```
!pip -q install langchain openai tiktoken
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 20.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 73.6/73.6 kB 8.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 75.6 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 61.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 90.0/90.0 kB 10.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 114.5/114.5 kB 9.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 268.8/268.8 kB 23.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 149.6/149.6 kB 13.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.1/49.1 kB 2.3 MB/s eta
0:00:00
```

```
import os

os.environ["OPENAI_API_KEY"] = ""
```

```
!pip show langchain
```

```
Name: langchain
Version: 0.0.206
Summary: Building applications with LLMs through composability
Home-page: https://www.github.com/hwchase17/langchain
Author:
Author-email:
License: MIT
Location: /usr/local/lib/python3.10/dist-packages
Requires: aiohttp, async-timeout, dataclasses-json, langchainplus-sdk,
numexpr, numpy, openapi-schema-pydantic, pydantic, PyYAML, requests,
SQLAlchemy, tenacity
Required-by:
```

Classification / Tagging

```
from langchain.chat_models import ChatOpenAI
from langchain.chains import create_tagging_chain,
create_tagging_chain_pydantic
from langchain.prompts import ChatPromptTemplate

llm = ChatOpenAI(temperature=0, model="gpt-3.5-turbo-0613")

schema = {
    "properties": {
        "sentiment": {"type": "string"},
```

```
        "stars": {"type": "integer"},
        "language": {"type": "string"},
    }
}

chain = create_tagging_chain(schema, llm)

chain.prompt.messages[0].prompt

PromptTemplate(input_variables=['input'], output_parser=None,
partial_variables={}, template='Extract the desired information from
the following passage.\n\nPassage:\n{input}\n', template_format='f-
string', validate_template=True)

print(chain.prompt.messages[0].prompt.template)

Extract the desired information from the following passage.

Passage:
{input}


chain.llm_kwargs

{'functions': [{'name': 'information_extraction',
   'description': 'Extracts the relevant information from the
passage.',
   'parameters': {'type': 'object',
    'properties': {'sentiment': {'title': 'sentiment', 'type':
'string'},
     'stars': {'title': 'stars', 'type': 'integer'},
     'language': {'title': 'language', 'type': 'string'}},
    'required': []}}],
 'function_call': {'name': 'information_extraction'}}

chain.output_parser

JsonOutputFunctionsParser()

review_01 = "Starts off kind of slow, but builds up quickly to give
the reader a good understanding of how things unfolded. I’m
anticipating a second book, and can’t wait for the rest of the story
to be unveiled!"
chain.run(review_01)

{'sentiment': 'positive', 'stars': 4}

review_02 = """It's the BIGGEST CON \n
This book has NOT been security checked for correct information , I
feel Meghan and Harry are laughing at me for buying this book , And
putting my money in there pockets. Harry can't come up with any dates,
says he can't remember the dates of anything , Said he was At college
on a Hot summers day when the phone call came about his Great Great
```

```
Grandmother passing , He was not at school he was Skiing in
Switzerland with his Brother ( I remember this) And the papers have
come back with the proof. Harry says that his mother bought him an
XBox for his 13th birthday in advance of her death that her sister
brought to the school for him, again not true the XBox didn't come out
till 4 years later . There is so many holes in the book . Every other
page has something on it that didn't happen. He takes NO reasonability
for ANYthing even when the evidence is staring him in his face. This
book is a BIG con, don't waste your money."""
```

```
chain.run(review_02)
```

```
{'sentiment': 'negative'}
```

```
review_03 = """ A Mixed Bag…..No One Wins Here… \n
I had to take a break from writing this review to separate my two
sons. It seems one has pushed the other over in some sort of
disagreement…Such is the nature of brothers, of siblings, of families.

Prince Harry’s memoir is equal parts refreshingly and cringe-worthily
honest. It leaves no stone unturned, and in that lies its greatness
and its tragedy.

Like any family, Prince Harry’s is a mix of differing personalities
and quirks, but unlike every family they are a part of a 1000 year
institution- in which the expectations, customs, and formality run
deep. In fact he is in a position so unlike any other human that this
is almost a must-read. But it’s more of a curiosity satisfier- and
when you’re done you kind of get the feeling you do when you
accidentally open a restroom stall and someone is using it. You
politely want to apologize excuse yourself from their business…

What I liked- the Prince’s detailed account in falling in love with
Meghan Markle. Marrying into a family culturally different than you is
no easy feat- and I can’t imagine the learning curve she had to face-
mostly alone. Her feelings of isolation and despair come across
clearly. This is a man who truly loves his wife. Perhaps, as the late,
remarkable Queen purportedly stated, maybe a little ‘over in love.’
I also truly felt the Prince captured the pain of losing his beloved
mother, in the most tragic of circumstances, poignantly and honestly.
In that he does a service to those also suffering grief (i.e.-all of
us). His openness and willingness to receiving mental health services
is admirable and I believe will help many. (If only all were able to
access the kinds of treatment and mental health support the Prince
did, but I digress…)

What I didn’t like:
There is sharing and then there is over-sharing. (Hence, me feeling
like I accidentally walked in on him in the restroom, indisposed). I
don’t need/want to know the details of his ‘first time,’ while his
```

```
military service is laudable, discussing the number of Taliban
fighters he killed was in poor taste, and the tiny snippets he throws
in regarding Kate seem rather petty.

As a woman who admires both Kate and Meghan, I felt he was perhaps
unintentionally adding to the common trope that women must always be
jealous and fighting.

Much of his disdain is aimed at the British press, and I can't deny it
is likely a brutal and ugly business. But perhaps too much disdain is
aimed at his family as well- imperfect as they are.

While I'm certain there was a level of catharsis in writing this
memoir, I worry that the long term consequences will far outweigh any
positive outcomes from such a blistering account. My prayers to this
entire family as they navigate these tough waters, that have now
seemingly become even more treacherous."""
```

```
chain.run(review_03)
```

```
{}
```

```
schema = {
    "properties": {
        "sentiment": {"type": "string", "enum": ["positive",
"neutral", "negative"]},
        "stars": {
            "type": "integer",
            "enum": [1, 2, 3, 4, 5],
            "description": "describes the number of stars give by a
reviewer on Amazon",
        },
        "language": {
            "type": "string",
            "enum": ["spanish", "english", "french", "german",
"italian"],
        },
    },
    "required": ["language", "sentiment", "stars"],
}

chain = create_tagging_chain(schema, llm)

chain.run(review_01)
```

```
{'sentiment': 'positive', 'stars': 4, 'language': 'english'}
```

```
chain.run(review_02)
```

```
{'sentiment': 'negative', 'stars': 1, 'language': 'english'}
```

```
chain.run(review_03)
```

```
{'sentiment': 'neutral', 'stars': 3, 'language': 'english'}
```

```
type(chain.run(review_03))
```

```
dict
```

*italicized text### Using Pydantic

```
from enum import Enum
from pydantic import BaseModel, Field

class Tags(BaseModel):
    sentiment: str = Field(..., enum=["positive", "neutral",
"negative"])
    stars: int = Field(
        ...,
        description="describes the number of stars give by a reviewer
on Amazon",
        enum=[1, 2, 3, 4, 5],
    )
    language: str = Field(
        ..., enum=["spanish", "english", "french", "german",
"italian"]
    )

chain = create_tagging_chain_pydantic(Tags, llm)

res = chain.run(review_01)

res
```

```
Tags(sentiment='positive', stars=4, language='english')
```

```
res.sentiment
```

```
{"type":"string"}
```

```
# This is formatted as code
```

Extraction, NER

```
from langchain.chat_models import ChatOpenAI
from langchain.chains import create_extraction_chain,
create_extraction_chain_pydantic
from langchain.prompts import ChatPromptTemplate

tc_01 = """Reddit CEO Steve Huffman is not backing down amid protests
against API changes made by the platform. In interviews with The
Verge, NBC News and NPR, Huffman defended business decisions made by
the company to charge third-party apps saying the API wasn't designed
to support these clients.

The Reddit co-founder also talked about protesting moderators,
changing site rules, and profitability in these interviews. The
```

```
platform is facing one of the strongest backlashes from the community,
but the CEO seemingly doesn’t want to budge.

What’s happening at Reddit?
In April, Reddit announced that it is going to charge for its API, but
didn’t announce any pricing. Earlier this month, Christian Selig, the
developer of a popular Reddit client for iOS called Apollo, posted
that he had a call with Reddit. API pricing quoted by them could cost
him $20 million a year to run the app. Selig later said that, because
the social network is not ready to make any changes to the pricing
structure, he is forced to shut down Apollo. Other third-party
developers of clients like Reddit is Fun and Relay for Reddit also
said that they will shut down their apps on June 30.

The only exception Reddit made was to allow free access to the API to
non-commercial apps providing accessibility features. The company has
made deals with apps like RedReader, Dystopia, and Luna and given them
exemptions from its “large-scale pricing terms.”

Thousands of subreddits went dark starting June 12 to protest those
changes — it caused a brief outage as well. Meanwhile, Huffman took a
strong stance in his AMA and took a dig at Apollo and Selig. As
moderators didn’t see anything changing, many subreddits decided to
extend the blackout."""

llm = ChatOpenAI(temperature=0, model="gpt-3.5-turbo-0613")

schema = {
    "properties": {
        "person_name": {"type": "string"},
        "startup": {"type": "string"},
        "news_outlet": {"type": "string"},
        "app_name": {"type": "string"},
        "month": {"type": "string"},
    },
    "required": ["person_name", "startup"],
}

chain = create_extraction_chain(schema, llm)

chain.prompt.messages[0]

HumanMessagePromptTemplate(prompt=PromptTemplate(input_variables=['inp
ut'], output_parser=None, partial_variables={}, template='Extract the
desired information from the following passage.\n\nPassage:\n{input}\
n', template_format='f-string', validate_template=True),
additional_kwargs={})

print(chain.prompt.messages[0].prompt.template)
```

```
Extract and save the relevant entities mentioned in the following
passage together with their properties.

Passage:
{input}


chain.llm_kwargs
```

```
{'functions': [{'name': 'information_extraction',
   'description': 'Extracts the relevant information from the
passage.',
   'parameters': {'type': 'object',
    'properties': {'info': {'type': 'array',
      'items': {'type': 'object',
       'properties': {'person_name': {'title': 'person_name',
         'type': 'string'},
        'startup': {'title': 'startup', 'type': 'string'},
        'news_outlet': {'title': 'news_outlet', 'type': 'string'},
        'app_name': {'title': 'app_name', 'type': 'string'},
        'month': {'title': 'month', 'type': 'string'}},
       'required': ['person_name', 'startup']}}},
    'required': ['info']}}],
 'function_call': {'name': 'information_extraction'}}
```

```
chain.output_parser
```

```
JsonKeyOutputFunctionsParser(key_name='info')
```

```
response = chain.run(tc_01)
response
```

```
[{'person_name': 'Steve Huffman',
  'startup': 'Reddit',
  'news_outlet': 'The Verge',
  'app_name': 'Apollo',
  'month': 'April'},
 {'person_name': 'Christian Selig',
  'startup': 'Apollo',
  'news_outlet': 'N/A',
  'app_name': 'Apollo',
  'month': 'April'},
 {'person_name': 'Christian Selig',
  'startup': 'Apollo',
  'news_outlet': 'N/A',
  'app_name': 'Apollo',
  'month': 'June'},
 {'person_name': 'Reddit',
  'startup': 'Reddit',
  'news_outlet': 'N/A',
  'app_name': 'Reddit is Fun',
```

```
  'month': 'June'},
 {'person_name': 'Reddit',
  'startup': 'Reddit',
  'news_outlet': 'N/A',
  'app_name': 'Relay for Reddit',
  'month': 'June'}]
```

```
def sort_objects(obj_list):
    # Initialize empty lists for each category
    people = []
    startups = []
    news_outlets = []
    apps = []
    months = []

    # Loop through each dictionary in the list
    for obj in obj_list:
        # Add each value to the appropriate list, if the key exists in
the dictionary and the value is not already in the list
        if 'person_name' in obj and obj['person_name'] not in people:
            people.append(obj['person_name'])
        if 'startup' in obj and obj['startup'] not in startups:
            startups.append(obj['startup'])
        if 'news_outlet' in obj and obj['news_outlet'] not in
news_outlets:
            news_outlets.append(obj['news_outlet'])
        if 'app_name' in obj and obj['app_name'] not in apps:
            apps.append(obj['app_name'])
        if 'month' in obj and obj['month'] not in months:
            months.append(obj['month'])

    # Return the lists as a dictionary
    return {'people': people, 'startups': startups, 'news_outlets':
news_outlets, 'apps': apps, 'months': months}

sort_objects(response)
```

```
{'people': ['Steve Huffman', 'Christian Selig', 'Reddit'],
 'startups': ['Reddit', 'Apollo'],
 'news_outlets': ['The Verge', 'N/A'],
 'apps': ['Apollo', 'Reddit is Fun', 'Relay for Reddit'],
 'months': ['April', 'June']}
```

```
tc_02 = """Reddit CEO Steve Huffman is not backing down amid protests
against API changes made by the platform. In interviews with The
Verge, NBC News and NPR, Huffman defended business decisions made by
the company to charge third-party apps saying the API wasn’t designed
to support these clients.

The Reddit co-founder also talked about protesting moderators,
changing site rules, and profitability in these interviews. The
```

```
platform is facing one of the strongest backlashes from the community,
but the CEO seemingly doesn’t want to budge. The Apollo app"""

response = chain.run(tc_02)
response
```

```
[{'person_name': 'Steve Huffman',
  'startup': 'Reddit',
  'news_outlet': 'The Verge',
  'app_name': 'Apollo',
  'month': 'June'},
 {'person_name': 'Steve Huffman',
  'startup': 'Reddit',
  'news_outlet': 'NBC News',
  'app_name': 'Apollo',
  'month': 'June'},
 {'person_name': 'Steve Huffman',
  'startup': 'Reddit',
  'news_outlet': 'NPR',
  'app_name': 'Apollo',
  'month': 'June'}]
```

```
sort_objects(response)
```

```
{'people': ['Steve Huffman'],
 'startups': ['Reddit'],
 'news_outlets': ['The Verge', 'NBC News', 'NPR'],
 'apps': ['Apollo'],
 'months': ['June']}
```