Auto-fixing parser

This output parser wraps another output parser, and in the event that the first one fails it calls out to another LLM to fix any errors.

But we can do other things besides throw errors. Specifically, we can pass the misformatted output, along with the formatted instructions, to the model and ask it to fix it.

For this example, we'll use the above Pydantic output parser. Here's what happens if we pass it a result that does not comply with the schema:

```
from langchain.prompts import PromptTemplate, ChatPromptTemplate, HumanMessagePromptTemplate
from langchain.llms import OpenAI
from langchain.chat_models import ChatOpenAI
from langchain.output_parsers import PydanticOutputParser
from pydantic import BaseModel, Field, validator
from typing import List
```

```
class Actor(BaseModel):
    name: str = Field(description="name of an actor")
    film_names: List[str] = Field(description="list of names of films they starred in")

actor_query = "Generate the filmography for a random actor."

parser = PydanticOutputParser(pydantic_object=Actor)
```

```
misformatted = "{'name': 'Tom Hanks', 'film_names': ['Forrest Gump']}"
```

```
parser.parse(misformatted)
```

```
    ---------------------------------------------------------------------------

    JSONDecodeError                           Traceback (most recent call last)

    File ~/workplace/langchain/langchain/output_parsers/pydantic.py:23, in PydanticOutputParser.parse(self, 
text)
         22     json_str = match.group()
    ---> 23 json_object = json.loads(json_str)
         24 return self.pydantic_object.parse_obj(json_object)


    File ~/.pyenv/versions/3.9.1/lib/python3.9/json/__init__.py:346, in loads(s, cls, object_hook, 
parse_float, parse_int, parse_constant, object_pairs_hook, **kw)
        343 if (cls is None and object_hook is None and
        344         parse_int is None and parse_float is None and
        345         parse_constant is None and object_pairs_hook is None and not kw):
    --> 346     return _default_decoder.decode(s)
        347 if cls is None:


    File ~/.pyenv/versions/3.9.1/lib/python3.9/json/decoder.py:337, in JSONDecoder.decode(self, s, _w)
        333 """Return the Python representation of ``s`` (a ``str`` instance
        334 containing a JSON document).
        335 
        336 """
    --> 337 obj, end = self.raw_decode(s, idx=_w(s, 0).end())
```

```
        338 end = _w(s, end).end()


    File ~/.pyenv/versions/3.9.1/lib/python3.9/json/decoder.py:353, in JSONDecoder.raw_decode(self, s, idx)
        352 try:
    --> 353     obj, end = self.scan_once(s, idx)
        354 except StopIteration as err:


    JSONDecodeError: Expecting property name enclosed in double quotes: line 1 column 2 (char 1)


    During handling of the above exception, another exception occurred:


    OutputParserException                     Traceback (most recent call last)

    Cell In[6], line 1
    ----> 1 parser.parse(misformatted)


    File ~/workplace/langchain/langchain/output_parsers/pydantic.py:29, in PydanticOutputParser.parse(self,
text)
         27 name = self.pydantic_object.__name__
         28 msg = f"Failed to parse {name} from completion {text}. Got: {e}"
    ---> 29 raise OutputParserException(msg)


    OutputParserException: Failed to parse Actor from completion {'name': 'Tom Hanks', 'film_names':
['Forrest Gump']}. Got: Expecting property name enclosed in double quotes: line 1 column 2 (char 1)
```

Now we can construct and use a `OutputFixingParser`. This output parser takes as an argument another output parser but also an LLM with which to try to correct any formatting mistakes.

```
from langchain.output_parsers import OutputFixingParser

new_parser = OutputFixingParser.from_llm(parser=parser, llm=ChatOpenAI())
```

```
new_parser.parse(misformatted)
```

```
    Actor(name='Tom Hanks', film_names=['Forrest Gump'])
```