Annotation of clinical datasets using composite information models

Aleksandar Zivaljevic

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) in Bioengineering, the University of Auckland, 2021.

Abstract

Annotation of clinical datasets is a process of discovery of clinical concepts and their association with machine readable constructs. The purpose of the process is to expose clinical information held in the datasets to automated agents in a structured format that facilitates and accelerates processes like document comparison, search or decision support.

Numerous annotation techniques currently exist and techniques vary from simple matching to words from a reference dictionary to complex supervised machine learning techniques. The results vary, there is no gold standard and the community constantly produces new techniques.

We developed an annotation technique that utilises ontological features of SNOMED-CT in discovering complex concepts contained in clinical documents and transforming them using term expansion into composite information models that reveal semantics of the messages conveyed in the document.

Our algorithm converts clinical documents into segments of ontology of reality and weights the concepts found based on their frequency and location in the expanded structure. The algorithm utilises term expansion methods and pre-defined weights and compares the similarities of the resulting structures by measuring distances between their *x* highest weight-bearing concepts based on the position of those concepts in the ontology of reality's graph structure represented by SNOMED CT.

We test the effectiveness of our algorithm by comparing the outputs of the headers and bodies of clinical discharge summaries. We use a machine agent to measure similarity between the composite information models discovered in the body and in the diagnosis section of the matched entity. The assumption that the model derived from the body of the discharge summary will be similar to the model derived from the diagnosis section has been tested and the results show that our algorithm is a valid solution for comparing and finding similarities between clinical documents. The results show lower distances between the concepts in the related bodies of texts compared to the distances between the concepts in the unrelated bodies of text.

To further test the application of the algorithm that we created and our novel approach, we explore the utility of the algorithm in comparing openEHR Archetypes with clinical

documents. Our results confirm that our approach is a step to the right direction as the results are promising.

Our method utilises SNOMED CT as an ontology of reality for expansion of clinical concepts found in the matched entity and introduces a novel technique of weighting that takes into consideration relations between entities, their position in the ontology of reality and their frequency in the clinical document text. As ontologies are quickly becoming content rich representations of reality, we consider it important to establish their utility as ontologies of reality in annotation processes. The method we developed is an alternative to the currently used search expansion methods and corpus annotation methods to name a few.

Dedication

To my son. May this accomplishment inspire his educational journey and his personal development.

Acknowledgements

I owe my deepest gratitude to my PhD mentors Professor Jim Warren, Dr Koray Atalag and Dr David Phillip Nickerson. In helping me complete this exciting and enjoyable project, they have shown not just technical expertise, but also ability to be patient and tactical with a student who, at times had life commitments to put his full focus on. I can't but single out Professor Jim Warren for his professionalism and dedication to ensuring student's success. Many thanks to the University of Auckland, I am utterly grateful for the facilities provided, including, but not limited to the Library and access to the NECTAR Cloud.

Contents

1 Preamble - Design Science Research (DSR)

The proposed research will be conducted as Design Science Research (DSR) as in Hevner et al. (2004) and structured as suggested by Johannesson & Perjons (2014). DSR is not meant to just describe, explain and predict, it is also meant to change the world by improving it. It is focused on improving practices, defined as sets of human activities, by resolving practical problems, defined as an undesirable state of affairs or gaps between the current and a desired state. Practical problems are not just denoting troublesome situations, but opportunities for improvement as well. DSR produces artefacts that are used to address particular problems of general interest. Artefacts are not just physical objects, but intangibles as well, including algorithms, information models, methods and guidelines among others. Artefacts and knowledge about artefacts are final outputs of the DSR process.

DSR prescribes a method framework for design science research that further prescribes a set of activities that are suggested to be undertaken as part of the research process. The activities are 1) Explicate problem, 2) Define requirements, 3) Design and develop artefact, 4) Demonstrate artefact and 5) Evaluate artefact.

Explicating a problem includes investigation and analysis of a problem and its underlying issues. The outcome of the Explicate problem activity is a justified and well formulated problem. Defining requirements is about transformation of the problem into demands of the proposed artefact. The outcome is a well-defined set of artefact characteristics. The Design and develop artefact activity outputs a fully functional artefact that is shown in the Demonstrate artefact activity as a proof of concept that is then evaluated in the last activity of the process.

No constraints are prescribed as to the use of research methods and research strategies in any of the activities. It is acceptable that different research strategies are utilised in different research activities that are undertaken as part of the DSR. It is also acceptable that different methods are used for data collection in different activities. As an example, activities 1) and 2) can be based on document research and activity 3) can be based on experiments conducted as part of the Action Research strategy.

Moreover, not all activities need to be undertaken in one DSR project. Some research projects will place more focus on one or two of the activities and will either skip or only lightly address others. The example would be a research project XYZ whose focus is on explicating a problem by carrying out a root cause analysis. The XYZ study would, rather than conducting all 5

activities, conduct activity 1) and possibly activity 2) and then not conduct activities 3), 4) and 5). Other studies that might be building on the XYZ study would focus on activities 3), 4) and 5) and lightly on activity 2) since the problem has been explicated already, and there will be no need for further focus on activity 1).

This research will deliver an algorithm as an artefact and all 5 activities of the DSR process will be undertaken. Activity 1) will be used to elaborate on the problem that the artefact will solve. To be able to do so, definitions of the relevant concepts by other authors will be given before use of these concepts is touched upon. Most of the phase 1) will be covered in the section 2 Explicate problem (DSR activity 1).

DSR activity 2) will define and elaborate on the requirements of the artefact that will be created in this project. Some of the requirements are not direct requirements of the artefact, but are actual prerequisites ensuring that methods used by the artefact in producing its output can be executed. The example of a direct requirement of the artefact is "The algorithm needs to be able to expand clinical concepts using SNOMED CT as the ontology of reality" and the example of a prerequisite for that is that "A method for converting SNOMED CT into a graph that can be traversed by available computational systems in realistic time is developed". Other examples are tasks that enabled caching of expansion results as well as the experiment that we conducted to test coverage of SNOMED CT to confirm its viability as an ontology of reality. Despite not being directly related to the goal of the project, the prerequisite tasks are of high importance because certain steps in the project cannot be completed without them.

Activity 3) in this project depicts not just the design of the artefact but also the details of the experiment where we tested coverage of SNOMED CT to ensure its viability as an ontology of reality. The design of the artefact is presented in detail including all the stages of information transformation from unstructured free text format to the structured graph-based format ready for the final step of comparison.

The artefact is demonstrated as an application developed in Microsoft Visual Studio in the C# programming language. That application is elaborated on in the chapter that fulfils the requirements of the activity 4).

The chapter that covers the final evaluation as prescribed by activity 5) of DSR contains the details of the experiment conducted to confirm the utility of the artefact and therefore to provide

confirmation for the assumptions made at the beginning of the project as well as to confirm that the goal of the project has been achieved.

The following chapter will provide background on the problem that prompted this research and will suggest the artefact as a start of the process towards finding the solution to that problem.

2 Explicate problem (DSR activity 1)

DSR instructs that the DSR activity 1 - Explicate Problem is about investigating and analysing a practical problem (Johannesson & Perjons, 2014, p. 76). We define the problem in the section 2.1 Introduction, and support it with the discussion in the section 2.2 Background. The problem is explicated as an issue of use of composite information models as annotation artefacts for the purpose of revealing messages, rather than just concepts, in clinical free text documents. The discussion covers current methods used in annotation of clinical documents and touches upon rule-based methods, machine learning and deep learning as well as methods common for pre-processing stage of each of the techniques. Three rule-based methods will be reviewed at the end before a short conclusion on rule-based methods is given.

2.1 Introduction

Clinical data, a foundation of clinical information, is a crucial factor in decision making processes in the domain of clinical practice. Data are acquired and then observed and interpreted by physicians to form the foundation for making conclusions in the process of care. Clinical data is made of inputs captured for variety of purposes and from variety of sources and range from determinants of health and measures of health and health status to documentation of care delivery (IOM et al., 2010).

Recognition of the value of clinical data is not new. The first verdicts on the value of data collected through patient observation as opposed to the unspecific philosophical postulates is given in the early works of Greek philosophers in the body of work called Hippocratic Corpus. The work “On Ancient Medicine”, arguably authored by Hippocrates, highlights the value of data collected using empirical and observation-based approaches over the philosophical approach that is based on generic knowledge, “empty postulates” and incursion of supernatural in treatment of conditions that can be explained using purely natural phenomena (Roth, 2008).

The first time the process of collection and recording of clinical data is documented in a written work was in the piece from the aforementioned Hippocratic Corpus titled “Epidemics”. The authors, reporting on epidemics and infectious diseases cases they encountered during their travel around the Greece, have documented their observations producing the first clinical record that many authors consider a “hallmark of clinical observation” (Roth, 2008, p. 79).

Galen, arguably the most famous Greek physician after Hippocrates, has made an important contribution to the discipline of Medicine by making available clinical data from the results of experiments and interventions he conducted (Rockland et al., 1998). Although there is a lot of debate around usefulness of the theory he produced (Avorn, 2008, p. 29), clinical data he left behind were repeatedly quoted by many in centuries to come, including Vesalius in the 16th and Thomas Willis in the 17th century (Rockland et al., 1998, p. 10).

Clinical documents carry information that is not just relevant at the point of care but as a secondary use information too. Information extracted from clinical documents like discharge summaries can be used in the processes of improving quality of patient care, medical research, decision making, for supporting population health statistics and care planning among other uses. Secondary use of clinical information is known to "enhance healthcare experiences for individuals, expand knowledge about disease and appropriate treatments, strengthen understanding about the effectiveness and efficiency of our healthcare systems, support public health and security goals, and aid businesses in meeting the needs of their customers" (Botsis et al., 2010, p. 1).

However, for the full potential of clinical information to be released, it needs to be put in a format that can be processed by automated agents and concepts and messages found in the data need to made comparable. The former is achieved through use of interoperability standards like ISO, openEHR and HL7 (Schloeffel et al., 2006). The latter is achieved through annotation.

Wilcock (2009a) defines annotation as process of attaching notes to an object for the purpose of providing more information on the object. Nagao (2003b) extends that, asserting that it is reasonable to generalise that any kind of content, not just notes can be used for this purpose. He adds that annotation content can annotate only part of the object and still be considered a valid annotation. In general, most of the authors call content added through annotation process "metadata" (Wilcock, 2009a; Passant, 2010; Pustejovsky & Stubbs, 2012a; Nagao, 2003b).

Annotation techniques associate concepts in annotated datasets to concepts that are understandable to the machine agent, including concepts found in ontologies (Al-Mubaid & Nguyen, 2009; Hsu et al., 2012; Jonquet et al., 2008; Papatheodorou et al., 2009; Roberts et al., 2007), taxonomies (W. Lee et al., 2007) or custom made data structures (Cimino & Barnett, 1990) among others. However, most of these annotation methods annotate only individual fields of the dataset, rather than recognise and annotate composite clinical information that is

usually of higher relevance to a project. For instance, a method annotating clinical text containing mention of concepts systolic and diastolic will recognise the fields and annotate them with SNOMED-CT codes SCTID: 271649006, Systolic blood pressure (observable entity) and SCTID: 271650006, Diastolic blood pressure (observable entity) respectively but will not recognise them as blood pressure concept.

We argue that annotation needs to take into consideration composite concepts in a clinical text, not just single concepts. We believe that recognising documents' messages, rather than concepts is crucial for document comparison and effective search among other processes.

However, these messages are not easy to recognise, not just because of their complexity, but because a single snapshot of a reality can be described by multitude of usually semantically similar, but still morphologically different messages. The messages are representations of a producing agent's interpretation of own idea/perception, meaning that a message constructed by one agent is likely to be different from the message created by a different agent that observes/considers the same phenomenon. In the case of clinical documents, the uniqueness of the producing agent's representation of the reality can be due to uniqueness of the agent's understanding and knowledge of the medical domain. Likewise, the agents reading these clinical documents will have their own interpretation of what is presented in them for all the same reasons.

Grace (2016, pp. 4–5) attributes differences between messages describing the same segment of reality to the language used by the producing agents. The crux of the Grace's position is that agents' epistemic base, including agent's assumptions of the nature of the world, culture and access to the knowledge of the world impacts the language used in describing the reality at the individual level. Although the issues of language are commonly seen as impacting one's representation of reality, language is certainly not the only layer observed. For example, Aristotle (as cited by Hudry, 2011) adds mental conception and Shore (1998) adds culture as additional layers that alter message structure before the meaning is finally formatted by the language.

The phenomenon that different agents can form different messages when communicating the same snapshot of the reality makes it difficult for the outputs to be compared and decided on their similarities and differences. This phenomenon particularly affects ability of automated

agents to discerning true meanings of messages put before them. That further makes processes like search through and classification of bodies of texts difficult and ineffective.

To overcome issues of semantic differences between messages, we propose that transposing the elements of a message into their roots will reveal the very meaning of the message and allow for comparison. We see medical documents as messages constructed from signs, therefore we see signs as elements of a message. Considering that signs are individual's interpretations of particulars found in the reality, we see particulars as signs' roots. Furthermore, we believe that if the signs are transposed back into particulars, or as close as possible to the particulars they represent, the messages will be comparable and their similarity measurable.

As particulars found in the messages have their own position in the ontology of reality, their interrelationships and distances from the other elements will be known, hence the final result of the process of transposing of message signs to particulars will be an information model whose elements' distances from other information models' elements will be known (or measurable).

We believe that calculating similarity of two such information models will be possible to be achieved by measuring distance between two information models. This is based on the assumption that proximity of sections of the ontology of reality is a function of their similarity and the more proximal the segments are, the more similar they are.

This work will create an artefact, an algorithm, that will annotate medical documents using complex information models so that medical documents can be positioned in an ontology of reality and utilised as such for comparison to other medical documents, search queries and similar. Annotation process will firstly extract messages from medical documents by converting signs found in these documents into particulars. It will than create complex information models from these particulars, in other words create annotation artefacts, and develop a method that will compare these information models by positioning them in the ontology of reality and measuring their distances.

We will utilise graphs as structures of information models we create in the process of annotation. Graph methods will be used to calculate distances of concepts in the ontology of reality. Due to its comprehensiveness, coverage and ontological structure, SNOMED CT is utilised as the ontology of reality.

We start the next section by defining and explaining our understanding of the construct of annotation. Then we touch upon the methods used in pre-processing of information before we elaborate on the methods of machine and deep learning. Finally, we talk about rule-based methods and itemise and describe some of the well-known methods in that group.

2.2 Background

2.2.1 Annotation

Annotated corpora are produced through the process of annotation. For us, annotation is association of a token with one or a group of codes representing predefined, unique concepts. For clinical texts, annotation is identification of tokens that represent medical concepts and their association with codes defined in one of the medical terminologies. The token can be a word, phrase, sentence, paragraph or a document. The outcome of an annotation process is a token labelled with one or more codes that are directly recognisable by automated agents and can be unambiguously converted into words or phrases that can be understood by human agents.

We base our understanding on Wilcock (2009b), who defines annotation as a process of attaching notes to an object for the purpose of providing more information on the object. We extend that definition with the assertions of Nagao (2003a) that any kind of content, not just notes, can be used for annotation and that annotation content can annotate only part of the object and still be considered a valid annotation.

Named entity recognition (Boag et al., 2018; Goeuriot et al., 2020; Wu et al., 2018) and concept normalisation (Luo et al., 2019; Pradhan et al., 2015) are terms describing similar processes to what we call annotation in this work. Named entity recognition is defined as identifying and locating of concepts and their categories in the body of text, while concept normalisation is seen as named entity recognition plus normalisation of recognised entities. Normalisation of recognised entities includes relating each of the recognised entities to a code in one of the clinical terminologies. Our work is in alignment to both named entity recognition and concept normalisation, but we go several steps further and expand recognised concepts before creating

graphs as annotation artefacts. Due to that fact and considering that there is no firm consensus on use of these terms, we use the term annotation in this work.

The most commonly annotated tokens in annotated corpora are words and phrases. Such annotations reveal diseases, anatomic parts and physiological elements among others and provide detailed and specific information on the entities found (e.g. Bada et al., 2012). Annotation of a sentence, paragraph and the document type tokens usually takes into consideration inferred meaning of the entire token (e.g. Wilbur et al., 2006). The annotation of paragraphs and sentences is utilised in identifying sections of the document (Liakata et al., 2010) and annotation of documents as a whole was seen as useful in classification of documents (e.g. Baker et al., 2016). In this work, we focus on annotation of documents.

2.2.1.1 Automated Vs manual annotation

Annotation process can be conducted by either a human or automated agents or a combination of the two. Annotations conducted by human agents (manual annotations) are considered a bottleneck in the overall process of utilisation of clinical data due to the time and resource requirements. The process of manual annotation usually involves training of experts, definition of annotation protocols and manual pre-processing of the corpus that is to be annotated (Pustejovsky & Stubbs, 2012b). The costs incurred during the process are high and affordability is seen as a major barrier of a manual annotation process (Velupillai et al., 2015).

In 2015 we conducted a series of case studies reviewing the utility of manual mapping in biomedical projects associated with the Virtual Physiological Human project (Zivaljevic et al., 2015b). We found that manual mapping of concepts found in clinical datasets impacted most significantly on the project timeframes, costs and validity of the reviewed projects. Elaborating on the negative effects of mapping, the authors of one of the project, Sittig et al. (2012), assert that "Design and development of these 'mapping' applications is one of the biggest challenges in any multi-institutional research project, because it is often the case that different organizations refer to the same activity, condition, or even procedure by different names, and the same names can refer to different things across institutions". Mapping applications are manually conducted mapping activities that required experts whose time was precious and expensive.

In other research, Zasada et al. (2012) reached a similar conclusion where they comment on the process of changing data to match the prescribed format (the process they call curation) saying that "the curation stage can be quite labour intensive". Similarly, Shi et al. (2011) report that manual mapping of the concepts used in variables resulted in them limiting the number of used variables. Due to that limitation, a limited number of hemodynamic variables are used in their calculations and as a result the authors state that their solution is based on a "rather narrow physiological envelope". They suggest that the "whole envelope of simulations should be performed under a range of physiological states" (p. 335) implying that a process more effective than manual mapping is required.

Another issue raised in relation to the manual annotation process is scalability. Although increasing the number of annotators through crowd sourcing and an increased number of texts made available through release of de-identified clinical material has improved creation and visibility of reference standards, manual annotation is still a process that is unable to satisfy annotation requirements of large databases of clinical data (Velupillai et al., 2015). An example of the volume of data produced nowadays is the fact that at least one clinical document needing annotation has been produced by each of the over 795,650 clinical incidents that took place in the New South West hospitals between 2016-2019 (NSW Clinical Excellence Commission, 2020). The amount of manual work required for annotation of just that corpus would greatly exceed available staff capable of providing that service.

Automated annotation is a process where automated agents annotate corpora using one or more automated methods. Automated techniques are seen to perform at least as well as human agents (Deleger et al., 2013). In their meta-analysis, Stanfill et al. (2010) found that of the 113 studies they included in their review, 26 found that the automated system performed better than, or as well as, humans, while only four found the opposite.

Scalability is another aspect that puts automated annotation ahead of manual. Automated methods are able to scale to millions of texts eliminating scalability as an issue and minimising the costs over a period of time. However, automated annotation methods are not without their issues. Generalisability is seen as one. Turchin et al. (2006) report that generalisability is an issue commonly affecting automated annotation methods and the example is given noting that a new set of regular expressions is needed to be created every time a new corpus is annotated. Another advantage of human agents was due to their intuition. Human agents have shown

ability to assume meanings of abbreviations even when misspelled and have shown better performance when errors in punctuations were present in the text.

2.2.2 Review of annotation methods

Review of the literature (Fu et al., 2020; Sheikhalishahi et al., 2019) reveals that the following methods are the most commonly used methods that annotation techniques are based on:

1) Machine learning,
2) Rule based and
3) Hybrid

Machine learning techniques are further divided into 2 groups: shallow and deep learning. As we do not utilise machine learning, neither its shallow nor deep alternatives, the machine learning (including deep learning) discussion on these topics will be rather concise as it will not delve into specific methods that utilise these techniques.

However, before we provide more context on each of the annotation techniques, a question of pre-processing is worth touching upon. This is due to the issue of heterogeneity of clinical text making pre-processing of clinical text imperative for the success of annotation processes, regardless of the technique deployed.

2.2.2.1 Pre-processing

One of the conclusions reached by Torii et al. (2011) is that performance of methods based on machine learning degrades when the methods are ported across clinical sources. This brings to mind the issue reported by Turchin et al. (2006) where the authors list the limitations of rule based systems and point out that, because of the difference in structures of the texts originating from different sources, rule based systems need to be adjusted for every source.

This clearly is an issue of structural heterogeneity of clinical texts and should not be taken as an issue directly related to the methods used. Heterogeneity of text structures, high level of noise, use of abbreviations and syntactical and grammar issues to name a few are well known to be present in clinical texts (Kaurova et al., 2011; Nguyen & Patrick, 2016). The solution to

eliminating these issues is pre-processing of texts and some techniques have been developed in the past (Nguyen & Patrick, 2016).

Pre-processing involves one or more distinct tasks and the selection of tasks is on authors of the overall method.

One group of tasks is exclusion of meaningless data. These tasks are usually among the first to be applied, usually before the task of tokenisation takes place. The tasks include processes like exclusion of stop words, punctuation and unused space between words, sentences and paragraphs. The tools used for this are pattern matching tools, like regex and direct matching.

Stop words are words deemed irrelevant for the task and are collated in lists called stop lists or negative dictionaries. Although no specific stop word list has been selected as a gold standard, several lists of stop words have seen more use than others. The example is a list of stop words derived from the Brown's corpus of over one million words drawn from a broad range of English literature (Fox, 1989).

After removal of stop words and other data from the text that is considered meaningless, tokenisation takes place. Tokenisation is a process where the text is separated into linguistically significant and methodologically useful parts, called tokens. Tokens are considered "atoms", of text processing, the "indivisible units" that are not to be broken down any further (Webster & Kit, 1992). However, that is not to mean that tokens are limited to single words only — tokens can be compound tokens, tokens that contain more than one word, even the whole documents (e.g. W. Huang et al., 2012; Rygl et al., 2016).

After tokenisation, the tokens need to be brought into a format that can be recognised by the machine agents conducting the process of annotation. This is particularly important if the agents working on a process use reference resources like dictionaries and terminologies that normally would contain only normalised formats of the words. The tasks used for normalisation of words include change of spelling and tasks like lemmatisation and stemming.

Stemming is converting derived forms of word tokens to their root forms by removing words' suffixes. The example of stemming is changing word "attending" to the word "attend" or word "learning" to the word "learn". However, stemming often leaves words meaningless as the suffix is just removed and not changed to the expected root suffix after stemming. The example is the word "decided" that when stemmed becomes "decid". The word "decid" will normally

not be found in dictionaries, terminologies or other word classification resources (Tool used for this example: NLTK, 2020).

Lemmatization, on the other hand, transforms a word to its dictionary format (Plisson et al., 2004). It removes the affixes, revealing the root and then adds missing characters to the root making the word semantically complete. Lemmatisation often utilises large word databases for lookups to find canonical formats. Lemmatisation is generally more complex process than stemming as it can involve analysis of the word, removal of affixes and lookup for a suitable word in large dictionaries.

However, for lemmatisation to produce correct results, the lookup resource must be appropriate so the results of the lemmatisation can be used further down in the process. The example is the word "exercised" that, when lemmatised using the WordNet lexical database (MIT Press, 2020) as a lookup source, produces the word "exercised". The word "exercised" is not in any of the terminologies included in the UMLS Metathesaurus (O. Bodenreider, 2004) on its own, so it would not be of use to an automated agent that would utilise results of that lemmatisation in the process of term expansion, for example.

Similar situation is with the word "exercis", which is the result of stemming of the word "exercised". The word "exercis" has no meaning to any of the UMLS Metathesaurus sources. However, use of the word "exercis" will result to a match in the sentence "Patient was recommended exercising 3 times a week" as the word "exercising" will be stemmed to its root, the word "exercis". However, if lemmatisation that is based on WordNet database is used, the match will not be found as the results of lemmatisation will be the words "exercised" and "exercising" and these two words will not be seen as matches by an automated agent conducting the matching. This example confirms that the choice of lookup source is an essential step if lemmatisation is to be used in the process of annotation.

From the discussion above, it can be concluded that the process of pre-processing, to provide desired results, need to take in consideration use of its outputs in the remaining steps of the process. In some cases, lemmatisation of the word exercise will make sense, especially if matching utilises StartWith or EndWidth functions that are common in many programming languages and can be achieved with SQL using literal "%". However, in some cases that would be counterproductive, especially if the matching function expects full word matches only, in which case lemmatisation might be a more productive option.

2.2.2.2 Machine learning methods

The field of machine learning seeks to enable computers to automatically improve themselves (and their methods) based on experience (Mitchell, 2006). These methods are not limited to data related operations, but include learning of robots on how to navigate themselves and learning of search engines on how to adjust the results based on the users preferences, among others. One of the aims of machine learning is either to exclude or to involve minimal human input (Zech et al., 2018).

Machine learning is fast maturing field and machine learning algorithms can be found in the fields like speech recognition, computer vision, bio surveillance, robot control and accelerating empirical sciences. The difference between the human learning and machine learning is that machine learning requires much more data to achieve the same or the similar effect of learning, but machine learning can learn from massive amounts of data, the amounts that are impossible to process by human brain (Halevy et al., 2009). It is perfectly feasible for a machine learning method to be trained by millions of charts and EHRs while the same amount of information is not possible to be used in human learning.

The goal of the machine learning algorithms is to decide on a function $f: A \rightarrow B$ based on the set of learning examples $\{x_i, y_i\}$ of inputs x_i and outputs $y_i = f(x_i)$. In other words, machine learning creates a function that produces best estimation of the next output based on transformations observed during the learning phase.

The examples of methods that fall under machine learning are the methods from Support Vector Machines (SVM), Bayesian classifiers and Genetic Algorithms among others (Mitchell, 2006). i2b2 challenge in 2010 (Uzuner et al., 2011) and in 2012 (Sun et al., 2013) have made great use of machine learning methods. The participants utilised SVM, conditional random field, hidden Markov model, ensemble method and hybrid methods and concluded that performance of machine learning methods measured through precision and recall is better compared to the rule based methods. This conclusion was not in line with their earlier conclusion made at the third i2b2 Workshop on Natural Language Processing Challenges for Clinical Records, claiming that rule based methods were better than machine learning methods and that the best choice of methods were hybrid methods that are combination of both, rule and machine learning methods (Uzuner et al., 2010).

Machine learning methods generally fall under one of the following subcategories: supervised machine learning, unsupervised machine learning and hybrid machine learning. A notion of distance is essential to all the machine learning methods as without knowing how similar two items in a sample are, the algorithms are unable to cluster them (Gentleman & Carey, 2008).

Supervised machine learning (SML) methods are by far the most commonly used machine learning methods in classification of clinical reports (Hastie et al., 2009; Mirończuk & Protasiewicz, 2018; Guzella & Caminhas, 2009; Al-garadi et al., 2016). They are based on the pre-defined training data that is either re-used if existing, extracted from another text, or prepared by field experts and provided to the algorithms. Unsupervised machine learning is also known as cluster analysis or class discovery. The prerequisite for unsupervised machine learning is selection of samples for clustering and definition on parameters such are features for clustering, similarity parameters and metrices and the choice of the algorithm to use. This technique does not utilise training, so cross validation is not possible (Gentleman & Carey, 2008).

Deep learning, as a form of machine learning is touched upon next.

2.2.2.3 Deep learning methods

Deep learning is a form of machine learning. Deep learning is based on multiple linear transformation of a representation of knowledge. Each time the transformation is applied, the knowledge becomes represented at a higher, more abstract level (LeCun et al., 2015). Each time the transformation is applied, a new layer of knowledge is created and the number of layers created in the process is not limited. Recognised (or potentially recognised) connections between the inputs and the outputs of the transformations are kept and form a chain that is called Credit Assignment Path (CAP).

The end result of these connections recorded in the CAP during the course of transformations is a neural network. Although one layer can be parameterised (transformed) multiple times, in which case the resulting network is called recurrent neural network, creating layers in a linear fashion is possible as well, in which case the result is called feedforward neural network (Schmidhuber, 2015).

The example of a deep learning algorithm is a break-through algorithm developed by Hinton (2009) in which he uses the Restricted Boltzmann Machine algorithm (Sutskever et al., 2008) as an unsupervised learning algorithm applied on each knowledge layer. This was a first successful attempt to deploy learning algorithms to greedily train on one source one layer at the time (Bengio, 2009).

Deep learning is found to be successful in finding intricate structures in multi-dimensional data and has found its use in image recognition (Farabet et al., 2013; Krizhevsky et al., 2012; Sutskever et al., 2014), speech recognition (Hashemi, 2012; G. Hinton et al., 2012) and natural language understanding (Collobert et al., 2011). However, it is still not clear whether deep learning methods outperform machine learning methods. For example, Kohler et al. (2019) tested 6 deep learning methods across the 14 datasets and found that classical machine learning methods outperform deep learning methods in their cell-type prediction based on gene-signature derived cell-type labels task. Similarly, Liu et al. (2019) found that deep Learning based models did not outperform health care professionals in diagnostic performance.

However, some authors disagree with these findings and state that Deep Learning methods outperform standard methods. For example, Koutsoukas et al. (2017) found that Deep Learning methods outperform classical machine learning methods, like Naïve Bayes, k-nearest neighbour, random forest and SVM methods when modelling bioactivity data. They found that the level of noise makes significant difference and confirmed that is the level of noise is higher than 30%, Naïve Bayes performs better than the Deep Learning methods that they deployed.

In favour of deep Learning was a verdict of Bouktif et al. (2018) who compared predictive power of deep learning model and a classic machine learning model. However, it appears that their selection of data that they considered as optimal for the deep learning model has increased the chances for deep learning method to be more successful.

Numerous deep learning methods have been deployed to tackle the task of concept recognition/extraction in the medical domain (Chalapathy et al., 2016; Ling et al., 2017; Lv et al., 2016; Q. Wei et al., 2020; Zhu et al., 2018). However, although some indications have been made available (Gehrmann et al., 2018), the issues of generalisability and presentability of results have prevented systematic reviews to come to the conclusion on the utility of the currently available methods (Kersloot et al., 2020).

2.2.2.4 Rule based methods

Rule based annotation methods are based on a set of rules that are manually defined. This dependency on manual input is considered a bottleneck (Sebastiani, 2002) as manual involvement raises the same or very similar questions compared to that of fully manual annotation. This dependency on manual input, along with further improvement of machine learning methods has seen rule-based methods decrease in popularity. However, increased availability of expert knowledge in the form of ontologies, terminologies and similar information and knowledge resources is able to minimise manual input and automate rule creation, bringing rule based methods back in focus (Solt et al., 2009).

The rule-based annotation systems usually start with recognition of zones in clinical documents, e.g. sections like diagnosis, clinical course and similar. Regex patterns are usually used for that task, utilising regularity in (sub) titles and other elements used for demarcation of document zones. That part of the process is normally different in each source and, as pointed out by Turchin et al. (2006) and mentioned in the text above, prevents that the methods are generalizable.

After zoning, classifiers are created. Classifiers are groups of rules that mimic work of the human agent annotators. Traditionally, the work of classifiers is called concept normalisation. Concept normalisation is one step away from annotation as its output is a set of codes that represent tokens found in the document.

Some classifiers operate by performing dictionary lookup where dictionary can be a pre-defined list, a terminology or ontology. The tokens are either considered in a structured format, where their location and relation to other tokens is considered or as a bag-of-words in which case each token is considered a feature (Goldberg, 2017, p. 67). Recent example of using classifiers can be found in Xu et al. (D. Xu et al., 2020) where a classifier based on the Bidirectional Encoder Representations from Transformers (BERT) neural network (Devlin et al., 2019) was constructed and used to rank the concepts recognised in the text.

There can be one or more classifiers defined and the only difference between the classifiers is the set of rules that they contain. Classifiers take in consideration the zone in which they operate, hence the classifier will place different value on the concept found in the diagnosis section compared to the concept found in the family history section. Classifiers can also place different value (weight) on a type of concept discovered, e.g. clinical concept can be given

more weight than a concept that represents non-clinical entity. In addition to the concepts, the rules in the classifiers can be programmed to discover and extract assertions and relations.

2.2.2.4.1 MetaMap

A well-known classifier is MetaMap (Aronson, 2001). MetaMap is a linear combination of measures that takes lexically processed text as an input. Lexical processing (or pre-processing of text) starts with tokenisation, where sentence boundaries, acronyms and abbreviations are recognised. Part of speech tagging and lexical lookup are the processes that take place before the text is syntactically analysed. Part of speech tagging ensures that the words in the text are assigned contextually appropriate grammatical descriptors (Mitkov, 2004) and lexical lookup ensures that the recognised words are found in the lexical reference material (McGray et al., 1987). This precludes meaningless tokens from being included in the process. Syntactic analysis in MetaMap has very similar role as lexical lookup, but its works at the level of phrases and sentences. The algorithm MetMap uses in the syntactic analysis is developed by McCray et al. (1993) and is part of the SPECIALIST, an experimental natural language processing system for the biomedical domain. The SPECIALIST has grown into a system that comprises of a lexicon and a set of lexical tools and is distributed by the National Library of Medicine as one of the Unified Medical Language System (UMLS) Knowledge Sources (Lu et al., 2020).

After pre-processing phase, in which lexical/syntactical analysis methods are applied to the text, MetaMap deploys methods that focus on the semantics of the discovered tokens. The first of the five methods that all flow in linear fashion, is generation of variants found using table lookup. It is not clear whether the authors used a custom-made table in this case, or the method utilises one of the external sources, like one of the sources underpinning UMLS meta-thesaurus (O. Bodenreider, 2004). Next in the MetaMap process is the candidate identification method in which candidate strings (intermediate results) are evaluated by how well they match input text, before they are combined and re-evaluated as such.

The evaluation methods used in the last two steps as part of the MetaMap are based on the measures of centrality, variation, coverage and cohesiveness.

Centrality measures quality of match between a phrase and a meta-thesaurus candidate. It reflects the position of the word (a token recognised in the text) and a meta-thesaurus candidate. The example are tokens “eye” and “complications” that have “ocular complications” as a

candidate concept description. In this case, the token “eye” gets assigned a score of 0 as it matches no parts of the candidate concept description and the token “complications” gets a score of 1 as it is a direct match to one part of it.

Variation is a measure estimating the difference of the candidate concept and the token. The authors define four bases for differentiation and assign each a different value they call distance value. Spelling differentiation is assigned 0, inflectional 1, if the token is a synonym or an abbreviation of a candidate, the value is 2 and if it is a derivative of the candidate concept description, the value is 3. The final variation value can include more than one of these values if present.

Variation for each token is calculated using the following formula:

$$V = \frac{4}{D + 4}$$

Where:

- D is a total distance value for the token - a sum of all 4 variation types listed above

Total V for the candidate is calculated as average V calculated for each of the tokens used to decide on the candidate.

Coverage value indicates how much of the meta-thesaurus concept description’s text and the token’s text are involved in the match, ignoring the gaps between the words (spaces). Authors prescribe values called metathesaurus span for matched part of the candidate and phrase span for matched part of each of the tokens involved in matching. The final result is weighted average of values for the candidate string and the token. Candidate string is given twice the weight as the token.

Cohesiveness measures connectedness of the words forming both a matched phrase (a token) and a candidate. The value is calculated based on the maximal length of continuous words participating in the match, adjusted for the length of each of the components. The value for the candidate is calculated as a sum of the squares of the connected candidate string component sizes divided by the square of the length of the string. The same logic is followed in calculating the value for each of the tokens. The final cohesiveness value is weighted average of the candidate token values where candidate is again given twice the weight as the token.

The rules used in the MetaMap are highly customisable. That allows for outputs of the algorithm to be very different depending on the initial parameters that can be supplied to the algorithm. The ratio for allowing this flexibility in the algorithm might be based on the fact that gold standards do not exist in the field and that different applications of the algorithm would expect different outputs.

The criticism of the MetaMap is that the algorithm is unable to provide semantic relationship between the extracted concepts and the authors have to combine it with surrogate approaches to identify information on the strength of relationships between extracted concepts (Ogallo & Kanter, 2016). The authors report that it "was difficult to determine whether a study truly investigated the association between two medical conditions simply based on the co-occurrence of the two conditions in the article title" (Sung et al., 2020, p. 1).

In some cases, the authors have found that MetaMap provides too many false positives (Pape-Haugaard et al., 2020), that it does not recognise acronyms (Le et al., 2020) and that it has limited ability to recognise misspelled words (Hanauer et al., 2020). However, considering MetaMap algorithm customisation options, it is difficult to say whether these findings have to do with MetaMap's validity or used parameters.

2.2.2.4.2 cTAKES

The rule based concept extraction algorithm that is commonly compared with MetMap is cTAKES (Savova et al., 2010). The algorithm is created by the Mayo Clinic, released as open source software and is currently maintained by Apache Foundation. The algorithm uses the methods from the NLP domain before rule based named entity recognition method is used to match the discovered tokens to the entries in several databases. The NLP methods utilised in cTAKES are sentence boundary detector, tokenizer, normalizer, part-of-speech (POS) tagger and shallow parser. As shallow parser in this case utilises Apache's OpenNLP ME machine learning method, cTAKES could also be seen as a hybrid algorithm too.

After NLP methods pre-process clinical text, cTAKES uses a named entity recognition (NER) method described in (Pakhomov et al., 2005). This method uses SNOMED-CT, MeSH, RxNorm and Mayo Synonym Clusters (MSC) as lookup resources. Access to the MSC is a significant advantage for this method as this resource contains a set of clusters each consisting of diagnostic statements considered to be synonymous. This source has been manually

compiled from Mayo Master Sheet repository, a collection of over 20 million manually coded diagnostic statements held by the Mayo clinics.

MetaMap and cTAKES have been compared and the results have shown similar performance, with cTAKES slightly outperforming MetaMap (Reátegui & Ratté, 2018; Rodríguez-González et al., 2018). Two methods are known to be combined in experiments that used hybrid approach and it has been reported that the performance of the combined method was satisfactory (Tang et al., 2013; Xia et al., 2013).

2.2.2.4.3 MedLEE

Another method that has been widely applied and has improved significantly over time is MedLEE (Friedman et al., 1994). The method consists of 3 stages, first of which is parsing. Parsing uses grammar and lexicon to convert unstructured text into its preliminary structured form. In the second stage, phrases are regularized to minimise the styling differences that are common in natural language. In the last stage, the standard forms are encoded into concepts found in a controlled vocabulary, called the Medical Entities Dictionary (MED). This last mapping is a one-to-one mapping as the forms have already been standardised in the second stage.

MedLEE was originally designed for decision support tasks in the radiology domain, chest x-ray reports specifically (Friedman et al., 1999). Later, it has been customised for use in the fields of pharmacovigilance (X. Wang et al., 2009), adverse event detection in discharge summaries (Melton & Hripcsak, 2005), nursing (Bakken et al., 2005) and biomedicine (Chen & Friedman, 2004) among others.

The main limitation of MedLEE is seen in its inability to make inferences from individual concepts and in its inability to reason about relationships among concepts (Chiang et al., 2010). Similar is noted in the (Lussier et al., 2001, p. 422) with the recommendation that MedLEE should consider adding functionality that would “gracefully transform the structured coded output of MedLEE into a standardized information model such as HL-7 for laboratories or DICOM for radiology”.

2.2.2.4.4 Rule-based methods - conclusion

All of the methods, regardless of whether they are rule based or based on machine or deep learning, put strong focus on what is in the target text, on the recognised entities and how these entities can be utilised in achieving whatever the goals are of the project that the method is deployed in. However, these entities can be in any one of many of their linguistic forms, making them hard to find, understand and relate to the rest of the text. For some tasks, finding several key entities could be enough, but other tasks will need to look at a bigger picture to perceive intended meaning of the clinical free text put before them so they can achieve their projects' goals.

The examples of the tasks that we refer to are classification of medical documents, comparison of medical documents and search through medical documents among others. We are concerned that, considering syntactic and semantic ambiguities of the natural language, the messages in these documents can go unnoticed or be misinterpreted. Although the current methods match the tokens they find to concept descriptions in the normalised sources, like medical terminologies and ontologies, we believe that that the process would reveal more if the concepts recognised are expanded and positioned in the reality before compared with another medical document, which again represent another segment of reality on its own.

We next provide our philosophical standpoint that we base our research goal and assumptions on, expand on application of ontology of reality in annotation and describe what we believe the solution to efficient annotation of clinical text should involve. Also in the next chapter, we discuss SNOMED CT as an ontology of reality and report on our research conducted to test its coverage and utility as an annotation ontology.

3 Research requirements (DSR activity 2)

As expected by the DSR process, we define project goals and underlying assumptions in this chapter (Hevner & Chatterjee, 2010, p. 20). After project goals and assumptions, we touch upon direct and indirect requirements for designing the artefact, which is an algorithm in the case of this research. Indirect requirements are prerequisites that make the main artefact possible to be created, tested or executed.

As mentioned earlier in the text, the example of a direct requirement of the artefact is "The algorithm needs to be able to expand clinical concepts using SNOMED CT as the ontology of reality" and the example of a prerequisite for that is that "A method for converting SNOMED CT into a graph that can be traversed by available computational systems in realistic time is developed". Despite not being directly related to the goal of the project, the prerequisite tasks are of high importance as they ensure that certain steps in the project can be completed.

3.1 Project goal and underlying assumptions

A choice of a method depends on the goals of the project that the method will be deployed in and a "research project involving text must start with a clear definition of its overall goal and the role text will play in achieving that goal" (Percha, 2020, p. 3). Hence, we start by explaining what our goals is, before we provide the assumptions that the goal is based on.

The goal of this project is to start a process towards defining a method for accurate annotation of human agent constructed messages contained in medical documents using composite information models, so that these documents can be positioned in the ontology of reality and their similarity to other clinical documents measured on a basis of their distance. The goal will be achieved by defining the artefact (an algorithm) that will provide a starting point in that process. The artefact will be implemented as a software product for the purpose of evaluation.

For that to be possible, an ontology of reality needs to be available for the composite information models used for annotation to be positioned in, so their distance can be measured for deciding on their similarity. ***It is assumed that distance between two information models positioned in the ontology of reality will depict the level of similarity between the documents they represent (Assumption 1)***. ***It is assumed that the closer the information models are in***

the ontology of reality, the more similar they are (Assumption 2). We consider that this approach will minimise impact of natural language ambiguity as it takes into consideration semantics of a message conveyed as a medical document, rather than just concepts contained in the message.

This implies that two important sub-processes need to take place in the overall process: 1) conversion of medical documents into information models that can be placed in the ontology of reality and 2) comparison of these models. Moreover, a suitable ontology of reality that contains these information models needs to be available. ***It is assumed that SNOMED CT can be used as ontology of reality (Assumption 3).***

We next explain the philosophical grounding for assumptions 1) and 2) before we explain the reasons for selecting SNOMED as the ontology of choice to represent a segment of reality that information models that we create for depicting semantics of messages conveyed in medical documents can be positioned in (assumption 3).

3.1.1 Philosophical grounding of the underlying assumptions

Our approach is rooted in the constructionist philosophical stance that hypothesises that reality is a conceptual system, a construction defined by concepts that in themselves are also constructions that can be observed in isolation as prescribers of their own realities. These concepts are seen as entities fundamental to our cognitive architecture that are immutable and final in their meaning and provide a layer of abstraction over the syntactic differences of their representations used in language (Smith, 2004).

This is in line with the Aristotelian proposition that reality is a construction made of particulars, often understood as objects, that have their definitions in universals, that are in modern times also known as classes. Although universals are seen as blueprints of the individuals ("individuals" is another term used for particulars), they are not seen as separate, independent entities. That means that it is a distinct individual that is a prerequisite for the existence of a universal and that without an individual, a universal would not exist (Vezina, 2007).

It is our view that this assumption of particulars and universals allows for reality, or segments of reality, to be presented using signs. These signs are the author's interpretations of particulars and each carries a meaning that is provided to a receiver (the reader) for further interpretation.

The field of semiotics approves this view in "*aliquid stat pro aliquo*" (or anything that stands for something is a sign), its definition for a sign. Expanded in their triadic model, a sign is defined as a representation of a meaning as it is derived through senses based on one's reference (Chandler, 2017).

The signs, for us, are interpretations of particulars found in the reality and concepts that communicate meaning. When grouped in messages and persisted in the corpuses of text, they represent knowledge of reality. This is in line with the anti-realist doctrine of constructivism. The anti-realists assert that there is no plausible evidence that proves that knowledge of reality can exist without of an agent that creates it based upon its perception (Potter, 1998). As opposed to that, the realist doctrine asserts that the knowledge of the reality exists independently of the representation of an observer and that it does not need a "knower" to create it (Searle & Slusser, 1995).

In the case of clinical messages contained in clinical documents, they are created by agents and contain signs (words, sentences, other text elements) that are agents' interpretations of the particulars that exist in their reality; hence, they represent a segment of that reality in which they can be positioned if converted into an appropriate model compatible with the model that represents a reality. This explains the origin of our ***Assumption 1***.

Notions of the ontology distances as measures of similarity are not new. Wang (2010) deployed a learning method that used ontology difference as a function of similarity. Huang et al. (2011) ranked ontology graph nodes and associated real numbers with each of the nodes. They then calculated similarity of the nodes based on the differences in the numbers that represented the nodes. Formica (2006) used ontology in deciding on similarity of concepts in the process of Formal Concept Analysis. They assigned similarity degrees to concepts found in an ontology and, based on these values as well as similarity on the concepts' relationships and their distance in that particular ontology, they calculated concept similarity. The examples of use of ontology distance as a measure of similarity are many, and many of them use SNOMED CT as the ontology of choice (Batet et al., 2011; W.-N. Lee et al., 2008; Tongphu & Suntisrivaraporn, 2017; D. Wei & Fu, 2017). Results reported by these authors makes us confident that our ***Assumption 2***, that proximity of two information models in the ontology of reality is a function of their similarity, is sound.

SNOMED CT is claimed to be the most comprehensive single resource of clinical terms available today. It has been created and is maintained by the SNOMED International organisation that currently has 37 countries as governing members. The number of concepts in the January 2019 edition of this resource was 349,548 and it is showing a steady growth in the number of concepts added in every edition. Its content is clinically verified and enables accurate representation of clinical concepts in electronic clinical records and clinical analytics systems (D. Lee et al., 2013).

SNOMED's structure consists of 19 large hierarchies that are joined through the top-level concept, called the SNOMED CT Concept. The hierarchies are made of concepts connected using the Is_A attribute. Hierarchies themselves are large acyclic graphs in nature and no concept from any of the hierarchies connects to any of the concepts from the other 18 hierarchies using Is_A attribute. In addition to the Is_A attribute, which is used purely for depicting hierarchy, other attributes, listed in the SNOMED CT Model Component (metadata) hierarchy, are used to depict relationships beyond taxonomical sub-typing. In contrast to the Is_A attribute, these attributes can 'connect' concepts located in different hierarchies. For example, *SCTID: 363698007, Finding site (attribute)* is used to represents the relationship between the concepts *SCTID: 64662007, Pulmonary infarction (disorder)* and *SCTID: 39607008, Lung structure (body structure)* that reside in different hierarchies.

The ability of SNOMED to provide axioms using a concept's hierarchical and non-hierarchical relationships allows for reasoning and classifies SNOMED as an ontology rather than just a taxonomy (J. Bodenreider, 2018). This is supported by the fact that the axioms in SNOMED CT are presented using Compositional Grammar that can be interpreted using Description Logic (SNOMED International, 2019). Considering that the number of concepts covered in SNOMED CT is significant and that SNOMED CT is a quality-controlled resource created by experts and scrutinised by many, it can be assumed that the axioms postulated in SNOMED CT are a close representation of the propositions in reality or in its representative ontology. These characteristics make SNOMED CT a suitable source for use in knowledge extraction and for term expansion in particular. These facts support our ***Assumption 3*** that SNOMED CT can be used as ontology of reality.

3.2 Utility of SNOMED CT as ontology of reality

We selected SNOMED as ontology of reality because of its focus on the medical domain, coverage (Zivaljevic et al., 2020), richness of relations between its concepts and presence of subgraphs that represent their own specific segments of reality. Moreover, as SNOMED is actively developed ontology, we expect that it will only increase in coverage and richness of inter-concept relations and that will further increase its utility as an ontology of reality.

We also use SNOMED CT in the process of creating composite information models that we annotate medical documents with. Specifically, we use SNOMED CT in the process of term expansion. Term expansion is a frequently used technique for knowledge extraction. Term expansion is used to overcome vocabulary mismatch issues where expansion increases the scope of analysis to cover syntactic patterns that are not otherwise present in the text. This technique keeps compound or obscure terms from preventing or confusing concept identification, making processes like knowledge extraction more effective (Alani et al., 2003).

In term expansion, a term is supplemented with associated terms found in a source selected as an expansion repository. This source can be any body of text, however, the most commonly used sources are domain-specific and domain-independent ontologies and databases. SNOMED CT, in particular, is used for work in the domain of medicine and human biology (Stokes et al., 2009; Y. Wang et al., 2008).

As part of our work on annotation of clinical datasets using openEHR Archetypes (Zivaljevic et al., 2015b), we need a repository that we can use as a source for expansion of clinical terms found in the free-text discharge summaries that we use as a sample. We expect that the source will be comprehensive and have its concepts and their relationships presented as closely as possible to reality, and thus be a valid representation of the clinical domain. Considering its comprehensiveness, quality control and involvement of clinicians in its creation, SNOMED CT is presumed to be an obvious choice. It is expected that its 19 hierarchies, that each cover different subdomains relevant to the clinical domain, will provide valuable guidance in noise minimisation efforts as they will allow for different weighting to be assigned to different (sub)domains of interest.

However, the research community reports issues that can potentially render SNOMED CT unsuitable for term expansion. Rodrigues et al. (2018) raise the question of modelling issues in SNOMED CT and suggest quality assurance improvements. However, their conclusions are

based on the limited sample, comparison of SNOMED CT concepts and mapped ICD11 classes from the circulatory and digestive chapters only.

Miñarro-Giménez et al. (2018) conducted a qualitative analysis testing the utility of SNOMED CT in coding clinical text by measuring inter-annotator disagreement. Their results show that their annotators matched what was defined in the reference standard only in 21.6% of the codes assigned to the terms found in the provided sample. This, as the authors term it, "astonishingly low" result could mean that SNOMED CT features a high level of ambiguity, which would render it unsuitable for our research due to the potential introduction of noise in the term expansion process. However, the issues that the authors list as factors causing this are mostly subjective or of a research methods nature and do not depict a failure of SNOMED CT. As mentioned in the text, the factors include annotators' lack of domain knowledge, carelessness, annotation guideline issues, interface term issues and language issues.

Bona and Ceusters (2018) alert that some of the concept tags in SNOMED CT incorrectly identify a concept's place in the hierarchy. The semantic tag is part of the concept's description and comes surrounded in parentheses. It functions to disambiguate the description of the particular concept from the descriptions of other concepts that might be the same but might belong to other concepts. The example that the authors give are the concepts SCTID: 35566002, Hematoma (morphologic abnormality) and SCTID: 385494008, Hematoma (disorder). It was found that 89 of over 300,000 concepts in the SNOMED CT 20170131 release have mismatched semantic tags. However, based on the magnitude of the problem (or lack of it), we discard this issue as minor to our study.

The issue of SNOMED CT's coverage has been raised by many (Rodrigues et al., 2018). This deficiency of SNOMED CT is reported to have practical implications on its utilisation in the clinical domain. For example, Liu et al. (2018) established that SNOMED CT's adoption among ophthalmologists in the US has not reached expected levels and find that the reason for that is the coverage of SNOMED CT's ophthalmology component. Similarly, Rastegar-Mojarad et al. (2017) raise an issue of coverage in their attempt to map the list of procedures from their Gynaecology Surgery Registry to SNOMED CT. They found that only a small percentage of procedure names can be mapped to the concepts in SNOMED CT. The reason behind this issue was not absence of suitable terms in SNOMED CT but the format in which the procedures are represented and use of what they call procedure modifiers. This issue observed by Rastegar-Mojarad et al. is an important one as it indicates that the issues seen as

SNOMED CT coverage could actually be unrelated to SNOMED CT and related to how concepts are presented in the corpus.

In their comparison of Nomenclature for Properties and Units (NPU), Logical Observation Identifiers Names and Codes (LOINC), and SNOMED CT for use in representing laboratory results, Bietenbeck, Boeker and Schulz (2018) suggest that for the best coverage SNOMED CT and LOINC should be used together as some of the concepts in SNOMED CT are represented in rather complex fashion, using post-coordinated, rather than pre-coordinated expression. However, they note that this particular issue is by design and is to ensure that duplication between SNOMED CT and LOINC is minimal.

The issues of SNOMED CT coverage raised in the literature have prompted us to question its suitability to our study. As our plan is to conduct ontological term expansion of the terms found in the sample corpus, we need an ontology that incorporates as many of our target concepts as possible. Ideally, the ontology will have all of the target concepts incorporated, but the possibility for that to be the case is low.

For that reason, as part of this work we test the suitability of SNOMED CT for use in expansion of clinical concepts in terms of its coverage. The main goal is to test whether the coverage of SNOMED CT will allow for the majority (if not all) of the clinical terms from our sample corpus of discharge summaries to be recognised. We expect that at least 90% of the clinical terms found in the discharge summaries are available in SNOMED CT. This is an arbitrary number based on a consensus of the research team members and work of the authors mentioned later in the text. If less than 90% coverage is found, we will consider terms from other ontologies for inclusion in SNOMED CT using extension. However, if other ontologies show similar results, we will revert to using SNOMED CT due to its future prospects in terms of development and expansion, as well as research team's expertise in the subject. We next discuss the study and its results.

3.3 Testing SNOMED CT coverage

Deidentified clinical records used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY (Uzuner et al., 2007). The sample contains 889 unannotated, de-identified discharge

summaries provided as free text. We extracted each record's text and stored it into an SQL database along with the related RECORD ID.

The free text extracted needed to be cleansed of elements that do not hold any value in the process. This particular corpus had section captions that divided free text. The captions were not standard and varied from discharge summary to discharge summary. Due to the fact that the captions contained no medical concepts related to the patient health record, but generic descriptions of the sections (Diagnosis, Family History, etc.), they were temporarily removed from the corpus. Some value was seen in classification of the concepts recognised under the headings, but we ignored that knowing that the headings are specific to this particular corpus and might not be found in other corpuses of clinical free text. This step will most likely be different for different corpuses, due to structural variations that can be expected. We call this step Text Cleaning.

Boundary Detection took place next. In this step we split the text in the array of sentences. For that to be possible, line breaks were normalised and changed to full stops if they were not present and double spaces were replaced with single space. It was ensured that the sentence boundaries were kept intact so that multi-word text fragments would not cross them.

The candidates for clinical concepts were extracted next, in the step we call Fragments Extraction. The algorithm extracts text fragments that are up to 5 words long, contain only a-z, A_Z, -,`, and space. We kept the interpunction in the sentence intact which ensured that the extracted text fragments would not span across sentence segments. The minimal number of characters for the text fragments that are one word in length is set to 3. The minimum length for the words in the text fragments made of 2 or more words has not been set. Text fragments are recorded once only, regardless of how many times they appeared in the corpus. However, the number of times that text fragments appear in the corpus has been recorded.

The algorithm used to extract the text fragments tokenised the text into 1-word tokens and then, starting from each token, the algorithm selected all n-word text fragments up to n=5. The duplicates were then removed and text fragments of one word in length were filtered for stop words. The fragments that contained more than one word were not filtered of stop words. The final array was cached into a database.

Due to our focus on expressiveness of terminology rather than higher level semantics of the text fragments retrieved, we were not concerned with recognising negation in the corpus. We see negation as a topic on its own and a question separate to this stage of our research.

Each of the extracted text fragments were then used to formulate a query to the Unified Medical Language System (UMLS). The UMLS Representational State Transfer REST application program interface's APIs ware used to query whether an exact match for the extracted text fragment exists in the UMLS Metathesaurus. If the exact match or matches existed, the source terminologies were recorded as associated to the text fragment used in the query. The information was returned by the UMLS in the JavaScript Object Notation (JSON) format and JSONs were saved in the database for future reference.

If SNOMED CT was listed as a source terminology for the discovered concept, we were also interested in which of the 19 SNOMED CT hierarchies the concept belongs to, so we made a recursive query that revealed that. The version of SNOMED CT available in UMLS was the US Edition version 20190301.

As our aim was to learn how good of a term expansion source SNOMED CT will be for our study, we wanted to establish the level of coverage that SNOMED CT will provide to our corpus. Elkin et al. (2006) conducted similar research testing how good of coverage SNOMED CT provided for coding the most common conditions seen at one of their large clinics. They found that the coverage was 92.3% and declared that as sufficient coverage.

Following Elkin et al., we used sensitivity as one measure. For us, sensitivity answers the question of how to quantify a UMLS source's (the Source) coverage. For example, for SNOMED CT, we calculate Sensitivity Ratio as the number of concepts found in SNOMED CT over the number of concepts found in the UMLS as if it was without SNOMED CT. We calculated Sensitivity Ratio for every Source where at least one text fragment matched a concept. The calculation is depicted in the following formula:

$$TPR_{Source} = \frac{TP_{Source}}{TP - UTP_{Source}}$$

Where:

- TPR_{Source} is Sensitivity (true positive rate)
- TP_{Source} is the number of true positives (text fragments matched to concepts) in the source

- TP is total number of text fragments matched to at least one concept in at least one source in UMLS – total number of true positives for all UMLS sources
- UTP_{Source} is text fragments uniquely matched to the concepts in the Source only

We also wanted to know the rate of false negatives as an indicator of the number of concepts that will not be recognised if the particular source is used. False negatives are the concepts found in the UMLS but not in the tested source over the total number of concepts found in the UMLS. We calculate false negatives using the following formula:

$$FNR_{Source} = \frac{TP - TP_{Source}}{TP}$$

Where:

- FNR_{Source} is false negative rate
- TP is total number of text fragments matched to at least one concept in at least one source in UMLS – total number of true positives for all UMLS sources
- TP_{Source} is the number of true positives (text fragments matched to concepts in the source)

As we were also interested in the coverage of distinct SNOMED CT hierarchies (organised as SNOMED CT axes), we calculated sensitivity of each one of the SNOMED CT hierarchies. For us, sensitivity of each hierarchy is the number of concepts found in that hierarchy over the number of total SNOMED CT concepts recognised:

$$TPR_{Axis} = \frac{TP_{Axis}}{TP_{SNOMED}}$$

Where:

- TPR_{Axis} is sensitivity of a particular SNOMED CT hierarchy (axis)
- TP_{Axis} is total number of true positives for a particular SNOMED CT hierarchy (axis)
- TP_{SNOMED} is the number of true positives for the entire SNOMED CT source

Considering that the corpus of text we work on consists of discharge summaries, we expected that significant sensitivity would be shown by the hierarchy with “Clinical finding (finding)” as its root concept.

Another point of interest for us was the coverage of other sources over SNOMED CT’s false negatives. For us, the number of false negatives for a Source is the number of text fragments

that were not matched to any of the concepts in that particular source but were matched to one or more concepts in any of the other Sources. We decided to single out the ontologies with significant coverage of SNOMED CT's false negatives so that we can, if necessary, expand SNOMED CT using those concepts and their relationships to the SNOMED CT concepts as defined in the UMLS. The formula is as follows:

$$TPR_{FNR_{SNOMED}} = \frac{TP_{Not\ in\ SNOMED}}{TP - TP_{SNOMED}}$$

Where:

- $TPR_{FNR_{SNOMED}}$ is coverage of SNOMED CT false negatives
- $TP_{Not\ in\ SNOMED}$ is the number of text fragments matched to concepts in the Source and not matched to concepts in SNOMED CT
- TP is total number of text fragments matched to at least one concept in at least one source in UMLS – total number of true positives for all UMLS sources
- TP_{SNOMED} is the number of text fragments found in SNOMED CT

We placed a threshold at 40% sensitivity per source ($TPR_{FNR_{SNOMED}} \geq 40\%$). In other words, if the ratio between the number of false negatives found in the Source X and the total false negatives is equal to or over 0.4, we would consider inclusion of the recognised concepts from the Source X into the SNOMED CT extension along with their relationships with the SNOMED CT concepts as they are depicted in the UMLS. If, after including sources with $TPR_{FNR_{SNOMED}} \geq 40\%$the coverage is still below 90%, we intend to go keep including sources until the coverage result reaches 90%.

A question might arise on why we have not used Specificity as a measure in this work. Specificity takes into consideration true negatives, and in the case at hand, Specificity would reveal the proportion between the number of text fragments that are correctly identified to have no meaning as concepts (true negatives) and the total number of text fragments that truly have no meaning as concepts (true negatives + false positives). The formula would be as follows:

$$Specificity_{Source} = \frac{TN_{Source}}{TN_{Source} + FP_{Source}}$$

Where:

- $Specificity_{Source}$ is specificity for a Source
- TN_{Source} is the number of true negatives for a Source
- FP_{Source} is the number of false positives for a Source

We could calculate the number of true negatives for a Source as the number of text fragments that have not been recognised neither in a particular Source nor in one of the other Sources. However, establishing a reliable number for the false positives was not possible without human agents. The question on whether a recognised match between a text fragment and a concept in a Source is a true negative or a false positive cannot be answered using automated methods at this point in time.

3.3.1 Results

Out of 208,513 text fragments of 1-5 words length, 9777 matched to at least one concept in at least one UMLS source. The number of UMLS sources that the matches are found in was 85. The top 5 sources were Consumer Health Vocabulary (CHV), National Cancer Institute (NCI), SNOMED CT US Edition (SNOMEDCT_US), LOINC (LNC) and MeSH (MSH).

Table 1 shows the experiment results for the top five sources. The field "Concepts" shows the number of text fragments recognised as concepts in a particular Source. The next field, named "Unique" shows the number of text fragments recognised as concepts only in one particular source. Sensitivity Ratio is given in the next field. It is explained in Equation 1 as the number of concepts found in the Source over the number of concepts found in the UMLS as if it was without that particular Source. The next field – false negatives is given in Equation 2 as the number concepts found in the UMLS but not in the tested source. And the last field in this table shows false negatives of the Source in relation to the false negatives of SNOMED CT.

Source	**Concepts**	**Unique**	**TPR** $_{Source}$	**FNR** $_{Source}$	**TPR FNR** $_{SNOMED}$
CHV	6932	916	78.23%	29.10%	57.91%
NCI	4750	642	52.00%	51.42%	44.10%
SNOMEDCT_US	4750	261	49.92%	51.42%	0.00%
LNC	2597	144	26.96%	73.44%	15.66%
MSH	2500	74	25.77%	74.43%	20.33%

Table 1 - Experiment result

Table 2 shows the results of the combined sources. The first field lists the number of text fragments recognised as concepts by the relevant combination of Sources. The next field shows the number of concepts that are uniquely recognised when the particular Sources are combined. Sensitivity and false negatives for the particular combination of the sources are given as the last two fields in this table (description as for the fields in Table 1).

Source	**Concepts**	**Unique**	**TPR** Source	**FNR** Source
SNOMED CT + NCI	6967	903	78.51%	28.74%
SNOMED CT + NCI + LNC	7356	1047	84.26%	24.76%
SNOMED CT + NCI + LNC + MSH	7889	1121	91.13%	19.31%

Table 2 - Selected sources and combined res

Table 3 shows a sample of text fragments, their occurrences in the discharge summaries and the number of matching concepts in each of the top 5 Sources.

Text fragment	**Occurrences**	**CHV**	**NCI**	**SNOMEDCT_US**	**LNC**	**MSH**
report	862	2	1	1	5	1
status	862	1	1	1	1	0
discharge	831	3	4	3	4	0
patient	723	1	2	1	3	1
summary	661	0	1	0	1	0
pain	518	1	6	1	3	1
stable	513	1	2	1	1	0
admission	511	1	1	0	1	0

Table 3 - Sample of matched text fragments, their occurrences in the text and the number of matching concepts in each Source

Table 4 shows a sample of text fragments that have been matched to concepts in Sources other than SNOMED CT, their occurrences in the discharge summaries and the number of matching concepts in each of the top 5 Sources.

Text fragment	**Occurences**	**CHV**	**NCI**	**SNOMEDCT_US**	**LNC**	**MSH**
summary	661	0	1	0	1	0
admission	511	1	1	0	1	0
not	504	0	1	0	1	0
admitted	479	1	0	0	2	0
discharged	463	0	1	0	1	0

treatment	431	1	3	0	3	2
room	407	0	1	0	2	0
service	386	1	1	0	2	0

Table 4 - Sample of text fragments matched to Sources other than SNOMED

Table 5 depicts the result of the coverage of distinct SNOMED CT hierarchies organised as SNOMED CT axes. It lists the SNOMED CT hierarchy axis followed by the number of concepts that belong to the particular SNOMED CT hierarchy axis and its coverage.

SNOMED CT Axes	**Concepts**	**TPR Axis**
Qualifier value (qualifier value)	1545	32.53%
Clinical finding (finding)	1087	22.88%
Substance (substance)	688	14.48%
Body structure (body structure)	617	12.99%
Procedure (procedure)	430	9.05%
Physical object (physical object)	195	4.11%
Observable entity (observable entity)	187	3.94%
SNOMED CT Model Component (metadata)	129	2.72%
Social context (social concept)	116	2.44%
Organism (organism)	109	2.29%
Environment or geographical location (environment / location)	100	2.11%
Physical force (physical force)	17	0.36%
Event (event)	13	0.27%
Pharmaceutical / biologic product (product)	10	0.21%
Situation with explicit context (situation)	9	0.19%
Staging and scales (staging scale)	7	0.15%
Record artifact (record artifact)	4	0.08%
Specimen (specimen)	3	0.06%
Special concept (special concept)	3	0.06%

Table 5 - Coverage of SNOMED CT axes

3.3.2 Discussion

In this study, we have demonstrated a fully automated approach for selecting sources for term expansion based on its coverage of the corpus of clinical text. Similar approaches were described in the past, some as theoretical concepts or frameworks (Alani et al., 2003; Li & Motta, 2010), some as solutions that require input of a human agent (Maiga & Ddembe, 2009) and some as complex sets of algorithms that include more than just a test of coverage in the process of source evaluation (Martínez-Romero et al., 2014, 2017).

The method we developed differs from other methods in several aspects. Firstly, it includes consideration for preparation of text for coverage testing. As we expect that the input is free text, we allow for text cleaning, stop word removal and tokenisation. Secondly, we deployed a version of brute-force testing of text fragments for presence in the sources. Starting from our one-word text fragments, we extracted all multi-word text fragments up to the length of 5 and tested them all for presence in the sources. We used caching to ensure that there was no performance penalty for doing so. And thirdly, our measure for selection of the sources to extend SNOMED CT as the preferred source was the sources' coverage of SNOMED CT's false negatives – $TPR_{FNR_{SNOMED}}$. This measure allowed us to find the sources that contain concepts that are the best match for the concepts missed by SNOMED CT. We also made sure that the concepts from other sources selected to supplement SNOMED CT do not overlap as we checked coverage for false negatives after adding every new ontology to the mix.

We showed that we were able to use this methodology to test coverage of SNOMED CT and to find sources whose concepts would be good contributors if included as source extension. The surprise in the study was the level of coverage provided by the open source Consumer Health Vocabulary (CHV). At 78.23%, this source indeed shows as the best choice for use in our study. However, our research indicates that the open source project that has created it is not active and that raises concerns for future support and expansion. Moreover, the source was created with the purpose of helping clinicians to communicate complex health related concepts to their patients (Mujib et al., 2018) and has no ontological properties. Due to these reasons, we decided not to use CHV as the main source in our next study.

We also considered using CHV as a source for extending the selected source. However, despite offering an impressive 916 concepts that are unique (not present in other sources), and 57.91% of coverage of SNOMED CT's false negatives, CHV's lack of ontological properties made us decide against that. Although the richness of this source would certainly increase the number

of text fragments recognised as concepts, its lack of ontological properties makes it unsuitable for use in semantics operations.

NCI and SNOMED CT are found to have the same number of text fragments recognised as concepts. Accordingly, their number of false negative results, represented as FNR_{Source} is the same at 51.42%. Coincidence is our only explanation for the same number of recognised concepts in the two sources that are known to have just 17.1% overlap (NLM.govt, 2019b). That prompted us to re-run experiment several times and to re-consider the source code. However, the results were confirmed to be correct.

The number of recognised unique concepts in NCI was significantly higher than in SNOMED CT or any other sources, even in CHV if considered relative to the number of recognised concepts (13.5% vs 13.2%). The difference in the number of unique concepts between NCI and SNOMED CT is reflected in the slight difference in the TPR_{Source} value where NCI has achieved slightly better result.

Considering that NCI is a terminology source highly specialised in oncology and based on the higher number of concepts found in NCI than in SNOMED CT, an assumption can be made that SNOMED CT does not have good coverage of oncology specialised concepts found in the corpus. This assumption would be in line with the reports on SNOMED CT's low coverage of cancer related concepts. For example, Raje and Bodenreider (2017) compared Disease Ontology (Schriml et al., 2012) and SNOMED CT and found that many of the Disease Ontology's cancer and neoplasm concepts were not found in SNOMED CT. Similarly, Melton et al. (2006) surveyed SNOMED CT for coverage of colorectal cancer surgery concepts and found that equipment and finding concepts were insufficiently represented.

However, none of the text fragments that our algorithm recognised as concepts in NCI and not in SNOMED CT are concepts specific to the field of oncology. Table 4 presents a sample of these text fragments. Although not sufficient to provide an indication on adequacy of SNOMED CT coverage of oncology related concepts (unless NCI is used as a gold standard), this observation might be indicating that SNOMED CT's coverage of oncology concepts is in line with that of NCI's. However, further testing, using methods other than deployed in this project, is needed before any conclusions on sufficiency of SNOMED CT's coverage of these concepts or comparison of coverages of SNOMED CT and NCI are made.

Nevertheless, some conclusions on the utility of extending SNOMED CT with the concepts from NCI can be made. NCI's $TPR_{FNR_{SNOMED}}$ value was also substantial at 44.10%, implying that almost half of the concepts found in NCI were not present in SNOMED CT and vice versa. This indicates that these two sources would be good contributors to one another. We calculated that, if the terms from NCI are used as an extension to SNOMED CT the resulting source would provide a total positive result of 78.51% and reduce the number of false negatives to 28.74%.

The number of concepts recognised by LOINC and MeSH was relatively small compared to the top 3 sources. However, their use as extensions of the main source is worth considering. LOINC's overlap with SNOMED CT is small at 6.7% (NLM.govt, 2019a) and considering its specialisation in laboratory results, we saw it as a good candidate for using as an extension to the main source. MeSH's overlap with SNOMED CT was also small at 8.1% and the prospect of the increase in coverage of close to 7% prompted us to decide that its concepts are worth inclusion.

The next source with the largest number of concepts that are not contained in SNOMED CT, NCI, LOINC and MeSH is MEDCIN. The number of concepts that MEDCIN would contribute would be 145 with 87 unique. That number of concepts does not significantly improve the resulting ontology, so we decided that it was not worth including. We present the results with multiple sources in Table 2 in the section 3.3.1 Results.

Although the results showing the root axes of SNOMED CT terms reveal that the greatest number of results comes from the Qualifier value hierarchy, close to 50% of the concepts come from the next 4 hierarchies Clinical finding, Substance, Body structure and Procedure that contain the concepts of highest relevance for expansion of clinical concepts. This clearly shows the need for assigning to or at least adjusting weight of the recognised concepts based on the semantics relevance of their roots.

3.3.3 Limitation

The algorithm we used for preparing text fragments for matching to concepts did not stem the text fragments before they were sent to UMLS for matching. Stemming is used to reduce variant word forms to common roots (J. Xu & Croft, 1998). Our tests conducted during the experiment show that deploying the Porter Stemmer (Porter, 1980) would improve recognition

of the text fragments by 8-10%. We tested just the text fragments that have been found in at least one of the sources. We believe that the utility of stemming should further be evaluated and that various stemming algorithms should be included in the testing process.

3.3.4 Conclusion

Our findings confirm that SNOMED CT satisfies our research requirements and that its coverage is sufficient for it to be used as an ontology of reality. The results also indicate that some other ontologies might be suitable for the same purpose as either 1) stand-alone ontologies of reality and/or as 2) combined with either SNOMED CT or other ontologies.

The open source Consumer Health Vocabulary (CHV) has shown as the best candidate for stand-alone use, at 78.23% of coverage. However, the CHV project seem to be not active, raising issues of sustainability of any artefact developed with dependency on this ontology. Also, documentation and peer-reviewed knowledge based on its structure and validity is lacking, making it difficult to understand its ontological qualities, a characteristic as important as coverage. NCI is also a potential candidate for stand-alone use. However, its strong focus on oncology and lack of coverage of other clinical areas, like ophthalmology or psychiatry has made us decide against it.

Combining sources seems a plausible option from the coverage point of view, combining SNOMED CT with the CHI would, for example, provide coverage of almost all concepts in our corpus. However, combining ontologies is not a trivial task and requires a significant number of experts performing that task. As we learned from Klein (2001), Pinto et al. (1999), Puhrer et al. (2010) and many others, issues are many, including technical and related to special knowledge.

The contribution of the testing coverage of SNOMED CT part of our project is the description and justification of the methodology that we used, which is a process that can be employed in many other projects. We believe that all projects that utilise ontologies for term expansion, concept recognition, coding or similar applications need to undergo an exercise similar to what we describe in the current work. We further believe that the use of SNOMED CT as a source of truth in these processes, in particular its use as ontology of reality, is novel and rapidly emerging and that the utility of SNOMED CT for this purpose will be increasing as the number of concepts and axiomatic relationship within SNOMED CT increase over time.

4 Design and develop artefact (DSR activity 3)

The DSR activity 3 defines an artefact capable of performing the work required for achieving the goal outlined in the DSR activity 2 for the purpose of addressing the problem explicated in the DSR activity 1.

We detail the process of designing the artefact in this chapter. The artefact developed by this project is an algorithm that is later integrated into a fully functional software solution and demonstrated in the DSR activity 4. We see the algorithm as a set of stages in the transformation of information from its original to its final form. The stages are graphically represented as the Figure 1 and each of the stages, along with the information transformation phases/methods will be detailed in a separate section.

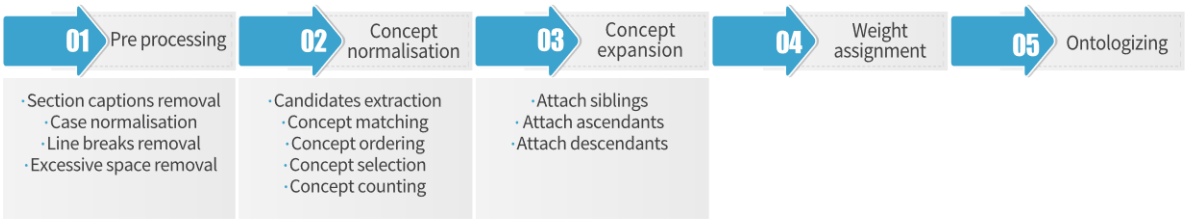


Figure 1 - Design and develop artefact (DSR activity 3) stages

4.1 Pre-processing used in this project

Our clinical corpus came as an XML file with 889 discharge summaries. Each discharge summary was distinct in the overall document and no XML was present within the discharge summary. We extracted each and every discharge summary from the XML document and uploaded it into an SQL server table one record per document. We used discharge summary ID as a unique identifier and indexed the table as such.

The text in the discharge summaries exhibited the majority of issues normally found in clinical text, called noise in the body of knowledge (Kaurova et al., 2011; Nguyen & Patrick, 2016). Examples of the issues include misspelled words, sparsity, medical measures and scores, abbreviations and grammatically incorrect sentences among others. Moreover, the text contained lines that had meaningless information, like a number or a date that had no explanation on its meaning attached. Another issue was frequent use of space character where

it does not belong, like in "(ACETAMINOPHEN)" or around interpunction characters, for example "diet , ambulating".

Each discharge summary in our corpus contains a number of headings that separate text into distinct sections. The examples of sections are "BRIEF HOSPITAL COURSE", "ALLERGIES", "PHYSICAL EXAMINATION" and "ASSESSMENT" among others. However, no rules were followed in including specific heading names, nor a list of expected heading names is prescribed for a discharge summary. In some discharge summaries, some of the heading names are misspelled. Although majority of the headings' names are written as uppercase, some are lowercase too.

Pre-processing requirements differ between projects (Turchin et al., 2006). The selection of methods deployed in pre-processing is customised to suit projects' specifics. In particular, specifics of corpus text as well as expectations of a concept normalisation algorithm used. Circumstances differ and some cases require minimal pre-processing (Wu et al., 2015), while some projects deploy complex and lengthy pre-processing methods (Friedrich & Dalianis, 2015).

We experimented combining natural language processing methods and found that our pre-processing produced best output when the process outlined as the Figure 2 is followed:

Figure 2 - Pre-processing process - phases

Some other methods that we tested in combination with the above are removal of punctuation characters, removal of numbers, stop words removal and removal of one-character orphan tokens. Removal of punctuation and number characters was abandoned as these characters are natural dividers of tokens and their removal introduced unexpected tokens, 2-grams and 3-grams in particular. The example is the text "vaginal bleeding , hemorrhoids , seasonal allergies" that normally produces 4 tokens: 1) vaginal bleeding, 2) bleeding 3) hemorrhoids and 4) seasonal allergies. If punctuation (commas) removed, 5 tokens would be extracted: 1) vaginal bleeding, 2) bleeding 3) bleeding haemorrhoids, 4) haemorrhoids and 5) seasonal allergies. We decided against stop words removal as the phrases like "Family history of

colorectal cancer" would not be recognised as a token as "of" is a stop word and would be removed in the process.

Section captions removal is used as a first step in the process. We took into consideration that majority of the captions were uppercase, can contain letters, numbers and other characters including spaces, that they are at least 3 characters long, that they begin at the beginning of the line, end at the end of the same line and that they usually end with either ":" or ";". Based on that the following is the regex pattern that we created to match section captions:

$$\text{^}[A-Z\backslash d\backslash W\_]\{3,\}?\,(?=(;|:)[\backslash s]\{0,\}[\backslash r\backslash n])$$

Where:

- $\text{^}[A-Z\backslash d\backslash W\_]\{3,\}$ means that the line begins with at least 3 characters that can be any character but lowercase letters
- $(?=(;|:)[\backslash s]\{0,\}[\backslash r\backslash n])$ means either : or ; followed by any number of spaces with the line ending with one of the new line characters

Next, function [Candidate_Token].ToLower() is used for bringing all text to lowercase (case normalisation). We were not concerned about capitalisation of tokens and we made sure that the function used for concept normalisation later took that into consideration too. As a result, matching to concept descriptions included [Candidate_Concept_Description].ToLower() transformation.

Line breaks removal and excessive pace removal used simple regex patterns "$(\backslash r\backslash n\backslash t|\backslash n|\backslash r\backslash t)$" and "$\backslash s\{2,\}$" respectively. Matches in both cases were replaced with single space. Both line breaks and excessive space were in some cases breaking ("disconnecting") sentences, hence decision to deploy these two methods. Bringing that to one space was "connecting" broken sentences again, e.g. "The patient was diagnosed with diabetes mellitus" became "The patient was diagnosed with diabetes mellitus". The methods were also not "connecting" normal sentences as these are properly terminated by full stop characters, e.g "The tests performed today were reassuring . Your chest scan did not show any evidence of blood clot or pneumonia ." became "The tests performed today were reassuring . Your chest scan did not show any evidence of blood clot or pneumonia ."

4.2 Concept normalisation and concept frequency

Natural language is known to contain a great deal of lexical variations in its vocabulary (McCray et al., 1994). Free text, as a form of presentation of natural language exhibits that same characteristic. For example, a clinical concept written in a discharge summary can be in one of its morphologically different forms (e.g. “suturing” and “closing by suture”) or be in one of its orthographic variations (e.g. “haematuria” and “hematuria” or “eye-patch” and “eye patch”). These anomalies, among others, make free text processing by automated agents difficult.

The purpose of concept normalisation is to recognise clinical concepts in a body of text. The output of concept normalisation is SNOMED CT code, one per distinct candidate extracted from text. Moreover, occurrences of distinct candidates are counted and recorded as a frequency of a concept and used in calculating concept’s representativeness of a body of text later. The process that is used to normalise concepts and count their frequency is presented as the Figure 3:

Figure 3 - Concept normalisation/concept frequency process phases

Grams of up to 3 words in length are extracted in the candidates extraction step of the concept normalisation process. Minimum length for a first word of each gram is selected to be 3 characters. The result of the candidates extraction are 1, 2 and 3-grams that are ready for the candidates matching step. A candidates extraction function “crawls” through the text and extracts every phrase of length x where $x \in \{1,2,3\}$. The function visits all words in the text and is not concerned with the meaningfulness of the candidates it extracts. The function does not skip words to extract 2 or 3-gram candidate but does stop if it encounters one of the gram-boundary characters. The gram-boundary characters used for boundary detection are punctuation characters and numbers, including numbers/letters combinations (e.g. “22yo”). A special case is when a dash is found between two words, when two words and a dash are considered a gram (e.g. “full-range” remains the same). We store the candidates in the List type object rather than a hash type object as we want to preserve duplicates.

The concepts matching process involves matching of each of the candidates to the descriptions of the concepts available in the SNOMED CT ontology. Firstly, an exact match is attempted

for each candidate from the list created in the candidates extraction step. An exact match is a match of lowercase modalities of strings A and B, where each and every character of the string A matches equally positioned character of the string B and vice versa. The example of matched strings are phrases “diabetes mellitus” and “Diabetes Mellitus”.

If a match is not found after the attempt to find an exact match, matching is attempted using [concept_description].StartsWith([candidate]) function. In this case, the candidate “diabetes” is matched to a description “diabetes mellitus” of a SCTID: 73211009 and description “diabetes type” of the concept SCTID: 405751000.

If a match is not found and a candidate is 1-gram, the candidate is stemmed and another match with concept descriptions is attempted, this time checking whether a concept description starts with a stem of a candidate string. The example is the 1-gram “vaccinate” that when stemmed using Porter stemmer becomes “vaccin” and then matches SNOMED CT concept description “vaccination”, which is acceptable synonym of a concept SCTID: 33879002, Administration of vaccine to produce active immunity (procedure). Due to the computational limitations that we face in this project, candidates other than 1-grams have not been stemmed. The function used for this is the function [concept_description].StartsWith([stem]).

It is not uncommon that more than one concept description match is found for a single candidate in this step. That usually means that a list of concepts is selected as a match to a candidate. To select a single matching concept for a candidate, matched concepts are ordered (concept ordering) and the top concept is selected (concept selection). These two steps in the process are explained next.

For concept ordering and selection, we considered two methods. The first method involves selection of a concept based on its centrality. This is not a new method as it has been made famous by the Page Rank algorithm (Page et al., 1999) that has subsequently been used and modified by many (Berkhin, 2005).

This method creates a graph out of the matching concepts positioning them as nodes. Concepts connections in SNOMED CT, both direct and indirect, are used as edges. Direct connections are where concepts representing given nodes are connected in SNOMED CT using Is_A attribute (SCTID: 116680003), while indirect are when concepts that represent given nodes are connected using Is_A attribute through other SNOMED CT concepts. An example of indirect connection is connection between SCTID: 201967009, Allergic arthritis of the hand connected

to SCTID: 3723001, Arthritis through SCTID: 16935003, Allergic Arthritis. That can be read as SCTID: 201967009 Is_A SCTID: 16935003 Is_A SCTID: 3723001.

The measure selects a single matching node that had the greatest number of connections. However, this method was not implemented because of its computational requirements and our hardware limitations. Also, we believe that the artefact needs to be extended using this method to achieve its full potential. The final computation needs to take in consideration not just the number of connections, but the frequency of candidates represented as nodes, the number of hops from each of the connecting nodes (distance measure) and weights of nodes connecting to the node in question. Implementing this extension would exceed the scope of this project and extend the timeframes that we had available. This assumption is based on the estimation of work required for development of the extension, testing and assessment of the newly extended method. However, this is certainly worth pursuing as future work towards enhancing the utility of this research.

The second method considered to order matching concepts and select a single matching concept will be explained next. We implemented that method in our algorithm. The method orders concepts based on their weight and selects a concept that has the highest weight. The weight of a concept is calculated as a multiple of concept's frequency in the document and concepts axial weight. Concepts' axial weights are pre-defined by the user and sent as input values to the algorithm. The user defines them based on his/her interest in the type of information contained by the SNOMED CT axis (branch) that the particular concept belongs too. The logic behind that is that a user should be able to select what type of information in the document is of her/his interest, so the algorithm is able to give it preference.

The SNOMED CT concept model provides 19 axes branching off the main concept SCTID: 138875005 (SNOMED International, 2020c). Each of the branches starts with a top-level concept and is distinct in terms of types of concepts that it contains. For example, the Clinical finding axis starts with a concept SCTID: 404684003, Clinical finding (finding) and contains concepts that are result of a clinical observation, assessment or judgment. On the other hand, the Social context axis starts with a concept SCTID: 48176007, Social context (social concept) and contains concepts that represent social conditions and circumstances significant to health care. Therefore, a user interested in finding information relevant to clinical findings should put more value in the Clinical finding axis than in Social context axis.

Concept counting is the last step in the algorithm. The concepts are counted as a sum of the frequency of candidates. The example is as follows: The candidate “vaccinate” has matched the concept SCTID: 33879002 and appeared 3 times in the text. The candidate “vaccination” has matched concept SCTID: 33879002 and appeared 6 times in the text. That results in 9 as the total frequency of the concept SCTID: 33879002.

The outcome of the Candidates matching process is a dictionary that contains pairs containing one concept per matched candidate and their frequency in the text represented by an integer. Considering that we match all candidates, regardless of their meaningfulness, to all concept descriptions in the SNOMED CT, we coin this type of candidates matching “Brute force candidates matching”.

4.3 Concept expansion

Each of the concepts discovered in the concept normalisation phase are expanded in the concept expansion phase. The purpose of concept expansion is to present a concept in question as it is positioned in its segment of the ontology of reality. We argue that the concepts surrounding the concept in question are important for conveying the concept’s meaning, and that the closer the concepts are the more important they are for that. The concepts surrounding the concept in question are ascendants, descendants and siblings. For us, sibling concepts are concepts attached to all first ascendants of the concept in question. First ascendants are also known as parent concepts and first descendants are also known as children concepts.

We used SNOMED CT, represented as a graph, as ontology of reality for the reasons described earlier in this document. The result of this process is a graph in which concepts are nodes and relations between concepts, as defined in the SNOMED CT, are edges. Figure 4 shows the phases of the process of concept expansion:

Figure 4 - Concept expansion process phases

Due to our computational limitations, we attach only one level of ascendant concepts and one level of descendant concepts to the concept in question. Also due to computational limitations, we do not attach concepts that are connected to the concept in question with attributes other than the Is_A attribute. We assume that attaching more than one level of ascendant and descendant concepts, as well as attaching concepts connected to the concept in question with other than Is_A attributes, would increase the accuracy of the process. However, experimental validation is required to test these assumptions that we are unable to provide, due to technical limitations. However, we allow for weight assignment provision in our application (Grapher) that we created to conduct this experiment.

The result of this process are separate graph structures created around each of the matched concepts.

4.4 Weight assignment

Each of the concepts in each of the graph structures created in the concept expansion phase are assigned weight in the weight assignment phase. In this work, the weight of a concept is a measure of its importance to the process of annotation. The weight calculated in this phase is used for calculation of the final weight of the concepts in the next step when the graph structures are merged (as described in the section 4.5 Ontologising).

Weights of the central concept of each of the graph structures and the concepts that are introduced in the process of concept expansion are calculated differently. We decided that the importance (weight) of the concept found in the text is a function of importance placed on the axis that the concept belongs to and the frequency of that particular concept in the text. The importance, or weight, of the axis is pre-defined by the user and a provision is made for that in the interface of the Grapher application created as part this project.

The weight of a central concept W_{cc} is calculated as follows:

$$W_{cc} = W_{Axe} * F$$

Where:

- W_{Axe} is concept's axial weight - the weight assigned to the axis that the concept belongs to

- F is the frequency of the concept – the number of times the concept appears in the text

We consider that the importance of the concepts added in the process of expansion should depend on the importance of the concept that they expanded and their position in the hierarchy of the reality in relation to that concept. However, we had to take in consideration the warnings that point to deviations in SNOMED concept distributions (Rector et al., 2011) and moderate for the number of concepts found in the groups of ancestors, descendants and siblings and included another variable to reflect that number. In other words, the weight of the concepts that are added in the process of expansion is a function of weight of the central concept of the structure that they belong to, a static parameter pre-defined by the user and the number of elements at the same level. The formula for calculating weight of the expanded concepts W_{ec} is as follows:

$$W_{ec} = \frac{W_{cc} * L_x}{\mathrm{N}} \text{ and } L_x \in \{L_a, L_d, L_s\}$$

Where:

- W_{cc} is the weight of the central concept,
- L_a is pre-defined level parameter for ascendant concepts
- L_d is pre-defined level parameter for descendant concepts
- L_s is pre-defined level parameter for sibling concepts and
- N is the number of concepts found on that particular level

We believe that the weight of the concepts should decrease with their distance from the central concept, hence we propose that the L_x is assigned to be proportional to the level's distance to the central concept and we leave provision in our software to do so. However, due to the computational limitations in our project, we did not include levels of ascendants and descendants more distant than parent and children levels.

4.5 Ontologising

What we call ontologising is merging of graph structures created in the process of concept expansion. At this point, all of the concepts that form these structures have weights assigned.

The reason for merging graph structures created by expansion is because these graphs are in many cases overlapping segments of reality. The example is the graph that has been formed around the concept SCTID: 73211009, Diabetes mellitus and the graph that has been formed around the concept SCTID: 271327008, Hypoglycemic syndrome. Both of these graphs will have concept SCTID: 126877002, Disorder of glucose metabolism as ascendant and concept SCTID: 80394007, Hyperglycemia as a sibling.

The goal of the ontologising phase is to merge overlapping segments of the reality into one segment that will be represented as a graph and whose nodes will be weighted taking into consideration the weights that the nodes had in the individual graphs. Each of the nodes in the merged graph will have weight equal to the sum of all of the nodes in the pre-merge graphs that represent the same concept. The formula to calculate concept weight (W_c) in the merged graph is as follows:

$$W_c = \sum_{k=0}^{n} W_{c_k}$$

Where:

- W_{c_k} is weight of a concept C in the graph k
- n is number of graphs

The equation reads: W_c, or weight of the concept C is a sum of weights of the concept C in all of the graphs where the concept C is found.

The result of a merge process is a graph that does not necessarily have all segments connected. That is because we excluded the ultimate root concept SCTID: 138875005, SNOMED CT Concept from the ontology of reality as we wanted to prevent nodes belonging to different axes from connecting.

At this stage, the document is completely annotated.

5 Demonstrate artefact - DSR activity 4

The DSR activity 4, Demonstrate artefact, presents the artefact created as part of the project that illustrates a real-life use case, also called a "proof of concept". In this project, considering that the artefact is an algorithm, we created a software application that utilises the artefact in part of its operation. The same software application is used for the evaluation of the utility of the artefact as well as for suggesting some future work and novel applications of the algorithm. We named the application Grapher.

We start with the description of the cloud platform, virtual machines, software used for visualisation, Cytoscape (Shannon et al., 2003), and GraphML (Brandes et al., 2013) data format used for building graph information models used by the graph visualisation platform. We then describe the corpus of free text clinical documents used as a sample in this project. Finally, we present the Grapher's interface and its functions.

5.1 Cloud platform and virtual machines

The platform we used to create and test the Grapher as well as to evaluate the artefact is the National eResearch Collaboration Tools and Resources (NECTAR) cloud platform (Australian eResearch Initiative, 2021). Access to the platform is provided by the University of Auckland (UoA), as a participating organisation. Authentication to the cloud services is managed by the REANNZ Tuakiri (REANNZ, 2021) and the eduGAIN, international inter-federation service (eduGAIN, 2021). Connectivity to the cloud services was through a Virtual Private Network (VPN) endpoint connection provided by the UoA. Connection software was a free version of the FortiClient client, version 6.4.0.1231 (Fortinet, 2021).

We actively used 2 virtual machines (VMs), one as the development/testing environment and one as the database server. The development VM had 4 virtual central processing units (VCPUs) Intel Skylake 2.2Ghz 64Bit, 8 GB RAM, 60GB root disk drive and 150 GB additional drive that was utilised as storage space. The database hosting VM configuration included 4 VCPUs, 8 GB RAM, 30GB root disk drive and 200 GB additional drive that was used to host the database.

Although the VMs have fulfilled their role in the development process, the issues experienced during the time of development are worth mentioning here as information to future researchers using the NECTAR platform provided through UoA.

The main issue was the limitation placed on configuration of the image that the VMs are built from; in particular, storage space of the root drive and memory size. The development VM originally had only 30GB root storage drive. That was increased to 60GB by NECTAR technical services after the VM exhibited operating system instability and subsequently failed to boot due to insufficient storage space on the boot partition. The VM needed to be recovered and re-built, which has taken significant time and effort.

Memory limitation of 8 GB has imposed some constraints to how Grapher operated. Grapher is a memory dependent application that utilises all CPUs in some of the processes where parallel execution was possible. The information artefacts that the Grapher produces and manages are typically 2-5 GB in size. The operating system used was 64Bit and the size of the information artefact that could be stored in memory was not constrained to 4GB as it is the case with 32Bit operating systems. A good example of a large information model stored in memory is the SNOMED CT graph that requires just over 2GB of memory.

However, large information artefacts used at the same time meant that not all artefacts could fit in the RAM memory and use of the swap partition was common. Unfortunately, use of the swap partition slowed down Grapher execution significantly and meant that we had to come up with solutions alternative to keeping all required information artefacts in memory. One of the solutions was minimisation of in-memory caching of segments of graphs resulted from concept expansion. The other was outsourcing some demanding work to the database server, through use of SQL views, functions and stored procedures, instead of using functionality available in the .NET's System.Data namespace.

Therefore, we believe that removing limitations on VM configurations would improve utility of the NECTAR platform in projects utilising large information artefacts.

5.2 Development environment software

The operating system used on both VMs was 64 Bit Windows Server 2012 Standard R2. The database hosting VM had Microsoft SQL Server 2008 R2 (SP3) installed with Microsoft SQL

Management Studio (SQLMS) used for management and coding purposes. The development VM had Microsoft Visual Studio 2017 (VS2017) application used for coding, and Cytoscape application used for graph visualisation purposes installed. We had to enable gcAllowVeryLargeObjects parameter in the App.Config file in the VS2017 for the objects larger that 4GB to be able to load into memory.

Grapher was developed in C# and has a strong SQL code component developed outside of VS2017, using SQLMS. The artefacts developed using SQL are SQL functions, SQL stored procedures and custom SQL types. The combination of SQL functions and SQL types was particularly useful in outsourcing work to the SQL server. For example, a custom SQL type like the following was useful in sending significant workload from development to SQL VM:

```
CREATE TYPE tpe_S_S AS TABLE
(
	[Key]  VARCHAR(50) NOT NULL,
	[Value] VARCHAR(MAX) NOT NULL
	PRIMARY KEY ([Key])
)
GO
```

Figure 5 - SQL Custom Table Type example

The following is the function where the type was utilised:

```
PROCEDURE [dbo].[Match_Connections_To_SubGraphs_Dic]
	@tpe_S_S READONLY
AS
BEGIN
	SELECT A.[Key] AS [Source], A.Value AS Connection,
		C.End_Node AS [Target] FROM @tpe_S_S AS A
	INNER JOIN
		All_Paths_SNOMED_Codes AS B ON B.SNOMED_CODE=A.Value
	INNER JOIN
		All_Paths_To_Root_For_End_Nodes AS C ON B.Path= C.ID
	GROUP BY A.[Key], A.Value, C.End_Node
END
```

Figure 6 - SQL Table-valued function where custom Table Type is used

This function would accept an entire Dictionary<String, String> as a parameter, rather than one by one KeyValue<String, String> sent as 2 SQL parameters. That means that, for a Dictionary

artefact of 10,000 entries, instead of 10,000 calls to the database, one can make a single call and wait for the SQL Function shown in Figure 6 to return a complete dataset as one result. This is especially useful in cases where the process cannot be conducted as asynchronous, meaning that the execution of the main thread would stay blocked for a long time – and 10,000 calls to the database can take a long period of time to complete rendering the application unresponsive.

Unfortunately, open source C# graph manipulation frameworks are sparse, and some of the available frameworks are poorly maintained. We utilised QuickGraph 3.6 framework (de Halleux, 2014/2020) that is a mature open source framework that provides generic directed/undirected graph data structures and algorithms. However, for some of the operations, the framework was either too demanding in terms of processing or lacked classes specific to some of the rather uncommon tasks, so we created custom structures (C# classes) according to our specifications.

As an example, we found that serialisation functions available in some of the QuickGraph classes from QuikGraph.Serialization namespace took a long time to execute and were creating textual structures that were unnecessarily large. The exact example is the function [graph].SerializeToGraphML< CustomClass, Edge<[CustomClass]>, AdjacencyGraph< CustomClass, Edge<[CustomClass]>>>([writer]) that was slow and in its generic form produced an output that was far larger than necessary.

At the same time, serialisation of a graph "outside" of the class, using Newtonsoft.Json or serialisation methods from the .NET's System.Text.Json namespace could not be used as some of the native QuickGraph classes used by the graph internally were not marked as serializable. Binary serialisation was not allowed by design (Dtosato, 2014). Since we had to conserve processing power and minimise time required to complete some of the operations, we decided to create classes that will provide custom serialisation for us. An example of such a class is given in the section 9.3 Custom graph class**.**

By design, QuickGraph framework does not have ability to visualise graphs. However, visualisation was important for us in the processes of troubleshooting and assessment of the created structures. To make visualisation possible, we included an application called Cytoscape (Shannon et al., 2003). The last version of Cytoscape used is 3.8.2 based on Java 11.0.6.

Cytoscape was created for the purpose of visualizing molecular interaction networks and gene expression profiles among other state data. It is an open source application created in Java that import several textual graph formats including JSON (Sporny et al., 2014) and GraphML (Brandes et al., 2013, 2002) as sources. The application was installed on the development VM and used to open and display textual files that represented graph outputs created by the Grapher application. There was no run-time integration between the Grapher and the Cytoscape, they operated as separate applications only.

Our format of choice was GraphML for Grapher outputs, as it was among the formats supported by the version of Cytoscape that we started with at the beginning of the project. JSON was not available as an option in Cytoscape at that time.

GraphML is an XML based format that has <graph></graph> element in its source. All attributes of the graph are supplied as direct attributes of that element and its edges and nodes are supplied as an unordered list of elements within the <graph></graph> element. Obligatory for node elements is to have id attribute and that the id attribute be unique among all elements. Edge elements, apart from the id attribute, must have source and destination attributes. The values of both source and destination must be valid node ids.

Cytoscape allows for custom arguments to be assigned to both nodes and graphs and later used in assigning visual arguments. For example, nodes can be assigned argument xyz and that argument could be assigned as a node colour argument in Cytoscape. Cytoscape would then use all valid colour values (name, hex and RGB) of the xyz argument as visual colour attributes of a node visualised as part of the graph. Other attributes that can be assigned include sizes, positions, and shapes.

To present a use-case for the use of Cytoscape in our project, we next present the textual and visual representations of a graph that contains two groups of nodes that are compared for similarity in the evaluation stage of our project. An example of the GraphML output can be seen in the EHR GraphML code section.

And the graphical output of the code is as follows:

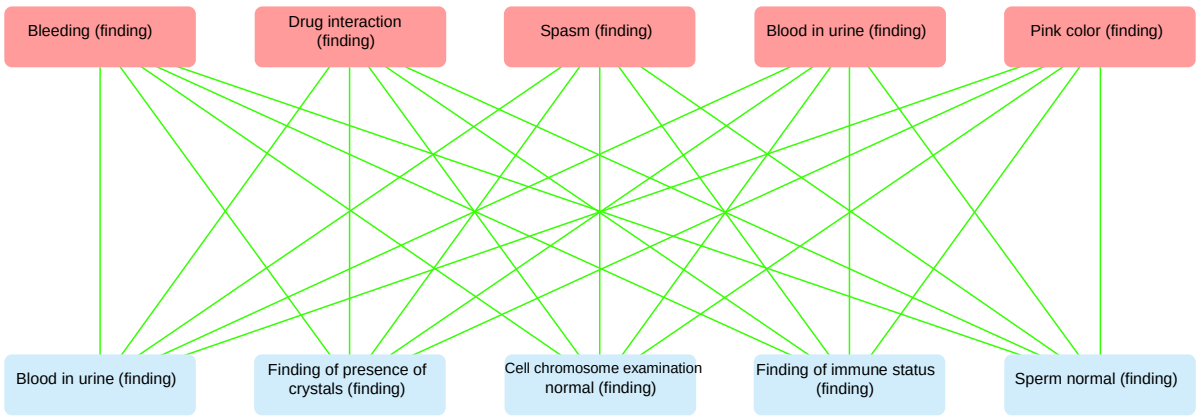


Figure 7 - Graphical representation of the GraphML code created by the Grapher application

The graph's visual appearance is controlled by the style sheet that has been created for this particular layout. The role of the style sheet is to define the layout of the graph elements on the screen so that they are easy to read for human agents. Making relations between graph elements easy to understand enabled us to be able to troubleshoot efficiently. We include the code for the style sheet in the section 9.5 Style sheet used in Cytoscape.

The Grapher application is developed as a Windows Presentation Foundation (WPF) application. WPF is a sub-system of .NET Framework. Grapher is a three-tier application that has a relational database, Microsoft SQL, as its backend, mid-tier developed in C# and front-end as Extensible Application Markup Language (XAML). The mid-tier and backend communicate using tools found in the System.Data and System.Linq namespaces.

C# was chosen because of the main author's experience with that programming language. The second language considered was Java. Although Java had more advanced libraries for working with graphs, including JGraphT (Michail et al., 2020), Google Guava (Google, 2014/2021) and Apache Commons (Apache, 2021), the learning required to make an application as complex as Grapher in a programming language and programming environment (Eclipse) that the main developer was not familiar with, was significant; hence the decision to use C#.

The Grapher is an application primarily created to demonstrate utility of the artefact developed in this project. However, some of the functionality of the Grapher is specifically developed to complete other tasks. Examples of the tasks include evaluation of the utility of the artefact, transformation of textual artefacts into formats easy to consume by the Grapher's functions and

caching of information models that were complex to re-create every time the application is started, or evaluation performed. The functionality of the Grapher that is built to demonstrate utility of the artefact will be touched upon in the next section. The other major functionality of the Grapher, evaluation of the artefact, will be discussed in the chapter 6 Evaluate artefact (DSR activity 5).

5.2.1 Demonstration of the artefact's utility

Demonstrating utility of the artefact includes presenting outputs of the relevant phases of the information transformation process. We chose to annotate two corpuses of documents, the corpus of EHR discharge summaries that we used in the evaluation process and a set of openEHR information models. We chose to annotate openEHR information models to demonstrate that the artefact is not specific to a particular corpus (e.g. discharge summaries), but that it can also work on other corpuses, including corpuses containing structured information, like openEHR archetypes.

The user interface provided by the Grapher used a combination of tab and listview controls that are both an integral part of the native .NET framework. The front tab of the user interface shows the outputs of the information transformation of a selected EHR document, as the document header and body outputs as well as the results of the transformation of the openEHR Archetypes that appear to be semantically similar to the selected EHR document. The tab "Weights" is the tab where the weights of the elements from the specific SNOMED CT axes are defined. That tab will be touched upon later in this section.

The visual appearance of the Grapher's front tab showing the EHR with ID = 17 is as follows:

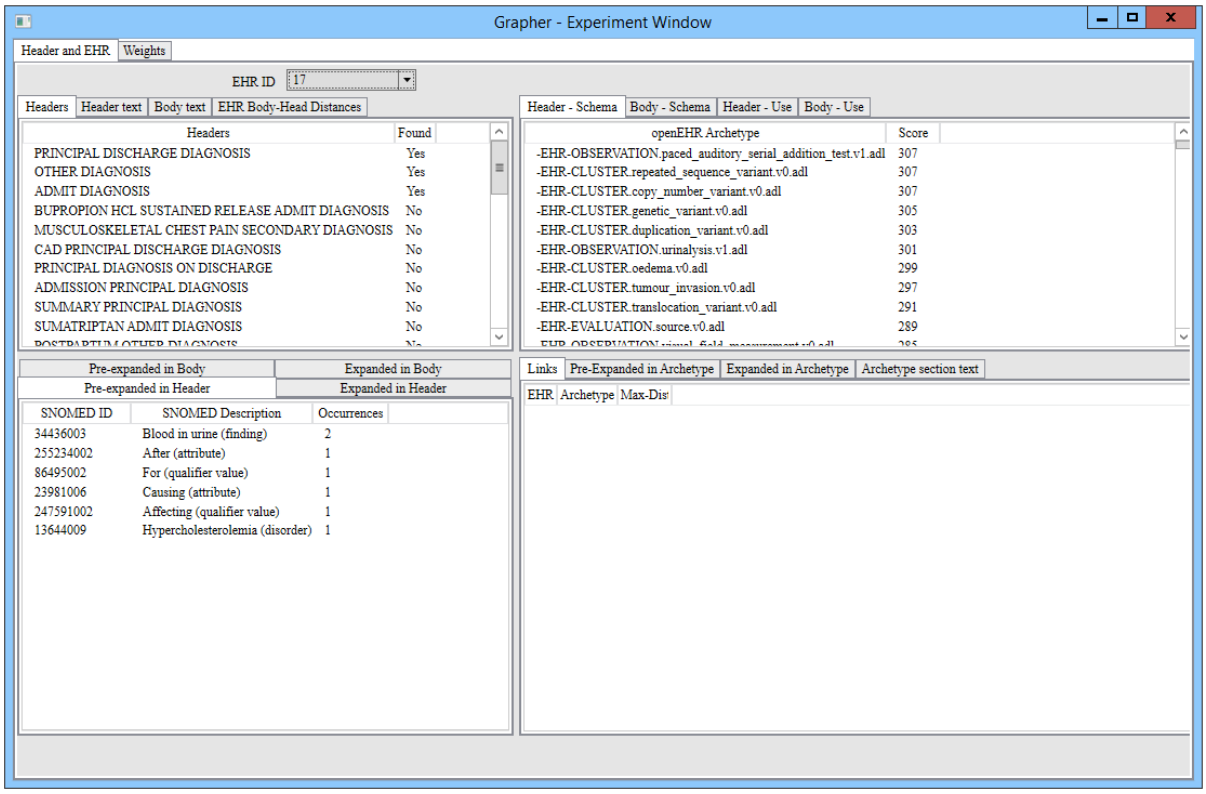


Figure 8 - Grapher user interface I

The dropdown control showing 110 on Figure 8 is the selected EHR ID. Upon selection of another EHR ID in that control, the data shown in the other controls change to the outputs of the selected EHR document.

The tab titled “Headers” shows the names of all diagnosis headers that are found in all documents in the corpus. The items on that list that are marked with “Yes” are the diagnosis headers that are found in the document whose outputs are displayed (e.g. document ID = 110 in this case).

The tabs “Header text” and “Body text” show the content of the diagnosis headers (“Header text”) and the body of the document (“Body text”). These tabs contain the free text that is annotated using information models created using the artefact (an algorithm) created in this project. The text is presented in its original format, as it is before the pre-processing phase of the process defined in the algorithm.

“EHR Body-Head Distances” tab shows the listview control with the list of concepts that are the final output of processing by the artefact (Figure 9). The descriptions of the concepts are

shown as they are compared with their distances in the SNOMED CT graph listed in the listview's most right column.

The group of tabs in the lower left corner of the front tab contains information generated in the process of coming to the output shown in the "EHR Body-Head Distances" tab. The lists show the tabs containing codes and descriptions of the concepts found in both, the diagnosis sections and body sections of the EHR. "Pre-expanded" means that the concepts shown are as found in the free text, before being expanded and "Expanded" means that the concepts shown are as they appear after expansion has been completed. In other words, "Expanded" tabs show the concepts that are part of the information model that is an annotation of a given text (Figure 9).

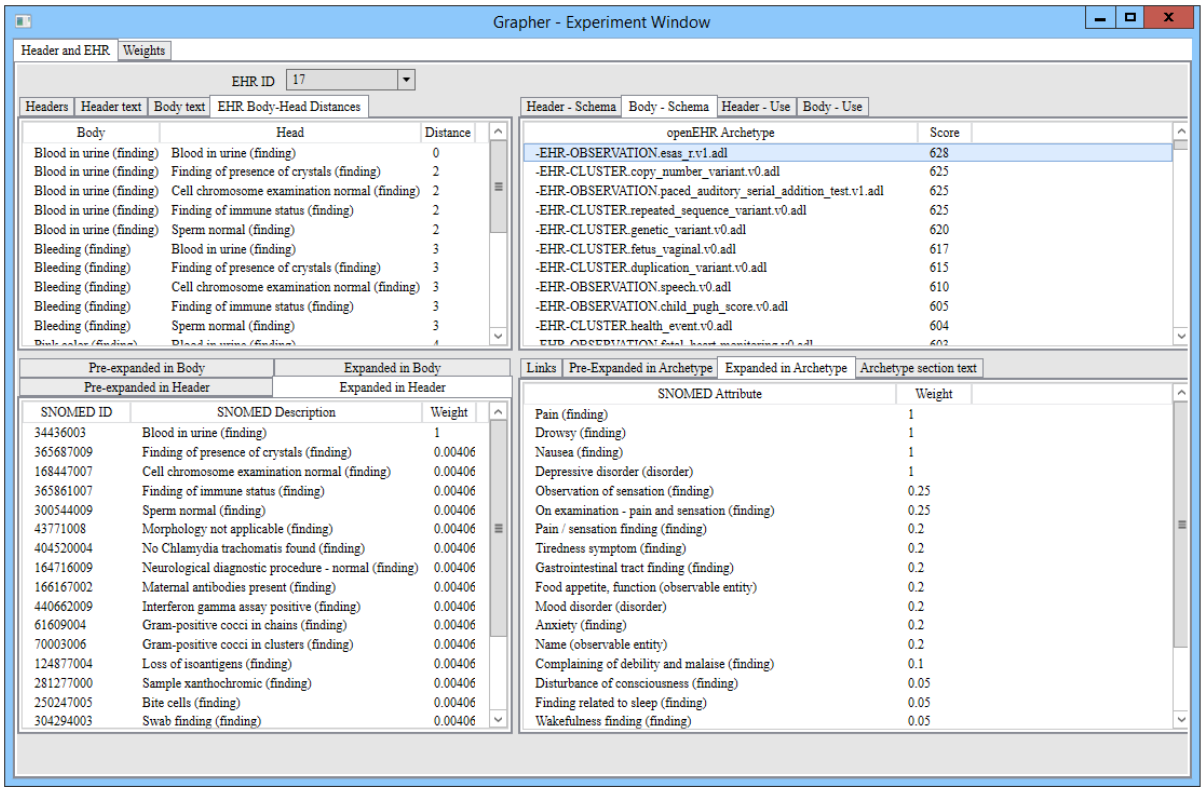


Figure 9 - Grapher user interface II

This interface shows all relevant stages of the information transformation as defined in the artefact created in this project. Firstly, free text input is shown in the group of tabs in the top left corner ("header text" and "Body text" tabs). Secondly, concepts found in the free text using only NLP techniques and before expansion are shown in the tabs in the lower left corner ("Pre-expanded" tabs). And lastly, the same section of the front tab contains the listviews with the list of concepts that are the final annotation of the concept whose ID is selected in the dropdown menu shown at the top of the front tab. The "EHR Body-Head Distances" tab shows the result

of the comparison of the concepts found in the diagnostic sections and the body of the EHR document whose ID is selected in the dropdown.

The tabs shown on the right hand side of the front tab show results of the comparison of the selected EHR document and the openEHR Archetypes that we collected from the openEHR repository (openEHR, 2015). OpenEHR Archetypes are information models that specify reusable data points that describe constructs that either directly or indirectly belong to the clinical domain. Examples of Archetypes that describe constructs that belong to the clinical domain are blood pressure, stroke risk and ECG result. Examples of Archetypes that represent constructs indirectly related to the constructs that belong to the clinical domain include ones that represent address, dwelling or absence of information.

openEHR Archetypes are selected as they are expected to be a rich source of information describing clinical constructs that are commonly found in free text clinical documents, for example discharge summaries that we use as a free text corpus in this project. Another reason why openEHR Archetypes are selected is because they are standardised, composite information models that we see as segments of reality – rich, multidimensional descriptors of the topic that they represent. In the chapter 7 Future work, we suggest that this topic is explored further, and we give some guidelines as to how this should be approached.

The goal of our testing of utility of openEHR Archetypes in annotation of clinical text is to deploy our algorithm and find Archetypes semantically similar to some or all of the messages communicated in the clinical text. openEHR Archetypes have two sections that we considered valuable for recognising the semantics of the concepts they represent, the Archetype's schema and the Archetype's Use sections.

An Archetype's schema is a graph of Archetype sub-concepts that clinicians, who are designers of openEHR Archetypes, consider relevant to the larger concept described by the Archetype. For example, the Blood Pressure Archetype contains sub-concepts like Systolic and Diastolic, Pulse Pressure and other relevant concepts that can be valuable in the process of annotation that follows the paradigm prescribing use of sections of reality as annotation models that we posit in this work. The structure of the Blood Pressure Archetype is presented in Figure 10. The entities listed under the Events, Protocol, State and Description branches are not very useful in terms of semantics of an Archetype and they are ignored in the process.

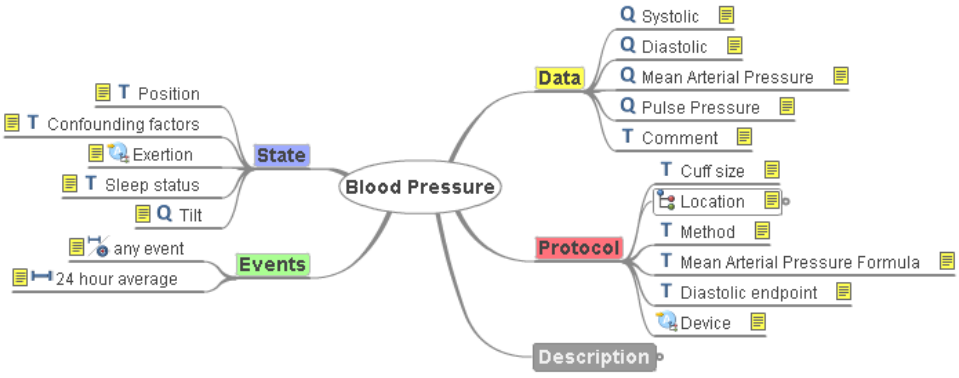


Figure 10 - Blood Pressure Archetype - adapted from openehr.org/ckm

The Grapher application lists all Archetypes that match the diagnostic section (header as we call it) and Body of the EHR document in the top right corner of the front tab (Figure 8 and Figure 9). We conduct matching to both Archetype sections, Schema and Use, hence four tabs are in total in that area of the front tab. The captions of the tabs are self-explanatory.

In the lower right section of the front tab, we present 4 tabs, "Links", "Pre-Expanded Archetype", "Expanded in Archetype" and "Archetype section text".

The section Links show the distances between the concepts found in the Archetype after expansion and the concepts found in the EHR document after expansion (Figure 8). The distance scores $InvDist$ shown in the third column in the listview control in this tab are the numbers that are the results of the following calculation:

$$InvDist = MaxDist - Dist$$

Where:

- $MaxDist$ is a maximum distance between any two concepts in the two lists
- $Dist$ is a distance between concepts

The first 5 concepts recognised in the expanded Archetype are compared with the first 5 concepts recognised in the expanded EHR document. Double clicking on an Archetype name reveals further detail in the tabs below.

The "Pre-Expanded in Archetype" tab contains a listview control that shows a list of descriptions of the concepts that are found in the Archetype prior to the expansion process. The

list shows the number of concepts found as well. The tab “Expanded in Archetype” shows the list of descriptions of the Archetype concepts after expansion along with their weights (Figure 9). The list shows 20 concepts. However, due to computational limitations, only the first 5 are used in the comparison process. The “Archetype section text” tab shows text entries of the Archetype schema.

Weights can be adjusted on the “Weights” tab. The controls on the tab provide data entry functionality upon double clicking on any of the entries. The adjustments can be made for 1) Roots (axis weights), 2) Ascendants (5 levels), 3) Descendants (5 levels), 4) Siblings and 5) Attributes. The Root and Attribute weights show both default and last used weights. The interface is as on Figure 11.

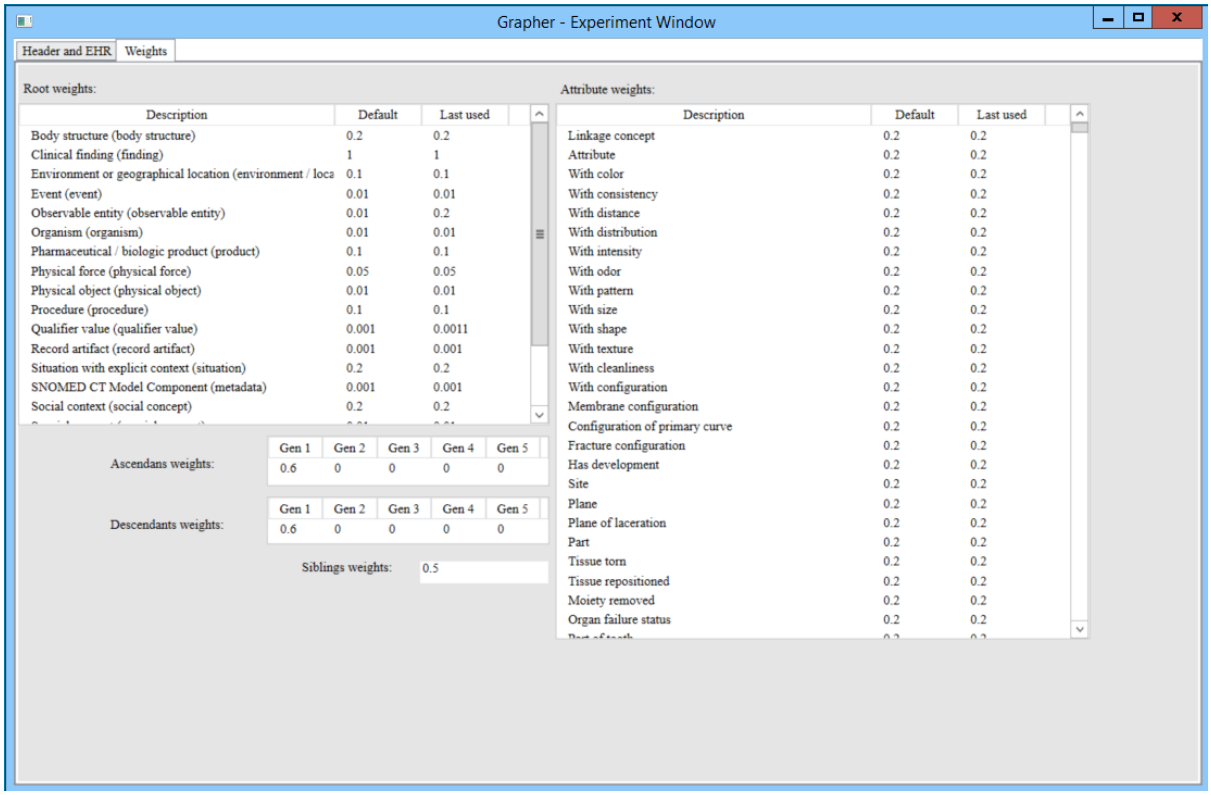


Figure 11 - Grapher user interface III

Ability to change weights enables the user to change the type of concepts that are found in the texts. For example, if the focus of the search is finding messages that contain concepts that represent procedures, rather than clinical information, then the Root weight of the Procedure axis would be increased. That would make the process of expansion be more biased towards the concepts that belong to the Procedure axis (that have SCTID: 71388002, Procedure concept as a root) in the process of assigning importance (weight) to the concepts.

The weights of ascendants, descendants and siblings concepts are also used in the process of expansion, hence changing them will impact the selection of the concepts used as expansion elements. As an example, if the focus is on revealing concepts with more detailed meaning, like SCTID: 73211009, Diabetes mellitus, rather than SCTID: 20957000, Disorder of carbohydrate metabolism, the Descendant weights should be set as proportionally higher than the Ascendant weights and the number of generations in the Descendant weights section with the weight above zero should be higher too.

The connections between the concepts in SNOMED CT are based on SNOMED CT attributes and they are listed under the SCTID: 106237007, Linkage Concept axis root. The Attribute weights list shows all possible attributes and assigns weights value to each one of them. In this project, we use only SCTID: 116680003, Is_A attribute due to the computational limitations we encountered. However, we believe that including further attribute connections in the process of expansion will improve the results and we suggest in the section 7 Future work that this claim is tested for validity.

For demonstration, the Grapher reads the pre-cached results of the concepts discovery, expansion and comparison as the real-time process would involve impractical delays. The weights as we present them in Figure 11 are used in the process of expansion and are the weights that the evaluation is based on.

6 Evaluate artefact (DSR activity 5)

The DSR activity 5, Evaluate artefact establishes how well an artefact fulfils the goal of the project that is outlined in the DSR activity 2 and to what extent it can solve, or alleviate, the problem as ascertained in the DSR activity 1.

In this section we detail the process of evaluation, outline the methods used in the process and show the evaluation results, before the results are discussed and the conclusion is made that the goal of the project has been achieved. However, to make the evaluation possible, significant preparations had to be made, including creation of new artefacts and imposing of some limitation on what we do. The artefacts include algorithm for creation of SNOMED CT graph from RF2 text-based files, algorithm for finding shortest path in the SNOMED CT graph that overcomes technological limitation that prevented us from direct application of available algorithms and the algorithm used for concept expansion. We describe that in the section 6.1 Prerequisites, before we report on the evaluation process results and methods.

6.1 Prerequisites

As described in the section 6.1.1 Creating SNOMED CT graph, SNOMED CT is made available as a set of RF2 files. RF2 files are flat text files and for use in this project, they had to be converted into a graph format and serialised for future use, as conversion on as-needed basis would be time demanding and impractical. Moreover, although the project utilised high-end technology, some of the computations, integral and not integral to the main algorithm were time demanding and could not be executed without extending evaluated software operation time beyond reasonably expected.

The example is the algorithm for finding shortest paths in the SNOMED CT graph. Although mature algorithms exist, the process of finding a shortest path is computationally demanding and its processing time impact application performance. For that reason, we had to model a new shortest path algorithm that splits the graph to manageable subgraphs.

Another example of excessive computational requirements is concept expansion process. That process involves re-creation of a section of reality around a concept that is being expanded and, depending on the number of nodes involved in the process, the time demands could be

excessive. Hence, we had to experiment with the expansion process and impose some limitations on how many levels are included in the expansion.

We next outline each of the prerequisites before we detail the evaluation method.

6.1.1 Creating SNOMED CT graph

SNOMED CT comes as a set of flat, tab delimited text files, encoded in UTF-8. The file specification used is called Release Format 2 (RF2) and was specified by SNOMED International. In this work, we used SNOMED CT US Edition version 20190301.

The SNOMED CT release package comes with 3 release types: full, delta and snapshot. A full release is a release type in which the release files contain every version of every component and reference set member ever released. A delta release is a release type in which the release files contain only rows that represent component versions and reference set member versions created since the previous release date. And a snapshot is a release type in which the release files contain only the most recent version of every component and reference set member released, as at the release date (SNOMED International, 2020a). We used the snapshot release type in our work as we needed the most recent versions of the components and components' information and were not interested in just the most recent changes (delta) or all historical changes (full).

Files that are important for defining the SNOMED CT graph were those that contain information on the concept, concepts' descriptions and concepts' relationships. All these files are located in the Terminology folder of the snapshot release type.

As mentioned, the files are tab delimited and contain column names in the first row of each file. We read these files using standard file reader functions from the .NET's System.IO namespace and saved the results in SQL Server as a set of relational tables. The concepts of the 3 mentioned files are provided in the tables below, followed with the table diagram as in the SQL database. The table structures are all from the (SNOMED International, 2020b).

The concept file structure is as below. The column id in this file contains concepts' unique identifies that are used to name nodes in the graphs.

Field	Data type	Purpose
id	SCTID	Uniquely identifies the concept.
effectiveTime	Time	Specifies the inclusive date at which the component version's state became the then current valid state of the component.
active	Boolean	Specifies whether the concept was active or inactive from the nominal release date specified by the effectiveTime.
moduleId	SCTID	Identifies the concept version's module.
definitionStatusId	SCTID	Specifies if the concept version is primitive or defined.

Table 6 - Concept RF2 file - Detailed Specification

The table below contains information as in the descriptions file. Each concept id from the table above can be associated with one or more descriptions in this table using conceptId field.

Field	Data type	Purpose
id	SCTID	Uniquely identifies the description.
effectiveTime	Time	Specifies the inclusive date at which the component version's state became the then current valid state of the component
active	Boolean	Specifies whether the state of the description was active or inactive from the nominal release date specified by the effectiveTime.
moduleId	SCTID	Identifies the description version's module.
conceptId	SCTID	Identifies the concept to which this description applies.
languageCode	String	Specifies the language of the description text using the two character ISO-639-1 code.
typeId	SCTID	Identifies whether the description is fully specified name a synonym or other description type.
term	String	The description version's text value, represented in UTF-8 encoding.
caseSignificanceId	SCTID	Identifies the concept enumeration value that represents the case significance of this description version.

Table 7 – Description RF2 file - Detailed Specification

The file that contains relationship information is as in the table below. The fields sourceId and destinationId are references to the concepts and typeId depicts the type of the relationship. The relationship typeId that we use for edges of the SNOMED CT graph is SCTID: 116680003.

Field	Data type	Purpose
id	SCTID	Uniquely identifies the relationship.
effectiveTime	Time	Specifies the inclusive date at which the component version's state became the then current valid state of the component.
active	Boolean	Specifies whether the state of the relationship was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SCTID	Identifies the relationship version's module.
sourceId	SCTID	Identifies the source concept of the relationship version.
destinationId	SCTID	Identifies the concept that is the destination of the relationship version.
relationshipGroup	Integer	Groups together relationship versions that are part of a logically associated relationshipGroup.
typeId	SCTID	Identifies the concept that represent the defining attribute (or relationship type) represented by this relationship version.
characteristicTypeId	SCTID	A concept enumeration value that identifies the characteristic type of the relationship version (i.e. whether the relationship version is defining, qualifying, etc.)
modifierId	SCTID	A concept enumeration value that identifies the type of Description Logic (DL) restriction (some, all, etc.)

Table 8 - Relationship RF2 file - Detailed specification

When imported into an MS SQL database, the tables appear as per the diagram below. Tables are named with the prefix S_ followed by the type of the construct described in the table. The constraints visible on the diagram impose the following:

- definitionStatusId field contains unique identifiers that are from the id field of the S_Concept table
- typeId field in the S_Description table contains unique identifiers that are from the id field in the S_Concept table

- caseSignificanceId field in the S_Description table contains unique identifiers that are from the id field in the S_Concept table
- conceptId field in the S_Description table contains unique identifiers that are from the id field in the S_Concept table
- sourceId field in the S_Relationship table contains unique identifiers that are from the id field in the S_Concept table
- destinationId field in the S_Relationship table contains unique identifiers that are from the id field in the S_Concept table
- typeId field in the S_Relationship table contains unique identifiers that are from the id field in the S_Concept table
- characteristicTypeId field in the S_Relationship table contains unique identifiers that are from the id field in the S_Concept table
- modifierId field in the S_Relationship table contains unique identifiers that are from the id field in the S_Concept table

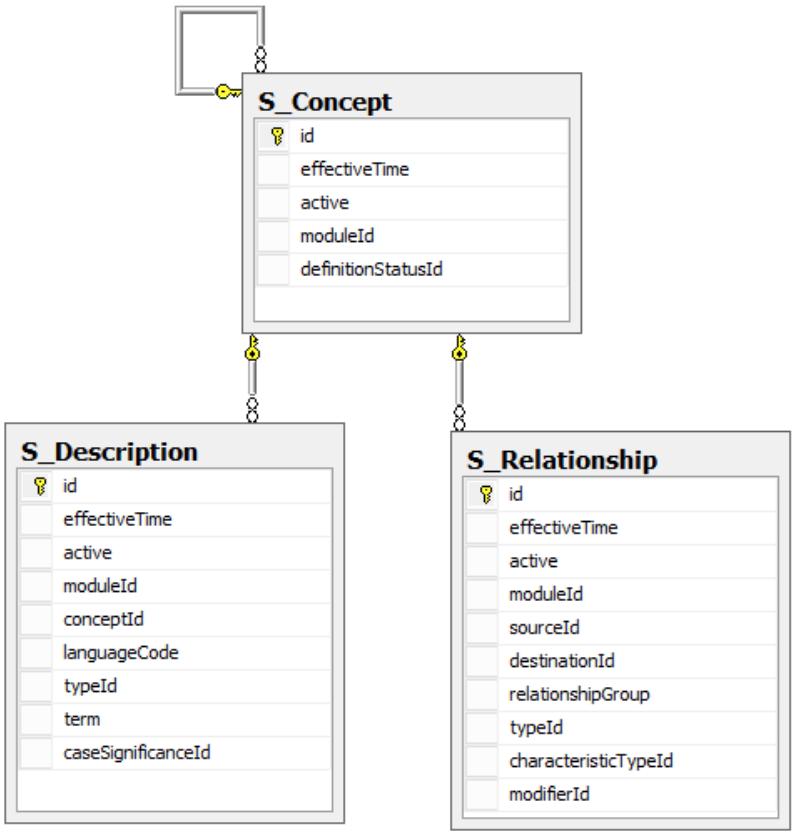


Figure 12 - SQL diagram of the imported SNOMED CT table structure

The SNOMED CT graph is created based on the information imported into the three tables shown in the diagram above. The SNOMED CT graph is created as an in-memory graph. It is

re-created every time the application starts and is preserved between its uses in one application instance.

QuickGraph framework (de Halleux, 2014/2020) is used in creation of an in-memory graph structure. The framework allows creation of the commonly used graph variations and also allows for querying of these structures. The framework allows for the nodes and edges to be defined as custom classes, as opposed to just strings or integers. This was helpful as it allowed us to include information as multiple definition strings and weights for the nodes and edges among others.

However, the QuickGraph framework has no visualisation functions. To be able to visualise segments of the graph, and we very often did that for visual testing purposes, we exported graph segments in XML format and visualised them using Cytoscape (Shannon et al., 2003). Although the QuickGraph framework has an XML export function, we did not find its output suitable for importing to Cytoscape and we created our own export model that we could use not just for visualisation of segments of the SNOMED CT graph but also for visualisation in the process of evaluation of the artefact created in this project. More on Cytoscape and on the XML export function will be included later in this chapter.

Although SCTID: 116680003, Is_A attribute is directed in nature ([specific concept] Is_A [more general concept]), for the purpose of finding the shortest path we render SCTID: 116680003 as undirected.

6.1.2 Finding shortest path in the SNOMED CT graph

We use SNOMED CT as ontology of reality that we represent as a unidirectional graph. SNOMED CT has well defined relationships between the concepts and we use its concepts as nodes and SNOMED CT's Is_A attributes between the concepts as edges connecting its nodes.

For deciding on semantic similarity of two concepts, we measure distance of the concepts in the SNOMED CT graph. We provided evidence justifying use of this method earlier in the section 3.2 Utility of SNOMED CT as ontology of reality.

The distance between two concepts in the SNOMED CT graph is the shortest distance between these two concepts as graph nodes using Is_A connections as edges. We do not consider

SCTID: 138875005, SNOMED CT Concept as a connecting node, hence the shortest paths are not allowed to use the edges of the top-level axial concepts connecting to the SCTID: 138875005.

One Is_A edge is considered 1 hop in the distance calculations and the edges in the graph, for the purpose of finding shortest paths, are considered non-directional. The edges are considered non-directional because we wanted to make sure that the paths between nodes at all levels are found. As an example, if edges were to be one-directional, as Is_A attribute directs, from the peripheral levels of the hierarchy towards its top, there would be no path between the node representing concept SCTID: 49817004, Neonatal diabetes mellitus and node representing SCTID: 73211009, Diabetes mellitus.

That means if, in text A, a 2-gram candidate "neonatal diabetes" was matched to the SCTID: 49817004, and in the text B, a 2-gram candidate "diabetes mellitus" was matched to the SCTID: 73211009, the two concepts would be seen as not connected, looking for text B to text A. The same would be the case if direction of edges is aimed towards the peripheral levels of the hierarchy, hence we decided that the edges in our ontology of reality graph are instantiated as non-directional.

We tested two algorithms for calculating shortest distance, or shortest path as it is referred to in the graph theory (Deo & Pang, 1984). The algorithms are Dijkstra's Shortest Path Algorithm (E. W. Dijkstra, 1959) and Floyd-Warshall Shortest Path Algorithm (Floyd, 1962). These 2 algorithms deploy different strategies in finding the shortest paths, hence our choice.

6.1.2.1 Dijkstra's shortest path algorithm

We present pseudocode for the Dijkstra's algorithm below. The pseudocode is based on the pseudocodes shown in Dijkstra (1959), Dijkstra (1959, p. 270) and Rabin (2019, p. 295).

1) With graph G that has total number of nodes $|V|$ and total number of edges $|E|$, select the start position of the path (source node)
2) Create empty set S. This is where the nodes that form the path will be stored
3) Store start node in the S and assign priority value W of 0 to it
4) Start node becomes the first connecting node CN
5) Work with each neighbouring node NN connected to the CN, one by one

6) Calculate W of the NN. The W of the NN is a sum of the weight of the edge between the NN and CN and the W of the CN
7) If the NN is not in S, or if W is less than the weight of the NN that is in S, place NN in S (or replace if already there) and label it with its name and W. Else, discard NN.
8) Repeat step 5 until there are nodes that are not in S

At the end of the process, the set S will contain all nodes with paths to the start node and their shortest distances to it.

Although this algorithm produces the desired result (shortest path between two nodes), the calculations that we performed to decide on nodes similarity in the process of comparison were many and the process took a long time to complete. The main issue was that Dijkstra's algorithm is practically re-creating a segment of a graph on each search. The segment of a graph re-created is a segment formed of all edges and nodes that form all paths between the source and the destination node. When concepts are proximal and edges sparse, creation of this segment is not a long process, but the process becomes particularly long when the source and the destination nodes are distant and with many connections between them. Moreover, to recreate the segment, the algorithm needs to visit edges that are for paths that are just potential candidates too, until the number of hops is lower than the number of hops in an already-discovered path.

To put this in perspective, we observe the time complexity of Dijkstra's algorithm's runtime. To understand the total time requirement for this algorithm, we need to understand time requirements of the individual algorithm's processes. We observe a simplest case that uses a standard binary min-heap priority queue. We take the values for operations from Cormen et al. (2009).

1) The algorithm keeps nodes and their priority values in the priority value queue. It updates nodes' values by removing and adding the updating node and its priority value back to the queue. The time costs of that operation is $O(\log |V|)$. For each node, this operation happens once per its connecting edge, totalling to $O(|E| \log |V|)$ for a completed Dijkstra's algorithm cycle
2) The algorithm can also keep nodes' priority values in a separate hash table or a matrix. However, this does not change operating time of the whole process

3) During the Dijkstra cycle, each of the nodes are removed from the priority queue exactly once and each removal takes $O(\log |V|)$ time. That adds up to $O(|V| \log |V|)$ for all nodes removed from the queue
4) The algorithm checks whether the priority queue is empty exactly $O(|V|)$ times, once for every node, just before that node is removed from the queue
5) Time for iteration through all node's neighbours is equal to $O(|E|)$ as iteration includes examination of every neighbouring node

Adding the times together, we get time complexity of $O(|E| \log |V|)$, $O(|V| \log |V|)$, $O(|E|)$ and $O(|V|)$. Values $O(|E|)$ and $O(|V|)$ are dominated by the other two values, hence the final time complexity is

$$O((|V| + |E|) \log |V|)$$

Where:

- $|V|$ is the number of nodes
- $|E|$ is the number of edges

For the subgraph rooted in the concept SCTID: 123037004, Body structure, the relevant values are $|E| = 196004$ and $|V| = 617$. That means that the number of calculations that potentially would take place for each short path search iteration is just over 548,000, depending on the positions of the nodes. Although some optimisations would be possible, like implementation of the Fibonacci queue (Barbehenn, 1998), considering the number of shortest path operations we needed to complete the task, we decided against using Dijkstra's algorithm in finding shortest path between the nodes in the SNOMED CT graph in the real time.

We also tested performance of this algorithm in off-line calculation of distances for the purpose of caching and re-using at the time of application execution. However, the tests have shown that the process was too long and that caching the number of shortest path results was not feasible.

6.1.2.2 Floyd-Warshall (FWI) algorithm

The next algorithm tested was Floyd-Warshall (FWI) algorithm. This algorithm creates a matrix of $|V| * |V|$ where $|V|$ is the number of nodes in a graph. The entries in the matrix

created by the FWI are weights of the shortest paths of the matrix elements. FWI creates the entire matrix in one pass of the algorithm, hence execution time of that first part of the process (matrix creation) can be very long for large graphs. However, subsequent operations (shortest path queries) just involve finding the value in the matrix, the process that is close to instantaneous on current systems.

The pseudo-code for this algorithm, as presented in the (Floyd, 1962) is as follows:

1) begin
2) integer i, j, k; real inf, s; inf:= $_{10}10$;
3) for i:= 1 step 1 until n do
4) for j:= 1 step 1 until n do
5) if m [j, i] < inf then
6) for k:= 1 step 1 until n do
7) if m [I, k] < inf then
8) begin s:= m [j, i] + m [i, k];
9) if s < m [j, k] then m [j, k]:= s
10) end
11) end shortest path

Where:

- n is the number of nodes - $|V|$
- m[i, j] is the shortest path between i and j
- $_{10}10$ means that no path is available between 2 nodes

Our tests have shown that the matrix of the SNOMED CT graph size (n ≈ 360,000 nodes) is impossible to create in reasonable time on the systems that we had access to in this project. However, as computational and storage capacity of the systems is constantly increasing, future research should consider revisiting utilisation of Floyd-Warshall algorithm in calculation of shortest path between SNOMED CT nodes.

6.1.2.3 Size issues affecting finding shortest path in the SNOMED CT graph

As mentioned in the text above, the SNOMED CT graph comes with a root node, SCTID: 138875005, SNOMED CT Concept and has 19 axial root nodes, as shown in Table 9. SCTID: 138875005 is the universal connector in the SNOMED CT graph, which means that any two nodes in the graph are connected through SCTID: 138875005, even if they are not in the same axis. That means the number of paths in SNOMED CT graph can be calculated as follows:

$$N = |V| * (|V| - 1)$$

Where $|V|$ is the number of concepts in the graph. Considering that the number of concepts is 356185 the number of paths is 126,867,398,040. As we are not interested in paths with zero distance (path from a node to itself), we correct for that as $(|V| - 1)$.

As our tests show that the average size of each of the paths is around 200 bytes, caching 126,867,398,040 paths would require just over 25 terabytes of data. Removing SCTID: 138875005 from the graph divides SNOMED CT graph into 19 smaller and disconnected graphs, reducing the number of total paths between the nodes. We calculated the total number of shortest paths as follows:

$$\mathrm{N} = \sum_{k=0}^{19} |V|_k * (|V|_k - 1)$$

Where $|V|_k$ is the number of nodes in the axial graph k.

Our calculations show 17,399,645,082 paths in this case. Considering the size of each of the paths, storage requirements change to 3,480 gigabytes or 3.48 terabytes for all of the 19 graphs. Although the paths for the some of the 19 graphs would be of manageable size, like one rooted in SCTID: 78621006, Physical force (just below 1GB), paths for some of the graphs would be far from that, like the graphs rooted in SCTID: 362981000, Qualifier value (1,555 GB) and SCID: 123037004, Body structure (714GB). That excluded caching of all full paths as an option as the size required for caching and computing power required for iterating though the records storing paths would exceed the characteristics of the technology available to us. We show number of concepts under each of the 19 axial roots in the Table 9.

k	Concept (axial root)	Description	Count
1	373873005	Drug	6948
2	404684003	Clinical finding	39609
3	48176007	Social context	6702
4	78621006	Physical force	976
5	123037004	Body structure	59768
6	308916002	Environment or geographical location	1840
7	370115009	Special concept	652
8	71388002	Procedure	21291
9	105590001	Substance	43404
10	123038009	Specimen	4181
11	254291000	Staging and scales	1601
12	272379006	Event	3794
13	362981000	Qualifier value	88183
14	410607006	Organism	35811
15	900000000000441003	SNOMED CT Model Component	1728
16	243796009	Context-dependent categories	1352
17	260787004	Artefact	16806
18	363787002	Observable entity	21051
19	419891008	Record artefact	488

Table 9 - SNOMED axial root concepts, their descriptions and counts

6.1.2.4 Novel technique for finding shortest path pseudo code

As neither Dijkstra's nor the Floyd-Warshall algorithm were efficient in calculating shortest paths in our environment, we explored options for enabling faster traversing, so we could find shortest paths in real-time. Graph partitioning into sub-graphs even smaller than the axial graphs was the strategy that we developed as a result of this inquiry. The process includes creation of smaller subgraphs connected by the normally internal edges that now become external connections between the newly connected sub-graphs.

The process that we followed is as on the following diagram:

Figure 13 - Finding shortest path process phases

Artefacts used to create SNOMED CT graph and tools and frameworks that we used in the process of graph creation are described in the section 6.1.1 Creating SNOMED CT graph and will not be covered here.

It is important to remind here that SNOMED CT graph is firstly created as a directed acyclic graph (DAG), where the Is_A attribute's direction is used as the direction of all edges. The graph is originally one connected structure with SCTID: 138875005, SNOMED CT Concept acting as a root. However, such a graph, when later converted to an undirected graph (UG), has all nodes connected which is an unwanted feature in our process, due to the paths between the nodes that are not in the same axes. The nodes located below different axial roots usually depict either very low or no semantic similarity between the concepts, hence our decision to eliminate these paths. The example are the nodes from the axes rooted in SCTID: 404684003, Clinical Finding and 254291000, Staging and scales. The concept SCTID: 297288000, Liver calculus has not much in common with the concept SCTID: 254364004, National Wilms' tumor study staging system, but the path between them would contain only 5 edges, which would indicate semantic similarity.

To ensure no paths are found between the nodes in different axial graphs, we eliminate the edges between the SCTID: 138875005, SNOMED CT Concept and 19 axial roots from the graph. The result of that process are 19 subgraphs rooted into 19 axial roots, that act as sink concepts. These graphs contain number of nodes as in the Table 9. Terms "*sink*" and "*source*" are the terms used in graph theory (Gross et al., 2013). Sink represents a node in a directed graph that has no outgoing edges. Source nodes are nodes in a directed graph that have no incoming edges. Our version of SNOMED CT graph has 1 sink and 218,012 sources originally. After SCTID: 138875005, SNOMED CT Concept is removed, the graph has 19 sinks and the same number of sources.

However, due to the number of edges, axial subgraphs are still too large for real-time finding of shortest paths between nodes. Hence we partition the subgraphs even further.

Our experiments show that the size of a graph that can be easily managed in our system is 20-250 edges. Graphs larger than 250 are slightly slower to traverse and graphs below 50 edges in most of the cases need to be merged with other graphs for the purpose of reduction of overhead in keeping track of all relevant connections. However, the numbers are not exact and are used for guidance only.

We start further partitioning by finding all source nodes in each of the graphs. Source nodes are the nodes that have no incoming edges. In the SNOMED CT hierarchy, these nodes are at the very bottom of the hierarchy, opposite to the SCTID: 138875005, SNOMED CT Concept. As mentioned earlier, the number of these nodes in the used version of SNOMED CT is 218,012.

When all source nodes are found we start traversing the graph up towards the sink nodes, capturing edges between all parents and their immediate children into a new graph, removing all captured nodes and their edges from the original graph. We also keep record of the edges that we break in the process. These edges are the edges connecting removed nodes to the nodes that remain in the original graph. We repeat this for every source node.

Every time we traverse the graph, the original graph becomes smaller, due to the decreased number of nodes and edges, and the number of new graphs (subgraphs) increases. The final result of this part of the process is no nodes nor edges in the original graph and 218,012 subgraphs.

However, some of the subgraphs in the group of 218,012 subgraphs are very small, some with only a single node in them. Likewise, some of the subgraphs have significantly more than 250 edges. In the next step, we find both, very small (less then 20 edges) and very large subgraphs (more than 300 edges). We do that by ordering all subgraphs by the number of edges from smaller to the largest. For smallest subgraphs, we try to attach as many of them to the subgraphs in the middle of the size-ordered graph array, making sure that the resulting subgraphs are not much larger than 250 edges. We make sure that we merge as many small subgraphs as possible in this process. For large subgraphs, we follow the same process used to break axial graphs, we break them by finding the source nodes and traversing the subgraph finding nodes' parents and their children and removing them from the original structure.

Summarised, pseudocode for this operation would look as follows:

1. Remove SCTID: 138875005, SNOMED CT Concept
2. Find all source nodes (nodes that have no incoming edges)
3. Traverse each subgraph up, towards the sink nodes
4. Create new graphs by capturing edges between all parents and their immediate children into a new graph,
5. Remove all captured nodes and their edges from the original graph
6. Go to 3 for each source node remaining in the original graph
7. Find small graphs and merge them with medium size graphs
8. Find large graphs and repeat steps from 2-8 for each one of them
9. Finish when the number of graphs and their size are acceptable 9processable by the available equipment)

The output of this process is just over 3,000 subgraphs of various sizes, with each subgraph containing information on its content (nodes and edges) and its connections with its neighbouring graphs. The connections with the neighbouring graphs are the edges of the subgraph's nodes that connect to the nodes of the neighbouring graphs. Each subgraph is given unique ID. We also have information on the location of each of the nodes, stored as pairs of concept SCTID and subgraph ID. From now on, all edges are considered undirected, which technically makes all subgraphs undirected graphs (UG).

We use graph IDs and information on their connections to create a "graph of graphs" (GoG). GoG is a structure where subgraphs are represented as GoG nodes and edges that were recorded as connections between the subgraphs are GoG edges. As we also recorded information on the locations of nodes (which subgraph a node is in), finding the shortest path between the nodes N1 and N2 follows the following pseudo code:

1) If the N1 and N2 shortest path result is in cache, return result and end
2) Find the name of the subgraph G1 node N1 is in
3) Find the name of the subgraph G2 node N2 is in
4) If G1 = G2, use Dijkstra's algorithm to find shortest path between nodes N1 and N2 in G1 and end
5) Use Dijkstra's algorithm to find the shortest path S between G1 and G2 in GoG.

6) Record node NC1 that is at the G1 end of the edge (connection) between G1 and its first neighbouring subgraph GN1 in S
7) Record node NC2 that is at the G2 end of the edge (connection) between G2 and its first neighbouring subgraph GN2 in S
8) If there is no connection between N1 and NC1 in G1, temporarily break the edge between NC1 and its connecting node in the GN1 and GOTO 4
9) If there is no connection between N2and NC2 in G2, temporarily break the edge between NC2 and its connecting node in the GN2 and GOTO 4
10) Temporarily merge graphs that form S in GM
11) Use Dijkstra's algorithm to find the shortest path between N1 and N2 in GM
12) Cache the result in memory

We used the above solution with success in finding shortest paths in real-time.

6.1.3 Concept expansion

Concept expansion is a process of creation of complex information models based on concepts discovered in the process of concept matching. We call concepts discovered in the process of concept matching the central concepts. They are a product of matching of 1-gram, 2-gram or 3-gram candidates to the concepts' descriptions in SNOMED CT.

Complex information models created in the concept expansion process are graph structures that consist of a central concept and expanding concepts. Expanding concepts are concepts proximal to a particular central concept, connected to it with an Is_A attribute in the SNOMED CT. The concepts proximal to the central concept that we use in the expansion process are first level ascendants (parents), first level descendants (children) and siblings. All concepts, central and expanding, become nodes in the graph representing newly created information model. The Is_A attribute connections between the nodes become graph edges.

This process of expansion involves querying the SNOMED CT graph for each type of proximal concepts, ascendants, descendants and siblings. Also, multiple queries to the SNOMED CT persisted in the SQL database are made if more than just a first level of ascendants and descendants is to be used. For example, if only one level of ascendants is used in expansion, one call is made to the SNOMED CT and nodes in a position of parents are requested only. However, if 2 or 3 levels of ascendants are used in the process of expansion, 2 or 3 calls are

made, one to get parents and one to get parents of each of the parents received in the first 2 calls to the SNOMED CT. We originally selected to expand 3 levels, before we reduced expansion to one level only, due to limitations that we experienced.

The limitations that we faced were related to the computation time required to expand the concept and the size of the expanded structure.

Computation time required for expansion of one concept depends on the number of nodes that the concept is directly connected to in the SNOMED CT graph. The greater the number of nodes attached to one concept, the longer the expansion time as more concepts need to be found, transported to the computation agent and assembled in the information model, a graph. Also, the greater the levels of expansion, the greater the number of concepts found, and as a result, the longer the expansion process.

Computation time also depends on the number of concepts found in the process of matching. The grater the number of concepts, the greater the number of expansions. Longer texts will have more concepts matched, which implies that the longer the text, the greater the number of expansions and the longer the computation time required to expand the concepts found in the text.

To overcome this problem, we resorted to offline caching of the information models created for each expanded concept found and matched in the text. That reduced the execution time as information models did not need to be generated in real time, but were downloaded from their place of persistence, SQL database in our case.

However, although offline caching has helped to reduce execution time of the main algorithm, the reduction of time was not significant and sufficient. The reason for that was the large size of the created information models caused by the number of concepts that the created information models consisted of. As information models are persisted in their serialised form, usually as JSON text, they needed to be de-serialised into their objects before they could be used in calculations. De-serialisation of a large JSON object is a complex undertaking and needs significant computational power and time. Hence, we decided to reduce the size of the created information models.

To reduce the size of the created information models, we minimised the number of levels of expansion of ancestor and descendant expanding concepts. We experimented with 1, 2 and 3

levels and decided on using only 1 level, including only parents, children and direct siblings of the central concept. For the same reason, we also excluded all other but Is_A relations of the central concepts.

Future work will suggest that experiments are conducted testing value of including more than one Is_A level as well as value of including concepts connected to the central concept by non-Is_A attributes. We leave provision in our application for that, allowing for definition of weights for other than first level of ascendants and descendants as well as for definition of weight for concepts connected to the central concept by each of the non-Is_A relationships.

As the processes of experimenting with the level of expansion and the process of caching expanded information models of the central concepts has taken place outside of the process of running the main algorithm, we include it here as the prerequisite.

6.2 Evaluation

We evaluate validity of our annotation method by testing similarity of the documents contained in the corpus containing 889 discharge summaries that we described in the section 3.3 Testing SNOMED CT coverage. The method that we designed and used in this process is explained next.

6.2.1 Methods used in documents comparison

An output of a process of document comparison is a number showing a level of semantic similarity of two documents. The method that we deployed is based on our **Assumption 2** that predicates that proximity of concepts in the ontology of reality is a function of their semantic similarity and that the more proximal the concepts are in the ontology of reality, the more similar they are.

For two documents to be compared, they need to be annotated. A document is annotated when a weighted graph (annotation artefact) constructed of SNOMED CT concepts as nodes and is_A relations as edges is associated with text in that document. The annotation artefact, a weighted graph, is a result of a complex process where information is extracted, transformed and formatted into a graph structure. The process includes extraction of concept candidates

from a body of free text, normalisation of concepts, expansion of concepts into graph structures and merging of created graph structures into a final graph structure before a process of comparison takes place. This process has been explained in the section 4 Design and develop artefact (DSR activity 3) in this document.

After annotation artefacts are created, their nodes are extracted in two separate lists where the artefacts are ordered by their weights in descending order. Next, the top x nodes are selected for comparison, an equal number from each of the two lists. If any of the two lists has less than x nodes ($y < x$), y becomes the number of nodes selected from each of the lists. In the case of our corpus, $x = 5$ for all documents.

At this stage we have two weighted lists populated with an equal number of nodes as list items, each list representing one document. The nodes are selected based on their weight, assuming that the higher the weight of a node is, the more semantically representative the node is of a document's content, or part of it.

Each of the selected nodes is assigned a rank value R, based on its position in the list. The first in the list, the node with the greatest weight, is given rank 1 and the rank value increases as the node's position decrease in the list. For example, if a list has 20 nodes, the node with the highest weight value, located at the start of the list will be assigned $R = 1$, and the node at the end of the list will be assigned $R = 20$.

The process of comparison of two documents involves measuring of distances between each of the nodes in the two sets, for the purpose of establishing semantic similarity of the concepts represented by the nodes. We explained that process in the section 6.1.2 Finding shortest path in the SNOMED CT graph. The outcome is information on how many connections are between the concepts in the two documents and how strong these connections are.

As the rank of the node, or R value, depicts the relevance of the concept represented by the node, we corrected the distances between the nodes with their ranks. The final result of the calculation is a Similarity Coefficient or Sco as named by us. The calculation looks like the following:

$$Sco = \sum_{k=1}^{x} \sum_{n=1}^{x} \frac{P_{max} + 1 - d(N_k, N_n)}{\left(R_{N_k} + R_{N_n}\right)}$$

Where:

- x is the number of nodes in the list
- $d(N_k, N_n)$ is the length of the shortest path between the nodes N_k and N_n
- P_{max} is the maximum distance between any two nodes in the lists that is not infinity
- R_{N_k} is the rank R of the N_k node
- R_{N_n} is the rank R of the N_n node

P_{max} corrects for the fact that shortest path distance is a positive number increasing as the similarity, that we measure with it, decreases. The greatest distance, P_{max} depicts the smallest similarity of two concepts before infinity (not similar at all). In the same time, 0 (zero) depicts full similarity (identical concepts). Sco is calculated only when $d(N_k, N_n)$ is smaller than infinity (when the shortest path exists). Based on results of our experiments confirming that no distances have value > 20, we selected 20 as the P_{max}.

Due to computational limitations, only top 5 nodes are taken in consideration in the process of document comparison, hence the equation looks like the following:

$$Sco = \sum_{k=1}^{5} \sum_{n=1}^{5} \frac{20 + 1 - d(N_k, N_n)}{(R_{N_k} + R_{N_n})}$$

We believe that concept of similarity is represented as a range, rather than an exact set of values. That means that for us, similarity is a range between the minimum and the maximum similarities, rather than a set of values that each depict a known similarity level. Hence, it bears no importance whether the maximum similarity is represented as a number lower than minimum similarity, or the other way around, as long as these limits are known.

Thus, we assumed that if the range $0 \rightarrow P_{max}$ depicts decreasing similarity, the same range can be presented as $-P_{max} \rightarrow 0$, with $-P_{max}$ as the minimum similarity and 0 as maximum similarity. If both ends of the range are expanded with P_{max} , the range becomes $0 \rightarrow P_{max}$ with 0 representing minimum similarity and P_{max} representing maximum similarity. We add 1 to P_{max} to avoid divisions by zero in the cases when $P_{max} == d(N_k, N_n)$, which happens in the cases of the nodes with the shortest paths in the graph spanning 20 edges.

However, there is no tested scale that can be used for validation of a number calculated using the technique presented above (Sco). Hence, we decided to test this technique deploying it on

the set of documents that we know are similar and on the set of documents that we know are not similar and compare the outputs. The logic is that outputs of testing similarity of similar documents will result in Sco values different (higher) than outputs of testing of similarity of documents that are not similar.

We split our discharge summaries to create a sample of similar and dissimilar documents. Each discharge summary was split into:

SD_k - diagnosis section and

SR_k - the rest of the document

Where:

- k is an ID of a discharge summary document, $k \in \{ID_1, ID_2 \dots ID_n\}$ and
- n is a number of discharge summaries that have diagnosis sections

Some of the discharge summaries did not have diagnosis sections and they were eliminated from the experiment. The number of discharge summaries that had entries in their diagnosis sections is 582, hence 307 documents were eliminated from the corpus.

We expected that diagnosis sections will contain concepts that are similar to, but most likely not exactly the same as, concepts in the remaining of the document. For example, if a diagnosis section contained concept SCTID: 73211009, Diabetes mellitus, it is very likely that the rest of the discharge summary would contain concepts like SCTID: 237598005, Hyperglycemic disorder or SCTID: 237622006, Poor glycemic control, all concepts proximal as they are positioned in the SNOMED CT graph. Therefore, our sample for calculating Sco of similar documents is a set of SD_k and SR_k pairs where k is an ID of a discharge summary. We call SD_k and SR_k documents similar documents pair.

Dissimilar documents, on the other hand, are expected to have semantically dissimilar concepts listed in them. We ensure that we compare dissimilar documents by comparing diagnosis sections of one discharge summary with a remaining of a document from a different discharge summary. In other words, we compare SD_k and SR_{k+j} where $(k + j) \in \{ID_1, ID_2 \dots ID_n\}$ and $j > 0$ and $j \in \{10, 20, \dots 490\}$.

We believe that Sco value of similar documents will be higher than Sco value of dissimilar documents. Later in this chapter, we outline statistical methods that we use to confirm that assumption.

In addition to Sco value for each of document pairs calculated in each of the iterations, we also calculate Siv value, that is a total sum of all distances for each of the discharge summaries in each of the iterations, unadjusted for ranking of concepts included in distance calculation. We measure that because we expect that the total sum of all shortest paths between nodes in the sets derived from annotation graphs will be greater when similar documents are compared than when dissimilar documents are compared. The formula that we used for calculating this type of similarity of two documents is as below. We call this measure Indicative Similarity Value or Siv. As with Sco, Siv is calculated only when $d(N_k, N_n)$ is smaller than infinity (when a shortest path exists).

$$Siv = \sum_{k=1}^{x} \sum_{n=1}^{x} P_{max} + 1 - d(N_k, N_n)$$

Where:

- x is the number of nodes in the list
- $d(N_k, N_n)$ is the length of the shortest path between the nodes N_k and N_n
- P_{max} is the maximum distance between any two nodes in the lists that is not infinity. We used 20 as the value for P_{max}

Considering that we calculate distances between 5 highest ranked concepts only and that P_{max} is 20, the final formula looks like the following:

$$Siv = \sum_{k=1}^{5} \sum_{n=1}^{5} 20 + 1 - d(N_k, N_n)$$

Another construct we measure is the average strength of the connection between the documents, $AvgSco$. We do that by dividing the Similarity Coefficient or Sco with the total number of shortest paths found between documents' nodes compared in the process. The formula is as follows:

$$AvgSco = \frac{Sco}{SP_n}$$

Where:

- SP_n is a total number of shortest paths found between compared nodes

We expect that $AvgSco$ for similar documents will be higher than $AvgSco$ of dissimilar documents.

6.2.2 Evaluation results and discussion

For our evaluation experiment, we conduct 50 iterations of calculating Sco for each of the 582 discharge summaries. 49 out of 50 iterations use dissimilar documents as a sample (Dis1 – Dis49 in Table 16) and 1 uses the standard sample of discharge summaries, a sample of similar documents (Sim1 in Table 16). We then summarise results of all iterations. Evaluation results are presented in Table 16 in the Appendices.

6.2.3 Discussion of the evaluation results

The results have shown the following:

1) Siv of the Sim1 is greater than any other Siv results of the Dis1 – Dis49 iterations
2) Sco of the Sim1 is greater than any other Sco results of the Dis1 – Dis49 iterations
3) $AvgSco$ of the Sim1 is greater than any other $AvgSco$ results of the Dis1-Dis49 iterations

Hence, the data shows that shortest paths unadjusted for the rank of nodes that they belong to (Siv) are shorter, that shortest paths adjusted for their ranks (Sco) are shorter and that the average shortest paths adjusted for the rank of nodes that form these shortest paths ($AvgSco$) are shorter when documents in the similar group (Sim1) are compared to when documents in the dissimilar groups (Dis1-Dis49) are compared. These conclusions are the result of a simple observation of the values in the Table 16.

However, these numbers by themselves are not sufficient evidence to support the claim that an artefact is able to correctly annotate a document for comparison. Hence, we next test whether

results of comparison of similar documents are somewhat different than results of comparison of dissimilar documents. For that, we use a statistical measure that finds outliers in the array of values.

Outliers test is a test used for detection of an unusually extreme value for a variable, given the statistical model in use (Barnett & Lewis, 1984). "An outlying observation, or 'outlier,' is one that appears to deviate markedly from other members of the sample in which it occurs" (Grubbs, 1969, p. 1). The value is either at the top (highest value) or the bottom (lowest value) of the value ordered model.

An outlier might be merely an extreme manifestation of a variability of information contained in a model. A variability might indicate errors in measurement process, deviation from experimental procedure or errors in supporting data if any supporting data is used (Grubbs, 1969). In case of the evaluation experiment that we conducted, we can exclude errors in measurement process, deviation from experimental procedures and errors in supporting data as our iterations were all automated, performed as a continuous process and used the same procedure, measures and supporting data. To increase validity of results, we conduct 50 iterations, each including 582 measurements and we sum results of each of the iterations to minimise the impact of results variability. Based on the number of iterations and measurements, as well as use of sums instead of individual results, we consider outlier a good measure for testing for extreme manifestations of variabilities in the results of our experiment.

The output of an outlier test is a representation of a chance that a value could have been encountered in a set of values. An output below 5% chance strongly indicates an outlier, or in other words, indicates that a value is an extreme manifestation of variability in a particular model.

Although there are several tests used for testing for outliers, tests by Dixon and Grubbs are usually applied (Reichenbächer & Einax, 2011, p. 43). However, Dixon's tests is only for sample sizes of up to 29 results ($n < 29 \,||\, n == 29$), hence we decide to utilise Grubb's test.

Grubb's test assumes normality of results presented as data, except for the outlier. Hence, the results are first tested for normal distribution before the Grubb's test for outliers is even applied. The normal distribution is a probability function that explains how values are distributed. For a distribution to be normal, values are expected to cluster around the central peak and to be

distributed equally on both sides of the central point. A Bell Curve is a specific representation of normally distributed data.

For testing normal distribution of data, we use the Rapid Test for Normal Distribution (David et al., 1954), further referred to as David's test, and the d'Agostino-Pearson (1973) test.

David's test calculates a value of normal distribution coefficient that is than found in the Significance Table for Testing Normal Distribution (Reichenbächer & Einax, 2011, p. 350; Sachs, 2013) shown as Table 17. The coefficient is calculated as follows:

$$\rho_{\acute{R}} = \frac{\gamma_{max} - \gamma_{min}}{s}$$

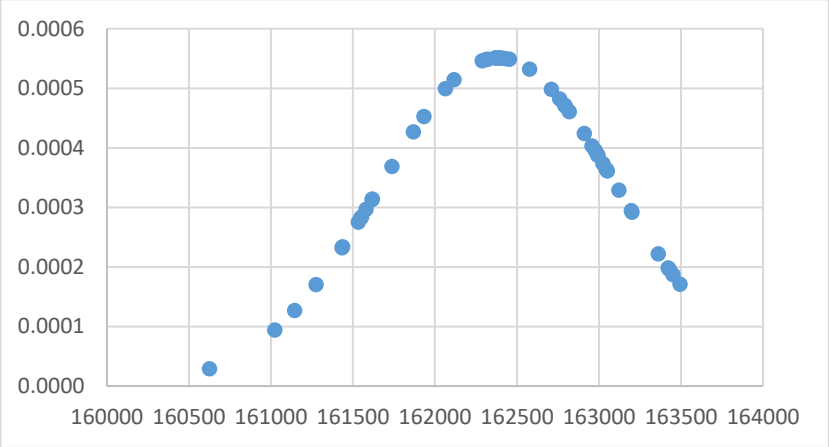
Where:

- γ_{max} is the maximum value in the range
- γ_{min} is the minimum value in the range and
- s is standard deviation of the range

As this test only provides a strong indication that a normal distribution exists, rather than that the data conforms to a normal distribution, it needs to be confirmed with one of the more robust tests. We selected the d'Agostino-Pearson test as confirmatory measure. This test is a robust test, an integration of Skewness and Kurtosis tests (Mardia, 1970). The d'Agostino-Pearson test was performed using the Real Statistics Resource Pack software, an Excel add-in created by Dr Charles Zaiontz (2020).

We start with a null hypothesis (H_0) that states that the Siv, Sco and $AvgSco$ results are normally distributed. To reject this hypothesis, we performed David's test first followed by the d'Agostino-Pearson test. The results of the tests are as follows:

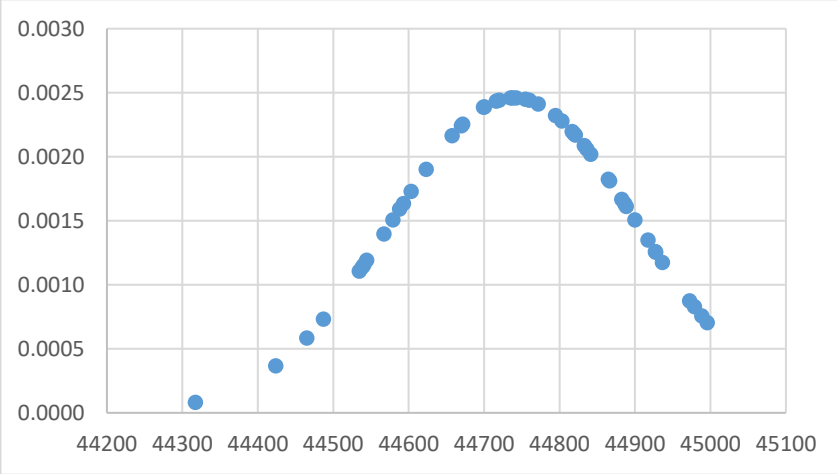
Siv



Measures	
David	3.921268506
d'Agostino-Pearson	
p-value:	0.17700975
α:	0.05
Normal:	**Yes**

Table 10 - Siv normal distribution testing results

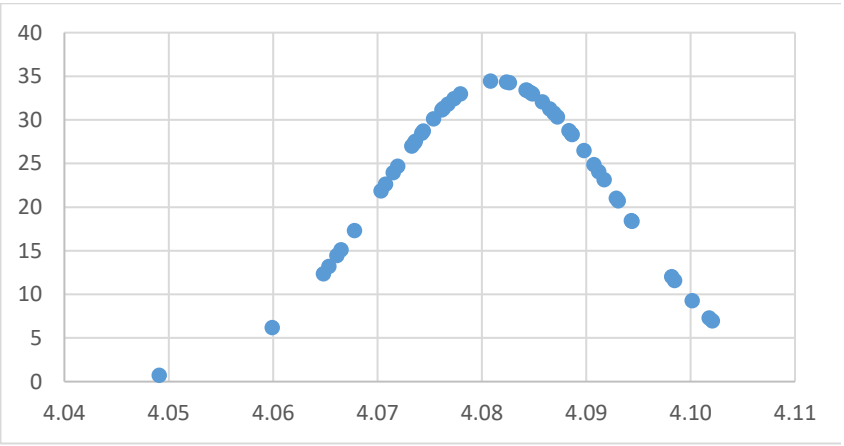
Sco



Measures	
David	4.143223288
d'Agostino-Pearson	
p-value:	0.328627127
α:	0.05
Normal:	**Yes**

Table 11 - Sco normal distribution testing results

AvgSco



Measures	
David	4.531572974
d'Agostino-Pearson	
p-value:	0.651485282
α:	0.05
Normal:	**Yes**

Table 12 - AvgSco normal distribution testing results

The following compares results of the David's test with the values in the significance table for testing normal distribution according to David, listed as Table 17:

- ***Siv***: $\rho_{\hat{R}} = 3.921268506;\ n = 49;$
  - $P = 95\%$: 3.83 < 3.921268506< 5.35 –> **normal distribution suggested**
  - $P = 99\%$: 3.62 < 3.921268506< 5.77 –> **normal distribution suggested**
- ***Sco***: $\rho_{\hat{R}} = 4.143223288;\ n = 49;$
  - $P = 95\%$: 3.83 < 4.143223288< 5.35 –> **normal distribution suggested**
  - $P = 99\%$: 3.62 < 4.143223288< 5.77 –> **normal distribution suggested**
- ***AvgSco*:** $\rho_{\hat{R}} = 4.531572974;\ n = 49;$
  - $P = 95\%$: 3.83 < 4.531572974< 5.35 –> **normal distribution suggested**
  - $P = 99\%$: 3.62 < 4.531572974< 5.77 –> **normal distribution suggested**

The results clearly provide no grounds for rejecting H_0. This allowed us to proceed to the d'Agostino-Pearson's test for normal distribution.

To perform d'Agostino-Pearson's test, we use Excel, Real Statistics Resource Pack add-on and $DPTEST(R1)$ formula where $R1$ is the array of measures for each of the variables, $Siv, Sco\ and\ AvgSco$. The results of the d'Agostino-Pearson's test are as follows

- ***Siv***: $n = 49;\ \alpha = 0.05;\ p - value = 0.17700975$
  - 0.17700975 > α –> **normal distribution confirmed**
- ***Sco***: $n = 49;\ \alpha = 0.05;\ p - value = 0.328627127$
  - 0.328627127 > α –> **normal distribution confirmed**
- ***AvgSco***: $n = 49;\ \alpha = 0.05;\ p - value = 0.651485282$
  - 0.651485282 > α –> **normal distribution confirmed**

As the H_0 has not been rejected, we proceed with the Grubb's test for testing for outliers.

Our H_0 in the case of outliers states that there are no outliers in the range of measured values of neither of the 3 variables: $Siv, Sco\ and\ AvgSco$. In the Grubb's test, we calculate two Grubb's defined values, G and G_{crit} and compare them. If $G > G_{crit}$ than we can reject null hypothesis (H_0). The following are the relevant formulas for G and G_{crit}:

$$G = \frac{x_{max} - \ddot{x}}{s}$$

Where:

- x_{max} is a suspected outlier
- $\ddot{x}$ is mean and
- s is a standard deviation

And

$$G_{crit} = \frac{(n-1) * t_{crit}}{\sqrt{n * (n-2+ t_{crit}^2)}}$$

Where:

- n is the sample size
- t_{crit} is the critical value of the t distribution $T(n-2)$ and the significance level is α/n

We calculated G and G_{crit} using Excel and Real Statistics Resource Pack add-on and $GRUBBS(R1, lab, alpha)$ function where $R1$ is an array of values (measurements), lab is the layout of the output expected (True for 4X4 output, False for single output of an outlier) and $alpha$ (α) is significance level. We used $\alpha = 0.05$ as a significance level. The results of the Grubb's outlier test for $Siv, Sco\ and\ AvgSco$ are as follows:

Siv

Original		**After outlier removed**	
Outlier	166202	Outlier	160625
G	4.140916757	G	2.406745803
G_{crit}	2.956974847	G_{crit}	2.949060371
Significant	**Yes**	**Significant**	**No**

Table 13 - Outlier test result for Siv

Sco

Original		**After outlier removed**	
Outlier	45817.69643	Outlier	44317.31587
G	4.748305914	G	2.578050259
G_{crit}	2.956974847	G_{crit}	2.949060371
Significant	**Yes**	**Significant**	**No**

Table 14 - Outlier test result for Sco

AvgSco

Original		**After outlier removed**	
Outlier	4.187323746	Outlier	4.049092359
G	5.483846621	G	2.760796284
G_{crit}	2.956974847	G_{crit}	2.949060371
Significant	**Yes**	**Significant**	**No**

Table 15 - Outlier test result for AvgSco

The results of the outlier tests reject the H_0, confirming the following:

1) Siv value of the Sim1 is an outlier compared to the Siv results of the Dis1 – Dis49 iterations
2) Sco value of the Sim1 is an outlier compared to the Sco results of the Dis1 – Dis49 iterations
3) $AvgSco$ value of the Sim1 is an outlier compared to the $AvgSco$ results of the Dis1-Dis49 iterations

The test points out the outliers as the Sim1 measurements in all, $Siv, Sco\ and\ AvgSco$ groups of results. Indicative is that after the outlier is removed, the next outlier test shows that there are no significant outliers in the results. That finding indicates high data validity.

However, despite strong data validity, large sample and mature and robust statistical methods utilized in the process, mere presence of an outlier in a pool of results does not provide evidence sufficient for confirming that our annotation method is valid. Nevertheless, the results are promising and certainly provide strong indication of validity and are enough to incite further

work on the subject of annotation using complex information models modelled upon ontologies as reference and a model of reality.

As the goal of this project is to start a process towards defining a method, rather than to fully define and fine tune a method, considering the complexity of the subject, the results of the evaluation strongly suggests that that the goal of this project has been achieved. The artefact created in this DSR guided project and the methods described in this document represent a blueprint that will help the next group of researchers to not start from ‘ground zero’ but to have a starting point that will provide them with a rather valuable guidance.

7 Future work

This method certainly merits further investigation that will confirm its validity (1), investigate benefits of inclusion of functions that have not been implemented due to technical limitations (2) and test its utility on other formats of information (3), e.g. structured information. Therefore, we suggest that the following is suitable continuation of work presented in this document:

- Evaluate the artefact using human agents (1)
- Include other than SCTID: 116680003, Is_A attributes as connections between the concepts (2)
- Include more than one generation of ancestors and descendants in the expansion process (2)
- Evaluate utility of Annotation of clinical datasets using openEHR Archetypes presented as a use-case in this document (3)

Each of these four suggested lines of future work will be expanded upon next.

7.1 Evaluate artefact using human agents

Although the evaluation presented in this document is fairly robust, we suggest that annotations created by the artefact we created are evaluated by human agents and that results are triangulated with the results presented in this document. However, considering that our artefact's outputs are complex information models, that are normally not easy to comprehend by human agents, the method used in this process will be difficult to design.

When evaluation by human agents is designed, the researcher will have to consider the fact that the algorithm we created as the artifact of this work annotates the messages, rather than individual concepts found in the document. Hence, just highlighting the concepts found in the document, which is commonly used as a technique in some evaluations, should not be considered as the concept recognised in the text might be part of the message that has meaning wider than the concept. For example, the sentence: "systolic and diastolic values have equalised" could indicate the message like SCTID: 58283004 | Narrow arterial pulse pressure

or one of the low or high blood pressure variants, which are more valuable for annotation than just concepts like 271649006 | Systolic blood pressure (observable entity) and 271650006 | Diastolic blood pressure (observable entity).

Also, important to consider is that the outputs of our algorithm are SNOMED CT coded entries organised in a graph, rather than just plain phrases. For example, the concept SCTID: 55382008 | Cerebral atherosclerosis (disorder) has 4 synonyms: Cerebral atherosclerosis, Atherosclerosis of intracranial artery, ICAD - intracranial atherosclerotic disease and ICAS - intracranial atherosclerosis. Hence, it is important that human agent evaluators use SNOMED CT codes as annotation elements, rather than just concept text.

Our recommendation of the design of the evaluation experiment will be given next. Please note that we provide guidelines only and not a prescription for an experiment.

Participants: The participants should have clinical knowledge sufficient to recognise clinical messages in the discharge summary documents from the corpus. They also need to be familiar with the SNOMED CT structure and proficient users of one of the SNOMED CT browsers. The number of participants in similar experiments ranges from 3 (e.g. Chapman et al., 2008) to undisclosed (e.g. Viani et al., 2019) and no gold standard exist, hence we are unable to suggest the optimal number.

Sample: The sample should be selected using Simple Random Sample. The researchers should use optimal system for selecting IDs of the documents from the sample of documents used in the evaluation described in this document. Only one set of documents should be selected and distributed to the participants. The sample should be presented in an easy-to-read electronic format.

Data collection: Participants should complete the experiment individually and in isolation from one another. Participants should also have full access to the SNOMED CT browser. They are to be presented with an electronic form where they will be able to list n number of concepts recognised as best descriptors of the messages conveyed in each of the sample documents. The most practical layout of the form would be n number of text boxes at the end of each document. We suggest that $n = 5$ as that was the same number of concepts used in our experiment. However, selection of a number of concepts was influenced by our technical limitations, rather than any other reason. This strategy is comparable to the strategy that we deployed in the process of comparison of documents.

Statistical analysis: After the representative concepts are selected by all of the participants, the concepts are to be compared with the concepts selected by the algorithm for each of the documents from the sample. We suggest that interrater agreement (Fleiss et al., 1981) is calculated between the participants only and between the participants and the algorithm altogether. If the interrater agreement result when the algorithm results are included is better or the same than the interrater agreement result when only the outputs of the participants are taken in consideration, the algorithm output is valid.

The null hypothesis for the experiment would be similar to: "***Interrater agreement calculating agreement between the outputs of the algorithm and human agents altogether is worse than the interrater agreement calculating agreement between the outputs produced by human agents only.***"

We suggest that distance of concepts in the SNOMED CT graph is included in measuring interrater agreement. For example, if one agent has selected 267036007 | Dyspnea (finding) and the other agent has selected 230145002 | Difficulty breathing (finding), the analysis method needs to take in consideration that these methods are just one hop away from one another in the SNOMED CT ontology and that the agreement of these two outputs is higher than if one agent has selected 267036007 | Dyspnea (finding) and the other agent has selected 707540007 | Acute respiratory distress in newborn (disorder), which are much further away from one another.

Triangulation of results of evaluation using human agents with result of the evaluation presented in this document would either substantiate or refute our claim that annotation of free text using complex information models is valid and that the artefact we developed in this work produces valid annotation information models.

7.2 Include other than SCTID: 116680003, Is_A attributes

Inclusion of other than Is_A attributes as connection between the concepts would provide richer and more detailed information about the real connectedness of concepts in a segment of reality they belong to.

For example, the current algorithm and the weighting structure will give very low weight to the concept SCTID: 72704001, Fracture (morphologic abnormality) even if found in the same

text as the concept SCTID: 23406007, Fracture of upper limb (disorder). The reason for that is the absence of an Is_A attribute between these two concepts in the SNOMED CT graph. However, if other than just Is_A attributes are included in the expansion, the algorithm would include the concept SCTID: 363698007, Finding site (attribute) as a connection between the two and therefore adjust the weight of the SCTID: 72704001, Fracture (morphologic abnormality) concept.

The same would be the case with the concepts SCTID: 371195002, Bone structure of upper limb (body structure) and SCTID: 23406007, Fracture of upper limb (disorder). The current algorithm would assign a low weight to the concept SCTID: 371195002 unless other than Is_A attributes are taken in consideration, in which case the concept SCTID: 363698007, Finding site (attribute) would be seen as a connecting attribute. Both examples are shown on the following diagram:

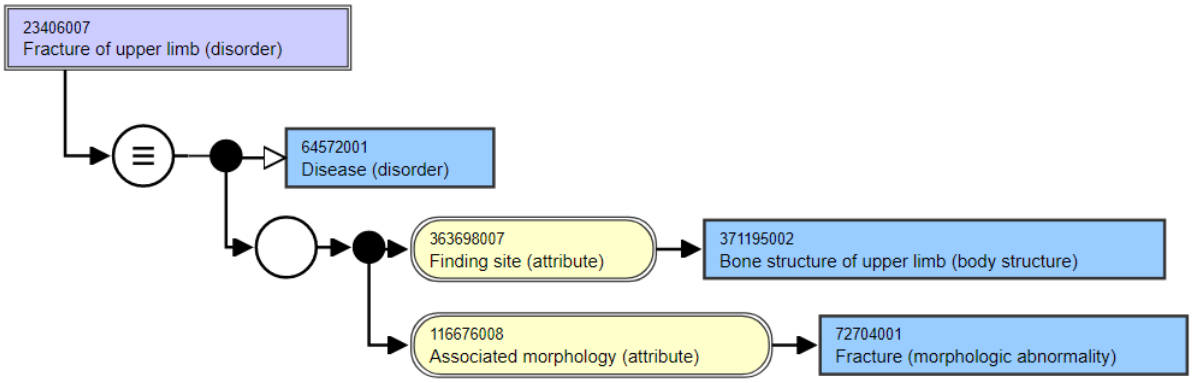


Figure 14 - Example of where inclusion of non-Is_A attributes would improve utility of the developed artefact

Provision for inclusion of non-Is_A connections in a process of expansion has already been made visually on the Weights tab in the Grapher application, and only minor changes would need to be made in the application's mid-tier layer for this to work. However, a system with significantly higher information processing power compared to the one we used in this project would be required, due to increased number of new concepts and new connections that will need to be integrated in a reality segment created by expansion.

7.3 Include more than one generation of ancestors and descendants in expansion process

Inclusion of more than one generation of ancestors or descendants would improve detail of description of a segment of reality represented after expansion. This ability would also allow that as required, only ascendants or only descendants are used as expanding elements. That would further allow for movement of focus towards more specialised concepts (more generations of descendants) or more general concepts (more generations of ascendants).

Provision for that has already been made in the Grapher application as Grapher application allows for weights of generations of ascendants and descendants to be defined. No changes in the mid-tier layer of the application would be needed as the Grapher has already been programmed to process any number of generations of ascendants and descendants as listed in the "Ascendants weights" and "Descendants weights" generation weights control.

However, introduction of new generations of concepts would increase processing requirements and a system with higher processing power compared to the one we used in this work will have to be used.

7.4 Evaluate utility of Annotation of clinical datasets using openEHR Archetypes presented as a use-case in this document

We present a novel use case of openEHR Archetypes in this work. We suggest openEHR Archetypes as annotation artefacts, rather than as just the general clinical information models that they have been designed to be. We suggest that the semantic power of openEHR Archetypes can be utilised for annotation of clinical documents and we suggest that our artefact is used as the intermediary algorithm that ensures that the right openEHR Archetypes are used in the process.

The Grapher application offers the required functionality for the proposed evaluation to be conducted. It processes openEHR Archetypes to the point where their semantic similarity with other documents, discharge summaries in the current work, is expressed quantitatively. Two sections of the openEHR Archetypes are processed, Archetype schema and Archetype "Use" section, and both results are offered separately. The evaluation design will decide whether just

one or both sections' results will be taken into consideration and whether the evaluation will utilise automated or human agents in the process or both.

8 Contribution of this PhD work

In reviewing PhD contribution to the body of knowledge, Gill and Dolan suggest that doctoral candidates should be able to critically elaborate on how and in what way their research makes a meaningful contribution to the body of knowledge (2015, p. 11). Clarke and Lunt' (2014) focus on originality of achievements of doctoral research and, although they confirm that originality is difficult to define, one of their findings is that examiners see publishability as an evidence of originality of achievements of doctoral projects. We take this as a guideline in defending the originality aspect of our project's achievements.

8.1 Originality

During the course of this work, we reported our findings in one international journal (Zivaljevic et al., 2020), on three conferences (Zivaljevic et al., 2015b, 2015c, 2019) and on two forums (Zivaljevic et al., 2015a, 2016). The journal article and the article presented on the HINZ2015 conference were peer reviewed. The work presented in the journal was one of the prerequisites for this project and the findings are presented in the thesis. The work presented on the HINZ2015 conference was awarded best scientific work award. The conference was a three-day conference where presenters from New Zealand and Australia presented their scientific and applied work. We believe that our publication achievements, peer reviewed publications in particular, are strong evidence supporting originality of our work presented in this thesis.

Phillips and Pugh (2010) list that being cross-disciplinary is a good indication of originality of a PhD research. We believe that we present original, cross-disciplinary thinking by producing an artefact that operates in the domain of technology on a basis of our assumptions grounded in the field of philosophy. We also use SNOMED CT in a new way, as an ontology of reality. This artefact has certainly not been made to serve that purpose, but the methods that we create and detail in our work, like conversion from SNOMED CT RF2 form into a graph and finding a shortest path in a large graph, make it suitable for that purpose. The algorithm that we produce to partition the SNOMED CT graph and to find a shortest path is unique. As PhD originality is hard to separate from PhD contributions, we provide further discussion on that topic in the section 8.2 Contribution.

8.2 Contribution

In our view, we made several contributions to the field of science in this work. The contributions can be separated into two distinct categories: 1) contribution to theory and 2) contribution to practice. We elaborate on each of the two next.

8.2.1 Contribution to theory

Our contribution to theory includes the following findings:

- Complex information models can be used in annotation of clinical documents
- Concepts semantics is a function of its representation and position in reality
- Segments of reality can be merged as well as information models that represent them
- Concepts can be assigned importance in reality to assist the algorithm to achieve its goal
- Knowledge of a concept is acquired through observation of its position in its environment, the reality
- Ontology can be used as representation of reality

Each of the contributions listed above will be touched upon next.

8.2.1.1 Complex information models can be used in annotation of clinical documents

Our approach taken in the process of finding a solution to the problem presented in this work was not born purely in the realm of technology. It rather stemmed out of the field of philosophy and was guided by the fundamentals of the field of epistemology that informed and steered development of the methodology applied. We approached the problem of meaning of entity by positioning it as a unit of reality, before we partially reveal its semantics through its expansion using semantically similar concepts that surround it. We then position that very segment of reality and compare it to another segment of reality that represents another meaning of another entity. The segments are therefore regarded not just as stand-alone structures, but as pieces of a whole, that themselves are constructed of pieces connected with links that govern the whole's structure.

We are not aware of other works in the area of clinical informatics, annotation of clinical datasets in particular, that willingly ground their approach to finding a meaning of an entity into a notion that is based on a postulate that a meaning of an entity is as a function of their position and representation in reality. Neither are we aware of a work that provides justification as clear as we do in this work on the use of complex information models as clinical information annotation elements.

8.2.1.2 Concepts semantics is a function of its representation and position in reality

The idea that a concept's semantics is a function of its representation and position in reality will provide a new perspective to the researchers working in this field. That thinking signifies importance of use of composite information models as annotation entities, rather than just codes from terminology systems. That opens perspective on text as a message, or a set of messages that neither can be understood nor compared without being observed as a segment of reality. This represents a significant shift of a paradigm in the field. In our view, this contribution to theory of science is also in support of this PhD's originality.

The results of our evaluation indicate that this approach is tenable and that the methods that we use in our work are good guidelines on how to approach solution development. We hope that, as a result, the researchers that now annotate (mostly singular) entities in clinical documents using techniques based on NLP (Zech et al., 2018), bag of words (Powell et al., 2017) or even ontologies (Tchechmedjiev et al., 2018) might decide to expand their methods and consider consulting an entity's environment, a reality segment where the entity is posited, when searching for an entity's semantics.

8.2.1.3 Segments of reality can be merged as well as information models that represent them

We also offer an idea that segments of reality, as representations of concepts' environments that are represented as information models, can be merged. After creating a segment of reality around each of the concepts found in a document (through concept expansion), we merge them and create another, larger segment of reality, suggesting that segments can be merged into one. In practical terms, our method for merging of reality segments eliminates duplication of concepts in an information model and unifies smaller reality segments into one.

8.2.1.4 Concepts can be assigned importance in reality to assist the algorithm to achieve its goal

By assigning weights to concepts, we introduce the notion of importance as a characteristic of concepts or groups of concepts that form a structured setting, like a reality segment. This approach is valuable as it allows for matching and expansion preferences to be pre-defined, so they impact the algorithm's decision on the type of concepts (including siblings, descendants and ascendants) to be integrated into reality segments as expansion outputs.

We put forward that importance of a concept depends on several factors. Firstly, it depends on the algorithm's pre-set interest in a specific segment of ontology of reality that a concept belongs to. We determine that by finding the root sink of a concept's branch and we record its root's pre-defined weight. Secondly, we posit that importance of a concept depends on how many times it has been repeated in the target text. The more times the concept is mentioned in the text, the higher its importance.

Thirdly, importance of a concept depends on whether it is found in text, or it has become part of the concept's expanded structure during the expansion process. For that, we allow that weights can be assigned to each type of concept (siblings, descendants, ascendants) used in expansion with a special provision for assigning different weights to different generations of these concepts. And finally, we claim that importance of concepts is cumulative as, when merging segments of reality, we sum weights of equivalent concepts that are combined into one. We are not aware of a work that takes the same or a similar approach to determining importance of concepts in merging segments of reality.

8.2.1.5 Knowledge of a concept is acquired through observation of its position in its environment, the reality

Finding a meaning of a concept through its expansion and positioning in reality also offers an answer to one of the core questions of epistemology "How is knowledge acquired?". It suggests that an entity, or a concept, is not learned nor understood when presented on its own, but only when "explained" as a segment of reality.

8.2.1.6 Ontology can be used as representation of reality

We also put forward that an information model, ontology, can be used as a representation of reality. We have shown that discovery of meaning of entities as well as comparing entities similarities depends on our ability to represent the reality where these entities reside. Our expectation is that ontologies that are used to represent reality will Improve as representational models, increasing in complexity and improving in semantic descriptiveness over time. Specifically, we suggest that, due to its axiomatic descriptiveness and its coverage, SNOMED CT can be used to represent a segment of reality that depicts the clinical field. We offer discussion on that topic in the section 3.2 Utility of SNOMED CT as ontology of reality.

8.2.2 Contribution to practice

Our contribution to practice includes the following:

- SNOMED CT's coverage tested and scientific-community informed
- A novel method for SNOMED CT graph partitioning and parsing developed
- A method for integration of Cytoscape in a C# project using GraphML as an intermediary developed
- A novel method for comparing similarity of composite information models
- A novel method for annotation of clinical datasets using composite information models

Each of the contributions listed above will be touched upon next.

8.2.2.1 SNOMED CT's coverage tested and scientific community informed

Testing suitability of SNOMED CT in terms of its coverage was a prerequisite for the project as we needed to understand whether SNOMED CT has sufficient coverage for use as an ontology of reality. We not only tested SNOMED CT for coverage, but several other ontologies as well, providing detailed statistics and discussion on our findings and methodology (listed in section 3.3 Testing SNOMED CT coverage). The output of that work has been published as an article in an international, peer-reviewed journal with impact factor 1.833 (Zivaljevic et al., 2020). We believe that other researchers will benefit from our contribution as our findings can be reused, and our methods can be replicated in other projects. We believe that this contribution belongs to both realms, theory and practice.

8.2.2.2 A novel method for SNOMED CT graph partitioning and parsing developed

Due to the computational limitations, caused by the state of the art in the current development of information systems (or at least those available for this thesis research), we needed to find an alternative, less computationally demanding method for calculating shortest path between the SNOMED CT graph's nodes. As a result, we developed a set of methods that partition the SNOMED CT graph into smaller graphs that are reconnected when needed for shortest path calculation. We detail the technique and include pseudocode in the section 6.1.2 Finding shortest path in the SNOMED CT graph.

Although graph partitioning algorithms are not new, they are generic (Hongke Xia et al., 2010; Schlicht & Stuckenschmidt, 2007) and it is up to a researcher to customise them to suit circumstances of a particular graph at hand. Moreover, the algorithms that are tested on the SNOMED CT graph (Ochieng & Kyanda, 2018) seem to focus on partitioning only, and do not specifically define methods to guide use of new structure in finding shortest paths between the graph's nodes after partitioning.

Our algorithm is not generic, as it has been customised to take into consideration specifics of SNOMED CT, and it also includes methods for keeping track of the location of the nodes and for re-joining of required graph segments when finding the shortest path. We believe that, after our method is published, the scientific community will benefit from a detailed set of instructions on how to find shortest path in the SNOMED CT graph, both in realistic time and using standard computing resources. Furthermore, other authors include pseudocode only, while we have a C# code base to include in the published material. Including working code, in particular in C#, will assist members of the scientific community as they will be able to re-use rather than re-develop the method.

8.2.2.3 A method for integration of Cytoscape in C# project using GraphML as an intermediary developed

Cytoscape is a tool widely used for visualisation of graphs. However, it is a stand-alone application built in Java and difficult to interact with from software utilising .NET technology. To overcome this issue and to be able to use this useful application in our environment, we mastered the GraphML language and used it as an output of our application to demonstrate results of document comparison. We also created a new style specification for the Cytoscape

application that makes visual outlook of graphs more explanatory. We will publish this style as a contribution to the open source community that maintains and develops Cytoscape. We believe that this artefact will be of use to others working in the graph comparison space. We see this as applications integration that demonstrates utility of currently available technologies, including .NET, Cytoscape and the GraphML language. As there is a lack of open source C# libraries with visualisation features as comprehensive as that of Cytoscape, this alternative will be of help to the future researchers using C# as a development environment. We include discussion on how Cytoscape is used in the section 5.2 Development environment software.

8.2.2.4 A novel method for comparing similarity of composite information models

Numerous approaches measure information models' similarity by using NLP methods in comparing concept labels and/or other textual meta-data available as concept descriptions (Smaili et al., 2019; Kiourtis et al., 2018). Others expect real numbers as concept identifiers that they can compare using mathematical methods (G. Liu et al., 2018). Another school of thought takes into consideration structures of information models compared, assuming that the more structurally similar the compared information models, the more semantically similar they are (Kiourtis et al., 2018).

Our method does not have to consider these specifics as the information models compared are already standardised to the ontology of reality standards. We suggest that the concept of similarity of two composite information models is a function of semantic similarity of concepts contained in two models. Furthermore, we propose that semantic similarity of any two concepts is a function of their distance in the ontology of reality that contains them. That implies that for deciding on a similarity of two composite information models, we need a third one that realistically represents the first two information models as they are in their shared reality. We use SNOMED CT as that third information model, or as an 'ontology of reality' as we refer to it in this document.

Measuring similarity of two information models based on the similarity of their concepts is not a new notion, but is far from well researched/developed and gold standards do not exist. The example of a current work in that field is the work of Gao et al. (Gao et al., 2017) where, similar to our work, the authors compare composite information models that they convert to graph structures for distance measuring. However, they condition their work to the learning

framework where information of each ontology vertex is expressed as a vector, that they use in the process of comparison. This solution creates a new structure that takes on the function of an ontology of reality providing axioms on concepts distances. We believe that inferring relations between concepts from their appearance in two models, which might not be as comprehensive as an acceptable ontology of reality would be, raises questions of validity. Our solution relies on a well-established, specialised ontology to provide axioms governing concepts' environment.

We believe that, using our graph partitioning method, calculation of concepts' distances becomes possible for any, not just small graphs. This further enables the similarity of composite information models to be calculated as a function of distances of concepts that form the information models being compared.

8.2.2.5 A novel and complete method for annotation of clinical datasets using composite information models

We present a novel and a complete algorithm for annotation of clinical datasets using composite information models as the primary output of this work. The novelty that our approach brings is its use of composite information models, rather than just single concepts as annotation elements. We ground that approach in the field of philosophy and offer a discussion on our underlying assumptions in the section 3.1 Project goal and underlying assumptions. We also offer discussion on the systems that employ similar, but not the same strategies, namely MetaMap (Aronson, 2001), cTAKES (Savova et al., 2010) and MedLEE (Friedman et al., 1994), noting that these systems use different annotation elements (single concepts).

We believe that the outcome of our work is an evolutionary step in development of the field of annotation of clinical datasets. We offer another perspective, that takes into consideration a message rather than just a concept and a whole rather than just a single unit. Our approach is not an end, but a continuation of efforts towards achieving efficient and valid conversion of clinical free text into machine-readable information.

A practical contribution that we make in this work is a complete algorithm for automated annotation of clinical text, that includes all stages of the process, from pre-processing to the final presentation of results. We include diagrams, pseudocode and code to help future researchers to easily reuse what we developed, and we include discussion detailing not just

how but also why we did what we did. We elaborate on the sections of the algorithm throughout the chapter 4 Design and develop artefact (DSR activity 3).

We believe that this practical contribution is valuable as it is detailed and allows for easy replication and reuse of the methods we created. This in particular because the issues that we experienced in the course of this work, and that we describe here, take time and resources to be addressed. As an example, the solutions that we offer for overcoming problems imposed by high computational requirements of some of the methods will certainly save some time to the researchers inquiring in this field in the near future.

9 Appendices

9.1 Evaluation results

Iteration	$\sum Siv$	$\sum Sco$	$\sum AvgSco$
Sim1	166202	45817.70	4.187324
Dis1	161276	44487.03	4.065342
Dis2	160625	44317.32	4.049092
Dis3	161145	44464.90	4.064805
Dis4	161024	44423.71	4.059926
Dis5	161618	44592.87	4.073524
Dis6	161547	44538.12	4.070754
Dis7	161737	44603.41	4.071512
Dis8	161434	44567.02	4.076749
Dis9	161437	44534.71	4.071931
Dis10	161580	44588.14	4.077935
Dis11	161932	44669.77	4.084653
Dis12	162368	44771.58	4.087236
Dis13	162117	44657.53	4.074220
Dis14	162287	44720.07	4.077322
Dis15	162453	44737.61	4.076319
Dis16	162393	44754.76	4.082346
Dis17	162577	44794.62	4.088593
Dis18	162372	44759.80	4.085787
Dis19	162995	44866.19	4.084314
Dis20	163024	44888.49	4.091186
Dis21	162760	44803.00	4.082650
Dis22	162958	44841.41	4.086894
Dis23	163121	44927.53	4.094370
Dis24	162976	44832.65	4.084235
Dis25	162819	44816.47	4.088348
Dis26	163362	44995.80	4.102088
Dis27	163451	44972.42	4.098462
Dis28	163427	44988.36	4.101784
Dis29	163421	44926.99	4.091711
Dis30	163494	44978.54	4.100141
Dis31	163197	44936.48	4.098174
Dis32	163052	44882.64	4.092891
Dis33	162991	44886.13	4.094329
Dis34	163202	44917.12	4.093049
Dis35	163044	44899.90	4.090735
Dis36	162710	44820.82	4.086508
Dis37	162910	44864.82	4.089774
Dis38	162789	44836.11	4.088647
Dis39	162795	44818.79	4.084833
Dis40	162323	44699.19	4.076161
Dis41	162404	44742.33	4.080840
Dis42	162311	44700.20	4.074396
Dis43	162389	44716.11	4.073619
Dis44	162429	44735.22	4.075359
Dis45	162064	44671.65	4.073279
Dis46	161868	44623.15	4.070341
Dis47	161616	44578.89	4.067788
Dis48	161552	44540.23	4.066116
Dis49	161533	44544.45	4.066501

Table 16 - Evaluation results

9.2 Statistical tables

n	Lower limit		Upper limit	
	$P = 95\%$	$P = 99\%$	$P = 95\%$	$P = 99\%$
5	2.15	2.02	2.753	2.803
6	2.28	2.15	3.012	3.095
7	2.40	2.26	3.222	3.338
8	2.50	2.35	3.399	3.543
9	2.59	2.44	3.552	3.720
10	2.67	2.51	3.685	3.875
11	2.74	2.58	3.80	4.012
12	2.80	2.64	3.91	4.134
13	2.86	2.70	4.00	4.244
14	2.92	2.75	4.09	4.34
15	2.97	2.80	4.17	4.44
16	3.01	2.84	4.24	4.52
17	3.06	2.88	4.31	4.60
18	3.10	2.92	4.37	4.67
19	3.14	2.96	4.43	4.74
20	3.18	2.99	4.49	4.80
25	3.34	3.15	4.71	5.06
30	3.47	3.27	4.89	5.56
35	3.58	3.38	5.04	5.42
40	3.67	3.47	5.16	5.56
45	3.75	3.55	5.26	5.67
50	3.83	3.62	5.35	5.77
55	3.90	3.69	5.43	5.86
60	3.96	3.75	5.51	5.94

Table 17 - Significance table for testing normal distribution according to David

9.3 Custom graph class

```
public class SubGraph : IDisposable
{
   public SubGraph(String _Name, String _Root_SNOMED_Code)
   {
      Name = _Name; Root_SNOMED_Code = _Root_SNOMED_Code;
      Elements = new HashSet<Pre_Edge_of_String>(new Pre_Edge_of_String_Comparer());
      Weight_Function = (S, T) => 1;
      Parent_Graph_Names = new HashSet<string>();
   }
   public String Name { get; set; }
   public HashSet<String> Parent_Graph_Names { get; set; }
   public String Root_SNOMED_Code { get; set; }
   [JsonIgnore]
   public Func<String, String, double> Weight_Function { get; set; }
   public HashSet<String> Get_Sources_And_Targets()
   {
      return new HashSet<string>(Get_Sources().Union(Get_Targets()));
   }
   public HashSet<String> Get_Sources()
   {
      return new HashSet<String>(Elements.Select(a => a.Source));
   }
   public HashSet<String> Get_Targets()
   {
      return new HashSet<String>(Elements.Select(a => a.Target));
   }
   [JsonIgnore]
   private IBidirectionalGraph<String, SEdge<String>> _Quick_Graph;
   public IBidirectionalGraph<String, SEdge<String>> Get_Quick_Graph(bool Add_Both_Directions = false)
   {
      if (_Quick_Graph == default)
      {
         HashSet<SEdge<String>> Edges;
         if (Add_Both_Directions)
         {
            Edges = new HashSet<SEdge<string>>(Elements.SelectMany(a => new HashSet<SEdge<String>>()
                  { new SEdge<string>(a.Source, a.Target), new SEdge<string>(a.Target, a.Source) }));
         }
         else
         {
            Edges = new HashSet<SEdge<string>>(Elements.Select(a => new SEdge<string>(a.Source, a.Target)));
         }
         _Quick_Graph = Edges.ToArray().ToBidirectionalGraph<String, SEdge<String>>(false);
      }
      return _Quick_Graph;
   }
   public void Add_Elements(IEnumerable<Pre_Edge_of_String> _Elements)
   {
      _Connections_Cache = default; _Quick_Graph = default;
      _Nodes = default;
      Clear_Matrix();
      Elements.UnionWith(_Elements);
   }
   public void Add_Sub_Graph(IEnumerable<SubGraph> Sub_Graphs, bool Check = false)
   {
      foreach (SubGraph S in Sub_Graphs) { Add_Sub_Graph(S, Check); }
   }
   public bool Is_SubGraph_Of_This(SubGraph Sub_Graph)
   {
      if (Sub_Graph.Get_Targets().Intersect(Get_Sources()).Count() < 1) { return false; } else { return true; }
   }
   public void Add_Sub_Graph(SubGraph Sub_Graph, bool Check = true)
   {
      if (Check) { if (!Is_SubGraph_Of_This(Sub_Graph)) { return; } }
      Add_Elements(Sub_Graph.Elements);
   }
   public HashSet<Pre_Edge_of_String> Elements { get; }
   private AdjacencyGraph<string, Edge<string>> Q_Graph;
   public double Get_Shortest_Path_Distance(String Source, String Target, int hoops = 5)
   {
```

```
    if (Q_Graph == default) { Create_Q_Graph(); }
    double edgeCost(Edge<string> e) => 1; double res = 100;
    TryFunc<String, IEnumerable<Edge<String>>> tryGetPaths = Q_Graph.ShortestPathsDijkstra(edgeCost, Source);
    if (tryGetPaths(Target, out IEnumerable<Edge<String>> path)) { res = path.Count(); }
    return res;
}
private void Create_Q_Graph()
{
    Q_Graph = new AdjacencyGraph<string, Edge<string>>();
    Q_Graph = Elements.Select(a => new Edge<string>(a.Source, a.Target)).ToAdjacencyGraph<String, Edge<String>>();
}
[JsonIgnore]
public HashSet<Matrix_Node> Matrix => CacheExtensions.GetOrStore(
        this.Name + "_Matrix", _CreateMatrix, Caching_Location.File);
public void Clear_Matrix() { CacheExtensions.DeCache(this.Name + "_Matrix"); }
private HashSet<Matrix_Node> _CreateMatrix()
{
    WinApi.TimeBeginPeriod(1);
    //add weights
    int c = Nodes().Count();
    Dictionary<String, int> tmp_Fast = new Dictionary<String, int>(c * c, StringComparer.Ordinal);
    String k;
    foreach (Pre_Edge_of_String a in Elements)
    {
        k = a.Source + "_" + a.Target;
        if (!tmp_Fast.ContainsKey(k)) { tmp_Fast.Add(k, 1); }
    }
    foreach (String S in Nodes())
    {
        foreach (String T in Nodes())
        {
            k = S + "_" + T;
            if (!tmp_Fast.ContainsKey(k)) { if (S == T) { tmp_Fast.Add(k, 0); } else { tmp_Fast.Add(k, 10); } }
        }
    }
    //Floyd warshal
    //Matrix_Node tmp1; Matrix_Node tmp2; Matrix_Node tmp3;
    String key_main; int key_calculated;
    HashSet<String> N = new HashSet<string>(Nodes());
    foreach (String A1 in N)
    {
        //System.Diagnostics.Stopwatch watch = System.Diagnostics.Stopwatch.StartNew();
        foreach (String A2 in N)
        {
            foreach (String A3 in N)
            {
                if (A2 != A3)
                {
                    key_main = A2 + "_" + A3;
                    key_calculated = tmp_Fast[A2 + "_" + A1] + tmp_Fast[A1 + "_" + A3];
                    if (key_calculated < tmp_Fast[key_main])
                    {
                        tmp_Fast[key_main] = key_calculated; //tmp3.Prev_Node = A1;
                    }
                }
            }
        }
        //MessageBox.Show(watch.ElapsedMilliseconds.ToString());
    }
    //Temp_Print_Matrix(M);
    WinApi.TimeEndPeriod(1);
    return new HashSet<Matrix_Node>(tmp_Fast.Select(a => new Matrix_Node(
        a.Key.Split('_')[0], a.Key.Split('_')[0], "", a.Value)));
}
[JsonIgnore]
private HashSet<String> _Nodes;
public HashSet<String> Nodes()
{
    if (_Nodes == default) { _Nodes = new HashSet<string>(Elements.Select(a => a.Source));
        _Nodes.UnionWith(Elements.Select(b => b.Target)); }
    return _Nodes;
}
[JsonIgnore]
public HashSet<Pre_Edge_of_String> Connections => _Connections_Cache != default ?
```

```
            _Connections_Cache : _Connections();
    [JsonIgnore]
    private HashSet<Pre_Edge_of_String> _Connections_Cache;
    private HashSet<Pre_Edge_of_String> _Connections()
    {
        return new HashSet<Pre_Edge_of_String>(
            Elements.Where(a => a.Target != Root_SNOMED_Code && Elements.FirstOrDefault(
             b => b.Source == a.Target) == default), new Pre_Edge_of_String_Comparer());
    }
    private void Temp_Print_Matrix(HashSet<Matrix_Node> M)
    {
        foreach (Matrix_Node m in M)
        {
            if (m.Weight < 1000 && m.Weight != 0)
            {
                Console.WriteLine(m.Source + " -> " + m.Target + " = " + m.Weight.ToString() + " - " + m.Prev_Node);
            }
        }
    }
    public String Create_JSON_Graph_String()
    {
        JArray JA_Nodes = new JArray();
        foreach (String Source in Get_Sources()) { JA_Nodes.Add(JSON_Node_Data(Source, false)); }
        foreach (String Target in Get_Targets().Except(Get_Sources())) { JA_Nodes.Add(JSON_Node_Data(Target, false)); }
        JProperty Nodes = new JProperty("nodes", JA_Nodes);
        JArray JA_Edges = new JArray();
        foreach (Pre_Edge_of_String Edge in Elements)
            {
                    JA_Edges.Add(JSON_Edge_Data(Edge, false, Get_Sources().Contains(Edge.Target) ? "blue" : "red"));
            }
        JProperty Edges = new JProperty("edges", JA_Edges);
        JObject El = new JObject(); El.Add(Nodes); El.Add(Edges);
        JProperty Fin_Prop = new JProperty("elements", El);
        return JsonConvert.SerializeObject(new JObject(Fin_Prop));
    }
    private JObject JSON_Node_Data(String Node, bool Selected)
    {
        JObject J = new JObject { { "data", JSON_Node(Node) }, { "selected", Selected.ToString().ToLower() } };
        return J;
    }
    private JObject JSON_Node(String Node)
    {
        JObject J = new JObject { { "id", Node }, { "name", Node } };
        return J;
    }
    private JObject JSON_Edge_Data(Pre_Edge_of_String Edge, bool Selected, String Color)
    {
        JObject J = new JObject { { "data", JSON_Edge(Edge, Color) }, { "selected", Selected.ToString() } };
        return J;
    }
    private JObject JSON_Edge(Pre_Edge_of_String Edge, String Color)
    {
        JObject J = new JObject { { "id", Edge.Source + "-" + Edge.Target }, { "source", Edge.Source },
            { "target", Edge.Target }, { "connection", Edge.Connection }, { "color", Color } };
        return J;
    }
    public void Dispose()
    {
        Clear_Matrix();
    }
}
```

9.4 EHR GraphML code

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml   xmlns="http://graphml.graphdrawing.org/xmlns"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"                                    xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key xmlns="" id="N" for="node" attr.name="SNOMED_CODE" attr.type="string"/>
  <key xmlns="" id="R" for="node" attr.name="rank" attr.type="double"/>
  <key xmlns="" id="D" for="node" attr.name="descriptions" attr.type="string"/>
  <key xmlns="" id="W" for="node" attr.name="weight" attr.type="double"/>
  <key xmlns="" id="L" for="node" attr.name="location" attr.type="string"/>
  <key xmlns="" id="Clr" for="node" attr.name="color" attr.type="string">
    <default>blue</default>
  </key>
  <key xmlns="" id="Dist" for="edge" attr.name="Distance_To_Target" attr.type="double"/>
  <graph xmlns="" id="G_110" edgedefault="directed">
    <node id="B_199047001">
      <data key="N">199047001</data>
      <data key="R">0</data>
      <data key="D">False labor (finding)</data>
      <data key="W">1.5</data>
      <data key="L">Body</data>
      <data key="Clr">red</data>
    </node>
    <node id="B_118212000">
      <data key="N">118212000</data>
      <data key="R">1</data>
      <data key="D">Parity finding (finding)</data>
      <data key="W">1.1</data>
      <data key="L">Body</data>
      <data key="Clr">red</data>
    </node>
    <node id="B_366322004">
      <data key="N">366322004</data>
      <data key="R">2</data>
      <data key="D">Finding of estimated date of delivery (finding)</data>
      <data key="W">1.1</data>
      <data key="L">Body</data>
      <data key="Clr">red</data>
    </node>
    <node id="B_70028003">
      <data key="N">70028003</data>
      <data key="R">3</data>
      <data key="D">Vertex presentation (finding)</data>
```

```
    <data key="W">1.0054945054945055</data>
    <data key="L">Body</data>
    <data key="Clr">red</data>
  </node>
  <node id="B_6096002">
    <data key="N">6096002</data>
    <data key="R">4</data>
    <data key="D">Breech presentation (finding)</data>
    <data key="W">1.0054347826086956</data>
    <data key="L">Body</data>
    <data key="Clr">red</data>
  </node>
  <node id="H_77386006">
    <data key="N">77386006</data>
    <data key="R">0</data>
    <data key="D">Pregnant (finding)</data>
    <data key="W">1</data>
    <data key="L">Header</data>
  </node>
  <node id="H_6383007">
    <data key="N">6383007</data>
    <data key="R">1</data>
    <data key="D">Premature labor (finding)</data>
    <data key="W">1</data>
    <data key="L">Header</data>
  </node>
  <node id="H_199047001">
    <data key="N">199047001</data>
    <data key="R">2</data>
    <data key="D">False labor (finding)</data>
    <data key="W">1</data>
    <data key="L">Header</data>
  </node>
  <node id="H_118185001">
    <data key="N">118185001</data>
    <data key="R">3</data>
    <data key="D">Finding related to pregnancy (finding)</data>
    <data key="W">0.2</data>
    <data key="L">Header</data>
  </node>
  <node id="H_289909005">
    <data key="N">289909005</data>
    <data key="R">4</data>
    <data key="D">Labor, function (observable entity)</data>
```

```
      <data key="W">0.2</data>
      <data key="L">Header</data>
    </node>
    <edge id="B_199047001-H_77386006" source="B_199047001" target="H_77386006">
      <data key="Dist">5</data>
    </edge>
    <edge id="B_199047001-H_6383007" source="B_199047001" target="H_6383007">
      <data key="Dist">2</data>
    </edge>
    <edge id="B_199047001-H_199047001" source="B_199047001" target="H_199047001">
      <data key="Dist">0</data>
    </edge>
    <edge id="B_199047001-H_118185001" source="B_199047001" target="H_118185001">
      <data key="Dist">4</data>
    </edge>
    <edge id="B_118212000-H_77386006" source="B_118212000" target="H_77386006">
      <data key="Dist">3</data>
    </edge>
    <edge id="B_118212000-H_6383007" source="B_118212000" target="H_6383007">
      <data key="Dist">6</data>
    </edge>
    <edge id="B_118212000-H_199047001" source="B_118212000" target="H_199047001">
      <data key="Dist">6</data>
    </edge>
    <edge id="B_118212000-H_118185001" source="B_118212000" target="H_118185001">
      <data key="Dist">2</data>
    </edge>
    <edge id="B_366322004-H_77386006" source="B_366322004" target="H_77386006">
      <data key="Dist">3</data>
    </edge>
    <edge id="B_366322004-H_6383007" source="B_366322004" target="H_6383007">
      <data key="Dist">6</data>
    </edge>
    <edge id="B_366322004-H_199047001" source="B_366322004" target="H_199047001">
      <data key="Dist">6</data>
    </edge>
    <edge id="B_366322004-H_118185001" source="B_366322004" target="H_118185001">
      <data key="Dist">2</data>
    </edge>
    <edge id="B_70028003-H_77386006" source="B_70028003" target="H_77386006">
      <data key="Dist">5</data>
    </edge>
    <edge id="B_70028003-H_6383007" source="B_70028003" target="H_6383007">
      <data key="Dist">8</data>
```

```
    </edge>
    <edge id="B_70028003-H_199047001" source="B_70028003" target="H_199047001">
      <data key="Dist">8</data>
    </edge>
    <edge id="B_70028003-H_118185001" source="B_70028003" target="H_118185001">
      <data key="Dist">4</data>
    </edge>
    <edge id="B_6096002-H_77386006" source="B_6096002" target="H_77386006">
      <data key="Dist">5</data>
    </edge>
    <edge id="B_6096002-H_6383007" source="B_6096002" target="H_6383007">
      <data key="Dist">8</data>
    </edge>
    <edge id="B_6096002-H_199047001" source="B_6096002" target="H_199047001">
      <data key="Dist">8</data>
    </edge>
    <edge id="B_6096002-H_118185001" source="B_6096002" target="H_118185001">
      <data key="Dist">4</data>
    </edge>
  </graph>
</graphml>
```

Formatted using the tool provided by (OneLogin, 2021).

9.5 Style sheet used in Cytoscape

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vizmap id="VizMap-2020_11_05-23_42" documentVersion="3.0">
  <visualStyle name="default">
    <network>
      <visualProperty default="0.0" name="NETWORK_CENTER_Z_LOCATION"/>
      <visualProperty default="550.0" name="NETWORK_WIDTH"/>
      <visualProperty default="#FFFFFF" name="NETWORK_BACKGROUND_PAINT"/>
      <visualProperty default="false" name="NETWORK_FORCE_HIGH_DETAIL"/>
      <visualProperty default="" name="NETWORK_TITLE"/>
      <visualProperty default="false" name="NETWORK_NODE_LABEL_SELECTION"/>
      <visualProperty default="0.0" name="NETWORK_CENTER_X_LOCATION"/>
      <visualProperty default="400.0" name="NETWORK_HEIGHT"/>
      <visualProperty default="true" name="NETWORK_EDGE_SELECTION"/>
      <visualProperty default="false" name="NETWORK_ANNOTATION_SELECTION"/>
      <visualProperty default="0.0" name="NETWORK_DEPTH"/>
      <visualProperty default="true" name="NETWORK_NODE_SELECTION"/>
      <visualProperty default="550.0" name="NETWORK_SIZE"/>
      <visualProperty default="0.0" name="NETWORK_CENTER_Y_LOCATION"/>
      <visualProperty default="1.0" name="NETWORK_SCALE_FACTOR"/>
    </network>
    <node>
      <dependency value="true" name="nodeCustomGraphicsSizeSync"/>
      <dependency value="false" name="nodeSizeLocked"/>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_8"/>
      <visualProperty default="255" name="NODE_BORDER_TRANSPARENCY"/>
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_2,    name=Node
Custom Paint 2)" name="NODE_CUSTOMPAINT_2"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_9"/>
      <visualProperty default="115.0" name="NODE_WIDTH"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_8"/>
      <visualProperty default="110.0" name="NODE_LABEL_WIDTH"/>
      <visualProperty default="0.0" name="NODE_BORDER_WIDTH"/>
      <visualProperty default="true" name="NODE_NESTED_NETWORK_IMAGE_VISIBLE"/>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_9"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_1"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_6"/>
      <visualProperty default="#000000" name="NODE_LABEL_COLOR"/>
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_4,    name=Node
Custom Paint 4)" name="NODE_CUSTOMPAINT_4"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_2"/>
```

```
      <visualProperty default="" name="NODE_LABEL">
        <passthroughMapping attributeName="descriptions" attributeType="string"/>
      </visualProperty>
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_8,    name=Node
Custom Paint 8)" name="NODE_CUSTOMPAINT_8"/>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_2"/>
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_7,    name=Node
Custom Paint 7)" name="NODE_CUSTOMPAINT_7"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_6"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_1"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_5"/>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_3"/>
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_6,    name=Node
Custom Paint 6)" name="NODE_CUSTOMPAINT_6"/>
      <visualProperty default="35.0" name="NODE_HEIGHT"/>
      <visualProperty default="ROUND_RECTANGLE" name="COMPOUND_NODE_SHAPE"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_5"/>
      <visualProperty default="35.0" name="NODE_SIZE"/>
      <visualProperty default="10.0" name="NODE_Y_LOCATION">
        <discreteMapping attributeName="location" attributeType="string">
          <discreteMappingEntry attributeValue="Header" value="200.0"/>
          <discreteMappingEntry attributeValue="Body" value="0.0"/>
        </discreteMapping>
      </visualProperty>
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_9,    name=Node
Custom Paint 9)" name="NODE_CUSTOMPAINT_9"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_7"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_2"/>
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_3,    name=Node
Custom Paint 3)" name="NODE_CUSTOMPAINT_3"/>
      <visualProperty default="SansSerif.plain,plain,12" name="NODE_LABEL_FONT_FACE"/>
      <visualProperty default="#1E90FF" name="NODE_PAINT"/>
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_1,    name=Node
Custom Paint 1)" name="NODE_CUSTOMPAINT_1"/>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_5"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_9"/>
      <visualProperty default="" name="NODE_TOOLTIP">
        <passthroughMapping attributeName="rank" attributeType="float"/>
      </visualProperty>
      <visualProperty default="10.0" name="COMPOUND_NODE_PADDING"/>
      <visualProperty default="0.0" name="NODE_Z_LOCATION"/>
      <visualProperty default="#CCCCCC" name="NODE_BORDER_PAINT"/>
      <visualProperty default="7" name="NODE_LABEL_FONT_SIZE"/>
      <visualProperty default="100" name="NODE_TRANSPARENCY"/>
```

```
      <visualProperty    default="DefaultVisualizableVisualProperty(id=NODE_CUSTOMPAINT_5,    name=Node
Custom Paint 5)" name="NODE_CUSTOMPAINT_5"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_7"/>
      <visualProperty default="#89D0F5" name="NODE_FILL_COLOR">
        <passthroughMapping attributeName="color" attributeType="string"/>
      </visualProperty>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_4"/>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_6"/>
      <visualProperty default="ROUND_RECTANGLE" name="NODE_SHAPE"/>
      <visualProperty default="0.0" name="NODE_DEPTH"/>
      <visualProperty default="SOLID" name="NODE_BORDER_STROKE"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_LABEL_POSITION"/>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_1"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_3"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_8"/>
      <visualProperty default="false" name="NODE_SELECTED"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_3"/>
      <visualProperty default="org.cytoscape.ding.customgraphics.NullCustomGraphics,0,[ Remove Graphics
]," name="NODE_CUSTOMGRAPHICS_7"/>
      <visualProperty default="0.0" name="NODE_X_LOCATION">
        <discreteMapping attributeName="rank" attributeType="float">
          <discreteMappingEntry attributeValue="0.0" value="0.0"/>
          <discreteMappingEntry attributeValue="1.0" value="150.0"/>
          <discreteMappingEntry attributeValue="2.0" value="300.0"/>
          <discreteMappingEntry attributeValue="4.0" value="600.0"/>
          <discreteMappingEntry attributeValue="3.0" value="450.0"/>
        </discreteMapping>
      </visualProperty>
      <visualProperty default="#FFFF00" name="NODE_SELECTED_PAINT"/>
      <visualProperty default="C,C,c,0.00,0.00" name="NODE_CUSTOMGRAPHICS_POSITION_4"/>
      <visualProperty default="true" name="NODE_VISIBLE"/>
      <visualProperty default="50.0" name="NODE_CUSTOMGRAPHICS_SIZE_4"/>
      <visualProperty default="255" name="NODE_LABEL_TRANSPARENCY"/>
    </node>
    <edge>
      <dependency value="false" name="arrowColorMatchesEdge"/>
      <visualProperty default="#000000" name="EDGE_LABEL_COLOR"/>
      <visualProperty default="#323232" name="EDGE_PAINT"/>
      <visualProperty default="#404040" name="EDGE_UNSELECTED_PAINT"/>
      <visualProperty default="#33FF00" name="EDGE_STROKE_UNSELECTED_PAINT"/>
      <visualProperty name="EDGE_BEND"/>
      <visualProperty default="#FFFF00" name="EDGE_SOURCE_ARROW_SELECTED_PAINT"/>
      <visualProperty default="false" name="EDGE_SELECTED"/>
```

```
        <visualProperty default="1.0" name="EDGE_WIDTH"/>
        <visualProperty default="255" name="EDGE_LABEL_TRANSPARENCY"/>
        <visualProperty default="255" name="EDGE_TRANSPARENCY"/>
        <visualProperty default="#000000" name="EDGE_SOURCE_ARROW_UNSELECTED_PAINT"/>
        <visualProperty default="#000000" name="EDGE_TARGET_ARROW_UNSELECTED_PAINT"/>
        <visualProperty default="#FFFF00" name="EDGE_TARGET_ARROW_SELECTED_PAINT"/>
        <visualProperty default="#FF0000" name="EDGE_SELECTED_PAINT"/>
        <visualProperty default="NONE" name="EDGE_TARGET_ARROW_SHAPE"/>
        <visualProperty default="#FF0000" name="EDGE_STROKE_SELECTED_PAINT"/>
        <visualProperty default="" name="EDGE_LABEL">
          <passthroughMapping attributeName="Distance_To_Target" attributeType="float"/>
        </visualProperty>
        <visualProperty default="Dialog.plain,plain,10" name="EDGE_LABEL_FONT_FACE"/>
        <visualProperty default="true" name="EDGE_CURVED"/>
        <visualProperty default="SOLID" name="EDGE_LINE_TYPE"/>
        <visualProperty default="6.0" name="EDGE_TARGET_ARROW_SIZE"/>
        <visualProperty default="true" name="EDGE_VISIBLE"/>
        <visualProperty default="" name="EDGE_TOOLTIP"/>
        <visualProperty default="200.0" name="EDGE_LABEL_WIDTH"/>
        <visualProperty default="12" name="EDGE_LABEL_FONT_SIZE"/>
        <visualProperty default="6.0" name="EDGE_SOURCE_ARROW_SIZE"/>
        <visualProperty default="NONE" name="EDGE_SOURCE_ARROW_SHAPE"/>
    </edge>
  </visualStyle>
</vizmap>
```

Formatted using the tool provided by (OneLogin, 2021).

9.6 Co-authorship form

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postoradinfo@auckland.ac.nz

Co-Authorship Form

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work. Section 6.3 - Testing SNOMED CT Coverage (Zivaljevic, A., Atalag, K., & Warren, J. (2020). Utility of SNOMED CT in automated expansion of clinical terms in discharge summaries: Testing issues of coverage. Health Information Management Journal, 1833358320934528. https://doi.org/10.1177/1833358320934528)

Nature of contribution by PhD candidate	Author, convceived and implemented study
Extent of contribution by PhD candidate (%)	95

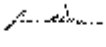
CO-AUTHORS

Name	Nature of Contribution
Jim Warren	Reviewing, editing; advice on study design
Koray Atalag	Reviewing, editing
David Nickerson	Reviewing, editing

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Jim Warren		29.01.2021
Koray Atalag		29.01.2021
David Nickerson		29.01.2021

Last updated: 28 November 2017

9.7 References

Alani, H., Sanghee Kim, Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H., & Shadbolt, N. R. (2003). Automatic ontology-based knowledge extraction from Web documents. *IEEE Intelligent Systems*, *18*(1), 14–21. https://doi.org/10.1109/MIS.2003.1179189

Al-garadi, M. A., Khan, M. S., Varathan, K. D., Mujtaba, G., & Al-Kabsi, A. M. (2016). Using online social networks to track a pandemic: A systematic review. *Journal of Biomedical Informatics*, *62*, 1–11. https://doi.org/10.1016/j.jbi.2016.05.005

Al-Mubaid, H., & Nguyen, H. A. (2009). Measuring Semantic Similarity Between Biomedical Concepts Within Multiple Ontologies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *39*(4), 389–398. https://doi.org/10.1109/TSMCC.2009.2020689

Apache. (2021). *Apache Commons – Apache Commons*. https://commons.apache.org/

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program. *Proceedings of the AMIA Symposium*, 17–21.

Australian eResearch Initiative. (2021). About NECTAR. *Nectar*. https://nectar.org.au/about/

Avorn, J. (2008). *Powerful Medicines: The Benefits, Risks, and Costs of Prescription Drugs*. Knopf Doubleday Publishing Group.

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner, W. A., Cohen, K. B., Verspoor, K., Blake, J. A., & Hunter, L. E. (2012). Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, *13*(1), 161. https://doi.org/10.1186/1471-2105-13-161

Baker, S., Silins, I., Guo, Y., Ali, I., Högberg, J., Stenius, U., & Korhonen, A. (2016). Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, *32*(3), 432–440. https://doi.org/10.1093/bioinformatics/btv585

Bakken, S., Hyun, S., Friedman, C., & Johnson, S. B. (2005). ISO reference terminology models for nursing: Applicability for natural language processing of nursing narratives. *International Journal of Medical Informatics*, *74*(7), 615–622. https://doi.org/10.1016/j.ijmedinf.2005.01.002

Barbehenn, M. (1998). A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. *IEEE Transactions on Computers*, *47*(2), 263-. https://doi.org/10.1109/12.663776

Barnett, V., & Lewis, T. (1984). Outliers in statistical data. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, Chichester: Wiley, 1984, 2nd Ed.* http://adsabs.harvard.edu/abs/1984osd..book.....B

Batet, M., Sánchez, D., & Valls, A. (2011). An ontology-based measure to compute semantic similarity in biomedicine. *Journal of Biomedical Informatics*, *44*(1), 118–125. https://doi.org/10.1016/j.jbi.2010.09.002

Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers Inc.

Berkhin, P. (2005). A Survey on PageRank Computing. *Internet Mathematics*, *2*(1), 73–120. https://doi.org/10.1080/15427951.2005.10129098

Bietenbeck, A., Boeker, M., & Schulz, S. (2018). NPU, LOINC, and SNOMED CT: A comparison of terminologies for laboratory results reveals individual advantages and a lack of possibilities to encode interpretive comments. *LaboratoriumsMedizin*, *42*(6), 267–275. https://doi.org/10.1515/labmed-2018-0103

Boag, W., Sergeeva, E., Kulshreshtha, S., Szolovits, P., Rumshisky, A., & Naumann, T. (2018). CliNER 2.0: Accessible and Accurate Clinical Concept Extraction. *ArXiv:1803.02245 [Cs]*. http://arxiv.org/abs/1803.02245

Bodenreider, J. (2018). *The New SNOMED CT International Medicinal Product Model*. http://ceur-ws.org/Vol-2285/ICBO_2018_paper_36.pdf

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, *32*(Database issue), D267–D270. https://doi.org/10.1093/nar/gkh061

Bona, J. P., & Ceusters, W. (2018). Mismatches between major subhierarchies and semantic tags in SNOMED CT. *Journal of Biomedical Informatics*, *81*, 1–15. https://doi.org/10.1016/j.jbi.2018.02.009

Botsis, T., Hartvigsen, G., Chen, F., & Weng, C. (2010). Secondary Use of EHR: Data Quality Issues and Informatics Opportunities. *Summit on Translational Bioinformatics*, *2010*, 1–5.

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches †. *Energies*, *11*(7), 1636. https://doi.org/10.3390/en11071636

Brandes, U., Eiglsperger, M., Herman, I., Himsolt, M., & Marshall, M. S. (2002). GraphML Progress Report Structural Layer Proposal. In P. Mutzel, M. Jünger, & S. Leipert (Eds.), *Graph Drawing* (pp. 501–512). Springer. https://doi.org/10.1007/3-540-45848-4_59

Brandes, U., Eiglsperger, M., Lerner, J., & Pich, C. (2013). *Graph markup language (GraphML)*.

Chalapathy, R., Borzeshi, E. Z., & Piccardi, M. (2016). Bidirectional LSTM-CRF for Clinical Concept Extraction. *ArXiv:1611.08373 [Cs, Stat]*. http://arxiv.org/abs/1611.08373

Chandler, D. (2017). *Semiotics: The Basics*. Taylor & Francis.

Chapman, W. W., Dowling, J. N., & Hripcsak, G. (2008). Evaluation of training with an annotation schema for manual annotation of clinical conditions from emergency department reports. *International Journal of Medical Informatics*, *77*(2), 107–113. https://doi.org/10.1016/j.ijmedinf.2007.01.002

Chen, L., & Friedman, C. (2004). Extracting phenotypic information from the literature via natural language processing. *Medinfo*, 758–762.

Chiang, J.-H., Lin, J.-W., & Yang, C.-W. (2010). Automated evaluation of electronic discharge notes to assess quality of care for cardiovascular diseases using Medical Language Extraction and Encoding System (MedLEE). *Journal of the American Medical Informatics Association*, *17*(3), 245–252. https://doi.org/10.1136/jamia.2009.000182

Cimino, J. J., & Barnett, G. O. (1990). Automated translation between medical terminologies using semantic definitions. *M. D. Computing*, *7*(2), 104–109.

Clarke, G., & Lunt, I. (2014). The concept of 'originality' in the Ph.D.: How is it interpreted by examiners? *Assessment & Evaluation in Higher Education*, *39*(7), 803–820. https://doi.org/10.1080/02602938.2013.870970

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(ARTICLE), 2493–2537.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.

D'AGOSTINO, R., & PEARSON, E. S. (1973). Tests for departure from normality. Empirical results for the distributions of b2 and √b1. *Biometrika*, *60*(3), 613–622. https://doi.org/10.1093/biomet/60.3.613

David, H. A., Hartley, H. O., & Pearson, E. S. (1954). The Distribution of the Ratio, in a Single Normal Sample, of Range to Standard Deviation. *Biometrika*, *41*(3/4), 482–493. https://doi.org/10.2307/2332728

de Halleux, J. (2020). *QuickGraph* [C#]. YaccConstructor. https://github.com/YaccConstructor/QuickGraph (Original work published 2014)

Deleger, L., Molnar, K., Savova, G., Xia, F., Lingren, T., Li, Q., Marsolo, K., Jegga, A., Kaiser, M., Stoutenborough, L., & Solti, I. (2013). Large-scale evaluation of automated clinical note de-identification and its impact on information extraction. *Journal of the American Medical Informatics Association: JAMIA*, *20*(1), 84–94. https://doi.org/10.1136/amiajnl-2012-001012

Deo, N., & Pang, C.-Y. (1984). Shortest-path algorithms: Taxonomy and annotation. *Networks*, *14*(2), 275–323. https://doi.org/10.1002/net.3230140208

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

Dijkstra, E. (1959). Dijkstra's algorithm. *Dutch Scientist Dr. Edsger Dijkstra Network Algorithm: Http://En. Wikipedia. Org/Wiki/Dijkstra's_algorithm*.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271.

Dtosato. (2014). *c#—.NET Binary Serialization in QuickGraph 3.6*. Stack Overflow. https://stackoverflow.com/questions/15898280/net-binary-serialization-in-quickgraph-3-6

eduGAIN. (2021). *EduGAIN – enabling worldwide access*. https://edugain.org/

Elkin, P. L., Brown, S. H., Husser, C. S., Bauer, B. A., Wahner-Roedler, D., Rosenbloom, S. T., & Speroff, T. (2006). Evaluation of the Content Coverage of SNOMED CT: Ability of SNOMED Clinical Terms to Represent Clinical Problem Lists. *Mayo Clinic Proceedings*, *81*(6), 741–748. https://doi.org/10.4065/81.6.741

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1915–1929. https://doi.org/10.1109/TPAMI.2012.231

Fleiss, J. L., Levin, B., Paik, M. C., & others. (1981). The measurement of interrater agreement. *Statistical Methods for Rates and Proportions*, *2*(212–236), 22–23.

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Communications of the ACM*, *5*(6), 345.

Formica, A. (2006). Ontology-based concept similarity in Formal Concept Analysis. *Information Sciences*, *176*(18), 2624–2641. https://doi.org/10.1016/j.ins.2005.11.014

Fortinet. (2021). *Fortinet | Enterprise Security Without Compromise*. Fortinet. https://www.fortinet.com

Fox, C. (1989). A stop list for general text. *ACM SIGIR Forum*, *24*(1–2), 19–21. https://doi.org/10.1145/378881.378888

Friedman, C., Alderson, P. O., Austin, J. H. M., Cimino, J. J., & Johnson, S. B. (1994). A General Natural-language Text Processor for Clinical Radiology. *Journal of the American Medical Informatics Association*, *1*(2), 161–174. https://doi.org/10.1136/jamia.1994.95236146

Friedman, C., Knirsch, C., Shagina, L., & Hripcsak, G. (1999). Automating a severity score guideline for community-acquired pneumonia employing medical language processing of discharge summaries. *Proceedings of the AMIA Symposium*, 256–260.

Friedrich, S., & Dalianis, H. (2015). Adverse drug event classification of health records using dictionary based pre-processing and machine learning. *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, 121–130.

Fu, S., Chen, D., He, H., Liu, S., Moon, S., Peterson, K. J., Shen, F., Wang, L., Wang, Y., Wen, A., Zhao, Y., Sohn, S., & Liu, H. (2020). Clinical concept extraction: A methodology review. *Journal of Biomedical Informatics*, *109*, 103526. https://doi.org/10.1016/j.jbi.2020.103526

Gao, W., Farahani, M. R., Aslam, A., & Hosamani, S. (2017). Distance learning techniques for ontology similarity measuring and ontology mapping. *Cluster Computing*, *20*(2), 959–968. https://doi.org/10.1007/s10586-017-0887-3

Gehrmann, S., Dernoncourt, F., Li, Y., Carlson, E. T., Wu, J. T., Welt, J., Jr, J. F., Moseley, E. T., Grant, D. W., Tyler, P. D., & Celi, L. A. (2018). Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PLOS ONE*, *13*(2), e0192360. https://doi.org/10.1371/journal.pone.0192360

Gentleman, R., & Carey, V. J. (2008). Unsupervised Machine Learning. In F. Hahne, W. Huber, R. Gentleman, & S. Falcon (Eds.), *Bioconductor Case Studies* (pp. 137–157). Springer. https://doi.org/10.1007/978-0-387-77240-0_10

Gill, P., & Dolan, G. (2015). Originality and the PhD: what is it and how can it be demonstrated? *Nurse Researcher*, *22*(6).

Goeuriot, L., Suominen, H., Kelly, L., Miranda-Escalada, A., Krallinger, M., Liu, Z., Pasi, G., Gonzalez Saez, G., Viviani, M., & Xu, C. (2020). Overview of the CLEF eHealth Evaluation Lab 2020. In A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, H. Joho, C. Lioma, C. Eickhoff, A. Névéol, L. Cappellato, & N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (pp. 255–271). Springer International Publishing. https://doi.org/10.1007/978-3-030-58219-7_19

Goldberg, Y. (2017). *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.

Google. (2021). *Google Guava* [Java]. Google. https://github.com/google/guava (Original work published 2014)

Grace, G. W. (2016). *The Linguistic Construction of Reality*. Routledge.

Gross, J. L., Yellen, J., & Zhang, P. (2013). *Handbook of Graph Theory*. CRC Press.

Grubbs, F. E. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics*, *11*(1), 1–21. https://doi.org/10.1080/00401706.1969.10490657

Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*, *36*(7), 10206–10222. https://doi.org/10.1016/j.eswa.2009.02.037

Halevy, A., Norvig, P., & Pereira, F. (2009). The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, *24*(2), 8–12. https://doi.org/10.1109/MIS.2009.36

Hanauer, D. A., Barnholtz-Sloan, J. S., Beno, M. F., Del Fiol, G., Durbin, E. B., Gologorskaya, O., Harris, D., Harnett, B., Kawamoto, K., May, B., Meeks, E., Pfaff, E., Weiss, J., & Zheng, K. (2020). Electronic Medical Record Search Engine (EMERSE): An Information Retrieval Tool for Supporting Cancer Research. *JCO Clinical Cancer Informatics*, *4*, 454–463. https://doi.org/10.1200/CCI.19.00134

Hashemi, H. (2012). Fuzzy Clustering of Seismic Sequences: Segmentation of Time-Frequency Representations. *IEEE Signal Processing Magazine*, *29*, 82–87. https://doi.org/10.1109/MSP.2012.2185897

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Overview of Supervised Learning. In T. Hastie, R. Tibshirani, & J. Friedman (Eds.), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (pp. 9–41). Springer. https://doi.org/10.1007/978-0-387-84858-7_2

Hevner, A., & Chatterjee, S. (2010). Design Science Research in Information Systems. In A. Hevner & S. Chatterjee (Eds.), *Design Research in Information Systems: Theory and Practice* (pp. 9–22). Springer US. https://doi.org/10.1007/978-1-4419-5653-8_2

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97. https://doi.org/10.1109/MSP.2012.2205597

Hinton, G. E. (2009). Deep belief networks. *Scholarpedia*, *4*(5), 5947. https://doi.org/10.4249/scholarpedia.5947

Hongke Xia, Xuefeng Zheng, & Xiang Hu. (2010). Graph-based partitioning of large-scale ontologies. *2010 2nd International Conference on Industrial and Information Systems*, *1*, 371–375. https://doi.org/10.1109/INDUSIS.2010.5565834

Hsu, W., Taira, R. K., El-Saden, S., Kangarloo, H., & Bui, A. A. T. (2012). Context-Based Electronic Health Record: Toward Patient Specific Healthcare. *IEEE Transactions on Information Technology in Biomedicine*, *16*(2), 228–234. https://doi.org/10.1109/TITB.2012.2186149

Huang, W., Kataria, S., Caragea, C., Mitra, P., Giles, C. L., & Rokach, L. (2012). Recommending citations: Translating papers into references. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 1910–1914. https://doi.org/10.1145/2396761.2398542

Huang, X., Xu, T., Gao, W., & Jia, Z. (2011). Ontology similarity measure and ontology mapping via fast ranking method. *International Journal of Applied Physics and Mathematics*, *1*(1), 54.

Hudry, J.-L. (2011). Aristotle on Meaning. *Archiv für Geschichte der Philosophie*, *93*(3), 253–280. https://doi.org/10.1515/agph.2011.012

IOM, McGinnis, M., Olsen, L., Goodby, A. W., Roundtable on Value & Science-Driven Health Care, & Institute of Medicine. (2010). *Clinical Data as the Basic Staple of Health*

Learning: Creating and Protecting a Public Good: Workshop Summary. National Academies Press.

Johannesson, P., & Perjons, E. (2014). *An Introduction to Design Science*. Springer.

Jonquet, C., Musen, M. A., & Shah, N. (2008). A System for Ontology-Based Annotation of Biomedical Data. In A. Bairoch, S. Cohen-Boulakia, & C. Froidevaux (Eds.), *Data Integration in the Life Sciences* (pp. 144–152). Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/978-3-540-69828-9_14

Kaurova, O., Alexandrov, M., & Blanco, X. (2011). Classification of free text clinical narratives (short review). *Business and Engineering Applications of Intelligent and Information Systems, 124*.

Kersloot, M. G., van Putten, F. J. P., Abu-Hanna, A., Cornet, R., & Arts, D. L. (2020). Natural language processing algorithms for mapping clinical text fragments onto ontology concepts: A systematic review and recommendations for future studies. *Journal of Biomedical Semantics*, *11*(1), 14. https://doi.org/10.1186/s13326-020-00231-z

Kiourtis, A., Mavrogiorgou, A., & Kyriazis, D. (2018). FHIR Ontology Mapper (FOM): Aggregating Structural and Semantic Similarities of Ontologies towards their Alignment to HL7 FHIR. *2018 IEEE 20th International Conference on E-Health Networking, Applications and Services (Healthcom)*, 1–7. https://doi.org/10.1109/HealthCom.2018.8531149

Klein, M. (2001). Combining and relating ontologies: An analysis of problems and solutions. *OIS@ IJCAI*.

Köhler, N. D., Büttner, M., & Theis, F. J. (2019). Deep learning does not outperform classical machine learning for cell-type annotation. *BioRxiv*, 653907. https://doi.org/10.1101/653907

Koutsoukas, A., Monaghan, K. J., Li, X., & Huan, J. (2017). Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of Cheminformatics*, *9*(1), 42. https://doi.org/10.1186/s13321-017-0226-y

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). *Proc. Advances in Neural Information Processing Systems 25*.

Le, N., Wiley, M., Loza, A., Hristidis, V., & El-Kareh, R. (2020). Prediction of Medical Concepts in Electronic Health Records: Similar Patient Analysis. *JMIR Medical Informatics*, *8*(7), e16008. https://doi.org/10.2196/16008

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, D., Cornet, R., Lau, F., & de Keizer, N. (2013). A survey of SNOMED CT implementations. *Journal of Biomedical Informatics*, *46*(1), 87–96. https://doi.org/10.1016/j.jbi.2012.09.006

Lee, W., Shah, N., Sundlass, K., & Musen, M. (2007). Comparison of ontology-based semantic-similarity measures. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*, 384–388.

Lee, W.-N., Shah, N., Sundlass, K., & Musen, M. (2008). Comparison of Ontology-based Semantic-Similarity Measures. *AMIA Annual Symposium Proceedings*, *2008*, 384–388.

Li, N., & Motta, E. (2010). Evaluations of User-Driven Ontology Summarization. In P. Cimiano & H. S. Pinto (Eds.), *Knowledge Engineering and Management by the Masses* (pp. 544–553). Springer Berlin Heidelberg.

Liakata, M., Teufel, S., Siddharthan, A., & Batchelor, C. (2010). *Corpora for the conceptualisation and zoning of scientific papers*. LREC 2010, 7th International Conference on Language Resources and Evaluation, Valletta, Malta.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.724.8203&rep=rep1&type=pdf

Ling, Y., Hasan, S. A., Datla, V., Qadir, A., Lee, K., Liu, J., & Farri, O. (2017). Diagnostic Inferencing via Improving Clinical Concept Extraction with Deep Reinforcement Learning: A Preliminary Study. *Machine Learning for Healthcare Conference*, 271–285. http://proceedings.mlr.press/v68/ling17a.html

Liu, G., Jia, Z., & Gao, W. (2018). *Ontology similarity computing based on stochastic primal dual coordinate technique*. https://doi.org/10.30538/OMS2018.0030

Liu, H., Hildebrand, P. L., Perl, Y., & Geller, J. (2018). Enrichment of SNOMED CT Ophthalmology Component to Support EHR Coding. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1990–1997. https://doi.org/10.1109/BIBM.2018.8621272

Liu, X., Faes, L., Kale, A. U., Wagner, S. K., Fu, D. J., Bruynseels, A., Mahendiran, T., Moraes, G., Shamdas, M., Kern, C., Ledsam, J. R., Schmid, M. K., Balaskas, K., Topol, E. J., Bachmann, L. M., Keane, P. A., & Denniston, A. K. (2019). A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: A systematic review and meta-analysis. *The Lancet Digital Health*, *1*(6), e271–e297. https://doi.org/10.1016/S2589-7500(19)30123-2

Lu, C. J., Payne, A., & Mork, J. G. (2020). The Unified Medical Language System SPECIALIST Lexicon and Lexical Tools: Development and applications. *Journal of the American Medical Informatics Association : JAMIA*, *27*(10), 1600–1605. https://doi.org/10.1093/jamia/ocaa056

Luo, Y.-F., Sun, W., & Rumshisky, A. (2019). MCN: A comprehensive corpus for medical concept normalization. *Journal of Biomedical Informatics*, *92*, 103132. https://doi.org/10.1016/j.jbi.2019.103132

Lussier, Y. A., Shagina, L., & Friedman, C. (2001). Automating SNOMED coding using medical language understanding: A feasibility study. *Proceedings of the AMIA Symposium*, 418–422.

Lv, X., Guan, Y., Yang, J., & Wu, J. (2016). Clinical relation extraction with deep learning. *International Journal of Hybrid Information Technology*, *9*(7), 237–248.

Maiga, G., & Ddembe, W. (2009). *A flexible biomedical ontology selection tool*. http://makir.mak.ac.ug/handle/10570/2021

Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, *57*(3), 519–530. https://doi.org/10.1093/biomet/57.3.519

Martínez-Romero, M., Jonquet, C., O'Connor, M. J., Graybeal, J., Pazos, A., & Musen, M. A. (2017). NCBO Ontology Recommender 2.0: An enhanced approach for biomedical ontology recommendation. *Journal of Biomedical Semantics*, *8*(1), 21. https://doi.org/10.1186/s13326-017-0128-y

Martínez-Romero, M., Vázquez-Naya, J. M., Pereira, J., & Pazos, A. (2014). BiOSS: A system for biomedical ontology selection. *Computer Methods and Programs in Biomedicine*, *114*(1), 125–140. https://doi.org/10.1016/j.cmpb.2014.01.020

McCray, A. T., Aronson, A. R., Browne, A. C., Rindflesch, T. C., Razi, A., & Srinivasan, S. (1993). UMLS knowledge for biomedical language processing. *Bulletin of the Medical Library Association*, *81*(2), 184–194.

McCray, A. T., Srinivasan, S., & Browne, A. C. (1994). Lexical methods for managing variation in biomedical terminologies. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 235–239.

McGray, A. T., Sponsler, J. L., Brylawski, B., & Browne, A. C. (1987). The Role of Lexical Knowledge in Biomedical Text Understanding. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 103–107.

Melton, G. B., & Hripcsak, G. (2005). Automated Detection of Adverse Events Using Natural Language Processing of Discharge Summaries. *Journal of the American Medical Informatics Association*, *12*(4), 448–457. https://doi.org/10.1197/jamia.M1794

Melton, G. B., Morrison, F. P., Cimino, J. J., Temple, L. K., Choti, M. A., Schulick, R. D., & Gearhart, S. L. (2006). How well do electronic systems represent colorectal cancer surgery concepts? Evaluation of SNOMED-CT, ICD9-CM, and CPT-4 for content coverage. *Journal of the American College of Surgeons*, *203*(3, Supplement), S69–S70. https://doi.org/10.1016/j.jamcollsurg.2006.05.182

Michail, D., Kinable, J., Naveh, B., & Sichi, J. V. (2020). JGraphT—A Java Library for Graph Data Structures and Algorithms. *ACM Transactions on Mathematical Software*, *46*(2), 1–29. https://doi.org/10.1145/3381449

Miñarro-Giménez, J. A., Martínez-Costa, C., Karlsson, D., Schulz, S., & Gøeg, K. R. (2018). Qualitative analysis of manual annotations of clinical text with SNOMED CT. *PLOS ONE*, *13*(12), e0209547. https://doi.org/10.1371/journal.pone.0209547

Mirończuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, *106*, 36–54. https://doi.org/10.1016/j.eswa.2018.03.058

MIT Press. (2020). *WordNet | The MIT Press*. The MIT Press. https://mitpress.mit.edu/books/wordnet

Mitchell, T. M. (2006). *The discipline of machine learning* (Vol. 9). Carnegie Mellon University, School of Computer Science, Machine Learning ….

Mitkov, R. (2004). *The Oxford Handbook of Computational Linguistics*. OUP Oxford.

Mujib, M. I., Yang, C. C., Zhao, M., & Williams, J. R. (2018). Expanding Consumer Health Vocabularies with Frequency-Conserving Internal Context Models. *2018 IEEE*

International Conference on Healthcare Informatics (ICHI), 241–246. https://doi.org/10.1109/ICHI.2018.00034

Nagao, K. (2003a). *Digital Content Annotation and Transcoding*. Artech House.

Nagao, K. (2003b). *Digital Content Annotation and Transcoding*. Artech House.

Nguyen, H., & Patrick, J. (2016). Text Mining in Clinical Domain: Dealing with Noise. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 549–558. https://doi.org/10.1145/2939672.2939720

NLM.govt. (2019a). *UMLS Metathesaurus—LNC (LOINC)—Statistics*. https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/LNC/stats.html

NLM.govt. (2019b). *UMLS Metathesaurus—NCI (NCI Thesaurus)—Statistics*. https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/NCI/stats.html

NLTK. (2020). *Python NLTK Stemming and Lemmatization Demo*. Stemming and Lemmatization with Python NLTK. https://text-processing.com/demo/stem/

NSW Clinical Excellence Commission. (2020). *Clinical incident data*. Clinical Incident Data. https://www.cec.health.nsw.gov.au/Review-incidents/Biannual-Incident-Report/Clinical-incident-data

Ochieng, P., & Kyanda, S. (2018). A K-way spectral partitioning of an ontology for ontology matching. *Distributed and Parallel Databases*, *36*(4), 643–673. https://doi.org/10.1007/s10619-018-7222-8

Ogallo, W., & Kanter, A. (2016). Using Natural Language Processing and Network Analysis to Develop a Conceptual Framework for Medication Therapy Management Research. *AMIA ... Annual Symposium Proceedings. AMIA Symposium*, *2016*, 984–993.

OneLogin. (2021). *XML Pretty Print Online Tool | SAMLTool.com*. OneLogin - SAML Developer Tools. https://www.samltool.com/prettyprint.php

openEHR. (2015). *Welcome to openEHR*. http://www.openehr.org/

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999, November 11). *The PageRank Citation Ranking: Bringing Order to the Web.* [Techreport]. Stanford InfoLab. http://ilpubs.stanford.edu:8090/422/

Pakhomov, S., Buntrock, J., & Duffy, P. (2005). High throughput modularized NLP system for clinical text. *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 25–28.

Papatheodorou, I., Crichton, C., Morris, L., Maccallum, P., $author.lastName, $author firstName, Davies, J., Brenton, J. D., & Caldas, C. (2009). A metadata approach for clinical data management in translational genomics studies in breast cancer. *BMC Medical Genomics*, *2*(1), 66. https://doi.org/10.1186/1755-8794-2-66

Pape-Haugaard, L. B., Lovis, C., & Madsen, I. C. (2020). *Digital Personalized Health and Medicine: Proceedings of MIE 2020*. IOS Press.

Passant, A. (2010). *Semantic Web Technologies for Enterprise 2.0*. IOS Press.

Percha, B. (2020). *Modern Clinical Text Mining: A Guide and Review*. https://doi.org/10.20944/preprints202010.0649.v1

Phillips, E., & Pugh, D. (2010). *How To Get A Phd: A handbook for students and their supervisors*. McGraw-Hill Education (UK).

Pinto, H. S., Gómez-Pérez, A., & Martins, J. P. (1999). Some issues on ontology integration. *Proceedings of the IJCAI*, *99*, 7–1.

Plisson, J., Lavrac, N., Mladenic, D., & others. (2004). A rule based approach to word lemmatization. *Proceedings of IS*, *3*, 83–86.

Porter, M. f. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137. https://doi.org/10.1108/eb046814

Potter, J. (1998). Fragments in the realization of relativism. *Social Constructionism, Discourse and Realism*, 27–45.

Powell, R. T., Olar, A., Narang, S., Rao, G., Sulman, E., Fuller, G. N., & Rao, A. (2017). Identification of Histological Correlates of Overall Survival in Lower Grade Gliomas Using a Bag-of-words Paradigm: A Preliminary Analysis Based on Hematoxylin & Eosin Stained Slides from the Lower Grade Glioma Cohort of The Cancer Genome Atlas. *Journal of Pathology Informatics*, *8*. https://doi.org/10.4103/jpi.jpi_43_16

Pradhan, S., Elhadad, N., South, B. R., Martinez, D., Christensen, L., Vogel, A., Suominen, H., Chapman, W. W., & Savova, G. (2015). Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. *Journal of the American Medical Informatics Association*, *22*(1), 143–154. https://doi.org/10.1136/amiajnl-2013-002544

Pührer, J., Heymans, S., & Eiter, T. (2010). Dealing with Inconsistency When Combining Ontologies and Rules Using DL-Programs. In L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, & T. Tudorache (Eds.), *The Semantic Web: Research and Applications* (pp. 183–197). Springer. https://doi.org/10.1007/978-3-642-13486-9_13

Pustejovsky, J., & Stubbs, A. (2012a). *Natural Language Annotation for Machine Learning*. O'Reilly Media, Inc.

Pustejovsky, J., & Stubbs, A. (2012b). *Natural Language Annotation for Machine Learning: A Guide to Corpus-Building for Applications*. O'Reilly Media, Inc.

Rabin, S. (2019). *Game AI Pro 360: Guide to Movement and Pathfinding*. CRC Press.

Raje, S., & Bodenreider, O. (2017). Interoperability of Disease Concepts in Clinical and Research Ontologies: Contrasting Coverage and Structure in the Disease Ontology and SNOMED CT. *Studies in Health Technology and Informatics*, *245*, 925–929.

Rastegar-Mojarad, M., Sohn, S., Wang, L., Shen, F., Bleeker, T. C., Cliby, W. A., & Liu, H. (2017). Need of informatics in designing interoperable clinical registries. *International*

Journal of Medical Informatics, *108*, 78–84. https://doi.org/10.1016/j.ijmedinf.2017.10.004

REANNZ. (2021). *Tuakiri—Trust and identity: REANNZ*. https://www.reannz.co.nz/products-and-services/tuakiri/

Reátegui, R., & Ratté, S. (2018). Comparison of MetaMap and cTAKES for entity extraction in clinical notes. *BMC Medical Informatics and Decision Making*, *18*(3), 74. https://doi.org/10.1186/s12911-018-0654-2

Rector, A. L., Brandt, S., & Schneider, T. (2011). Getting the foot out of the pelvis: Modeling problems affecting use of SNOMED CT hierarchies in practical applications. *Journal of the American Medical Informatics Association: JAMIA*, *18*(4), 432–440. https://doi.org/10.1136/amiajnl-2010-000045

Reichenbächer, M., & Einax, J. W. (2011). *Challenges in Analytical Quality Assurance*. Springer Science & Business Media.

Roberts, A., Gaizauskas, R., Hepple, M., Davis, N., Demetriou, G., Guo, Y., Kola, J. (Subbarao), Roberts, I., Setzer, A., Tapuria, A., & Wheeldin, B. (2007). The CLEF Corpus: Semantic Annotation of Clinical Text. *AMIA Annual Symposium Proceedings*, *2007*, 625–629.

Rockland, K. S., Peters, A., & Kaas, J. H. (1998). *Cerebral Cortex: Volume 12: Extrastriate Cortex in Primates*. Springer Science & Business Media.

Rodrigues, J., Schulz, S., Mizen, B., Rector, A., & Serir, S. (2018). Is the Application of SNOMED CT Concept Model sufficiently Quality Assured? *AMIA Annual Symposium Proceedings*, *2017*, 1488–1497.

Rodríguez-González, A., Costumero, R., Martinez-Romero, M., Wilkinson, M. D., & Menasalvas-Ruiz, E. (2018). Extracting Diagnostic Knowledge from MedLine Plus: A

Comparison between MetaMap and cTAKES Approaches. *Current Bioinformatics*, *13*(6), 573–582. https://doi.org/10.2174/1574893612666170727094502

Roth, A. D. (2008). *Reciprocal Influences Between Rhetoric and Medicine in Ancient Greece*. ProQuest.

Rygl, J., Sojka, P., Ruzicka, M., & Rehurek, R. (2016). ScaleText: The Design of a Scalable, Adaptable and User-Friendly Document System for Similarity Searches. *RASLAN*, 79–87.

Sachs, L. (2013). *Angewandte Statistik: Anwendung statistischer Methoden*. Springer-Verlag.

Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., & Chute, C. G. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, *17*(5), 507–513. https://doi.org/10.1136/jamia.2009.001560

Schlicht, A., & Stuckenschmidt, H. (2007). Criteria-based partitioning of large ontologies. *Proceedings of the 4th International Conference on Knowledge Capture*, 171–172.

Schloeffel, P., Beale, T., Hayworth, G., Heard, S., & Leslie, H. (2006). *The Relationship between CEN 13606, HL7, and OpenEHR*. http://search.informit.com.au.ezproxy.auckland.ac.nz/documentSummary;dn=950616334398351;res=IELHEA

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Schriml, L. M., Arze, C., Nadendla, S., Chang, Y.-W. W., Mazaitis, M., Felix, V., Feng, G., & Kibbe, W. A. (2012). Disease Ontology: A backbone for disease semantic integration. *Nucleic Acids Research*, *40*(D1), D940–D946. https://doi.org/10.1093/nar/gkr972

Searle, J. R., & Slusser, M. (1995). *The Construction of Social Reality*. Simon and Schuster.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47. https://doi.org/10.1145/505282.505283

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, *13*(11), 2498–2504. https://doi.org/10.1101/gr.1239303

Sheikhalishahi, S., Miotto, R., Dudley, J. T., Lavelli, A., Rinaldi, F., & Osmani, V. (2019). Natural Language Processing of Clinical Notes on Chronic Diseases: Systematic Review. *JMIR Medical Informatics*, *7*(2), e12239. https://doi.org/10.2196/12239

Shi, Y., Brown, A. G., Lawford, P. V., Arndt, A., Nuesser, P., & Hose, D. R. (2011). Computational modelling and evaluation of cardiovascular response under pulsatile impeller pump support. *Interface Focus*, *1*(3), 320–337. https://doi.org/10.1098/rsfs.2010.0039

Shore, B. (1998). *Culture in Mind: Cognition, Culture, and the Problem of Meaning*. Oxford University Press.

Sittig, D. F., Hazlehurst, B. L., Brown, J., Murphy, S., Rosenman, M., Tarczy-Hornoch, P., & Wilcox, A. B. (2012). A survey of informatics platforms that enable distributed comparative effectiveness research using multi-institutional heterogeneous clinical data. *Medical Care*, *50*(Suppl), S49–S59. https://doi.org/10.1097/MLR.0b013e318259c02b

Smaili, F. Z., Gao, X., & Hoehndorf, R. (2019). OPA2Vec: Combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinformatics*, *35*(12), 2133–2140. https://doi.org/10.1093/bioinformatics/bty933

Smith, B. (2004). *Beyond concepts: Ontology as reality representation*. https://philarchive.org

SNOMED International. (2019). *Compositional Grammar—Specification and Guide*. https://confluence.ihtsdotools.org/display/DOCSCG

SNOMED International. (2020a). *3.1.1 General Structure of Release Files—Release File Specification*. Release File Specification. https://confluence.ihtsdotools.org/display/DOCRELFMT/3.1.1+General+Structure+of+Release+Files

SNOMED International. (2020b). *4.2 File Format Specifications—Release File Specification*. File Format Specifications. https://confluence.ihtsdotools.org/display/DOCRELFMT/4.2+File+Format+Specifications

SNOMED International. (2020c). *SNOMED CT Starter Guide*. SNOMED CT Concept Model. https://confluence.ihtsdotools.org/display/DOCSTART/6.+SNOMED+CT+Concept+Model

Solt, I., Tikk, D., Gál, V., & Kardkovács, Z. T. (2009). Semantic Classification of Diseases in Discharge Summaries Using a Context-aware Rule-based Classifier. *Journal of the American Medical Informatics Association*, *16*(4), 580–584. https://doi.org/10.1197/jamia.M3087

Sporny, M., Longley, D., Kellogg, G., Lanthaler, M., & Lindström, N. (2014). JSON-LD 1.0. *W3C Recommendation*, *16*, 41.

Stanfill, M. H., Williams, M., Fenton, S. H., Jenders, R. A., & Hersh, W. R. (2010). A systematic literature review of automated clinical coding and classification systems. *Journal of the American Medical Informatics Association*, *17*(6), 646–651. https://doi.org/10.1136/jamia.2009.001024

Stokes, N., Li, Y., Cavedon, L., & Zobel, J. (2009). Exploring criteria for successful query expansion in the genomic domain. *Information Retrieval*, *12*(1), 17–50. https://doi.org/10.1007/s10791-008-9073-9

Sun, W., Rumshisky, A., & Uzuner, O. (2013). Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. *Journal of the American Medical Informatics Association*, *20*(5), 806–813. https://doi.org/10.1136/amiajnl-2013-001628

Sung, S.-F., Hsieh, C.-Y., & Hu, Y.-H. (2020). Two Decades of Research Using Taiwan's National Health Insurance Claims Data: Bibliometric and Text Mining Analysis on PubMed. *Journal of Medical Internet Research*, *22*(6), e18457. https://doi.org/10.2196/18457

Sutskever, I., Hinton, G. E., & Taylor, G. W. (2008). The Recurrent Temporal Restricted Boltzmann Machine. *Advances in Neural Information Processing Systems*, *21*, 1601–1608.

Sutskever, I., Vinyals, O., & Le, Q. (2014). *Proc. Advances in Neural Information Processing Systems 27*.

Tang, B., Wu, Y., Jiang, M., Denny, J. C., & Xu, H. (2013). Recognizing and Encoding Discorder Concepts in Clinical Text using Machine Learning and Vector Space Model. *CLEF (Working Notes)*, *665*.

Tchechmedjiev, A., Abdaoui, A., Emonet, V., Melzi, S., Jonnagaddala, J., & Jonquet, C. (2018). Enhanced functionalities for annotating and indexing clinical text with the NCBO Annotator+. *Bioinformatics*, *34*(11), 1962–1965. https://doi.org/10.1093/bioinformatics/bty009

Tongphu, S., & Suntisrivaraporn, B. (2017). Algorithms for Measuring Similarity Between ELH Concept Descriptions: A Case Study on Snomed ct. *COMPUTING AND INFORMATICS*, *36*(4), 733–764.

Torii, M., Wagholikar, K., & Liu, H. (2011). Using machine learning for concept extraction on clinical documents from multiple data sources. *Journal of the American Medical Informatics Association*, *18*(5), 580–587. https://doi.org/10.1136/amiajnl-2011-000155

Turchin, A., Kolatkar, N. S., Grant, R. W., Makhni, E. C., Pendergrass, M. L., & Einbinder, J. S. (2006). Using Regular Expressions to Abstract Blood Pressure and Treatment Intensification Information from the Text of Physician Notes. *Journal of the American Medical Informatics Association*, *13*(6), 691–695. https://doi.org/10.1197/jamia.M2078

Uzuner, Ö., Luo, Y., & Szolovits, P. (2007). Evaluating the State-of-the-Art in Automatic De-identification. *Journal of the American Medical Informatics Association*, *14*(5), 550–563. https://doi.org/10.1197/jamia.M2444

Uzuner, Ö., Solti, I., & Cadag, E. (2010). Extracting medication information from clinical text. *Journal of the American Medical Informatics Association*, *17*(5), 514–518. https://doi.org/10.1136/jamia.2010.003947

Uzuner, Ö., South, B. R., Shen, S., & DuVall, S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association : JAMIA*, *18*(5), 552–556. https://doi.org/10.1136/amiajnl-2011-000203

Velupillai, S., Mowery, D., South, B. R., Kvist, M., & Dalianis, H. (2015). Recent Advances in Clinical Natural Language Processing in Support of Semantic Analysis. *Yearbook of Medical Informatics*, *10*(1), 183–193. https://doi.org/10.15265/IY-2015-009

Vezina, B. (2007). Universals and particulars: Aristotle's ontological theory and criticism of the Platonic forms. *Undergraduate Review*, *3*(1), 101–103.

Viani, N., Miller, T. A., Napolitano, C., Priori, S. G., Savova, G. K., Bellazzi, R., & Sacchi, L. (2019). Supervised methods to extract clinical events from cardiology reports in Italian.

Journal of Biomedical Informatics, *95*, 103219. https://doi.org/10.1016/j.jbi.2019.103219

Wang, X., Hripcsak, G., Markatou, M., & Friedman, C. (2009). Active Computerized Pharmacovigilance Using Natural Language Processing, Statistics, and Electronic Health Records: A Feasibility Study. *Journal of the American Medical Informatics Association*, *16*(3), 328–337. https://doi.org/10.1197/jamia.M3028

Wang, Y., Gao, W., Zhang, Y., & Gao, Y. (2010). Ontology similarity computation use ranking learning method. *The 3rd International Conference on Computational Intelligence and Industrial Application*, 20–22.

Wang, Y., Patrick, J., Miller, G., & O'Hallaran, J. (2008). A computational linguistics motivated mapping of ICPC-2 PLUS to SNOMED CT. *BMC Medical Informatics and Decision Making*, *8*(1), S5. https://doi.org/10.1186/1472-6947-8-S1-S5

Webster, J. J., & Kit, C. (1992). Tokenization as the initial phase in NLP. *COLING 1992 Volume 4: The 15th International Conference on Computational Linguistics*.

Wei, D., & Fu, G. (2017). Using SNOMED Distance to Measure Semantic Similarity of Clinical Trials. *Studies in Health Technology and Informatics*, *245*, 1341–1341.

Wei, Q., Ji, Z., Li, Z., Du, J., Wang, J., Xu, J., Xiang, Y., Tiryaki, F., Wu, S., Zhang, Y., Tao, C., & Xu, H. (2020). A study of deep learning approaches for medication and adverse drug event extraction from clinical text. *Journal of the American Medical Informatics Association*, *27*(1), 13–21. https://doi.org/10.1093/jamia/ocz063

Wilbur, W. J., Rzhetsky, A., & Shatkay, H. (2006). New directions in biomedical text annotation: Definitions, guidelines and corpus construction. *BMC Bioinformatics*, *7*(1), 356. https://doi.org/10.1186/1471-2105-7-356

Wilcock, G. (2009a). *Introduction to Linguistic Annotation and Text Analytics*. Morgan & Claypool Publishers.

Wilcock, G. (2009b). Introduction to Linguistic Annotation and Text Analytics. *Synthesis Lectures on Human Language Technologies*, *2*(1), 1–159. https://doi.org/10.2200/S00194ED1V01Y200905HLT003

Wu, Y., Jiang, M., Lei, J., & Xu, H. (2015). Named Entity Recognition in Chinese Clinical Text Using Deep Neural Network. *Studies in Health Technology and Informatics*, *216*, 624–628.

Wu, Y., Jiang, M., Xu, J., Zhi, D., & Xu, H. (2018). Clinical Named Entity Recognition Using Deep Learning Models. *AMIA Annual Symposium Proceedings*, *2017*, 1812–1819.

Xia, Y., Zhong, X., Liu, P., Tan, C., Na, S., Hu, Q., & Huang, Y. (2013). Combining MetaMap and cTAKES in Disorder Recognition: THCIB at CLEF eHealth Lab 2013 Task 1. *CLEF (Working Notes)*.

Xu, D., Gopale, M., Zhang, J., Brown, K., Begoli, E., & Bethard, S. (2020). Unified Medical Language System resources improve sieve-based generation and Bidirectional Encoder Representations from Transformers (BERT)–based ranking for concept normalization. *Journal of the American Medical Informatics Association*, *27*(10), 1510–1519. https://doi.org/10.1093/jamia/ocaa080

Xu, J., & Croft, W. B. (1998). Corpus-based Stemming Using Cooccurrence of Word Variants. *ACM Trans. Inf. Syst.*, *16*(1), 61–81. https://doi.org/10.1145/267954.267957

Zaiontz, C. (2020). *Real Statistics Resource Pack software* (7.2) [Computer software]. www.real-statistics.com

Zasada, S. J., Wang, T., Haidar, A., Liu, E., Graf, N., Clapworthy, G., Manos, S., & Coveney, P. V. (2012). IMENSE: An e-infrastructure environment for patient specific multiscale data integration, modelling and clinical treatment. *Journal of Computational Science*, *3*(5), 314–327. https://doi.org/10.1016/j.jocs.2011.07.001

Zech, J., Pain, M., Titano, J., Badgeley, M., Schefflein, J., Su, A., Costa, A., Bederson, J., Lehar, J., & Oermann, E. K. (2018). Natural Language–based Machine Learning Models for the Annotation of Clinical Radiology Reports. *Radiology*, *287*(2), 570–580. https://doi.org/10.1148/radiol.2018171093

Zhu, H., Paschalidis, I. C., & Tahmasebi, A. (2018). Clinical Concept Extraction with Contextual Word Embedding. *ArXiv:1810.10566 [Cs]*. http://arxiv.org/abs/1810.10566

Zivaljevic, A., Atalag, K., & Warren, J. (2019, November 20). *Automated annotation of clinical free text using SNOMED annotated graphs*. Health Informatics New Zealand 2019, Hamilton. https://www.hinz.org.nz/page/ProgDHWNZ19

Zivaljevic, A., Atalag, K., & Warren, J. (2020). Utility of SNOMED CT in automated expansion of clinical terms in discharge summaries: Testing issues of coverage. *Health Information Management Journal*, 1833358320934528. https://doi.org/10.1177/1833358320934528

Zivaljevic, A., Atalag, K., Warren, J., Cooling, M., Nickerson, D., & Hunter, P. (2015a). *Annotation of Clinical Datasets Using openEHR Archetypes*. Auckland Bioengineering Institute Research Forum, Auckland, New Zealand. http://www.abi.auckland.ac.nz/en/about/events/2016/2016-research-forum.html

Zivaljevic, A., Atalag, K., Warren, J., Cooling, M., Nickerson, D., & Hunter, P. (2015b). Annotation of clinical datasets using openEHR Archetypes as a solution for data access issues faced in biomedical projects. *Health Informatics New Zealand 2015*. https://www.researchgate.net/profile/Aleksandar_Zivaljevic/publication/282278618_Annotation_of_clinical_datasets_using_openEHR_Archetypes_as_a_solution_for_data_access_issues_faced_in_biomedical_projects/links/560a475b08ae576ce63fbbfd.pdf

Zivaljevic, A., Atalag, K., Warren, J., Cooling, M., Nickerson, D., & Hunter, P. (2015c, November 26). *Annotation of clinical datasets using openEHR Archetypes—Information discovery*. MedTech Core D4 Conference, Dunedin. https://www.regonline.com.au/custImages/380000/381616/PROGRAMME-ABSTRACTSFINAL.pdf

Zivaljevic, A., Atalag, K., Warren, J., Cooling, M., Nickerson, D., & Hunter, P. (2016). *Extracting meaning from openEHR clinical information models*. Auckland Bioengineering Institute Research Forum, Auckland, New Zealand. http://www.abi.auckland.ac.nz/en/about/events/2016/2016-research-forum.html