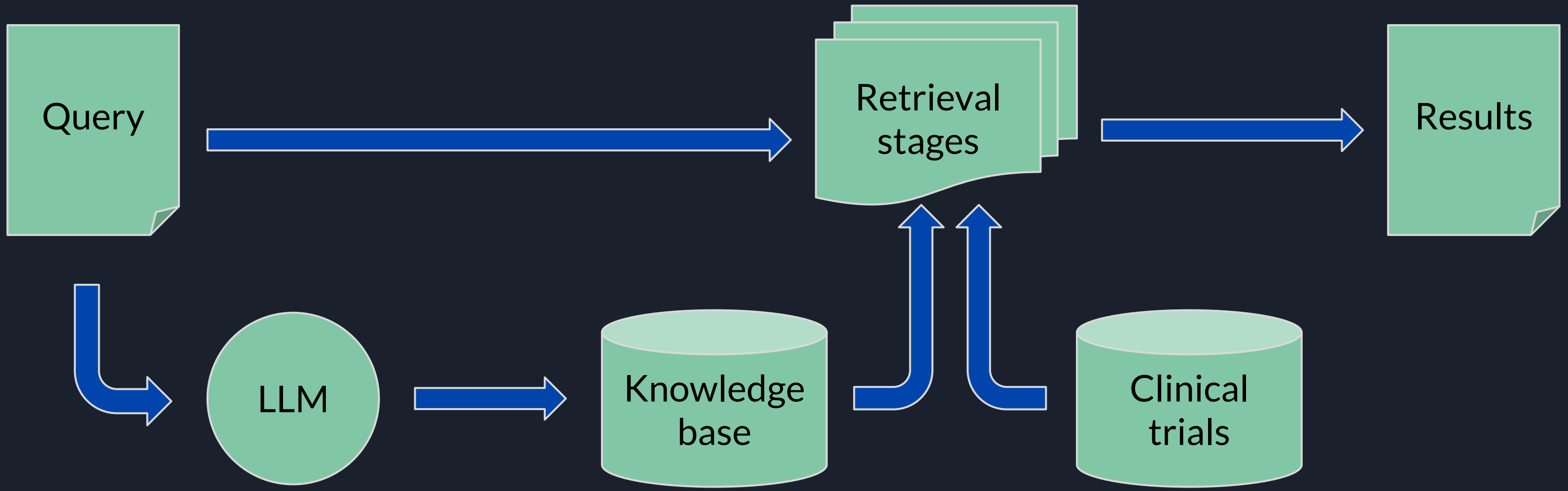
Information retrieval

Group 1

Patrizio Acquadro
Lorenzo Capalbo
Mattia Moro

Problem

Project architecture



Knowledge base

ClinicalTrial

Condition
- Diabetes
- Cancer
- etc.

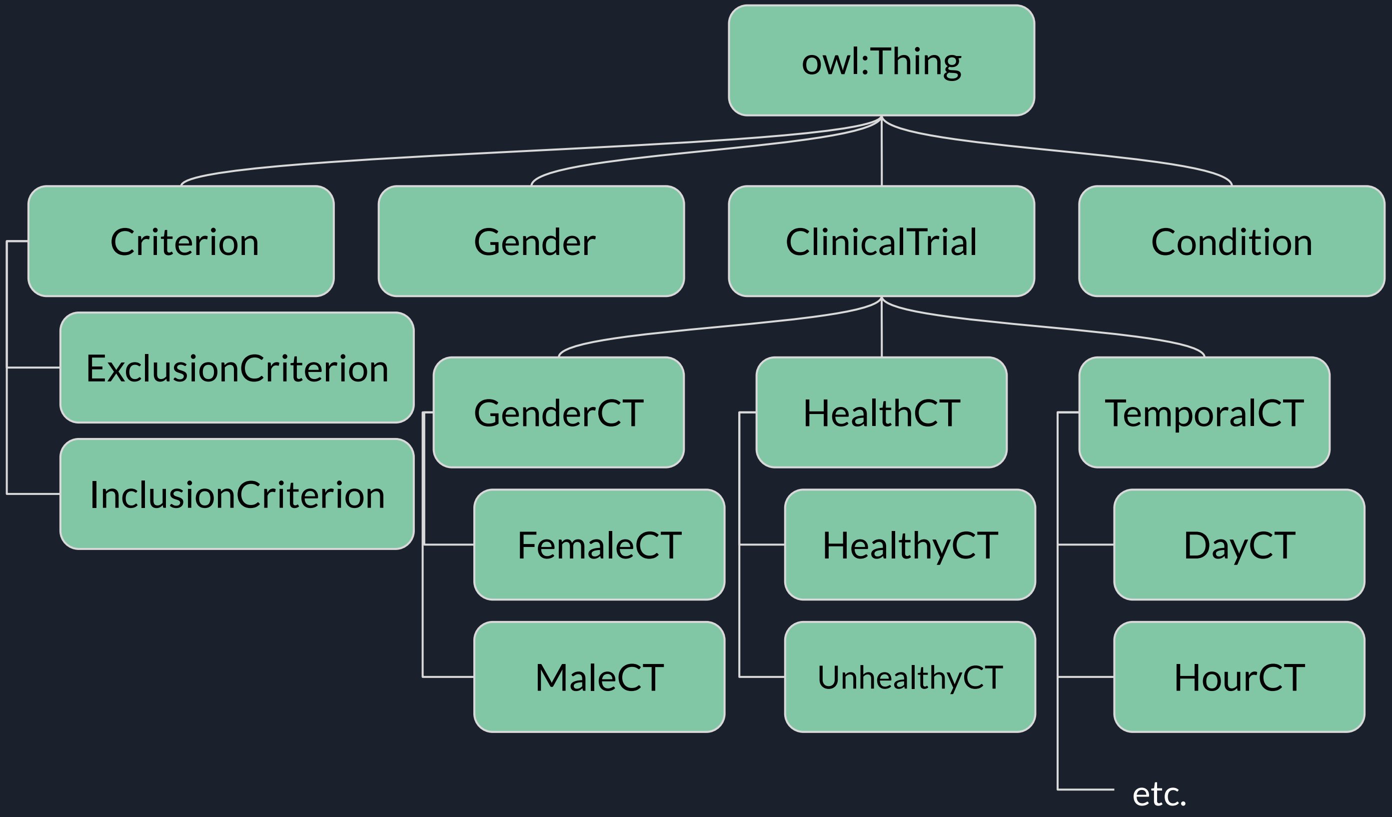
Gender
- Female
- Male

Age
- 21 years
- 18 months
- etc.

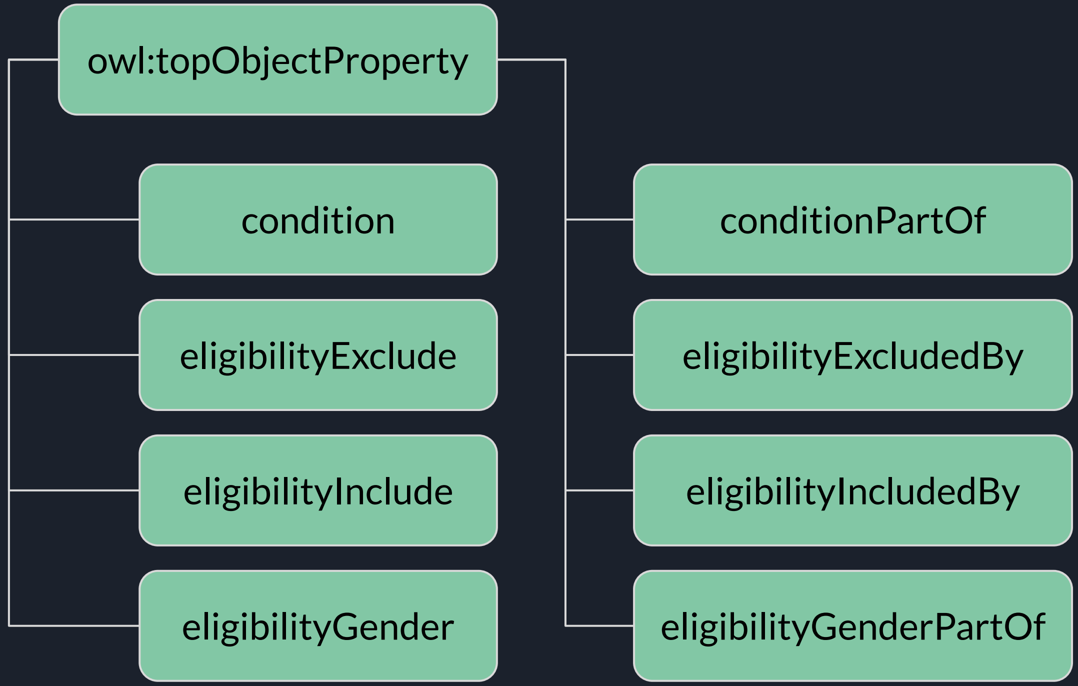
Healthy
- True
- False

Criteria
- Pregnant
- Smoker
- etc.
- *therapy
- *scopy

Ontology: Classes



Ontology: Object properties



Ontology: Data properties

- owl:topDataProperty
  - eligibilityHealthy
  - eligibilityMaximumAge
    - eligibilityMaximumDays
    - eligibilityMaximumHours
    - eligibilityMaximumMinutes
    - eligibilityMaximumMonths
    - eligibilityMaximumWeeks
    - eligibilityMaximumYears
  - eligibilityMinimumAge
    - eligibilityMinimumDays
    - eligibilityMinimumHours
    - eligibilityMinimumMinutes
    - eligibilityMinimumMonths
    - eligibilityMinimumWeeks
    - eligibilityMinimumYears

Knowledge base retrieval

SPARQL

Query Extraction and Inspection

Inspecting and Extracting Queries:

- Parsing the XML file using xml.etree.ElementTree.parse
- Extracting query numbers and texts into a queries dictionary
- Verification of the extraction process by displaying first three entries.

```
[('1',
  'A 19-year-old male came to clinic with some sexual concern.  He recently engaged in a relationship and is worried about the satisfaction of his girlfriend. He has a "baby face" according to his girlfriend\'s statement and he is not as muscular as his classmates.  On physical examination, there is some pubic hair and poorly developed secondary sexual characteristics. He is unable to detect coffee smell during the examination, but the visual acuity is normal. Ultrasound reveals the testes volume of 1-2 ml. The hormonal evaluation showed serum testosterone level of 65 ng/dL with low levels of GnRH.'),
 ('2',
  'A 32-year-old woman comes to the hospital with vaginal spotting.  Her last menstrual period was 10 weeks ago. She has regular menses lasting for 6 days and repeating every 29 days. Medical history is significant for appendectomy and several complicated UTIs. She has multiple male partners, and she is inconsistent with using barrier contraceptives. Vital signs are normal.  Serum β-hCG level is 1800 mIU/mL, and a repeat level after 2 days shows an abnormal rise to 2100 mIU/mL.  Pelvic ultrasound reveals a thin endometrium with no gestational sac in the uterus.'),
 ('3',
  "A 51-year-old man comes to the office complaining of fatigue and some sexual problems including lack of libido. The patient doesn't smoke or use any illicit drug.  Blood pressure is 120/80 mm Hg and pulse is 70/min.  Oxygen saturation is 99% on room air.  BMI is 24 kg/m2.  Skin examination shows increased pigmentation. Genotype testing is consistent with homozygosity for the C282Y mutation. Laboratory study shows transferrin saturation of 55% and serum ferritin of 550 µg/L. He is diagnosed as a case of hemochromatosis.")]
```

Contextual Information Extraction

Contextual Information Extraction:

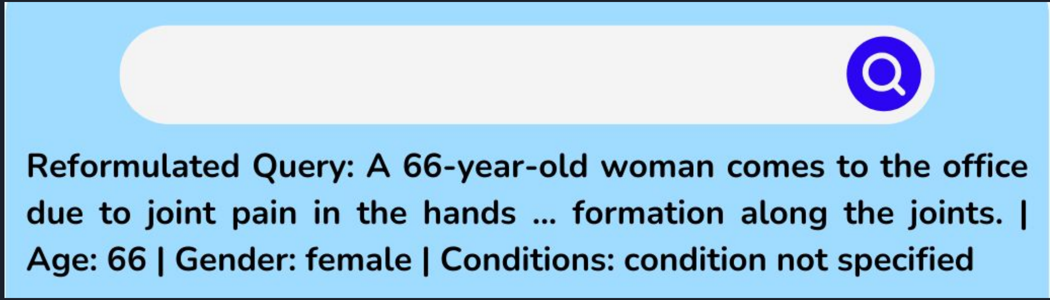
- Utilizing functions like identify_age, identify_gender, and identify_conditions.
- Extracting specific demographic and health data from queries.
- Tailoring queries to include essential contextual details.

```
# Function to identify age from a query
def identify_age(query):
    # Regular expression to find age-related terms
    age_match = re.search(r'\b(\d{1,3})\b(?:\s*|-)*(years|year|months|month|old|year-old)\b', query)
    # Return the age if found, otherwise a default message
    return age_match.group(1) if age_match else "age not specified"
```

Query Reformulation

Reformulating Queries for LLM:

- Integrating extracted data into original queries.
- Creating structured queries with contextual tags.
- Enhanced understanding and accuracy in keyword extraction.



LLM Keyword Extraction

Utilizing LLM for Keyword Extraction:

- Leveraging GPT-3.5 Turbo for detailed information extraction.
- The importance of the prompt
- Error handling and updating to new OpenAI versions for efficiency.
- Generating responses to reformulated queries with reduced error rate.

```python
# Function to create the prompt for OpenAI's GPT-3.5 Turbo model
def create_prompt(query_text):
    return [
        {"role": "system", "content": "You are a helpful medical assistant."},
        {"role": "user", "content": query_prompt_base + ' ' + query_text}
    ]
```

Processing LLM Responses

Processing LLM Responses:

- Loading LLM responses from Excel file.
- Parsing responses into structured dictionaries for each query.
- Transforming raw data into analyzable format for keyword extraction.


```
{0: {'conditions': 'poorly developed secondary sexual characteristics, low levels of GnRH',
  'gender': 'male',
  'healthy': True,
  'age_number': 19,
  'age_type': 'Years',
  'disease': 'None',
  'health_status': None},
```

Summary

Summary of Steps:

1) Initial file uploads and query inspection.
2) Contextual information extraction and query reformulation.
3) Efficient use of LLM for keyword extraction and response processing.
4) Creation of the dictionaries for each query.

Search engine pipeline

The pipeline is composed by:

- a common baseline;
- four possibilities.

The four possibilities are developed on top of the baseline

Baseline

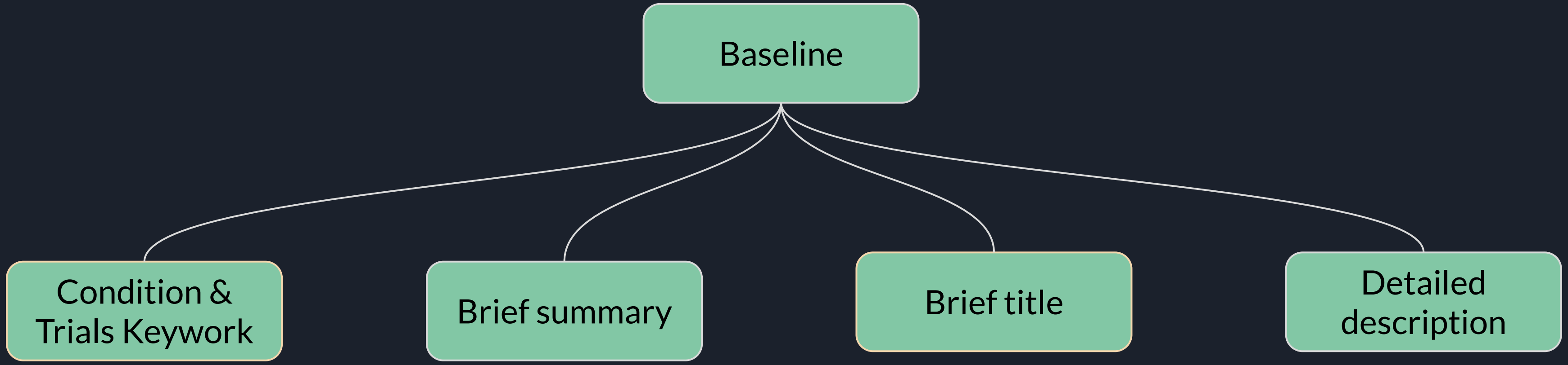
The baseline consists in:

- Two stages of hard filtering;
- a stage of matching;
- a stage of 'not' matching.

It's performed as a series of SPARQL queries to the KG

Retrieval alternatives

The baseline obtained from the KG is then tested on four different columns (or combined columns) from the dataset, to see which are the documents retrieved from each one, and with which score



Reality check

We want to be transparent, admitting that unfortunately our initial proposal couldn’t be tested, mainly due to:

- Limited storage capacity from Google Colab (also from our machines);
- Lack of time

BUT, we were still able to test the four pipelines on a single query, with some adjustments.

Results

With the randomly chosen query, we did run the four retrieval pipelines, with these results

Pipeline 1

	qid	docid	docno	rank	score	query
0	1	1207	NCT02999399	0	11.159350	smoker hemochromatosis
1	1	1356	NCT02019368	1	10.693071	smoker hemochromatosis
2	1	1076	NCT02575885	2	10.074489	smoker hemochromatosis

Pipeline 2

	qid	docid	docno	rank	score	query
0	1	151	NCT00369616	0	10.399582	smoker hemochromatosis
1	1	80	NCT00327808	1	10.177895	smoker hemochromatosis
2	1	978	NCT02141633	2	9.968684	smoker hemochromatosis

Pipeline 3

	qid	docid	docno	rank	score	query
0	1	1200	NCT02599337	0	10.381643	smoker hemochromatosis
1	1	978	NCT02141633	1	9.098075	smoker hemochromatosis
2	1	902	NCT01657487	2	8.901488	smoker hemochromatosis

Pipeline 4

	qid	docid	docno	rank	score	query
0	1	287	NCT00284856	0	11.244928	smoker hemochromatosis
1	1	1773	NCT04711629	1	10.071834	smoker hemochromatosis
2	1	1292	NCT02329353	2	9.733366	smoker hemochromatosis

Conclusions

Our solution presented a hybrid and ambitious approach between cutting-edge methods as LLMs and more traditional (yet innovative) ones as KGs, leveraging them in a search engine with PyTerrier.
Despite the different problems we faced, we were still able through collaboration and patience to
create a final product with acceptable results.

Project organization