MapperGPT: Large Language Models for Linking and Mapping Entities

This manuscript (permalink) was automatically generated from cmungall/gpt-mapping-manuscript@c7c67a1 on June 2, 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

When do two identifiers indicate the same thing? Linking the same or related entities at scale is crucial for knowledge base and ontology integration. For example, if two different disease databases, one with information about disease genes and the other with information about disease symptoms, are to be merged, then it is important to precisely know which disease in one database corresponds to which disease in the other.

A common method to automate ontology matching is to use lexical methods, in particular matching on primary or alternative labels that have been assigned to concepts, sometimes in combination with lexical normalization. These can often provide very high recall, but low precision, due to lexical ambiguity. Examples are provided in 1, including a false match between an aeroplane part and an insect part due to sharing the same name (wing) based on analogous function.

Table 1: Example of entity matching problem

Resource A	Concept A	Resource B	Concept B	Predicted	True Predicate
UK Auto Ontology	Car	Industrial Ontology	Automobile	n/a	`exactMatch`
Train Ontology	Car	Industrial Ontology	RailwayCarriage	n/a	`closeMatch`
Fly ontology	Wing	Industrial ontology	Wing	`exactMatch`	`differentFrom`

An example of this approach is the LOOM algorithm used in the Bioportal ontology resource, which provides very high recall, including NNN mappings across over a thousand vocabularies.

A number of approaches can give higher precision mappings, many of these make use of other relationships or properties in the ontology. The Ontology Alignment Evaluation Initiative (OAEI) provides a yearly evaluation of different methods for ontology matching. One of the top-performing methods in OAEI is the LogMap tool, which makes use of logical axioms in the ontology to assist in mapping.

A number of tools such as LogMap been used to build or link ontologies and knowledge bases. However, these approaches are usually used in conjunction with manual curation of mappings, which can be resource intensive.

Deep learning approaches and in particular Language Models (LMs) have been applied to ontology matching tasks. Some methods make use of embedding distance OntoEmma, e.g [Wang et al., 2018], DeepAlignment [Kolyvakis et al., 2018], VeeAlign [Iyer et al., 2020]. More recently the Truveta Mapper [1] treats matching as a Translation task and involves pre-training on ontology structures.

The most recent development in LMs are so-called Large Language Models (LLMs), exemplified by ChatGPT, which involved billions of parameters and pre-training on instruction-prompting tasks. The resulting models have generalizable abilities to perform a wide range of tasks, including question answering, information extraction. However, one challenge with LLMs is the problem of *hallucination*.

One possibility is using GPT to generate mappings de-novo. However, the problem of hallucination makes this highly unreliable, in particular, due to the propensity for LLMs to hallucinate database or ontology identifiers when these are requested.

We devised an alternative approach called *MapperGPT* that does not use GPT to generate mappings de-novo, but instead works in concert with existing high-recall methods such as LOOM. We use GPT to refine and predict relationships (predicates) as a post-processing step. We use an in-context knowledge-driven semantic approach, in which examples of different mapping categories are provided, and information about the two concepts in a mapping is provided.

We evaluated this on a series of alignment tasks from different domains, including anatomy, developmental biology, and renal diseases. We devised a collection of tasks that are designed to be particularly challenging for lexical methods. We show that when used in combination with high-recall methods such as LOOM or LexMatch, MapperGPT can provide a substantial improvement in accuracy beating SOTA methods such as LogMap.

Our contributions are as follows:

- creation of a series of new matching tasks expressed using the SSSOM standard
- An algorithm and tool MapperGPT that uses GPT to predict relationships between concepts

Methods

Algorithm

Our method MapperGPT takes as input two ontologies *O1* and *O2* and a set of candidate mappings *M*. These mappings are assumed to be have been generated from an existing high-recall method such as LOOM.

```
M' = {}
For m in M:
  prompt = GeneratePrompt(m.a, m.b, O1, O2)
  response = CompletePrompt(prompt, model)
  m' = Parse(response)
  add m' to M'
return M'
```

Prompt generation

The method `GeneratePrompt` generates 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) }}
```

The use of examples makes this 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 `Describe` function will generate a textual description of an ontology or database concept, showing the following properties:

- name
- synonyms
- definition
- parents (superclasses)
- other relationships

Prompt Completion

The prompt is then passed to a GPT model, which generates a response. In principle the method should work with any instruction-based model, either local or remotely accessed via an API. In practice we have only evaluated this against the OpenAI API and the two leading instruction-based models, `gpt-3.5-turbo` and `gpt-4`.

Response Parsing

The response is parsed to retrieve the key data model elements: category, confidence, similarities, differences.

The result object can be exported to SSSOM format.

Example

As an example, two concepts from the fruitfly and zebra fish anatomy ontologies are candidate matches due to sharing a lexical element (the “PC” 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.
```

The consumer may typically only make use of the *predicate* slot, but the list of similarities and differences may prove informative.

Implementation

We use the OAK library to connect to a variety of ontologies in OBO and Bioportal. 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
```

Generation of test sets

To evaluate the method, we created a collection of test sets from biological domains. We chose to devise new test sets as we wanted to base these on up-to-date, precise, validated mappings derived from ontologies such as Mondo, CL, and Uberon.

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. We used the same method for developmental stages.

We also generated a renal disease test set by taking all heritable renal diseases from Mondo, all renal diseases from NCIT, and generating a test set based on validated curated mappings between Mondo and NCIT.

TODO: more test sets

TODO: table showing sizes

Tool evaluation

We evaluate MapperGPT with two models: gpt-3.5-turbo and gpt-4. MapperGPT is capable of providing refined predicates from SKOS but for this task we only take exactMatch as a predicted mapping, and discard all others.

We also evaluated against the OAK lexmatch tool, as a high-recall baseline. Although lexmatch allows for customizable rules, we ran this without any prior knowledge of the domains, and considered any lexical match to be a predicted match.

We selected 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)

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).

Results

MapperGPT with GPT4 improves on state of the art across all tasks

On all tasks combined, summarized in 2, MapperGPT with GPT4 has an accuracy of 0.647, which is a considerable improvement over the SOTA, demonstrating the validity of the approach.

Table 2: Combined results over all tasks

method	f1	P	R
lexmatch	0.012	0.006	**0.865**
logmap	0.538	0.463	0.641
gpt3	0.473	**0.598**	0.391
gpt4	**0.647**	0.594	0.712

LogMap returns a score rather than a binary answer - we took the best performing cutoff. The distribution of F1 scores with different thresholds are show in 1.

Couldn't load plugin.

Figure 1: **LogMap Results**

Anatomy Task Results

We assessed methods against an anatomy ontology matching task containing all vetted mappings between the Fly anatomy ontology (FBbt) and the Zebra fish anatomy ontology (ZFA).

Table 3: Results of *Drosophila* to *Danio rerio* anatomy matching

method	f1	P	R
lexmatch	0.350	0.220	**0.847**
logmap	0.489	0.402	0.625
gpt3	0.435	0.519	0.375
gpt4	**0.651**	**0.560**	0.778

In this task, GPT-4 scored highest in both accuracy and precision.

Couldn't load plugin.

Figure 2: **LogMap Results**

We also assessed Fly to Worm:

Table 4: Results of *Drosophila* to *C elegans* anatomy matching

method	f1	P	R
lexmatch	0.264	0.156	**0.857**
logmap	0.589	0.528	0.667
gpt3	0.345	**0.625**	0.238
gpt4	**0.630**	0.580	0.690

Couldn't load plugin.

Figure 3: **LogMap Results**

Developmental Stage ontology task results

Table 5: Results of human developmental stages (HsapDv) vs mouse developmental stages (MmusDv)

method	f1	P	R
lexmatch	**0.839**	0.929	**0.765**

method	f1	P	R
logmap	0.643	0.818	0.529
gpt3	0.522	**1.000**	0.353
gpt4	0.381	**1.000**	0.235

Couldn't load plugin.

Figure 4: **LogMap Results**

Disease matching task results

We evaluated methods against a disease ontology matching task, which was to match all heritable renal diseases from Mondo to all renal diseases from NCIT.

Table 6: Results of MONDO vs NCIT (renal subset)

method	f1	P	R
lexmatch	0.003	0.002	**1.000**
logmap	0.680	**0.680**	0.680
gpt3	0.679	0.643	0.720
gpt4	**0.759**	0.667	0.880

Couldn't load plugin.

Figure 5: **LogMap Results**

In this task, the previous SOTA achieves slight gain in precision over GPT based methods.

Discussion

Study limitations

The anatomy task is particularly challenging for traditional methods as the curated mappings are quite conservative - e.g

Malpighian tubule <-> renal tubule

Narrative explanations of results

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

We did not perform a qualitative evaluation of the explanations

Limitations of the method

Our best results were achieved using GPT-4. However, at this time, GPT-4 is expensive to run, so we do recommend its use in cases where simpler lexical methods should suffice. We are exploring use of open models that can be executed locally.

Future Work

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

Conclusions

blah

References

1. **Truveta Mapper: A Zero-shot Ontology Alignment Framework**
Mariyam Amir, Murchana Baruah, Mahsa Eslamialishah, Sina Ehsani, Alireza Bahramali, Sadra Naddaf-Sh, Saman Zarandioon
arXiv (2023) https://doi.org/gr934s
DOI: 10.48550/arxiv.2301.09767