

# Dataset enrichment using LLM's ✨

Jeroen Overschie

PyData Eindhoven — 27th of November 2023

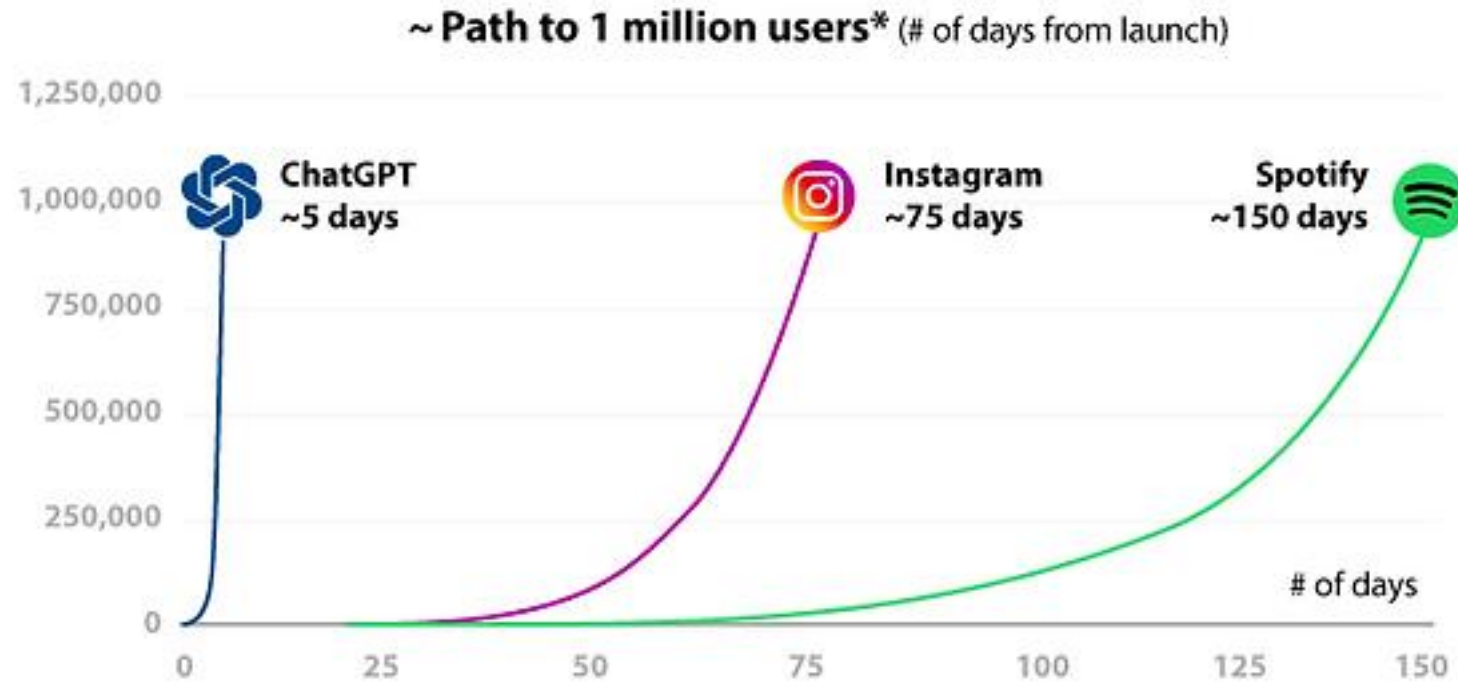


Jeroen Overschie

Machine Learning Engineer

LLM's are booming

# LLM's are booming



Sources: Google, Subredditstats, Media Reports

# LLM's are booming



Information  
extraction



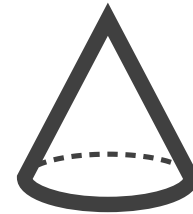
Search



Summarization



Text  
generation

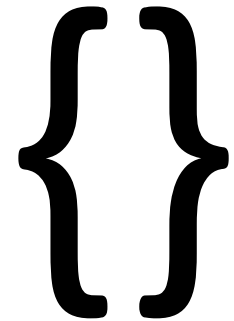
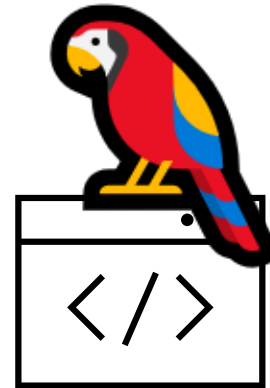
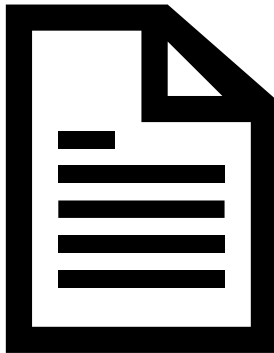


Classification

”

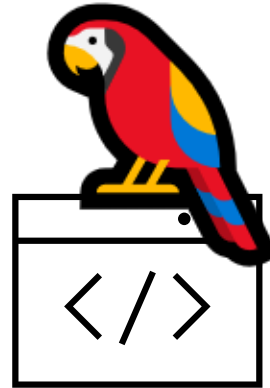
Information  
extraction

“Given the text that  
follows, extract  
properties X Y Z”



“Given the text that  
follows, extract  
properties X Y Z”

?



{ }



Foto's Locatie

**Nieuw**

**Te huur: Appartement Hageland 104 in Amsterdam**  
1066 SB (Sloten/Nieuw-Sloten)

**€ 1.900 per maand**

70 m<sup>2</sup> 3 kamers Gemeubileerd

**Beschrijving**

Apartment on 1 minute walking distance of the shopping center, supermarket and bus station.

Details:

- Size of the property: 70 m<sup>2</sup>
- Number of bedrooms: 2 (the 2nd bedroom has no windows)
- Number of bathrooms: 1
- Type of house: Apartment
- Construction year of the house: 1992
- Interior decoration: Furnished
- Flooring: Laminat
- Quality of public transportation: Good

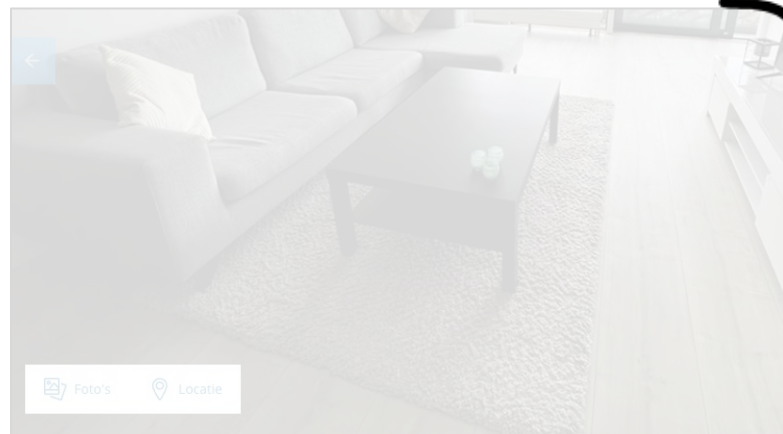
Also in this rental house:

- Parking possibility: Private parking place outside
- Separate shower
- Separate toilet
- Elevator
- Storage: box
- 3rd floor
- Balcony (facing South-West)

“Given the text that follows, extract properties X Y Z”







Foto's Locatie

### Nieuw

## Te huur: Appartement Hageland 104 in Amsterdam

1066 SB (Sloten/Nieuw-Sloten)

€ 1.900 per maand

70 m<sup>2</sup> 3 kamers Gemeubileerd

### Beschrijving

Apartment on 1 minute walking distance of the shopping center, supermarket and bus station.

Details:

- Size of the property: 70 m<sup>2</sup>
- Number of bedrooms: 2 (the 2nd bedroom has no windows)
- Number of bathrooms: 1
- Type of house: Apartment
- Construction year of the house: 1992
- Interior decoration: Furnished
- Flooring: Laminate
- Quality of public transportation: Good

Also in this rental house:

- Parking possibility: Private parking place outside
- Separate shower
- Separate toilet
- Elevator
- Storage: box
- 3rd floor
- Balcony (facing South-West)

pararius

Plaats, wijk, buurt of straat



+ Woning verhuren

Inloggen



Filters

Prijs

Slaapkamers

Oppervlakte

Interieur

Kaart

5.145 Huurwoningen in Nederland

Nieuwste eerst



Nieuw

### Appartement Hageland 104

1066 SB Amsterdam (Sloten/Nieuw-Sloten)

€ 1.900 per maand

70 m<sup>2</sup> 3 kamers Gemeubileerd

Amsterdam Housing



Uitgelicht

### Appartement Lichtstraat

5611 XD Eindhoven (Witte Dame)

€ 2.195 per maand

103 m<sup>2</sup> 2 kamers Gemeubileerd

Housing Totaal



Nieuw

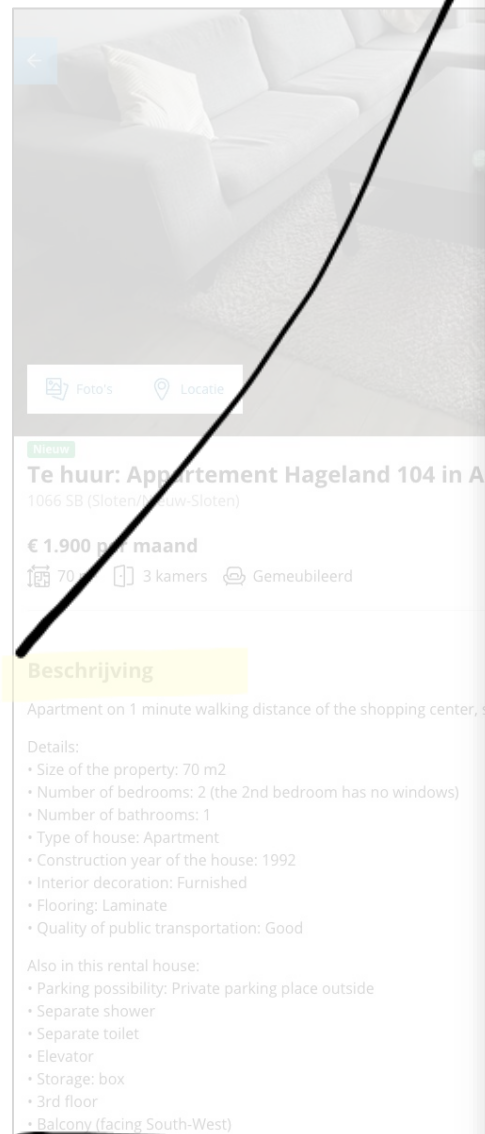
### Huis Bredestraat 11

5242 CD Rosmalen (Molenhoek)

€ 2.195 per maand

118 m<sup>2</sup> 3 kamers Gemeubileerd

Kingstay



## Beschrijving

Apartment on 1 minute walking distance of the shopping center, supermarket and bus station.

Details:

- Size of the property: 70 m2
- Number of bedrooms: 2 (the 2nd bedroom has no windows)
- Number of bathrooms: 1
- Type of house: Apartment
- Construction year of the house: 1992
- Interior decoration: Furnished
- Flooring: Laminate
- Quality of public transportation: Good

Also in this rental house:

- **Parking possibility:** Private parking place outside
- Separate shower
- Separate toilet
- **Elevator**
- Storage: box
- 3rd floor
- **Balcony (facing South-West)**

Conditions:

- **Pets not allowed**
- **Sharing not allowed**
- **Students not allowed**
- **Smoking not allowed**
- Children allowed
- Tenancy agreements subject to owner's consent
- Measurements not conform NEN 2580

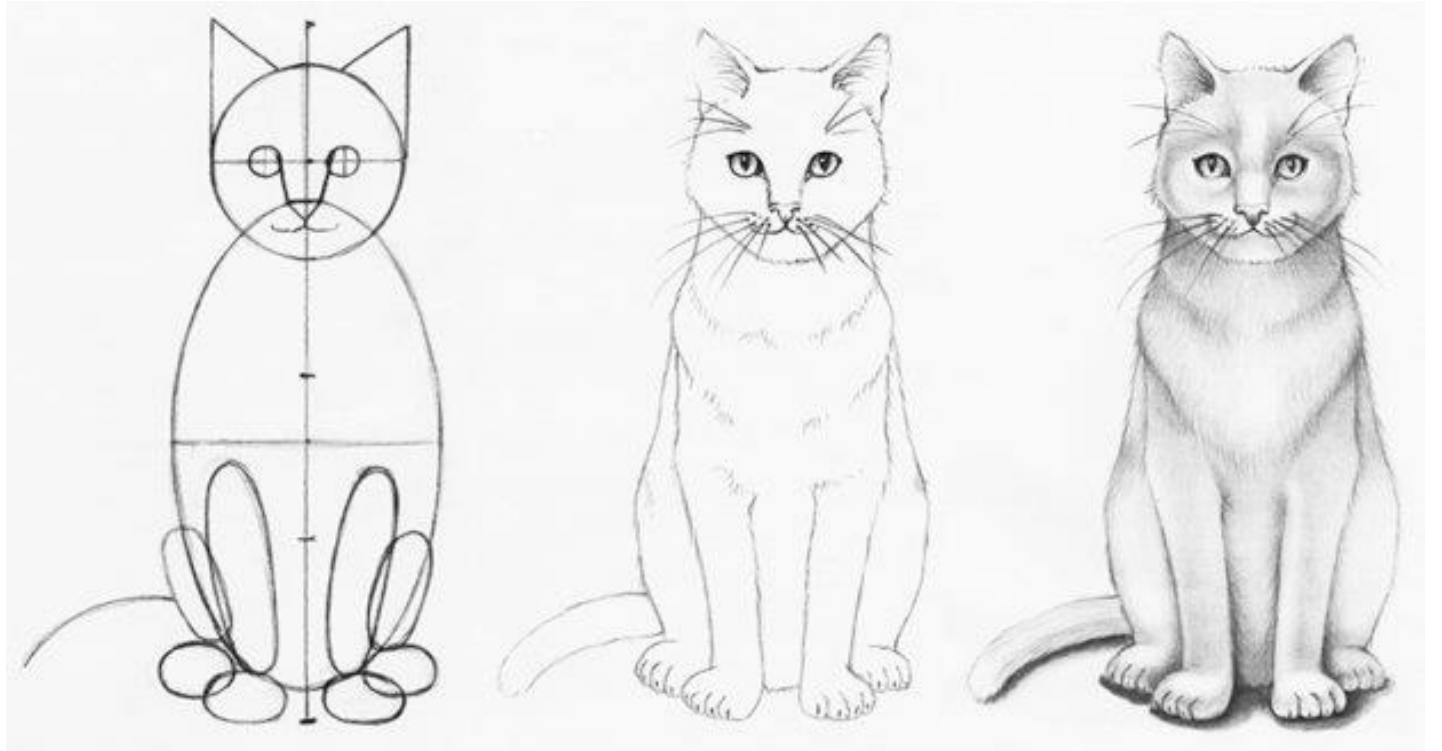
The rental price of this house is exclusive Gas/Electricity/Water, TV/Internet and local Taxes.

— Minder

Can we  
extract these  
properties  
using a **LLM**?

# Let's start prompting the LLM

We will **gradually progress** from  
**naïve** to **sophisticated** approaches



## Prompt engineering

### Prompt

A house listing will follow, as it appeared on a housing website. Please extract the following information from the listing:

- Is pet friendly (yes/no/unknown)

## House listing

{house}

## Prompt

A house listing will follow, as it appeared on a housing website. Please extract the following information from the listing:

- Is pet friendly (yes/no/unknown)

## House listing  
{house}



 Appartement aan de Hageland 104 (€1900.0)

### ## Description

Apartment on 1 minute walking distance of the shopping center, supermarket and bus station.

Details:

- Size of the property: 70 m2
- Number of bedrooms: 2 (the 2nd bedroom has no windows)
- Number of bathrooms: 1
- Type of house: Apartment
- Construction year of the house: 1992
- Interior decoration: Furnished
- Flooring: Laminate
- Quality of public transportation: Good

Also in this rental house:

- Parking possibility: Private parking place outside
- Separate shower
- Separate toilet
- Elevator
- Storage: box
- 3rd floor
- Balcony (facing South-West)

Conditions:

- Pets not allowed
- Sharing not allowed
- Students not allowed
- Smoking not allowed
- Children allowed
- Tenancy agreements subject to owner's consent
- Measurements not conform NEN 2580

The rental price of this house is exclusive Gas/Electricity/Water, TV/Internet and local Taxes.

### ## Eigenschappen

 Neighborhood: Sloten/Nieuw-Sloten

 Rooms: 3

 Surface area: 70.0 m<sup>2</sup>

 Real estate agent: Amsterdam Housing

 URL: <https://www.pararius.nl/appartement-te-huur/amsterdam/2a54c6e5/hageland>

 Google Maps link: <https://maps.google.com/?q=Hageland+104%2C+Sloten%2FNieuw-Sloten>

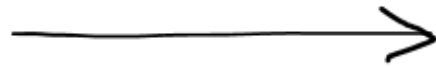
## Prompt engineering

### Prompt

A house listing will follow, as it appeared on a housing website. Please extract the following information from the listing:

**- Is pet friendly (yes/no/unknown)**

## House listing  
{house}



### Output

Pet friendly: No

Actual result from GPT-3.5 turbo (OpenAI)  
- using temperature = 0.0

## Prompt engineering

### Prompt

A house listing will follow, as it appeared on a housing website. Please extract the following information from the listing:

- Is pet friendly (yes/no/unknown, quote: "...")

Provide a quote from the listing if possible.

## House listing

{house}



### Output

- Is pet friendly: No ("Pets not allowed")

Quote: "Pets not allowed"

Actual result from GPT-3.5 turbo (OpenAI)  
- using temperature = 0.0

## Prompt engineering

### Prompt

A house listing will follow, as it appeared on a housing website. Please extract the following information from the listing:

- Is pet friendly (yes/no/unknown, quote: "...")
- Has a garden (yes/no/unknown, quote: "...")
- Has a parking place (yes/no/unknown, quote: "...")
- Is suitable for a couple (yes/no/unknown, quote: "...")
- Has a balcony (yes/no/unknown, quote: "...")

## House listing  
{house}



### Output

- Is pet friendly: No
- Has a garden: Unknown
- Has a parking place: Yes (private parking place outside)
- Is suitable for a couple: Unknown
- Has a balcony: Yes (facing South-West)

Actual result from GPT-3.5 turbo (OpenAI)  
- using temperature = 0.0



Going  
structured

## Prompt engineering

### Prompt

A house listing will follow, as it appeared on a housing website.  
Please extract the following information from the listing:

- ...  
- ...  
...

**Provide the output in a JSON object.**

## House listing



### Output

```
{  
  "pet_friendly": "no",  
  "has_garden": "unknown",  
  "has_parking_place": "yes",  
  "suitable_for_couple": "unknown",  
  "has_balcony": "yes"  
}
```

Actual result from GPT-3.5 turbo (OpenAI)  
- using temperature = 0.0

# Prompt engineering

## Prompt

A house listing will follow, as it appeared on a housing website. Please extract the following information from the listing in a JSON response:

```
...
{
  "is_pet_friendly": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
  "has_garden": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
  "has_parking_place": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
  "is_suitable_for_couple": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
  "has_balcony": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
}
```



## Output

```
{
  "is_pet_friendly": {
    "value": "no",
    "quote": "Pets not allowed"
  },
  "has_garden": {
    "value": "unknown",
    "quote": "-"
  },
  "has_parking_place": {
    "value": "yes",
    "quote": "Parking possibility: Private parking place outside"
  },
  "is_suitable_for_couple": {
    "value": "unknown",
    "quote": "-"
  },
  "has_balcony": {
    "value": "yes",
    "quote": "Balcony (facing South-West)"
  }
}
```

Actual result from GPT-3.5 turbo (OpenAI)  
- using temperature = 0.0

# Prompt engineering

## Prompt

```
obj = json.loads(  
    ...  
    {  
        "is_pet_friendly": {  
            "value": ..., # yes/no/unknown  
            "quote": ..., # exact quote from listing  
        },  
        "has_garden": {  
            "value": ..., # yes/no/unknown  
            "quote": ..., # exact quote from listing  
        },  
        "has_parking_place": {  
            "value": ..., # yes/no/unknown  
            "quote": ..., # exact quote from listing  
        },  
        "is_suitable_for_couple": {  
            "value": ..., # yes/no/unknown  
            "quote": ..., # exact quote from listing  
        },  
        "has_balcony": {  
            "value": ..., # yes/no/unknown  
            "quote": ..., # exact quote from listing  
        },  
    }  
    ...  
)
```

## Output

```
{  
    "is_pet_friendly": {  
        "value": "no",  
        "quote": "Pets not allowed"  
    },  
    "has_garden": {  
        "value": "unknown",  
        "quote": "-"  
    },  
    "has_parking_place": {  
        "value": "yes",  
        "quote": "Parking possibility: Private parking place outside"  
    },  
    "is_suitable_for_couple": {  
        "value": "unknown",  
        "quote": "-"  
    },  
    "has_balcony": {  
        "value": "yes",  
        "quote": "Balcony (facing South-West)"  
    }  
}
```

Actual result from GPT-3.5 turbo (OpenAI)  
- using temperature = 0.0

# Prompt engineering

## Prompt

```
obj = json.loads(
```

```
A house listing will follow, as it appeared on a real estate website. Please extract the following information from the listing in a JSON response:
```

```
)
```

```
{
```

```
  "is_pet_friendly": {
```

```
    "value": "yes",
```

```
    "quote": "Pets allowed with additional deposit."
```

```
  },
```

```
  "has_garden": {
```

```
    "value": "yes",
```

```
    "quote": "Large backyard garden with fruit trees."
```

```
  },
```

```
  "has_parking_place": {
```

```
    "value": "yes",
```

```
    "quote": "Two-car garage and additional street parking available."
```

```
  },
```

```
  "is_suitable_for_couple": {
```

```
    "value": "yes",
```

```
    "quote": "Perfect for a couple or small family."
```

```
  },
```

```
  "has_balcony": {
```

```
    "value": "yes",
```

```
    "quote": "Spacious balcony with beautiful views."
```

```
  },
```

```
  "is_suitable_for_couple": {
```

```
    "value": "yes",
```

```
    "quote": "Perfect for a couple or small family."
```

```
  },
```

```
  "has_balcony": {
```

```
    "value": "yes",
```

```
    "quote": "Spacious balcony with beautiful views."
```

```
  },
```

```
  "is_suitable_for_couple": {
```

```
    "value": "yes",
```

```
    "quote": "Perfect for a couple or small family."
```

## Output

Actual result from GPT-3.5 turbo (OpenAI)  
- using temperature = 0.0

```
"House listing"
```

```
{
```

```
  "title": "Beautiful Modern House",
```

```
  "description": "Spacious and modern house with stunning views. Perfect for a couple or small family.",
```

```
  "location": "123 Main Street, Anytown, USA",
```

```
  "price": "$500,000",
```

```
  "is_pet_friendly": {
```

```
    "value": "yes",
```

```
    "quote": "Pets allowed with additional deposit."
```

```
  },
```

```
  "has_garden": {
```

```
    "value": "yes",
```

```
    "quote": "Large backyard garden with fruit trees."
```

```
  },
```

```
  "has_parking_place": {
```

```
    "value": "yes",
```

```
    "quote": "Two-car garage and additional street parking available."
```

```
  },
```

```
  "is_suitable_for_couple": {
```

```
    "value": "yes",
```

```
    "quote": "Perfect for a couple or small family."
```

```
  },
```

```
  "has_balcony": {
```

```
    "value": "yes",
```

```
    "quote": "Spacious balcony with beautiful views."
```

```
  },
```

```
}
```

# Prompt engineering

## Prompt

```
obj = json.loads(  
  ...  
)
```

A house listing will follow, as it appeared on a real estate website. Please extract the following information from the listing in a JSON response:

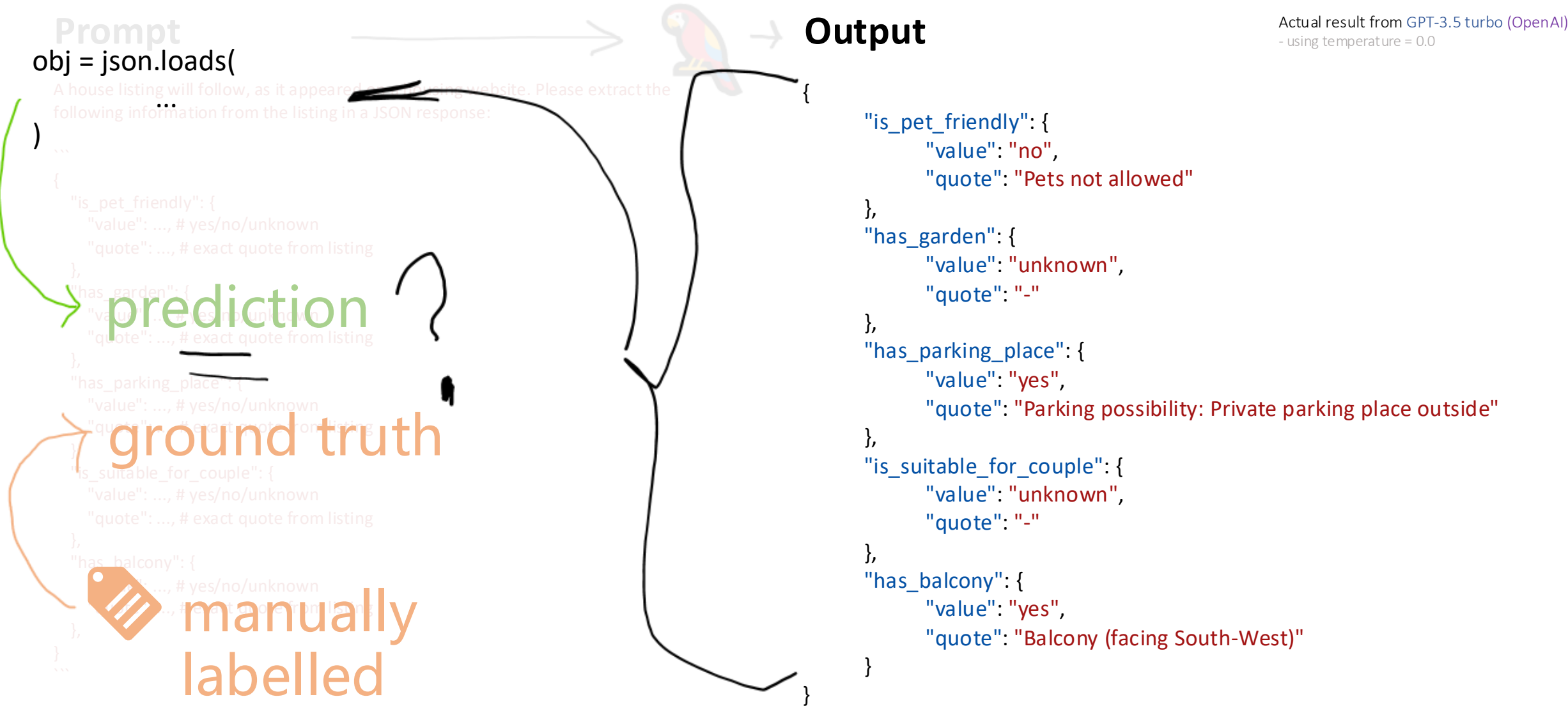
```
{  
  "is_pet_friendly": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
  "has_garden": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
  "has_parking_place": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
  "is_suitable_for_couple": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
  "has_balcony": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
}
```

## Output

```
{  
  "is_pet_friendly": {  
    "value": "no",  
    "quote": "Pets not allowed"  
  },  
  "has_garden": {  
    "value": "unknown",  
    "quote": "-"  
  },  
  "has_parking_place": {  
    "value": "yes",  
    "quote": "Parking possibility: Private parking place outside"  
  },  
  "is_suitable_for_couple": {  
    "value": "unknown",  
    "quote": "-"  
  },  
  "has_balcony": {  
    "value": "yes",  
    "quote": "Balcony (facing South-West)"  
  },  
}
```

Actual result from GPT-3.5 turbo (OpenAI)  
- using temperature = 0.0

# Prompt engineering



# Evaluating LLM performance

For information extraction

ground truth

```
{
  "is_pet_friendly": {
    "was_extracted": true,
    "quote": "Pets not allowed",
    "value": false
  },
  "has_garden": {
    "was_extracted": false,
    "quote": null,
    "value": null
  },
  "has_parking_place": {
    "was_extracted": true,
    "quote": "Parking possibility: Permit with the
municipality (please check waiting list on the website
of the municipality)",
    "value": true
  },
  "is_suitable_for_couple": {
    "was_extracted": true,
    "quote": null,
    "value": true
  },
  "has_balcony": {
    "was_extracted": true,
    "quote": "French balcony",
    "value": true
  }
}
```

prediction

```
{
  "is_pet_friendly": {
    "was_extracted": true,
    "quote": "Pets not allowed",
    "value": false
  },
  "has_garden": {
    "was_extracted": false,
    "quote": null,
    "value": null
  },
  "has_parking_place": {
    "was_extracted": true,
    "quote": "Parking possibility: Permit with the
municipality (please check waiting list on the website
of the municipality)",
    "value": null
  },
  "is_suitable_for_couple": {
    "was_extracted": false,
    "quote": null,
    "value": null
  },
  "has_balcony": {
    "was_extracted": true,
    "quote": "French balcony",
    "value": true
  }
}
```

Score 

60% of entries  
correct

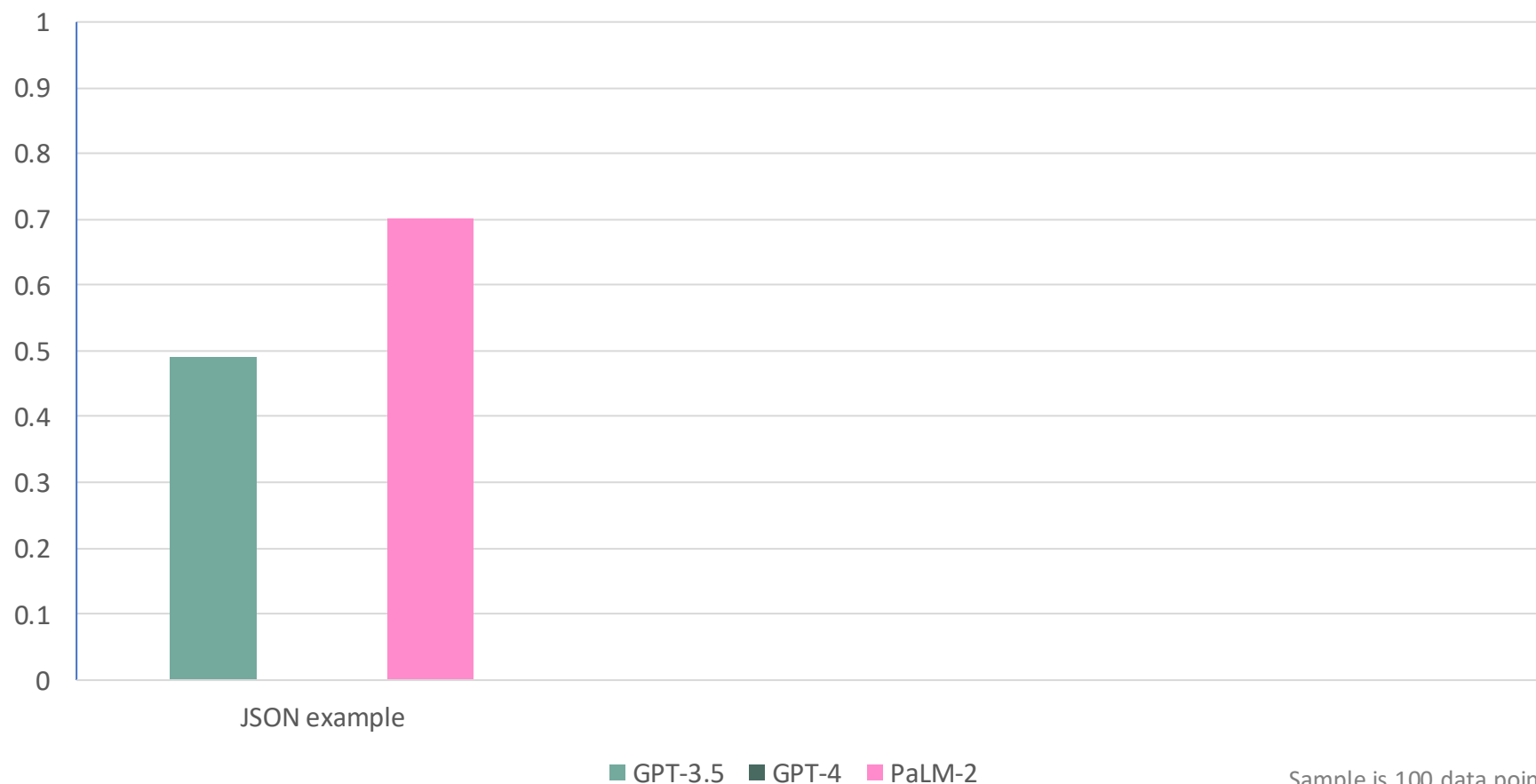
=> **0.6 score**



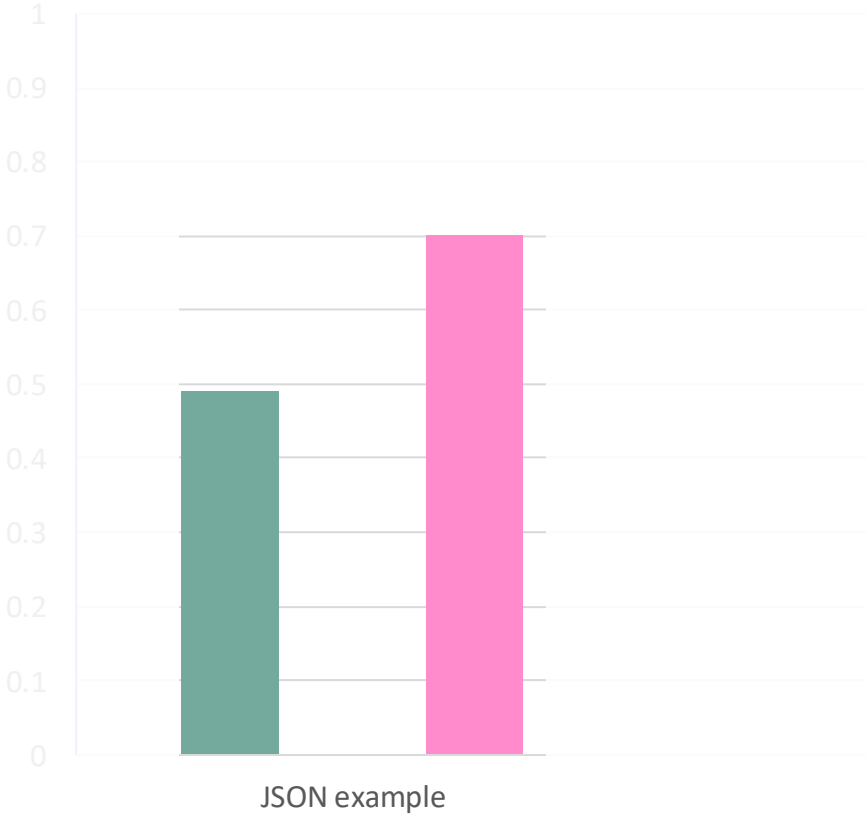
# Evaluating LLM performance

For information extraction

GPT 3.5, GPT 4 and PaLM compared



# GPT-4 got 100% failure rate for decoding JSON



■ GPT-3.5 ■ GPT-4

GPT 3.5, GPT 4 are

Batch 1 [Provide Feedback](#)

Experiment ID: 496807957802400095    Artifact Location: file:///Users/dunnkers/git/pararius-scraper/mlruns/496807957802400095

> Description [Edit](#)

Q metrics.rmse < 1 and params.model = "tree"

i

Time created ▾

State: Active ▾

Sort: Created ▾

Columns ▾

Table   Chart   Evaluation   **Experimental**

|                          |                       |                      |                |          | Metrics       |               | Parameters  |  |
|--------------------------|-----------------------|----------------------|----------------|----------|---------------|---------------|-------------|--|
| <input type="checkbox"/> | <input type="radio"/> | Run Name             | Created        | Duration | percentage_co | method        | temperature |  |
| <input type="checkbox"/> | <input type="radio"/> | clumsy-fly-857       | ✓ 17 hours ago | 5.5s     | 0.8           | palm_2_jso... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | resilient-auk-682    | ✓ 17 hours ago | 20.1s    | 0.6           | palm_2_jso... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | powerful-grouse-53   | ✓ 17 hours ago | 7.0s     | 1             | palm_2_jso... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | big-lamb-468         | ✗ 17 hours ago | 30.6s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | painted-skink-612    | ✗ 18 hours ago | 21.1s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | illustrious-mare-761 | ✗ 18 hours ago | 10.2s    | -             | gpt_4_json... | -           |  |
| <input type="checkbox"/> | <input type="radio"/> | legendary-sloth-657  | ✗ 18 hours ago | 10.1min  | -             | gpt_4_json... | -           |  |
| <input type="checkbox"/> | <input type="radio"/> | rumbling-robin-590   | ✗ 18 hours ago | 11.3s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | thoughtful-lamb-796  | ✗ 18 hours ago | 14.1s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | zealous-cat-719      | ✗ 18 hours ago | 12.9s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | wistful-crane-575    | ✗ 18 hours ago | 13.3s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | dashing-snipe-236    | ✗ 18 hours ago | 23.5s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | luxuriant-lynx-611   | ✗ 18 hours ago | 10.1s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | tasteful-kit-346     | ✗ 19 hours ago | 8.6min   | -             | gpt_4_json... | -           |  |
| <input type="checkbox"/> | <input type="radio"/> | exultant-mink-510    | ✗ 19 hours ago | 16.5s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | funny-stag-716       | ✗ 19 hours ago | 6.0min   | -             | gpt_4_json... | -           |  |
| <input type="checkbox"/> | <input type="radio"/> | brawny-carp-935      | ✗ 19 hours ago | 14.7s    | -             | gpt_4_json... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | secretive-shrew-956  | ✓ 19 hours ago | 5.6s     | 0.3           | gpt_35_jso... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | bemused-wolf-623     | ✓ 19 hours ago | 5.6s     | 0.3           | gpt_35_jso... | 0.0         |  |
| <input type="checkbox"/> | <input type="radio"/> | righteous-zebra-905  | ✓ 19 hours ago | 8.2s     | 0.5           | gpt_35_jso... | 0.0         |  |

GPT-4 got 100%  
failure rate for  
decoding JSON

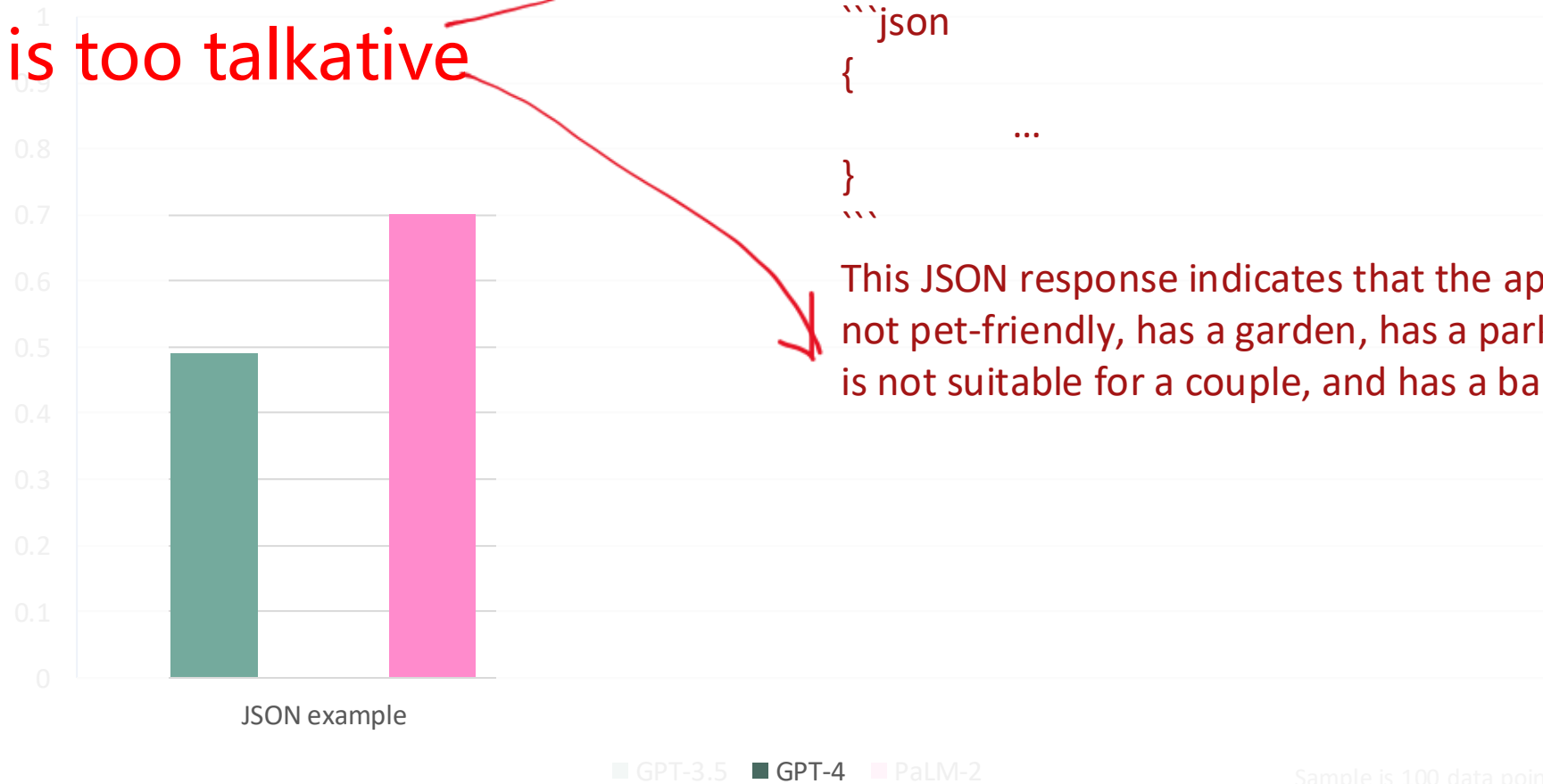
GPT-4 is too talkative

## Output

Based on the house listing provided, here is the  
extracted information in a JSON response:

```
```json  
{  
  ...  
}  
```
```

This JSON response indicates that the apartment is  
not pet-friendly, has a garden, has a parking place,  
is not suitable for a couple, and has a balcony.

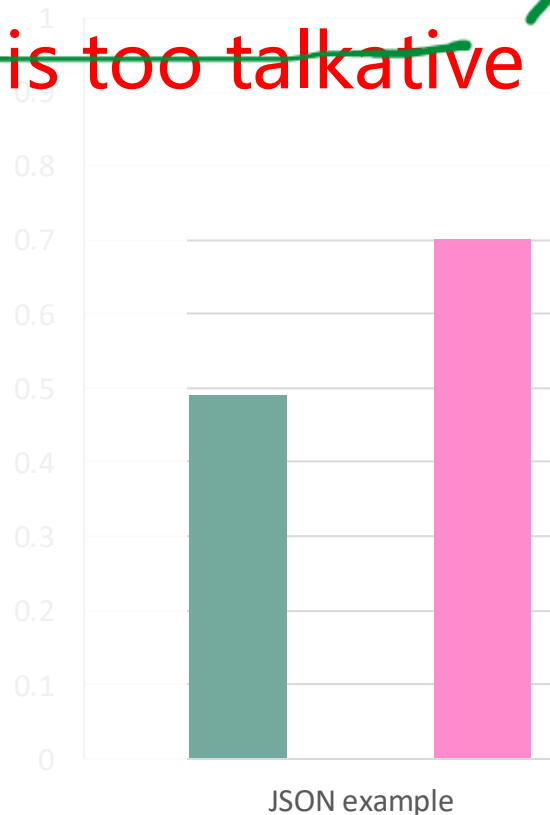


GPT-4 got 100%  
failure rate for  
decoding JSON

~~GPT-4 is too talkative~~

Remediated with an **extra statement** in the prompt:

Return nothing else except a valid JSON object. Do not surround the JSON object with backticks or any other text.



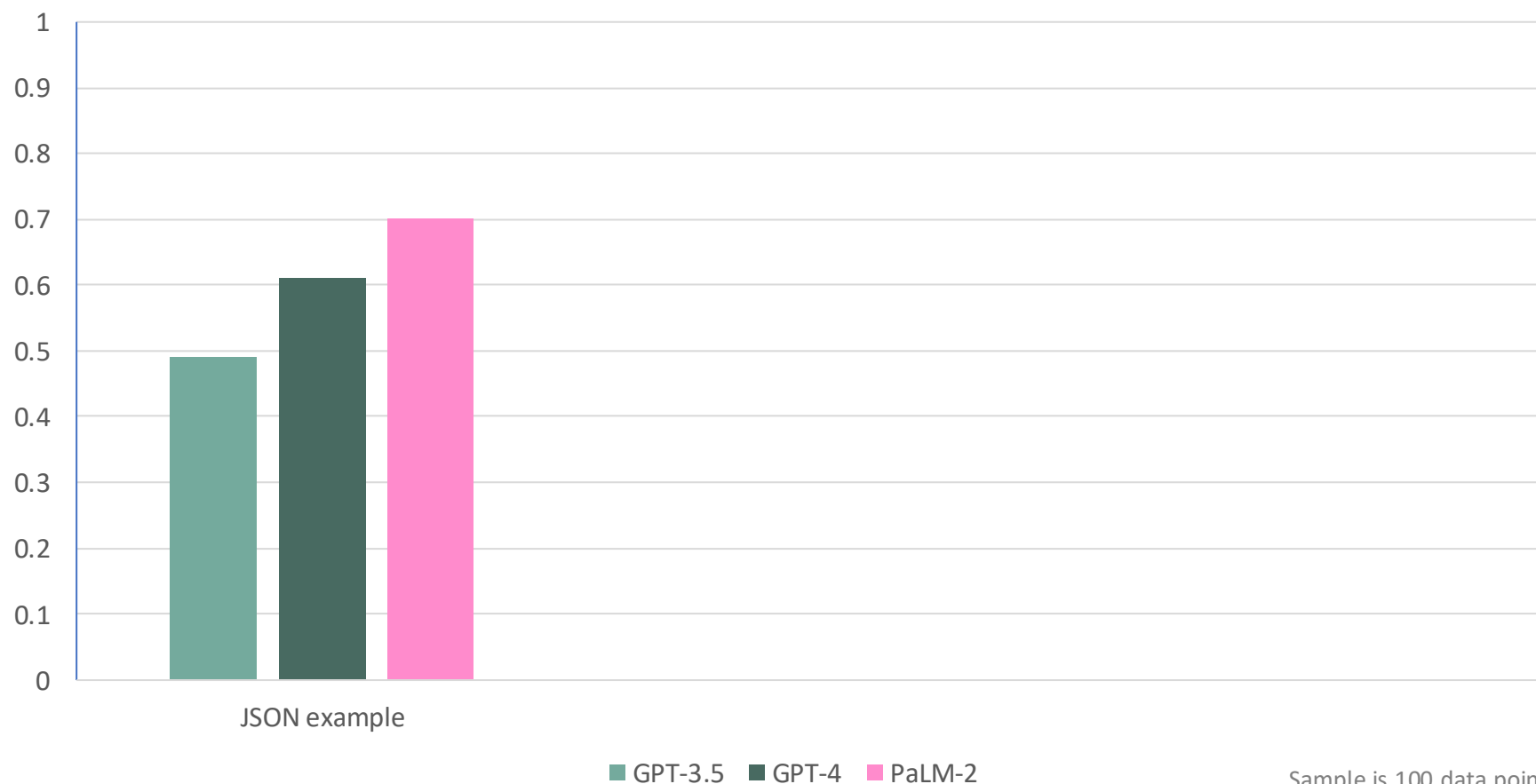
■ GPT-3.5 ■ GPT-4 ■ PaLM-2

Sample is 100 data points

# Evaluating LLM performance

For information extraction

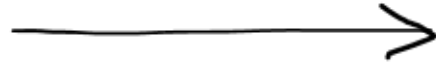
GPT 3.5, GPT 4 and PaLM compared



Can we do  
better?

# Prompt engineering

## Prompt



## Output

A house listing will follow, as it appeared on a housing website. Please extract the following information from the listing in a JSON response:

```
...  
{  
  "is_pet_friendly": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
  "has_garden": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
  "has_parking_place": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
  "is_suitable_for_couple": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
  "has_balcony": {  
    "value": ..., # yes/no/unknown  
    "quote": ..., # exact quote from listing  
  },  
}  
...
```

# Prompt engineering

Prompt



Output

A house listing will follow, as it appeared on a housing website. Please extract the following information from the listing in a JSON response:

`output_template = HouseFeatures.model_json_schema()`

`output_template_str = json.dumps(output_template)`

```
{
  "is_pet_friendly": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
  "has_garden": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
  "has_parking_place": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
  "is_suitable_for_couple": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
  "has_balcony": {
    "value": ..., # yes/no/unknown
    "quote": ..., # exact quote from listing
  },
}
```

Pydantic model

```
ex.json
Open with Xcode

{
  "$defs": {
    "HouseFeature": {
      "properties": {
        "was_extracted": {
          "description": "Whether the feature was extracted",
          "title": "Was Extracted",
          "type": "boolean"
        },
        "quote": {
          "anyOf": [
            {
              "type": "string"
            },
            {
              "type": "null"
            }
          ],
          "default": null,
          "description": "Exact quote from the listing",
          "title": "Quote"
        },
        "value": {
          "anyOf": [
            {
              "type": "boolean"
            },
            {
              "type": "null"
            }
          ],
          "default": null,
          "description": "Whether the house has the feature",
          "title": "Value"
        }
      },
      "required": [
        "was_extracted"
      ],
      "title": "HouseFeature",
      "type": "object"
    }
  },
  "description": "Correctly extracted house listing features",
  "properties": {
    "is_pet_friendly": {
      "$ref": "#/$defs/HouseFeature"
    },
    "has_garden": {
      "$ref": "#/$defs/HouseFeature"
    },
    "has_parking_place": {
      "$ref": "#/$defs/HouseFeature"
    },
    "is_suitable_for_couple": {
      "$ref": "#/$defs/HouseFeature"
    },
    "has_balcony": {
      "$ref": "#/$defs/HouseFeature"
    }
  },
  "required": [
    "is_pet_friendly",
    "has_garden",
    "has_parking_place",
    "is_suitable_for_couple",
    "has_balcony"
  ],
  "title": "HouseFeatures",
  "type": "object"
}
```



# Prompt engineering

## Prompt

```
output_template = HouseFeatures.model_json_schema()  
output_template_str = json.dumps(output_template)
```

A house listing will follow. Your task is to extract details from the house listing. Fill in the following template:

```
# template  
{output_template_str}
```

```
# house listing  
{str(house)}
```



## Output

```
{  
  "is_pet_friendly": {  
    "was_extracted": false  
  },  
  "has_garden": {  
    "was_extracted": false  
  },  
  "has_parking_place": {  
    "was_extracted": true,  
    "value": true,  
    "quote": "Parking in front of the building is available through a permit system"  
  },  
  "is_suitable_for_couple": {  
    "was_extracted": true,  
    "value": true,  
    "quote": "preference for single household"  
  },  
  "has_balcony": {  
    "was_extracted": true,  
    "value": true,  
    "quote": "east-facing balcony (morning sun) spans the entire width of the apartment and overlooks the beautiful courtyard"  
  }  
}
```

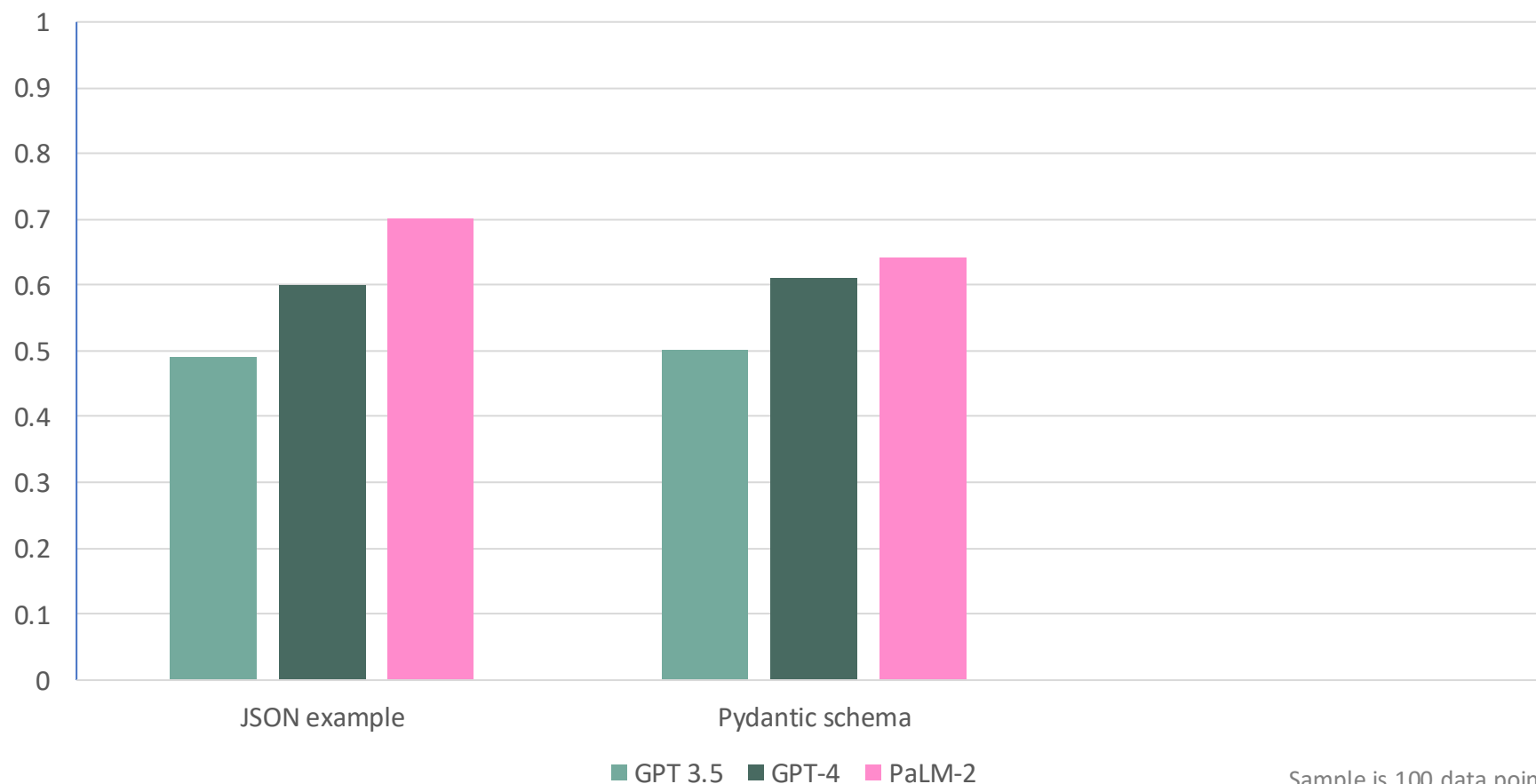


```
house_features = HouseFeatures.model_validate_json(  
  response_text  
)
```

# Evaluating LLM performance

For information extraction

GPT 3.5, GPT 4 and PaLM compared



Can we do even  
better?

# Function calling and other API updates

We're announcing updates including more steerable API models, function calling capabilities, longer context, and lower prices.

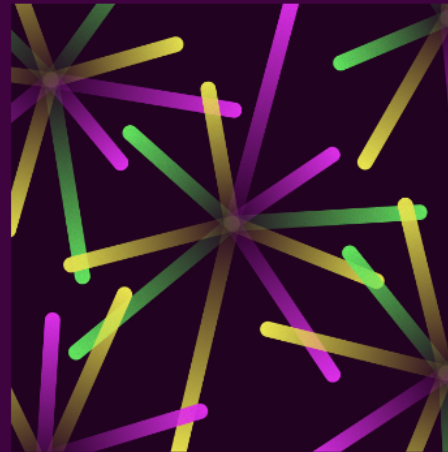


Illustration: Ruby Chen

June 13, 2023

## Authors

[Atty Eleti](#) ↓

[Jeff Harris](#) ↓

[Logan Kilpatrick](#) ↓

[Announcements](#), [Product](#)

*July 20, 2023 update:*

We previously communicated to developers that `gpt-3.5-turbo-0301`, `gpt-4-0314` and `gpt-4-32k-0314` models were scheduled for sunset on Sept 13, 2023. After reviewing feedback from customers and our community, we are extending support for those models until at least June 13, 2024.

## Prompt engineering

```
response = openai.ChatCompletion.create(  
    model="gpt-3.5-turbo-0613",  
    deployment_id="gpt-35-turbo-us",  
    messages=messages,  
    functions=functions,  
    function_call={  
        "name": "extract_house_features"  
    },  
)
```

Force  
calling  
function

```
functions = [  
    {  
        "name": "extract_house_features",  
        "type": "function",  
        "description": "Extract features and properties from a house listing text.",  
        "parameters": {  
            "type": "object",  
            "properties": {  
                "is_pet_friendly": {  
                    "type": "boolean",  
                    "description": "Whether the house allows pets"  
                },  
                "has_garden": {  
                    "type": "boolean",  
                    "description": "Whether the house has a garden"  
                },  
                "has_parking_place": {  
                    "type": "boolean",  
                    "description": "Whether the house has a parking place"  
                },  
                "is_suitable_for_couple": {  
                    "type": "boolean",  
                    "description": "Whether the house is suitable for couples"  
                },  
                "has_balcony": {  
                    "type": "boolean",  
                    "description": "Whether the house has a balcony"  
                },  
            },  
            "required": [  
                "is_pet_friendly",  
                "has_garden",  
                "has_parking_place",  
                "is_suitable_for_couple",  
                "has_balcony"  
            ]  
        }  
    }  
]
```

# Prompt engineering

## Prompt

{house}



## Output

{

```
"name": "extract_house_features",  
"arguments": "{\n  \"is_pet_friendly\": false,\n  \"has_garden\": false,\n  \"has_parking_place\": true,\n  \"is_suitable_for_couple\": true,\n  \"has_balcony\": true\n}"
```

}

```
response = openai.ChatCompletion.create(  
    model="gpt-3.5-turbo-0613",  
    deployment_id="gpt-35-turbo-us",  
    messages=messages,  
    functions=functions,  
    function_call={"name": "extract_house_features"},  
)
```

## Context

Assistant is a large language model **designed to extract structured data from text.**



```
function_call_args = json.loads(  
    function_call["arguments"]  
)  
extract_house_features(  
    **function_call_args  
)
```

# Some extra juice



jxnl / instructor

<> Code

Issues 11

Pull requests 4

Actions

Projects

Security

Insights

instructor

Public

Watch 21

Fork 72

Starred 944

main

42 branches

15 tags

Go to file

Add file

<> Code

jxnl add comments on why dumb rag

21b6c7d yesterday

194 commits

|                      |                                                                          |              |
|----------------------|--------------------------------------------------------------------------|--------------|
| .github              | Update workflow to only run on pushes to main branch                     | 2 months ago |
| .vscode              | Improve Documentation (#96)                                              | 2 weeks ago  |
| docs                 | add comments on why dumb rag                                             | yesterday    |
| examples             | Merge branch 'helpers'                                                   | 5 days ago   |
| instructor           | fix bug                                                                  | 5 days ago   |
| tests                | Reasking logic on validations (#98)                                      | 2 weeks ago  |
| .gitignore           | deploy                                                                   | 2 months ago |
| LICENSE              | Initial commit                                                           | 3 months ago |
| README.md            | Update README.md                                                         | 2 weeks ago  |
| blog.md              | reorganized blog vs readme                                               | 3 months ago |
| mkdocs.yml           | update docs                                                              | 5 days ago   |
| poetry.lock          | Upgrade typer (#90)                                                      | 3 weeks ago  |
| pyproject.toml       | Maybe helper to dynamically create a class with errors on extraction ... | 2 weeks ago  |
| requirements-doc.txt | add insider                                                              | last week    |
| requirements.txt     | Docstring parsing (#83)                                                  | last month   |

README.md

## Instructor (openai\_function\_call)

stars 944

forks 72

issues 11 open

license MIT

docs available

Buy Me a Coffee

Donate

Follow @jxnico

Structured extraction in Python, powered by OpenAI's function calling api, designed for simplicity, transparency, and control.

This library is built to interact with openai's function call api from python code, with python structs / objects. It's designed to be intuitive, easy to use, but give great visibility in how we call openai.

### Requirements

This library depends on **Pydantic** and **OpenAI** that's all.

### Installation

### About

openai function calls for humans

[jxnl.github.io/instructor/](https://jxnl.github.io/instructor/)

python openai pydantic-v2

Readme

MIT license

Activity

944 stars

21 watching

72 forks

Report repository

### Releases

15

0.2.7

Latest

2 weeks ago

+ 14 releases

### Packages

No packages published

### Used by

11

+ 3

### Contributors

19

+ 8 contributors

### Deployments

142

github-pages yesterday

Preview

# Prompt engineering

```
from instructor import OpenAISchema
```

```
class HouseFeatures(OpenAISchema):  
    """Correctly extracted house listing features"""
```

```
    is_pet_friendly: bool  
    has_garden: bool  
    has_parking_place: bool  
    is_suitable_for_couple: bool  
    has_balcony: bool
```

```
HouseFeatures.openai_schema
```

```
functions = [  
    {  
        ...  
    }  
]
```

```
prompt = str(house)  
response = openai.ChatCompletion.create(  
    model="gpt-3.5-turbo-0613",  
    deployment_id="gpt-35-turbo-us",  
    functions=[HouseFeatures.openai_schema],  
    function_call={"name": HouseFeatures.openai_schema["name"]},  
    messages=[  
        {"role": "system", "content": "Assistant is a large language model  
designed to extract structured data from text."},  
        {"role": "user", "content": prompt}  
    ],  
)  
house_features = HouseFeatures.from_response(response)  
house_features
```

```
HouseFeatures(is_pet_friendly=False,  
has_garden=False, has_parking_place=True,  
is_suitable_for_couple=True, has_balcony=True)
```

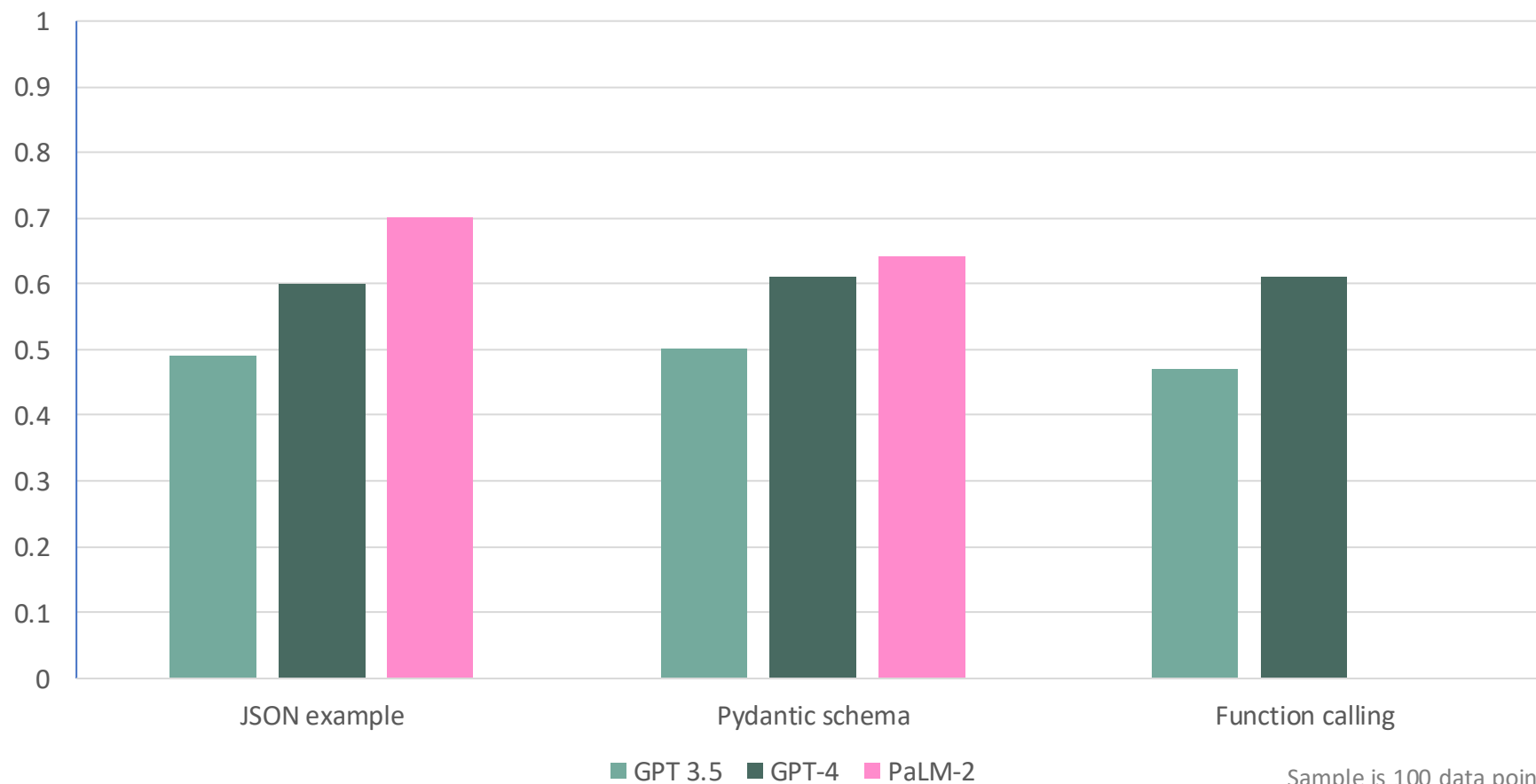




# Evaluating LLM performance

For information extraction

GPT 3.5, GPT 4 and PaLM compared



Now  
what?

We can enrich our original dataset using these new properties

```
{
  'id': 'edfebeac',
  'url': 'https://www.pararius.nl/appartement-te-huur/amsterdam/edfebeac/van-ostadestraat',
  'price': 2150.0,
  'surface_area': 58.0,
  'rent_or_buy': 'huur',
  'city': 'Amsterdam',
  'neighborhood': 'Nieuwe Pijp',
  'interior': 'Gestoffeerd',
  'rooms': 2,
  'is_pet_friendly': HouseFeature(was_extracted=False, quote='', value=None),
  'has_garden': HouseFeature(was_extracted=True, quote='with terrace-garden at the back', value=True),
  'has_parking_place': HouseFeature(was_extracted=False, quote='', value=None),
  'is_suitable_for_couple': HouseFeature(was_extracted=False, quote='', value=None),
  'has_balcony': HouseFeature(was_extracted=False, quote='', value=None)
}
```

# Wrapping it up

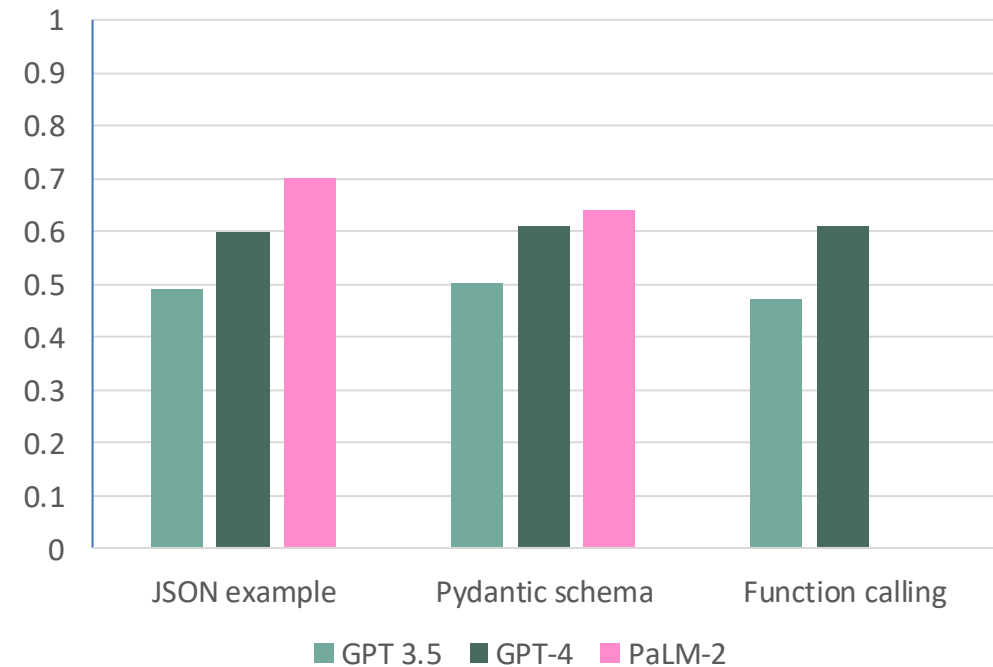
We saw different **methods** of achieving the same thing

## *Methods*

- JSON example
- Pydantic schema
- Function calling

With varying **performances** for different models

GPT 3.5, GPT 4 and PaLM compared



# Using LLM's for Dataset enrichment



Good if:

- Instructions are explicit
- Output format is explicit
- Input data is comprehensive
- Data is valuable enough to justify cost

Do **exploit**:

- Structured output possibilities (JSON, pydantic)
- Function calling (if using GPT)
  - Instructor library

Using LLM's  
for Dataset  
enrichment

What's  
your  
usecase?



Thanks!



Questions?

GitHub repo

