













Data and text mining

Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES): a method for populating knowledge bases using zero-shot learning

J. Harry Caufield ^{1,*}, Harshad Hegde ¹, Vincent Emonet ², Nomi L. Harris ¹,
Marcin P. Joachimiak ¹, Nicolas Matentzoglou ³, HyeonSik Kim ⁴, Sierra Moxon ¹,
Justin T. Reese ¹, Melissa A. Haendel ⁵, Peter N. Robinson ⁶, Christopher J. Mungall ¹

¹Biosystems Data Science, Division of Environmental Genomics and Systems Biology, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States

²Institute of Data Science, Faculty of Science and Engineering, Maastricht University, 6200 MD Maastricht, The Netherlands

³Semanticly, Athens, Greece

⁴Robert Bosch LLC, Sunnyvale, CA 94085, United States

⁵Department of Biomedical Informatics, University of Colorado, Anschutz Medical Campus, Aurora, CO 80217, United States

⁶Berlin Institute of Health at Charité, 10178 Berlin, Germany

*Corresponding author. Biosystems Data Science, Division of Environmental Genomics and Systems Biology, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mailstop 977-0257, Berkeley, CA 94720, United States. E-mail: jhc@lbl.gov (J.H.C.)

Associate Editor: Jonathan Wren

Abstract

Motivation: Creating knowledge bases and ontologies is a time consuming task that relies on manual curation. AI/NLP approaches can assist expert curators in populating these knowledge bases, but current approaches rely on extensive training data, and are not able to populate arbitrarily complex nested knowledge schemas.

Results: Here we present Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES), a Knowledge Extraction approach that relies on the ability of Large Language Models (LLMs) to perform zero-shot learning and general-purpose query answering from flexible prompts and return information conforming to a specified schema. Given a detailed, user-defined knowledge schema and an input text, SPIRES recursively performs prompt interrogation against an LLM to obtain a set of responses matching the provided schema. SPIRES uses existing ontologies and vocabularies to provide identifiers for matched elements. We present examples of applying SPIRES in different domains, including extraction of food recipes, multi-species cellular signaling pathways, disease treatments, multi-step drug mechanisms, and chemical to disease relationships. Current SPIRES accuracy is comparable to the mid-range of existing Relation Extraction methods, but greatly surpasses an LLM's native capability of grounding entities with unique identifiers. SPIRES has the advantage of easy customization, flexibility, and, crucially, the ability to perform new tasks in the absence of any new training data. This method supports a general strategy of leveraging the language interpreting capabilities of LLMs to assemble knowledge bases, assisting manual knowledge curation and acquisition while supporting validation with publicly-available databases and ontologies external to the LLM.

Availability and implementation: SPIRES is available as part of the open source OntoGPT package: <https://github.com/monarch-initiative/ontogpt>.

1 Introduction

Knowledge Bases and ontologies (here collectively referred to as KBs) encode domain knowledge in a structure that is amenable to precise querying and reasoning. General purpose KBs such as Wikidata (Vrandečić 2014) contain broad contextual knowledge, and are used for a wide variety of tasks, such as integrative analyses of otherwise disconnected data and enrichment of web applications (e.g. a recipe website may want to dynamically query Wikidata to retrieve information about ingredients or country of origin). In the life sciences, KBs such as the Gene Ontology (GO) (The Gene Ontology Consortium 2019) and the Reactome biological pathway KB (Fabregat *et al.* 2018) contain extensive curated knowledge detailing cellular mechanisms that involve

interacting gene products and molecules. These domain-specific KBs are used for tasks such as interpreting high-throughput experimental data. All KBs, whether general purpose or domain-specific, owe their existence to curation, often a concerted effort by human experts.

However, the vast majority of human knowledge is communicated via natural language, with scientific knowledge communicated textually in journal abstracts and articles, which has historically been largely opaque to machines. The latest Natural Language Processing (NLP) techniques making use of Large Language Models (LLMs) have shown great promise in interpreting highly technical language, as demonstrated by their performance on question-answering benchmarks (Ateia and Kruschwitz 2023).

Received: 4 May 2023; Revised: 16 December 2023; Editorial Decision: 9 February 2024; Accepted: 20 February 2024

© The Author(s) 2024. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

These techniques have known limitations, such as being prone to hallucinations (Ji et al. 2023) (i.e. generating incorrect statements) and insensitivity to negations (Ettinger 2020). Applications such as clinical decision support require precision and reliability not yet demonstrated by LMs of any size, though recent demonstrations offer promise (Wang et al. 2020, Khambete et al. 2021, Luo et al. 2022, Wachter and Brynjolfsson 2023).

If instead of passing the unfiltered results of LLM queries to users, we use LLMs to build KBs using NLP at the time of KB construction, then we can assist manual knowledge curation and acquisition while validating facts prior to insertion into the KB. NLP can assist KB construction at multiple stages. Literature triage aids selection of relevant texts to curate; Named Entity Recognition (NER) can identify textual spans mentioning relevant things or concepts such as genes or ingredients; grounding maps these spans to persistent identifiers in databases or ontologies; Relation Extraction (RE) connects named entities via predicates such as ‘causes’ into simple triple statements. Deep Learning methods such as autoregressive LMs (Vaswani et al. 2017) have made considerable gains in all these areas. The first generation of these methods relied heavily on task-specific training data, but the latest generation of LLMs such as GPT-3 and GPT-4 are able to generalize and perform zero-shot or few-shot learning on these tasks by reframing them as prompt-completion tasks (Brown et al. 2020).

Most KBs are built upon detailed knowledge schemas which prove challenging to populate. Schemas describe the forms in which data should be structured within a domain. For example, a food recipe KB may break a recipe down into a sequence of dependent steps, where each step is a complex knowledge structure involving an action, utensils, and quantified inputs and outputs. Inputs and outputs might be a tuple of a food type plus a state (e.g. cooked) (Fig. 1). Ontologies such as FOODON (Wang et al. 2020) may be used to provide identifiers for any named entities. Similarly, a biological pathway database might break down a cellular program into subprocesses and further into individual steps, each step involving actions, subcellular locations, and inputs and outputs with activation states and stoichiometry. Adapting existing pipelines to custom KB schemas requires considerable engineering.

A schema provides a structure for data. For example, the recipe schema in Fig. 1 could be used in a recipe database, with each record instantiating the recipe class, with additional linked records instantiating contained classes, e.g. individual ingredients or steps. Figure 2 shows an example of an instantiated schema class, rendered using YAML (<https://yaml.org/spec/1.2.2/>) syntax.

Here we present Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES), an automated approach for population of custom schemas and ontology models. The objective of SPIRES is to generate an instance (i.e. an object) from a text, where that instance has a collection of attribute-value associations. Each value is either a primitive (e.g. string, number, or identifier) or another inlined instance (Fig. 2). SPIRES integrates the flexibility of LLMs with the reliability of publicly-available databases and ontologies (Fig. 3). This strategy allows SPIRES to fill out schemas with linked data while bypassing a need for training examples. A major advantage of SPIRES over more traditional RE is its ability to populate schemas that exhibit nesting, in which

complex classes may have attributes whose ranges are themselves complex classes. SPIRES also makes use of a flexible grounding approach that can leverage over a thousand ontologies in the OntoPortal Alliance (Graybeal et al. 2019), as well as biomedical lexical grounders such as Gilda (Gyori et al. 2022) and OGER (Furrer et al. 2019). This grounding method offers far more consistent mapping to unique identifiers than hallucination-prone LLM querying alone.

2 System and methods

In SPIRES, A knowledge *schema* is a structure for constraining the shape of instances for a given domain. A schema is a collection of *classes* or templates, each of which can be instantiated by instances. Each class has a collection of *attribute constraints*, which control the attribute-value pairs that can be associated with each instance. The *range* of an attribute specifies the allowed value or values. A range can be either (i) a primitive type such as a string or number; (ii) a class; or (iii) an *enumeration* of permissible value tokens (e.g. an enumeration of days of the week may include ‘Monday’, ‘Tuesday’, and so on). Attributes also have *cardinality*, specifying the minimum and maximum number of values each instance can take. In addition, each schema element can have arbitrary metadata associated with it.

Formally, a schema S consists of n classes:

$$\text{Classes}(S) = \{c_1, \dots, c_n\} \quad (1)$$

Classes correspond to the kinds of entities present in a database (e.g. in a recipe database, this would include recipes, as well as ingredients and steps; see example in Fig. 1).

Each class c_i has an ordered list of attributes:

$$\text{Attributes}(c_i) = \{c_i a_1, \dots, c_i a_m\} \quad (2)$$

Instances of c_i may have *values* specified for each of these attributes. An attribute a can have associated properties:

- $\text{Name}(a)$: the name of the attribute; e.g. ‘summary’ or ‘steps’.
- $\text{Multivalued}(a) = \{\text{True}, \text{False}\}$, indicating whether the value of a is a list, or single-valued. A recipe might have a single-valued attribute for the name of the recipe, and a multivalued attribute for the steps.
- $\text{Identifier}(a) = \{\text{True}, \text{False}\}$, indicating whether a is a persistent identifier for instances, such as the FOODON identifiers in Fig. 2.
- $\text{Prompt}(a)$ = string, which is a user-specified custom prompt for that attribute.
- $\text{Range}(a)$: the allowable values for this attribute; this can be a class c in S , or a primitive type such as string or number, or a value set (see below). In Fig. 1, the range of the *ingredients* attribute is Ingredient, and the range of the *id* attribute is a string.
- $\text{ValueSets}(c)$: a list of atomic values from which values of a can be drawn, where a value set is either an extensional list (fixed/static) or intensional (specified by ontology query). For example, a value set for a food element in an ingredient may be drawn from the food branch of the Food Ontology.

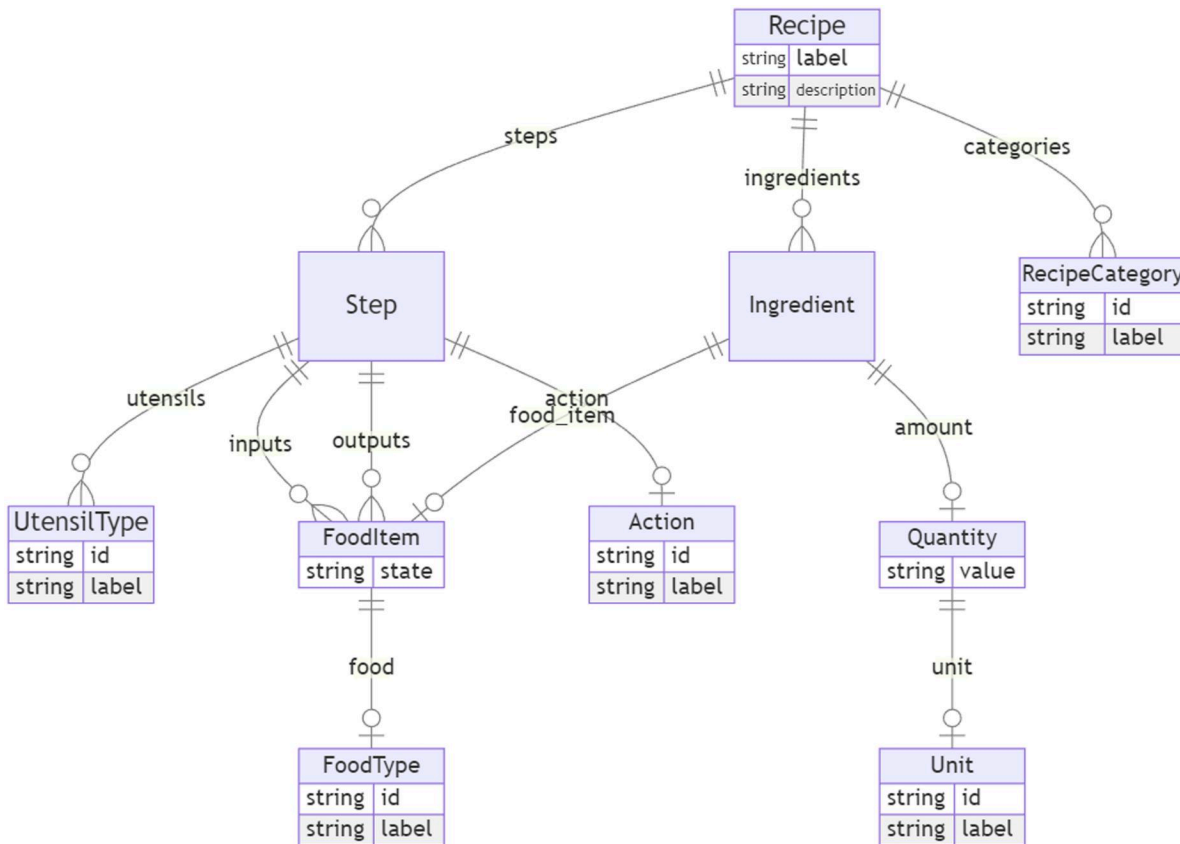


Figure 1. Example schema. Boxes denote classes and arrows denote attributes whose range are classes (compound attributes). Crows feet above boxes denote multivalued attributes. Attributes whose ranges are primitives or value sets are shown within each box. Here, the top level container class ‘Recipe’ is composed of a label, description, categories, steps, and ingredients. Steps and ingredients are further decomposed into food items, quantities, etc.

- $Inlined(a) = \{True, False\}$, indicating, when the range is a class, if the object should be nested/embedded, or passed by reference.

In addition, a class c can include a set of identifier constraints:

$$IDSpaces(c_i) = \{prefix_i, \dots, prefix\} \quad (3)$$

The constraint set is a list of strings that are the allowable prefixes that the identifier can take, e.g. ‘WIKIDATA’, ‘MESH’, ‘GO’, or ‘FOODON’. The prefixes should come from a standard prefix registry such as BioRegistry (Hoyt *et al.* 2022) to ensure consistency across schemas and projects; SPIRES expects upper-case prefixes.

2.1 Evaluation of entity grounding

To determine the extent to which SPIRES improves entity grounding relative to prompting alone, we queried two GPT models with sets of ontology term labels with and without our grounding. We selected 100 terms at random from each of three ontologies: the Gene Ontology (GO), the Mouse Developmental Anatomy Ontology (EMAPA), and the MONDO Disease Ontology. The 16k GPT-3.5-turbo (gpt-3.5-turbo-16k) and the newly available GPT-4-turbo (gpt-4-1106-preview) models were each queried with the full term list in a single prompt each along with text requesting corresponding identifiers from the specified ontology (or, for SPIRES, a structured query based on a minimal schema). A match was considered successful for each pair of identifier and label in which the

label text was parsed as a single entity, remained unchanged in the output, and matched to the correct identifier. The full evaluation and results are available in a code notebook online (https://github.com/monarch-initiative/ontogpt-experiments/blob/main/experiments/ground_compare/Comparing_Grounding.ipynb).

2.2 Evaluation against chemical disease relation task

We evaluated SPIRES on the Biocreative Chemical-Disease-Relation task (Li *et al.* 2016). We used all 500 abstracts of the BC5CDR test set and evaluated against the set of 1066 chemical-induces-disease (CID) triples. For each triple, the predicate is fixed, and the subject and object are always identifiers drawn from the Medical Subject Headings (MeSH) vocabulary (Lipscomb 2000). Grounding was performed using multiple ontologies beyond MeSH, including three resources for chemical and drug information: Chemical Entities of Biological Interest (ChEBI) (Hastings *et al.* 2016), DrugBank (Wishart *et al.* 2018), and MedDRA (Brown *et al.* 1999) (see Supplementary Table S1 for a full list of external resources used for grounding). We used the Translator NodeNormalizer (Fecho *et al.* 2022) to normalize these to MeSH IDs to permit comparison with the test set. No fine tuning was performed. The training set was used to enhance our mappings of named entity spans to MeSH identifiers; after building this lexicon, the training set was discarded.

We provided SPIRES with a model of chemical to disease (CTD) associations based on the Biolink Model (Unni

On medium heat melt the butter and sautee the onion and bell peppers.
 Add the hamburger meat and cook until meat is well done...
 Ingredients: 1 small onion, 2 bell peppers, 2 tablespoons garlic powder...
 ...

```

label: Simple Spaghetti
description: A tomato sauce spaghetti dish with hamburger meat and vegetables.
category:
- dbpedia:Main_course          ## dbpedia ontology
- dbpedia:Italian_cuisine      ## dbpedia ontology
ingredients:
- food_item: FOODON:03301704    ## onion (whole, raw)
  quantity: 1
- food_item: FOODON:00003485    ## sweet red bell pepper (whole)
  quantity: 2
- food_item: FOODON:03301844    ## garlic powder
  quantity: 2
  unit: "[tbs_us]"              ## UCUM standard
- food_item: FOODON:03310351    ## butter
  quantity: 3
  unit: "[tbs_us]"
- food_item: FOODON:00001649    ## black or white pepper product
  quantity: 1
  unit: "[tbs_us]"
...
steps:
- action: chop
  inputs:
    - FOODON:03301704              ## onion (whole, raw)
  outputs:
    - _:ChoppedOnion              ## (no term in ontology)
- action: chop
  inputs:
    - FOODON:00003485              ## sweet red bell pepper (whole)
  outputs:
    - _:ChoppedBellPepper         ## (no term in ontology)
...
- action: add
  inputs:
    - FOODON:03301217              ## tomato sauce
    - FOODON:00002221              ## salt product
    - FOODON:00001649              ## black or white pepper product
    - FOODON:03301644              ## garlic powder
  outputs:
    - FOODON:03304014              ## spaghetti sauce with meat
  ...

```

Figure 2. Example of a portion of text to parse and a corresponding instantiation of the recipe schema from Fig. 1, using YAML syntax. Input text is truncated for brevity; the full input is available at <https://github.com/monarch-initiative/ontogpt/blob/main/tests/input/cases/recipe-spaghetti.txt>. In each attribute-value pair, the attribute is shown in bold, followed by a colon and then the value or values. For multivalued attributes, each list element value is indicated with a hyphen at the beginning of the line. Terminal elements that are value sets from ontologies and standards such as FOODON (Dooley *et al.* 2018), UCUM (Schadow *et al.* 1999), and DBPedia (Bizer *et al.* 2009) are shown here with their human-readable labels after the double-hash comment symbol. Dynamic elements are indicated via RDF blank node syntax (e.g. _:ChoppedOnion does not correspond to a named entity and serves as a placeholder).

et al. 2022). Biolink extends the simple triple model of associations to include qualifiers on the predicate, subject, and object. Subject and object qualifier information was discarded in this evaluation as extracting these details was not tested for in the original CDR benchmark. Statements with predicate qualifiers of ‘NOT’ were discarded. We configured value sets for MeSH Disease and Chemical entries manually (see the full list of identifiers used to define these sets in [Supplementary Table S2](#)). NER of chemical and disease entities was also evaluated based on ability to identify a corresponding MeSH. We compared two pre-processing approaches: a ‘chunking’ approach in which input documents were processed as separate subsegments

(essentially a sliding window approach) and a ‘no chunking’ approach in which the entirety of the test corpus document title and abstract was passed in a prompt. Two OpenAI models were used in these comparisons: gpt-3.5-turbo and gpt-4.

3 Algorithm

The SPIRES extraction procedure takes as input (i) a schema S , (ii) an entry point class C , and (iii) a text T (Fig. 4, top). It returns a structured instance i conforming to S , making use of a large language model (LLM) that allows prompt completion, such as GPT-3 and its more recent versions. The procedure is detailed below:

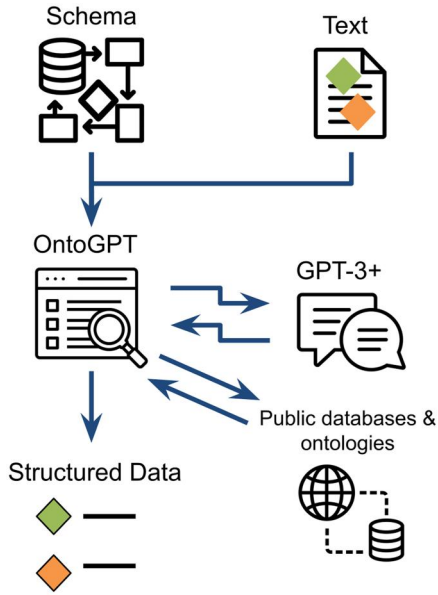


Figure 3. Overview of the SPIRES approach. A knowledge schema and text containing instances defined in the schema are processed by OntoGPT, yielding a query for GPT-3 or newer, accessed through the OpenAI API. OntoGPT parses the result, grounding extracted instances with specific entries and terms retrieved from queries of databases and ontologies where possible. The final product is a set of structured data (instances and relationship) in the shapes defined by the schema. Icons by user Khoirin from the Noun Project (<https://thenounproject.com/besticon/>).

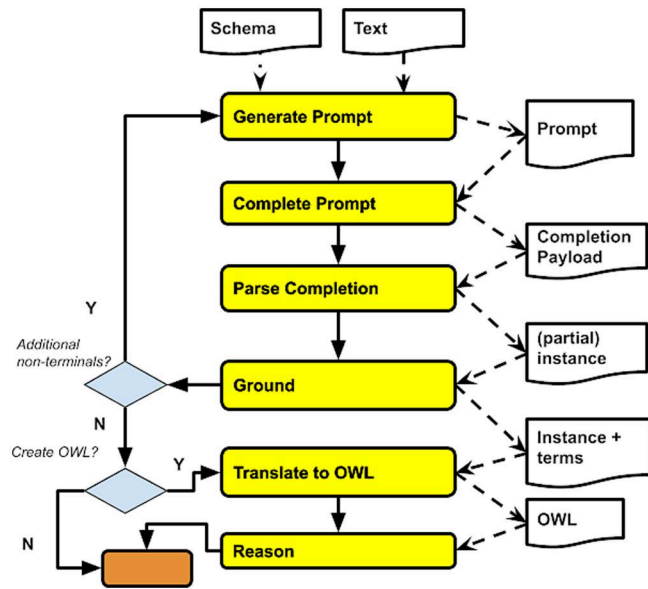


Figure 4. Flowchart depicting the SPIRES algorithm.

$SPIRES(S, C, T) :$

- 1) Generate the prompt: $p = GeneratePrompt(S, C, T)$
- 2) Perform prompt completion: $r = CompletePrompt(p)$
- 3) Parse results and recurse over nested structures:

$$iu = ParseCompletion(r, S, C)$$

- 4) Ground results using ontologies: $i = Ground(iu, S, C)$

- 5) (optional) translation to OWL: $ont = TranslateToOWL(i)$

3.1 Step 1: Generate prompt

SPIRES first generates text for a prompt (Fig. 4, *Generate Prompt*) to be provided to the LLM:

$$GeneratePrompt(S, C, T) = Instructions() + AttributeTemplate(S, C, T) + TextIntro() + T + Break() \quad (4)$$

Here, the *Instructions* function returns a piece of text such as ‘From the text below, extract the following entities in the following format’.

The *AttributeTemplate* function generates a pseudo-YAML structure that is a template for results. For each a in $Attributes(C)$, we write:

$$Name(a) + " : " + Prompt(a) + " n" \quad (5)$$

If Prompt is undefined for attribute a , then it is automatically generated from the name. If $Multivalued(a)$ is True, then the text is preceded with ‘A semicolon-separated list’.

The *TextIntro* function introduces a break between the template and the input text and is a fixed string ‘Text:’. The *Break* function is also a fixed string that serves to demarcate the end of the text and is a sequence of three break characters, e.g. ‘===’. As an example, when calling this function when $S=RecipeSchema$, $C=Ingredient$, and $T=‘garlic powder (2 tablespoons)’$, the following prompt would be generated:

Split the following piece of text into fields in the following format:

```
food_item: <the food item>
amount: <the quantity of the ingredient>
Text:
garlic powder (2 tablespoons)
===
```

Note that typical input texts will be larger, except when the function is called recursively.

3.2 Step 2: Complete the prompt

The generated prompt is provided to the LLM using a completion API (Fig. 4, *Complete Prompt*). The nature of the prompt can be adapted for different language models; the OntoGPT implementation defaults to the GPT-3.5-turbo model but is compatible with any model capable of delivering a payload that conforms to a prompt-specified structure. The intended completion results are a pseudo-YAML structure conforming to the specified template. For example, when passing the example prompt in Step 1, the return payload may be the following text:

```
food_item: garlic powder
amount: 2 tablespoons
```

3.3 Step 3: Completion result parsing and recursive extraction

The *ParseCompletion*(r, S, C) function returns a pre-grounded instance object i partially conforming to C . This step consists of two sub-steps: (i) parsing the pseudo-YAML;

(ii) recursively calling SPIRES on any inlined attributes. For the parsing step (Fig. 4, *Parse Completion*), the completion provided by the LLM is not guaranteed to be strict YAML or even conform directly to the specified template, so a heuristic approach is used. The response is separated by newlines into a list. Each line is split on the first instance of a ‘:’; the part before is matched against the attribute name, and the part after is the value, which is parsed as detailed below. Attribute matching is case-insensitive. All whitespace is normalized to underscores.

Each value v is parsed according to the range and cardinality of the matched attribute a , populating each attribute a of i :

$$i[a] = \text{ParseValue}(v) \quad (6)$$

If a is *multivalued*, then v is first split according to a delimiter (default ‘,’), and the rules below are applied on each token; otherwise the rules below are applied directly.

Rule 1: If the range is a primitive data type (i.e. string, number, or boolean) then the value is returned as-is.

Rule 2: If the range of the attribute is a class, and the attribute is noninlined (i.e. a reference) or an enumeration, then the value will be grounded, as specified in Step 4 below.

Rule 3: if the range of the attribute is an inlined class, then SPIRES is called recursively:

$$\text{SPIRES}(S, \text{Range}(a), v) \quad (7)$$

This proceeds until a noninlined class is reached. For example, given the example payload from the previous step, the attribute *food item* is a reference to an ontology class, so the value ‘garlic powder’ is grounded using the grounding procedure (Step 4). The attribute amount is a reference to an inlined class *Quantity*, so this will be recursively parsed by calling *Generate Prompt(RecipeSchema, Quantity, "2tablespoons")*.

3.4 Step 4: Grounding and normalization

All leaf nodes of the instance tree that correspond to named entities are grounded, i.e. mapped to an identifier in an existing vocabulary, ontology, or database (Fig. 4, *Ground*). Classes representing named entities can each be annotated with one or more vocabularies. Each vocabulary is identified by a unique prefix. For example, in Fig. 1, the *FoodItem* class could be annotated with both FOODON and Wikidata, indicating that grounding on labels can be performed using these vocabularies. Grounding on the string ‘garlic powder’ may then yield FOODON:03301844 when the BioPortal (Whetzel et al. 2011) or AgroPortal annotator (Jonquet et al. 2018) is used, and WIKIDATA: Q10716334 when a Wikidata normalizer is used. The final results are normalized via validation against identifier constraints for the class. If $IDS\text{Spaces}(c)$ is set, then the prefix of the identifier is checked against the list of valid prefixes. If $ValueSets(c)$ is set, then the value returned must be present in the value set.

3.5 Step 5: Translation to OWL and reasoning

Step 4 produces an instance tree that can be directly represented in JSON or YAML syntax (both of which allow for arbitrary nesting of objects). For some KBs, this is sufficient. Further conversion to an ontological representation in OWL (Fig. 4, *Translate to OWL*), and additional reasoning steps, then support checking for consistency and population of

missing axioms. There are multiple methods for translating to OWL, including ROBOT templates (Jackson et al. 2019), DOSDPs (Osuni-Sutherland et al. 2017), and OTTR (Kindermann et al. 2018).

4 Implementation

We provide an implementation of SPIRES in Python as part of the OntoGPT Python package (<https://github.com/monarch-initiative/ontogpt>), which provides both a command line interface (CLI) and a simple web application (Supplementary Fig. S1). SPIRES uses LinkML (Moxon et al. 2021) as its Knowledge Schema language. This allows for a full representation of the necessary schema elements while incorporating LinkML’s powerful mechanism for specifying static and dynamic value sets. For example, a value set can be constructed as a declarative query of the form ‘include branches A, B, and C from ontology O_1 , excluding sub-branch D, and include all of ontology O_2 ’. The LinkML framework also supports converting schemas to LinkML from forms such as SHACL (Pareti and Konstantinidis 2022), JSON-Schema (<http://json-schema.org/>), or SQL Data Definition Language, allowing their use with SPIRES.

SPIRES performs grounding and normalization with the Ontology Access Kit library (OAKlib) (<https://github.com/INCATools/ontology-access-kit>), which provides interfaces for multiple annotation tools (i.e. those providing links to external vocabularies and ontologies), including the Gilda entity normalization tool (Gyori et al. 2022), the BioPortal annotator (Jonquet et al. 2009), and the Ontology Lookup Service (Jupp et al. 2015). For identifier normalization a number of services can be used, including OntoPortal mappings, with the default being the NCATS Biomedical Translator Node Normalizer (Fecho et al. 2022).

The results of extraction can optionally be further processed using LinkML-OWL (<https://zenodo.org/record/7384531>), which generates an OWL representation of instance data using mappings specified in a LinkML schema. This OWL file can be used as an input to ROBOT (Jackson et al. 2019) to run OWL reasoning to check for logical inconsistencies and perform automated classification.

4.1 Standard templates for multiple applications

The SPIRES implementation comes with a growing collection of ready-made schemas for multiple applications. These are primarily life-science focused, e.g. deriving a pathway from a Mechanism of Action description in a database such as DrugBank. We also include a schema for food recipes to demonstrate general applicability in domains beyond the environmental and life sciences. Table 1 lists a selection of the pre-made schemas.

4.2 Extraction of recipe ontologies from websites

To demonstrate the full functionality of OntoGPT we created a pipeline for extracting recipes from websites and generating an OWL ontology from the combined outputs. Recipes are extracted using the recipe-scrappers Python module (<https://github.com/hhursey/recipe-scrappers>). The pipeline takes the output of scraping, concatenates the results into a text, then feeds this to OntoGPT using the recipe template. We use LinkML-OWL to map the recipe template to OWL axioms, such that each recipe is represented as a class defined by its ingredients and its steps. We use ROBOT to extract the

Table 1. Pre-made schemas.^a

Schema	Use Case(s)	Identifiers	Text inputs
Food Recipes	Enforcing consistent structure on stepwise processes	FOODON, UO	Unstructured and semi-structured recipes
Drug mechanisms	Integrating drug descriptions	MONDO, CHEBI, MESH	Mechanism of Action (MOA) descriptions
Chemical-disease interactions	Assembling knowledge graphs of chemical-impacted phenotypes	MESH	Abstracts describing effects of chemicals on conditions
Metagenomic Samples	Standardizing metadata for metagenomics	ENVO	Descriptions of environmental samples
Mendelian Diseases	Extracting disease relationships from literature	MONDO, HPO	Case studies or descriptions of Mendelian diseases

^a Example use cases are included but are not comprehensive. Note the CTD schema is deliberately restricted to only use the MESH vocabulary for purposes of evaluation. Identifiers refers to all ontologies, value sets, and other unique term sets incorporated in a given schema.

relevant parts of the FOODON ontology, and merge this with the extraction results, combined with a manually coded simple recipe classification with defined classes for groupings such as ‘Meat Recipe’ and ‘Wheat Based Recipe’. We use the Elk reasoner (Kazakov and Klinov 2015) to classify the results. The results of this process are highlighted in [Supplementary Fig. S3](#).

4.3 Entity grounding

Grounding entities with ontology terms is part of the core functionality of SPIRES and its value is well demonstrated in a direct comparison with the straightforward approach of directly querying an LLM with term descriptions. If we request the GO term for ‘integrase activity’ we expect the response to include GO:0008907, e.g. Of 100 GO terms chosen at random, SPIRES returned the correct identifiers for 98 when using GPT-3.5-turbo and 97 with GPT-4-turbo. Without SPIRES, GPT-3.5-turbo returned just 3 correct identifiers. Though it yielded 100 putative matches, few included correct GO identifiers. This ‘mass hallucination’ may be an artifact of prompting with terms lacking surrounding context. Even so, it may be challenging to determine how much context is sufficient to improve grounding. GPT-4-turbo demonstrated a different challenge by consistently refusing to retrieve identifiers, returning responses such as ‘As an AI developed before 2023, I do not have real-time access to databases...’. For the EMAPA mouse anatomy ontology, SPIRES returned correct identifiers for all 100 term descriptions, while GPT-3.5-turbo repeatedly provided identifiers from the EHDAA2 human anatomy ontology instead. GPT-4-turbo refused to ground EMAPA terms as it had with GO. MONDO terms posed some surprising difficulty: SPIRES with GPT-3.5-turbo correctly returned 97 of 100 identifiers but SPIRES with GPT-4-turbo returned just 18 correct matches. In some cases, this may have been due to incorrectly parsing entities (e.g. parsing ‘UV-induced skin damage, susceptibility to’ as ‘skin damage’). As with GO, prompting without SPIRES only returned one correct identifier at most from both GPT-3.5-turbo and GPT-4-turbo.

4.4 Evaluation on BioCreative chemical disease relation task

We evaluated SPIRES on the BioCreative Chemical-Disease-Relation (BC5CDR) task corpus. To demonstrate the zero-shot learning (ZSL) approach, we did not perform any fine tuning using the training set. The training set was used to enhance our mappings of named entity spans to MeSH identifiers and was then discarded. For our CTD schema (see [Supplementary Fig. S2](#)), we follow the Biolink Model (Unni

et al. 2022) which extends the simple triple model of associations to include qualifiers on the predicate, subject, and object. This yields finer-grained predictions; e.g. SPIRES correctly parses the statements in [Table 2](#). In these cases, SPIRES grounds the drug entity Cromakalim to its corresponding MeSH identifier and extracts its relationship with vasodilation along with a qualifier noting the observation is specific to ‘large and small coronary vessels’, an anatomical entity worthy of further grounding (though this was not explored within the original BC5CDR task). Similarly, the correctly extracted relationship between lithium and hypercalcemia includes the qualifier that the observation pertains to chronic lithium exposure.

When evaluating, we discard subject and object qualifier information, as this is not tested for in the original CDR benchmark. If the predicate qualifier is ‘NOT’ then we discard the whole statement. Note that in the examples in [Table 2](#), even though we evaluated the first two statements to be a correct interpretation of the abstract, they were counted as false negatives; the corresponding triple was not in the test set, presumably an error of omission.

For SPIRES, we saw initially encouraging results on the BC5CDR task with chunking and GPT-3.5-turbo: we observed an F-score of 41.16, precision of 0.43, and recall of 0.39. Using the ‘no chunking’ approach (i.e. no preprocessing of the test document) yielded an F-score of 36.64 (precision 0.63, recall 0.26) with GPT-3.5-turbo and an F-score of 43.80 (precision 0.69, recall 0.32). For NER results alone (i.e. correct grounding against MeSH for chemical and disease entities), see [Supplementary Table S3](#).

These results place SPIRES just below the average of all 18 teams that participated in the original CDR challenge. We assume all 18 teams used the full training set, whereas with SPIRES there was no task-specific training or fine tuning. For comparison, Luo *et al.* report an F-score of 44.98 on BC5CDR with their biomedical domain-specific, trained-from-scratch BioGPT model (Luo *et al.* 2022). We note that the best-scoring RE results from the CDR task achieved an impressive score of 0.57, though with a model trained on a large and carefully engineered set of training examples (Xu *et al.* 2015). SPIRES bypasses this step but may see further improvement with fine-tuned and/or domain-specific LLMs.

5 Discussion

5.1 Comparable methods

SPIRES is a well-developed and generally model-agnostic approach for information extraction designed with structured

Table 2. Extracted relation examples.^a

Source	Subject	Subject qualifier	Object	Object qualifier
2160002	MESH: D019806 Cromakalim	Chronic	MESH: D014664 Vasodilation	Large and small coronary vessels
2160002	MESH: D020110 Pinacidil		MESH: D014664 Vasodilation	Large and small coronary vessels
19154241	MESH: D008094 Lithium		MESH: D006934 Hypercalcemia	Transient
10327032	MESH: D005472 Fluorouracil		MESH: D001927 Brain Diseases	

^a All predicates are 'INDUCES'. Sources are PubMed identifiers (PMIDs). PMID 2160002 is 'Vasodilation of large and small coronary vessels and hypotension induced by cromakalim and pinacidil' (Giudicelli *et al.* 1990). PMID 19154241 is a case report on lithium therapy (Rizwan and Perrier 2009). PMID 10327032 is a study of hyperammonemic encephalopathy risks in cancer patients (Liaw *et al.* 1999).

schemas and standardized ontologies in mind. Some recent efforts have made great strides in leveraging the first type of resource, i.e. they address the task of aligning extracted information with pre-defined data models. The approach described by Dagdelen and Dunn *et al.* (2024) employs engineered schemas to extract structured relationships from unstructured text in materials chemistry (Dunn *et al.* 2022). The authors of the LLMs4OL approach also explored application of LLMs to information extraction, but concluded that the models are not yet sufficiently flexible for ontology-driven needs (Babaei Giglou *et al.* 2023). We also consider the task of ontology alignment to be related to our efforts; we have found that LLMs can noticeably improve accuracy in ontology alignment (Matentzoglou *et al.* 2023) and the development of general frameworks such as Agent-OM (Qiang *et al.* 2023) may further improve the grounding inherent to information extraction.

5.2 Choosing a model

OntoGPT currently supports both select open LLMs and the OpenAI API. Running OntoGPT across a large corpus with OpenAI models may be prohibitively expensive for some users. In addition, the use of this API involves closed models with inscrutable training data, which may be plagued by biases (Bender *et al.* 2021). Though our experiments here generally concern GPT-3 and 4, the rapid pace of model development will ensure access to progressively more capable (and ideally, more transparent) language models. Smaller LMs such as LLaMA have been shown to outperform models ten times their size (Touvron *et al.* 2023), and it is possible to fine-tune these into instruction following models (Zhang *et al.* 2023). LLMs based on LLaMA2 and adapted for biomedical language, including BioMedGPT-LM (Luo *et al.* 2023) and Radiology-Llama2 (Liu *et al.* 2023), may complement the grounding provided through SPIRES.

5.3 Reliability and hallucinations

A common problem with LLMs is hallucination of results (producing factually invalid statements that are not consistent with the input text) (Ji *et al.* 2023, Bender *et al.* 2021). We crafted prompts to limit hallucination, asking only for the LM to extract what was found in the text, and keeping default low-creativity settings. On examination we found that hallucinations were generally infrequent, with most false positives and negatives attributable to incorrect RE. It is worth noting that LLM interfaces designed for direct function calling may duplicate some of the data structure enforcement afforded by SPIRES but do not alleviate the issue of hallucination: a model may still improperly associate real or fictional ontology identifiers with extracted entities when queried without aid of our approach.

Some text generation may yield technically correct results. For example, one result extracted from the title 'Increased frequency and severity of angio-oedema related to long-term therapy with angiotensin-converting enzyme inhibitor in two patients', yielded 'Lisinopril INDUCES angio-oedema'. Lisinopril is in fact a subtype of ACE inhibitor, and the extracted association is supported by other literature. However, this more precise statement is not the one that is in the original text. Presumably the LM is substituting the class of drug with a specific member here, but it is unclear why it does it on this occasion. Until there are better methods to control this hallucination and explain justifications for statements in terms of the text and prior knowledge, results from LMs should be carefully validated before being entered into KBs.

SPIRES is a new approach to information extraction that leverages recent advances in large language models to populate complex knowledge schemas from unstructured text. It uses ZSL to identify and extract relevant information from query text, which is then normalized and grounded using existing ontologies and vocabularies. SPIRES requires no model tuning or training data. The approach is customizable, flexible, and can be used to populate knowledge schemas across varied domains. We envision SPIRES being used not in isolation, but rather in synergistic strategies combining human expertise, linguistic pattern recognition, deep learning and classical deductive reasoning approaches. SPIRES is one component of a growing toolkit of methods for transforming noisy, heterogeneous information into actionable knowledge.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

This work was supported by the National Institutes of Health National Human Genome Research Institute [RM1 HG010860]; National Institutes of Health Office of the Director [R24 OD011883]; and the Director, Office of Science, Office of Basic Energy Sciences, of the US Department of Energy [DE-AC0205CH11231 to J.H.C., H. H., N.L.H., M.J., S.M., J.T.R., and C.J.M.]. We also gratefully acknowledge Bosch Research for their support of this research project.

Data availability

The data underlying this article are available in Zenodo at <https://dx.doi.org/10.5281/zenodo.7894107>.

References

- Atea S, Kruschwitz U. Is ChatGPT a biomedical expert? – exploring the Zero-Shot performance of current GPT models in biomedical tasks. In: *CLEF 2023: Conference and Labs of the Evaluation Forum*, Thessaloniki, Greece: CLEF Initiative, 2023.
- Babaei Giglou H, D'Souza J, Auer S. LLMs4OL: large language models for ontology learning. In: *The Semantic Web – ISWC 2023*. Switzerland: Springer Nature, 2023, 408–27. <https://doi.org/10.1007/978-3-031-47240-4>
- Bender EM, Gebru T, McMillan-Major A *et al.* On the dangers of stochastic parrots: can language models be too big? In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, New York, NY, USA: Association for Computing Machinery, 2021, 610–23. ISBN 9781450383097. <https://doi.org/10.1145/3442188.3445922>
- Bizer C, Lehmann J, Kobilarov G *et al.* DBpedia – a crystallization point for the web of data. *J Web Semant* 2009;7:154–65. <https://doi.org/10.1016/j.websem.2009.07.002>
- Brown EG, Wood L, Wood S. The medical dictionary for regulatory activities (MedDRA). *Drug Saf* 1999;20:109–17. <https://doi.org/10.2165/00002018-199920020-00002>
- Brown TB, Mann B, Ryder N *et al.* Language models are Few-Shot learners. arXiv, <https://doi.org/10.48550/arXiv.2005.14165>, arXiv:2005.14165, 2020, preprint: not peer reviewed.
- Dooley DM, Griffiths EJ, Gosal GS *et al.* FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. *NPJ Sci Food* 2018;2:23. <https://doi.org/10.1038/s41538-018-0032-6>
- Dagdelen J, Dunn A, Lee S *et al.* Structured information extraction from scientific text with large language models. *Nat Commun* 2024;15:1. <https://doi.org/10.1038/s41467-024-45563-x>.
- Ettinger A. What BERT is not: lessons from a new suite of psycholinguistic diagnostics for language models. *Trans Assoc Comput Linguist* 2020;8:34–48. <https://doi.org/10.1162/tacla00298>
- Fabregat A, Jupe S, Matthews L *et al.* The reactome pathway knowledgebase. *Nucleic Acids Res* 2018;46:D649–55. <https://doi.org/10.1093/nar/gkx1132>
- Fecho K, Thessen AT, Baranzini SE *et al.*; Biomedical Data Translator Consortium. Progress toward a universal biomedical data translator. *Clin Transl Sci* 2022;15:1838–47. <https://doi.org/10.1111/cts.13301>
- Furrer L, Jancso A, Colic N *et al.* OGER: hybrid multi-type entity recognition. *J Cheminform* 2019;11:7. <https://doi.org/10.1186/s13321-018-0326-3>
- Giudicelli JF, la Rochelle CD, Berdeaux A. Effects of cromakalim and pinacidil on large epicardial and small coronary arteries in conscious dogs. *J Pharmacol Exp Ther* 1990;255:836–42.
- Graybeal J, Jonquet C, Fiore N *et al.* Adoption of BioPortal's ontology registry software: the emerging OntoPortal community. In: *13th Research Data Alliance Plenary Meeting (RDA P13)*, Philadelphia, United States: Research Data Alliance 2019.
- Gyori BM, Hoyt CT, Steppi A. Gilda: biomedical entity text normalization with machine-learned disambiguation as a service. *Bioinform Adv* 2022;2:vbac034. <https://doi.org/10.1093/bioadv/vbac034>
- Hastings J, Owen G, Dekker A *et al.* ChEBI in 2016: improved services and an expanding collection of metabolites. *Nucleic Acids Res* 2016;44:D1214–9. <https://doi.org/10.1093/nar/gkv1031>
- Hoyt CT, Balk M, Callahan TJ *et al.* Unifying the identification of biomedical entities with the bioregistry. *Sci Data* 2022;9:714. <https://doi.org/10.1038/s41597-022-01807-3>
- Jackson RC, Balhoff JP, Douglass E *et al.* ROBOT: a tool for automating ontology workflows. *BMC Bioinformatics* 2019;20:407. <https://doi.org/10.1186/s12859-019-3002-3>
- Ji Z, Lee N, Frieske R *et al.* Survey of Hallucination in Natural Language Generation. *ACM Comput Surv* 2023;55:12. <https://doi.org/10.1145/3571730>.
- Jonquet C, Shah NH, Musen MA. The open biomedical annotator. *Summit Transl Bioinform* 2009;2009:56–60.
- Jonquet C, Toulet A, Arnaud E *et al.* AgroPortal: a vocabulary and ontology repository for agronomy. *Comput Electron Agric* 2018;144:126–43. <https://doi.org/10.1016/j.compag.2017.10.012>
- Jupp S, Burdett T, Malone J *et al.* A new ontology lookup service at EMBL-EBI. In: *Proceedings of the 8th International Conference on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2015)*, Cambridge, United Kingdom: CEUR Workshop Proceedings, Vol. 1546, 2015, 118–9.
- Kazakov Y, Klinov P. Advancing ELK: not only performance matters. In: *Proceedings of the 28th International Workshop on Description Logics (DL-15)*, Athens, Greece: CEUR Workshop Proceedings, Vol. 1350, 2015.
- Khambete MP, Su W, Garcia JC *et al.* Quantification of BERT diagnosis generalizability across medical specialties using semantic dataset distance. *AMIA Jt Summits Transl Sci Proc* 2021;2021:345–54. <https://doi.org/10.1371/journal.pone.0112774>
- Kindermann C, Lupp DP, Sattler U *et al.* Generating Ontologies from Templates: A Rule-Based Approach for Capturing Regularity. In: *Proceedings of the 31st International Workshop on Description Logics (DL 2018)*, Tempe, AZ, USA: CEUR Workshop Proceedings, Vol. 2211, 2018.
- Li J, Sun Y, Johnson RJ *et al.* BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database (Oxford)* 2016;2016:baw068. <https://doi.org/10.1093/database/baw068>
- Liaw CC, Wang HM, Wang CH *et al.* Risk of transient hyperammonemic encephalopathy in cancer patients who received continuous infusion of 5-fluorouracil with the complication of dehydration and infection. *Anticancer Drugs* 1999;10:275–81. <https://doi.org/10.1097/00001813-199903000-00004>
- Lipscomb CE. Medical subject headings (MeSH). *Bull Med Libr Assoc* 2000;88:265–6.
- Liu Z, Li Y, Shu P *et al.* Radiology-Llama2: Best-in-Class large language model for radiology. arXiv, <https://doi.org/10.48550/arXiv.2309.06419>, arXiv:2309.06419, 2023, preprint: not peer reviewed.
- Luo R, Sun L, Xia Y *et al.* BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Brief Bioinform* 2022;23:bbac409. <https://doi.org/10.1093/bib/bbac409>
- Luo Y, Zhang J, Fan S *et al.* BioMedGPT: open multimodal generative pre-trained transformer for BioMedicine. arXiv, <https://doi.org/10.48550/arXiv.2308.09442>, arXiv:2308.09442, 2023, preprint: not peer reviewed.
- Matentzoglou N, Caufield JH, Hegde HB *et al.* MapperGPT: large language models for linking and mapping entities. arXiv, <https://doi.org/10.48550/arXiv.2310.03666>, arXiv:2310.03666, 2023, preprint: not peer reviewed.
- Moxon S, Solbrig H, Unni D *et al.* The linked data modeling language (LinkML): a general-purpose data modeling framework grounded in machine-readable semantics. In: *Proceedings of the International Conference on Biomedical Ontologies (ICBO 2021)*, Bolzano, Italy: CEUR Workshop Proceedings, Vol. 3073, 2021, 148–51.
- Osumi-Sutherland D, Courtot M, Balhoff JP *et al.* Dead simple OWL design patterns. *J Biomed Semantics* 2017;8:18. <https://doi.org/10.1186/s13326-017-0126-0>
- Pareti P, Konstantinidis G. A review of SHACL: from data validation to schema reasoning for RDF graphs. In: Šimkus M, Varzinczak I (eds.), *Reasoning Web. Declarative Artificial Intelligence*. Cham, Switzerland: Springer International Publishing, 2022, 115–44. <https://doi.org/10.1007/978-3-030-95481-9>
- Qiang Z, Wang W, Taylor K. Agent-OM: leveraging large language models for ontology matching. arXiv, <https://doi.org/10.48550/arXiv.2312.00326>, arXiv:2312.00326, 2023, preprint: not peer reviewed.
- Rizwan MM, Perrier ND. Long-term lithium therapy leading to hyperparathyroidism: a case report. *Perspect Psychiatr Care* 2009;45:62–5. <https://doi.org/10.1111/j.1744-6163.2009.00201.x>

- Schadow G, McDonald CJ, Suico JG *et al.* Units of measure in clinical information systems. *J Am Med Inform Assoc* 1999;6:151–62. <https://doi.org/10.1136/jamia.1999.0060151>
- The Gene Ontology Consortium. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Res* 2019;47:D330–8. <https://doi.org/10.1093/nar/gky1055>
- Touvron H, Lavril T, Izacard G *et al.* Llama: open and efficient foundation language models. arXiv, <https://doi.org/10.48550/arXiv.2302.13971>, arXiv:2302.13971, 2023, preprint: not peer reviewed.
- Unni DR, Moxon SAT, Bada M *et al.*; The Biomedical Data Translator Consortium. Biolink model: a universal schema for knowledge graphs in clinical, biomedical, and translational science. *Clin Translational Sci* 2022;15:1848–55. <https://doi.org/10.1111/cts.13302>
- Vaswani A, Shazeer N, Parmar N *et al.* Attention is all you need. In: *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA. Red Hook, NY, USA: Curran Associates Inc. 2017. <https://doi.org/10.48550/arXiv.1706.03762>
- Vrandečić D. Wikidata: a free collaborative knowledgebase. *Commun ACM* 2014;57:10. <https://doi.org/10.1145/2629489>.
- Wachter RM, Brynjolfsson E. Will generative artificial intelligence deliver on its promise in health care? *JAMA* 2023;331:65–9. <https://doi.org/10.1001/jama.2023.25054>
- Wang Y, Fu S, Shen F *et al.* The 2019 n2c2/OHNL track on clinical semantic textual similarity: overview. *JMIR Med Inform* 2020;8:e23375. <https://doi.org/10.2196/23375>
- Whetzel PL, Noy NF, Shah NH *et al.* BioPortal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic Acids Res* 2011;39:W541–5. <https://doi.org/10.1093/nar/gkr469>
- Wishart DS, Feunang YD, Guo AC *et al.* DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res* 2018;46:D1074–82. <https://doi.org/10.1093/nar/gkx1037>
- Xu J, Wu Y, Zhang Y *et al.* UTH-CCB@BioCreative V CDR task: identifying chemical-induced disease relations in biomedical text. In: *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, Sevilla, Spain: CNIO Centro Nacional de Investigaciones Oncológicas, 2015, 254–9.
- Zhang R, Han J, Zhou A *et al.* Llama-adapter: efficient fine-tuning of language models with zero-init attention. arXiv, <https://doi.org/10.48550/arXiv.2303.16199>, arXiv:2303.16199, 2023, preprint: not peer reviewed.