第五章 推断

在这一章中，我们将通过一个故事，引领你了解如何从产品评价和新闻文章中推导出情感和主题。

让我们先想象一下，你是一名初创公司的数据分析师，你的任务是从各种产品评论和新闻文章中提取出关键的情感和主题。这些任务包括了标签提取、实体提取、以及理解文本的情感等等。在传统的机器学习流程中，你需要收集标签化的数据集、训练模型、确定如何在云端部署模型并进行推断。尽管这种方式可能会产生不错的效果，但完成这一全流程需要耗费大量的时间和精力。而且，每一个任务，比如情感分析、实体提取等等，都需要训练和部署单独的模型。

然而，就在你准备投入繁重工作的时候，你发现了大型语言模型（LLM）。LLM 的一个明显优点是，对于许多这样的任务，你只需要编写一个 Prompt，就可以开始生成结果，大大减轻了你的工作负担。这个发现像是找到了一把神奇的钥匙，让应用程序开发的速度加快了许多。最令你兴奋的是，你可以仅仅使用一个模型和一个 API 来执行许多不同的任务，无需再纠结如何训练和部署许多不同的模型。

让我们开始这一章的学习，一起探索如何利用 LLM 加快我们的工作进程，提高我们的工作效率。

一、情感推断

1.1 情感倾向分析

让我们以一则电商平台上的台灯评论为例，通过此例，我们将学习如何对评论进行情感二分类（正面/负面）。

```
lamp_review = """
我需要一盏漂亮的卧室灯，这款灯具有额外的储物功能，价格也不算太高。\
我很快就收到了它。在运输过程中，我们的灯绳断了，但是公司很乐意寄送了一个新的。\
几天后就收到了。这款灯很容易组装。我发现少了一个零件，于是联系了他们的客服，他们很快就给我寄来了缺失的零件！\
在我看来，Lumina 是一家非常关心顾客和产品的优秀公司！
"""
```

接下来，我们将尝试编写一个 Prompt ，用以分类这条商品评论的情感。如果我们想让系统解析这条评论的情感倾向，只需编写"以下商品评论的情感倾向是什么？"这样的 Prompt ，再加上一些标准的分隔符和评论文本等。

然后，我们将这个程序运行一遍。结果表明，这条商品评论的情感倾向是正面的，这似乎非常准确。尽管这款台灯并非完美无缺，但是这位顾客对它似乎相当满意。这个公司看起来非常重视客户体验和产品质量，因此，认定评论的情感倾向为正面似乎是正确的判断。

```
from tool import get_completion

prompt = f"""
以下用三个反引号分隔的产品评论的情感是什么？

评论文本: ```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
情感是积极的。
```

如果你想要给出更简洁的答案，以便更容易进行后期处理，可以在上述 Prompt 基础上添加另一个指令：*用一个单词回答：「正面」或「负面」*。这样就只会打印出 “正面” 这个单词，这使得输出更加统一，方便后续处理。

```
prompt = f"""
以下用三个反引号分隔的产品评论的情感是什么？

用一个单词回答：「正面」或「负面」。

评论文本：```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
正面
```

1.2 识别情感类型

接下来，我们将继续使用之前的台灯评论，但这次我们会试用一个新的 Prompt 。我们希望模型能够识别出评论作者所表达的情感，并且将这些情感整理为一个不超过五项的列表。

```
# 中文
prompt = f"""
识别以下评论的作者表达的情感。包含不超过五个项目。将答案格式化为以逗号分隔的单词列表。

评论文本：```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
满意,感激,赞赏,信任,满足
```

大型语言模型非常擅长从一段文本中提取特定的东西。在上面的例子中，评论所表达的情感有助于了解客户如何看待特定的产品。

1.3 识别愤怒

对于许多企业来说，洞察到顾客的愤怒情绪是至关重要的。这就引出了一个分类问题：下述的评论作者是否流露出了愤怒？因为如果有人真的情绪激动，那可能就意味着需要给予额外的关注，因为每一个愤怒的顾客都是一个改进服务的机会，也是一个提升公司口碑的机会。这时，客户支持或者客服团队就应该介入，与客户接触，了解具体情况，然后解决他们的问题。

```
# 中文
prompt = f"""
以下评论的作者是否表达了愤怒？评论用三个反引号分隔。给出是或否的答案。

评论文本：```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
否
```

上面这个例子中，客户并没有生气。注意，如果使用常规的监督学习，如果想要建立所有这些分类器，不可能在几分钟内就做到这一点。我们鼓励大家尝试更改一些这样的 Prompt ，也许询问客户是否表达了喜悦，或者询问是否有任何遗漏的部分，并看看是否可以让 Prompt 对这个灯具评论做出不同的推论。

二、信息提取

2.1 商品信息提取

信息提取是自然语言处理（NLP）的重要组成部分，它帮助我们从文本中抽取特定的、我们关心的信息。我们将深入挖掘客户评论中的丰富信息。在接下来的示例中，我们将要求模型识别两个关键元素：购买的商品和商品的制造商。

想象一下，如果你正在尝试分析一个在线电商网站上的众多评论，了解评论中提到的商品是什么、由谁制造，以及相关的积极或消极情绪，将极大地帮助你追踪特定商品或制造商在用户心中的情感趋势。

在接下来的示例中，我们会要求模型将回应以一个 JSON 对象的形式呈现，其中的 key 就是商品和品牌。

```
# 中文
prompt = f"""
从评论文本中识别以下项目：
- 评论者购买的物品
- 制造该物品的公司

评论文本用三个反引号分隔。将你的响应格式化为以“物品”和“品牌”为键的 JSON 对象。
如果信息不存在，请使用“未知”作为值。
让你的回应尽可能简短。

评论文本：```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
{
  "物品": "卧室灯",
  "品牌": "Lumina"
}
```

如上所示，它会说这个物品是一个卧室灯，品牌是 Luminar，你可以轻松地将其加载到 Python 字典中，然后对此输出进行其他处理。

2.2 综合情感推断和信息提取

在上面小节中，我们采用了三至四个 Prompt 来提取评论中的“情绪倾向”、“是否生气”、“物品类型”和“品牌”等信息。然而，事实上，我们可以设计一个单一的 Prompt ，来同时提取所有这些信息。

```
# 中文
prompt = f"""
从评论文本中识别以下项目：
- 情绪（正面或负面）
```

```
- 审稿人是否表达了愤怒？（是或否）
- 评论者购买的物品
- 制造该物品的公司

评论用三个反引号分隔。将你的响应格式化为 JSON 对象，以 “情感倾向”、“是否生气”、“物品类型” 和 “品牌” 作为键。
如果信息不存在，请使用 “未知” 作为值。
让你的回应尽可能简短。
将 “是否生气” 值格式化为布尔值。

评论文本: ```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
{
  "情感倾向": "正面",
  "是否生气": false,
  "物品类型": "卧室灯",
  "品牌": "Lumina"
}
```

这个例子中，我们指导 LLM 将“是否生气”的情况格式化为布尔值，并输出 JSON 格式。你可以尝试对格式化模式进行各种变化，或者使用完全不同的评论来试验，看看 LLM 是否仍然可以准确地提取这些内容。

三、主题推断

大型语言模型的另一个很酷的应用是推断主题。假设我们有一段长文本，我们如何判断这段文本的主旨是什么？它涉及了哪些主题？让我们通过以下一段虚构的报纸报道来具体了解一下。

```
# 中文
story = """
在政府最近进行的一项调查中，要求公共部门的员工对他们所在部门的满意度进行评分。
调查结果显示，NASA 是最受欢迎的部门，满意度为 95%。

一位 NASA 员工 John Smith 对这一发现发表了评论，他表示：
“我对 NASA 排名第一并不感到惊讶。这是一个与了不起的人们和令人难以置信的机会共事的好地方。我为成为这样一个创新组织的一员感到自豪。”

NASA 的管理团队也对这一结果表示欢迎，主管 Tom Johnson 表示：
“我们很高兴听到我们的员工对 NASA 的工作感到满意。
我们拥有一支才华横溢、忠诚敬业的团队，他们为实现我们的目标不懈努力，看到他们的辛勤工作得到回报是太棒了。”

调查还显示，社会保障管理局的满意度最低，只有 45%的员工表示他们对工作满意。
政府承诺解决调查中员工提出的问题，并努力提高所有部门的工作满意度。
"""
```

3.1 推断讨论主题

以上是一篇关于政府员工对其工作单位感受的虚构报纸文章。我们可以要求大语言模型确定其中讨论的五个主题，并用一两个词语概括每个主题。输出结果将会以逗号分隔的Python列表形式呈现。

```
# 中文
prompt = f"""
确定以下给定文本中讨论的五个主题。

每个主题用1-2个词概括。

请输出一个可解析的Python列表，每个元素是一个字符串，展示了一个主题。

给定文本: ```{story}```
"""
response = get_completion(prompt)
print(response)
```

```
['NASA', '满意度', '评论', '管理团队', '社会保障管理局']
```

3.2 为特定主题制作新闻提醒

假设我们有一个新闻网站或类似的平台，这是我们感兴趣的主题：美国航空航天局、当地政府、工程、员工满意度、联邦政府等。我们想要分析一篇新闻文章，理解其包含了哪些主题。可以使用这样的Prompt：确定以下主题列表中的每个项目是否是以下文本中的主题。以 0 或 1 的形式给出答案列表。

```
# 中文
prompt = f"""
判断主题列表中的每一项是否是给定文本中的一个话题，

以列表的形式给出答案，每个元素是一个Json对象，键为对应主题，值为对应的 0 或 1。

主题列表：美国航空航天局、当地政府、工程、员工满意度、联邦政府

给定文本: ```{story}```
"""
response = get_completion(prompt)
print(response)
```

```
[
  {"美国航空航天局": 1},
  {"当地政府": 1},
  {"工程": 0},
  {"员工满意度": 1},
  {"联邦政府": 1}
]
```

从输出结果来看，这个 `story` 与关于“美国航空航天局”、“员工满意度”、“联邦政府”、“当地政府”有关，而与“工程”无关。这种能力在机器学习领域被称为零样本（Zero-Shot）学习。这是因为我们并没有提供任何带标签的训练数据，仅凭 Prompt ，它便能判定哪些主题在新闻文章中被包含。

如果我们希望制定一个新闻提醒，我们同样可以运用这种处理新闻的流程。假设我对“美国航空航天局”的工作深感兴趣，那么你就可以构建一个如此的系统：每当出现与'美国宇航局'相关的新闻，系统就会输出提醒。

```
result_lst = eval(response)
topic_dict = {list(i.keys())[0] : list(i.values())[0] for i in result_lst}
print(topic_dict)
if topic_dict['美国航空航天局'] == 1:
    print("提醒: 关于美国航空航天局的新消息")
```

```
{'美国航空航天局': 1, '当地政府': 1, '工程': 0, '员工满意度': 1, '联邦政府': 1}
提醒: 关于美国航空航天局的新消息
```

这就是我们关于推断的全面介绍。在短短几分钟内，我们已经能够建立多个用于文本推理的系统，这是以前需要机器学习专家数天甚至数周时间才能完成的任务。这一变化无疑是令人兴奋的，因为无论你是经验丰富的机器学习开发者，还是刚入门的新手，都能利用输入 Prompt 快速开始复杂的自然语言处理任务。

英文版

1.1 情感倾向分析

```
lamp_review = """
Needed a nice lamp for my bedroom, and this one had \
additional storage and not too high of a price point. \
Got it fast.  The string to our lamp broke during the \
transit and the company happily sent over a new one. \
Came within a few days as well. It was easy to put \
together.  I had a missing part, so I contacted their \
support and they very quickly got me the missing piece! \
Lumina seems to me to be a great company that cares \
about their customers and products!!
"""
```

```
prompt = f"""
What is the sentiment of the following product review,
which is delimited with triple backticks?

Review text: ```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
The sentiment of the product review is positive.
```

```
prompt = f"""
What is the sentiment of the following product review,
which is delimited with triple backticks?

Give your answer as a single word, either "positive" \
or "negative".

Review text: ```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
positive
```

1.2识别情感类型

```
prompt = f"""
Identify a list of emotions that the writer of the \
following review is expressing. Include no more than \
five items in the list. Format your answer as a list of \
lower-case words separated by commas.

Review text: ```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
satisfied, pleased, grateful, impressed, happy
```

1.3 识别愤怒

```
prompt = f"""
Is the writer of the following review expressing anger?\
The review is delimited with triple backticks. \
Give your answer as either yes or no.

Review text: ```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
No
```

2.1 商品信息提取

```
prompt = f"""
Identify the following items from the review text:
- Item purchased by reviewer
- Company that made the item

The review is delimited with triple backticks. \
```

```
Format your response as a JSON object with \
"Item" and "Brand" as the keys.
If the information isn't present, use "unknown" \
as the value.
Make your response as short as possible.

Review text: ```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
{
  "Item": "lamp",
  "Brand": "Lumina"
}
```

2.2 综合情感推断和信息提取

```
prompt = f"""
Identify the following items from the review text:
- Sentiment (positive or negative)
- Is the reviewer expressing anger? (true or false)
- Item purchased by reviewer
- Company that made the item

The review is delimited with triple backticks. \
Format your response as a JSON object with \
"Sentiment", "Anger", "Item" and "Brand" as the keys.
If the information isn't present, use "unknown" \
as the value.
Make your response as short as possible.
Format the Anger value as a boolean.

Review text: ```{lamp_review}```
"""
response = get_completion(prompt)
print(response)
```

```
{
  "Sentiment": "positive",
  "Anger": false,
  "Item": "lamp",
  "Brand": "Lumina"
}
```

3.1 推断讨论主题

```
story = """
In a recent survey conducted by the government,
public sector employees were asked to rate their level
of satisfaction with the department they work at.
The results revealed that NASA was the most popular
department with a satisfaction rating of 95%.
```

```
One NASA employee, John Smith, commented on the findings,
stating, "I'm not surprised that NASA came out on top.
It's a great place to work with amazing people and
incredible opportunities. I'm proud to be a part of
such an innovative organization."

The results were also welcomed by NASA's management team,
with Director Tom Johnson stating, "We are thrilled to
hear that our employees are satisfied with their work at NASA.
We have a talented and dedicated team who work tirelessly
to achieve our goals, and it's fantastic to see that their
hard work is paying off."

The survey also revealed that the
Social Security Administration had the lowest satisfaction
rating, with only 45% of employees indicating they were
satisfied with their job. The government has pledged to
address the concerns raised by employees in the survey and
work towards improving job satisfaction across all departments.
"""
```

```
prompt = f"""
Determine five topics that are being discussed in the \
following text, which is delimited by triple backticks.

Make each item one or two words long.

Format your response as a list of items separated by commas.
Give me a list which can be read in Python.

Text sample: ```{story}```
"""
response = get_completion(prompt)
print(response)
```

```
survey, satisfaction rating, NASA, Social Security Administration, job
satisfaction
```

```
response.split(sep=',')
```

```
['survey',
 ' satisfaction rating',
 ' NASA',
 ' Social Security Administration',
 ' job satisfaction']
```

3.2 为特定主题制作新闻提醒

```
topic_list = [
    "nasa", "local government", "engineering",
    "employee satisfaction", "federal government"
]
```

```
prompt = f"""
Determine whether each item in the following list of \
topics is a topic in the text below, which
is delimited with triple backticks.

Give your answer as list with 0 or 1 for each topic.\

List of topics: {", ".join(topic_list)}

Text sample: ```{story}```
"""
response = get_completion(prompt)
print(response)
```

```
[1, 0, 0, 1, 1]
```

```
topic_dict = {topic_list[i] : eval(response)[i] for i in
range(len(eval(response)))}
print(topic_dict)
if topic_dict['nasa'] == 1:
    print("ALERT: New NASA story!")
```

```
{'nasa': 1, 'local government': 0, 'engineering': 0, 'employee satisfaction': 1,
'federal government': 1}
ALERT: New NASA story!
```