6. LLM评估方式

文本任务评估指标

1.1 基于词重叠率（方便计算，缺点大部分计算方式无法考虑近义词）

评估方法	简要介绍	对应链接
ROUGE	在单词、短语的角度去衡量两个句子的形似度(简单好计算)	LLM评估方式
BLEU	生成词在Label中的准确率	LLM评估方式
NIST	得到信息量累加起来再除以整个译文的n-gram片段	LLM评估方式
METEOR	考虑了基于整个语料库上的准确率和召回率	LLM评估方式

（1）ROUGE(Recall-Oriented Understudy for Gisting Evaluation)

原始论文🔗： Lin C Y. Rouge: A package for automatic evaluation of summaries[C]//Text summarization branches out. 2004: 74-81.

ROUGE指标是在机器翻译、自动摘要、问答生成等领域常见的评估指标。ROUGE通过将模型生成的摘要或者回答与参考答案（一般是人工生成的）进行比较计算，得到对应的得分。

缺点是这种方法只能在单词、短语的角度去衡量两个句子的形似度。并不能支持同义词、近义词等语意级别去衡量。

ref = "I'm very happy!"

hyp = "I'm very sad!"

hyp1 = "I'm very cheerful!"

hyp1和hyp2的rouge得分是一样的。但显然hyp1才是与ref更相近的。

好处是这种方式计算高效，在忽略近义词等情况下，做到比较合理的判断。

Rouge的实现有**pyrouge, files2rouge, rouge, py-rouge**

这里给出rouge样例代码：

Plain Text

```
# pip install rouge
from rouge import Rouge
hypothesis = "the #### transcript is a written version of each day 's cnn student news program use this transcript to he    lp students with readin g comprehension and vocabulary use the weekly newsquiz to test your knowle dge of storie s you     saw on cnn student news"
reference = "this page includes the show transcript use the transcript to help students with reading comprehension and     vocabulary at the bottom of the page , comment for a chance to be mentioned on cnn student news . y ou must be a teac    her or a student age # # or older to request a mentio n on the cnn student news roll call . the weekly newsquiz tests     studen ts ' knowledge of even ts in the news"
rouger = Rouge()
scores = rouger.get_scores(hypothesis, reference)


# print(socres)
[
  {
    "rouge-1": {
      "f": 0.4786324739396596,
      "p": 0.6363636363636364,
      "r": 0.3835616438356164
    },
    "rouge-2": {
      "f": 0.2608695605353498,
      "p": 0.3488372093023256,
      "r": 0.20833333333333334
    },
    "rouge-l": {
      "f": 0.44705881864636676,
      "p": 0.5277777777777778,
      "r": 0.3877551020408163
    }
  }
]
```

Chinese-ROUGE：

github仓库：https://github.com/Isaac-JL-Chen/rouge_chinese

参考链接：chinese-rouge 知乎教程

```
# pip install rouge-chinese
from rouge_chinese import Rouge
import jieba # you can use any other word cutting library
hypothesis = "###刚刚发声，A股这种情况十分罕见！大聪明逆市抄底330亿，一篇研报引爆全球，市场逻辑生变？"
hypothesis = ' '.join(jieba.cut(hypothesis))
reference = "刚刚过去的这个月，美股总市值暴跌了将近6万亿美元（折合人民币超过40万亿），这背后的原因可能不仅仅是加息这么简单。最近瑞士信贷知名分析师Zoltan Polzsar撰写了一篇极其重要的文章，详细分析了现有世界秩序的崩坏本质以及美国和西方将要采取的应对策略。在该文中，Zoltan Polzsar直指美国通胀的本质和其长期性。同期，A股市场亦出现了大幅杀跌的情况。"
reference = ' '.join(jieba.cut(reference))
rouge = Rouge()
scores = rouge.get_scores(hypothesis, reference)


# output print(score)
[
  {
    "rouge-1": {
      "f": 0.4786324739396596,
      "p": 0.6363636363636364,
      "r": 0.3835616438356164
    },
    "rouge-2": {
      "f": 0.2608695605353498,
      "p": 0.3488372093023256,
      "r": 0.20833333333333334
    },
    "rouge-l": {
      "f": 0.44705881864636676,
      "p": 0.5277777777777778,
      "r": 0.3877551020408163
    }
  }
]
```

（2） BLEU (bilingual evaluation understudy)

原始论文🔗：https://aclanthology.org/P02-1040.pdf

它的总体思想就是准确率，假如给定标准译文reference，神经网络生成的句子是candidate，句子长度为n，**candidate中有m个单词出现在reference，m/n就是bleu的1-gram的计算公式。**

BLEU还有许多变种。根据n-gram可以划分成多种评价指标，常见的指标有BLEU-1、BLEU-2、BLEU-3、BLEU-4四种，其中n-gram指的是连续的单词个数为n。

Python-NLTK-BLEU评分方法：

Plain Text

```
from nltk.translate.bleu_score import sentence_bleu

reference=[['The', 'new', 'translator', 'will', 'stand', 'on', 'the', 'exhibition', 'on', 'behalf', 'of', 'the', 'four', 'times', 'group', 'at', 'the', 'exhibition', 'We', 'will', 'introduce', 'the', 'new', 'star`s', 'business', 'the', 'advantages', 'and', 'the', 'successful', 'cases', 'so', 'that', 'you', 'can', 'understand', 'the', 'new', 'translator', 'more', 'comprehensively', 'We', 'have', 'a', 'stable', 'full-time', 'international', 'team', 'that', 'ensures', 'punctual', 'efficient', 'translation', 'and', 'dubbing', 'and', 'provides', 'a', 'full', 'range', 'of', 'control', 'through', 'the', 'perfect', 'quality', 'control', 'and', 'project', 'management', 'system', 'providing', 'a', 'one-stop', 'service', 'for', 'translation', 'dubbing', 'subtitle', 'production', 'post', 'production', 'broadcasting', 'and', 'ratings', 'surveys'],['The', 'new', 'translator', 'star', 'will', 'represent', 'sida', 'times', 'group', 'in', 'the', 'exhibition', 'when', 'we', 'will', 'introduce', 'the', 'new', 'translator', 'star`s', 'business', 'advantages', 'successful', 'cases', 'and', 'other', 'dimensions', 'so', 'that', 'you', 'can', 'have', 'a', 'more', 'comprehensive', 'understanding', 'of', 'the', 'new', 'translator', 'star', 'We', 'have', 'a', 'stable', 'full-time', 'international', 'team', 'which', 'can', 'ensure', 'timely', 'and', 'efficient', 'translation', 'and', 'dubbing', 'Through', 'perfect', 'quality', 'control', 'and', 'project', 'management', 'system', 'we', 'provide', 'translation', 'dubbing', 'subtitle', 'production', 'post-production', 'broadcasting', 'and', 'rating', 'survey']]

candidate=['New', 'Transtar', 'will', 'present', 'itself', 'at', 'the', 'Exhibition', 'on', 'behalf', 'of', 'StarTimes', 'and', 'we', 'will', 'give', 'a', 'comprehensive', 'introduction', 'of', 'ourselves', 'including', 'the', 'current', 'services', 'we', 'offer', 'the', 'advantages', 'we', 'hold', 'and', 'the', 'projects', 'we', 'have', 'completed', 'to', 'help', 'you', 'understand', 'us', 'more', 'New', 'Transtar', 'boasts', 'of', 'an', 'international', 'team', 'of', 'professionals', 'and', 'is', 'capable', 'of', 'providing', 'fast', 'and', 'quality-guaranteed', 'services', 'including', 'translating', 'dubbing', 'subtitle', 'making', 'post-production', 'broadcasting', 'and', 'collecting', 'of', 'viewership', 'ratings', 'thanks', 'to', 'our', 'strict', 'streamlined', 'and', 'developed', 'quality', 'control', 'and', 'project', 'management', 'system']

score = sentence_bleu(reference, candidate)
```

ROUGE指标与BLEU指标非常类似，均可用来衡量生成结果和标准结果的匹配程度，不同的是ROUGE基于召回率，BLEU更看重准确率。

（3）NIST(National Institute of standards and Technology)

最主要的是引入了每个n-gram的信息量(information) 的概念。BLEU算法只是单纯的将n-gram的数目加起来，而nist是在得到信息量累加起来再除以整个译文的n-gram片段数目。这样相当于对于一些出现少的重点的词权重就给的大了。

$$Info(w_1 \dots w_n) = \log_2 \left(\frac{\text{the \# of occurrences of } w_1 \dots w_{n-1}}{\text{the \# of occurrences of } w_1 \dots w_n} \right)$$

分母是n元词在参考译文中出现的次数，分子是对应的n-1元词在参考译文中的出现次数（对于一元词汇，分子的取值就是整个参考译文的长度）。这里之所以这样算，应该是考虑到出现次数少的就是重点词这样的一个思路。

$$scorce = \sum_{n=1}^{N} \left\{ \sum_{\substack{\text{all } w_1 \dots w_n \\ \text{that co-occur}}} Info(w_1 \dots w_n) \Big/ \sum_{\substack{\text{all } w_1 \dots w_n \\ \text{in sysoutput}}} (1) \right\} \bullet \exp\left\{ \beta \log^2 \left[\min\left(\frac{L_{sys}}{\overline{L}_{ref}}, 1 \right) \right] \right\}$$

N一般取5，β是一个常数，在Lsys/Lref=2/3 时，β使得长度罚分率为0.5，它是个经验值。Lref 是参考答案的平均长度（注意L的上方有一个平均符号），Lsys是译文的长度。

```
from nltk.translate.nist_score import sentence_nist
hypothesis1 = ['It', 'is', 'a', 'guide', 'to', 'action', 'which', 'ensures', 'that', 'the', 'military', 'always','obeys', 'the', 'commands', 'of', 'the', 'party']
hypothesis2 = ['It', 'is', 'to', 'insure', 'the', 'troops','forever', 'hearing', 'the', 'activity', 'guidebook','that', 'party', 'direct']
reference1 = ['It', 'is', 'a', 'guide', 'to', 'action', 'that',
...               'ensures', 'that', 'the', 'military', 'will', 'forever',
...               'heed', 'Party', 'commands']
reference2 = ['It', 'is', 'the', 'guiding', 'principle', 'which',
...               'guarantees', 'the', 'military', 'forces', 'always',
...               'being', 'under', 'the', 'command', 'of', 'the',
...               'Party']
 reference3 = ['It', 'is', 'the', 'practical', 'guide', 'for', 'the',
...               'army', 'always', 'to', 'heed', 'the', 'directions',
...               'of', 'the', 'party']
sentence_nist([reference1, reference2, reference3], hypothesis1)
>>> 3.3709
sentence_nist([reference1, reference2, reference3], hypothesis2)
>>> 1.4619
```

（4）METEOR

原文链接🔗：http://www.cs.cmu.edu/~alavie/METEOR/pdf/Banerjee-Lavie-2005-METEOR.pdf

该指标考虑了基于整个**语料库**上的准确率和召回率，而最终得出测度。

METEOR扩展了BLEU有关“共现”的概念，提出了三个统计共现次数的模块：

- 一是“绝对”模块（"exact" module），即统计待测语句与参考语句中绝对一致单词的共现次数；
- 二是“波特词干”模块（porter stem module），即基于波特词干算法计算待测译文与参考译文中词干相同的词语**“变体”**的共现次数，如happy和happiness将在此模块中被认定为共现词；
- 三是“WN同义词”模块（WN synonymy module），即基于WordNet词典匹配待测译文与参考译文中的**同义词**，计入共现次数，如sunlight与sunshine。

同时METEOR将**词序**纳入评估范畴，**设立基于词序变化的罚分机制，当待测译文词序与参考译文不同时，进行适当的罚分。**最终基于共现次数计算准确率、召回率与F值，并考虑罚分最终得到待测译文的METEOR值。

```
from nltk.translate.meteor_score import meteor_score

reference3 = '我 说 这 是 怎 么 回 事, 原 来 明 天 要 放 假 了'
reference2 = '我 说 这 是 怎 么 回 事'
hypothesis2 = '我 说 这 是 啥 呢 我 说 这 是 啥 呢'
# reference3：参考译文
# hypothesis2：生成的文本
res = round(meteor_score([reference3, reference2], hypothesis2), 4)
print(res)

# output:
0.4725
```

1.2 Data to Text 指标（输入三元组生成语句后评估）[故事生成]

data to text 和翻译、摘要等生成式任务最大的不同是，**input是类似于table或者三元组等其他形式的数据**。在评估生成结果时，我们还需要考虑文本是否准确的涵盖了data的信息。

评估方法	简介	实现
relation generation (RG)	指从生成的句子中抽取出关系，然后对比有多少关系也出现在了source中	哈佛代码实现：https://github.com/harvardnlp/data2text
content selection (CS)	一般指data当中的内容有多少出现在了生成的句子中，一般有precision和recall两个指标	
content ordering (CO)	使用归一化 Damerau–Levenshtein距离计算生成句和参考句的“sequence of records”	

1.3 image caption（图片标注评价指标）[图生文]

评估方法	简介	实现

CIDEr	将每个句子都看作“文档”，将其表示成 tf-idf向量的形式，计算参考 caption 与模型生成的 caption 的余弦相似度	https://github.com/wangleihitcs/CaptionMetrics
SPICE	使用基于图的语义表示来编码 caption 中的 objects, attributes 和 relationships。	https://github.com/peteanderson80/SPICE (Java)

1.4 词向量评价指标

词向量则是通过**Word2Vec、Sent2Vec等方法将句子转换为向量表示**，这样一个句子就被映射到一个低维空间，句向量在一定程度上表征了其含义，在通过**余弦相似度**等方法就可以计算两个句子之间的相似程度。

使用词向量的好处是，可以一定程度上增加答案的多样性，因为这里大多采用词语相似度进行表征，相比词重叠中要求出现完全相同的词语，限制降低了很多。**（但是很少有Paper用到，因为计算困难，且转换模型也需要考量）**

评估方式	简介	实现
Greedy Matching	主要关注两句话之间最相似的那些词语，即关键词	https://blog.csdn.net/qq_33772192/article/details/88936473
Embedding Average	直接使用句向量计算真实响应和生成响应之间的相似度，而句向量则是每个词向量加权平均而来	https://blog.csdn.net/qq_36332660/article/details/128160295 (from nlgeval import NLGEval)
Vector Extrema	句向量采用向量极值法进行计算，然后计算真实响应和生成响应之间的相似度	https://blog.csdn.net/qq_33772192/article/details/88948555

1.5 基于语言模型的方法

基于语言模型的评价指标**通过使用语言模型，来计算参照文本和生成文本的相似度**，主要包括BertScore、BARTScore、MoverScore、BLEURT及Perplexity。

评估方式	简介	实现
BertScore	生成文本和参照文本，分别用bert提取特征，然后对两个句子的每一个词分别计算内积，可以得到一个相似性矩阵	https://huggingface.co/spaces/evaluate-metric/bertscore
BARTScore	采用无监督学习对生成文本的不同方面进行评估。	https://github.com/neulab/BARTScore
MoverScore	对BertScore的改进	https://github.com/AIPHES/emnlp19-moverscore
BELURT	**通过预训练结合人工评估数据的微调来同时满足度量方法的鲁棒性和表达度**	https://huggingface.co/spaces/evaluate-metric/bleurt
Perplexity（困惑度）	用来评估生成文本的流畅性，其值越小，说明语言模型的建模能力越好，即生成的文本越接近自然语言。	https://huggingface.co/spaces/evaluate-metric/perplexity
MoverScore	**采用了推土机距离计算和参考句的相似程度，而不是单纯的像bertscore只考虑最相似的词的距离。这样我觉得可以防止候选句的某一个词过于强大**	https://arxiv.org/abs/1909.02622

1.6 总结

- **BLEU，ROUGE等评价指标依然是主流的评价方式(计算简单)**
- 从短句惩罚、重复、重要信息缺失、多样化等方面，衍生出例如METEOR、SPICE等评价指标
- 以**bertscore**为代表的评价指标近年来受到广泛的关注，与人工评价的相关性也越来越高

大模型评估（对话能力，推理）

- 当模型仅被用于执行单一任务的时候，我们可以出考题（benchmark）来评估其能力；
- 但当 LLM 成为一个综合模型，我们想要**将其应用于多类型任务时候，就意味着需要进行多维度考察**；

- 而当其成为一个对话应用的时候，那么我们更希望可以对其进行拟人化的考察，除了硬性能力，还希望可以 check 其是否更像人。

2.0 Holistic Evaluation of Language Models(HELM)

- 广泛的覆盖面和对不完整性的认识。我们定义了我们理想中想要评估的场景的分类法，选择场景和指标来覆盖空间并明确缺失的内容。
- 多指标测量。HELM不是专注于诸如准确性之类的孤立指标，而是针对每个场景同时测量多个指标（例如，准确性、稳健性、校准、效率），从而允许进行权衡分析。
- 标准化。我们使用相同的适应策略（例如，提示）评估HELM在相同场景下可以访问的所有模型，从而允许进行受控比较。
    - https://crfm.stanford.edu/helm/latest/
    - https://github.com/stanford-crfm/helm
    - Holistic Evaluation of Language Models (HELM), a framework to increase the transparency of language models (https://arxiv.org/abs/2211.09110). - GitHub - stanford-crfm/helm: Holistic Evaluation of ...
    - https://github.com/stanford-crfm/helm

2.1 UCB 引入 Elo 进行对抗评估

Elo等级分制度（Elo rating system）是一种计算玩家相对技能水平的方法，广泛应用在竞技游戏和各类运动当中。其中，Elo评分越高，那么就说明这个玩家越厉害。

这个**Elo评分的数值是绝对的**。也就是说，当未来加入新的聊天机器人时，我们依然可以直接通过Elo的评分来判断哪个聊天机器人更厉害。

具体来说，如果玩家A的评分为Ra，玩家B的评分为Rb，玩家A获胜概率的精确公式（使用以10为底的logistic曲线）为：

$$E_{\mathrm{A}} = \frac{1}{1 + 10^{(R_{\mathrm{B}} - R_{\mathrm{A}})/400}} .$$

然后，玩家的评分会在每场对战后线性更新。

假设玩家A（评分为Ra）预计获得Ea分，但实际获得Sa分。更新该玩家评分的公式为：

$$R'_{\mathrm{A}} = R_{\mathrm{A}} + K \cdot (S_{\mathrm{A}} - E_{\mathrm{A}}) .$$

https://chat.lmsys.org/?arena

https://36kr.com/p/2243109425885057

2.2 爱丁堡大学的 Fuyao 在研究从推理上评价模型能力

Model	Param.	Type	GSM8K	MATH	MMLU	BBH	HumanEval	C-Eval	TheoremQA
gpt-4	?	RLHF	92	42.5	86.4	–	67	68.7*	43.4
claude-v1.3	?	RLHF	81.8*	–	74.8*	67.3*	–	54.2*	24.9
PaLM-2	?	Base	80.7	34.3	78.3	78.1	–	–	31.8
gpt-3.5-turbo	?	RLHF	74.9*	–	67.3*	70.1*	48.1	54.4*	30.2
claude-instant	?	RLHF	70.8*	–	–	66.9*	–	45.9*	23.6
text-davinci-003	?	RLHF	–	–	64.6	70.7	–	–	22.8
code-davinci-002	?	Base	66.6	19.1	64.5	73.7	47	–	–
text-davinci-002	?	SIFT	55.4	–	60	67.2	–	–	16.6

Minerva	540B	SIFT	58.8	33.6	–	–	–	–	–
Flan–PaLM	540B	SIFT	–	–	70.9	66.3	–	–	–
Flan–U–PaLM	540B	SIFT	–	–	69.8	64.9	–	–	–
PaLM	540B	Base	56.9	8.8	62.9	62	26.2	–	–
LLaMA	65B	Base	50.9	10.6	63.4	–	23.7	38.8*	–
PaLM	64B	Base	52.4	4.4	49	42.3	–	–	–
LLaMA	33B	Base	35.6	7.1	57.8	–	21.7	–	–
InstructCodeT5+	16B	SIFT	–	–	–	–	35	–	11.6
StarCoder	15B	Base	8.4	15.1	33.9	–	33.6	–	12.2
Vicuna	13B	SIFT	–	–	–	–	–	–	12.9
LLaMA	13B	Base	17.8	3.9	46.9	–	15.8	–	–
Flan–T5	11B	SIFT	16.1*	–	48.6	41.4	–	–	–
Alpaca	7B	SIFT	–	–	–	–	–	–	13.5
LLaMA	7B	Base	11	2.9	35.1	–	10.5	–	–

Flan-T5	3B	SIFT	13.5*	–	45.5	35.2	–	–	–

Base 表示预训练的检查点。SIFT 表示监督指令微调后的检查点。RLHF 表示从人类反馈中强化学习之后的检查点。标有星号 * 的数字来自我们自己的运行，否则来自我们在下面解释的多个来源。

与HeLM和其他评估有何不同?

- **HeLM 使用仅回答提示，我们使用链式思维促进**
- **HeLM 评估一切。该方法只关注复杂推理，这是 LLM 能力的关键区别。**

挑战任务:

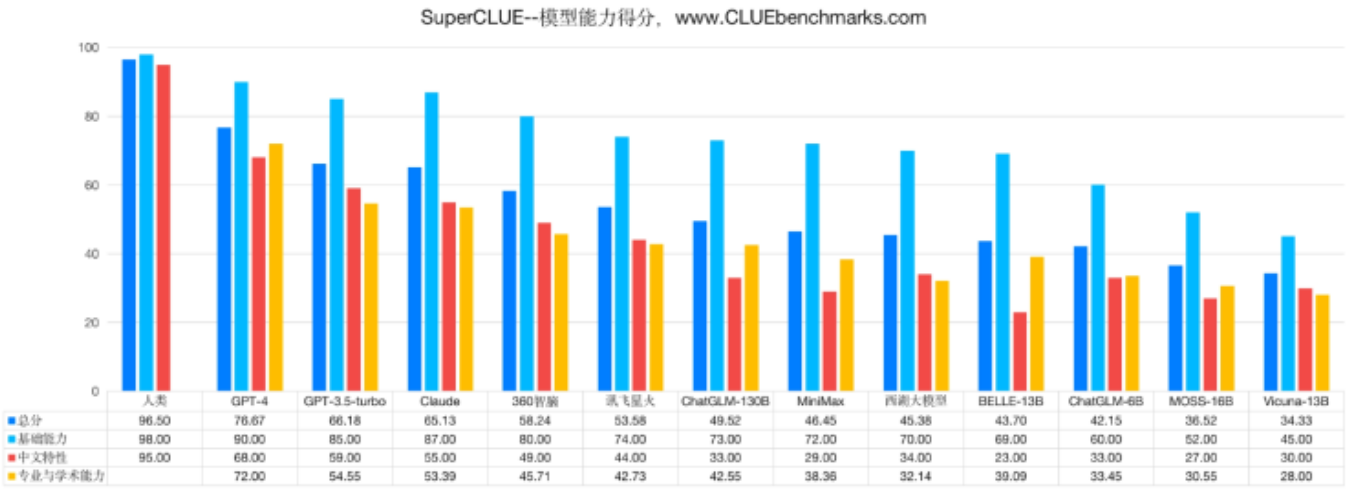
- MMLU：高中和大学知识
- GSM8K：小学数学。-- 在与 LLM 交互时，此数据集的性能改进直接转化为日常数学能力
- MATH（难！）：非常难的数学和自然科学。当前所有模型都在挣扎。
- BBH：27 个硬推理问题的集合 https://github.com/FranxYao/chain-of-thought-hub/tree/main
- HumanEval：用于评估编码能力的经典数据集。
- C-Eval：中文52门学科知识测试合集
- TheoremQA（困难！）：由 STEM 定理驱动的问答数据集

2.3 CLUE发布SuperCLUE进行中文通用大模型综合性评测基准

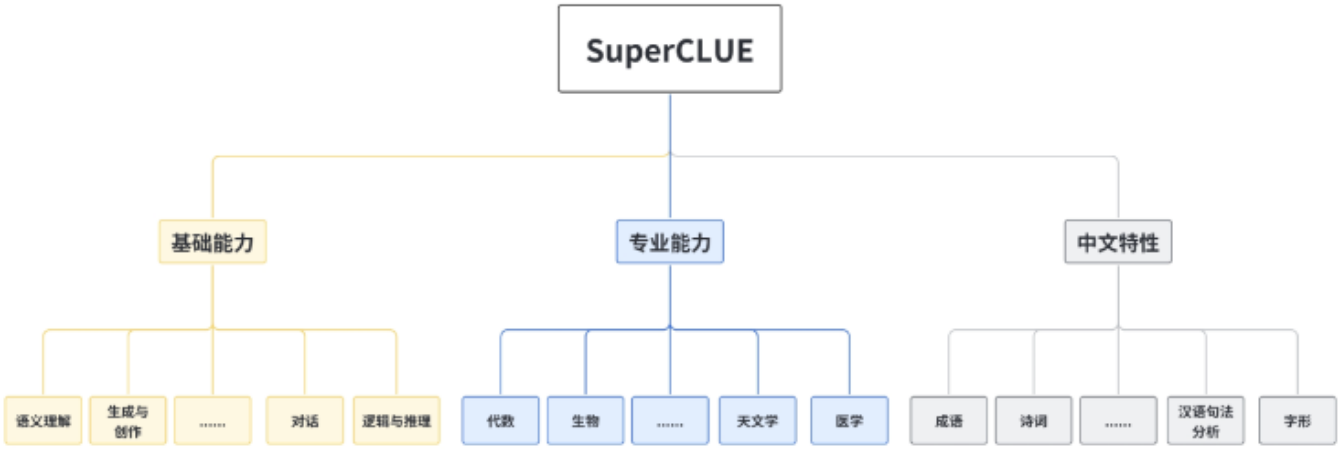
中文通用大模型综合性测评基准（SuperCLUE），是针对中文可用的通用大模型的一个测评基准。

它主要要回答的问题是：在当前通用大模型大力发展的情况下，中文大模型的效果情况。包括但不限于：这些模型哪些相对效果情况、相较于国际上的代表性模型做到了什么程度、 这些模型与人类的效果对比如何?

它尝试在一系列国内外代表性的模型上使用多个维度能力进行测试。**SuperCLUE，是中文语言理解测评基准（CLUE）在通用人工智能时代的进一步发展。**



	人类	GPT-4	GPT-3.5-turbo	Claude	360智脑	讯飞星火	ChatGLM-130B	MiniMax	西湖大模型	BELLE-13B	ChatGLM-6B	MOSS-16B	Vicuna-13B
总分	96.50	76.67	66.18	65.13	58.24	53.58	49.52	46.45	45.38	43.70	42.15	36.52	34.33
基础能力	98.00	90.00	85.00	87.00	80.00	74.00	73.00	72.00	70.00	69.00	60.00	52.00	45.00
中文特性	95.00	68.00	59.00	55.00	49.00	44.00	33.00	29.00	34.00	23.00	33.00	27.00	30.00
专业与学术能力		72.00	54.55	53.39	45.71	42.73	42.55	38.38	32.14	39.09	33.45	30.55	28.00



SuperCLUE的特点:

1）多个维度能力考察（3大类70+子能力）：从三个不同角度对中文大模型进行测试，以考察模型的综合能力；并且每一个子能力又含有十项或以上不同的细分能力。

2）自动化测评（一键测评）：通过自动化测评方式以相对客观形式测试不同模型的效果，可以一键对大模型进行测评。

3）广泛的代表性模型（9个模型）：选取了多个国内外有代表性的可用的模型进行测评，以反应国内大模型的发展现状并了解与国际领先模型的差距或相对优劣势。

4）人类基准：在通用人工智能发展的情况下，也提供了模型相对于人类效果的指标对比。

评测方式

1、统一prompt：针对每一个题目，构造了统一的prompt供模型和人类使用；

2、预测：系统使用模型进行预测，要求模型选取ABCD中的某一个选项；

3、打分：如果模型的回答不是标准的答案，而是一段文字，系统会采取特定的策略自动提取出模型的答案。该策略结合模型的表现进行优化和完善。

（注：当无法提取有效答案的时候，则表明模型没有按照人类的要求做题，未正确理解指令，则认为模型回答错误。）

```
示例：
语义理解：
    两个男人正常交谈，其中一个男人夸赞对方办事能力强，对方回答“哪里，哪里”。这里的“哪里，哪里”是什么意思？
    A. 讲话十分含糊不清。
    B. 要求说出具体的优点。
    C. 表达自己的谦虚。
    D. 挑衅对方。

逻辑与推理：
    小明的妻子生了一对双胞胎。以下哪个推论是正确的？
    A. 小明家里一共有三个孩子。
    B. 小明家里一共有两个孩子。
    C. 小明家里既有男孩子也有女孩子。
    D. 无法确定小明家里孩子的具体情况。
```

2.4 C-Eval

https://cevalbenchmark.com/static/leaderboard.html

C-Eval 是一个针对基础模型的综合中文评估套件。它包含 13948 道多项选择题，涵盖 52 个不同的学科和四个难度级别https://github.com/SJTU-LIT/ceval

#	模型名称	发布机构	提交时间	平均 ▼	平均 (Hard)	STEM	社会科学	人文科学	其他
0	GPT-4	OpenAI	2023/5/15	68.7	54.9	67.1	77.6	64.5	67.8
1	InternLM	SenseTime & Shanghai AI Laboratory (equal contribution)	2023/6/1	62.7	46	58.1	76.7	64.6	56.4
2	ChatGPT	OpenAI	2023/5/15	54.4	41.4	52.9	61.8	50.9	53.6
3	Claude-v1.3	Anthropic	2023/5/15	54.2	39	51.9	61.7	52.1	53.7
4	Claude-instant-v1.0	Anthropic	2023/5/15	45.9	35.5	43.1	53.8	44.2	45.4
5	GLM-130B	Tsinghua	2023/5/15	40.3	30.3	34.8	48.7	43.3	39.8
6	CubeLM-13B	CubeLM	2023/6/5	40.2	27.3	34.1	49.7	43.4	39.6
7	Bloomz-mt	BigScience	2023/5/15	39	30.4	35.3	45.1	40.5	38.5
8	LLaMA-65B	Meta	2023/5/15	38.8	31.7	37.8	45.6	36.1	37.1
9	ChatGLM-6B	Tsinghua	2023/5/15	34.5	23.1	30.4	39.6	37.4	34.5
10	Chinese LLaMA-13B	Cui et al.	2023/5/15	33.3	27.3	31.6	37.2	33.6	32.8
11	MOSS	Fudan	2023/5/15	31.1	24	28.6	36.8	31	30.3
12	Chinese Alpaca-13B	Cui et al.	2023/5/15	26.7	27.1	26	27.2	27.8	26.4

测试数据：

Plain Text

```
id: 1
question: 25 °C时，将pH=2的强酸溶液与pH=13的强碱溶液混合，所得混合液的pH=11，则强酸溶液与强碱溶液 的体积比是(忽略混合后溶液的体积变化)____
A: 11:1
B: 9:1
C: 1:11
D: 1:9
answer: B
explantion:
1. pH=13的强碱溶液中c(OH−)=0.1mol/L, pH=2的强酸溶液中c(H+)=0.01mol/L，酸碱混合后pH=11，即c(OH−)=0.001mol/L。
2. 设强酸和强碱溶液的体积分别为x和y，则：c(OH−)=(0.1y−0.01x)/(x+y)=0.001，解得x:y=9:1。
```

2.5 KoLA 清华评测排行

清华大学提出KoLA评测基准，从**掌握和利用世界知识的角度**，衡量大语言模型的表现。

KoLA基于19个关注实体、概念和事件的任务。参考了Bloom认知体系，KoLA从知识的记忆、理解、应用和创造4个层级，从深度而非广度去衡量大语言模型处理世界知识的能力。

- 论文链接: https://arxiv.org/pdf/2306.09296.pdf
- 评测榜单: https://kola.xlore.cn

LEADERBOARD

Season S1(2023-6-01-2023-6-30) week S1(6-3)

Rank	Model	Team	Rank Last Week	GitHub/URL	Total Points	1-1	1-2	1-3	2-1	2-2	2-3
1	GPT-4	OpenAI	--		100.0	51.4	55.5	54.6	63.5	42.9	46.0
2	GPT-3.5-turbo	OpenAI	--		79.9	41.7	47.6	42.0	37.5	43.8	44.8
3	InstructGPT davinci v2	OpenAI	--		71.8	30.8	37.2	32.4	26.6	42.5	36.5
4	Cohere-command	Cohere	1		65.2	46.6	42.6	56.8	33.1	41.2	40.6
5	FLAN-UL2	Google	2		59.4	41.3	31.9	53.0	52.7	41.2	47.8
6	FLAN-T5	Google	--		54.9	44.1	39.9	49.6	57.0	42.1	43.6
7	J2-Jumbo-Instruct	AI21	--		52.4	23.0	24.0	17.6	20.1	15.8	24.5
8	ChatGLM	Tsinghua/Zhipu	--		47.0	27.8	44.5	36.1	23.3	42.1	46.6
9	InstructGPT curie v1	OpenAI	--		44.2	19.0	33.1	33.1	22.3	34.9	35.9
10	LLaMa	MetaAI	3		42.0	15.5	16.7	9.9	14.6	10.3	10.7