# 2.2.4 示例提示选择器

示例提示对模型输出结果产生的影响是非常显著的。这点在实际操作中的反馈是明确无误的。但问题在于，我们可能有大量这样的示例，我们不可能全部输入给模型。而且，尝试适应所有示例可能会很快变得非常昂贵，尤其是在计算资源和时间上。这就是示例选择器发挥作用的地方，它帮助我们选择最适合的示例来提示模型。

以金融财报的摘要为例，如果举例的摘要过长，成本会变得很昂贵。大量、冗长的例子可能会占用模型可处理的 Token 数量，这使得模型无法充分理解和处理用户的真正输入，从而影响输出质量。

幸运的是，LangChain 的示例提示选择器（Example Selector）提供了一套工具，来解决这个问题。这些工具能基于策略选择合适的例子，如根据例子的长度、输入与例子之间的n-gram重叠分数来评估其相似度打分、找到与输入具有最大余弦相似度的例子, 以及多样性等因素来选择例子, 从而保持提示成本的相对稳定。

根据长度选择示例，是很普遍和现实的需求，以下是根据长度选择示例的代码：

from langchain.prompts import PromptTemplate  
from langchain.prompts import FewShotPromptTemplate  
from langchain.prompts.example\_selector import LengthBasedExampleSelector  
  
examples = [  
 {"word": "happy", "antonym": "sad"},  
 {"word": "tall", "antonym": "short"},  
 {"word": "energetic", "antonym": "lethargic"},  
 {"word": "sunny", "antonym": "gloomy"},  
 {"word": "windy", "antonym": "calm"},  
]  
  
example\_prompt = PromptTemplate(  
 input\_variables=["word", "antonym"],  
 template="Word: {word}\nAntonym: {antonym}",  
)  
  
example\_selector = LengthBasedExampleSelector(  
 examples=examples,   
 example\_prompt=example\_prompt,   
 max\_length=25,  
)  
  
dynamic\_prompt = FewShotPromptTemplate(  
 example\_selector=example\_selector,  
 example\_prompt=example\_prompt,  
 prefix="Give the antonym of every input",  
 suffix="Word: {adjective}\nAntonym:",   
 input\_variables=["adjective"],  
)  
  
print(dynamic\_prompt.format(adjective="big"))

todo: 返回的结果