的信息量是更多的**。因为要用到更多的信息量，所以**样本的类别分布在 prompt 视角之下都是稀疏的**，这应该也是 prompt 在 few-shot 上效果显著的原因。

```
# 使用我们开发的预训练语言模型直接预测的结果
input:  送元二使安西是一首[MASK][MASK]。
output:  送元二使安西是一首诗词。
```

我们的文本分类任务，类别标签都是有语义的，且这个语义是自然语义，所以，我们假设，在自编码预训练语言模型中，样本标签和样本之间天然就存在潜在的关联，而在 prompt-tuning 上，我们只需要将这个关联抽取出来，强化它就好。

与之不同的是，fine-tuning 形式中，对于模型来讲，每个类别标签只是一个 id，所以在训练过程中，模型是根据样本什么样的分布来调整参数的对于我们来讲是不可感知的，但是单从模型的表现来看，我们可以注意到，模型是很容易走捷径的，如果某一类样本中存在明显的分布 bias，则对其他少量分布的样本就是相当不友好的。而如果那个 bias 和类别本身在语义上不相关，则更加是一个灾难。

总之，**prompt 更加依赖先验，而 fine-tuning 更加依赖后验**。也正因为这种特点，prompt 生效的最基本前提是**预训练语料已经覆盖了任务语料的分布**，所幸，得力于超大规模的公开语料，现在绝大多数任务都是可以做到的，做不到的也可以利用领域适配等 trick 去补足。

以上是我对 prompt 的一些个人理解，同时，我猜想 prompt 可能还存在一些特点（但我不确定）：

1. 使用 prompt 的前提是，任务中，输入和答案之间需要存在语言的自然语义分布中的关联，由于预训练语言模型使用的是海量的自然语料，我们便理想化地认为预训练语言模型中学习到的知识就是自然的语义分布；
2. 也正因为第1点，同时也因为 prompt 所检索出来的信息量是更多的，所以 prompt 对样本中的噪音、混淆会很敏感，对任务的数据质量要求很高（这或许也是在大规模数据上，prompt 可能干不过 fine-tuning 的原因）；
3. 在自编码模型上，prompt 的重点可能是关联，而非一定要符合自然的表达习惯，但在自回归模型上，可能就更要求 prompt 符合自然表达习惯（这点是我基于两个模型的概率公式口胡的）。

接下来我介绍一下我们在prompt上的一些实践。

直接使用 prompt 进行细粒度分类

我们的第一个作为 prompt 试水的项目，是一个短文本分类的任务，这个任务的输入是一个名词性短语，比如：刘德华、李达康、豉汁蒸排骨，预测这个名词性短语的类别，比如：人、菜品等。起初，这个任务我们只去做粗粒度类别的分类，总共是42个类别，而决定使用 prompt 之后，我们认为，prompt 这么好的性质，不直接做细粒度分类简直浪费了啊！所以我们根据词语后缀（名词性短语的类别绝大多数由后缀决定）将原本的任务转换成了2100个类别，准备开始做。

实际上，在开始做之前，我还是有一点担忧，毕竟，初期实验，我们打算直接用预训练的参数，做 token 级的分类，但是由于**中文是字级别的 token，但是类别标签都是多于一个字的，而且长度是不固定的**（毕竟在此之前好像我看到的 prompt 的答案多数是一个 token 的），所以就有两个担心：

- 最终预测的结果是否会控制在类别标签集合中，预测出来的 token 是否在类别标签集合对应的字符集合中
- 模型能否学会变长的 prompt

于是我去查找文章，看到一些文章里给出的结论说 prompt 的长度可以是10~20，并且自己也打算用 padding 的方式直接去控制不定长答案学习，也不管字符集的问题，直接使用最朴素的方法去训练，训练过后，测试结果直接让我震惊了：精度是88.9%！而之前42个类的分类精度，也仅仅是89%。

后续我们不断地迭代数据，去除样本中的噪音之后，现在测试精度已经达到了90.17%，并且还有继续改进的空间。

这个过程中，超出了我的预期的是，首先**预测结果中不在类别标签集合中的结果占比仅仅在2%左右，且这些预测结果与已有标签仅仅相差1~2个莱文斯坦距离**（levenstein distance，将字符串 A 编辑转换为字符串 B 最少需要经过多少次单字符增加、删除、替换操作），这些预测结果，如果考虑到用户体验，我只需要使用BK树修正即可。并且**预测结果中完全没有预测到标签集合所对应字符集之外的 token**，且我用 padding 控制长度也是生效的。而这全程，我使用的都是平常 fine-tuning 时使用的学习率（相比于预训练小了两个数量级）。

同时，我的训练样本类别分布是极其不平衡的，样本最多的类可能有几十万，样本最少的类可能仅仅有几十个，但无论多少，模型最终预测的结果都是正确的，这又恰恰证明了 prompt 视角下，样本的类别分布都是稀疏的，进一步给出了一个启发：**使用 prompt 可以轻松地解决不平衡分类的问题。**

在做这个任务的过程中，还发生了一个小插曲，就是我把标签列表的 offset 设置串位了，所有的标签 token 提前了两位，不在 mask 的位置上了，但是最终的测试精度居然仍是88.9%，原本是一个低级错误，但我讲出来，主要还是认为它佐证了我所说的，**自编码模型的 prompt 似乎只需要考虑关联，而不需要考虑自然表达习惯**。当然这个验证不合理，我需要在其他任务上使用不那么符合自然分布的 prompt 验证一下效果。

在我做这个项目的同时，我的同事也尝试了一些其他的项目，例如使用 span 结果直接去询问NER类别，也达到了93%的F1（那份数据集本身的质量大概也就是那个数了，例如“中国人民银行”的类别有10:7的矛盾，景点和地址的类别定义混淆不清），使用 prompt 做关系分类，也达到了96%左右的F1。

这个模型，我们也将于11月在**PaddleNLP-解语**项目中开源，有兴趣的小伙伴们敬请期待。

以上这些实验结果，直接坚定了我在其他任务上继续使用 prompt 的信心，而在做这些任务的过程中，我也发现了更加有趣的事情。

使用 prompt 检测数据噪音

前文说到了，prompt 对噪音、混淆数据是非常敏感的，那么我们就想到，可以反过来利用这个特性，去探测数据中的噪音，迭代优化数据。最直接的，就是 prompt 方式预测出的**不在标签集里面的数据**。

因为**模型的预测表现一定反映了训练样本的情况**，那么，模型出现了预测结果不在标签集里面的情况，则一定反映了训练样本中，某一类数据出现了类别混淆。所以使用模型预测全量的训练样本之后，将此类样本找出，即可判定训练样本中需要做什么样的调整。例如：某一版模型将“软刺裸裂尻鱼”预测成了“鱼物”，则反映出训练样本中的鱼类一定出现了类别标签混淆，一部分被标注成了“鱼类”，一部分被标注成了“动物”，且二者在语义上都是成立的，所以模型也无法分辨最终想要的是哪一个知识。这种现象不仅仅在这个名词性短语分类中见到，也在上面提到的NER任务中有体现。

除根据类别混淆情况探测之外，我们同样也可以使用**训练样本的真实标签与预测标签的错误，找到那些根本无法区分的边缘样本**。对于名词性短语分类这个任务，我们知道，命名实体类样本，如作品名、角色名、品牌名等，是极易与其他类样本混淆，因为这一类名字可以任意起，且很容易与一般词语重合，例如作品中各种专业书籍、教科书、人物传记等，角色名与人名也极易出现混淆。在没有上下文的情况下，哪怕是人类也难以区分这些短语的正确类别。但是，比如“高等数学”，当这个词单独出现的时候，我们大概率会认为它所指的是那个学科，而不是某一本教科书。

所以，根据这一原则，我们找到那些冲突的类别，直接从训练样本中删掉，避免模型困惑。果真，基于这一原则删掉了几万条数据之后，模型的效果甚至提升了1个点。

我们认为，这个结果是可以推广到所有基于 prompt 的任务上的。**prompt 模型、样本、预测错误三者之间形成了一个对抗、迭代的关系**，基于这样一种机制，充分利用预训练语言模型从海量语料里面学习到的知识，我们可以更加轻松地去迭代优化数据，同时也能够标注生成更大规模的高质量数据，帮助其他的模型得到更好的结果。

用于知识增强

至此，我们就没有必要拘泥于 prompt 的任务形式了，试着开一开其他的脑洞。

前面提到，prompt 思路本质上**是将预训练语言模型看作是一个知识库**（Language Models as Knowledge Bases），而 prompt 奏效了，给我们的启发是，**知识增强的方式，可以更加自然**。

首先，我们假设海量的预训练语料中已经包含了比如对于某一个实体的知识，包括它的描述知识，和它的分布知识（即上下文提示）。那么，不仅限于 prompt 的任务形式，我们直接使用包含了知识的自然文本，到预训练语言模型中检索出它的表示，是否就可以得到一种**更加自然，且与预训练语言模型天生更加契合**的知识增强方式呢?

实际上，在细粒度命名实体识别任务 CLUENER 上，就存在类似的例子。CLUENER 任务中，有几个非常难以区分的类别，分别是电影、游戏、书籍。这三种都是作品类的命名实体，在上下文中没有相应的信息的时候，极易发生混淆，实际上数据中的标签上也有一些混淆。但是在实验中，我们发现，**在输入文本中明确提示了答案的时候，模型往往能正确标注出命名实体的类别**。这间接佐证了，模型是有能力感知到输入文本中的提示的。

那么我们当即决定先采取最朴素的方法，在输入文本的后面追加一些知识文本做提示，实验过后，在一些易混淆 case 上取得了一些成效，但是整体效果提升不大，这当然是符合预期的，毕竟这种朴素的插入并没有什么目的，没有让模型有“观察的重点”，而下面的一篇工作[2]，则是我们的想法的更完善版本。

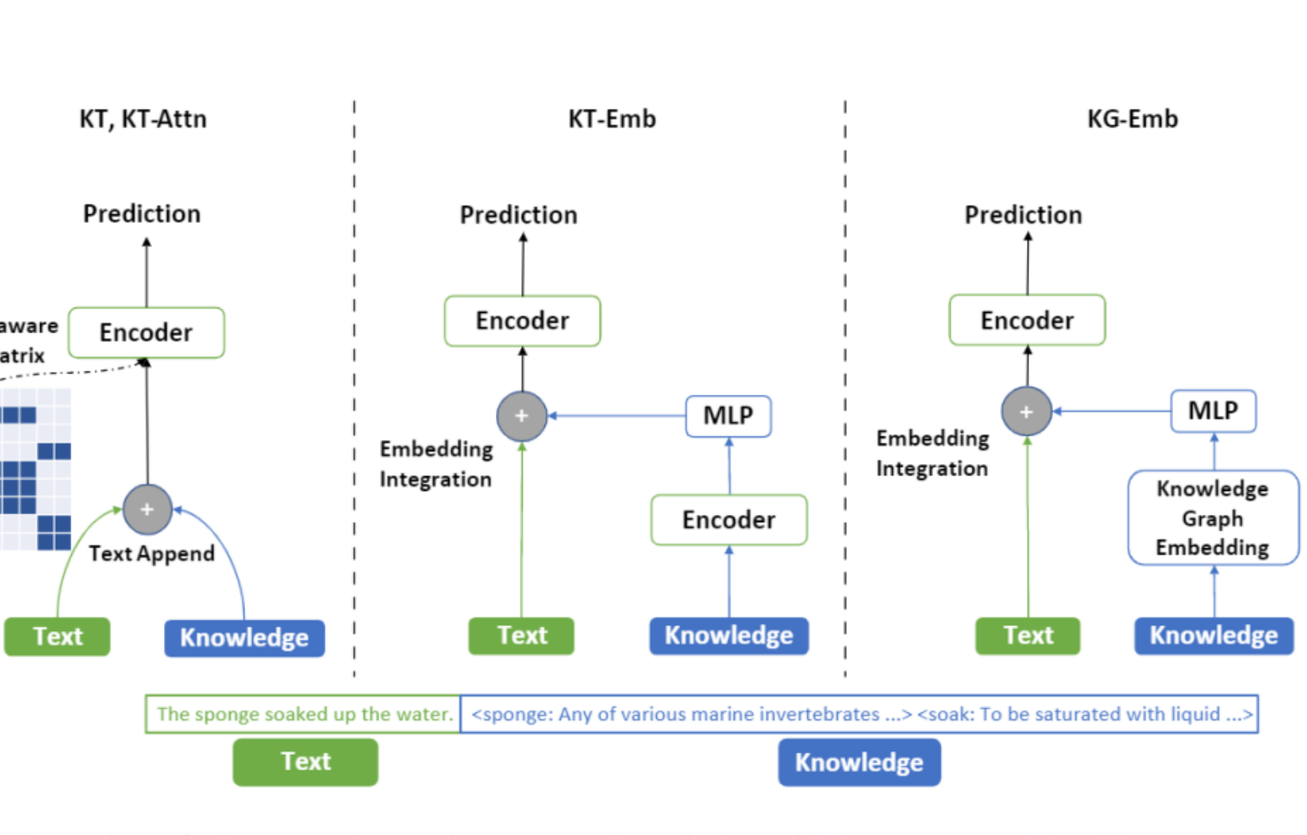


Figure 1: Illustration of all approaches to incorporate knowledge with language models. Here we assume the encoder module has the flexibility to take either a sequence of tokens or a sequence of token embeddings.

这篇工作使用了4种方式将知识融合到 NLP 模型中：

- KT 方式和我前面尝试的方式一样，直接将知识文本（KT）追加到输入之中；
- KT-Attn 则是在 attention 上叠加了一个可见性矩阵，控制**只有知识描述的 token 能够看到自身描述的知识**；
- KT-Emb 是使用 encoder 先计算得到 KT 的表示，再叠加到文本表示上；
- KG-Emb 则是使用知识图谱表示去增强模型，图谱表示则是使用 TransE 计算得到

其中，KT 方式的知识来自于 Wikitonary 词典数据，通过词性标注等方式查到对应的词，将词典中的释义看作是知识，加入到任务之中。KG-Emb 的方式，则是使用 Wikidata 的数据，将图谱知识融合到任务中。**其中 KT-Attn 方式是我心中比较理想的增强方式，而 KT-Emb 方法是我没有想到过的，但是看了文章之后我也尝试理解了这个方法，感觉上也是 make sense 的。**

实验结果如下：

Model	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	Avg
Metrics	Matt. corr.	Acc.	Acc.	Pear. corr.	Acc.	Acc.	Acc.	Acc.	
RoBERTa-Large (Liu et al., 2019)	68.0	96.4	90.9	92.4	**92.2**	90.2	94.7	86.6	88.93
RoBERTa-Large (ours)	67.02	96.22	90.93	92.71	92.15	90.59	**94.73**	90.61	89.37
+ KT	68.52	96.33	89.22	92.39	92.01	90.49	94.55	90.97	89.31
+ KT-Attn	**68.84**	96.44	**91.18**	92.61	92.09	**90.63**	94.67	**91.7**	**89.77**
+ KT-Emb	68.22	**96.56**	90.69	**92.8**	92.08	90.56	**94.73**	90.97	89.58
+ KG-Emb	68.03	96.44	90.69	92.42	92.19	**90.63**	94.55	90.61	89.45
RoBERTa-Base (ours)	60.07	94.72	**89.71**	90.95	**91.58**	87.73	92.84	75.09	85.34
+ KT	**62.89**	94.72	88.24	89.87	91.57	87.78	92.75	69.68	84.69
+ KT-Attn	62.35	94.84	89.22	**90.98**	**91.58**	87.92	92.90	**76.17**	**85.75**
+ KT-Emb	62.43	94.84	**89.71**	90.9	91.49	**88.02**	92.77	73.29	85.43
+ KG-Emb	61.62	**95.18**	88.97	90.45	91.5	88.01	**93.06**	73.65	85.31

Table 3: Results for RoBERTa on classification and regression (CR) tasks. All results are medians over five runs with different seeds on the development set. To validate our results, we follow RoBERTa (Liu et al., 2019) to finetune starting from the MNLI model for RoBERTa-large instead of the baseline pretrained model on RTE, STS-B and MRPC tasks. Complete results on other pretrained language models can be found in the Appendix B.

可以看到，**KT-Attn 的方式和 KT-Emb 方式在两种规模的模型里面取得的增益也是最多的**，这也符合我这一节想要表述的观点。同时，作者还做了一个 case 分析，在 CoLA （语法检测）任务上，作者分别比对了知识增强前和知识增强后，各个 token 对结果的重要程度贡献，结果如下：

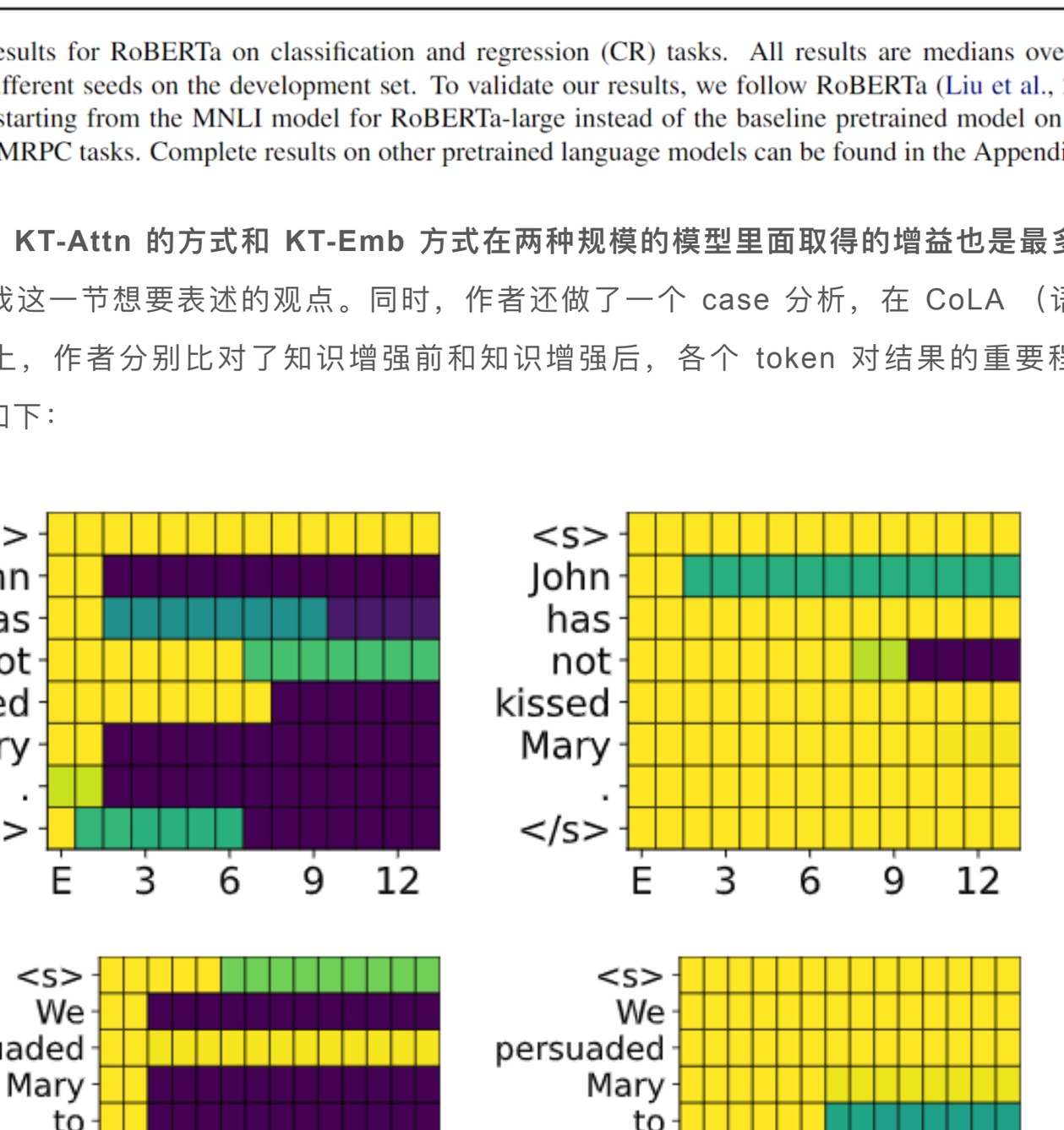


Figure 3: DiffMask plot for CoLA task with RoBERTa-Base model. CoLA task is to predict the linguistic acceptability of a sentence. A purple cell means that the model's corresponding layer thinks the token on the left is not important for the end task and can be ignored. A yellow cell means the opposite and green cells mean neutrality. The left column shows the result from the vanilla RoBERTa-Base and the right column shows **KT-Attn** which is one of our knowledge integrated language model. Clearly, KT-Attn has a better understanding of the end task as it correctly identifies words and phrases such as “not” and “and Sue to stay” which would not change the linguistic acceptability if being ignored.

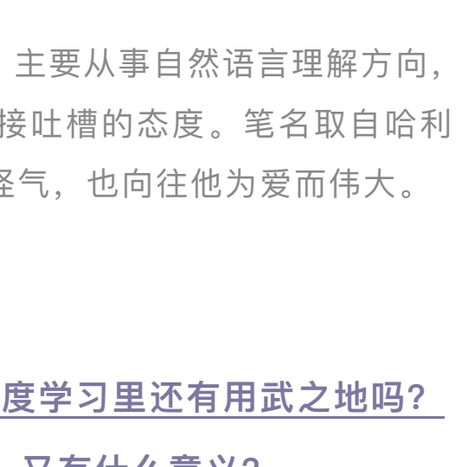
其中，黄色代表重要，紫色代表不重要。可以看到，知识增强之后，第一句 has 和 kissed 最终的重要度显著提升，我们也知道，这两个词对语法正确的判定尤为重要，第二句中 Sue to stay 最终被视作不重要，也符合我们对英文语法的认知。

后面作者也做了各种分析，以及分析了**什么样的知识在什么样的任务上会起到作用**，以及分别评估了在不同的预训练模型上，使用这4种方式分别取得了什么样的效果。我就不再赘述了，感兴趣的读者们可以去看原文。

我在另一篇论文[3]的解读中，也提到了另外一种利用预训练语言模型做图谱 linking 消歧的方式，这里也不再重复解读了。

小结

总而言之，我认为 prompt 作为一种可以**更加充分利用预训练语言模型**的方法，其未来是无限可期的，毕竟现在的预训练语言模型，还是**基于大量的自然语料**训练而成的，也就是说，**预训练语言模型里面的知识是正确的**，所以更需要我们充分抽取、利用它。prompt 还是一个年轻的方向，还有大量的可能性可以去发掘，例如，**怎样将它优雅地应用在序列标注任务上**，就是我一直想要弄明白的问题，我们也会在我们未来的中文理解项目上不断地去探索 prompt 的应用方式，争取也能够做出我们理想中的各种中文文本理解的模型。



萌屋作者：Severus
Severus，在某厂工作的老程序员，主要从事自然语言理解方向，资深死宅，日常愤青，对个人觉得难以理解的同行工作都采取直接吐槽的态度。笔名取自哈利波特系列的斯内普教授，觉得自己也像他那么自闭、刻薄、阴阳怪气，也向往他为爱而伟大。

作品推荐

1. 深度学习，路在何方？
2. 数据还是模型？人类知识在深度学习里还有用武之地吗？
3. 在错误的数据上，刷到 SOTA 又有什么意义？

参考文献

[1] Liu, Pengfei, et al. "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing." arXiv preprint arXiv:2107.13586 (2021).

[2] Xu, Ruochen, et al. "Does Knowledge Help General NLU? An Empirical Study." arXiv preprint arXiv:2109.00563 (2021).

[3] Islam, Sk Mainul, and Sourangshu Bhattacharya. "Scalable End-to-End Training of Knowledge Graph-Enhanced Aspect Embedding for Aspect Level Sentiment Analysis." arXiv preprint arXiv:2108.11656 (2021).