# 大模型分享

文从应用的角度出发，探讨大模型在下游任务中的应用，以及相较于微调模型，大模型在哪些任务上更有优势。 主要从以下几个方面展开：

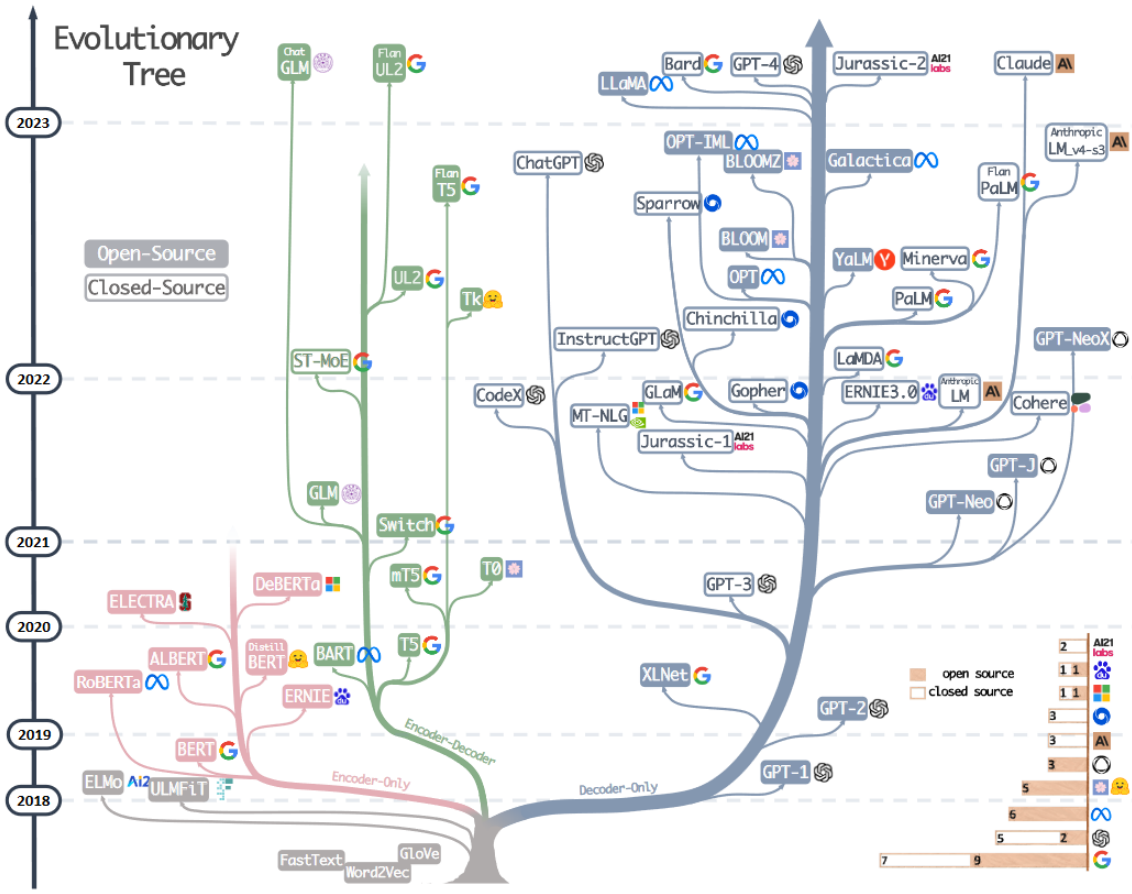
- 大模型速览
- 大模型在下游任务中的适用性讨论
- 具体应用案例

# 1.大模型速览

## 1.1 微调模型（fine-tuned models) 和 大语言模型（LLM）的区别？

微调模型是指，使用预训练数据在较小的语言模型上训练的模型，然后在特定任务上使用更小的数据集进行微调。 大语言模型是指，在大规模的数据集上进行训练的模型，没有对特定任务进行微调。 论文 [1]中从实践角度出发，将参数量少于20B的模型划分为微调模型。

## 1.2 大模型的发展历程




由模型架构进行分类，大模型主要有两种类型：Encoder-Decoder 和 Decoder-Only。Encoder-Decoder 类型主要清华的 GLM 系列、谷歌的 T5 系统、以及早期 Meta 的 BART 等；Decoder-Only 类型的主要代表有 OpenAI 的 GPT 系列、谷歌的 PaLM 等。

Table 1. Summary of Large Language Models.

| | Characteristic | LLMs |
|---|---|---|
| Encoder-Decoder or Encoder-only (BERT-style) | Training: Masked Language Models<br>Model type: Discriminative<br>Pretrain task: Predict masked words | ELMo [80], BERT [28], RoBERTa [65], DistilBERT [90], BioBERT [57], XLM [54], Xlnet [119], ALBERT [55], ELECTRA [24], T5 [84], GLM [123], XLM-E [20], ST-MoE [133], AlexaTM [95] |
| Decoder-only (GPT-style) | Training Autoregressive Language Models<br>Model type: Generative<br>Pretrain task: Predict next word | GPT-3 [16], OPT [126]. PaLM [22], BLOOM [92], MT-NLG [93], GLaM [32],Gopher [83], chinchilla [41], LaMDA [102], GPT-J [107], LLaMA [103], GPT-4 [76], BloombergGPT [117] |

# 2.大模型在下游任务中的适用性讨论

本节从数据角度和任务角度两个视角讨论大模型在下游任务中的适用性讨论。

## 2.1 数据角度

大模型的训练依托于大量且种类丰富的训练数据，如何将训练好的大模型良好的适配到下游任务中，数据是一个重要的考虑角度。从大模型的训练数据角度来看，训

练数据为模型提供了单词和句子的语法、句法和语义的丰富理解，以及识别上下文和生成连贯文本的能力，大模型在各类任务上表现出的强大性能与它选择的训练数据有很大的关联性。因此当将大模型应用到下游任务时，选择的大模型的训练数据在任务类别和领域类别上，应当尽可能的与下游任务具有相似性。

根据下游任务的有标注数据的多少，可以将下游任务分为零样本（zero shot）、少样本（few shot）和样本充足的场景。

- 零样本。在零样本场景下，大模型已被证明优于之前的方法。
- 少样本。在少样本场景下，通过 Prompt Learning 可以有效地将大模型泛化到下游任务，达到与微调模型相媲美的效果。
- 样本充足。在训练样本充足的情况下，微调模型可以更好的拟合数据分布，此条件下微调模型和大模型具有相当的性能。选择微调模型还是大模型需要从性能、计算存储资源等多个角度考虑。

## 2.2 任务角度

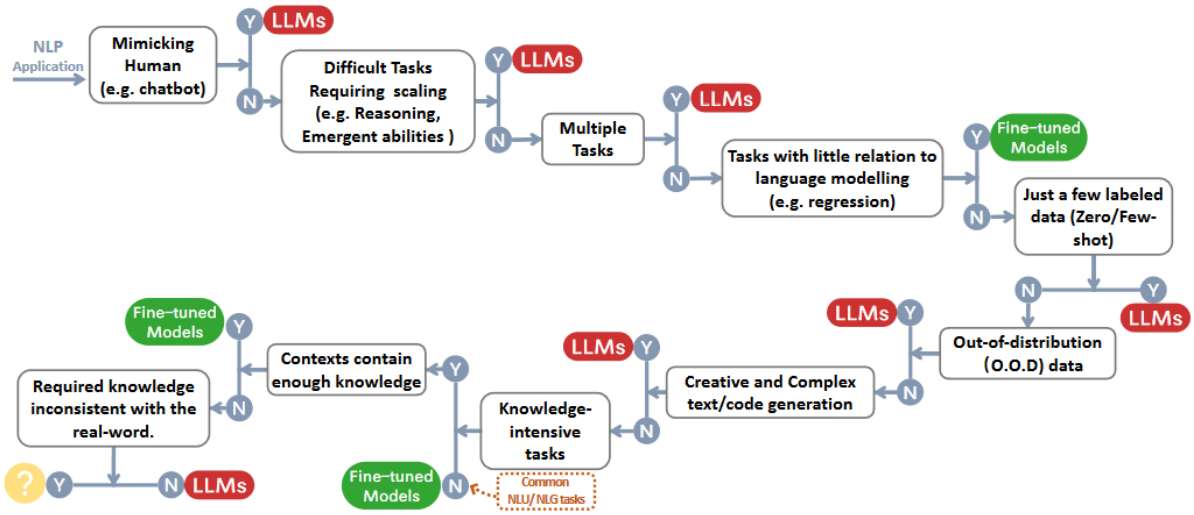



Fig. 2. The decision flow for choosing LLMs or fine-tuned models [2] for user's NLP applications. The decision flow helps users assess whether their downstream NLP applications at hand meet specific conditions and, based on that evaluation, determine whether LLMs or fine-tuned models are the most suitable choice for their applications. During the decision process in the figure, Y means meeting the condition, and N means not meeting the condition. The yellow circle for Y of the last condition means there's no model working well on this kind of application.

### 2.2.1 自然语言理解任务

自然语言理解任务是文本分类、蕴含预测、命名实体识别等自然语言处理任务的基础，从目前的研究来看，在传统的自然语言理解任务上，微调模型相比于大模型通常是更好的选择。例如，在文本分类、情感分析、问答、检索等领域，微调模型相比大模型表现得更好[1]。（需要参考文献说明，在这类任务中，微调模型更优的原因是什么？）以检索任务为例，检索任务需要从成千上万个候选集中挑选出相似度高的文档，或者是对上百甚至成千的粗排集合中重新排序，目前还没有一种合适的

方法将成千上百的候选文本转化为大模型的输入形式。一种尝试是，利用T5模型，以候选文档为输入，计算生成查询的概率，从而用于重排[4]。

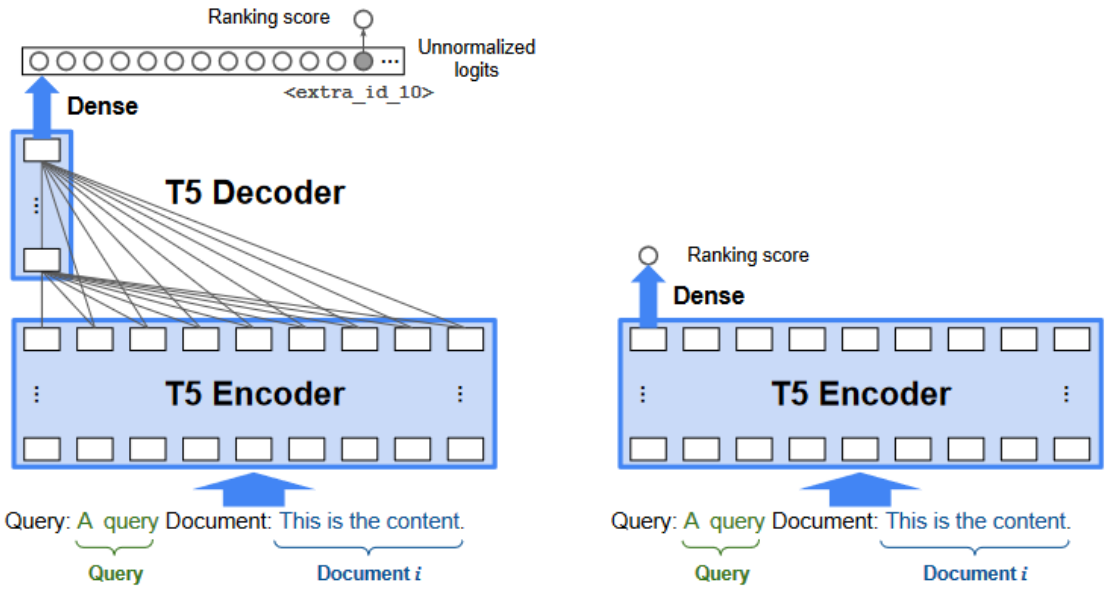



**Figure 1: Model structures of the two variants of RankT5.**

### 2.2.2 自然语言生成任务

在自然语言生成任务上，大模型相较于微调模型具有明显优势。一个自然的原因是，大模型的训练范式是生成式的，且大模型是在多种任务、不同数据集上训练得来，具有强大的语义表示和逻辑推理能力，能够进行连贯正确的输出。例如，摘要生成、机器翻译以及文章生成等开发任务上，大模型都表现得由于微调模型。一个有趣的例子是，GPT-4可以通过leetcode上25%的题目。

## 小结

总体上来说，大模型在自然语言生成任务上表现得更好，在自然语言理解任务上还有长足的提升空间。当然也有一些任务，大模型和微调模型都表现出相当的效果，选用那种方案是应用到下游任务需要可拓展性、计算存储资源等方面进行综合考虑。此外，大模型和下游任务数据的匹配性也是重要的考虑因素。

# 3.具体应用案例（以文本分类为例）

上下文学习（In-context Learning，ICL）

Text Classification via Large Language Models Sun, Xiaofei, et al. "Text Classification via Large Language Models." *arXiv preprint arXiv:2305.08377* (2023).

文本分类任务需要模型具有强大的推理能力，使其能够理解语言的复杂表达，例如从句构造、让步、否定、强化以及反讽等，近期提高大模型推理能力的工作主要侧重于模型的数学推理能力。此外，对于大模型的上下文学习能力，受上下文长度的限制，能够利用的有标签样本数量是很少的，表现得不如微调模型。

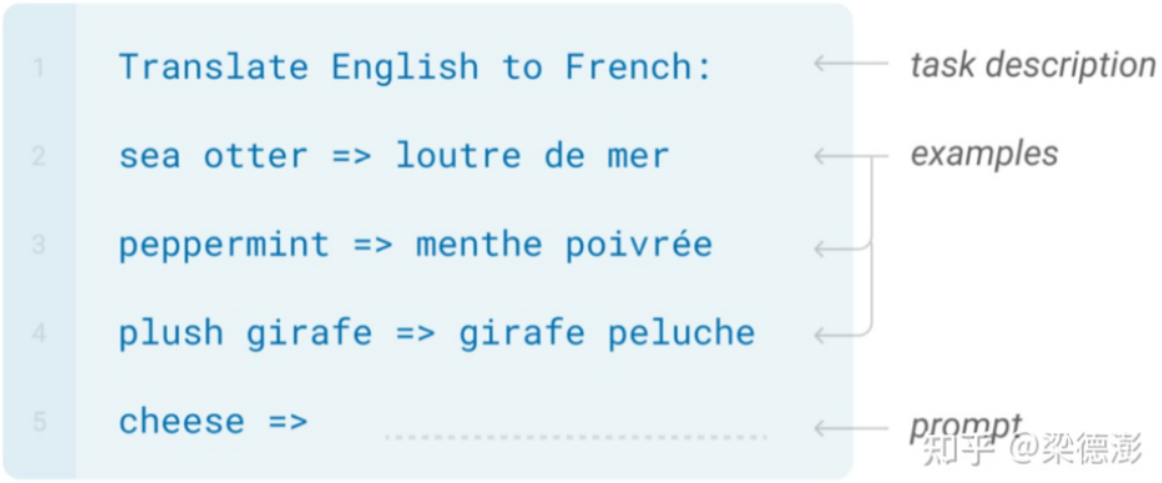
为了激发大模型在文本分类任务上的潜力，论文提出了一种名为 CARP 的推理框架；为了在长度受限的上下文中有效利用有标签的样本，CARP 使用了一个微调的模型在有标签数据集上通过 kNN 搜索用于 上下文学习的示例，从而让模型充分利用 LLM 的泛化能力和有标签数据提供的特定提示信息。

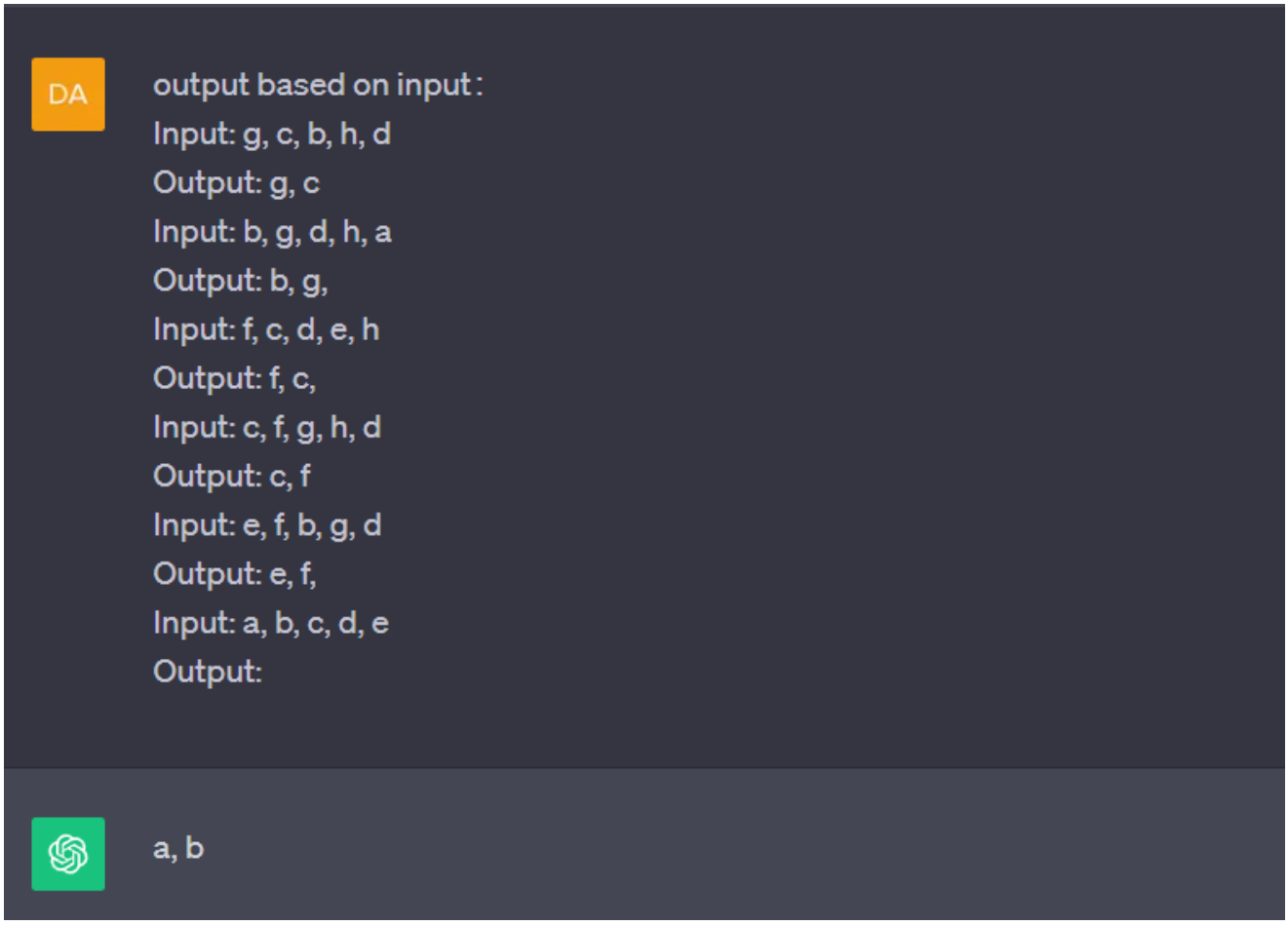
CARP 框架的整理过程为：

- 首先构造提示，使得大模型能够找出文本的表面线索（例如关键词、语气、语义关系等）；
- 然后再次构造提示，并将输入和上一步的线索作为当前的输入，生成推理细节；
- 然后将上面两步生成的线索和推理细节构建新的提示，用于最后的输出。

提示构造（Prompt Construction）由三部分组成：

- 任务描述
- 示例，示例由一些输入和输出对组成。
- 输入（测试输入）。




**示例采样** 随机采样，从训练集中随机采样 k 个样本。 KNN搜索采样；随机采样的缺点是，采样得到的示例与测试输入在语义上不一定是语义相关的。论文中采用的方法类似于编码搜索，使用一个编码模型，对训练集编码，然后对测试输入同样进行编码，进行KNN搜索，对于编码模型，文中采用了 SimCES 和 微调模型。

**一种渐进式的推理策略：** 线索搜集 ==> 推理 ==> 决策。

**Collecting Clues** For a test sequence, clues are local fact evidence such as keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references, etc. The following is an example for clues of an input:

**Input**: *Steers turns in a snappy screenplay that curls at the edges; it's so clever you want to hate it.*

**Clues**: *"snappy", "clever", "want to hate it" are clues for determining the sentiment of the input sentence.*

**Reasoning** For reasoning, the LLM is prompted to go beyond superficial keywords to mine deeper perspectives, considering language phenomenon such as negation, intensification, irony, etc), and piece together local evidence to form the final decision. The following example shows the reasoning process to decide the sentiment of the above example based on the evidence collected:

*1. The phrase "snappy screenplay" implies that the screenplay is of a high quality and is well-crafted.*
*2. The phrase "curls at the edges" implies that the screenplay is cleverly written.*
*3. The phrase "so clever you want to hate it" is a paradoxical statement, which suggests that the sentiment is positive despite the use of the word "hate".*

**零样本（zero-shot）设置下如何收集线索和进行推理？** 因为在零样本设置下，没有可用的示例，因此，直接构造 Prompt，然后输出结果。

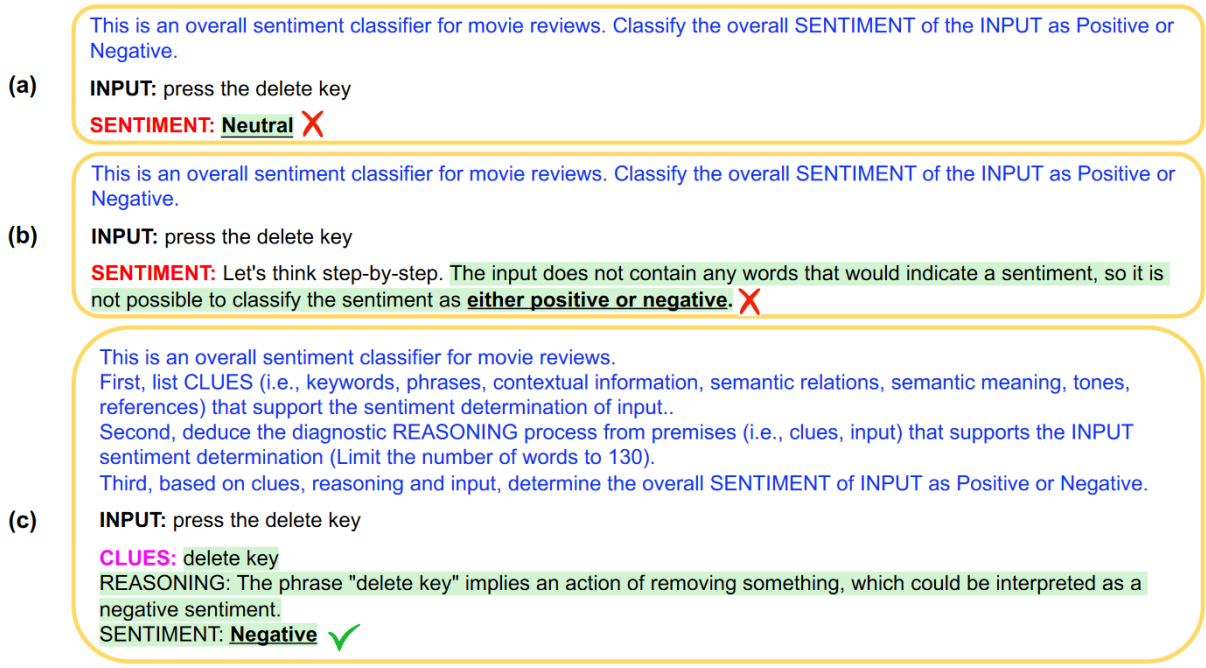



Figure 1: Examples of zero-shot prompting methods for the text classification task: **(a)** represents for the **vanilla** prompting method; **(b)** denotes for the **Chain-of-Thought (CoT)** (Kojima et al., 2022) prompting method; **c** represents for the proposed **CARP** prompting method.

在少样本（few-shot）设置下，给定训练样本（text, label），构建 prompt，然后生成线索。

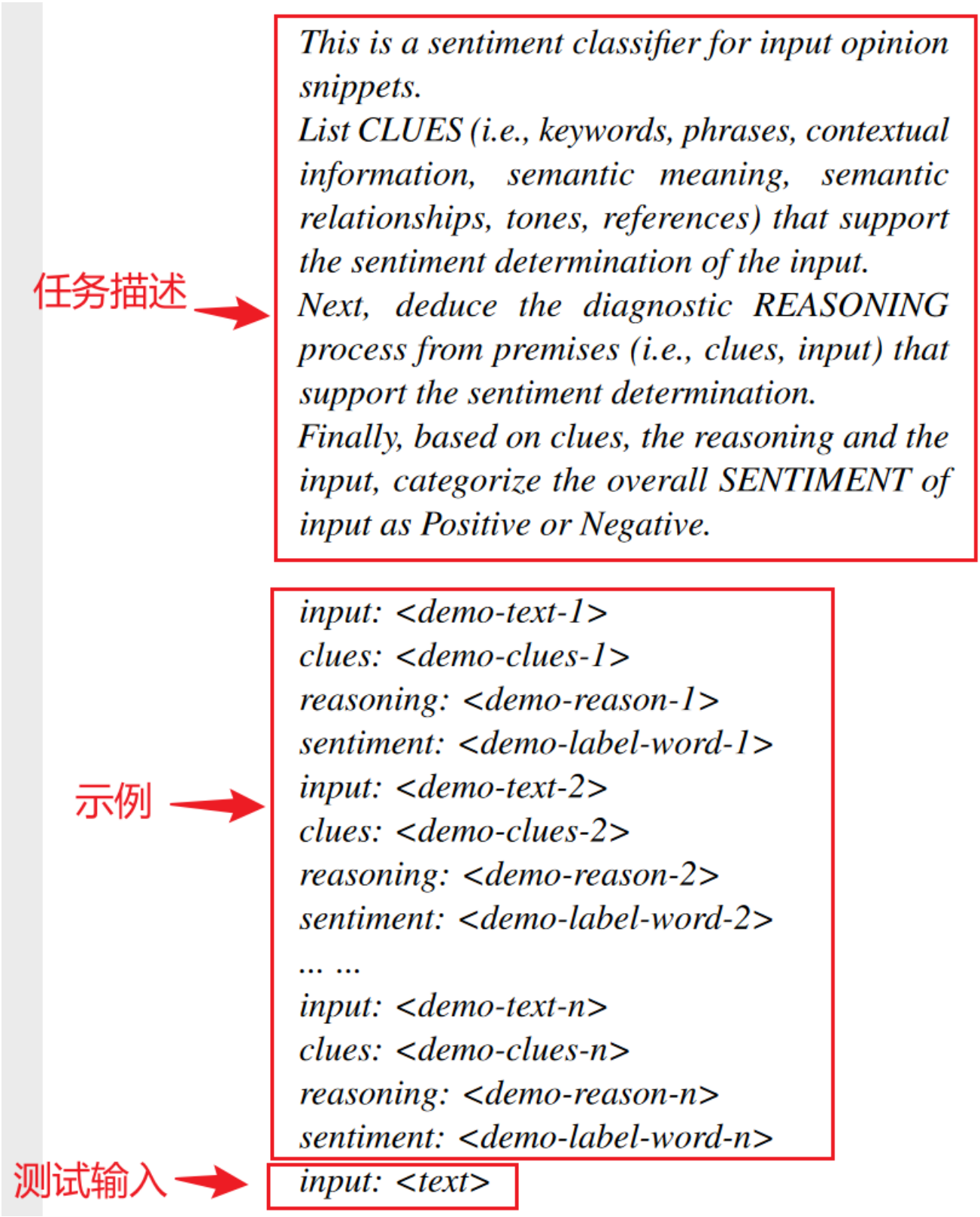
*List CLUES (i.e., keywords, phrases, contextual information, semantic meaning, semantic relationships, tones, references) that support the sentiment determination of the input (limit to 15 words).*
*INPUT: <text>*
*SENTIMENT: <label-word>*

然后根据生成的线索，构建 prompt，生成详细的推理。

*Based on the input and clues, articulate the diagnostic reasoning process that supports the sentiment determination of the input.*
*INPUT: <text>*
*LABEL: <label-word>*
*CLUES: <clues>*
*REASONING:*

最后，为有标签数据全部生成线索和推理，在测试阶段，使用使用 kNN 选择示例，然后构成 prompt 作为输入，使大模型做最后的输出。prompt 的形式为：(text, clues,

reasons, golden label word)




由于是使用大模型的生成任务来做分类任务，使用投票机制来提高准确性，文中考虑了两种投票策略，少数服从多数和加权投票。

实验结果（略）

| | SST-2 | AGNews | R8 | R52 | MR | Average |
|---|---|---|---|---|---|---|
| **Supervised Methods** | | | | | | |
| RoBERTa-Large (Liu et al., 2019) | 95.99 | 95.55 | 97.76 | 96.42 | 91.16 | 95.38 |
| DeBERTa (He et al., 2020) | 94.75 | 95.32 | 98.33 | 96.32 | 90.19 | 94.99 |
| RoBERTa-GCN (Lin et al., 2021) | 95.80 | **95.68*** | 98.2 | 96.1 | 89.7 | 95.10 |
| XLNet (Yang et al., 2019) | **96.10*** | 95.55 | - | - | - | - |
| VLAWE (Ionescu and Butnaru, 2019) | - | - | - | - | **93.3*** | - |
| GCN-SB (Zeng et al., 2022) | - | - | **98.53*** | **96.35*** | 87.59 | - |
| **Zero-shot Setting** | | | | | | |
| Vanilla (Brown et al., 2020) | 91.55 | 90.72 | 90.19 | 89.06 | 88.69 | 90.04 |
| CoT (Kojima et al., 2022) | 92.11 | 91.25 | 90.48 | 91.24 | 89.37 | 90.89 |
| **CARP** | 93.01 | 92.60 | 91.75 | 91.80 | 89.94 | 91.82 |
| **Few-shot Setting ($k$=16)** | | | | | | |
| *Random Sampler* | | | | | | |
| Vanilla (Brown et al., 2020) | 92.36 | 91.74 | 91.58 | 91.56 | 89.15 | 91.28 |
| CoT (Kojima et al., 2022) | 94.56 | 95.02 | 92.49 | 92.03 | 89.91 | 92.80 |
| **CARP** | 96.20 | 95.18 | 97.60 | 96.19 | 90.03 | 95.04 |
| *SimCSE kNN-Sampler* | | | | | | |
| Vanilla (Brown et al., 2020) | 93.90 | 93.50 | 94.36 | 92.40 | 89.59 | 94.05 |
| CoT (Kojima et al., 2022) | 94.21 | 94.28 | 95.07 | 92.98 | 90.27 | 93.69 |
| **CARP** | 95.69 | 95.25 | 97.83 | 96.27 | 90.74 | 95.16 |
| *FT kNN-Sampler* | | | | | | |
| Vanilla (Brown et al., 2020) | 94.01 | 94.14 | 95.57 | 95.79 | 90.90 | 94.08 |
| CoT (Kojima et al., 2022) | 95.48 | 94.89 | 95.59 | 95.89 | 90.17 | 94.40 |
| **CARP** | 96.80 | 95.99 | 98.29 | 96.82 | 91.90 | 95.97 |
| **CARP** (WP Vote) | 97.39 | 96.40 | 98.78 | 96.95 | 92.39 | 96.38 |

Table 2: Accuracy performances of different settings on benchmarks. We report mean and standard deviation results over 5 runs. The GPT-3 denotes `text-davinci-003`. In few-shot experiments, we sample 16 annotated examples ($k$=16) for every test instance. * indicates previous state-of-the-art results. "MJ Vote" is short for majority vote. "WP Vote" denotes weighted probability vote.

# 参考文献


1. Yang, Jingfeng, et al. "Harnessing the power of llms in practice: A survey on chatgpt and beyond." *arXiv preprint arXiv:2304.13712* (2023).
2. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877 – 1901, 2020.
3. https://zhuanlan.zhihu.com/p/143221527
4. Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’ 23), July 23 – 27, 2023, Taipei, Taiwan. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3539618.3592047
5. . https://www.zhihu.com/question/595298808

6. https://www.zhihu.com/people/zibuyu9/answers
7. https://zhuanlan.zhihu.com/p/629087587
8. https://arxiv.org/abs/2305.08377
9. https://zhuanlan.zhihu.com/p/630552148
10. https://new.qq.com/rain/a/20230803A07DSB00
11. https://zhuanlan.zhihu.com/p/635911283
12. https://zhuanlan.zhihu.com/p/606788655?utm_campaign=shareopn&utm_id=0