MapperGPT: Large Language Models for Linking and Mapping Entities

This manuscript (permalink) was automatically generated from cmungall/gpt-mapping-manuscript@ed03cbf on June 1, 2023.

Authors

- **John Doe**
  XXXX-XXXX-XXXX-XXXX · johndoe · johndoe · @johndoe@mastodon.social
  Department of Something, University of Whatever · Funded by Grant XXXXXXXX

- **Chris Mungall** ✉
  0000-0002-6601-2165 · cmungall
  Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720

✉ — Correspondence possible via GitHub Issues or email to Chris Mungall <cjmungall@lbl.gov>.

Abstract

Mapping...

Introduction

The unification of diverse knowledge bases, lexicons, and ontologies necessitates the interrelation or mapping of entities. An instance of this process can be seen when merging multiple disease terminologies, where it's crucial to ascertain the equivalence of a disease concept across different vocabularies. This challenge is commonly referred to as the ontology matching problem.

The procedure of ontology matching generally employs a combination of automated processes and human intervention. The automated part usually comprises lexical matches of labels, synonyms, or other vocabulary components. This strategy can yield high recall; however, due to lexical ambiguities and homonyms, its precision may suffer depending on the sources. Consequently, a manual filtration step, usually undertaken by a domain expert, is often mandated. Nonetheless, this final filtration phase can prove laborious, as lexical methods typically generate an abundance of potential mappings.

Our team has developed a method that conducts automatic filtration and categorization of prospective mappings using large language models. We approach this as an in-context learning challenge where the model is presented with assorted concept pairs, accompanied by the correct elucidation of their relationship.

We subsequently put this methodology to the test on an anatomy ontology matching task.

The method....

Methods

GPT-based mapping agent

We generate a prompt according to the following template:

```
What is the relationship between the two specified concepts?

Give your answer in the form:

category: <one of: EXACT_MATCH, BROADER_THAN, NARROWER_THAN, RELATED_TO,
DIFFERENT>
confidence: <one of: LOW, HIGH, MEDIUM>
similarities: <semicolon-separated list of similarities>
differences: <semicolon-separated list of differences>

Make use of all provided information, including the concept names,
definitions, and relationships.

Examples:

{{ examples }}

Here are the two concepts:

{{ describe(conceptA) }}
{{ describe(conceptB) }}
```

We use a few-shot learning approach. Examples are provided in-context in the following form:

```
[Concept A]
id: FOO:125
name: wing
def: part of a bird that is flapped to enable flight
is_a: Limb
relationship: part_of Bird
relationship: has_part Feather

[Concept B]
id: BAR:458
name: wing
relationship: part_of Aeroplane

category: DIFFERENT
confidence: HIGH
similarities: function
differences: A is an anatomical part; B is a part of a vehicle
```

For each candidate mapping between concepts A and B, we generate a description of each concept, incorporating key elements: name, synonyms, definition, relationships.

The prompt is then passed to a GPT model, which generates a response. The response is parsed to retrieve the key data model elements: category, confidence, similarities, differences.

Example

As an example, two concepts from the fruitfly and zebrafish anatomy ontologies are candidate matches due to sharing a lexical element (abbreviation). This is a false positive match in reality, as the concept are entirely different.

The two concept descriptions are generated from respective ontologies as follows:

```
[Concept A]
id: FBbt:00001906
name: embryonic/larval Malpighian tubule Type I cell
def: Type I cell of the embryonic/larval Malpighian tubules.
synonyms:  PC ;  embryonic/larval Malpighian tubule Type I cell ;  larval
        Malpighian tubule Type I cell ;  larval Malpighian tubule principal
        cell ;
is_a:  embryonic/larval specialized Malpighian tubule cell ;  Malpighian
        tubule Type I cell ;

[Concept B]
id: ZFA:0000320
name: caudal commissure
def: Diencephalic tract which is located in the vicinity of the dorsal
        diencephalon and mesencephalon and connects the pretectal nuclei.
        From Neuroanatomy of the Zebrafish Brain.
synonyms:  PC ;  caudal commissure ;  posterior commissure ;
is_a:  diencephalic white matter ;
relationship: part of synencephalon
relationship: start stage unknown
relationship: end stage adult
```

The payload for this using gpt-3.5-turbo is:

```
category: DIFFERENT
confidence: HIGH
similarities: NONE
differences: A is a type of cell in the embryonic/larval Malpighian tubules;
        B is a diencephalic tract in the zebrafish brain.
subject: FBbt:00001906
object: ZFA:0000320
```

This is then parsed to a YAML object:

```
predicate: DIFFERENT
confidence: HIGH
similarities:
  - NONE
differences:
  - A is a type of cell in the embryonic/larval Malpighian tubules
  - B is a diencephalic tract in the zebrafish brain.
```

Implementation

We use the OAK library to binding to ontologies.

the overall framework is implemented in OntoGPT.

The input is an SSSOM file. The output is SSSOM with predicate_id filled with predicted value.

```
ontogpt categorize-mappings --model gpt-4 -i foo.sssom.tsv -o bar.sssom.tsv
```

Evaluation

We evaluate against LogMap, which is one of the top-performing mappers in the OAEI.

We convert LogMap results to SSSOM [doi@10.1093/database/baac035]. (Harshad to write)

To generate anatomy test sets, we generated pairwise mappings between species-specific anatomy ontologies, using the Uberon and CL mappings as the gold standard. i.e. if a pair of concepts are transitively linked via Uberon or CL, then they are considered a match.

To evaluate against the gold standard we only considered “best” mappings from each method

LogMap produces a score with each mapping, so we scanned all scores to determine the optimal score threshold in terms of accuracy (F1) (note this gives LogMap an advantage over our method, which does not produce a score).

For the MapperGPT method, we filtered any mapping that is not predicted to be EXACT.

Results

Task

We generated 325 candidate lexical matches between FBbt and ZFA (see methods).

We ran these through MapperGPT.

We also ran LogMap over these two ontologies.

We treat entities linked via Uberon and CL as the Gold Standard.

Core Results

	method	f1	P	R
0	lexmatch	0.34957	0.220217	**0.847222**
1	logmap	0.48913	0.401786	0.625
2	gpt3	0.435484	0.519231	0.375
3	gpt4	**0.651163**	**0.56**	0.777778

LogMap

LogMap returns a score rather than a binary answer - we took the best performing cutoff:

Couldn't load plugin.

img

Discussion

Unlike traditional ontology mapping tools, MapperGPT can provide narrative explanations of why two concepts are predicted to be related in a certain way.

Future Work

MapperGPT is expensive to run with GPT-4. We recommend its use in cases where simpler lexical methods should suffice. We are exploring use of open models that can be executed locally.

We are planning to integrate MapperGPT into our Boomer pipeline to make BoomerGPT, a hybrid neurosymbolic mapping tool that integrates probabilistic inference, description logic reasoning, lexical methods, rule-based methods, and LLMs.

Conclusions

blah

References