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 Joachimiak**[1], **Nicolas Matentzoglu**[3], **HyeongSik Kim**[4], **Sierra Moxon**[1], **Justin T. Reese**[1], **Melissa A. Haendel**[5], **Peter N. Robinson**[6], **and Christopher J. Mungall**[1]

[1]Division of Environmental Genomics and Systems Biology, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[2]Institute of Data Science, Faculty of Science and Engineering, Maastricht University, Maastricht, The Netherlands
[3]Semanticly Ltd, Athens, Greece
[4]Robert Bosch LLC, Sunnyvale, CA 94085, USA
[5]Anschutz Medical Campus, University of Colorado, Aurora, CO 80217, USA
[6]The Jackson Laboratory, Bar Harbor, ME 04609, USA

Abstract

Creating knowledge bases and ontologies is a time consuming task that relies on a 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 arbitrary complex nested knowledge schemas.

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 (ZSL) 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 GPT-3+ to obtain a set of responses matching the provided schema. SPIRES uses existing ontologies and vocabularies to provide identifiers for all matched elements.

We present examples of use of SPIRES in different domains, including extraction of food recipes, multi-species cellular signaling pathways, disease treatments, multi-step drug mechanisms, and chemical to disease causation graphs. Current SPIRES accuracy is comparable to the mid-range of existing Relation Extraction (RE) methods, but has the advantage of easy customization, flexibility, and, crucially, the ability to perform new tasks in the absence of any 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.

SPIRES is available as part of the open source OntoGPT package: `https://github.com/monarch-initiative/ontogpt`.

Keywords large language models · linked data · schemas · text mining · ontologies · artifician intelligence

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 [1] contain broad contextual knowledge about multiple domains of knowledge, and are used for a wide variety of tasks, such as integrative analyses of otherwise disconnected data and enriching web applications (for example, 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) [2] and the Reactome biological pathway KB [3] 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 Language Models (LMs) such as BERT [4] have shown promise on question-answer benchmarks over natural language [5], but still lack the ability to generalize [6]. These techniques have other limitations, such as the inability to leverage existing knowledge without additional domain-specific engineering [7, 8], as well as being prone to hallucinations [9] (i.e., generating incorrect statements) and insensitivity to negations [10]. Applications such as clinical decision support require precision and reliability not yet demonstrated by LMs, though recent demonstrations offer promise [11, 12, 13].

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 potentially assist manual knowledge curation and acquisition while validating facts prior to insertion into the KB. Validation can employ both manual and automated approaches. One powerful validation approach that leverages prior knowledge is to perform logical reasoning using an ontology that makes use of expressed Web Ontology Language (OWL) Axioms [14]. 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 [15] 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 are able to generalize and perform zero-shot or few-shot learning on these tasks, by reframing these tasks as prompt-completion tasks [16].

Most KBs are built upon rich 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 itself a complex knowledge structure, involving an action, utensils, and quantified inputs and outputs, where inputs and outputs might be a tuple of a food type plus a state (e.g. cooked) (Figure 1). Ontologies or vocabularies such as FOODON [12] 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 and tailoring.

A schema is used to provide a structure for data. For example, the recipe schema used in Figure 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 [17] syntax.

There are a number of frameworks for representing schemas. JSON-Schema [21] provides a means of structuring JSON data, while SQL includes a Data Definition Language (DDL) for structuring data stored in relational databases. Semantically aware schema languages such as the Shapes Constraint Language (SHACL) [22], Fast Healthcare Interoperability Resources (FHIR) [23], and the Linked Data Modeling Language (LinkML) [24] enhance schemas through the use of interoperable ontologies, and can also serve as schemas for Linked Data and Knowledge Graphs (KGs) [25].

Here we present Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES), an automated approach for population of custom schemas and ontology models in any domain. The objective of SPIRES is to generate an instance (aka object) from a text, where that instance has a collection of attribute-value associations, with each value being either a primitive (e.g. string, number, or identifier), or another inlined (i.e., embedded) instance (Figure 2). SPIRES integrates the flexibility of LLMs with the reliability of publicly-available databases and ontologies (Figure 3). This strategy allows the method to fill out schemas with linked data while bypassing a need for training examples. Unlike simple Relation Extraction (RE) methods, SPIRES can be used to populate schemas that exhibit nesting, in

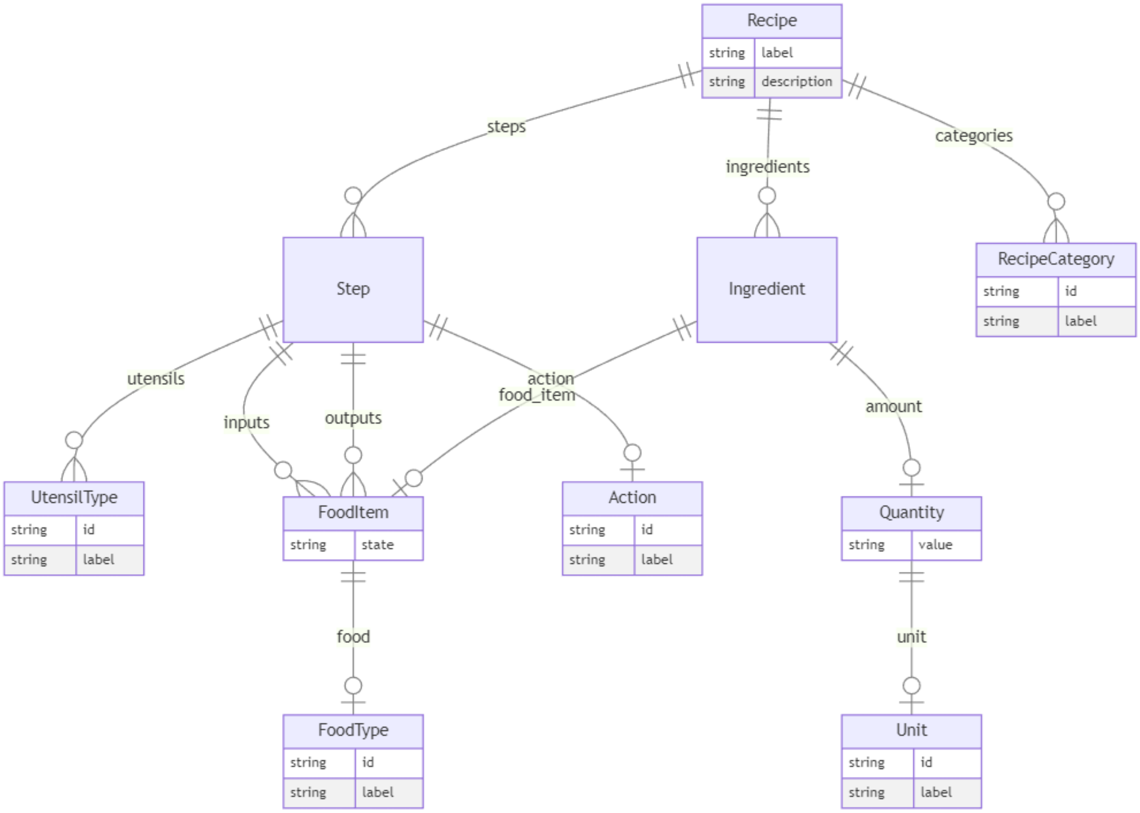


Figure 1: Examples of a recipe schema. Boxes denote classes, and arrows denote attributes whose range are classes (compound attributes). Crows feet above boxes denote multivalued attributes. Attributes whose ranges are primitives or value sets are shown within each box. In this case, the top level container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

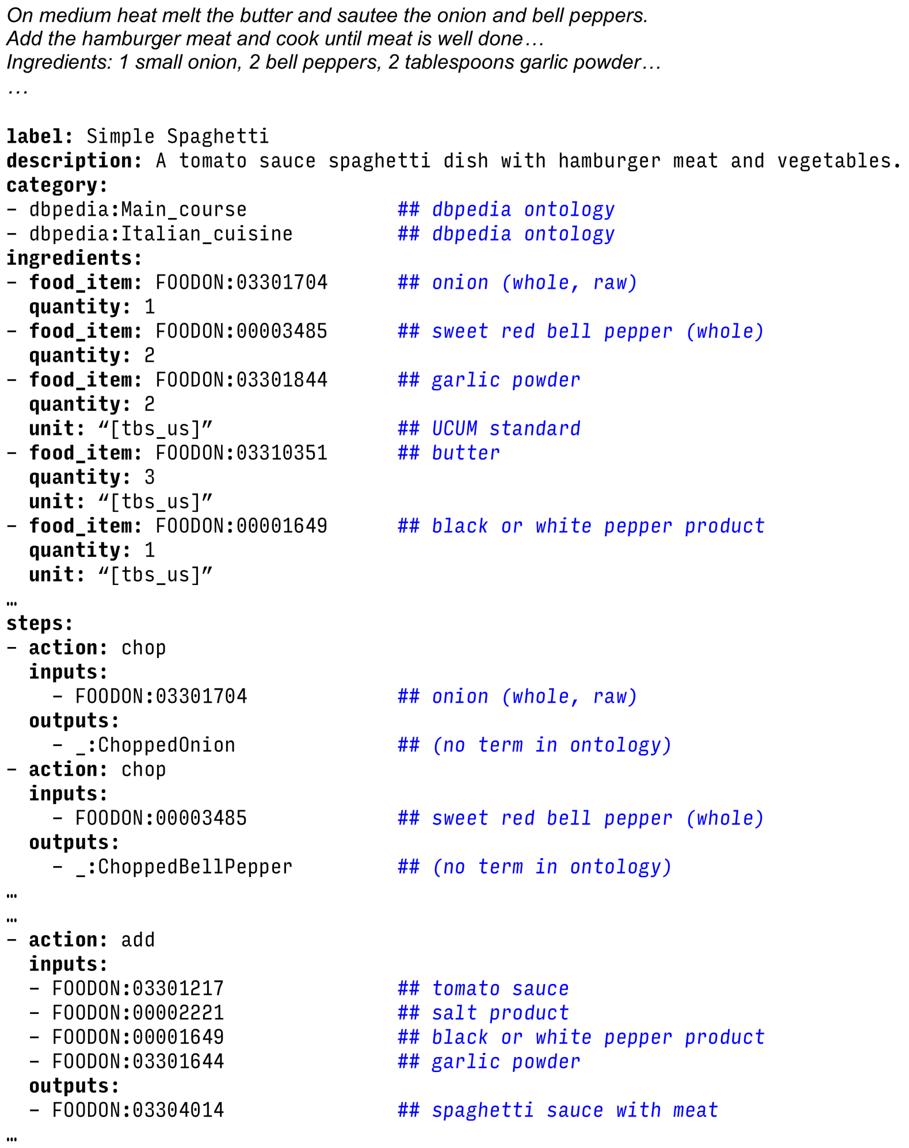
which complex classes may have attributes whose ranges are themselves complex classes. SPIRES also makes use of a highly flexible grounding approach that can leverage over a thousand ontologies in the OntoPortal Alliance [26], as well as biomedical lexical grounders such as Gilda [27] and OGER [28].

2 Methods

2.1 Knowledge Schema Formulation

All guiding knowledge schema used by SPIRES are defined in LinkML. This modeling language supports extensive semantic linking and high flexibility across data processing toolchains.

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 (1) a primitive type such as a string or number; (2) a class; or (3) 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. Additionally, each schema element can have arbitrary metadata associated with it.



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 Figure 1, using YAML syntax, which allows for nesting of structures. 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 [18], UCUM [19], and DBPedia [20] are shown here with their human-readable labels in blue after the double-hash comment symbol. Dynamic elements are indicated via RDF blank node syntax (e.g. `_:ChoppedOnion` doesn't correspond to an existing named entity in a vocabulary), which serves as a placeholder for the term.

Formally, a schema S consists of n classes:

$$Classes\,(S) = \{c_1, \cdots, c_n\} \tag{1}$$

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 Figure 1).

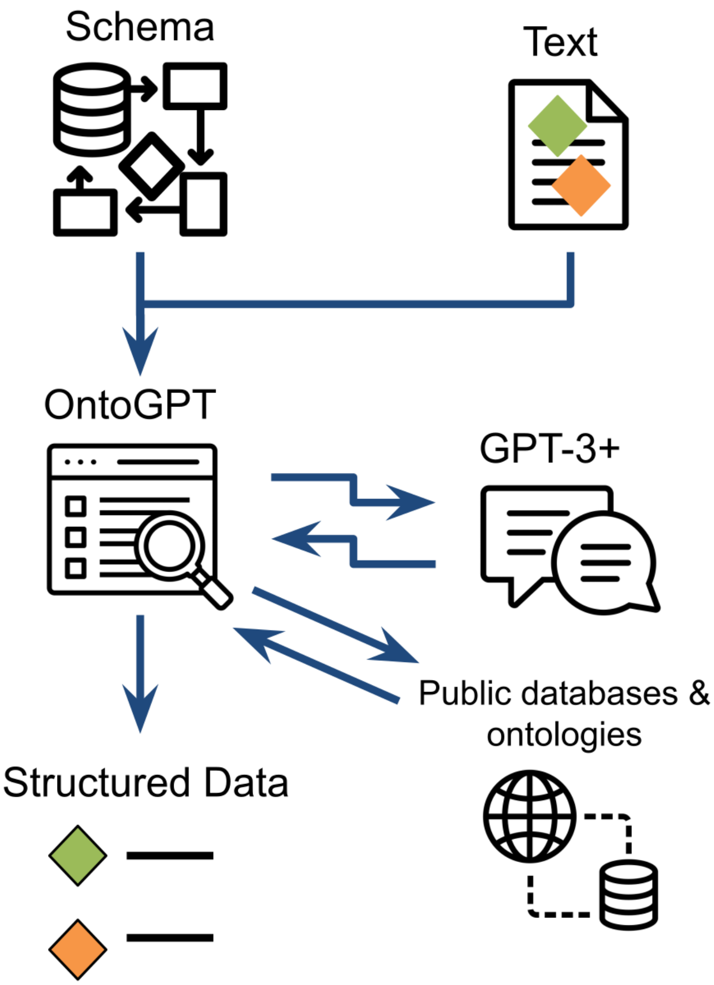


Figure 3: Overview of the SPIRES approach. A knowledge schema and text containing instances defined in the schema are processed by OntoGPT, yielding a query for GPT-3 or newer, accessed through the OpenAI API. OntoGPT parses the result, grounding extracted instances with specific entries and terms retrieved 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/`).

Each class c_i has an ordered list of attributes:

$$Attributes\,(c_i) = \{c_i a_1, \cdots, c_i a_m\} \tag{2}$$

Instances of c_i may have *values* specified for each of these attributes.

An attribute *a* can have a number of properties associated with it:

- $Name(a)$: the name of the attribute; for example, "summary" or "steps".
- $Multivalued(a)$ = $\{True, False\}$, 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.
- $Identifier(a)$ = $\{True, False\}$, indicating whether *a* is a persistent identifier for instances, such as the FOODON identifiers in Figure 2.
- $Prompt(a)$ = string, which is a user-specified custom prompt for that attribute.

- $Range(a)$: 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 Figure 1, the range of the *ingredients* attribute is Ingredient, and the range of the *id* attribute is a string.
- $Inlined(a)$ = $\{True, False\}$, indicating, when the range is a class, if the object should be nested/embedded, or passed by reference.

Additionally, a class *c* can include a set of constraints on the identifier:

$$IDSpaces\,(c_i) = \{prefix_i, \cdots, prefix\} \tag{3}$$

The constraint set is a list of strings that are the allowable prefixes that the identifier can take–for example, “WIKIDATA”, “MESH”, “GO”, or “FOODON”. The prefixes should come from a standard prefix registry such as BioRegistry [29] to ensure consistency across schemas and projects.

A *ValueSet* specifies an allowable list of string or identifier values:

$ValueSets(c)$: a list of atomic values from which values of a can be drawn from, 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.

2.2 SPIRES Algorithm

The SPIRES extraction procedure takes as input (1) a schema *S*, (2) an entry point class *C*, and (3) a text *T*, and 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 broken into steps, illustrated in Figure 4, and detailed below:

$SPIRES(S, C, T)$:

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: $ont = TranslateToOWL(i)$

2.2.1 Step 1: Generate Prompt

The first step is to generate text for a prompt that is to be fed to the LLM:

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

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:

$$Name\,(a) + \text{"} : \text{"} + Prompt\,(a) + \text{"}\backslash n\text{"} \tag{5}$$

If Prompt is not defined for attribute *a*, then one 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. “===”.

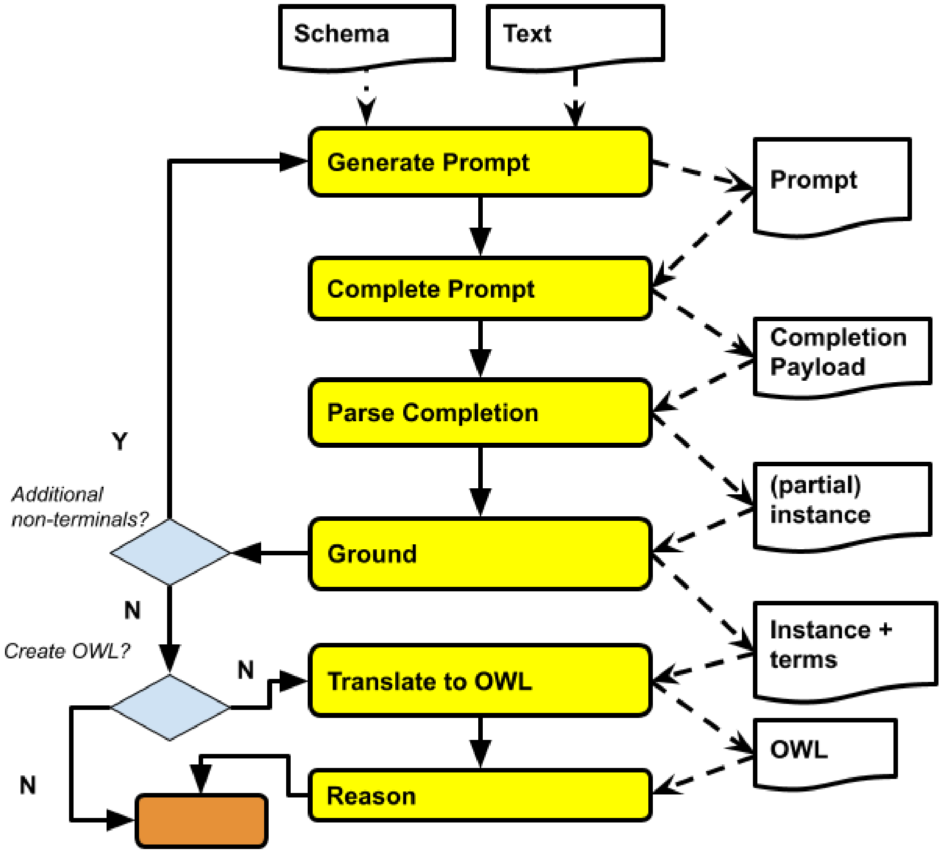


Figure 4: Flowchart depicting the SPIRES algorithm. The input documents are a knowledge schema and a text.

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 typically input texts will be larger, except when the function is called recursively (see below).

2.2.2 Step 2: Complete the Prompt

The generated prompt is fed directly to the LLM using a completion API.

The nature of the prompt can be adapted for different language models; the OntoGPT implementation depends on the GPT-3 text-davinci-003 model [30], which is 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 above, the return payload may be the following text:

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

2.2.3 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: (1) parsing the pseudo-YAML; (2) recursively calling SPIRES on any inlined attributes.

For the parsing step, 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 (it is not uncommon for GPT-3+ to creatively modify the response, introducing whitespace and capitalization).

Each value *v* is parsed according to the range and cardinality of the matched attribute *a*. This is used to populate attribute *a* of *i*:

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

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 non-inlined (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:

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

This proceeds until a non-inlined 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 $GeneratePrompt(RecipeSchema, Quantity,$ `"2 tablespoons"`$)$.

2.2.4 Step 4: Grounding and Normalization

As a final step, all leaf nodes of the instance tree that correspond to named entities are grounded, i.e., mapped to an identifier in some existing vocabulary, ontology, or database.

Classes that represent named entities can be annotated with the different vocabularies that can supply values. Each vocabulary is identified by a unique prefix. For example, in the schema in Figure 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 [31] or AgroPortal annotator [32] 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 $IDSpaces(c)$ 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.

2.2.5 Step 5: Translation to OWL and Reasoning

The output of the previous step is 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. In other cases, we may wish to do further conversion to an ontological representation in OWL, and then perform additional reasoning steps, to check for consistency or populate missing axioms. There are multiple methods for translating to OWL, including ROBOT templates [33], DOSDPs [34], and OTTR [35].

2.3 Implementation

We provide an implementation of SPIRES in Python as part of the OntoGPT Python package (`https://github.com/monarch-initiative/ontogpt`).

SPIRES uses LinkML [24] as the Knowledge Schema language. This allows for a full representation of the necessary schema elements to implement SPIRES. In particular, LinkML has a 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 can also be used to convert schemas from forms such as SHACL [22], JSON-Schema [21], or SQL Data Definition Language into LinkML so that they can be used in SPIRES.

For parse completion, we use the *text-davinci-003* model in GPT-3 via the OpenAI API. Note that this requires the user to have an OpenAI account and sufficient credits to run the query. At the time of writing, costs are negligible for using SPIRES over small text corpora ($0.02 USD for every thousand input tokens), but running across all of PubMed would incur higher costs. Pricing models may also change at any time.

For grounding and normalization we use the Ontology Access Kit library (OAKlib) [36], which provides interfaces for multiple annotation tools (i.e., those providing links to external vocabularies and ontologies), including the Gilda entity normalization tool [27], the BioPortal annotator [37], and the Ontology Lookup Service [38]. For identifier normalization a number of services can be used, including OntoPortal mappings, with the default being the NCATS Biomedical Translator Node Normalizer [39].

The results of extraction can optionally be further processed using LinkML-OWL [40], 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 [33] to run OWL reasoning to check for logical inconsistencies and perform automated classification.

SPIRES has both a command line interface (CLI) and a simple web application (Figure 5). The CLI is implemented using the Click framework, and provides a number of sub-commands for extraction as well as preparing training sets (not currently used in the evaluation below). The web application is implemented using FastAPI, and provides an easy way to extract instances from text and visualize results. The web application can be launched by the command `web-ontogpt`. Note the web application lacks authentication, so we caution against deploying it publicly.

2.4 Evaluation Against Chemical Disease Relation Task

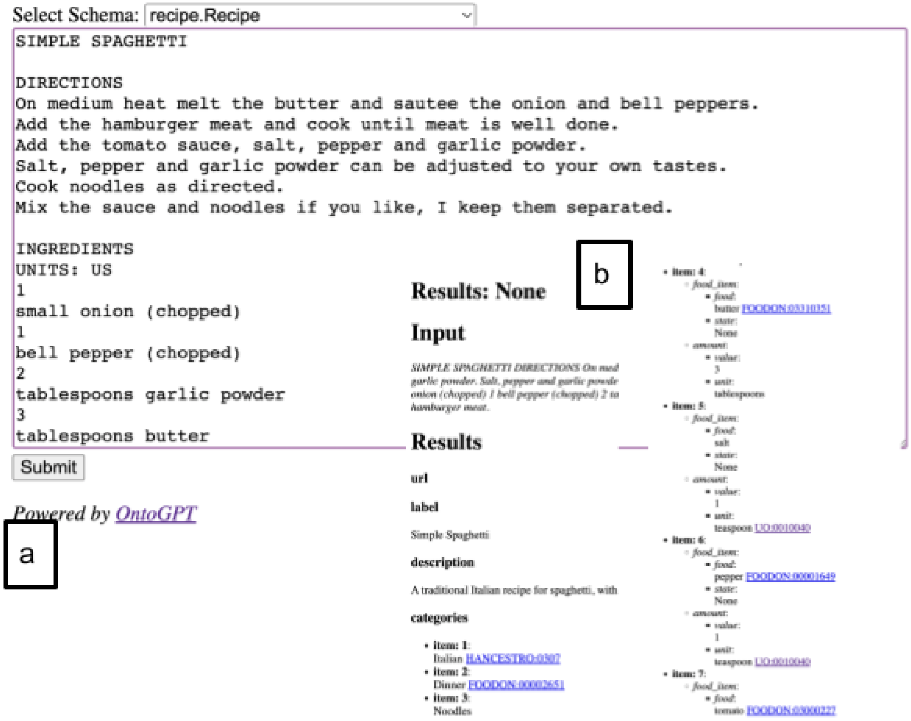
For evaluation on the Biocreative Chemical-Disease-Relation task [41], we used all 500 abstracts of the BC5CDR test set and evaluated against the set of 1066 chemical-induces-disease (CID) triples. Matching of named entities was not considered in this evaluation. For each triple, the predicate is fixed, and the subject and object are always identifiers drawn from the Medical Subject Headings (MeSH) vocabulary [42]. Grounding was performed using multiple ontologies beyond MeSH, including three resources for chemical and drug information: Chemical Entities of Biological Interest (ChEBI) [43], DrugBank [44], and MedDRA [45] (See Supplementary Table 1) for a full list of external resources used for grounding). We used the Translator NodeNormalizer [39] 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 the Biolink Model [25] as a model of chemical to disease (CTD) associations. 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 2.

3 Results

3.1 Standard Templates for Multiple Applications

The SPIRES implementation comes with ready-made schemas for multiple applications. These are primarily life-science focused, for example, deriving a pathway from a Mechanism of Action description in a database such as DrugBank. We



Select Schema: recipe.Recipe

```
SIMPLE SPAGHETTI

DIRECTIONS
On medium heat melt the butter and sautee the onion and bell peppers.
Add the hamburger meat and cook until meat is well done.
Add the tomato sauce, salt, pepper and garlic powder.
Salt, pepper and garlic powder can be adjusted to your own tastes.
Cook noodles as directed.
Mix the sauce and noodles if you like, I keep them separated.

INGREDIENTS
UNITS: US
1
small onion (chopped)
1
bell pepper (chopped)
2
tablespoons garlic powder
3
tablespoons butter
```

Powered by OntoGPT

Results: None

Input

SIMPLE SPAGHETTI DIRECTIONS On med garlic powder. Salt, pepper and garlic powde onion (chopped) 1 bell pepper (chopped) 2 ta hamburger meat.

Results

url

label

Simple Spaghetti

description

A traditional Italian recipe for spaghetti, with

categories

- **item: 1:** Italian HANCESTRO:0307
- **item: 2:** Dinner FOODON:00002651
- **item: 3:** Noodles

- **item: 4:**
  - *food_item:*
    - *food:* butter FOODON:03310351
    - state: None
  - *amount:*
    - value: 3
    - unit: tablespoons
- **item: 5:**
  - *food_item:*
    - *food:* salt
    - state: None
  - *amount:*
    - value: 1
    - unit: teaspoon UO:0010040
- **item: 6:**
  - *food_item:*
    - *food:* pepper FOODON:00001649
    - state: None
  - *amount:*
    - value: 1
    - unit: teaspoon UO:0010040
- **item: 7:**
  - *food_item:*
    - *food:* tomato FOODON:03000227

Figure 5: Screenshot of web-ontogpt. (a) Form entry page, allowing selection of schema, plus input text. (b) Sample of results as structured object rendered as nested HTML. Note that both input text and results are truncated for brevity.

also include a schema for food recipes to demonstrate general applicability in domains beyond the environmental and life sciences.

Table 1 lists the pre-made schemas included with SPIRES.

The OntoGPT repository contains examples of running SPIRES on example texts through these different schemas: `https://github.com/monarch-initiative/ontogpt/tree/main/tests/output`

3.2 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-scrapers Python model (`https://github.com/hhursev/recipe-scrapers`). 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 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 [46] to classify the results.

The results of this process are highlighted in Figure 6.

3.3 Evaluation on BioCreative Chemical Disease Relation Task

We evaluated SPIRES on the BioCreative Chemical-Disease-Relation (BC5CDR) task, as described in Methods. To demonstrate the zero-shot learning 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; after building this lexicon, the training set was discarded.

Table 1: Pre-made schemas. 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.

Schema and URI	Identifiers	Text inputs
Food Recipes https://w3id.org/ontogpt/recipe	FOODON, UO, UCUM	Unstructured and semi-structured recipes from the web
Drug mechanisms https://w3id.org/ontogpt/drugmech	MONDO, CHEBI, MESH	Mechanism of Action (MOA) descriptions, e.g from DrugBank
Multi-species signaling pathways https://w3id.org/ontogpt/gocam	GO, HGNC, UniProtKB, Reactome	Abstracts describing infection or symbiosis pathways
Chemical-disease interactions https://w3id.org/ontogpt/ctd	MESH	Abstracts describing effects of chemicals on conditions
Treatments and therapies https://w3id.org/ontogpt/treatment	MAXO, MONDO	Summaries of treatments of diseases
Biochemical reactions https://w3id.org/ontogpt/reaction	RHEA, GO, UniProt	Descriptions of biochemical reactions, their chemical participants, and the gene products that catalyze them
Metagenomic Samples https://w3id.org/ontogpt/metagenomic	ENVO	Textual descriptions of samples taken from environments, describing environmental context and parameters
Mendelian Diseases https://w3id.org/ontogpt/mendelian	MONDO, HPO, GENO	Case studies or descriptions of Mendelian diseases, including modes of inheritance, phenotypes, etc

For our CTD schema, we followed the Biolink Model [25] which extends the simple triple model of associations to also include qualifiers on the predicate, subject, and object. This yielded finer-grained predictions; for example, for a study on cromakalim and pinacidil on coronary arteries [47], SPIRES correctly parsed the statements in Table 2. The corresponding schema is outlined in Figure 7.

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.

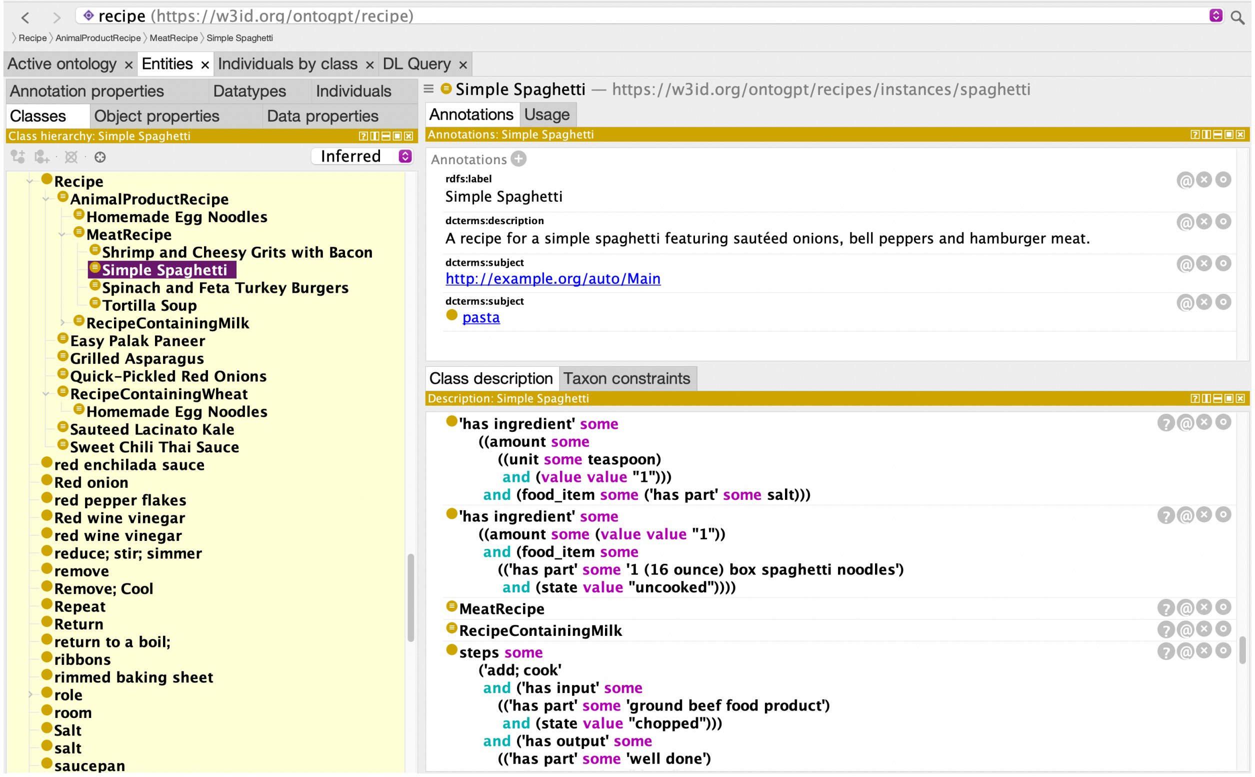
SPIRES had an F-score of 40.65, precision of 0.42, and recall of 0.38. This places it just below the average of all 18 teams that participated in the original CDR challenge. We assume that all 18 teams used the full training set, whereas with SPIRES there was no task-specific training or fine tuning. For comparison, Luo et al. report an F-score of 44.98 on BC5CDR with their biomedical domain-specific, trained-from-scratch BioGPT model [13].

The results of our experiments are available in Zenodo [50].

4 Discussion

4.1 Dependency on Centralized LLMs

Currently OntoGPT depends on the user having a subscription to the OpenAI API, which limits its uses – running OntoGPT across a large corpus would be prohibitively expensive for most users. Additionally, the use of this service introduces a dependency on a closed model with inscrutable training data, which may be plagued by biases[51]. Our experiments here were limited to GPT-3+, but it is likely that GPT-4 will yield even better results. However, dependence on even larger models increases this centralization problem, at the same time as coming with larger environmental costs.



```
Recipe
  AnimalProductRecipe
    Homemade Egg Noodles
    MeatRecipe
      Shrimp and Cheesy Grits with Bacon
      Simple Spaghetti
      Spinach and Feta Turkey Burgers
      Tortilla Soup
    RecipeContainingMilk
  Easy Palak Paneer
  Grilled Asparagus
  Quick-Pickled Red Onions
  RecipeContainingWheat
    Homemade Egg Noodles
  Sauteed Lacinato Kale
  Sweet Chili Thai Sauce
red enchilada sauce
Red onion
red pepper flakes
Red wine vinegar
red wine vinegar
reduce; stir; simmer
remove
Remove; Cool
Repeat
Return
return to a boil;
ribbons
rimmed baking sheet
role
room
Salt
salt
saucepan
```

Simple Spaghetti — https://w3id.org/ontogpt/recipes/instances/spaghetti

rdfs:label
Simple Spaghetti

dcterms:description
A recipe for a simple spaghetti featuring sautéed onions, bell peppers and hamburger meat.

dcterms:subject
http://example.org/auto/Main

dcterms:subject
pasta

```
'has ingredient' some
    ((amount some
        ((unit some teaspoon)
         and (value value "1")))
     and (food_item some ('has part' some salt)))
'has ingredient' some
    ((amount some (value value "1"))
     and (food_item some
        (('has part' some '1 (16 ounce) box spaghetti noodles')
         and (state value "uncooked"))))
MeatRecipe
RecipeContainingMilk
steps some
    ('add; cook'
     and ('has input' some
        (('has part' some 'ground beef food product')
         and (state value "chopped")))
     and ('has output' some
        (('has part' some 'well done')
```

Figure 6: Protege screenshot of extracted recipes merged OWL file), centered on the simple spaghetti recipe. The recipe is correctly classified under MeatRecipe, due to the presence of an ingredient that is classified as a meat-based product in FOODON. The right hand panel shows OWL logical axioms for the recipe, including its ingredients, and the steps involved.

Table 2: Relation extraction examples. PMID, PubMed Identifier; Qual, qualifier. PMID 2160002 is "Vasodilation of large and small coronary vessels and hypotension induced by cromakalim and pinacidil" [47]. PMID 19154241 is "Long-term lithium therapy leading to hyperparathyroidism: a case report" [48]. PMID 10327032 is "Risk of transient hyperammonemic encephalopathy in cancer patients who received continuous infusion of 5-fluorouracil with the complication of dehydration and infection" [49].

Source (PMID)	Subject	Sub. qual.	Predicate	Object	Object qual.
2160002	MESH:D019806 Cromakalim	-	INDUCES	MESH:D014664 Vasodilation	large and small coronary vessels
2160002	MESH:D020110 Pinacidil	-	INDUCES	MESH:D014664 Vasodilation	large and small coronary vessels
19154241	MESH:D008094 Lithium	Chronic	INDUCES	MESH:D006934 Hypercalcemia	-
10327032	MESH:D005472 Fluorouracil	-	INDUCES	MESH:D001927 Brain Diseases	Transient

However, recent research has shown that smaller models such as LLaMA are able to outperform models ten times their size[52], and it is possible to fine-tune these into instruction following models[53]. Furthermore, some of these such as Stanford Alpaca (`https://github.com/tatsu-lab/stanford_alpaca`) are open source.

4.2 Reliability and Hallucinations

A common problem with language models is hallucination of results (producing factually invalid statements that are not consistent with the input text) [9][51].

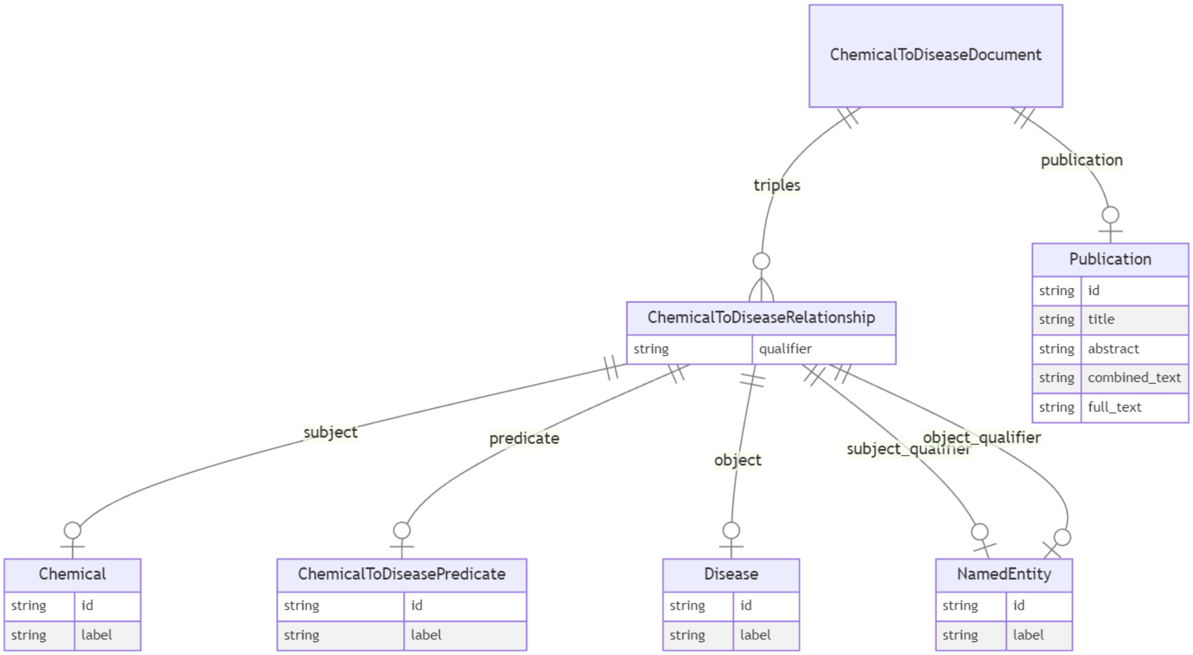


Figure 7: Chemical to Disease (CTD) schema (available from `https://w3id.org/ontogpt/ctd`).

We crafted prompts to try and avoid these as much as possible, asking only for the LM to extract what was found in the text, and keeping default low-creativity settings. On examination we found that in general hallucinations were infrequent, with most false positives and negatives attributable to incorrect relation extraction. Some apparently hallucinated results may still be technically correct: for example, suggesting *Alendronate induces Musculoskeletal pain* (MESH:D059352) whereas the test set contains the more general *Alendronate induces pain* (MESH:D010146) in a case where the nature of the pain was clearly Musculoskeletal in nature. When extracting a statement from the title "Increased frequency and severity of angio-oedema related to long-term therapy with angiotensin-converting enzyme inhibitor in two patients", the prompt completion was: "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's not clear why it does it on this occasion, and not on other ones. 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.

We envision SPIRES being used not in isolation, but rather in dual synergistic strategies, combining both human expertise and linguistic pattern recognition, deep learning and classical deductive reasoning approaches using ontologies.

5 Conclusion

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 zero-shot learning 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 highly customizable, flexible, and can be used to populate knowledge schemas across varied domains. We view SPIRES as one component of a growing toolkit of methods for transforming noisy, heterogeneous information into actionable knowledge.

References

[1] Denny Vrandečić. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pages 1063–1064, New York, NY, USA,

2012. ACM. ISBN 9781450312301. doi:10.1145/2187980.2188242.

[2] The Gene Ontology Consortium. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Res.*, 47(D1):D330–D338, January 2019. ISSN 0305-1048, 1362-4962. doi:10.1093/nar/gky1055.

[3] Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, Marija Milacic, Corina Duenas Roca, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Guilherme Viteri, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The reactome pathway knowledgebase. *Nucleic Acids Res.*, 46(D1):D649–D655, January 2018. ISSN 0305-1048, 1362-4962. doi:10.1093/nar/gkx1132.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, October 2018. doi:10.48550/arXiv.1810.04805.

[5] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. doi:10.18653/v1/d19-1250.

[6] Kate Pearce, Tiffany Zhan, Aneesh Komanduri, and Justin Zhan. A comparative study of Transformer-Based language models on extractive question answering. *arXiv*, October 2021. doi:10.48550/arXiv.2110.03142.

[7] Pedro Colon-Hernandez, Catherine Havasi, Jason Alonso, Matthew Huggins, and Cynthia Breazeal. Combining pre-trained language models and structured knowledge. *arXiv*, January 2021. doi:10.48550/arXiv.2101.12294.

[8] Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Sci. Rep.*, 12(1):1040, January 2022. ISSN 2045-2322. doi:10.1038/s41598-021-04590-0.

[9] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Wenliang Dai, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *arXiv*, February 2022. doi:10.1145/3571730.

[10] Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Trans. Assoc. Comput. Linguist.*, 8:34–48, December 2020. ISSN 2307-387X. doi:10.1162/tacl_a_00298.

[11] Mihir P Khambete, William Su, Juan C Garcia, and Marcus A Badgeley. Quantification of BERT diagnosis generalizability across medical specialties using semantic dataset distance. *AMIA Jt Summits Transl Sci Proc*, 2021:345–354, May 2021. ISSN 2153-4063. doi:10.1371/journal.pone.0112774.

[12] Yanshan Wang, Sunyang Fu, Feichen Shen, Sam Henry, Ozlem Uzuner, and Hongfang Liu. The 2019 n2c2/OHNLP track on clinical semantic textual similarity: Overview. *JMIR Med Inform*, 8(11):e23375, November 2020. ISSN 2291-9694. doi:10.2196/23375.

[13] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Brief. Bioinform.*, 23(6), November 2022. ISSN 1467-5463, 1477-4054. doi:10.1093/bib/bbac409.

[14] Christopher J Mungall, Heiko Dietze, and David Osumi-Sutherland. Use of OWL within the gene ontology. *bioRxiv*, page 010090, October 2014. doi:10.1101/010090. Accessed: 2023-3-28.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, June 2017. doi:10.48550/arXiv.1706.03762.

[16] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are Few-Shot learners. *arXiv*, May 2020. doi:10.48550/arXiv.2005.14165.

[17] Oren Ben-Kiki, Clark Evans, and Ingy Döt Net. YAML ain't markup language (YAML™) version 1.2.2. `https://yaml.org/spec/1.2.2/`, 2021. Accessed: 2023-3-28.

[18] Damion M Dooley, Emma J Griffiths, Gurinder S Gosal, Pier L Buttigieg, Robert Hoehndorf, Matthew C Lange, Lynn M Schriml, Fiona S L Brinkman, and William W L Hsiao. FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. *NPJ Sci Food*, 2:23, December 2018. ISSN 2396-8370. doi:10.1038/s41538-018-0032-6.

[19] G Schadow, C J McDonald, J G Suico, U Föhring, and T Tolxdorff. Units of measure in clinical information systems. *J. Am. Med. Inform. Assoc.*, 6(2):151–162, 1999. ISSN 1067-5027. doi:10.1136/jamia.1999.0060151.

[20] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, September 2009. ISSN 1570-8268. doi:10.1016/j.websem.2009.07.002.

[21] JSON schema. `http://json-schema.org/`, 2022. Accessed: 2023-3-28.

[22] Paolo Pareti and George Konstantinidis. A review of SHACL: From data validation to schema reasoning for RDF graphs. In *Reasoning Web. Declarative Artificial Intelligence*, pages 115–144. Springer International Publishing, 2022. doi:10.1007/978-3-030-95481-9_6.

[23] Duane Bender and Kamran Sartipi. HL7 FHIR: An agile and RESTful approach to healthcare information exchange. In *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pages 326–331, June 2013. doi:10.1109/CBMS.2013.6627810.

[24] Sierra Moxon, Harold Solbrig, Deepak Unni, Dazhi Jiao, Richard Bruskiewich, James Balhoff, Gaurav Vaidya, William Duncan, Harshad Hegde, Mark Miller, and Others. The linked data modeling language (LinkML): A General-Purpose data modeling framework grounded in Machine-Readable semantics. In *CEUR Workshop Proceedings*, volume 3073, pages 148–151, 2021.

[25] Deepak R Unni, Sierra A T Moxon, Michael Bada, Matthew Brush, Richard Bruskiewich, J Harry Caufield, Paul A Clemons, Vlado Dancik, Michel Dumontier, Karamarie Fecho, Gustavo Glusman, Jennifer J Hadlock, Nomi L Harris, Arpita Joshi, Tim Putman, Guangrong Qin, Stephen A Ramsey, Kent A Shefchek, Harold Solbrig, Karthik Soman, Anne E Thessen, Melissa A Haendel, Chris Bizon, Christopher J Mungall, Liliana Acevedo, Stanley C Ahalt, John Alden, Ahmed Alkanaq, Nada Amin, Ricardo Avila, Jim Balhoff, Sergio E Baranzini, Andrew Baumgartner, William Baumgartner, Basazin Belhu, Mackenzie Brandes, Namdi Brandon, Noel Burtt, William Byrd, Jackson Callaghan, Marco Alvarado Cano, Steven Carrell, Remzi Celebi, James Champion, Zhehuan Chen, Mei-Jan Chen, Lawrence Chung, Kevin Cohen, Tom Conlin, Dan Corkill, Maria Costanzo, Steven Cox, Andrew Crouse, Camerron Crowder, Mary E Crumbley, Cheng Dai, Vlado Dančík, Ricardo De Miranda Azevedo, Eric Deutsch, Jennifer Dougherty, Marc P Duby, Venkata Duvvuri, Stephen Edwards, Vincent Emonet, Nathaniel Fehrmann, Jason Flannick, Aleksandra M Foksinska, Vicki Gardner, Edgar Gatica, Amy Glen, Prateek Goel, Joseph Gormley, Alon Greyber, Perry Haaland, Kristina Hanspers, Kaiwen He, Kaiwen He, Jeff Henrickson, Eugene W Hinderer, Maureen Hoatlin, Andrew Hoffman, Sui Huang, Conrad Huang, Robert Hubal, Kenneth Huellas-Bruskiewicz, Forest B Huls, Lawrence Hunter, Greg Hyde, Tursynay Issabekova, Matthew Jarrell, Lindsay Jenkins, Adam Johs, Jimin Kang, Richa Kanwar, Yaphet Kebede, Keum Joo Kim, Alexandria Kluge, Michael Knowles, Ryan Koesterer, Daniel Korn, David Koslicki, Ashok Krishnamurthy, Lindsey Kvarfordt, Jay Lee, Margaret Leigh, Jason Lin, Zheng Liu, Shaopeng Liu, Chunyu Ma, Andrew Magis, Tarun Mamidi, Meisha Mandal, Michelle Mantilla, Jeffrey Massung, Denise Mauldin, Jason McClelland, Julie McMurry, Philip Mease, Luis Mendoza, Marian Mersmann, Abrar Mesbah, Matthew Might, Kenny Morton, Sandrine Muller, Arun Teja Muluka, John Osborne, Phil Owen, Michael Patton, David B Peden, R Carter Peene, Bria Persaud, Emily Pfaff, Alexander Pico, Elizabeth Pollard, Guthrie Price, Shruti Raj, Jason Reilly, Anders Riutta, Jared Roach, Ryan T Roper, Greg Rosenblatt, Irit Rubin, Sienna Rucka, Nathaniel Rudavsky-Brody, Rayn Sakaguchi, Eugene Santos, Kevin Schaper, Charles P Schmitt, Shepherd Schurman, Erik Scott, Sarah Seitanakis, Priya Sharma, Ilya Shmulevich, Manil Shrestha, Shalki Shrivastava, Meghamala Sinha, Brett Smith, Noel Southall, Nicholas Southern, Lisa Stillwell, Michael " Michi" Strasser, Andrew I Su, Casey Ta, Anne E Thessen, Jillian Tinglin, Lucas Tonstad, Thi Tran-Nguyen, Alexander Tropsha, Gaurav Vaidya, Luke Veenhuis, Adam Viola, Marcin Grotthuss, Max Wang, Patrick Wang, Paul B Watkins, Rosina Weber, Qi Wei, Chunhua Weng, Jordan Whitlock, Mark D Williams, Andrew Williams, Finn Womack, Erica Wood, Chunlei Wu, Jiwen Kevin Xin, Hao Xu, Colleen Xu, Chase Yakaboski, Yao Yao, Hong Yi, Arif Yilmaz, Marissa Zheng, Xinghua Zhou, Eric Zhou, Qian Zhu, Tom Zisk, and The Biomedical Data Translator Consortium. Biolink model: A universal schema for knowledge graphs in clinical, biomedical, and translational science. *Clin. Transl. Sci.*, June 2022. ISSN 1752-8054, 1752-8062. doi:10.1111/cts.13302.

[26] John Graybeal, Clement Jonquet, Nicola Fiore, and Mark A Musen. Adoption of BioPortal's ontology registry software: The emerging OntoPortal community. In *RDA P13 2019 - 13th Research Data Alliance Plenary Meeting*, April 2019.

[27] Benjamin M Gyori, Charles Tapley Hoyt, and Albert Steppi. Gilda: biomedical entity text normalization with machine-learned disambiguation as a service. *Bioinformatics Advances*, 2(1), January 2022. doi:10.1093/bioadv/vbac034.

[28] Lenz Furrer, Anna Jancso, Nicola Colic, and Fabio Rinaldi. OGER : hybrid multi-type entity recognition. *Journal of Cheminformatics*, 11(1), 2019. doi:10.1186/s13321-018-0326-3.

[29] Charles Tapley Hoyt, Meghan Balk, Tiffany J Callahan, Daniel Domingo-Fernández, Melissa A Haendel, Harshad B Hegde, Daniel S Himmelstein, Klas Karis, John Kunze, Tiago Lubiana, Nicolas Matentzoglu, Julie McMurry, Sierra Moxon, Christopher J Mungall, Adriano Rutz, Deepak R Unni, Egon Willighagen, Donald Winston, and Benjamin M Gyori. Unifying the identification of biomedical entities with the bioregistry. *Sci Data*, 9(1):714, November 2022. ISSN 2052-4463. doi:10.1038/s41597-022-01807-3.

[30] OpenAI. OpenAI API. `https://platform.openai.com/docs/models`, 2023. Accessed: 2023-3-27.

[31] P L Whetzel, N F Noy, N H Shah, P R Alexander, C Nyulas, T Tudorache, and M A Musen. 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.*, 39(suppl):W541–W545, July 2011. ISSN 0305-1048. doi:10.1093/nar/gkr469.

[32] Clément Jonquet, Anne Toulet, Elizabeth Arnaud, Sophie Aubin, Esther Dzalé Yeumo, Vincent Emonet, John Graybeal, Marie-Angélique Laporte, Mark A Musen, Valeria Pesce, and Pierre Larmande. AgroPortal: A vocabulary and ontology repository for agronomy. *Comput. Electron. Agric.*, 144:126–143, January 2018. ISSN 0168-1699. doi:10.1016/j.compag.2017.10.012.

[33] Rebecca C Jackson, James P Balhoff, Eric Douglass, Nomi L Harris, Christopher J Mungall, and James A Overton. ROBOT: A tool for automating ontology workflows. *BMC Bioinformatics*, 20(1):407, July 2019. ISSN 1471-2105. doi:10.1186/s12859-019-3002-3.

[34] David Osumi-Sutherland, Melanie Courtot, James P Balhoff, and Christopher Mungall. Dead simple OWL design patterns. *J. Biomed. Semantics*, 8(1):18, 2017. ISSN 2041-1480. doi:10.1186/s13326-017-0126-0.

[35] Christian Kindermann, Daniel P Lupp, Uli Sattler, and Evgenij Thorstensen. Generating ontologies from templates: A Rule-Based approach for capturing regularity. *arXiv*, page 13, 2018. doi:10.48550/arXiv.1809.10436.

[36] Chris Mungall, Harshad, Patrick Kalita, Charles Tapley Hoyt, Sujay Patil, Marcin p Joachimiak, Joe Flack, David Linke, Deepak, Sierra Moxon, Nico Matentzoglu, Vinícius de Souza, Glass, Harry Caufield, Jules Jacobsen, Justin Reese, Nomi Harris, and Shawn Tan. INCATools/ontology-access-kit: v0.2.1. `https://github.com/INCATools/ontology-access-kit`, March 2023.

[37] Clement Jonquet, Nigam H Shah, and Mark A Musen. The open biomedical annotator. *Summit Transl Bioinform*, 2009:56–60, March 2009. ISSN 2153-6430.

[38] Simon Jupp, Tony Burdett, James Malone, Catherine Leroy, Matt Pearce, Julie Mcmurry, and Helen Parkinson. A new ontology lookup service at EMBL-EBI. `http://ceur-ws.org/Vol-1546/paper_29.pdf`, 2015. Accessed: 2023-1-3.

[39] Karamarie Fecho, Anne T Thessen, Sergio E Baranzini, Chris Bizon, Jennifer J Hadlock, Sui Huang, Ryan T Roper, Noel Southall, Casey Ta, Paul B Watkins, Mark D Williams, Hao Xu, William Byrd, Vlado Dančík, Marc P Duby, Michel Dumontier, Gustavo Glusman, Nomi L Harris, Eugene W Hinderer, Greg Hyde, Adam Johs, Andrew Su, Guangrong Qin, Qian Zhu, and Biomedical Data Translator Consortium. Progress toward a universal biomedical data translator. *Clin. Transl. Sci.*, May 2022. ISSN 1752-8054, 1752-8062. doi:10.1111/cts.13301.

[40] Chris Mungall, Sujay Patil, and Nomi Harris. linkml/linkml-owl: v0.2.4. `https://zenodo.org/record/7384531`, December 2022.

[41] Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016:baw068, May 2016. ISSN 0162-4105, 1758-0463. doi:10.1093/database/baw068.

[42] C E Lipscomb. Medical subject headings (MeSH). *Bull. Med. Libr. Assoc.*, 88(3):265–266, July 2000. ISSN 0025-7338.

[43] Janna Hastings, Gareth Owen, Adriano Dekker, Marcus Ennis, Namrata Kale, Venkatesh Muthukrishnan, Steve Turner, Neil Swainston, Pedro Mendes, and Christoph Steinbeck. ChEBI in 2016: Improved services and an expanding collection of metabolites. *Nucleic Acids Res.*, 44(D1):D1214–9, January 2016. ISSN 0305-1048, 1362-4962. doi:10.1093/nar/gkv1031.

[44] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.*, 46(D1):D1074–D1082, January 2018. ISSN 0305-1048. doi:10.1093/nar/gkx1037.

[45] Elliot G Brown, Louise Wood, and Sue Wood. The medical dictionary for regulatory activities (MedDRA). *Drug Saf.*, 20(2):109–117, 1999. ISSN 0114-5916. doi:10.2165/00002018-199920020-00002.

[46] Yevgeny Kazakov and Pavel Klinov. Advancing ELK: Not only performance matters. In Diego Calvanese and Boris Konev, editors, *Proceedings of the 28th International Workshop on Description Logics (DL-15). CEUR Workshop Proceedings 2015.*, 2015.

[47] J F Giudicelli, C D la Rochelle, and A Berdeaux. Effects of cromakalim and pinacidil on large epicardial and small coronary arteries in conscious dogs. *J. Pharmacol. Exp. Ther.*, 255(2):836–842, November 1990. ISSN 0022-3565.

[48] Mian M Rizwan and Nancy D Perrier. Long-term lithium therapy leading to hyperparathyroidism: a case report. *Perspect. Psychiatr. Care*, 45(1):62–65, January 2009. ISSN 0031-5990, 1744-6163. doi:10.1111/j.1744-6163.2009.00201.x.

[49] C C Liaw, H M Wang, C H Wang, T S Yang, J S Chen, H K Chang, Y C Lin, S J Liaw, and C T Yeh. Risk of transient hyperammonemic encephalopathy in cancer patients who received continuous infusion of 5-fluorouracil with the complication of dehydration and infection. *Anticancer Drugs*, 10(3):275–281, March 1999. ISSN 0959-4973. doi:10.1097/00001813-199903000-00004.

[50] Christopher J Mungall and J Harry Caufield. Evaluation of SPIRES on Chemical-Disease-Relation extraction task 2023-01. `https://zenodo.org/record/7657763`, February 2023.

[51] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 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, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi:10.1145/3442188.3445922. URL `https://doi.org/10.1145/3442188.3445922`.

[52] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[53] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention, 2023.

[54] Nicholas Sioutos, Sherri de Coronado, Margaret W Haber, Frank W Hartel, Wen-Ling Shaiu, and Lawrence W Wright. NCI thesaurus: a semantic model integrating cancer-related clinical and molecular information. *J. Biomed. Inform.*, 40(1):30–43, February 2007. ISSN 1532-0464, 1532-0480. doi:10.1016/j.jbi.2006.02.013.

[55] Christopher J Mungall, Julie A McMurry, Sebastian Köhler, James P Balhoff, Charles Borromeo, Matthew Brush, Seth Carbon, Tom Conlin, Nathan Dunn, Mark Engelstad, Erin Foster, J P Gourdine, Julius O B Jacobsen, Dan Keith, Bryan Laraway, Suzanna E Lewis, Jeremy NguyenXuan, Kent Shefchek, Nicole Vasilevsky, Zhou Yuan, Nicole Washington, Harry Hochheiser, Tudor Groza, Damian Smedley, Peter N Robinson, and Melissa A Haendel. The monarch initiative: an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Res.*, 45(D1):D712–D722, January 2017. ISSN 0305-1048, 1362-4962. doi:10.1093/nar/gkw1128.

[56] Sebastian Köhler, Michael Gargano, Nicolas Matentzoglu, Leigh C Carmody, David Lewis-Smith, Nicole A Vasilevsky, Daniel Danis, Ganna Balagura, Gareth Baynam, Amy M Brower, Tiffany J Callahan, Christopher G Chute, Johanna L Est, Peter D Galer, Shiva Ganesan, Matthias Griese, Matthias Haimel, Julia Pazmandi, Marc Hanauer, Nomi L Harris, Michael J Hartnett, Maximilian Hastreiter, Fabian Hauck, Yongqun He, Tim Jeske, Hugh Kearney, Gerhard Kindle, Christoph Klein, Katrin Knoflach, Roland Krause, David Lagorce, Julie A McMurry, Jillian A Miller, Monica C Munoz-Torres, Rebecca L Peters, Christina K Rapp, Ana M Rath, Shahmir A Rind, Avi Z Rosenberg, Michael M Segal, Markus G Seidel, Damian Smedley, Tomer Talmy, Yarlalu Thomas, Samuel A Wiafe, Julie Xian, Zafer Yüksel, Ingo Helbig, Christopher J Mungall, Melissa A Haendel, and Peter N Robinson. The human phenotype ontology in 2021. *Nucleic Acids Res.*, 49(D1):D1207–D1217, January 2021. ISSN 0305-1048, 1362-4962. doi:10.1093/nar/gkaa1043.

[57] Lynn M Schriml, Elvira Mitraka, James Munro, Becky Tauber, Mike Schor, Lance Nickle, Victor Felix, Linda Jeng, Cynthia Bearer, Richard Lichenstein, Katharine Bisordi, Nicole Campion, Brooke Hyman, David Kurland, Connor Patrick Oates, Siobhan Kibbey, Poorna Sreekumar, Chris Le, Michelle Giglio, and Carol Greene. Human disease ontology 2018 update: classification, content and workflow expansion. *Nucleic Acids Res.*, 47(D1): D955–D962, January 2019. ISSN 0305-1048, 1362-4962. doi:10.1093/nar/gky1032.

6 Supplementary Information

Supplementary Table 1: Resources used for grounding during evaluation of SPIRES with relations in the BC5CDR test corpus. These resources were used for initial annotation and are subsequently normalized to MeSH. Annotations from the Gilda text entity normalization tool are retrieved through its API (`http://grounding.indra.bio/apidocs`) using the Ontology Access Kit.

Entity type	Resource	Prefix	Source
Chemical	Medical Subject Headings 2022	MESH	[42]
	Chemical Entities of Biological Interest	CHEBI	[43]
	National Cancer Institute Thesaurus	NCIT	[54]
	Mapping of Drug Names and MeSH 2022	MDM	[42]
	DrugBank	DRUGBANK	[44]
	Gilda	N/A	[27]
Disease	Medical Subject Headings 2022	MESH	[42]
	Mondo Disease Ontology	MONDO	[55]
	Human Phenotype Ontology	HP	[56]
	National Cancer Institute Thesaurus	NCIT	[54]
	Human Disease Ontology	DOID	[57]
	Medical Dictionary for Regulatory Activities	MEDDRA	[45]

Supplementary Table 2: MeSH identifiers used to define value sets during evaluation of SPIRES with relations in the BC5CDR test corpus. All identifiers in this table were treated as root nodes of a hierarchy, i.e., the value sets include all child MeSH terms.

Entity type	MeSH identifier	MeSH term
Chemical	D602	Amino Acids, Peptides, and Proteins
	D1685	Biological Factors
	D2241	Carbohydrates
	D4364	Pharmaceutical Preparations
	D6571	Heterocyclic Compounds
	D7287	Inorganic Chemicals
	D8055	Lipids
	D9706	Nucleic Acids, Nucleotides, and Nucleosides
	D9930	Organic Chemicals
	D11083	Polycyclic Compounds
	D13812	Therapeutics
	D19602	Food and Beverages
	D45424	Complex Mixtures
	D45762	Enzymes and Coenzymes
	D46911	Macromolecular Substances
Disease	D001423	Bacterial Infections and Mycoses
	D001523	Mental Disorders
	D002318	Cardiovascular Diseases
	D002943	Circulatory and Respiratory Physiological Phenomena
	D004066	Digestive System Diseases
	D004700	Endocrine System Diseases
	D005128	Eye Diseases
	D005261	Female Urogenital Diseases and Pregnancy Complications
	D006425	Hemic and Lymphatic Diseases
	D007154	Immune System Diseases
	D007280	Disorders of Environmental Origin
	D009057	Stomatognathic Diseases
	D009140	Musculoskeletal Diseases
	D009358	Congenital, Hereditary, and Neonatal Diseases and Abnormalities
	D009369	Neoplasms
	D009422	Nervous System Diseases
	D009750	Nutritional and Metabolic Diseases
	D009784	Occupational Diseases
	D010038	Otorhinolaryngologic Diseases
	D010272	Parasitic Diseases
	D012140	Respiratory Tract Diseases
	D013568	Pathological Conditions, Signs and Symptoms
	D014777	Virus Diseases
	D014947	Wounds and Injuries
	D017437	Skin and Connective Tissue Diseases
	D052801	Male Urogenital Diseases
	D064419	Chemically-Induced Disorders