snomedizer: R Interface to the SNOMED CT Terminology Server REST API

Peter Dutey-Magni[1] **and Anika Cawthorn**[1]

1 University College London

Summary

Electronic health record systems are expanding across the world and increasingly being used for operational planning and applied research. Electronic health records largely consist of unstructured data, which present challenges to analysts and researchers. Solutions to these challenges lie in making better use of a global standard for the representation of unstructured medical information: SNOMED CT. SNOMED CT terminology references and describes a wide range of concepts ranging from clinical anatomy to findings, procedures, and even medicines. SNOMED CT is now present in over 70% of clinical systems commercialised in Europe and North America.

`snomedizer` is an R package to interrogate a SNOMED CT terminology server. It supports operations such as: extracting attributes of a target concept; reclassifying concepts into parent concepts; building codelists; and extracting basic information from freetext information. Such operations will often be a small part of a larger analytical project. `snomedizer` is designed for non-specialists: it does not require prior knowledge of ontologies or API requests, and is developed in R, one of the most commonly used languages in healthcare analytics. Providing access to these operations directly from a familiar data wrangling environment will lower the barrier/threshold to the SNOMED CT ontology faced by non-expert users.

Background

SNOMED CT

An international project merging and expanding the College of American Pathologists' Systematized Nomenclature of Medicine (SNOMED) and the UK National Health Service's Clinical Terms Version 3 (CTV3) resulted in the first release of SNOMED CT in 2002.

SNOMED CT is a clinical terminology system: it includes a database of terms/synonyms used in health and healthcare. But it is also an ontology: it is made up of 'concepts', i.e real-world entities defined in relation to each other by 'relationships': being a subtype of another concept, or attributes, for instance a medical disorder having a particular anatomical finding site. See Figure 1 below for an example. SNOMED CT includes services for querying the ontology, complete with reasoning capability to logically infer relationships that are not already explicitly stated in SNOMED CT.

SNOMED CT concepts and relationships span an extremely wide range spanning anatomical structures, clinical findings and diagnoses, healthcare procedures, test results and their interpretation, pharmaceutical products and devices.
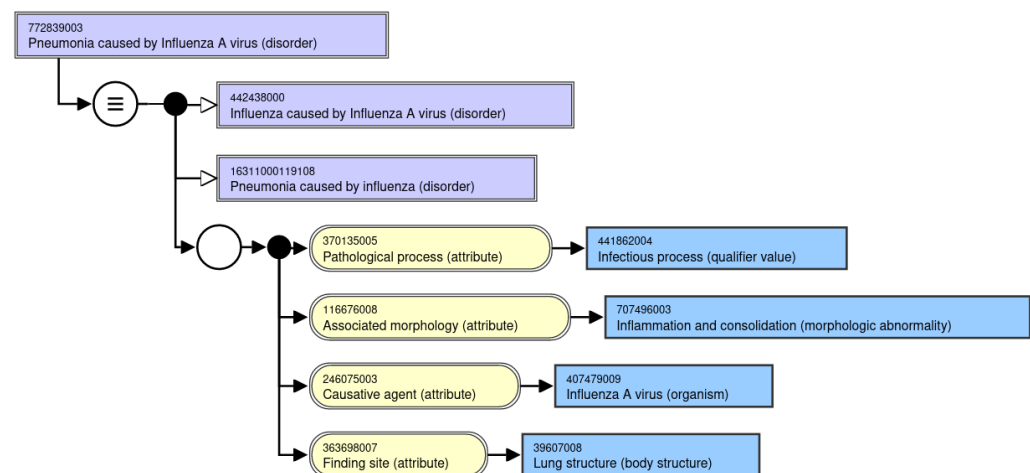
Figure 1: Diagram of stated relationships from SNOMED CT concept 772839003 | Pneumonia caused by Influenza A virus (disorder) | as produced by the SNOMED International browser

The characterisation of SNOMED CT by Bhattacharyya (2016) is as follows:

- semantically interoperable: SNOMED CT is compatible with the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001)
- polyhierarchical: a concept can be a subtype of more than just one parent concepts, as opposed to monohierarchical classifications such as the International Classification of Diseases 10th Edition (ICD-10, World Health Organisation, 1993).
- multi-lexical: one concept can be designated by one or more terms/synonyms
- multi-language support.

SNOMED CT is currently used in more than 80 countries and is present in over 70% of clinical systems commercialised in Europe and North America (SNOMED International, 2021a, 2021b). Some jurisdictions now mandate its use by healthcare providers (New Zealand Ministry of Health, 2021; NHS Digital, 2016, 2021a, 2021b).

The SNOMED CT International Edition is released twice a year by a non-for-profit international organisation named SNOMED International. It contains over 354,000 concepts at the time of writing. Member organisations of SNOMED International are responsible for publishing national extensions. For example, the UK's national release centre publishes a UK clinical extension and a UK drug extension which, together, add in the region of 30,000 concepts.

Clinical terminology services

Terminology services are the software applications used by analysts for interrogating and retrieving information from concepts, their synonyms, or their relationships. SNOMED International standards have developed dedicated terminology service specifications (SNOMED International, 2020) which are separate from (but interoperable; see SNOMED International, 2021c) global Semantic Web standards maintained by W3C (RDF, RDFS and OWL). The Expression Query Language (ECL, SNOMED International, 2021d) is the main component allowing complex queries and expression of relationships of interest. Examples are provided in section XXX.

Snowstorm (SNOMED International, 2021e) is the official implementation of these standards in the shape of an open source Java application used via an HTTP REST interface.

Snowstorm facilitates importing, authoring and maintaining multiple SNOMED CT edition in an ElasticSearch server backend. It also offers a FHIR-compliant terminology REST interface (HL7, 2019). Snowstorm is likely to remain the most relevant terminology services to end users:

- it is free and open source software (released under MIT licence) without any restrictions, meaning any healthcare organisation can build it on premise without charge. This is important, because queries retrieved via REST have the potential to disclose personal information if intercepted (for instance querying concepts related to human immunodeficiency virus infections, or looking up a code using a free-text term). Operating the service on premise, behind a firewall ensures patient data are kept confidential.
- it is actively development in parallel with SNOMED CT terminology service specifications (SNOMED International, 2020). eg. has the complete implementation of ECL
- if features a high-performance architecture based on ElasticSearch (Divya & Goyal, 2015)
- it has a wide user base, and serves as the terminology service for many other SNOMED services (such as the SNOMED International browser). This provides welcome community support, for example when facing difficulties loading particular terminology extensions.

Other terminology services exist which can store and retrieve SNOMED CT terminology, though they offer fewer functionalities:

- Ontoserver (Metke-Jimenez, Steel, Hansen, & Lawley, 2018) is a proprietary terminology service developed by the CSIRO. It supports SNOMED CT and other terminology systems. It features a FHIR terminology interface (HL7, 2019) and supports ECL queries.
- the National Library of Medicine's Unified Medical Language System (UMLS) REST API (US National Library of Medicine, n.d.) supports several terminology systems, but not ECL queries.
- OHDSI Athena (OHDSI, n.d.) is a web service which supports SNOMED CT and other terminology systems. It currently does not provide an API and is therefore limited to browsing the terminology in a web interface without complex queries.
- Other standard ontology services such as Protégé, OWL API or Apache Jena can provide terminology services but do not offer any dedicated support for SNOMED CT.

Design of snomedizer

Problem space

We considered a specific group of end users:

- healthcare analysts and health service researchers
- with data science skills foundations
- for who interrogating SNOMED CT is only a modest proportion of their analytical workload.

The adoption of SNOMED CT in healthcare analytics is slowed down by three main obstacles:

1. Build and maintenance of a on-premise terminology server is required for optimal data protection. Loading SNOMED CT into a terminology server can be complex.
2. Snowstorm is used via an HTTP REST API. Though highly interoperable, this does require personal skills and time to author suitable REST queries and handle JSON responses. Typical operations may require multiple (sequential) calls to the REST API, requiring additional programming.
3. There is a lack of hands-on training resources with real-life examples.

Typical tasks (use cases) performed by the target end users include:

- building codelists (Davé & Petersen, 2009; Watson, Nicholson, Hamilton, & Price, 2017)
- rapidly reclassifying concepts into higher-level categories (ascendants), eg `312371005 | Acute infective bronchitis (disorder) |` to `50417007 | Lower respiratory tract infection (disorder) |`
- extracting characteristics of a concept, for instance the `363698007 | Finding site (attribute) |` of a disorder, or the active

Solution

Snowstorm is designed to address obstacle (1) above. snomedizer (Dutey-Magni & Cawthorn, 2021) addresses obstacles (2) and (3) by providing easy functions to send and retrieve common queries to and from an existing Snowstorm terminology service. These functions are available directly from user's the analytical suite in the form of an R library (under GPLv3 licence).

The key requirements of snomedizer are:

- The software must not require advanced knowledge of ontology reasoning, natural language processing or software engineering.
- The software must be free and open source.
- The software must be interoperable with popular data wrangling software commonly used by the target user group. R is a leading leading health data science language along with Python (Meyer, 2019), with a strong community (https://nhsrcommunity.com/). R use has been popularised by the data wrangling library dplyr, developed library, tidy data (Wickham, 2014).
- The software must help retrieve ascendants/descendants of one or more concept.
- The software must help retrieve attributes of one or more concept.
- The software must support bespoke and complex queries created by the user.
- The software must include extensive documentation.
- The software must include tutorials which can run without building a Snowstorm endpoint. Public Snowstorm endpoints are available online for reference use only (no use in production systems) under terms of the SNOMED International SNOMED CT Browser License Agreement.
- The software must issue a warning in case of incompatibility with the Snowstorm endpoint.

snomedizer contains a direct implementation of 22 out of 50 Snowstorm API operations to meet the above requirements. The remaining 28 are for terminology authoring and maintenance and are not within scope. This is supplemented with utility functions for configuring the connection to the Snowstorm endpoint, and six wrapper functions designed to carry out common operations in a user-friendly way:

- `concept_ancestors()` and `concept_descendants()`: fetch active ancestors/descendants of one or more concepts
- `concept_descriptions()`: fetch descriptions of one or more concepts
- `concept_find()`: search SNOMED CT concepts by term, ECL query, or concept identifiers
- `concept_is()`: determine whether one or more concept are subtypes of a target set of concepts
- `concept_map()`: map SNOMED CT concepts to other terminology or code systems (using map reference sets).

All wrapper functions return results in a data frame and can handle vectors of inputs to conform with tidy data principles (Wickham, 2014). They also specify default options relevant to most users and better handle errors.

snomedizer has a companion website with all documentation and tutorials (https://snomedizer.web.app/). Every release is documented in a changelog to indicate key changes and compatibility with specific Snowstorm releases. This is in addition to utility functions, which generate user warnings if the configured endpoint is not fully compatible with the current version of snomedizer.

Examples

Setup

The latest release of snomedizer may be installed directly from its GitHub repository thanks to the devtools library:

```r
install.packages("devtools")
devtools::install_github("ramses-antibiotics/snomedizer")
```

On load, snomedizer tries to connect to the "MAIN" branch of a public Snowstorm endpoint, usually one of the endpoints maintained by SNOMED International.

```r
library(snomedizer)
```

```
## The following SNOMED CT Terminology Server has been selected:
## https://snowstorm.ihtsdotools.org/snowstorm/snomed-ct
## This server may be used for reference purposes only.
## It MUST NOT be used in production. Please refer to ?snomedizer for details.
```

The user is able to connect to another endpoint and/or branch:

```r
snomedizer_options_set(
    endpoint = "https://snowstorm.ihtsdotools.org/snowstorm/snomed-ct",
    branch = "MAIN/SNOMEDCT-US"
)
```

Basic operations

Using the `concept_find()` wrapper function, users can retrieve concepts using either an identifier or a term search. The default `limit` is set to 50 results, but a user may raise this to up to a maximum of 10,000 results.

```
concept_233604007 <- concept_find(conceptId = "233604007")
str(concept_233604007)
```

```
## 'data.frame':    1 obs. of  16 variables:
## $ conceptId      : chr "233604007"
## $ active         : logi TRUE
## $ definitionStatus: chr "FULLY_DEFINED"
## $ moduleId       : chr "900000000000207008"
## $ effectiveTime  : chr "20150131"
## $ id             : chr "233604007"
## $ idAndFsnTerm   : chr "233604007 | Pneumonia (disorder) |"
## $ fsn.term       : chr "Pneumonia (disorder)"
## $ fsn.lang       : chr "en"
## $ pt.term        : chr "Pneumonia"
## $ pt.lang        : chr "en"
## $ total          : int 1
## $ limit          : int 50
## $ offset         : int 0
## $ searchAfter    : chr "WyIyMzM2MDQwMDciXQ=="
## $ searchAfterArray: chr "233604007"
```

```
concepts_pneumo <- concept_find(term = "pneumonia", limit = 2)
```

```
## Warning:
## This server request returned just 2 of a total 601 results.
## Please increase the server `limit` to fetch all results.
```

```
str(concepts_pneumo)
```

```
## 'data.frame':    2 obs. of  16 variables:
## $ conceptId      : chr  "233604007" "161525004"
## $ active         : logi  TRUE TRUE
## $ definitionStatus: chr  "FULLY_DEFINED" "FULLY_DEFINED"
## $ moduleId       : chr  "900000000000207008" "900000000000207008"
## $ effectiveTime  : chr  "20150131" "20040731"
## $ id             : chr  "233604007" "161525004"
## $ idAndFsnTerm   : chr  "233604007 | Pneumonia (disorder) |" "161525004 | His
## $ fsn.term       : chr  "Pneumonia (disorder)" "History of pneumonia (situati
## $ fsn.lang       : chr  "en" "en"
## $ pt.term        : chr  "Pneumonia" "H/O: pneumonia"
## $ pt.lang        : chr  "en" "en"
## $ total          : int  601 601
## $ limit          : int  2 2
## $ offset         : int  0 0
## $ searchAfter    : chr  "WzE2MTUyNTAwNF0=" "WzE2MTUyNTAwNF0="
## $ searchAfterArray: int  161525004 161525004
```

Wrapper functions can handle more than a single concept:

```
dm_and_pneumo <- concept_descendants(conceptId = c("233604007", "73211009"))
```

```
## Warning:
## This server request returned just 50 of a total 228 results.
## Please increase the server `limit` to fetch all results.


## Warning:
## This server request returned just 50 of a total 118 results.
## Please increase the server `limit` to fetch all results.
```

```r
str(dm_and_pneumo[["233604007"]])
```

```
## 'data.frame':    50 obs. of  16 variables:
##  $ conceptId      : chr  "882784691000119100" "10625751000119106" "10625711000
##  $ active         : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ definitionStatus: chr  "FULLY_DEFINED" "FULLY_DEFINED" "FULLY_DEFINED" "FULL
##  $ moduleId       : chr  "900000000000207008" "900000000000207008" "9000000000
##  $ effectiveTime  : chr  "20200731" "20150731" "20150731" "20150731" ...
##  $ id             : chr  "882784691000119100" "10625751000119106" "10625711000
##  $ idAndFsnTerm   : chr  "882784691000119100 | Pneumonia caused by Severe acut
##  $ fsn.term       : chr  "Pneumonia caused by Severe acute respiratory syndrom
##  $ fsn.lang       : chr  "en" "en" "en" "en" ...
##  $ pt.term        : chr  "Pneumonia caused by SARS-CoV-2" "Bronchopneumonia du
##  $ pt.lang        : chr  "en" "en" "en" "en" ...
##  $ total          : int  228 228 228 228 228 228 228 228 228 228 ...
##  $ limit          : int  50 50 50 50 50 50 50 50 50 50 ...
##  $ offset         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ searchAfter    : chr  "WzcyNDQ5ODAwNF0=" "WzcyNDQ5ODAwNF0=" "WzcyNDQ5ODAwNF
##  $ searchAfterArray: int  724498004 724498004 724498004 724498004 724498004 724
```

```r
str(dm_and_pneumo[["73211009"]])
```

```
## 'data.frame':    50 obs. of  16 variables:
##  $ conceptId      : chr  "530558861000132104" "10754881000119104" "10753491000
##  $ active         : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ definitionStatus: chr  "PRIMITIVE" "PRIMITIVE" "PRIMITIVE" "FULLY_DEFINED" .
##  $ moduleId       : chr  "900000000000207008" "900000000000207008" "9000000000
##  $ effectiveTime  : chr  "20170131" "20140731" "20140731" "20180131" ...
##  $ id             : chr  "530558861000132104" "10754881000119104" "10753491000
##  $ idAndFsnTerm   : chr  "530558861000132104 | Atypical diabetes mellitus (dis
##  $ fsn.term       : chr  "Atypical diabetes mellitus (disorder)" "Diabetes mel
##  $ fsn.lang       : chr  "en" "en" "en" "en" ...
##  $ pt.term        : chr  "Atypical diabetes mellitus" "Diabetes mellitus in mo
##  $ pt.lang        : chr  "en" "en" "en" "en" ...
##  $ total          : int  118 118 118 118 118 118 118 118 118 118 ...
##  $ limit          : int  50 50 50 50 50 50 50 50 50 50 ...
##  $ offset         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ searchAfter    : chr  "WzYwOTU2MzAwOF0=" "WzYwOTU2MzAwOF0=" "WzYwOTU2MzAwOF
##  $ searchAfterArray: int  609563008 609563008 609563008 609563008 609563008 609
```

Functions from the dplyr library such as `select()`, `mutate()`, `transmute()`, `filter()` and `arrange()` can be chained to wrapper functions using the `%>%` piping operator

```
library(dplyr)
concept_ancestors(conceptId = "233604007") %>%
   .[["233604007"]] %>%
   select(conceptId, fsn.term) %>%
   filter(grepl("finding", fsn.term))
```

```
##    conceptId                                   fsn.term
## 1  609623002          Finding of upper trunk (finding)
## 2  406123005        Viscus structure finding (finding)
## 3  404684003                 Clinical finding (finding)
## 4  302292003      Finding of trunk structure (finding)
## 5  301230006                     Lung finding (finding)
## 6  301226008 Lower respiratory tract finding (finding)
## 7  298705000    Finding of region of thorax (finding)
## 8  106048009              Respiratory finding (finding)
```

Functions comply with the tidy data principles (Wickham, 2014). For instance, the `concept_is()` function can be used within the `mutate()` call to create a new column in a data frame indicating whether concepts listed in a given column are bacterial infections (as opposed to viral, or unspecified infection types).

```
pneumonia_type <- dm_and_pneumo[["233604007"]] %>%
   select(conceptId, pt.term) %>%
   mutate(bacterial = concept_is(
      concept_ids = conceptId,
      target_ecl = "87628006 | Bacterial infectious disease (disorder) |"
   ))

# printing the first 5 concepts
head(pneumonia_type[, c("pt.term", "bacterial")], 5)
```

```
##                                         pt.term bacterial
## 1                 Pneumonia caused by SARS-CoV-2     FALSE
## 2                    Bronchopneumonia due to virus     FALSE
## 3 Bronchopneumonia due to Streptococcus pneumoniae      TRUE
## 4         Bronchopneumonia due to Streptococcus      TRUE
## 5    Bronchopneumonia due to Staphylococcus aureus      TRUE
```

Advanced operations

More advanced operations may be carried out using SNOMED CT's Expression Constraint Language (SNOMED International, 2021d). ECL consists in simple queries facilitated by operators listed in the ECL quick reference table.

For instance, one can use the SNOMED CT ontology (with full inference) to search all infections that can be caused by bacteria belonging to `106544002 | Family Enterobacteriaceae (organism) |`:

1. the complete set of infection concepts is defined as concepts descending from `40733004 | Infectious disease (disorder) |`, using the "descendant or self" ECL operator notated `<<`

2. the complete set of concepts caused by bacteria of family *Enterobacteriaceae* is search with the attribute contraint `246075003 |Causative agent| = <<106544002 | Family Enterobacteriaceae (organism) |` using an equality operator
3. constraints (1.) and (2.) can be chained with the "refine" ECL operator notated `:`.

This query shows *Enterobacteriaceae* bacteria are associated with infections affecting a range of body systems, including the respiratory, urinary, and digestive systems.

```
enterobac_infections <- concept_find(
  ecl = "<<40733004 | Infectious disease (disorder) | :
           246075003 |Causative agent| = <<106544002",
  limit = 10000
)

# print the names of the first 10 concepts
head(enterobac_infections$pt.term, 10)
```

```
## [1] "Bronchopneumonia due to Klebsiella pneumoniae"
## [2] "Bronchopneumonia due to Escherichia coli"
## [3] "Salmonella pyelonephritis"
## [4] "Urinary tract infection caused by Klebsiella"
## [5] "Infection due to Shiga toxin producing Escherichia coli"
## [6] "Infection due to Escherichia coli O157"
## [7] "Septic shock co-occurrent with acute organ dysfunction due to Serratia"
## [8] "Sepsis without acute organ dysfunction caused by Serratia species"
## [9] "Enteritis of small intestine caused by enterotoxigenic Escherichia coli"
## [10] "Pneumonia caused by Enterobacter"
```

Future developments

The first major release (1.0.0) of snomedizer focused on implementing API operations of Snowstorm version 7.X.X (UPDATE!!). Bug reports and user requests are welcome and can be submitted on the package repository (https://github.com/ramses-antibiotics/snomedizer/issues/).

Further developments are considered based on user feedback. Possible future features include:

- functions to query the ElasticSearch service directly, rather than via the Snowstorm REST API. This would facilitate term searches and natural language processing. This feature is not currently implemented because requires that this service is published to the end user, which is not the case on public servers.
- supporting the FHIR terminology server (HL7, 2019), which is provided since Snowstorm release 4.10.2.

Acknowledgements

References

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), 34–43. Retrieved from https://web.archive.org/web/20171010210556/https://pdfs.semanticscholar.org/566c/1c6bd366b4c9e07fc37eb372771690d5ba31.pdf

Bhattacharyya, S. B. (2016). *Introduction to SNOMED CT*. Springer. Retrieved from https://doi.org/10.1007/978-981-287-895-3_7

Davé, S., & Petersen, I. (2009). Creating medical and drug code lists to identify cases in primary care databases. *Pharmacoepidemiology and Drug Safety*, *18*(8), 704–707. doi:https://doi.org/10.1002/pds.1770

Divya, M. S., & Goyal, S. K. (2015). ElasticSearch: An advanced and quick search technique to handle voluminous data. *COMPUSOFT: An International Journal of Advanced Computer Technology*, *2*(6), 171–175.

Dutey-Magni, P., & Cawthorn, A. (2021). *snomedizer: R Interface to the SNOMED CT Terminology Server REST API*. Zenodo. Retrieved from https://doi.org/10.5281/zenodo.5705568

HL7. (2019). FHIR Terminology Service Specification (v4.0.1: R4 - Mixed Normative and STU). Retrieved from http://hl7.org/fhir/R4/terminology-service.html

Metke-Jimenez, A., Steel, J., Hansen, D., & Lawley, M. (2018). OntoServer: A syndicated terminology server. *Journal of Biomedical Semantics*, *9*(1), 24. doi:10.1186/s13326-018-0191-z

Meyer, M. A. (2019). Healthcare data scientist qualifications, skills, and job focus: a content analysis of job postings. *Journal of the American Medical Informatics Association*, *26*(5), 383–391. doi:10.1093/jamia/ocy181

New Zealand Ministry of Health. (2021). *HISO 10048 emergency care data standard*. Retrieved from https://www.health.govt.nz/publication/hiso-10048-emergency-care-data-standard

NHS Digital. (2016). *SCCI0034 SNOMED CT Requirements Specifications Amd 35/2016*. Retrieved from https://digital.nhs.uk/data-and-information/information-standards/information-standards-and-data-collections-including-extractions/publications-and-notifications/standards-and-collections/scci0034-snomed-ct

NHS Digital. (2021a). *DAPB4013: Medicine and Allergy/Intolerance Data Transfer Requirements Specification (Amd 5/2021)*. Retrieved from https://digital.nhs.uk/data-and-information/information-standards/information-standards-and-data-collections-including/publications-and-notifications/standards-and-collections/dapb4013-medicine-and-allergy-intoleran

NHS Digital. (2021b). *DAPB4017: Pathology Test and Results Standard Specification (Amd 44/2020)*. Retrieved from https://digital.nhs.uk/data-and-information/information-standards/information-standards-and-data-collections-including-extractions/publications-and-notifications/standards-and-collections/dapb4017-pathology-test-and-results-sta

OHDSI. (n.d.). *Athena Vocabularies Repository*. Retrieved from https://github.com/OHDSI/Athena

SNOMED International. (2020). *SNOMED CT Terminology Services Guide. Publication date: 2020-09-30*. Retrieved from http://snomed.org/tsg

SNOMED International. (2021a). *2020 Annual Report*. Retrieved from https://www.paperturn-view.com/?pid=MTY165474

SNOMED International. (2021b). *SNOMED CT: Articulating Stakeholder Value*. Retrieved from https://www.paperturn-view.com/?pid=MTU155774

SNOMED International. (2021c). *SNOMED CT OWL Guide. Publication date: 2021-08-02.* Retrieved from http://snomed.org/owl

SNOMED International. (2021d). *SNOMED CT Expression Constraint Language Specification and Guide. Version 1.6. Publication date: 2021-10-05.* Retrieved from http://snomed.org/ecl

SNOMED International. (2021e). *Snowstorm: SNOMED CT Terminology Server Using Elasticsearch.* Retrieved from https://github.com/IHTSDO/snowstorm

US National Library of Medicine. (n.d.). UMLS REST API. Retrieved from https://documentation.uts.nlm.nih.gov/rest/home.html

Watson, J., Nicholson, B. D., Hamilton, W., & Price, S. (2017). Identifying clinical features in primary care electronic health record studies: Methods for codelist development. *BMJ Open*, *7*(11). doi:10.1136/bmjopen-2017-019637

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(10), 1–23. doi:10.18637/jss.v059.i10

World Health Organisation. (1993). *The ICD-10 Classification of Mental and Behavioural Disorders. Clinical descriptions and diagnostic guidelines.* Geneva: World Health Organisation. Retrieved from https://www.who.int/classifications/icd/en/bluebook.pdf