Few shot examples for chat models

This notebook covers how to use few shot examples in chat models.

There does not appear to be solid consensus on how best to do few shot prompting. As a result, we are not solidifying any abstractions around this yet but rather using existing abstractions.

Alternating Human/AI messages

The first way of doing few shot prompting relies on using alternating human/ai messages. See an example of this below.

```python
from langchain.chat_models import ChatOpenAI
from langchain import PromptTemplate, LLMChain
from langchain.prompts.chat import (
    ChatPromptTemplate,
    SystemMessagePromptTemplate,
    AIMessagePromptTemplate,
    HumanMessagePromptTemplate,
)
from langchain.schema import AIMessage, HumanMessage, SystemMessage
```

```python
chat = ChatOpenAI(temperature=0)
```

```python
template = "You are a helpful assistant that translates english to pirate."
system_message_prompt = SystemMessagePromptTemplate.from_template(template)
example_human = HumanMessagePromptTemplate.from_template("Hi")
example_ai = AIMessagePromptTemplate.from_template("Argh me mateys")
human_template = "{text}"
human_message_prompt = HumanMessagePromptTemplate.from_template(human_template)
```

```python
chat_prompt = ChatPromptTemplate.from_messages(
    [system_message_prompt, example_human, example_ai, human_message_prompt]
)
chain = LLMChain(llm=chat, prompt=chat_prompt)
# get a chat completion from the formatted messages
chain.run("I love programming.")
```

```
    "I be lovin' programmin', me hearty!"
```

System Messages

OpenAI provides an optional `name` parameter that they also recommend using in conjunction with system messages to do few shot prompting. Here is an example of how to do that below.

```python
template = "You are a helpful assistant that translates english to pirate."
system_message_prompt = SystemMessagePromptTemplate.from_template(template)
example_human = SystemMessagePromptTemplate.from_template(
    "Hi", additional_kwargs={"name": "example_user"}
)
```

```python
example_ai = SystemMessagePromptTemplate.from_template(
    "Argh me mateys", additional_kwargs={"name": "example_assistant"}
)
human_template = "{text}"
human_message_prompt = HumanMessagePromptTemplate.from_template(human_template)
```

```python
chat_prompt = ChatPromptTemplate.from_messages(
    [system_message_prompt, example_human, example_ai, human_message_prompt]
)
chain = LLMChain(llm=chat, prompt=chat_prompt)
# get a chat completion from the formatted messages
chain.run("I love programming.")
```

```
    "I be lovin' programmin', me hearty."
```