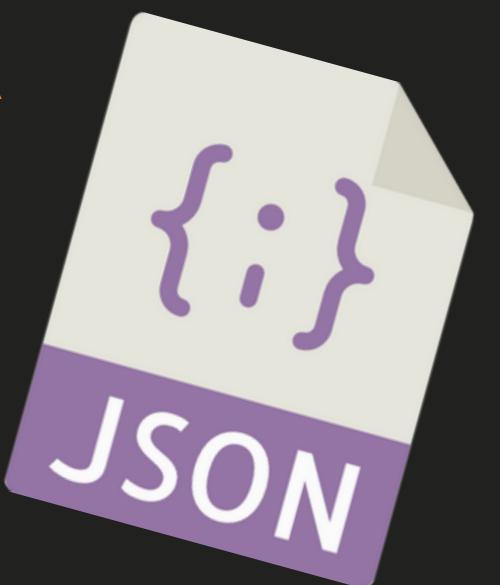


PRODUCT PARSER

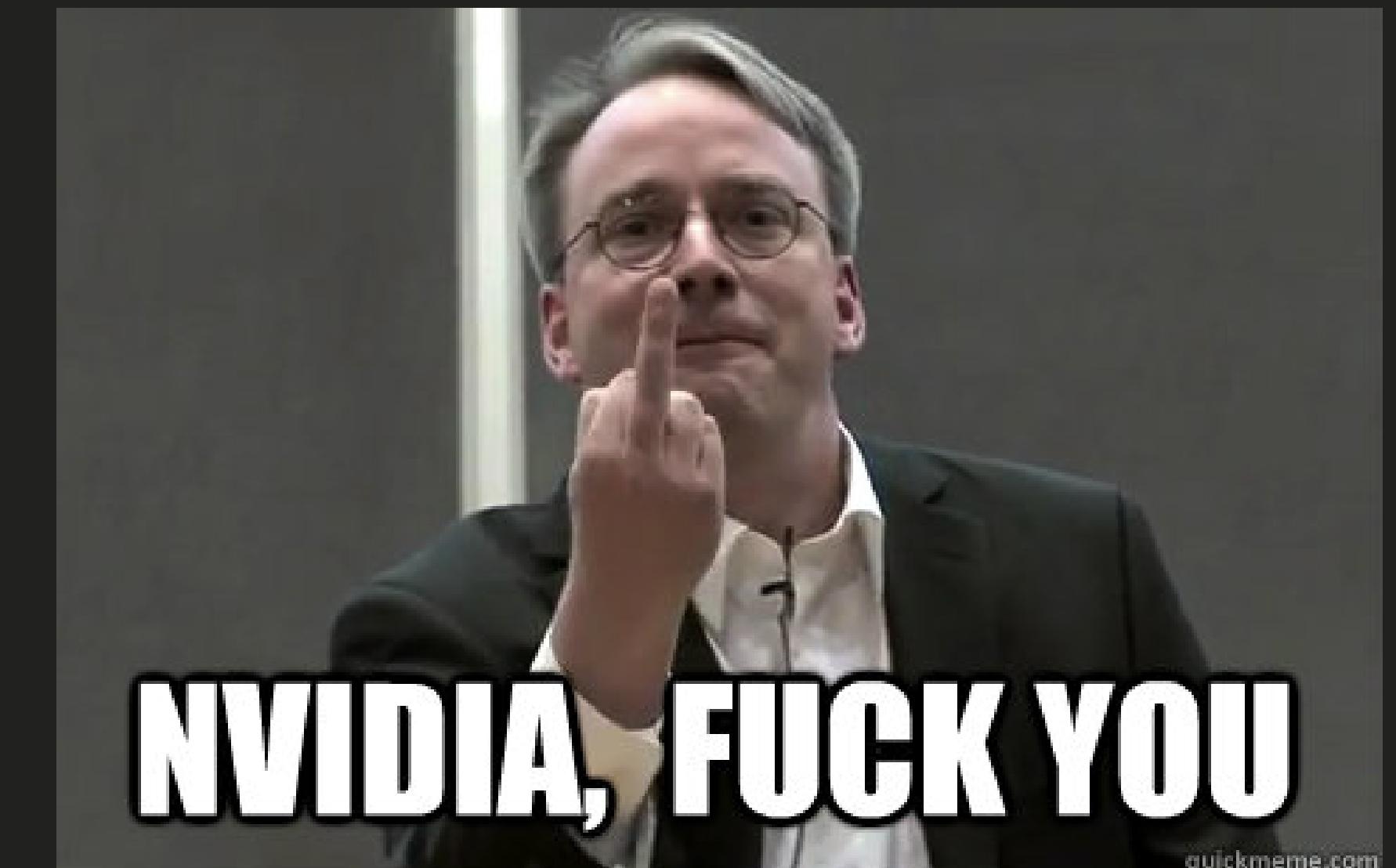
Analizator składniowy opisów produktów ecommerce, z wykorzystaniem ollamy oraz struktury json



CZY KONIECZNYM
BYŁO ZROBIENE
POWERPOINTA?



OH, I USE
ARCH BTW.
NOW MY
COMPTUER
IS DEAD...



NVIDIA, FUCK YOU

CEL



Zamiana opisu słownego na ustrukturyzowaną formę json w oparciu o wcześniej stworzony schemat strukturalny.

- Uproszczenie i przyśpieszenie dodawania nowych ofert w serwisach ecommerce (nie trzeba ręcznie wybierać wartości z dropdownów)
- Dodawanie nowych zmiennych na istniejących serwisach. (chcemy dodać filtr np. koloru, bez konieczności ręcznego labelowania istniejących ofert)
- Scrapery (brak konieczności programowania scrapera pod każdy serwis, wystarczy ekstrakcja całego tekstu na raz)

Input Description

Paste a product description below to extract details.

Description

elektryczne nożyce firmy "Fiskars": narzędzia ogrodowe, ręczne, w kolorze zielonym, ze stali hartowanej za 20zł



JSON Output

Raw extraction result from the API.

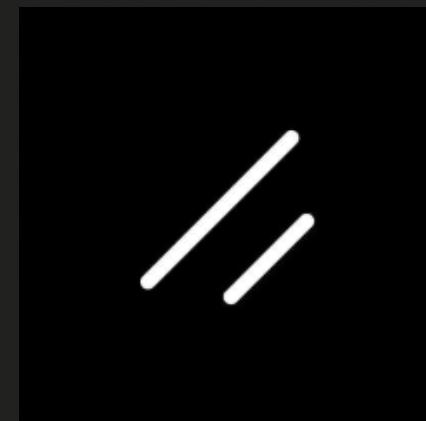
```
{  
    "category": "garden_tools",  
    "subcategory": "hand_tools",  
    "brand": "Fiskars",  
    "color": "green",  
    "material": "hardened steel",  
    "price": "20zł",  
    "product_name": "elektryczne nożyce Fiskars",  
    "features": [],  
    "image_url": "https://finesio.b-cdn.net/product-pim/f/4/d/b/f4dbb033a3f517a9fdf43a6e926f2326488f323f_fiskars_powerg class=public"  
}
```

TECH STACK

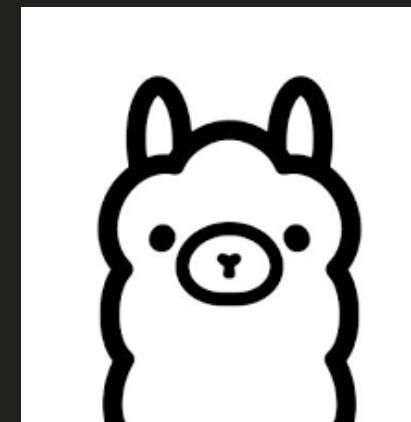
Next.js



ShadCN/ui



Ollama



Fastapi



Mistral-nemo (12b)



DuckDuckGo



FRONTEND PLAYGROUND

Opis wklejony do input fielda



Fastapi endpoint



Przywołanie schematu i przestanie do parsera



Sklejenie system promptu ze schematem i opisem



Endpoint ollamy odbiera prompt i zwraca jsona



Fuzzy matching outputu ze schematem



DuckDuckGo szuka obrazu dla jsona



Fastapi zwraca finalną odpowiedź na frontend



Json i obraz są mapowane na kartę produktu

AI Product Parser

Input Description
Paste a product description below to extract details.

Description

```
elektryczne nożyce firmy "Fiskars": narzędzia ogrodowe, ręczne, w kolorze zielonym, ze stali hartowanej za 20zł
```

Extract Details

JSON Output
Raw extraction result from the API.

```
{
  "category": "garden_tools",
  "subcategory": "hand_tools",
  "brand": "Fiskars",
  "color": "green",
  "material": "hardened steel",
  "price": "20zł",
  "product_name": "elektryczne nożyce Fiskars",
  "features": [],
  "image_url": "https://finesio.b-cdn.net/product-pim/f/4/d/b/f4dbb033a3f517a9fdf43a6e926f2326488f323f_fiskars_powerg class=public"
}
```

Product Card

Product Name: elektryczne nożyce Fiskars **Category:** garden_tools

Image:

Price: 20zł **Color:** Green

Material: hardened steel **Size:** Not specified

Subcategory: hand_tools

SCHEMAT JSON

Na jego podstawie model wie jakich zmiennych szukać w tekście oraz jakie mogą/pominny przybierać wartości. Wartości w schemacie mogą być strict (enumy i fuzzy matching) lub loose (mini prompty)

```
    "features": {
        "type": "array",
        "items": { "type": "string" },
        "description": "A list of key features or highlights of the product"
    },
    "category": {
        "type": "string",
        "description": "The main category of the product"
    },
    "subcategory": {
        "type": "string",
        "description": "The specific subcategory"
    },
    "brand": {
        "type": "string",
        "description": "The brand manufacturer"
    }
},
"inference_rules": [
    "Extract brand names exactly as they appear.",
    "Categorize items based on standard e-commerce categories.",
    "Extract price with currency symbol."
]
```

```
schema.json > ...
1  {
2      "properties": {
3          "category": {
4              "type": "enum",
5              "description": "The main category of the product",
6              "values": [
7                  "appliances",
8                  "arts, crafts & sewing",
9                  "automotive",
10                 "baby products",
11                 "beauty",
12                 "beauty & personal care",
13                 "cell phones & accessories",
14                 "clothing, shoes & jewelry",
15                 "electronics",
16                 "grocery & gourmet food",
17                 "health & household",
18                 "home & kitchen",
19                 "industrial & scientific",
20                 "office products",
21                 "patio, lawn & garden",
22                 "pet supplies",
23                 "sports & outdoors",
24                 "tools & home improvement",
25                 "toys & games",
26                 "video games"
27             ]
28         },
29         "subcategory": {
30             "type": "enum",
31             "description": "The specific subcategory",
32             "values": [
33                 "3d printers",
34                 "3d wall panels",
35                 "accessories",
36             ]
37         }
38     }
39 }
```

KONSTRUKCJA PROMPTU

System prompt składa się ze szczegółowego opisu zadania dla modelu, z miejscami w które wtłaczany jest schemat. User prompt to opis produktu. Oba prompty trafiają do ollamy z opcją wymuszenia jsona.

```
try:
    response = ollama.chat(
        model=MODEL_NAME,
        messages=[
            {
                'role': 'system',
                'content': system_prompt,
            },
            {
                'role': 'user',
                'content': description,
            },
        ],
        format='json',
    )
    content = response['message']['content']
    raw_result = json.loads(content)

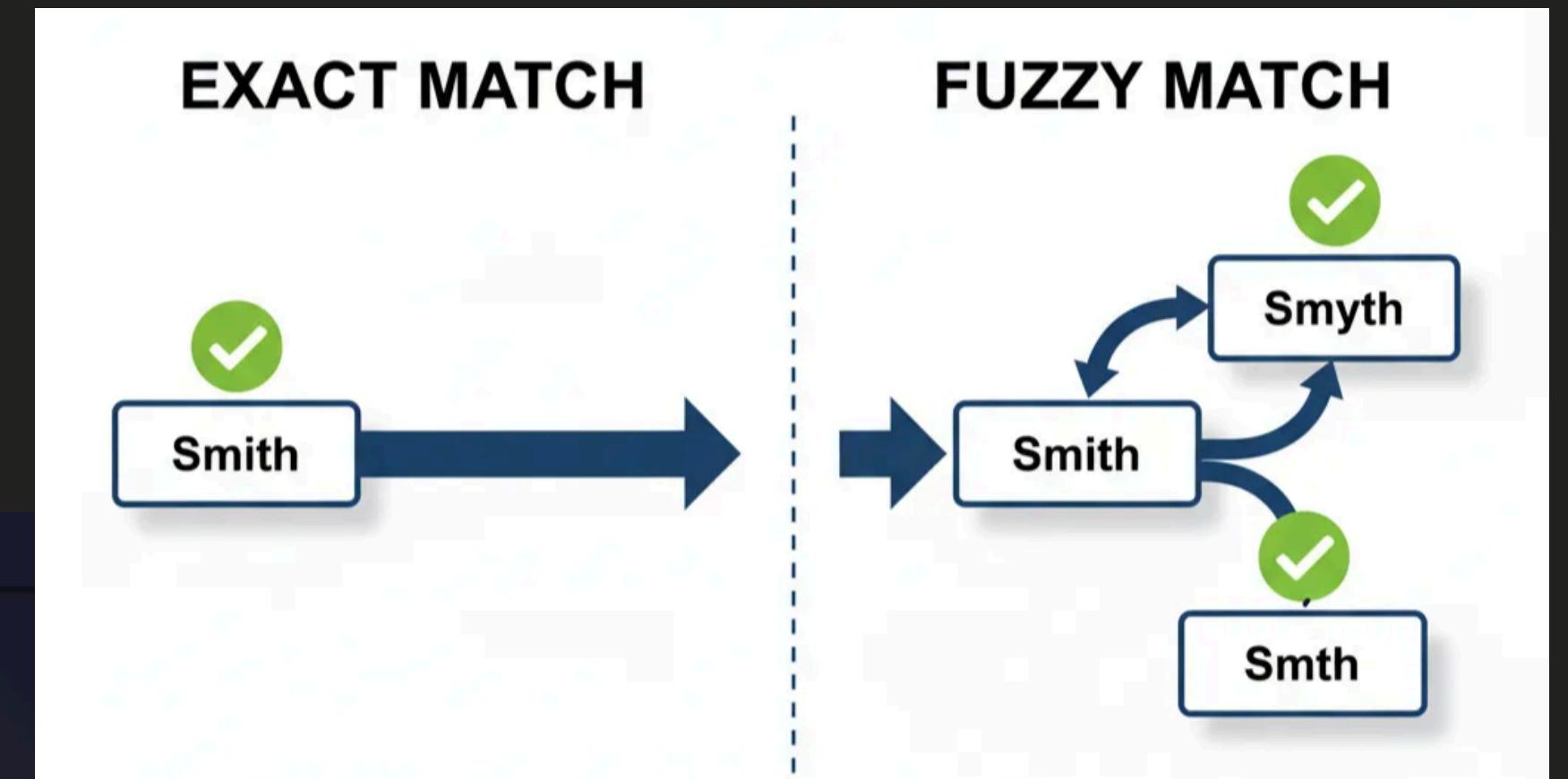
    # Post-process validation
    return validate_and_normalize(raw_result, schema)
```

```
24 def construct_prompt(schema):
25
26     prompt = f"""
27     You are a helpful assistant that parses product descriptions into a JSON structure.
28     You MUST output ONLY valid JSON.
29     You must extract the following properties from the input text.
30
31     Fields to Extract:
32     {fields_str}
33
34     Valid Categories (Choose exactly one for 'category'):
35     {cat_list_str}
36
37     Rules:
38     1. Return JSON only.
39     2. For 'category', you MUST infer and choose the best fit from the 'Valid Categories' list above.
40     3. For ALL OTHER fields (brand, subcategory, etc.), extract the EXACT text found in the description.
41     4. If a property is not found, return null (or empty list for arrays).
42     5. 'features' should be a list of short strings highlighting key product features.
43
44     Inference Rules:
45     {rules_str}
46
47     Example Input: "Bright, big orange and black fedora made with quality polyester. Brand: HatMaster. Di
48
49     Example Output:
50     {{
51         "category": "clothing, shoes & jewelry",
52         "subcategory": "hats",
53         "brand": "HatMaster",
54         "color": "orange",
55         "material": "polyester",
56         "size": "xl",
57         "dimensions": "10x10x5 inches",
58         "weight": "0.5 lbs",
59         "features": ["Waterproof", "sun protection"]
60     }}
61 """
62
63     return prompt
```

SPRAWDZENIE OUTPUTU

Po otrzymaniu odpowiedzi od modelu, json jest porównywany ze schematem. Jeżeli to enum to szukany jest exact match, jeżeli nie jest to enum to szukamy najbardziej podobnego, jeżeli nic nie uda się znaleźć, używamy oryginalnej wartości jaką zwrócił model

```
def validate_and_normalize(result, schema):
    def match_val(v):
        if not v: return None
        v = str(v).lower().strip()
        if v in valid_values:
            return v # Exact match
        # Fuzzy match
        matches = difflib.get_close_matches(v, valid_values, n=1, cutoff=0.6)
        if matches:
            return matches[0]
        return v # Return original value if no match found (soft validation)
```



TEST NA DANYCH ZASTANYCH

Dane produktów na amazonie

- Opis, kategoria, podkategoria, producent, kolor, wymiary
- 1000 rekordów do testów, z nich 50 losowych do walidacji parsera

Skuteczność samej metody działania jest naprawdę wysoka. Głównym ograniczeniem jest brak konkretnych danych w opisie (nie wyodrębnimy z niego np. koloru jeżeli autor nie zawał tej informacji w opisie).

Kolejnym ograniczeniem jest dobry schemat (posiadając realny schemat sklepu, parser dopasowuje enumy 1 do 1, zamiast samemu tworzyć wartości)

Ostatnim ograniczeniem jest mały rozmiar modelu, 12 miliardów parametrów.



Metric	Accuracy
Category	44.0%
Subcategory	22.0%
Brand	62.0%
Color	32.0%
Dimensions Found	26/50

PRZYKŁADOWY OPIS Z ALLEGRO

Stan: Nowy

Sauna ogrodowa QUADRO 240 cm 2QR zestaw do samodzielnego montażu

4,87 ★★★★★ 54 oceny i 36 recenzji | 11 osób kupiło ostatnio



od Oficjalny sklep AZS-ENGINEERING

Firma | [poleca 100%](#)

Megaraty 9 190,00 zł
Najniższa cena oferty z 30 dni przed obniżką

8908,20 zł
445,41 zł x 20 rat [raty zero](#) sprawdź

1 osoba kupiła tę ofertę

Pokaż warianty od innych sprzedających

Kolor/Wzór



Szerokość produktu

220 cm

Długość

180 cm 240 cm 340 cm

Liczba sztuk | oferta limitowana ⓘ

- 1 + z 48 sztuk

Garden Sauna BECKA made of Swedish wood

home_and_garden



N/A

Material

Swedish spruce

Subcategory

outdoor_saunas

Size

2.2m diameter, 2.4m length

Features

- Easy to assemble
- Requires no foundation
- Quick heating
- Portable
- Can be customized during assembly

INNE PRZYKŁADY

Input Description

Paste a product description below to extract details.

Description

Touhou Patchouli Fumo Doll 10x19 for 19.99

Extract Details

Touhou Patchouli Fumo Doll

toys & hobbies

N/A



\$19.99

N/A

Material
N/A

Size
10x19

Subcategory
action figures & dolls

Features

N/A

INNE PRZYKŁADY

Input Description

Paste a product description below to extract details.

Description

nokia 3310, gray, for free

Extract Details

Nokia 3310

electronics & photo

Nokia



free

Gray

Material

Not specified

Subcategory

mobile phones

Size

Not specified

INNE PRZYKŁADY

Input Description

Paste a product description below to extract details.

Description

this offer shows "Smart Drone" by Google made out of wood & plastic, 499, red, pocket size

Extract Details

Smart Drone

Google

electronics



\$499

Red

Material

wood & plastic

Size

pocket size

Subcategory

drones

INNE PRZYKŁADY

Input Description

Paste a product description below to extract details.

Description

Figurka kolekcjonerska Grzegorz Braun z Konfederacji, w zestawie z gaśnicą, brunatny, 271000 rubli, rozmiaru od morza do morza, wykonany z węgla. Unikatowymi funkcjami jest piracka opaska na oko, funkcja mówienia "Szczęść Boże" oraz zgłaszania wniosków formalnych

Extract Details

Unknown Product

toys & hobbies

Grzegorz Braun

271000 rubles



Material
carbon

Size
seawater to seawater

Subcategory
collectible figurines

Features

pirate eyepatch

saying 'Szczęść Boże'

formal proposal function

GITHUB.COM/NELIQ/JSON-OLLAMA-PARSER

```
{  
  "message": {  
    "type": "koniec",  
    "content": "dziękuję za uwagę",  
  },  
}
```