Redes Neuronales para Lenguaje Natural

2024

Grupo de Procesamiento de Lenguaje Natural
Instituto de Computación


Agentes

ReAct Prompting

Agentes

- Espacio de acciones: $\mathcal{A} \rightarrow$ Acciones que puede realizar el agente.
- Espacio de observaciones: $\mathcal{O} \rightarrow$ “Respuestas” del ambiente.
- Espacio de estados: $\mathcal{C} \rightarrow$ Contexto del agente.
- En el paso t, el agente recibe una observación $o_t \in \mathcal{O}$ del ambiente y realiza una acción $a_t \in \mathcal{A}$ siguiendo una política $\pi(a_t|c_t)$ donde $c_t = (o_1, a_1, \cdots, o_{t-1}, a_{t-1}, o_t)$.

ReAct (Reasoning + Acting)

- Aumentar el espacio de acciones $\hat{\mathcal{A}} = \mathcal{A} \cup \mathcal{L}$ donde $\mathcal{L}$ es el espacio del lenguaje.

ReAct: Synergizing Reasoning and Acting in Language Models
Wei et al., (2022)

ReAct Prompting

En paso *t*:

- A partir del contexto (los ejemplos Few-Shot y los pasos anteriores) el modelo de lenguaje genera:
  - **Thought *t***
  - **Action *t***
- Mediante *parsing*, se detecta la acción a realizar:
  - Si la acción es **Finish**[Respuesta]: **termina**
  - Si es otra acción, se ejecuta obteniendo: **Observation *t***
- Se avanza al paso *t+1*

ReAct: Synergizing Reasoning and Acting in Language Models
Wei et al., (2022)

ReAct Prompting - Ejemplo (Few-Shot)

Usuario

Question: Musician and satirist Allie Goertz wrote a song about the "The Simpsons" character Milhouse, who Matt Groening named after who?

LLM

Thought 1: The question simplifies to "The Simpsons" character Milhouse is named after who. I only need to search Milhouse and find who it is named after.

Action 1: Search[Milhouse]

Ambiente

Observation 1: Milhouse Mussolini Van Houten is a recurring character in the Fox animated television series The Simpsons voiced by Pamela Hayden and created by Matt Groening.

LLM

Thought 2: The paragraph does not tell who Milhouse is named after, maybe I can look up "named after".

Action 2: Lookup[named after]

Ambiente

Observation 2: (Result 1 / 1) Milhouse was named after U.S. president Richard Nixon, whose middle name was Milhous.

LLM

Thought 3: Milhouse was named after U.S. president Richard Nixon, so the answer is Richard Nixon.

Action 3: Finish[Richard Nixon]

Function Calling (tool calling)

Los modelos más nuevos son entrenados para ser capaces de ejecutar funciones (*acciones*).

- Reciben en la prompt una definición de las funciones posibles a ejecutar siguiendo un formato particular.
- Pueden decidir ejecutar una función usando tokens especiales, indicando:
  - Nombre de la función a ejecutar
  - Parámetros de entrada
- Luego, se debe ejecutar la función y el resultado se agrega a la prompt (*observación*), para que el modelo actúe en consecuencia.

Function Calling con Llama 3.1 (Hugging Face)

```python
from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline

tokenizer = AutoTokenizer.from_pretrained("meta-llama/Meta-Llama-3.1-8B-Instruct")
model_llm = AutoModelForCausalLM.from_pretrained(
    "meta-llama/Meta-Llama-3.1-8B-Instruct",
    device_map="auto"
)
```

```python
def get_current_temperature(location: str) -> float:
    """
    Get the current temperature at a location.

    Args:
        location: The location to get the temperature for, in the format
        "City, Country"
    Returns:
        The current temperature at the specified location in celsius degrees,
        as a float.
    """
    return 22.
```

Function Calling con Llama 3.1 (Hugging Face)

```
tools = [get_current_temperature]

messages = [
    {
        "role": "system",
        "content": "Eres un asistente que responde preguntas sobre el clima."
    },
    {
        "role": "user",
        "content": "¿Hace calor hoy en Montevideo?"
    }
]

prompt = tokenizer.apply_chat_template(
    messages,
    tools=tools,
    tokenize=False,
    add_generation_prompt=True
)
```

Function Calling con Llama 3.1 (Hugging Face)

```
print(prompt)
```

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

Environment: ipython
Cutting Knowledge Date: December 2023
Today Date: 26 Jul 2024

Eres un asistente que responde preguntas sobre el clima.<|eot_id|><|start_header_id|>user<|end_header_id|>

Given the following functions, please respond with a JSON for a function call with its proper arguments that best answers the given prompt.

Respond in the format {"name": function name, "parameters": dictionary of argument name and its value}.Do not use variables.

{
    "type": "function",
    "function": {
        "name": "get_current_temperature",
        "description": "Get the current temperature at a location.",
        "parameters": {
            "type": "object",
            "properties": {
                "location": {
                    "type": "string",
                    "description": "The location to get the temperature for, in the format \"City, Country\""
                }
            },
            "required": [
                "location"
            ]
        },
        "return": {
            "type": "number",
            "description": "The current temperature at the specified location in celsius degrees, as a float."
        }
    }
}

¿Hace calor hoy en Montevideo?<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Function Calling con Llama 3.1 (Hugging Face)

```
pipe = pipeline("text-generation", model=model_llm, tokenizer=tokenizer)

response = pipe(
    prompt,
    return_full_text=False,
    max_new_tokens=250,
    pad_token_id=tokenizer.eos_token_id
  )[0]["generated_text"]
```

```
print(response)
```

```
{"name": "get_current_temperature", "parameters": {"location": "Montevideo, Uruguay"}}
```

Function Calling con Llama 3.1 (Hugging Face)

```
import json

response_dict = json.loads(response)

weather = get_current_temperature(response_dict["parameters"]["location"])

tool_call = {
    "name": response_dict["name"],
    "arguments": response_dict["parameters"]
}

messages.append({
    "role": "assistant",
    "tool_calls": [{"type": "function", "function": tool_call}]
})
messages.append({
    "role": "tool",
    "name": response_dict["name"],
    "content": weather
})
```

Function Calling con Llama 3.1 (Hugging Face)

```
prompt = tokenizer.apply_chat_template(
    messages,
    tools=tools,
    tokenize=False,
    add_generation_prompt=True
)

response = pipe(
    prompt,
    return_full_text=False,
    max_new_tokens=250,
    pad_token_id=tokenizer.eos_token_id
)[0]["generated_text"]
```

```
print(response)
```

```
La temperatura en Montevideo, Uruguay es de 22.0°C.
```


Evaluación

Evaluación

Hasta ahora mencionamos la loss como métrica para predecir cómo se va a comportar el modelo

Pero esta no es una métrica de evaluación, solo nos ayuda a analizar si el entrenamiento es bueno

- Evaluación extrínseca: probarlo en tareas, por ejemplo con benchmarks como GLUE o benchmarks generativos
- Evaluación intrínseca: Perplejidad

Evaluación - Perplejidad

Intuición: Un modelo es mejor si asigna mayores probabilidades a datos no vistos.

Dado un conjunto de test: $w_1 w_2 w_3 \ldots w_n = w_{1:n}$

$$\text{Perplexity}_\theta(w_{1:n}) = P(w_{1:n})^{-\frac{1}{n}} = \sqrt[n]{\frac{1}{P(w_{1:n})}}$$

Probabilidad inversa

Normalizado por el largo

$$\text{Perplexity}_\theta(w_{1:n}) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i|w_{<i})}}$$

Evaluación - Perplejidad

Es inversa a la probabilidad: cuanto mayor sea la probabilidad de la secuencia, menor será la perplejidad

- Por lo tanto cuando menor la perplejidad, mejor el modelo
- **Minimizar la perplejidad es lo mismo que maximizar la probabilidad**

Cuidado: la perplejidad también es sensible al largo, que depende de la tokenización, por lo que es mejor comparar LMs con la misma tokenización

Evaluación - Otros factores

Tamaño

- Modelos grandes toman mucho tiempo y GPUs para entrenar y usan mucha memoria de almacenamiento

Gasto de energía

- Se puede medir en kWh, o kg de CO_2 emitido

Igualdad (fairness)

- Se utilizan benchmarks para medir sesgos de género, raza, estereotipos, y performance sobre grupos minoritarios

Evaluación - Benchmarks

Conjuntos de evaluación para distintas categorías:

General	MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024b), IFEval (Zhou et al., 2023)
Math and reasoning	GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), GPQA (Rein et al., 2023), ARC-Challenge (Clark et al., 2018)
Code	HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), HumanEval+ (Liu et al., 2024a), MBPP EvalPlus (base) (Liu et al., 2024a), MultiPL-E (Cassano et al., 2023)
Multilinguality	MGSM (Shi et al., 2022), Multilingual MMLU (internal benchmark)
Tool-use	Nexus (Srinivasan et al., 2023), API-Bank (Li et al., 2023b), API-Bench (Patil et al., 2023), BFCL (Yan et al., 2024)
Long context	ZeroSCROLLS (Shaham et al., 2023), Needle-in-a-Haystack (Kamradt, 2023), InfiniteBench (Zhang et al., 2024)

The Llama 3 Herd of Models
Llama Team, AI @ Meta (2024)

Evaluación - Benchmarks

Ejemplo - Algunos resultados reportados de Llama 3:

Meta Llama 3 Instruct model performance

	Meta Llama 3 8B	Gemma 7B - It Measured	Mistral 7B Instruct Measured
MMLU 5-shot	**68.4**	53.3	58.4
GPQA 0-shot	**34.2**	21.4	26.3
HumanEval 0-shot	**62.2**	30.5	36.6
GSM-8K 8-shot, CoT	**79.6**	30.6	39.9
MATH 4-shot, CoT	**30.0**	12.2	11.0

	Meta Llama 3 70B	Gemini Pro 1.5 Published	Claude 3 Sonnet Published
MMLU 5-shot	**82.0**	81.9	79.0
GPQA 0-shot	39.5	**41.5** CoT	38.5 CoT
HumanEval 0-shot	**81.7**	71.9	73.0
GSM-8K 8-shot, CoT	**93.0**	91.7 11-shot	92.3 0-shot
MATH 4-shot, CoT	50.4	**58.5** Minerva prompt	40.5

https://ai.meta.com/blog/meta-llama-3/

Evaluación - MMLU

- **15908 preguntas múltiple opción** de conocimiento y razonamiento
- **57 áreas** incluyendo medicina, matemáticas, computación, derecho, entre otros

One of the reasons that the government discourages and regulates monopolies is that
(A) producer surplus is lost and consumer surplus is gained.
(B) monopoly prices ensure productive efficiency but cost society allocative efficiency.
(C) monopoly firms do not engage in significant research and development.
(D) consumer surplus is lost with higher prices and lower levels of output.

Evaluación - MMLU

Se mide el accuracy comparando el string exacto.

Ejemplo de prompt 2-shot

The following are multiple choice questions about high school mathematics.
How many numbers are in the list 25, 26, ..., 100?
(A) 75 (B) 76 (C) 22 (D) 23
Answer: B

Compute $i + i^2 + i^3 + \cdots + i^{258} + i^{259}$.
(A) -1 (B) 1 (C) i (D) -i
Answer: A

If 4 daps = 7 yaps, and 5 yaps = 3 baps, how many daps equal 42 baps?
(A) 28 (B) 21 (C) 40 (D) 30
Answer:

Evaluación - Tarea específica

También me puede interesar evaluar el LM sobre la tarea particular que estoy intentando resolver.

- Usar métricas y datasets específicos a mi tarea.

Por ejemplo, en traducción automática vimos:

- BLEU
- ChrF

Evaluación - Question Answering

Exact match

Porcentaje de respuestas generadas que son exactamente iguales a las respuestas de referencia

F1 score

Promedio de F1 score en función del solapamiento de tokens entre la respuesta generada y la de referencia (bag of tokens).

Sirven para QA extractivo, no tanto para QA generativo

Evaluación - Question Answering

BERTScore

Usar BERT para generar embeddings contextualizados de cada token y obtener un score que tenga en cuenta la semántica.

- **Recall:** Para cada token de la respuesta de referencia x, se obtiene el token de la respuesta generada $\tilde{x}$ más similar.

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\tilde{x}_j \in \tilde{x}} x_i \cdot \tilde{x}_j$$

- **Precisión:** Para cada token de la respuesta generada $\tilde{x}$, se obtiene el token de la respuesta generada x más similar.

$$P_{\text{BERT}} = \frac{1}{|\tilde{x}|} \sum_{\tilde{x}_j \in \tilde{x}} \max_{x_i \in x} x_i \cdot \tilde{x}_j$$

Evaluación - Question Answering - Ejemplo

Pregunta: *¿Es posible estudiar dos carreras a la par, como ing. en sistemas e ing. química?*

Respuesta correcta: *No hay problema. Puedes abrir dos carreras, y las materias que sean comunes a ambas se revalidan de una carrera para la otra de forma automática.*

Respuesta candidata 1: *Sí, es posible estudiar Ingeniería en Sistemas e Ingeniería Química en simultáneo. Si lo haces, ten en cuenta que las unidades curriculares (materias) que apruebes y sean comunes a ambas carreras se validarán automáticamente en las dos escolaridades.*

Respuesta candidata 2: *No, no es posible estudiar Ingeniería en Sistemas e Ingeniería Química en simultáneo. Ten en cuenta que las unidades curriculares (materias) que apruebes y sean comunes a ambas carreras solo se validarán para la escolaridad correspondiente a la que estés cursando.*

	Exact Match	F1 Score	BERTScore F1
Candidata 1	0.0	0.323	0.714
Candidata 2	0.0	0.375	0.700

No capturan bien la semántica!

Evaluación - Question Answering

LLM-as-a-Judge

Usar LLM para evaluar, comparando respuesta de referencia con respuesta generada (o comparando entre respuestas generadas).

Se ha observado que hay un alto acuerdo con la evaluación humana utilizando GPT-4 como evaluador *(Zheng et al., 2023).*

Problemas:

- **Sesgo de posición:** Favorecer ciertas posiciones sobre otras
- **Sesgo de verbosidad:** Favorecer respuestas más largas
- **Sesgo de autopercepción:** Favorecer respuestas generadas por si mismo.

Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena
Zheng et al. (2023)

Evaluación - Question Answering - Ejemplo

LLM-as-a-Judge (1 al 5)

Candidata 1:

- **CoT:** La respuesta a evaluar coincide completamente con la respuesta de referencia de una manera veraz y tiene toda la información que se expresa en la respuesta de referencia. Ambas respuestas afirman que es posible estudiar dos carreras simultáneamente y que las materias comunes se revalidan automáticamente entre las dos carreras.
- **Score: 5**

Candidata 2:

- **CoT:** La respuesta a evaluar contradice la respuesta de referencia de una manera que la información no es veraz en comparación con la respuesta de referencia. La respuesta a evaluar indica que no es posible estudiar Ingeniería en Sistemas e Ingeniería Química simultáneamente, mientras que la respuesta de referencia afirma que sí es posible abrir dos carreras y que las materias comunes se revalidan automáticamente de una carrera a otra.
- **Score: 1**

Evaluación Humana

- La evaluación automática en tareas generativas es un área de investigación muy abierta
- No hay un sustituto claro para la evaluación humana *(Kamalloo et al., 2023)*
- Chatbot Arena (https://lmarena.ai/) es una plataforma que permite a las personas comparar y evaluar modelos, y ofrece una tabla de posiciones bastante confiable

Evaluating Open-Domain Question Answering in the Era of Large Language Models
Kamalloo et al. (2023)

Bibliografía

- Function Call: Documentación de Hugging Face (transformers)
- Perplejidad: Jurafsky & Martin, 3rd Ed. (draft) - Capítulo 10
- MMLU: Jurafsky & Martin, 3rd Ed. (draft) - Capítulo 12
- BERTScore: Jurafsky & Martin, 3rd Ed. (draft) - Capítulo 13
- Evaluación QA: Jurafsky & Martin, 3rd Ed. (draft) - Capítulo 14
- Papers…