MapperGPT: Large Language Models for Linking and Mapping Entities

This manuscript (<u>permalink</u>) was automatically generated from <u>monarch-initiative/gpt-mapping-manuscript@24d851c</u> on September 8, 2023.

Authors

- Nicolas Matentzoglu

Semanticly, Athens, Greece · Funded by NIH NHGRI 7RM1HG010860-02

- Harshad Hegde

Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720

- Hyeongsik Kim
 - **ⓑ** 0000-0002-3002-9838 ⋅ **♀** yy20716

Robert Bosch LLC

- 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

Aligning terminological resources, including ontologies, controlled vocabularies and taxonomies, is a critical part of data integration in many domains such as healthcare, chemistry and biomedical research.

Entity mapping is the process of determining correspondences between entities across these resources, such as gene identifiers, disease concepts or chemical entity identifiers. Many tools have been developed to compute such mappings based on common structural features and lexical information such as labels and synonyms. Lexical approaches in particular often provide very high recall, but low precision, due to lexical ambiguity.

Large Language Models (LLMs), such as the ones employed by ChatGPT, have generalizable abilities to perform a wide range of tasks, including question answering and information extraction.

Here we present *MapperGPT*, an approach based on LLMs to refine and predict mapping relationships as a post-processing step, that works in concert with existing high-recall methods that are based on lexical and structural heuristics.

We evaluated *MapperGPT* 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 *MapperGPT* can provide a substantial improvement in accuracy beating state of the art methods such as LogMap.

Introduction

Tackling global challenges, including rare disease and climate change, requires the integration of a large number of disparate data sources. Due to the decentralised nature of data standardisation, where different data providers inevitably employ different controlled vocabularies and ontologies to standardise their data, it becomes crucial to integrate such "semantic spaces" (i.e. data spaces that are described using divergent sets of ontologies).

Linking entities across often huge semantic spaces at scale is crucial. For example, integrating genetic associations for disease provided by a disease data resource such as the Online Mendelian Inheritance in Man (OMIM) with the phenotypic associations to the same disease provided by Orphanet requires mapping different disease identifiers that refer to the exact same real-world disease concept. Manually mapping thousands of disease concepts for two semantic spaces is potentially doable manually, but in the real world, dozens of resources providing information about the same data type (disease, genes, environment, organisms) need to be integrated, which makes a purely manual approach infeasible.

Semantic entity matching is the process of associating a term or identifier A in one semantic space to one or more terms or identifiers B in another, where A and B refer to the same or related real-world concepts. A common method to automate semantic entity matching is to use lexical methods, in particular matching on primary or alternative labels (synonyms) that have been assigned to concepts, sometimes in combination with lexical normalization. These methods 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	differentFro m

An example of this approach is the LOOM algorithm used in the Bioportal ontology resource [pubmed:20351849?], which provides very high recall mappings across over a thousand ontologies and other controlled 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.

Deep learning approaches and in particular Language Models (LMs) have been applied to ontology matching tasks. Some methods make use of embedding distance OntoEmma [1], DeepAlignment [2], VeeAlign [3]. More recently the Truveta Mapper [4] 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*. An hallucination describes a situation where an Al model "fabricates" information that does not directly correspond to the provided input.

Given their performance on any tasks related to the understanding and generation natural language, it seems obvious to employ LLMs directly as a powerful, scalable alternative to current state-of-the-art (SOTA) methods for entity matching. One possibility is using LLMs like the ones employed by ChatGPT to generate mappings de-novo. However, the problem of hallucination makes this 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 models to generate mappings de-novo, but instead works in concert with existing high-recall methods such as LOOM [pubmed:20351849?]. We use a GPT model to refine and predict relationships (predicates) as a post-processing step, essentially for the purpose of isolating and removing false positive mappings. We use an in-context knowledge-driven semantic approach, in which examples of different mapping categories and information about the two concepts in a mapping is provided to the model to determine an appropriate mapping relationship. We use the Simple Standard for Sharing Ontological Mapping (SSSOM) [5] for sharing and comparing entity mappings across systems.

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 OAK LexMatch [6], MapperGPT can provide a substantial improvement in accuracy beating SOTA methods such as LogMap.

Our contributions are as follows:

- The creation of a series of new matching tasks expressed using the SSSOM standard
- An algorithm and tool, *MapperGPT*, that uses a GPT model 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: F00:125
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 Drosophila (fruitfly) and zebrafish anatomy ontologies [8] 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 response for this using gpt-3.5-turbo (August 2023) is:

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 Ontology Access Kit (OAK) library [6] to connect to a variety of ontologies in the Open Biological and Biomedical Ontology (OBO) Library [9] and Bioportal [pubmed:19483092?]. OAK generally enables accessing ontologies, but we also make use of its ability to extract subsets of ontologies, perform lexical matching using OAK LexMatch, extract mappings from ontologies and ontology portals such as Bioportal, and add labels to mapping tables which typically only include the mapped identifiers, for better readability. We make use of ROBOT [10] for converting between different ontology formats. The overall mapping framework is implemented in OntoGPT [11] (https://github.com/monarch-initiative/ontogpt) in a method called categorize-mappings, where the input is a SSSOM mapping file (usually generated by a lexical matching tool) and the output is a SSSOM mapping file with predicate_id filled with predicted value. Example:

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

The entire pipeline is implemented as a fully reproducible Makefile (https://github.com/cmungall/gpt-mapping-manuscript/blob/main/Makefile).

Generation of test sets

To evaluate the method, we created a collection of test sets from the biological and biomedical 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 [12], Cell Ontology (CL) [13], and the Uberon Anatomy Ontology [14].

To generate anatomy test sets, we generated pairwise mappings between species-specific anatomy ontologies, using the Uberon and CL mappings as the gold standard. If a pair of concepts are transitively linked via Uberon or CL, then they are considered a match. For example, UBERON:0000924 (ectoderm) is mapped to FBbt:00000111 (ectoderm (fruitfly)) and ZFA:0000016 (ectoderm (zebrafish)), so we assume that FBbt:00000111 is an exact match to ZFA:0000016. We used the same method for linking species-specifc developmental stage terms.

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.

Table: Breakdown of the existing test sets. {#tbl:main_results_testsetsize}

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

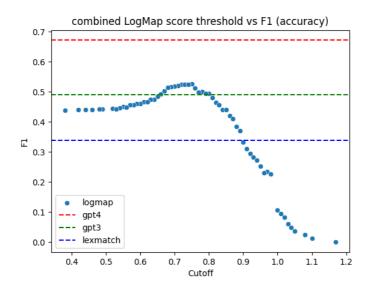
MapperGPT with GPT4 improves on state of the art across all tasks

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	0.340	0.210	0.881
logmap	0.527	0.458	0.619
gpt3	0.490	0.500	0.481
gpt4	0.672	0.601	0.762

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.



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	Р	R
lexmatch	0.349	0.219	0.847
logmap	0.486	0.404	0.611
gpt3	0.511	0.557	0.472
gpt4	0.644	0.543	0.792

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

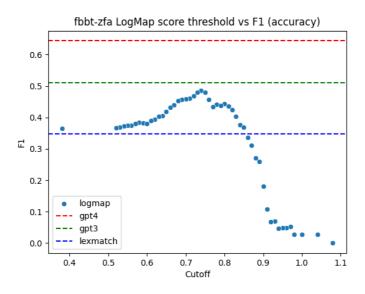


Figure 2: LogMap Results

We also assessed Fly to Worm:

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

method	f1	Р	R
lexmatch	0.257	0.152	0.854
logmap	0.520	0.441	0.634
gpt3	0.427	0.471	0.390
gpt4	0.660	0.585	0.756

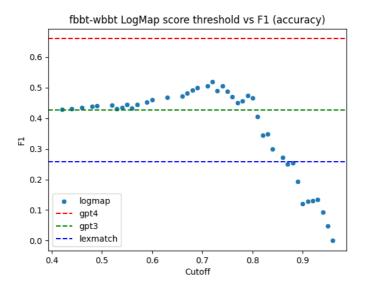


Figure 3: LogMap Results

Developmental Stage ontology task results

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

method	f1	P	R
lexmatch	0.606	0.455	0.909
logmap	0.531	0.405	0.773
gpt3	0.556	0.714	0.455
gpt4	0.647	0.917	0.500

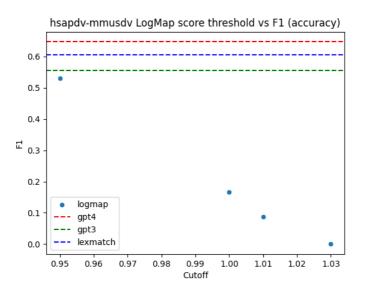


Figure 4: LogMap Results

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	Р	R
lexmatch	0.352	0.214	1.000
logmap	0.721	0.611	0.880
gpt3	0.486	0.378	0.680
gpt4	0.793	0.697	0.920

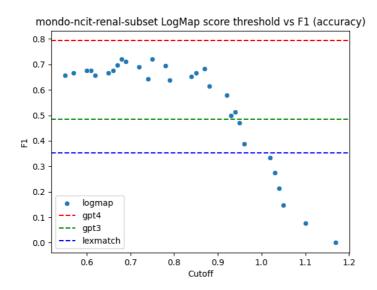


Figure 5: LogMap Results

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. Ontology Alignment in the Biomedical Domain Using Entity Definitions and Context

Lucy Lu Wang, Chandra Bhagavatula, Mark Neumann, Kyle Lo, Chris Wilhelm, Waleed Ammar *arXiv* (2018) https://doi.org/gspf3t

DOI: 10.48550/arxiv.1806.07976

2. DeepAlignment: Unsupervised Ontology Matching with Refined Word Vectors

Prodromos Kolyvakis, Alexandros Kalousis, Dimitris Kiritsis *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018) https://doi.org/gspf3s

DOI: 10.18653/v1/n18-1072

3. Multifaceted Context Representation using Dual Attention for Ontology Alignment

Vivek Iyer, Arvind Agarwal, Harshit Kumar *arXiv* (2020) https://doi.org/gspf3v

DOI: 10.48550/arxiv.2010.11721

4. 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

5. A Simple Standard for Sharing Ontological Mappings (SSSOM)

Nicolas Matentzoglu, James P Balhoff, Susan M Bello, Chris Bizon, Matthew Brush, Tiffany J Callahan, Christopher G Chute, William D Duncan, Chris T Evelo, Davera Gabriel, ... Christopher J Mungall

Database (2022-01-01) https://doi.org/gspf3p

DOI: <u>10.1093/database/baac035</u> · PMID: <u>35616100</u> · PMCID: <u>PMC9216545</u>

6. INCATools/ontology-access-kit: v0.5.19-rc1

Chris Mungall, Harshad, Patrick Kalita, Charles Tapley Hoyt, Sujay Patil, Marcin P Joachimiak, Joe Flack, David Linke, Nomi Harris, Harry Caufield, ... Tiago Lubiana Zenodo (2023-09-02) https://doi.org/gspf3x

DOI: 10.5281/zenodo.8310471

7. The Drosophila anatomy ontology

Marta Costa, Simon Reeve, Gary Grumbling, David Osumi-Sutherland Journal of Biomedical Semantics (2013-10-18) https://doi.org/gspf3q
DOI: 10.1186/2041-1480-4-32 • PMID: 24139062 • PMCID: PMC4015547

8. The zebrafish anatomy and stage ontologies: representing the anatomy and development of Danio rerio

Ceri E Van Slyke, Yvonne M Bradford, Monte Westerfield, Melissa A Haendel Journal of Biomedical Semantics (2014) https://doi.org/gspf3r
DOI: 10.1186/2041-1480-5-12 · PMID: 24568621 · PMCID: PMCID: PMC3944782

9. OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies

Rebecca Jackson, Nicolas Matentzoglu, James A Overton, Randi Vita, James P Balhoff, Pier Luigi Buttigieg, Seth Carbon, Melanie Courtot, Alexander D Diehl, Damion M Dooley, ... Bjoern Peters *Database* (2021-10-01) https://doi.org/gspf3n

DOI: 10.1093/database/baab069 · PMID: 34697637 · PMCID: PMC8546234

10. **ROBOT: A Tool for Automating Ontology Workflows**

Rebecca C Jackson, James P Balhoff, Eric Douglass, Nomi L Harris, Christopher J Mungall, James A Overton

BMC Bioinformatics (2019-07-29) https://doi.org/ggkjnc

DOI: 10.1186/s12859-019-3002-3 · PMID: 31357927 · PMCID: PMC6664714

11. OntoGPT

JHarry Caufield, Harshad Hegde, Vincent Emonet, Nomi L Harris, Marcin P Joachimiak, Nicolas Matentzoglu, HyeongSik Kim, Sierra AT Moxon, Justin T Reese, Melissa A Haendel, ... Christopher J Mungall

Zenodo (2023-08-24) https://doi.org/gspf3w

DOI: 10.5281/zenodo.8278168

12. Mondo: Unifying diseases for the world, by the world

Nicole A Vasilevsky, Nicolas A Matentzoglu, Sabrina Toro, Joseph E Flack IV, Harshad Hegde, Deepak R Unni, Gioconda F Alyea, Joanna S Amberger, Larry Babb, James P Balhoff, ... Melissa A Haendel

Cold Spring Harbor Laboratory (2022-04-16) https://doi.org/ggx27c

DOI: <u>10.1101/2022.04.13.22273750</u>

13. The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability

Alexander D Diehl, Terrence F Meehan, Yvonne M Bradford, Matthew H Brush, Wasila M Dahdul, David S Dougall, Yongqun He, David Osumi-Sutherland, Alan Ruttenberg, Sirarat Sarntivijai, ... Christopher J Mungall

Journal of Biomedical Semantics (2016-07-04) https://doi.org/gg99b9

DOI: <u>10.1186/s13326-016-0088-7</u> · PMID: <u>27377652</u> · PMCID: <u>PMC4932724</u>

14. Uberon, an integrative multi-species anatomy ontology

Christopher J Mungall, Carlo Torniai, Georgios V Gkoutos, Suzanna E Lewis, Melissa A Haendel *Genome Biology* (2012) https://doi.org/fxx6gr

DOI: <u>10.1186/gb-2012-13-1-r5</u> · PMID: <u>22293552</u> · PMCID: <u>PMC3334586</u>