RetryOutputParser

While in some cases it is possible to fix any parsing mistakes by only looking at the output, in other cases it can't. An example of this is when the output is not just in the incorrect format, but is partially complete. Consider the below example.

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

```
template = """Based on the user question, provide an Action and Action Input for
what step should be taken.
{format_instructions}
Question: {query}
Response:"""
class Action(BaseModel):
    action: str = Field(description="action to take")
    action_input: str = Field(description="input to the action")

parser = PydanticOutputParser(pydantic_object=Action)
```

```
prompt = PromptTemplate(
    template="Answer the user query.\n{format_instructions}\n{query}\n",
    input_variables=["query"],
    partial_variables={"format_instructions": parser.get_format_instructions()}
)
```

```
prompt_value = prompt.format_prompt(query="who is leo di caprios gf?")
```

```
bad_response = '{"action": "search"}'
```

If we try to parse this response as is, we will get an error

Skip to main content

```
parser.parse(bad_response)
```

```
---------------------------------------------------------------------------
ValidationError                           Traceback (most recent call last)
File ~/workplace/langchain/langchain/output_parsers/pydantic.py:24, in
PydanticOutputParser.parse(self, text)
     23     json_object = json.loads(json_str)
---> 24     return self.pydantic_object.parse_obj(json_object)
     26 except (json.JSONDecodeError, ValidationError) as e:

File ~/.pyenv/versions/3.9.1/envs/langchain/lib/python3.9/site-
packages/pydantic/main.py:527, in pydantic.main.BaseModel.parse_obj()

File ~/.pyenv/versions/3.9.1/envs/langchain/lib/python3.9/site-
packages/pydantic/main.py:342, in pydantic.main.BaseModel.__init__()

ValidationError: 1 validation error for Action
action_input
  field required (type=value_error.missing)

During handling of the above exception, another exception occurred:

OutputParserException                     Traceback (most recent call last)
Cell In[6], line 1
----> 1 parser.parse(bad_response)

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 Action from completion {"action":
"search"}. Got: 1 validation error for Action
action_input
  field required (type=value_error.missing)
```

If we try to use the `OutputFixingParser` to fix this error, it will be confused - namely, it doesn't know what to actually put for action input.

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

```
fix_parser.parse(bad_response)
```

Skip to main content

Instead, we can use the RetryOutputParser, which passes in the prompt (as well as the original output) to try again to get a better response.

```
from langchain.output_parsers import RetryWithErrorOutputParser
```

```
retry_parser = RetryWithErrorOutputParser.from_llm(parser=parser,
llm=OpenAI(temperature=0))
```

```
retry_parser.parse_with_prompt(bad_response, prompt_value)
```

```
Action(action='search', action_input='who is leo di caprios gf?')
```