Clinical Practice Guideline Formalization: Translating Clinical Practice Guidelines to Computer Interpretable Guidelines

by

Wessam Gad El-Rab

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Clinical Practice Guidelines (CPGs) offer concise instruction on the optimal care for the patient based on the latest clinical findings. The main benefit of a CPG is to improve the quality of care, and the consistency of care. It is been shown that passive dissemination of CPGs, like publishing in a medical journal, is ineffective in changing practice behavior. Nevertheless, integrating CPG knowledge into clinical systems, such as decision support systems, has shown to be more effective.

In order to best benefit from the knowledge in the CPGs, an interest in automatically formalizing medical knowledge contained in CPGs has grown. This dissertation describes a new framework to automate a subset of the common CPGs formalization research problems. Our framework follows a multi-step approach, which has been shown to be a good strategy for CPG formalization. One of the major sub-problem to automate the formalization of CPGs is to detect ambiguity in CPGs and resolve it automatically. In this dissertation we described two unsupervised algorithms to the resolve ambiguities in CPGs.

Preface

Chapter 6 of this thesis has been accepted by the Journal of Health Informatics (JHI) but not yet published. The paper will be publsihed as Wessam Gad El-Rab, Osmar Zaiane and Mohammad El-Hajj, "Formalizing Clinical Practice Guideline for Clinical Decision Support Systems", Journal of Health Informatics (JHI). I was responsible for the data collection and analysis as well as the manuscript composition. Osmar Zaiane and Mohammad El-Hajj were the supervisory authors and were involved with concept formation and contributed to manuscript edits.

Acknowledgments

I am indebted to many people who have been an integral part of my research. I would like to sincerely thank my supervisors, Dr. Osmar Zaïane and Dr. Mohammad El-Hajj for their guidance, support throughout this work. I've learned a lot from you and consider myself very fortunate to have you as my supervisors. I also would like to thank Dr. Greg Kondrak and Dr. Randy Goebel for their discussions, and feedback.

Thank you to my loving parents, who inculcated in me a love of learning and the value of hard work. You have ensured that I received the best education; Without you, I would never be where I stand today.

Finally, I would like to thank my wife Rola. You have supported and stood by me throughout this journey and I am grateful for your unconditional patience and understanding. Rola, I love you. I dedicate this thesis to you.

Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
CDSS	Clinical Decision Support System
CIG	Computer Interpretable Guideline
CPG	Clinical Practice Guideline
GDL	Guideline Definition Language
IE	Information Extraction
NGC	National Guideline Clearinghouse
NLM	National Library of Medicine
NLP	Natural Language Processing
UIMA	Unstructured Information Management Architecture
UMLS	Unified Medical Language System
WSD	Word Sense Disambiguation

Chapter 1

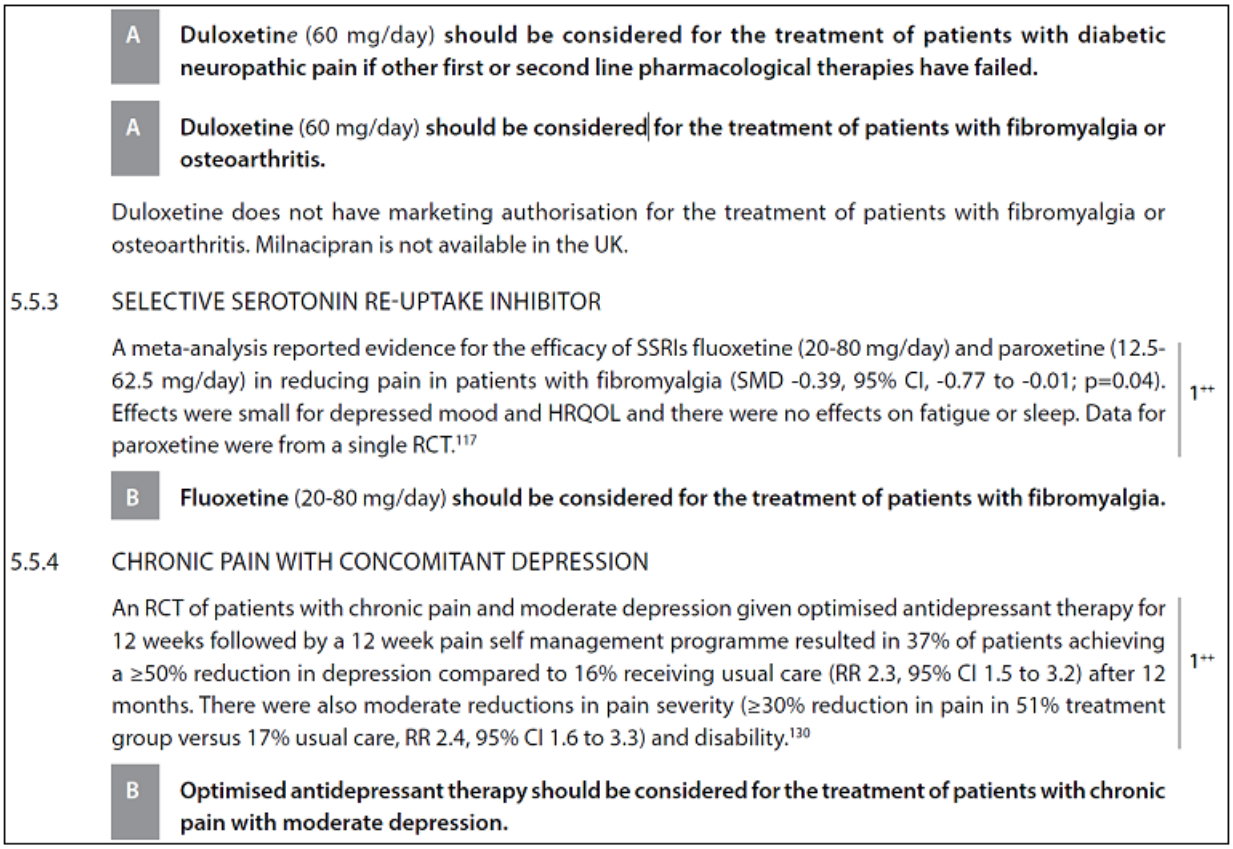
Introduction

1.1 Context and motivation

Clinical Practice Guidelines (CPGs) offer concise instruction on the optimal care for the patient based on the latest clinical findings. The main benefit of a CPG is to improve the quality of care, and the consistency of care. For a health care professional, a CPG can help offer explicit recommendations when a health care professional is uncertain about how to proceed, and can alert a health care professional when an ineffective practice is pursued [145].

It is been shown that passive dissemination of CPGs, like publishing in a medical journal, is ineffective in changing practice behavior [42]. Many health care practitioners are not aware of the existence of the CPG, and even when they are directed to the relevant CPG, they experience difficulties using it in their daily practice [77]. Nevertheless, integrating CPG knowledge into clinical systems, such as decision support systems, has shown to be more effective [144]. In order to best benefit from the CPGs knowledge by following an active CPG dissemination approach, an interest in formalizing medical knowledge contained in CPGs has grown.

CPGs are authored by different organizations such as the Scottish Intercollegiate Guidelines Network (SIGN), and the Guidelines Advisory Committee (GAC) in Canada. Each organization has it is own standard in developing and reviewing guidelines and therefore CPGs formats and structures vary across multiple organizations, which further



A **Duloxetine** (60 mg/day) **should be considered for the treatment of patients with diabetic neuropathic pain if other first or second line pharmacological therapies have failed.**

A **Duloxetine** (60 mg/day) **should be considered for the treatment of patients with fibromyalgia or osteoarthritis.**

Duloxetine does not have marketing authorisation for the treatment of patients with fibromyalgia or osteoarthritis. Milnacipran is not available in the UK.

5.5.3 SELECTIVE SEROTONIN RE-UPTAKE INHIBITOR

A meta-analysis reported evidence for the efficacy of SSRIs fluoxetine (20-80 mg/day) and paroxetine (12.5-62.5 mg/day) in reducing pain in patients with fibromyalgia (SMD -0.39, 95% CI, -0.77 to -0.01; p=0.04). Effects were small for depressed mood and HRQOL and there were no effects on fatigue or sleep. Data for paroxetine were from a single RCT.[117] 1++

B **Fluoxetine** (20-80 mg/day) **should be considered for the treatment of patients with fibromyalgia.**

5.5.4 CHRONIC PAIN WITH CONCOMITANT DEPRESSION

An RCT of patients with chronic pain and moderate depression given optimised antidepressant therapy for 12 weeks followed by a 12 week pain self management programme resulted in 37% of patients achieving a ≥50% reduction in depression compared to 16% receiving usual care (RR 2.3, 95% CI 1.5 to 3.2) after 12 months. There were also moderate reductions in pain severity (≥30% reduction in pain in 51% treatment group versus 17% usual care, RR 2.4, 95% CI 1.6 to 3.3) and disability.[130] 1++

B **Optimised antidepressant therapy should be considered for the treatment of patients with chronic pain with moderate depression.**

Figure 1.1: Extract from the Management of chronic pain CPG [3]

complicates the automation of the CPG formalization. In this research work we used the "Management of chronic pain. A national clinical guideline." CPG [3], which is a 71 pages PDF and is publicly available on the Scottish Intercollegiate Guidelines Network. Figure 1.1 shows an extract from page 19 of the "Management of chronic pain. A national clinical guideline." CPG [3].

There are several formal languages developed to help modelling clinical guidelines into a Computer Interpretable Guideline (CIG); a review study presents many of the popular formal languages[105]. Assuredly, the development of the guideline modelling languages is an important step toward facilitating the CPG formalization process, yet the formalization task remains laborious and complex, mainly because it requires two different areas of expertise: a medical expertise to correctly interpret the medical knowledge of CPGs, and a knowledge engineer expertise to correctly represent the medical knowledge using the

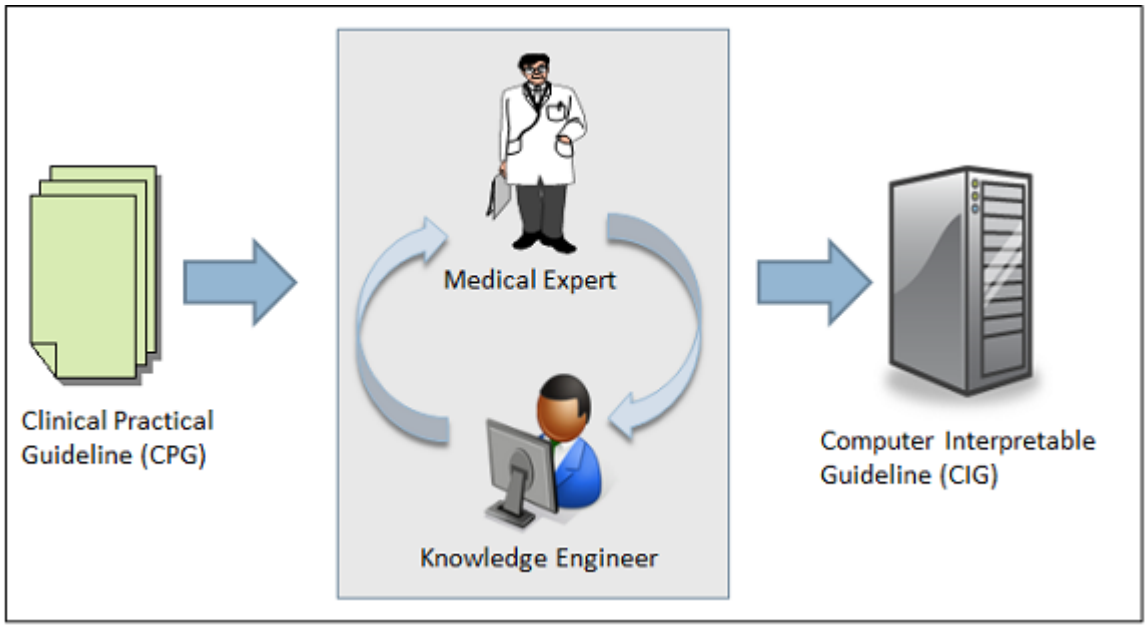


Figure 1.2: Manual CPG formalization

syntax of the modelling language. Figure 1.2 shows a typical manual CPG formalization process, where medical experts and knowledge engineers collaborate to map CPGs to CIGs.

The aim of this research is to minimize the effort required by human modellers to integrate the knowledge contained within CPGs into clinical systems such as clinical decision support systems (CDSS) by automating the manual CPG formalization process, a process that translates CPGs to CIGs. Full automation of the CPG formalization is not possible due to the computational complexity of natural language understanding; therefore, we narrowed our focus to automate the formalization of parts of the CPG text that conform to predefined lexical patterns that are used by medical experts to express specific clinical recommendations such as prescribing a medication or ordering a lab test for a patient.

To achieve our research target we set three specific goals. The first goal was automatically disambiguate the narrative text of CPGs using medical knowledge bases and graph-based algorithms. To achieve this goal, we created two unsupervised disambiguation algorithms that are further detailed in Chapter 4. The second goal was to develop

a system upon the algorithms resulted from our first goal to transform CPGs into CIGs using a multi-step approach. In Chapter 6, we presented a detailed description of all the components we developed to build CPG formalization system for one clinical recommendation type. The system proposed in Chapter 6 can be extended to find contradictions that exist between different CPGs by comparing the extracted rules in multiple CIGs and highlighting the rules that have the same conditions but with opposite or different actions. The third goal was to allow human modellers to refine and add other types of clinical recommendations without rebuilding the system. In Chapter 6, we presented a detailed 4 steps designed for human modellers to add and refine clinical recommendations without rebuilding the system.

1.2 Contribution to knowledge

This thesis makes contributions to knowledge in the following areas:

1. A Graph-based Disambiguation approach using the Unified Medical Language System (UMLS) semantic network [67](Section 4.3). This work is published in the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining [46].

2. A Graph-based Disambiguation approach using the UMLS Metathesaurus (Section 4.4). This work is published in the IEEE 15th International Conference on e-Health Networking, Applications Services (Healthcom) [47].

3. An analysis on the impact of using different UMLS subsets as a knowledge source on the unsupervised type of Word Sense Disambiguation (WSD) algorithms (Section 4.5). This work is published in the 3rd International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare [45].

4. A formalization system that can be effectively used to formalize the medication prescriptions of CPGs into CDSS friendly format. The system provides human modellers a process to extend the system to formalize other clinical recommendation

of CPGs (Chapter 6). This work is submitted to the Journal of Health Informatics (JHI) and is been recommended for publication with changes.

5. A scalability approach for our presented CPG formalization system to adapt new clinical recommendation types. The approach is based on using transitive verbs associated to clinical Action Palletes and evaluated the system against the Yale Guideline Recommendation Corpus (YGRC) (Chapter 7).

1.3 Structure of manuscript

Chapter 2 gives an overview of the CPG formalization processes, and the different tasks involved. The alternative approaches for knowledge extraction are discussed and the rational of choosing between them is highlighted. Chapter 3 reports on a systematic review of the literature on the different disambiguation approaches and knowledge source used. Chapter 4 presents two unsupervised disambiguation algorithms for biomedical text using the UMLS knowledge sources. The first algorithm uses the UMLS Semantic Network and the second algorithm uses the UMLS Metathesaurus. Chapter 4 also present the impact of using different subsets of the UMLS Metathesaurus on the accuracy of an unsupervised disambiguation algorithm. Chapter 5 presents the tools and technologies leveraged to build our CPG formalization system. Chapter 6 presents a CPG formalization system that follows a multi-step approach. The system is designed to set boundaries around each of the aspects of the CPG formalization, where each aspect is implemented as a separate autonomous component in a CPG formalization pipeline to serve the objective discussed in Chapter 2. Chapter 7 discusses the scalability of the CPG formalization system presented in Chapter 6.

Finally, Chapter 8 discusses the results of the research in the wider context of extracting clinical actions from CPG, considers directions for future research, and draws conclusions. The Appendices to this thesis contain additional material for Chapters 2-7.

Chapter 2

CPG Formalization - Review

2.1 Definitions

2.1.1 Clinical Practice Guidelines

Clinical Practice Guidelines (CPGs) as defined by the Institute of Medicine are "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" [52]. Such statements contain recommendations for best practice that aim to reduce variation in medical care by promoting the most effective treatments.

The gap between evidence and practice is one of the most consistent findings in research of health services [21]. CPGs compose an important source of evidence that rarely get adopted in practice [59]. The most notable reason for poor CPG adoption, is the traditional dissemination of CPG on paper alone which has proven to be generally insufficient [117, 60, 61, 147]. There are other reasons that contributed to the low adoption of CPG, such as:

1. Variation in the level of details in CPGs; Some CPGs are too general or too specific, making them hard to adopt in practice. In the "Calcium supplementation in pregnant women" CPG [1] the hypertension term is used but not detailed, while in the "First-trimester abortion in women with medical conditions" CPG [2] hypertension is defined as uncontrolled blood pressure (BP) (systolic BP >160 or diastolic BP

Dosage	1.5–2.0 g elemental calcium/day[a]
Frequency	Daily, with the total daily dosage divided into three doses (preferably taken at mealtimes)
Duration	From 20 weeks' gestation until the end of pregnancy
Target group	All pregnant women, particularly those at higher risk of gestational hypertension[b]
Settings	Areas with low calcium intake

Figure 2.1: Suggested scheme for calcium supplementation in pregnant women, adapted from [1]

>105).

2. Variation of CPGs formats; CGPs are authored by different organizations such as the Scottish Intercollegiate Guidelines Network (SIGN), and the Guidelines Advisory Committee (GAC), NHS Clinical Knowledge Summaries; the National Institute for Clinical Excellence (NICE). Each of these organizations structure CPG content differently. As an example in the "Calcium supplementation in pregnant women" CPG [1] the suggested scheme for calcium supplementation in pregnant women is provided in a table format as shown in Figure 2.1 while in the "Management of chronic pain" CPG [3] similar type of recommendation is provided as plain sentence as shown in Figure 2.2.

3. Ambiguity and vagueness in CPGs; Use of ambiguous and vague terms hampers communication and leads to uncertainty and to variable interpretation [37]. The usage of a term like "elderly" in many CPGs is a good example of underspecification which a form vagueness.

4. Lack of the knowledge and information that are necessary to implement CPGs in practice [125].

The formalization of CPGs has been proposed as a method to increase adoption of CPGs [30]. But to be most useful, CPGs information should be available at the time and place it

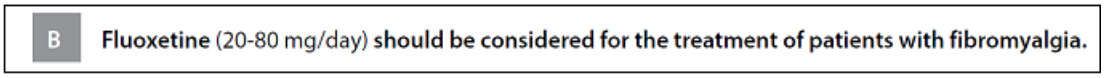


Figure 2.2: Fluoxetine recommendationf for patient with fibromyalgia, adapted from [3]

is needed and be specific to the task at hand [133]. Such push dissemination approach of CPGs can be achieved by integrating the information contained within CPGs into clinical systems.

2.1.2 Computer-interpretable Guidelines

Computer-interpretable Guidelines (CIGs) are formalized models of CPGs. Research on CIGs started about two decades ago and became more wide-spread in the late 1990s and early 2000s [105].

At a high level, the published guideline modelling formalisms fall into four categories, each type guideline formalism can model CPGs from different perspective.

1. Rule-based, is focused on modelling conditions and actions, examples includes: Arden Syntax [65], Guideline Definition Language (GDL) [110]

2. Document-based, is focused on organizing the heterogeneous information contained CPG documents in a formal model, examples includes: GEM [124]

3. Decision-logic expression languages, are focused on formalizing standard queries and expressions for decision support. examples includes: GEL [106], GELLO [129]

4. Task-network models, is focused on modelling hierarchical clinical steps "clinical workflow". eaxmples includes: GLIF [25], PROForma [53], SAGE [136], Asbru [122], GUIDE [108], EON [99], GASTON [82, 43], GLARE [11, 132], HELEN [128], NewGuide [103, 108, 36]

2.1.3 Information Extraction

Information Extraction (IE) is the process of automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities

from unstructured sources such as natural language.

Broadly, the IE techniques fall into three categories:

1. Rule-based methods [70, 93, 123], which are driven by predicates, and
2. Statistical methods [29, 146], which are based on a weighted sum of predicate firings.
3. Hybrid models [31, 35, 39, 50, 92, 111], that attempt to gain the benefits of both statistical and rule-based methods.

2.2 Introduction

To achieve our objective in minimizing the effort required by human modellers to integrate the knowledge contained within CPG into clinical systems we need to answer the following three questions:

- Which CIG to target?
- What IE approach to use?
- Which CDSS type to target?

Answering any of the above three questions would limit our alternatives in answering the other two questions. Considering that the human-related factor is critical in the formalization process, because regardless of the formalization approach followed, we would have partial results due to the computational complexity of natural language understanding, it is important to choose an approach that would allow human modellers not to only understand resulting CIG, but also to control the formalization process. Therefore, we needed to answer the second question first and decide on the IE approach. There are various tasks involved in the CPG formalization process, such as text parsing and tokenization, but the tasks that inflict the biggest challenges to the CPG formalization process are the IE ones, such as entity extraction and relationship extraction. In the following section we explore the landscape of IE approaches and advantages and disadvantages of each.

2.2.1 Rule-based vs ML Information Extraction

The landscape of Information Extraction as discussed in section 2.2.4 is clustered at a high level into rule-based approaches and statistical machine learning approaches or hybrid of the two. Each approach has its advantages and disatvantages which are depicted in Table 1, adapted from [34]

Table 2.1: Advantages and disadvantages of Rule-based and ML-based Information Extraction Systems

	Advantages	**Disadvantages**
Rule-based	Declarative Easy to comprehend Easy to maintain Easy to incorporate domain knowledge Easy to trace and fix the cause of errors	Heuristics Requires tedious manual labour
ML-based	Trainable Adaptable Reduces manual effort	Requires labeled data Requires retraining for domain adaptation Requires ML expertise to use or maintain Opaque

In a recent study that surveyed the IE technologies landscape the authors identified a major disconnect between industry and academia: "while rule-based IE dominates the commercial world, it is widely regarded as dead-end technology" [34].

Figure 2 shows evidence of this trend drawn from a survey of published research papers. The study examined the EMNLP, ACL, and NAACL conference proceedings from 2003 through 2012 and identified 177 different EMNLP research papers on the topic of entity extraction. The study classified these papers into three categories, based on the techniques used: purely rule-based, purely machine learning-based, or a hybrid of the two. The left side of Figure 2 shows the breakdown of research papers according to this categorization. The right side of Figure 2.2 shows the result of an industry survey of commercial entity extraction products from 54 different vendors listed in [150]. The industry survey was conducted in 2013, one year after the end of this ten-year run of NLP papers. Interestingly, the industrial landscape is not reflecting the research efforts of the

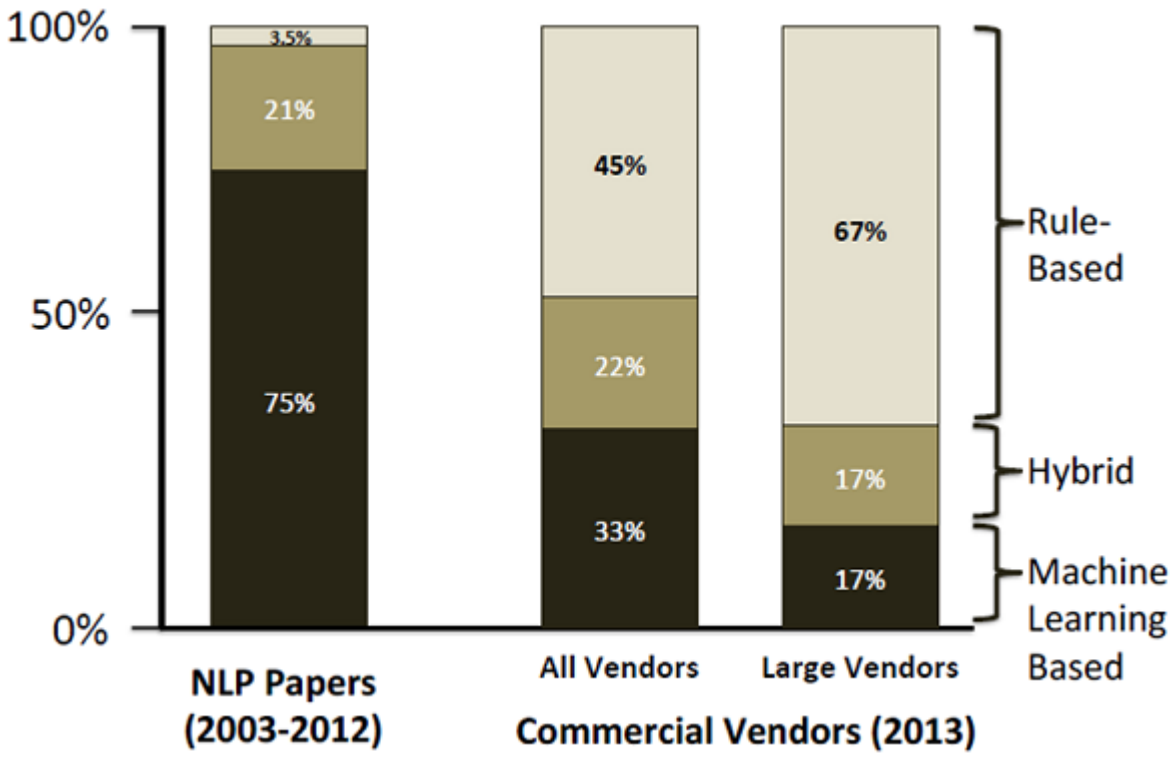


Figure 2.3: Implementation of Entity Extraction

previous 10 years. As shown in Figure 2.3, only 1/3 of the of the commercial IE product relied entirely on machine learning. The authors of [34] attribute the disconnect between the two communities to the way each community measures the benefits and costs of IE, as well as academia's perception that rule-based IE is devoid of research challenges.

Despite the greater accuracy of the ML-based IE, the commercial world is still more attracted to rule-based IE for its interpretability, which has the benefits of easier adoption and maintainability [79, 14]. In contrast, ML-based systems are riskier to adopt and more difficult to maintain [55, 140, 91].

One of the most notable reasons behind the academic community's steering away from rule-based IE systems is the perception of rule-based IE system lack of research problems [34]. While manually authoring rules is an easy task, automating it opens many research questions such as:

1) How can we prevent the system from generating complex rule sets which would be

difficult to understand or maintain,

2) How can we evaluate the generated rule systematically.

2.3 CPG Formalization System

There are multiple systems for CPG formalization each designed with a different focus, some are focused on the traceability between CPG and CIG, while others are more focused on hierarchical representation of CIGs. However, the existing systems are not focused on providing the human modellers a mechanism to control the granularity of the clinical knowledge to be extracted. We believe this is an important feature for a CPG formalization system in order to have the right level of details in the extract clinical knowledge. Providing the human modellers a mechanism to control the granularity of the clinical knowledge to be extracted is the main focus of the system we are proposing in Chapter 6. In the following subsections we present some of the existing CPG formalization systems.

2.3.1 Stepper

Stepper [118] is a mark-up tool for narrative guidelines, developed by the EuroMISE centrum – Kardio and the University of Economics, Prague, Czech Republic. The Stepper project has two main goals:

1. To develop a step-wise method for formalization (in this context, XML transformation) of text documents of clinical guidelines

2. To develop the Stepper tool, an XML editor enhanced with features to support the above method

Stepper has been designed as a document-centric tool, which takes a guideline text as its starting point and splits the formalization process into multiple user definable steps, each of which corresponds to an interactive XML transformation. The result of each step is an increasingly formalized version of the source document. An embedded XSLT processor carries out non-interactive transformation. Both the mark-up and the iterative transformation process are carried out by rules expressed in a new transformation language

based on XML, the so-called XKBT (XML Knowledge Block Transformation). This was because the well-known standard for transformation XSLT did not solve all problems when explicitly expressing transformations of knowledge in each step. Hence, a tailored transformation language was developed. The transformation process with Stepper consists of six steps:

1. Input text format. The format of the original guideline text is XHTML, the XML version of HTML.

2. Coarse-grained semantic mark-up. Basic blocks of the text are marked (e.g., headings, sentences) and parts without operation semantics are removed.

3. Fine-grained semantic mark-up. Complex sentences are rearranged into simpler ones and background knowledge is added. In addition, a data dictionary is created, which describes the clinical parameters involved.

4. Universal knowledge base. The original document is transformed into a universal knowledge base. This involves changing the structure of the document to achieve modularity, which is assumed to involve medical experts in part.

5. Export-specific knowledge base. The representation is adapted to ease the export to the target representation. Therefore, an export-specific knowledge base is produced from the universal one.

6. Target computational representation. The ultimate format is produced by the knowledge engineer. This step is assumed to be performed fully automatically using XSL style sheets.

2.3.2 GEM

The GEM Cutter [107] is a tool with the aim to facilitate the transformation of CPGs into the Guideline Elements Model (GEM) format [124]. It was developed by Yale Center for Medical Informatics at Yale University School of Medicine.

The GEM format is an XML-based guideline document model that can store and organize the heterogeneous information contained in practice guideline documents. GEM

is intended to facilitate the translation of natural language guideline documents into a standard computer interpretable format. It encodes considerable information about guideline recommendations in addition to the recommendations themselves, including the reason for each recommendation, the quality of evidence that supports it, and the recommendation strength assigned by the developers. For encoding guideline knowledge no programming knowledge is required, but a markup process is applied. The authoring process for GEM guidelines takes place in three steps:

1. The GEM document, which has an XML-based syntax, is created based on the original guideline using the GEM Cutter. The elements of the GEM document are then stored in a relational design database.

2. Knowledge Customization: meta-information is added, the guideline can be locally adapted, and abstract concepts of the guideline can be implemented. This step is guided by the knowledge customization wizard.

3. Knowledge Integration into the clinical workflow depending on local circumstances.

GEM is constructed as a hierarchy with more than 100 discrete elements and more than nine major branches. The majority of the elements describes properties of the guideline as a whole (e.g., title, developer, purpose, target, target population). The content of the guideline can be described in detail by three groups: Recommendations, definitions, and algorithm. Recommendations can be conditional or mandatory. For the recommended action the benefit, risk, and cost are stored and reason, evidence quality, strength of recommendation, costs, and so on are annotated. Definitions consist of term and term meaning, which are both free text. Algorithm consists of an action step, a conditional step, a branch step, and a synchronization step.

Unlike the Stepper approach, GEM Cutter does not retain the connection between the CPG and the formal GEM format.

2.3.3 Document Exploration and Linking Tool / Addons (DELT/A)

The Institute of Software Technology and Interactive Systems at the Vienna University of Technology is developing a tool to provide a relatively easy way to translate free text

into various (semi-)formal, XML-based representations. It achieves this by displaying both the original text and the translation, and showing the user which parts of the formal code correspond to which elements of the original text. This not only makes it easier to author plans, but also to understand the resulting constructs in terms of the original guideline.

DELT/A [81, 138, 139] provides two main features:

1. linking between a textual guideline and its formal representation, and
2. applying design patterns in the form of macros.

DELT/A allows the definition of links between the original guideline and the target representation. Therefore, if someone wants to know the origin of a specific value in the XML file DELT/A can be used to jump to the correlating point in the text file where the value is defined and the other way round.

The second feature of DELT/A is the usage of macros. A macro combines several XML elements, which are usually used together. Thus, using macros allows creating and extending specific XML files more easily through the usage of common design patterns.

DELT/A supports the following tasks:

1. Authoring and augmenting guidelines.
2. Understanding the (semi-)formal representation of guidelines.
3. Structuring the syntax of the (semi-)formal representation. DELT/A provides a structured list of elements of the target language – the macros – that need to be done in a way that best supports the authoring of plans.

By means of these features, the original text parts need not be stored as part of the target representation elements. The links clearly show the source of each element in the target representation. Additionally, there is no need to produce a guideline in natural language from the target representation since the original text remains unaltered.

The described approaches (2.3.1- 2.3.3) are mostly based on manual steps to gradually convert CPGs into CIGs; although the accuracy of the manual steps is straightforwardly

controlled, as the resulted accuracy is as good as the input provided by the human modellers, these approaches are expensive to use in formalizing large numbers of CPGs. The motivation of our work is to build a semi-automated information extraction based system that provides the human modellers a mechanism to control the granularity of the clinical knowledge to be extracted.

Chapter 3

Text Disambiguation - Review

3.1 Introduction

Human language is ambiguous. Words can have multiple meanings depending on the context in which they occur. In the study [102]authors manually tagged 192,800 English words with senses from WordNet [97], and found that the 121 most frequently used nouns have an average of 7.8 senses. Text disambiguation is the process of finding the correct meaning of every word in the text; this process is relatively easy for human but for machines it is as hard as an AI-complete problem, which means solving it would require solving all the difficult problems in artificial intelligence (AI), such as natural language understanding [69]. In the field of computational linguistic, the problem is called Word Sense Disambiguation (WSD), and is defined as the problem of finding the correct sense of a word when used in a particular context.

According to theoretical linguistics literature [90, 38], words with lexical ambiguity are divided into two types, namely polysemous and homographs. A word is polysemous if it can be used to express different unrelated meanings (e.g., "Astragalus" which refers to "ankle bone in the human body" and to the "Astragalus plant species"). On the other hand, a word is a homograph if it can be used to express different related meanings (e.g., "lens" which refers to "human body part in the eye" and to an "optical device"). There are, however, many other cases for which this decision is not clear.

The WSD is one of the oldest computational linguistic problems; it goes back to the

early days of machine translation (MT) in the late 1940s [141]. The difficulty of the problem was recognized in the MT research in the 1960s [17]. In the 1970s the artificial intelligence (AI) research community worked on the WSD problem [143, 115]. In the 1980s, large-scale lexical resources became available, which hugely benefited the progress of the WSD; for example in 1986, Lesk [88] used Oxford Advanced Learner's Dictionary of Current English (OALD) to address WSD. In the 1990s, WordNet became available, and had a greater impact on the WSD compared to the already existing machine readable dictionaries because of its hierarchical organization of word meanings called synsets. In the late 1990s, Senseval [75] started a community-based evaluation exercise for WSD, and it became easier to compare different WSD systems. Before Seneseval, the evaluation of different WSD systems required to consider the disparities in test words, and sense inventories used by the different systems which made the evaluation process extremely difficult. WSD has many applications in machine translation and Information Retrieval. In this thesis we are using WSD for Information Extraction purposes. Analyzing text accurately requires resolving ambiguities; for example solving multiple lexical matches in Named-entity recognition (NER).

Approaches to WSD are classified as either as unsupervised approaches which do not use a pre-annotated corpora, or a supervised approaches which use pre-annotated corpora to train itself. As WSD algorithms mostly rely on existing knowledge sources such as dictionaries or a lexical knowledge bases to get the full list of senses for a given word, this created another dimension to classify WSD approaches based on the knowledge source used, so that some approaches are called dictionary-based or knowledge-based.

3.2 WSD Task Description

WSD is essentially a classification task, where word senses are the classes, and each ambiguous term in the text could be assigned to more than one class. A WSD algorithm is the task of classifying ambiguous to one or more class "sense". More formally, a text T can be viewed as a sequence of words $(w_1, w_2, ..., w_n)$, and $Senses$ is a function that maps from an input word w to a discrete output space $S = \{s_1, ..., s_k\}$, such a mapping is usually given by means of external knowledge sources such as dictionaries. A WSD can be viewed

as the task of identifying a mapping A from words to senses such that $A(w_i) \subseteq Senses(w_i)$. The ultimate goal of WSD is to find a mapping A such that $|A(w_i)| = 1$, meaning to map each ambiguous word to just one sense, without such condition, a mapping A is considered to reduce ambiguity rather than resolving it. Another alternative of the WSD is the Word Sense Induction (WSI) or Word sense discrimination, where the $Senses$ function is not given, and therefore senses need to be induced. The WSI is essentially a clustering task where each cluster represents a sense. WSI is sometimes referred in the literature as the unsupervised approach to WSD. In this thesis our focus is on the approaches where the inventory of senses is provided, which at high level could be categorized as either supervised approaches or knowledge-based approaches.

3.3 Supervised WSD

Generally, supervised WSD have shown better results than the knowledge-based "unsupervised". However, it is difficult to find the required minimum number of occurrences per each sense of a word in any tagged corpora, a problem that is called knowledge acquisition bottleneck [56]. The knowledge acquisition bottleneck remains a big challenge for adopting supervised WSD at a large scale.

Probabilistic Methods

The Naïve Bayes classifier [56] is one of the simplest probabilistic classifiers. It has been used for resolving WSD [48, 85, 104, 57]. The classifier does no feature selection, but has the ability to combine evidence from multiple features. The Naïve Bayes assumes conditional independence of the attributes used for description, and consequently the structure and ordering of words is ignored. In spite of this simplifying assumption, the Naïve Bayes classifier perform well compared to the other supervised methods [28, 98, 102].

Decision Lists and Decision Trees

A decision list [114] is an ordered set of rules for categorizing test instances. The rules are tested in order based on their score until one matches the test instance. Decision lists

have already been successfully applied to WSD [149] and performed well.

A decision tree partitions the feature space into classes of word senses using a minimal set of features, and assembles them into a tree. Each node in the decision tree represents a choice point between a number of different possible values for a feature. WSD with decision tree is to find a path in the tree from the root to a leaf node that corresponds with the observed features. Decision trees have already been applied to WSD [98]using the C4.5 algorithm [109] to generate the decision tree.

Neural Networks

A Neural Network, is an interconnected assembly of artificial neurons that have associate weight reflecting the strength of connections. The neuron computes by forming a weighted sum of its input. Neural networks are trained to find the correct weight values that partition the training contexts into non-overlapping sets corresponding to the desired responses. A learning algorithm is used to tune the weights; usually it start by assigning random weight and with each learning step, it slightly modifies the weights toward the values that properly partition the training contexts. Neural networks have been applied to WSD and performed well compared to other supervised methods [84, 134].

Support Vector Machines (SVM)

SVM [24] is based on the idea of finding the linear hyper-plane between the two classes in a training set by maximizing the margin between the classes. As SVM is a binary classifier and WSD requires multiple class "sense" classification, the problem needs to be casted to multiple binary classification. In [86], the authors evaluated various learning algorithms for WSD and the result claimed that linear SVM is the best classifier compared to other supervised approaches.

Instance-Based Learning

Instance-Based Learning approaches perform classification by searching the training instances for the one that closely resembles the instance to be classified. There is no model

learned compared to other supervised approaches. The k-Nearest-Neighbor (kNN) algorithm is one of the widely used Instance-Based Learning approaches. The kNN classifier finds the k nearest training instances to the ambiguous test instance, and then classifies the ambiguous test instance based on the majority vote of k nearest training instances. In [101], the authors evaluate the kNN classifier and the result claimed that the kNN is one of the highest-performing classifier in WSD.

3.4 Knowledge-based WSD

The objective of knowledge-based disambiguation is to use knowledge resources such as dictionaries, and ontologies to infer the senses of words. Knowledge-based methods do not rely on annotated corpora and in some literature the definition of Knowledge-based WSD is further restricted to methods that do not rely on any corpora even the non annotated ones. The Knowledge-based methods usually have lower performance than their supervised alternatives, but they do not suffer from the knowledge acquisition bottleneck [56]problem, and therefore they have a wider coverage.

Lesk Algorithm

The Lesk algorithm [88] is based on the calculation of the word overlap between the sense definitions (glosses). Given two words, the algorithm selects those senses whose definitions have the maximum overlap. The Lesk algorithm inspired many sense definition overlap algorithms. In[76] the author presented a variation of the Lesk algorithm that disambiguates each word separately, which have been shown to outperform the original Lesk algorithm [137]. In [15] the authors proposed another variation of the Lesk algorithm called the adapted Lesk algorithm, in which each gloss is extended to include a definition of related words based on the WordNet semantic relations.

Semantic Similarity

The underlying hypothesis of the semantic similarity based approaches is the fact that words in a discourse must be related in meaning for the discourse to be coherent [62].

This is a property of human languages which presents a semantic constraint "Similarity constraint" that can be very useful for automating the disambiguation process. Similarity measures usually rely on semantic networks such as WordNet for computing metrics; The list below provide some of the similarity measures that could be used as a measure of semantic relatedness between a pair of concepts:

- Leacock–Chodorow [83] semantic relatedness between two concepts c_1 and c_2 is defined as:

  $Rel_{lch}(c_1, c_2) = -log\frac{length(c_1,c_2)}{2D}$

  where$length(c_{1,}c_2)$ is the length of the shortest path between c_1 and c_2 using node-counting, and D is the maximum depth of the hierarchy.

- Resnik [112] semantic relatedness between two concepts c_1 and c_2 is defined as

  $Rel_{rsn}(c_1, c_2) = IC(LCS(c_1, c_2)$

  whereIC is defined as:

  $IC(c) = -logP(c)$ and $P(c)$ is the probability of encountering an instance of concept c in a large corpus.

  $LCS(c_1, c_2)$ is the least common subsumer of c_1 and c_2.

- Jiang–Conrath [71] semantic relatedness between two concepts c_1 and c_2 is defined as

  $Rel_{jcn}(c_1, c_2) = \frac{1}{IC(c_1)+IC(c_2)-2*IC(LCS(c_1,c_2))}$

- The Lin [89] semantic relatedness between two concepts c_1 and $c_2$is defined as

  $Rel_{lin}(c_1, c_2) = \frac{2*IC(LCS(c_1,c_2))}{IC(c_1)+IC(c_2)}$

3.5 Knowledge sources for WSD

Knowledge-based WSD methods use diverse types of knowledge resources such as dictionaries, treasuries and lexical knowledge base which is one of the commonly used resources. In the lexical knowledge base category, WordNet [130] is one the most used resource in NLP domain-independent applications, while in the context of biomedical domain, the Unified Medical Language System (UMLS) is the biggest lexical resource publicly available. UMLS is the knowledge base we used in the disambiguation algorithms presented in Chapter 4. In the next subsection we describe the different components of the UMLS.

3.5.1 Unified Medical Language System

The Unified Medical Language System (UMLS) [67] is a repository of multiple controlled bio-medical vocabularies developed by the U.S. National Library of Medicine (NLM) and is composed of the following three knowledge sources:

1. The *Metathesaurus*, a vocabulary database of biomedical concepts with their various names, and the relationships among them. The Metathesaurus of the UMLS 2011AB release contains more than 2.6 million concepts collected from 161 vocabularies, such as: SNOMED Clinical Terms (SNOMED-CT) and Medical Subject Headings (MSH). The Metathesaurus organises knowledge based on concepts, where each concept is identified by a Concept Unique Identifier (CUI). A CUI may refer to multiple terms from the individual terminologies. These concepts are labeled with Atomic Unique Identifiers (AUIs). For example, the AUI Cold Temperature [A15588749] from MeSH and the AUI Low Temperature [A3292554] from SNOMED-CT are mapped to the CUI Cold Temperature [C0009264].

2. The *Semantic Network*, a set of semantic types to categorize all concepts represented in the Metathesaurus, and a set of semantic relations to define possible relationships between semantic types. The Semantic Network in the UMLS 2011AB release contains:

(a) 133 semantic types. Examples of semantic types include: Enzyme, Genetic Function, Therapeutic or Preventive Procedure, Laboratory Procedure.

(b) 54 semantic relations. Examples of semantic relations include: affects, treats, disrupts, prevents, process_of. Semantic relations are interconnected by semantic types. For example, the semantic types Enzyme and Genetic Function are interconnected by the semantic relation affects.

3. The *SPECIALIST Lexicon*, a set of lexical entries with one entry for each spelling or set of spelling variants in a particular part of speech and describes the morphologic, orthographic and syntactic properties of a word.

3.6 Mapping biomedical text to UMLS concepts

Mapping biomedical text to concepts in the UMLS Metathesaurus is not always a simple lexical exact match, as a term in a biomedical text can appear with a different variation than the one in the UMLS Metathesaurus. There are different approaches to map biomedical text to concepts in the UMLS Metathesaurus, some approaches [78] can detect noise in the text such as spelling errors and unfinished sentences while other methods are based on lexical analysis [12, 13] like MetaMap which we describe in the following subsection. We used the MetaMap tool in the CPG formalization system presented in Chapter 6 to map the text of CPG to concepts in the UMLS Metathesaurus.

3.6.1 MetaMap

MetaMap [12, 13] is a program developed by the NLM and is composed of the following five components:

1. Lexical/Syntactic Analysis: This component segments the biomedical text into phrases and then into terms. The text is Xerox part-of-speech tagged using the Xerox POS tagger.

2. Variant Generation: This component generates a variant for each phrase identified by the Lexical/Syntactic Analysis component. A variant is one or more phrase words accompanied with its spelling variants, derivational variants.

3. Candidate Identification: This component retrieves the set of concepts from the UMLS Metathesaurus that contain at least one variant identified by the Variant Generation component.

4. Candidate Evaluation: This component evaluates each candidate against the input text. The mapping score is computed using a combination of four linguistical measures: centrality; variation; coverage; and cohesiveness. The four measures are combined linearly such that coverage and cohesiveness get twice the weight of centrality and variation. The score is normalised to a value between 0 and 1,000, where a score of 1,000 means a perfect candidate.

5. Mapping Construction: This component combines all the Metathesaurus candidates that match the input text.

3.7 Evaluation data set

The availability of different test data sets complicate the task of comparing the accuracy of the different WSD algorithms, as a WSD algorithm does not perform with the same reported accuracy on all other data sets, since each data set has different coverage of terms and concepts.

Selecting a data set for the purpose of evaluating a WSD algorithm is a critical task as it impacts our understanding of the strength and weakness of the WSD algorithm. Obviously the broader the coverage of data set the better, but as the test data set has to be finite, it becomes impossible to build a test data set that can cover all possible terms in all plausible contexts and therefore it is crucial to define a few key properties required in a data set to be considered as a proper test data set for the disambiguation task. The main goal of a WSD algorithm is to properly disambiguate senses. Therefore, the richness of ambiguous terms should be the most important property of the test data set – and not

only that but to have equal distribution of the different senses of any ambiguous term. Below we provide a brief description of two data sets that are rich with ambiguous UMLS concepts:

- The NLM WSD [142] data set consists of 50 frequently occurring ambiguous terms from the 1998 MEDLINE baseline. Each ambiguous term in the data set contains 100 instances. The total number of instances is 5,000.
- The MSH WSD [72] data set contains 203 ambiguous words. The 203 words are composed of 106 ambiguous terms, and 88 ambiguous acronyms, and 9 words that are combinations of both. The data set has up to 100 instances for each possible sense. The total number of instances is 37,888.

The accuracy of the disambiguation algorithms presented in Chapter 4 is evaluated based on the MSH WSD data set.

Chapter 4

Disambiguation using UMLS

4.1 Introduction

Extracting information automatically from biomedical documents is challenged by the ambiguity of natural language, in which words can have multiple meanings. For instance the word "*lens*" has different meanings in the following two article title sentences which we captured from the MSH-WSD dataset [72].

a) Lens cadmium, lead, and serum vitamins C, E, and beta carotene in cataractous smoking patients.

b) A simple solution to lens fogging during robotic and laparoscopic surgery.

In the first sentence, lens is used to refer to a human body part, while in the second sentence lens is used to refer to a medical device part. Disambiguation is an essential task for the CPG formalization. Therefore, automating the CPGs disambiguation is required for the automation of the CPG formalization.

As detailed in Chapter 2, supervised learning approaches outperform unsupervised ones, but in the biomedical domain it is very expensive to create a manually annotated corpus for algorithm training purposes, which makes the unsupervised approach a more practical choice [74, 44, 96, 7]. In a WSD study focused in the biomedical domain [120] the authors believe that combining unsupervised learning and established knowledge proved to be most effective.

This chapter presents two unsupervised graph-based approaches to WSD in the biomedical domain that use the UMLS [67] as its knowledge base.

4.2 Background and Related Work

Most unsupervised WSD studies are domain ignorant, meaning that they are not customised for a specific field or domain. The key component that classifies an unsupervised WSD as domain specific is the knowledge base, for example the UMLS is commonly leveraged by WSD focused on the biomedical domain [95, 9] while WordNet [130] is commonly leveraged by domain-independent WSD [8, 127, 100, 135]. In Table 4.1. we list six recent unsupervised graph-based WSD algorithms along with their reported accuracy.

In domain-independent WSD the Senseval [74, 44, 96, 7], with its different versions, is the commonly used data set for algorithms evaluation. WordNet and Senseval can still be applied to biomedical text disambiguation but would result in lower accuracy when compared to a biomedical knowledge base and dataset. We can clearly see the difference when we compare the results between [8] and [9], where the authors applied the same algorithm, but used WordNet and Senseval in the first attempt [8] and UMLS and NLM-WSD in the second attempt [9] in which they achieved close to 10% accuracy improvement.

Evaluating our algorithms by comparing accuracies with all algorithms presented in Table 4.1 was not possible because these algorithms' implementation are either not publicly available or use different test data sets than the MSH-WSD test that we used. Instead, we defined our baseline algorithm to be a disambiguation algorithm that randomly chooses one sense from the set of plausible senses for each ambiguous term, which would result in an accuracy of 50% as ambiguous terms have two or more plausible senses.

4.3 Using UMLS Semantic Network

The disambiguation algorithm we propose is based on the hypothesis [57] that words closely located to each other in a text must have some degree of semantic relatedness, examples of semantic relatedness metrics are provided in section 3.4. We used the UMLS

Table 4.1: Recent Unsupervised Graph-based WSD Approaches

	Knowledge base	**Evaluation Dataset**	**Accuracy**
Bridget McInnes, Ted Pedersen, Ying Liu,Genevieve Melton (2011) [95]	UMLS Metathe-saurus	MSH-WSD	72.0%
Eneko Agirre, Aitor Soroa, Mark Stevenson (2010) [9]	UMLS Metathe-saurus	NLM-WSD	68.1%
Eneko Agirre, Aitor Soroa (2009) [8]	WordNet	Senseval-2, Senseval-3	58.6% - 57.4%
Ravi Sinha, Rada Mihalcea (2007) [127]	WordNet	Senseval-2, Senseval-3	56.4% - 52.4%
Roberto Navigli, Mirella Lapata (2007) [100]	WordNet / EnWordNet	SemCor, Senseval-3	–
George Tsatsaronis, Michalis Vazirgiannis, Ion Androutsopoulos (2007) [135]	WordNet	Senseval-2	49.2%

Algorithm 4.1 Graph-based WSD algorithm using UMLS semantic network

```
1: procedure WordSenseDisambiguate(W, t, s)
2:     Load UMLS semantic network as a graph G
3:     Map words W_1..n to UMLS semantic types
4:     let A = { sematic types of W_t}
5:     let B = { sematic types of W_l | l = (t − 1..t − s) ∪ (t + 1..t + s)}
6:     for each a in A do
7:         for each b in B do
8:             RelatednessDist(a) = RelatednessDist(a) + Shortestpath(a, b, G)
9:         end for
10:    end for
11:    let m = minimum{RelatednessDist(a) | a ∈ A}
12:    return {a | a ∈ A ∧ RelatednessDist(a) = m}
13: end procedure
```

Semantic Network as our knowledge base to find the relatedness between words. For example, an ambiguous term x (i.e. for which we have different semantic types) we take the neighbouring words before and after in a given window and check their respective semantic types using MetaMap [12, 13]. We select the semantic type of x the one which has the smallest distance from the set of neighbouring word semantic types based on UMLS Semantic Network. Algorithm 4.1 shows the pseudo-code of our approach.

In line 1, the algorithm takes three input parameters:

- W, a sequence of n words,
- t, an index in W pointing to the word we need to disambiguate,
- s, a window size of the words before and after t to include in the analysis.

In line 2, we convert the UMLS Semantic Network to a directed graph G, where each semantic type is a node, and semantic relations between semantic types are the edges between the nodes. In line 3, we map all words in W to UMLS concepts using the MetaMap [12, 13] tool. In line 4, we populate set A with all the semantic types of the

word we need to disambiguate W_t. In line 5, we populate set B with the semantic types of the words located before and after W_t by the given window size $2s + 1$. In lines 6-10, we measure the relatedness distance between each semantic type in set A to all semantic types in set B based on their closeness to each other in G . The semantic type of A that receives the lowest relatedness distance is deemed the semantic type of the correct sense. To prevent the algorithm from favouring one central edge and consequently resulting in equal relatedness distances, we added weights to edges in G using the betweeness centrality [26]. The betweeness centrality helped us rank two nodes in set A that have same distances to nodes in set B; modifying the edges in graph G using the betweenes centrality let the uncommon edges between the different nodes in set A have different weights and consequently different relatedness to nodes in set B.

As a running example we will use the following sentence from the MSH-WSD [72] data set.

- *A simple solution to lens fogging during robotic and laparoscopic surgery.*

The word we need to disambiguate is "*lens*". As provided by the MSH-WSD data set the word lens can have any of the three possible UMLS concepts and their corresponding semantic types are shown in Table 4.2.

Table 4.2: Ambiguous UMLS Concepts

UMLS Concept		**Semantic Type**
Unique Id	Name	
C0023308	Lens Diseases.	Disease or Syndrome
C0023318	Lens (device).	Medical Device
C0023317	Lens, Crystalline.	Body Part, Organ, or Organ Component

The correct concept of the word "*lens*" in the given sentence is C0023318 which has the sense of a medical device lens.

Figure 4.1 illustrates this running example. For simplicity, in the example we only take a size window of 1 and draw only 133 semantic types of graph G without the edges. We show elements of both set A and set B. Set A elements are the grey nodes representing

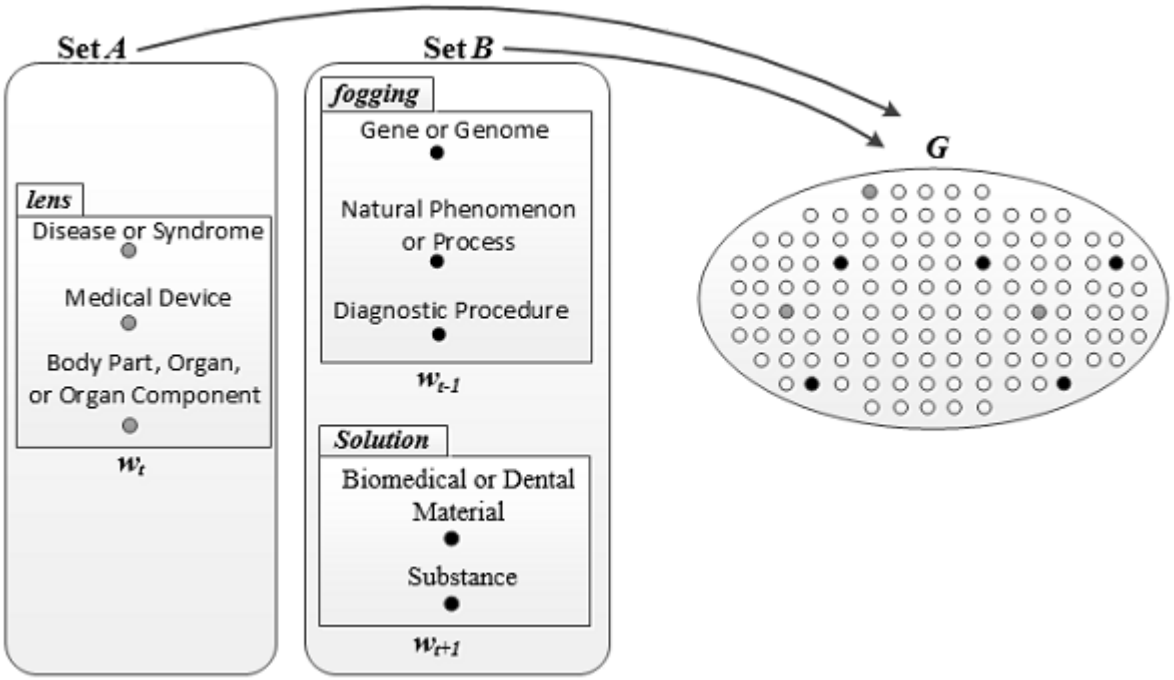


Figure 4.1: Graph-based WSD using UMLS semantic network

the three candidate semantic types of W_t word lens, and set B elements are the black nodes representing the semantic types we extracted from MetaMap for the W_{t-1} word "*solution*" and the W_{t+1} word "*fogging*".

We know that all the grey and black nodes of set A and set B must be nodes in the graph G, so we highlighted them in G. After having the graph G with highlighted grey and black nodes, the problem can be described as: which of the grey nodes is more related to the black nodes. To answer this question we calculate the sum of the shortest paths from each grey node to all the black nodes, and the grey node that receives the lowest values is deemed to have the highest relatedness.

4.3.1 Algorithm Evaluation

We evaluated our method using the MSH-WSD [72] dataset containing 203 ambiguous words. The 203 words are composed of 106 ambiguous terms, and 88 ambiguous acronyms, and 9 words that are combinations of both. The dataset has up to 100 instances for each

possible sense. The total number of instances is 37,888. We ran our algorithm on the MSH-WSD dataset with a window of size 3 and the resulting average accuracy was 60.3%. Table 4.3. shows the 10 words with highest accuracies and Table 4.4. shows the 10 words with lowest accuracies. The small size window of 3 is chosen for scalability reasons as the semantic types of the neighbouring words add a combinatorial set of distances to compute. One important fact worth of note is that in our algorithm we use a relatively small knowledge base that we do not alter. Comparing our accuracy with [95] who used a graph-based algorithm, the authors reported an average accuracy of 59% for a window size of 2 words, which are similar to our algorithm accuracy. Our algorithm has the advantage of using lightweight approach as it only uses semantic types in its graph representation which is a much smaller knowledge base than the one used in [95].

Table 4.3: Graph-based WSD using UMLS semantic network (Highest 10 accuracies)

Word	**True Positive**	**False Positive**	**False Negative**	**Accuracy**
CDA	192	6	0	97%
CTX	177	6	0	97%
FAS	190	8	0	96%
MCC	124	7	0	95%
BPD	186	12	0	94%
BSE	186	12	0	94%
DAT	187	10	1	94%
Epi	187	11	0	94%
SS	136	7	1	94%
CRF	185	13	0	93%

4.4 Using UMLS Metathesaurus

The limitation of the algorithm presented in Section 4.3 is the inability to disambiguate distinct terms that have the same UMLS semantic type. It is limitation resulted from using the UMLS Semantic Network as a knowledge base, in which many UMLS concepts

Table 4.4: Graph-based WSD using UMLS semantic network (Lowest 10 accuracies)

Word	**True Positive**	**False Positive**	**False Negative**	**Accuracy**
Lupus	12	0	285	4%
Medullary	8	0	190	4%
TPO	8	0	190	4%
TSF	2	0	51	4%
MBP	4	0	139	3%
TNC	5	0	162	3%
CCD	3	0	138	2%
RA	5	13	279	2%
Gamma-Interferon	1	0	197	1%
Murine sarcom virus	0	0	180	0%

are mapped to more than one semantic type. In the following algorithm we propose to leverage the UMLS Metathesaurus as our knowledge base source to solve the limitation from the algorithm presented in Section 4.3. We represented the UMLS Metathesaurus as a graph K, such that UMLS concepts are the nodes and relation between UMLS concepts are the edges. The proposed algorithm is inspired by the approach presented in [100].

The nodes and edges of graph K are created based on the following UMLS tables:

- The MRCONSO table, which contains mappings of concepts in individual source vocabularies to concepts in the UMLS, is used as the source of our nodes in the graph K, and we used the CUI filed as our node identity.

- The MRREL table, which contains contains both hierarchical and non-hierarchical relations between UMLS concepts, is used as the source of our edges in the graph K.

Table 4.5 shows a subset of relations between concepts that we extracted MRREL tables. The MRREL table contains ten different types of relations between concepts; for the performance consideration we focused on the following six relation types:

- PAR, the *parent* relation
- CHD, the *child* relation
- RB, the *broader* relation
- RN the *narrower* relations
- SIB, the *sibling* relation
- RO, the *other* relation

Table 4.5: UMLS Concepts Relations

UMLS Concept	**Relation**	**UMLS Concept**
Metabolisms, Energy	CHD	Rates, Basal Metabolic
Metabolisms, Energy	PAR	Processes, Metabolic
Drug-Induced Abnormality	RN	Fetal hydantoin syndrome
Drug-Induced Abnormality	PAR	Deformity
Drug-Induced Abnormality	RN	Warfarin syndrome

After building the knowledge source and represent as the graph K, it is fed into our algorithm along with the following inputs:

- W, a sequence of n words, representing the text containing the word to be disambiguated,
- t, an index in W pointing to the word we need to disambiguate,
- s, a window size of the words before and after t,
- A, a set of plausible senses for the word being disambiguated.

Algorithm 4.2 shows the pseudocode of our approach. We progressively build a graph for each W_t word to be disambiguated; the graph is composed of:

Algorithm 4.2 Graph-based WSD using UMLS Metathesaurus

1: **procedure** WordSenseDisambiguate(K, W, t, s, A)
2: let $V = \{$ UMLS concept of $W_l \mid l = (t-1..t-s) \cup (t+1..t+s)\}$
3: let $V = V \cup A$
4: **for** each v in V **do**
5: $X = DFS(K, v, p)$
6: **for** each x in X **do**
7: **if** $(x \notin V)$ **then**
8: let $V = V \cup \{x\}$
9: **end if**
10: **end for**
11: **end for**
12: let $E = GetEdges(V, K)$
13: let $VRanks = Betweenness(V, E)$
14: let $m = maximum\{VRanks(a) \mid a \in V \wedge a \in A\}$
15: return m
16: **end procedure**

- V, a set of nodes representing the UMLS concepts of the words before and after W_t within a window of size s, combined with the set A. We used the MetaMap tool for mapping words to UMLS concepts. In line 4-11, we loop through all nodes in V, and for each node in V we search for its neighbour nodes in the graph K using depth-first search. All neighbour nodes found in K that do not exist in V are added to the V.
- E, the edges that interconnect all nodes in V based on the K graph.

The algorithm uses the following 3 functions:

- **DFS**(K , v , p), which returns the set of nodes encountered when performing depth-first search starting from node v in the graph K at a maximum depth p.
- **GetEdges**(V , K), which returns the set of edges in graph K that interconnect all nodes in the V set.

- **Betweenness** (V , E), which returns a set of all nodes in V with their betweeness metric.

We compute the betweenness score [54] of all nodes of the graph (V , E), the node in V that exist in A and receive the highest betweenness score is assumed to be the node of the correct sense of the W_t word.

4.4.1 Algorithm Evaluation

We evaluated our method using the MSH-WSD [72] dataset containing 203 ambiguous words. The 203 words are composed of 106 ambiguous terms, and 88 ambiguous acronyms, and 9 words that are combinations of both. The dataset has up to 100 instances for each possible sense. The total number of instances is 37,888. We ran our algorithm on the MSH-WSD dataset with a window of size 2 and the resulting average accuracy was 59.2%. Table 4.6 shows the 10 words with highest accuracies and Table 4.7 shows the 10 words with lowest accuracies.

Table 4.6: Graph-based WSD using UMLS Metathesaurus (Highest 10 accuracies)

Word	**True Positive**	**False Positive**	**False Negative**	**Accuracy**
Lawsonia	99	16	0	86.09%
Eels	104	26	0	80.00%
HR	87	10	12	79.82%
DE	98	27	1	77.78%
PCB	93	28	6	73.23%
Torula	89	33	0	72.95%
PAF	82	33	0	71.30%
Callus	99	51	0	66.00%
EM	82	47	0	63.57%
CCD	88	42	11	62.41%

Sources of ambiguity varies but one of the common sources of ambiguity is that many medical abnormalities share the same lexical term with the physical object associated

Table 4.7: Graph-based WSD using UMLS Metathesaurus (Lowest 10 accuracies)

Word	**True Positive**	**False Positive**	**False Negative**	**Accuracy**
Hemlock	19	54	4	24.68%
PCP	72	225	0	24.24%
CP	70	227	0	23.57%
Arteriovenous Anastomoses	30	99	0	23.26%
DON	26	100	0	20.63%
ORI	22	101	0	17.89%
MAF	21	99	0	17.50%
PCA	79	390	22	16.09%
WBS	17	111	0	13.28%
PHA	12	98	0	10.91%

with it. As an example from Table 4.6, the term callus is ambiguous because it is one lexical term that is used to refer to the callus tissue and the acquired abnormality from the callus tissue, as shown in the following two sentences from the MSH-WSD [72] test dataset.

- Callus tissue sense: Myostatin (GDF-8) deficiency increases fracture callus size, Sox-5 expression, and callus bone volume.Myostatin (GDF-8) is a negative regulator of skeletal muscle growth and mice lacking myostatin show increased muscle mass.

- Callus abnormality sense: The association between callus formation, high pressures and neuropathy in diabetic foot ulceration.

Comparing our accuracy with [95] who used a graph-based approach, the authors reported an average accuracy of 59% to 61% for a window size of 2 words, which is similar to our algorithm accuracy. Using the betweenness score [54] as the semantic similarity metric in our disambiguation algorithm resulted in similar accuracy obtained from the semantic similarity metrics used in [95] such as the Leacock–Chodorow metric [83] and the Resnik

metric[112] that were described in Section 3.4.

4.5 Analyzing the impact of UMLS relations on the Word Sense Disambiguation accuracy

Comparing accuracies of both algorithms in Section 4.3 and 4.4 motivated us to find why the accuracy are very close despite the big difference in the size and details of the knowledge sources. Our expectation was to experience accuracy improvement when switching from the UMLS Semantic Network (Section 4.3) to the UMLS Metathesaurus (Section 4.4). Therefore, we analyzed the impact of using different subsets of the UMLS Metathesaurus on the resulted accuracy of the unsupervised WSD algorithm.

Our focus in this analysis is on the unsupervised WSD algorithms that leverage the UMLS Metathesaurus as a knowledge source. There have only been a few attempts in this research area with different reported accuracies. Interestingly the difference in accuracy cannot only be credited to the rigorousness of the algorithm as each algorithm used different subsets of the UMLS, which could have a special impact. Moreover not all algorithms were evaluated using the same dataset.

There are multiple unsupervised WSD algorithms that leverage the UMLS. Some algorithms used the UMLS Metathesaurus knowledge source [9, 95], while others used the UMLS Semantic Network knowledge source [66, 10]. Generally, WSD algorithms that leverage the UMLS Semantic Network would run faster compared to the WSD algorithms that leverage the UMLS Metathesaurus, because of the smaller size of the Semantic Network knowledge base. But the main disadvantage of leveraging the UMLS Semantic Network is the fact that it restricts the WSD algorithm to only disambiguate words with concepts that belong to different UMLS semantic types. In the following subsections we provide a brief description of two different types of unsupervised WSD algorithms that used the UMLS Metathesaurus knowledge source.

4.5.1 Similarity-based unsupervised WSD

The similarity-based unsupervised WSD measures the similarity of each sense of the word being disambiguated to the words in the surrounding text, and the sense that has the highest similarity is assumed to be the correct one. The approach presented in [95] is a recent implementation of a similarity based unsupervised WSD.

4.5.2 Graph-based unsupervised WSD

The graph-based unsupervised WSD builds a graph representing all possible senses of the word being disambiguated. The nodes in the graph correspond to the senses and the edges in the graph correspond to the relation type between senses (e.g. parent, child, broader). Next, the graph is assessed to determine the importance of each node: the node "*sense*" that is considered the most important of the word being disambiguated is assumed to be the correct one. The approach presented in [9] is a recent implementation of a graph-based unsupervised WSD.

4.5.3 Methods

For the purpose of our analysis we implemented the graph-based unsupervised WSD algorithm presented in Section 4.4 with a slight variation to the way we compute the vertex ranks in our graph. We used the PageRank metric [27] rather than the Betweenness metric [54] to compute the vertex ranks. The algorithm is inspired by the approach presented in [100]; Algorithm 4.3 shows the pseudo-code of our approach. We ran the WSD algorithm against different subsets of the UMLS Metathesaurus. The way we split the UMLS Metathesaurus into smaller knowledge bases is by the different relations defined in the MRREL table, so each subset contains all the UMLS concepts but with only specific types of relations interconnecting them. We created four Metathesaurus subsets:

- PAR/CHD, a subset that contains only the parent and child relations;
- RB/RN, a subset that contains only the broader and the narrower relations;
- SIB, a subset that contains only the sibling relation;

Algorithm 4.3 Graph-based WSD using UMLS Metathesaurus

```
1: procedure WordSenseDisambiguate(K, W, t, s, A)
2:     let V = { UMLS concept of W_l | l = (t − 1..t − s) ∪ (t + 1..t + s)}
3:     let V = V ∪ A
4:     for each v in V do
5:         X = DFS(K, v, p)
6:         for each x in X do
7:             if (x ∉ V) then
8:                 let V = V ∪ {x}
9:             end if
10:        end for
11:    end for
12:    let E = GetEdges(V, K)
13:    let VRanks = PageRank(V, E)
14:    let m = maximum{VRanks(a) | a ∈ V ∧ a ∈ A}
15:    return m
16: end procedure
```

- RO, a subset that contains only the other relation.

Input:

- K, a graph representing the subset the UMLS Metathesaurus,
- W, a sequence of n words,
- t, an index in W pointing to the word we need to disambiguate,
- s, a window size of the words before and after t to include in the analysis,
- A, a set of plausible senses for the word being disambiguated. Only one element of A is the correct sense.

The algorithm uses the following 3 functions:

- **DFS**(K , v , p), which returns the set of nodes encountered when performing depth-first search starting from node v in the graph K at a maximum depth p.
- **GetEdges**(V , K), which returns the set of edges in graph K that interconnect all nodes in the V set.
- **PageRank**(V , E), which returns a set of all nodes in V with their PageRank metric.

Each of the four UMLS Metathesaurus subsets is represented as a K graph, where the UMLS concepts are the nodes, and the UMLS relations between concepts are the edges. For the mapping step (line 2 of the WordSenseDisambiguate function), we used the MetaMap tool. In the DFS function we set p (the maximum depth of the depth-first search) to 1 for execution time purposes.

4.5.4 Results and Discussion

We evaluated our method using the MSH-WSD [72] dataset containing 203 ambiguous words. The 203 words are composed of 106 ambiguous terms, and 88 ambiguous acronyms, and 9 words that are combinations of both. The dataset has up to 100 instances for each possible sense. The total number of instances is 37,888. We executed our algorithm against the MSH-WSD test dataset, with a window size of 2, and we executed the algorithm using the 4 subsets of the MRREL table (PAR/CHD, RB/RN, RO, SIB). Table 4.8 shows the average accuracy for the usage of each of the 4 subsets of the MRREL table. The results shows the PAR/CHD subset have the best average accuracy, but from our observation of the terms and acronyms with top accuracies of each MRREL table subset as shown in Table 4.9-4.12 we can conclude that there is no real winner. The RO relation which is worst performing among the 4 subsets was able to detect specific terms and acronyms with higher accuracy compared to the other 3 subsets that have higher average accuracy.

Table 4.8: Graph-based WSD - Average accuracy

Category	**PAR/CHD**	**RB/RN**	**RO**	**SIB**
Term	66.00%	38.42%	26.38%	61.16%
Acronym	64.14%	30.95%	30.96%	60.84%
Average	65.07%	34.69%	28.67%	61.00%

Table 4.9: Graph-based WSD - Highest 5 accuracies of the PAR/CHD relation

Term/Acronym	**PAR/CHD**	**RB/RN**	**RO**	**SIB**
dC	**94.44%**	51.01%	5.56%	50.51%
HC1	**93.94%**	49.49%	50.51%	60.16%
PCD	**93.94%**	49.49%	49.49%	36.36%
BPD	**93.43%**	0.00%	0.00%	50.51%
SCD	**92.93%**	0.51%	49.49%	50.00%

Table 4.10: Graph-based WSD - Highest 5 accuracies of the RB/RN relation

Term/Acronym	**PAR/CHD**	**RB/RN**	**RO**	**SIB**
PHA	15.45%	**86.36%**	9.09%	15.45%
PAF	22.61%	**86.09%**	16.52%	96.52%
PCB	77.95%	**83.46%**	0.80%	68.50%
lymohogranulomatosis	19.33%	**82.35%**	83.19%	15.13%
DON	2.38%	**78.57%**	3.97%	76.19%

Table 4.11: Graph-based WSD - Highest 5 accuracies of the RO relation

Term/Acronym	**PAR/CHD**	**RB/RN**	**RO**	**SIB**
HR	25.69%	0.00%	**88.07%**	22.94%
lymohogranulomatosis	19.33%	82.32%	**83.19%**	15.13%
sex factor	8.40%	0.00%	**71.76%**	29.01%
CDR	41.50%	33.33%	**68.03%**	40.82%
Callus	21.33%	2.00%	**66.00%**	36.00%

Table 4.12: Graph-based WSD - Highest 5 accuracies of the SIB relation

Term/Acronym	**PAR/CHD**	**RB/RN**	**RO**	**SIB**
PAF	22.61%	86.09%	16.52%	**96.52%**
MCC	86.26%	3.05%	25.95%	**93.89%**
Eels	19.23%	4.62%	0.00%	**91.54%**
BAT	46.46%	50.00%	0.00%	**90.91%**
CAD	54.55%	49.49%	50.00%	**90.40%**

4.6 Summary

In this chapter we propose two unsupervised disambiguation algorithms for biomedical text using the UMLS knowledge sources. In the first algorithm we used the UMLS Semantic Network and in the second algorithm we used the UMLS Metathesaurus. In the UMLS Semantic Network based approach, we built a graph to represent the UMLS semantic network, such that nodes represent the UMLS semantics types, and edges represent the semantic relations between the semantic types. In the the UMLS Metathesaurus based approach, we built a graph to represent the UMLS Metathesaurus such that nodes represent the UMLS concepts extracted from the MRCONSO table and edges represent the relation between UMLS concepts extracted from the MRREL table.

Both algorithms are evaluated using the MSH-WSD dataset, the UMLS semantic network based approach resulted an average accuracy of 60.3%, while the UMLS Metathesaurus based approach resulted an average accuracy of 59.2%. Despite, the big difference in the size of the knowledge source used in both algorithm, the resulted accuracy is close. Therefore, we analyzed the impact of using different subsets of the UMLS Metathesaurus on the resulted accuracy. In the analysis we divided the UMLS Metathesaurus into 4 subsets based on relation type and we found that each subset excels in disambiguating different terms/acronyms, which indicates that using all relations of the UMLS MRREL table is not necessarily the best approach.

Chapter 5

Tools and Technologies used

5.1 Introduction

Building Information Extraction (IE) system requires a synthesis of different components to inter-work. Many tools have been developed to support IE for domain-independent text, but less tools have been built for the biomedical domain. This chapter discusses the generic and biomedical-domain specific tools and technologies that we leveraged to perform common IE tasks. Some of the tools discussed in this chapter had to be programmatically customized in order to inter-work with the rest of the components of our system, which we cover in detail in Chapter 6.

5.2 UIMA

The Unstructured Information Management Architecture (UIMA) [51] is a component architecture and software framework (Apache UIMA) implementation for the analysis of unstructured information like natural language text, speech, images or videos. UIMA uses the concept of an Analysis Engine, which analyzes a segment of unstructured data "called artifact" and saves the information in a comment analysis structure (CAS) object. CAS is the data structure to represent the data artifact and the metadata annotations which is exchanged between the UIMA Analysis Engines. The artifact is encapsulated in one or more Subjects of Analysis (Sofas). The definition of an annotation structure is

called the Type System which describes a domain model.

In a survey study [16] the authors compared UIMA to other frameworks such as GATE [40] and assessed UIMA as the most evolved and comprehensive architecture. In another study [63], the authors valued UIMA over GATE for two reasons: it separates the engineering problems from the NLP issues and takes in charge many of the engineering needs like the data transmission or the data serialization; Second, it provides a programming framework for defining and managing NLP objects present in analysis tasks such as creating or getting the annotations of a given type.

5.3 UIMA Ruta

Apache UIMA Ruta [80] is a rule-based script language supported by Eclipse-based tooling called the Apache UIMA Ruta Workbench. There are other rule-based systems for information extraction such as is JAPE [41], AFST [23], SystemT [33], TokensRegex [32]. JAPE or "Java Annotation Patterns Engine" is one of the most noted systems, and it is integrated into the GATE [40] framework. JAPE creates finite state transducers that operates over both text and annotations, based on regular expressions. One of the main advantages of the UIMA Ruta is that it offers a more compact representation of rules. A rule in the UIMA Ruta language is composed of a sequence of rule elements that consists of the following four parts:

1. "*Matching reference condition*" is a type of annotation by which the rule element matches on the covered text of one of those annotations.

2. "*Optional quantifier*" specifies whether it is necessary that the rule element successfully matches and how often the rule element may match.

3. "*List of conditions*" specifies additional constraints that the matched text or annotations need to fulfill.

4. "*List of actions*" defines the consequences of the rule and often creates new annotations or modifies existing annotations.

Both the Ruta rule language and the UIMA Ruta Workbench integrate smoothly with Apache UIMA and are designed to enable rapid development of text processing applications supporting all aspects of the development process like authoring of rules, syntax checking, debugging, and quality assessment.

Ruta rule language provides:

- 41 Actions (MARK, UNMARK, CREATE, ADD, ...)
- 27 Conditions (CONTAINS, PARTOF, REGEXP, AFTER ...)

5.4 openEHR

OpenEHR is a non-proprietary standard for EHR architecture which purpose is to facilitate the creation and sharing of health records by consumers and clinicians. OpenEHR standard is based on modelling the clinical domain using the so called two-level approach [19]. The two-level approach distinguishes a Reference Model, used to represent the generic properties of health record information, and Archetypes, which are meta-data used to represent the specific characteristics of the various kinds of clinical data [73].

The Reference Model (RM) represents the general features of the components of the EHR, how they are organized and the context information needed to satisfy both the ethical and legal requirements of the record. It includes a flexible syntax and some generic types of clinical information as observations, evaluations, instructions and actions. Then, instances or specialisations of that RM are devised in the form of constraints expressed through more concrete "archetypes", which serve as a shared language for common and specialised clinical concepts as "blood pressure", "medication order", and "heart rate".

All clinical information created in the openEHR EHR is ultimately expressed in "Entries". An Entry is logically a single 'clinical statement', and may be a single short narrative phrase, but may also contain a significant amount of data, e.g. an entire microbiology result, a psychiatric examination note, a complex medication order. In terms of actual content, the Entry classes are the most important in the openEHR EHR Information Model, since they define the semantics of all the 'hard' information in the record. They are intended to be archetyped, and in fact, archetypes for Entries and sub-parts

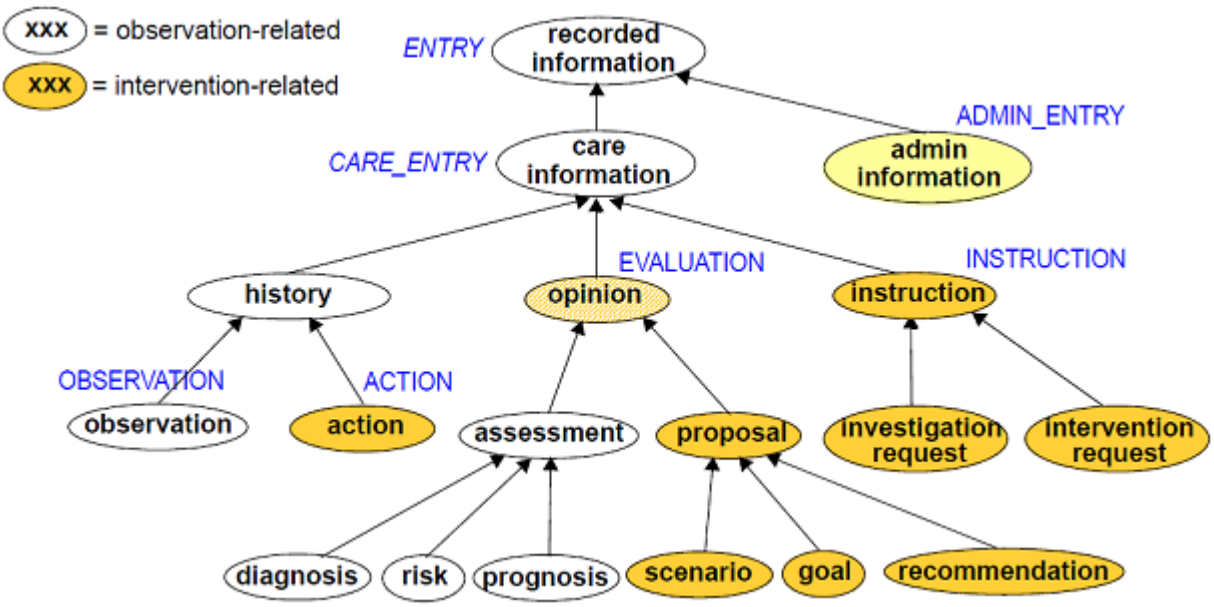


Figure 5.1: Ontology of recorded clinical information. Adapted from [20]

of Entries make up the vast majority of archetypes defined for the EHR. The openEHR ENTRY subtypes are shown in the ontolgy in Figure 5.1.

There are five concrete subtypes: ADMIN_ENTRY, OBSERVATION, EVALUATION, INSTRUCTION and ACTION, of which the latter four are kinds of CARE_ENTRY. The choice of these types is based on the clinical problem-solving process shown in Figure 5.2 [20].

Archetype instances themselves conform to a formal model, known as an Archetype Model (which is related to the Reference Model) and are specified using the Archetype Definition Language (ADL), Figure 5.3 shows an example of an the ADL of the “Tobacco Use” observation Archetype downloaded from the openEHR Clinical Knowledge Manager (CKM) [6], which is a web-based repository allowing for archetype search, browse and download.

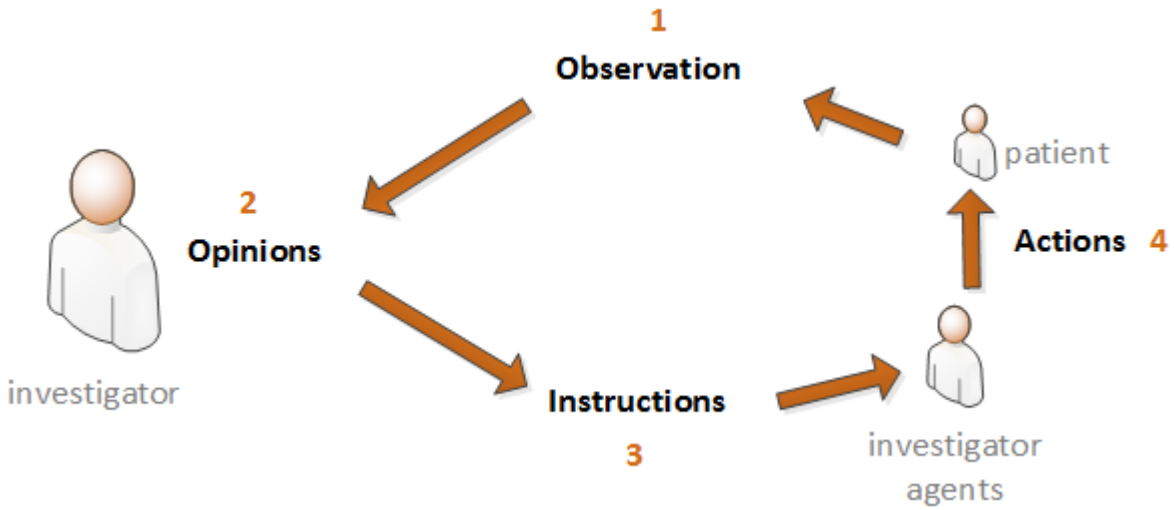


Figure 5.2: Relationship of information types to the investigation process. Based on [19]

5.5 Guideline Definition Language (GDL)

The Guideline Definition Language (GDL) [110] leverage the designs of openEHR Reference Model and Archetype Model, Figure 7.4 shows the classes that are based on the openEHR specifications in blue colour. The yellow classes are the ones that were introduced to represent guidelines. The Guide class is the parent class used to model a CPG, in our formalization process we are most interested to capture clinical knowledge which would be represent as a set of rules using the Rule class.

The Rule class has two main members:

- *whenStatments*, a list of expressions that represent the condition that must be evaluated before the rule get fired.
- *thenStatments*, a list of expressions that represent the action that must be execute if the condition was met.

Both *whenStatments* and *thenStatments* are of type ExpressionItem which is the core of all the rules that we will extract from our narrative text.

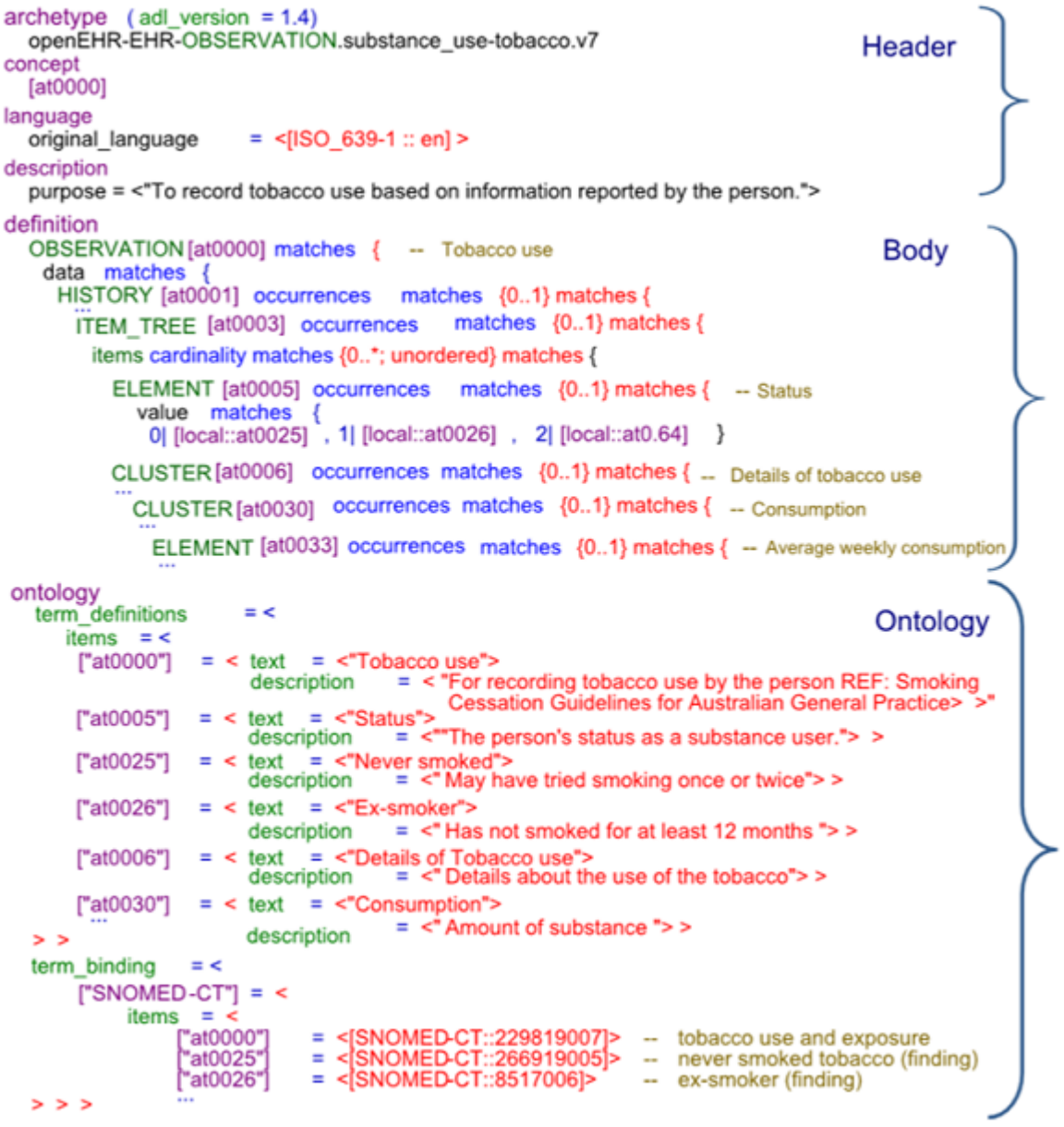


Figure 5.3: Tobacco use Archetype

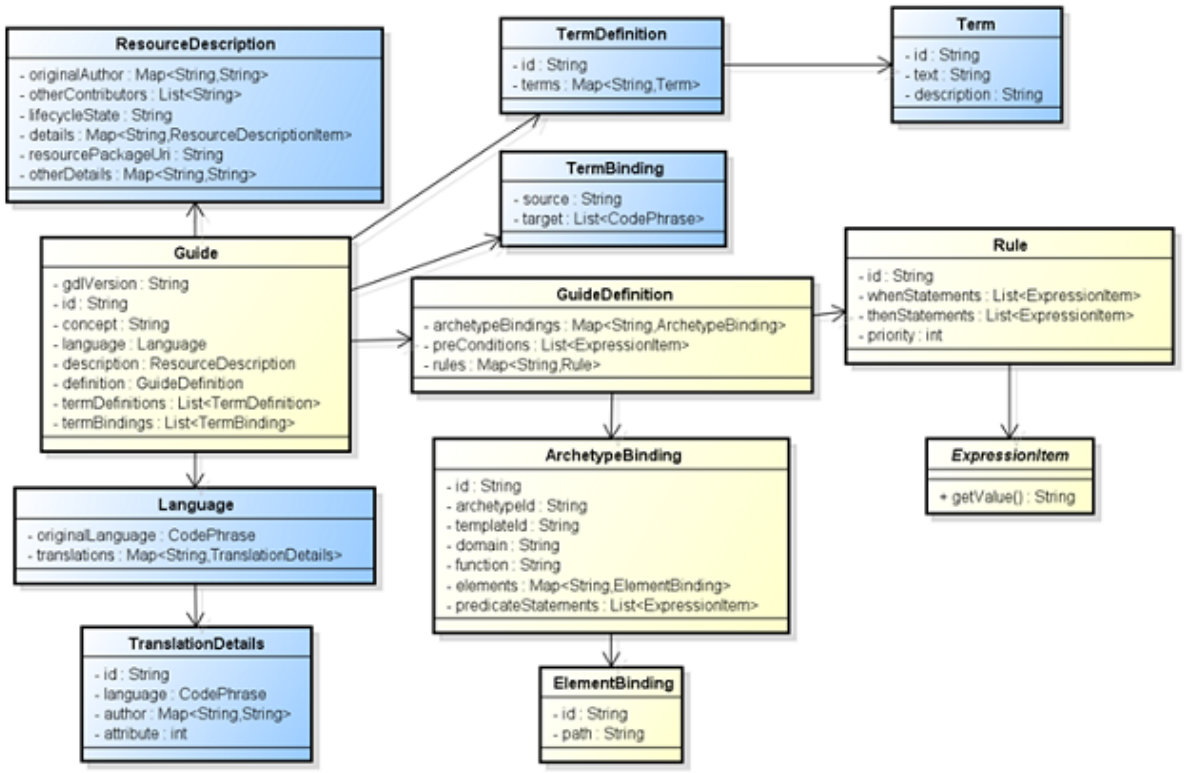


Figure 5.4: GDL Guide Package

5.6 Summary

This chapter has outlined the technologies and tools used for the development of our CPG formalization system. The following two chapters discuss how these tools are leveraged and customized. Chapters 6 provides the details of each of CPG formalization system components, and how these interconnect with the rest of other components of the system. Chapter 7 provide a slight modification to the system for the purpose of evaluating text coverage across multiple CPGs.

Chapter 6

Putting it all together: The CPG formalization system

6.1 Introduction

CPG formalization approaches that have been published and described in Chapter 2 are either based on a set of manual steps to gradually convert CPGs into CIGs, or based on automated information extraction mechanisms frequently using linguistic patterns. While the accuracy of the manual approaches is straightforwardly controlled, as the resulted accuracy is as good as the input provided by the human modellers, these approaches are expensive to use in formalizing large numbers of CPGs. On the other hand, the automated and semi-automated information extraction based approaches are in theory more suited to formalize a relative large number of CPGs but these approaches do provide the human modellers a mechanism to control the granularity of the clinical knowledge to be extracted. Therefore, the motivation of our work in building a CPG formalization system is to balance between accuracy and the specificity of CPG knowledge extracted. This chapter describe how all the tools fit together to form a system for CPG formalization, with a focus on the Information Extraction automation task.

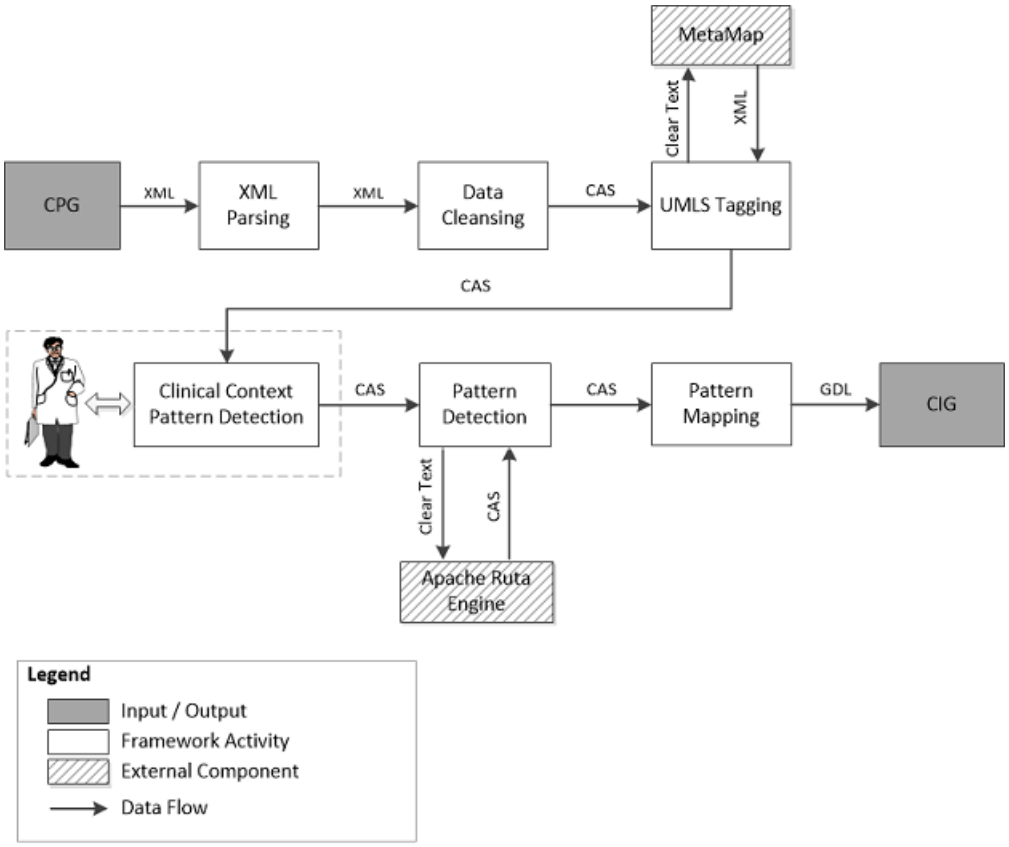


Figure 6.1: CPG formalization activities

6.2 Methods

The proposed system follows a multi-step approach, which has been shown to be a good strategy for CPG formalization [121]. We designed the system to set boundaries around each of the aspects of the CPG formalization, where each aspect is implemented as a separate autonomous component in a CPG formalization pipeline as illustrated in Figure 6.1.

The system is based on the Unstructured Information Management Architecture (UIMA) [51] and data interchange between text analysis components which is performed using the UIMA Common Analysis Structure (CAS) [58], a commonly shared data structure to represent the artifact as well as according metadata. In the following subsections, we provide a description of each component in our CPG formalization system.

6.2.1 XML parsing

We used CPGs extracted from the National Guideline Clearinghouse (NGC) [5] in XML format. The XML parsing component extracts the content of the XML CPG documents into a structured object. Although we extract the content of all the sections in the XML document, we only use the Major Recommendations section, which contains the diagnosis and/or treatment narrative text of the clinical knowledge that we try to identify and formalize.

6.2.2 Text cleansing

Most of the sections extracted by the XML parsing component contain narrative text mixed with HTML tags. HTML tags are used by Web browsers to render text for visual display, but as we are not interested in composing the text for web browsers, we removed all HTML tags from the text.

6.2.3 Medical Concept tagging

Medical Concept tagging is a component to map CPG text to a medical vocabulary; we used the Unified Medical Language System (UMLS) Metathesaurus as our biomedical vocabulary database; The UMLS Metathesaurus contains more than 2.6 million concepts each assigned to at least one semantic type from the set of the 133 semantic types of the UMLS semantic network. We used MetaMap [12, 113], to map CPG text to the UMLS Metathesaurus concepts; For integrating MetaMap with the UIMA framework we leveraged the MetaMap UIMA Annotator [116] which is a wrapper that makes the MetaMap tool usable as an UIMA analysis engine.

Due to a technical limitation in the UIMA Ruta [80] dealing with arrays (UIMA Ruta will be disccused in the following subsection), we modified the MetaMap UIMA Annotator to output multiple UMLS concept annotations for concepts associated with multiple semantic types instead of just one UMLS concept annotation with an array of semantic types. Figure 6.2 illustrates the change we applied. The “Candidate” data structure in Figure 6.2 is one of the MetaMap UIMA Annotator output objects; we defined a more

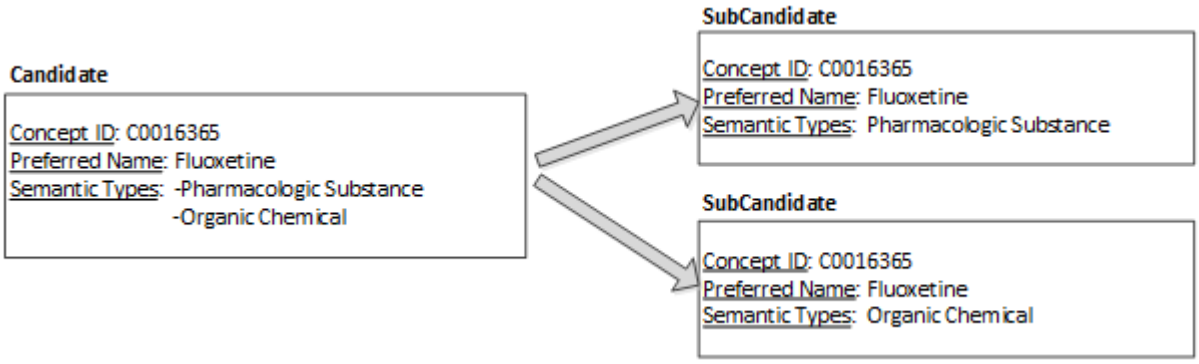


Figure 6.2: MetaMap UIMA Annotator output annotation "Candidate" for text Fluoxetine split into two separate "SubCandidate" annotations

flat data structure, the "SubCandidate" and modified the MetaMap UIMA Annotator to generate the described flat data structure instead of the hierarchical "Candidate" data structure.

6.2.4 Medical Tags Disambiguation

Medical Tags Disambiguation is the process of finding the correct UMLS concepts, when multiple concepts are assigned by MetaMap with the same score. For example the word lens could get annotated by MetaMap with three different UMLS concepts that have different meanings as shown in Table 4.2.

To solve this type of ambiguity we used the graph-based disambiguation algorithm [47] that we described in Section 4.4. The disambiguation algorithm ranks the generated MetaMap UMLS concepts based on their relatedness to the context of co-located text using the betweeness centrality metric [26].

6.2.5 Clinical recommendation pattern detection

Clinical recommendation pattern detection is a rule-based extraction component. This component is the first level of our clinical recommendation extraction mechanism; its function is to extract parts of CPG text that contain the minimum necessary features of the clinical recommendation in question. Extracting clinical recommendation based on

```
DECLARE Sentence;
(# PERIOD){-> Sentence};
PERIOD (# PERIOD){-> Sentence};

DECLARE CHEM;
DECLARE DISO;
SubCandidate{ SubCandidate.semanticType=="aapp" -> MARKONCE(CHEM)};
SubCandidate{ SubCandidate.semanticType=="acab" -> MARKONCE(DISO)};

DECLARE Annotation EVALUATION_problem_diagnosis
                   (DISO problem_diagnosisRef);
DECLARE Annotation INSTRUCTION_medication_order
                   (CHEM medicineRef);

Sentence{CONTAINS(DISO)) ->
            CREATE(EVALUATION_problem_diagnosis,
                  "problem_diagnosisRef" = DISO};

Sentence{CONTAINS(CHEM) ->
            CREATE(INSTRUCTION_medication_order,
                   "medicineRef" = CHEM};

DECLARE Annotation Diagnosis_Medication_Relation
                   (EVALUATION_problem_diagnosis diagnosisRef,
                    INSTRUCTION_medication_order medicationRef);

Sentence{CONTAINS(INSTRUCTION_medication_order),
         CONTAINS(EVALUATION_problem_diagnosis)  ->
            CREATE(Diagnosis_Medication_Relation,
                  "diagnosisRef" = EVALUATION_problem_diagnosis,
                  "medicationRef" = INSTRUCTION_medication_order)};
```

Step 1
Set text boundaries

Step 2
Cluster UMLS semantic types

Step 3
A) Clinical data structure

B) Conditions for clinical data structure

Step 4
A) Clinical context semantic relation

B) Conditions for clinical context

Figure 6.3: Ruta extraction rules

the minimum necessary features follow the top-down approach [119] where only general rules that cover as many possible instances of clinical recommendation need to be defined, which means rules that have high coverage and poor precision. Because general extraction rules tend to be in small numbers and simple to define, the rule authoring task is a good fit for the medical experts who usually lack extensive knowledge in rule authoring. To further simplify the rule authoring task for the medical expert we used UIMA Ruta [80] as it has a defined rule-based language with the ability to build rules against the text as well as against the semantic annotations of the text. We also defined a process of four steps to structure the effort required. The human modeller might need to do multiple iterations of these four steps to either increase the accuracy of rules or to author extraction rules for multiple clinical recommendations. In the following subsections we describe each of the four steps to author drug recommendation extraction rules with sample Ruta syntax highlighted in Figure 6.3.

6.2.5.1 Step 1) Set text analysis boundaries

While analyzing the CPG as one big unit is too complex, analyzing the CPG text as a bag of tokens is impractical; therefore, we need to break down the CPG document into text chunks small enough for easiness of analysis. Step 1 in Figure 6.3, shows the Ruta code to break CPG text into a set of sentences, where the first line defines a new type "Sentence" using the DECLARE keyword, the second and third lines define two rules to identify any set of tokens that either end with a period or are enclosed between two periods, and then annotate them with the new type "Sentence".

6.2.5.2 Step 2) Cluster UMLS semantic types

Each UMLS tag "SubCandidate" is assigned to one UMLS semantic type, and as we have a set of 133 semantic types in the UMLS semantic network, this gives us a wide spectrum of semantic types that is too granular for our analysis. Clustering the UMLS semantic network into smaller set of semantic types helps eliminate duplicate rules across UMLS semantic types. To achieve this goal we followed the approach presented in [22] and aggregated semantic types. In Table 6.1 and Table 6.2 we show two groups of semantic types, the Chemical & Drugs (CHEM) and the Disorder (DISO) used for drug recommendations.

Step 2 in Figure 6.3, shows the Ruta code to define two new types (CHEM and DISO), then scan all tokens in the CPG that are annotated with the SubCandidate type, and assign them one of the two new annotations based on their current UMLS semantic type. Similar rules apply to all elements in Table 6.1 and Table 6.2.

6.2.5.3 Step 3) Structuring clinical data

In this step, the human modeller A) defines the clinical data structures and B) provides conditions to assign the newly defined clinical data structure to tokens in the CPG text. Defining clinical data structures could be coarse, e.g. the drug prescription data structure composed of the medicine name, and the dose; or more granular to include the dose timing and the duration of the treatment. The expressivity of the clinical recommendation extraction rules heavily depends on the granularity of data structures used, the

Table 6.1: Chemical & Drugs semantic types group (CHEM)

Semantic Type	**Abbreviation**
Amino Acid, Peptide, or Protein	aapp
Antibiotic	antb
Biologically Active Substance	bacs
Biomedical or Dental Material	bodm
Carbohydrate	carb
Chemical	chem
Chemical Viewed Functionally	chvf
Chemical Viewed Structurally	chvs
Clinical Drug	clnd
Eicosanoid	eico
Element, Ion, or Isotope	elii
Enzyme	enzy
Hazardous or Poisonous Substance	hops
Hormone	horm
Immunologic Factor	imft
Indicator, Reagent, or Diagnostic Aid	irda
Inorganic Chemical	inch
Lipid	lipd
Neuroreactive Substance or Biogenic Amine	nsba
Nucleic Acid, Nucleoside, or Nucleotide	nnon
Organic Chemical	orch
Organophosphorus Compound	opco
Pharmacologic Substance	phsu
Receptor	rcpt
Steroid	strd
Vitamin	vita

Table 6.2: Disorders emantic types group (DISO)

Semantic Type	**Abbreviation**
Acquired Abnormality	acab
Anatomical Abnormality	anab
Cell or Molecular Dysfunction	comd
Congenital Abnormality	cgab
Disease or Syndrome	dsyn
Experimental Model of Disease	emod
Finding	fndg
Injury or Poisoning	inpo
Mental or Behavioral	mobd
Neoplastic Process	neop
Pathologic Function	patf
Sign or Symptom	sosy

more granular the clinical data structures the more expressive rules can be authored but also the more complex the rule authoring task becomes. Therefore a good balance needs to be achieved by defining the least granular level of clinical data structure that is sufficient for the required expressivity of the extraction rules. To follow pre-reviewed clinical data structures we defined our clinical data structures based on the openEHR archetypes [18]; openEHR archetypes are a set of common specialised clinical concepts in the form of structured constraint statements based on the openEHR Reference Model [18]. The openEHR archetypes have different types such as observations, evaluations, instructions and actions, which are adapted from the problem solving process model illustrated in Figure 6.4. The role of a health care practitioner in this model is to make observations, form opinions, and then prescribe instructions. Generally, a human modeller needs to specify at least two clinical data structures to formalize clinical recommendation knowledge, one to represent the input to the health practitioner and another to represent the output from the health practitioner.

Detecting an instance of the defined clinical data structure in the CPG text is achieved

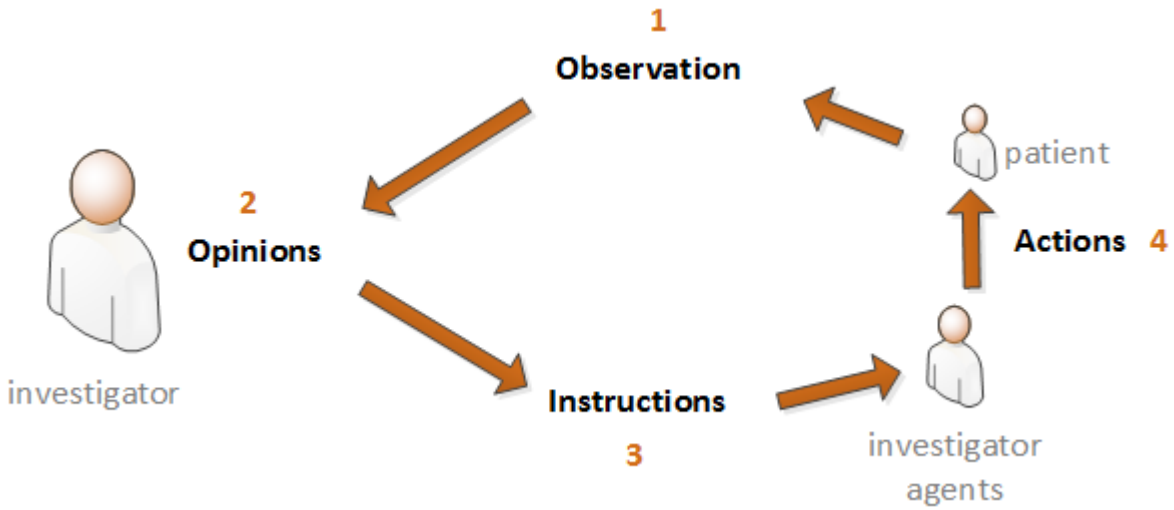


Figure 6.4: Relationship of information types to the investigation process, based on [19]

by annotating the CPG text with the clinical data structure types based on predefined conditions. The conditions could be based on specific lexicon, syntax or previously annotated semantics; Step 3 in Figure 6.3 shows the Ruta code of our version of the "problem diagnosis" evaluation archetype and the "medication order" instructions archetype. The defined clinical data structures contain one element for simplicity, but each of these data structures can contain multiple elements; we also defined relaxed conditions that capture tokens annotated with the DISO and CHEM semantic groups, and tagged the former as an element of the "problem diagnosis" evaluation archetype, and the latter as the medicine element of the "medication order" instructions archetype.

6.2.5.4 Step 4) Clinical recommendation semantic relations

Each clinical recommendation could be modelled as an instance of semantic relation between clinical data, for example drug recommendation could be modeled as a disease-to-drug semantic relation or symptoms-to-drug semantic relation. Annotating CPG text with clinical recommendation semantic relation requires the human modeller to 1) define a semantic relation, and 2) define conditions for mapping instances of clinical data structures to a semantic relation. Step 4 in Figure 6.3 shows the Ruta code to define a binary

relation between the "problem diagnosis" data structure, and the "medication order" data structure, we also defined relaxed condition that capture instances of "problem diagnosis" and "medication order" that are co-located in the same sentence, and then create an annotation of the "Diagnosis Medication Relation" semantic relation. The condition we used is relaxed for simplicity but medical expert can define more strict conditions for the lexical patterns of the text between "problem diagnosis" data structure, and the "medication order" data structure; a recent study [148] shows that drug-disease treatment pair can be identified with high precision using a set of lexical patterns such as in patients with, for treatment of, in the management of.

Clinical recommendation filtering, this component is responsible for removing the clinical recommendation instances wrongly labelled by the clinical recommendation pattern detection component. We used logistic regression [64] classification algorithm to decide on the correctness of the drug recommendation labels. Because this classification algorithm is supervised, which means it requires to be trained using a correctly annotated data set; we generated a training data set composed of 117 recommendation sentences extracted from the Yale Guideline Recommendation Corpus (YGRC) [68]. The YGRC is composed of 1275 recommendations which cover a broad range of diseases and mental disorders extracted from the NGC. We annotated all YGRC sentences with MetaMap and then selected 117 sentences that have tokens in the DISO semantic group in addition to other tokens in the Procedure (PROC) or CHEM semantic group. We manually tagged each sentence as either drug/procedure recommendation or non-drug/procedure recommendation.

Clinical recommendation mapping, a component to map instances of clinical recommendation semantic relations to their target CIG constructs as a set of rules. We used the openEHR Guideline Definition Language (GDL) [110] as our target CIG; GDL leverage the designs of openEHR Reference Model and Archetype Model that we used in the pattern detection step, therefore, the mapping is straightforward. In GDL, the Guide data structure is the parent data structure used to model a CPG, in our formalization process we are most interested to capture clinical knowledge which would be represented as a set of rules using the Rule data structure that is decomposed into main members: *whenStatments*, a list of expressions that represent the condition that must be evaluated

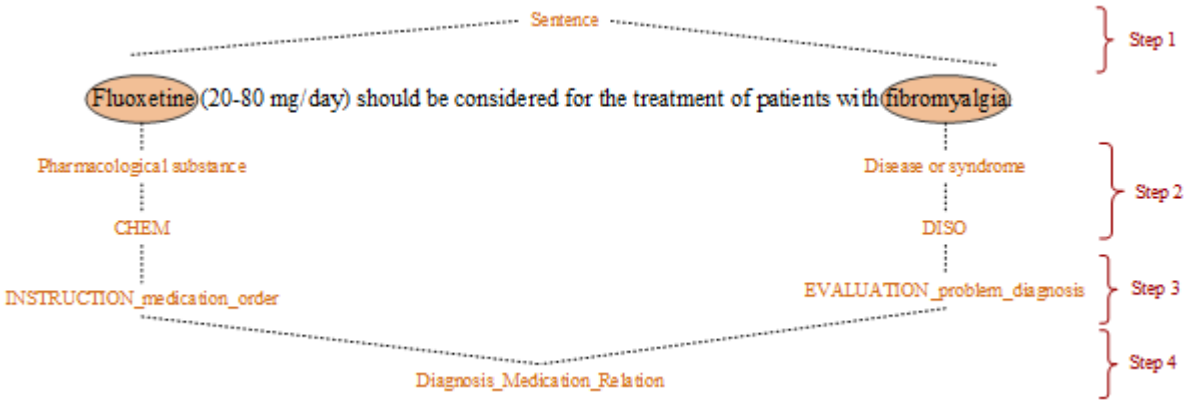


Figure 6.5: Annotations for the drug recommendation

before the rule get fired. *thenStatments*, a list of expressions that represent the action that must be execute if the condition was met. Although the clinical recommendation mapping is still a manual task to be done by the knowledge engineer, it can be fully automated if the medical expert modeller used a standard naming convention for the clinical data structures used in the clinical recommendation pattern detection component.

To demonstrates an example of drug recommendation CPG clinical sentence annotated using the 4-steps method detailed in the above sub-sections, we used the following sentence: *"Fluoxetine (20-80 mg/day) should be considered for the treatment of patients with fibromyalgia"* from the "Management of chronic pain. A national clinical guideline." CPG. And in Figure 6.5 [3], we show all annotations for this sentence based on the UIMA Ruta 4-steps drug recommendation pattern defined in Figure 6.3.

6.3 Results and Discussion

We implemented our formalization system in JAVA and integrated it with the GDL editor. Figure 6.6 shows the rule extracted from the example given in Figure 6.5 in the native GDL format, and Figure 6.7 shows the extracted rule as it display in the GDL editor; due to the lack of access to independent human modellers we could not measure the manual effort saving introduced by our system; nevertheless, we evaluated the accuracy of the drug recommendations knowledge extracted by our system.

Table 6.3: Recommendation sentences classification evaluation

Recommendation	**Precision**	**Sensitivity/Recall**	**Specificity**
Chemicals & Drugs	78%	71%	73%
Procedures	70%	75%	79%
Average	**74%**	**73%**	**76%**

To build our gold standard for the drug recommendation clinical context to measure against, we used all recommendations from the "Management of chronic pain" CPG [3] and the "Management of lung cancer" CPG [4], and then we manually tagged each recommendation as either drug/procedure recommendation or non-drug/procedure recommendation. The resulted test data set is composed of 169 recommendation sentences. Our evaluation was based on measuring the precision, sensitivity/recall and specificity of the extracted drug recommendation rules form the above CPG. The precision, sensitivity/recall and specificity are measured based on the correctness of our system in finding instances for the UIMA Ruta patterns defined by the medical expert. More formally, assume that I is the set of all sentences in a CPG, and I_G denotes the subset of I that contains sentences with both a medication and a disease; $I_{G'}$ denotes for all sentences in I that do not contain both a medication and a disease; I_F denotes the set of sentences extracted by our system; $I_{F'}$ denotes set of sentences not extracted by our system

- $Precision = \frac{|I_G \cap I_F|}{|I_F|} = \frac{TruePositives}{TruePositives+FalsePositives}$
- $Recall/Sensitivity = \frac{|I_G \cap I_F|}{|I_G|} = \frac{TruePositives}{TruePositives+FalseNegatives}$
- $Specificity = \frac{|I_{F'}|}{|I_{F'} \cup (I_F \cap I_{G'})|} = \frac{TrueNegatives}{FalsePositives+TrueNegatives}$

In Table 6.3 we show the accuracy of our framework on classifying the 169 recommendation sentences.

We evaluated the correctness of the formalized recommendations by manually checking the extracted rules and we assigned a coefficient of: 1 for rules that are correctly coded and complete, 0.5 for rules that are correct but partial (e.g. not all elements of the rule conditions are captured), 0 for rules that are wrong. In Table 6.4 we show the accuracy of the extracted rules based on the described metric. There are two main sources of errors

Table 6.4: Recommendation sentences extracted rules accuracy

Recommendation	**Accuracy**
Chemicals & Drugs	87%
Procedures	81%
Average	**84%**

for the wrong rules: either wrong MetaMap annotations or wrong classification from our clinical context filtering component using logistic regression.

The precision is impacted by the size and the quality of our training data set; in the presented example we used a training data set made of 169 sentences which is small to provide high precision. This issue could be lessened by feeding the outputted rules of the system back to the training data set, a step that requires a minor manual tagging of which rule are correctly extracted and which ones are wrongly extracted. The sensitivity/recall is impacted by how we split our CPG into smaller text chunks, e.g. in the presented example we split the CPG into sentences, but some drug recommendations within the CPG have the drug and the medication located in two separate sentences, and therefore, these ones are missed by our extraction rules. This issue could be lessened by changing the size of our unit of analysis from one sentence to two consecutive sentences or to the whole paragraph, but such a modification would hurt the precision unless we add more rules to handle cross sentences extraction. Different cross sentences extraction approaches can be applied. One approach would be to perform cross sentence extraction when a sentence only contains one part of the clinical recommendation such as a sentence with only a disease, followed by a sentence that only contains the other part of the clinical recommendation such as a sentence with only a medication. This approach is very conservative and would not impact the precision of the in-sentence extraction rule; Incorporating other cross sentences extraction approaches that have more coverage would likely interfere with other in-sentence extraction rules. Therefore, with every cross sentence extraction approach we need to evaluate the cross sentence extraction precision gain to the in-sentence precision loss. The specificity is impacted by how strict are our conditions for tagging a sentence with a specific semantic relation. In the presented example we achieved high specificity

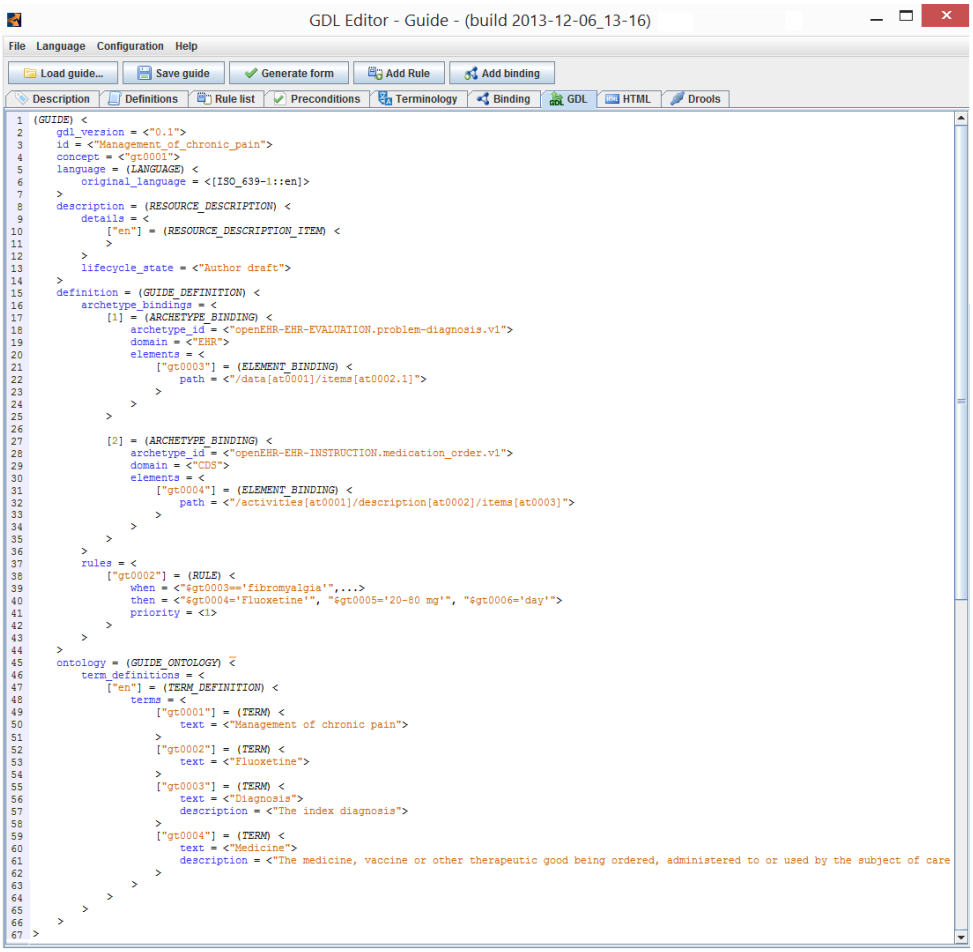


Figure 6.6: Drug recommendation extracted GDL rule

because our conditions for tagging drug medication semantic relations are very strict.

We found that different clinical recommendations of CPG do not only necessitate different types of text analysis, but could also require different target knowledge representations, e.g. a knowledge representation using rules works for the medication recommendations of CPG, but it would not be the best representation for clinical workflow. Therefore, we had to consider specific quality attributes when constructing our formalization system to be generic and yet easily reusable by a human modeller. We found that there are three quality attributes intrinsic to achieving our goal:

Clinical Interoperability: The ability to consistently express clinical meanings within electronic health record (EHR) systems and medical knowledge repositories. The Clinical

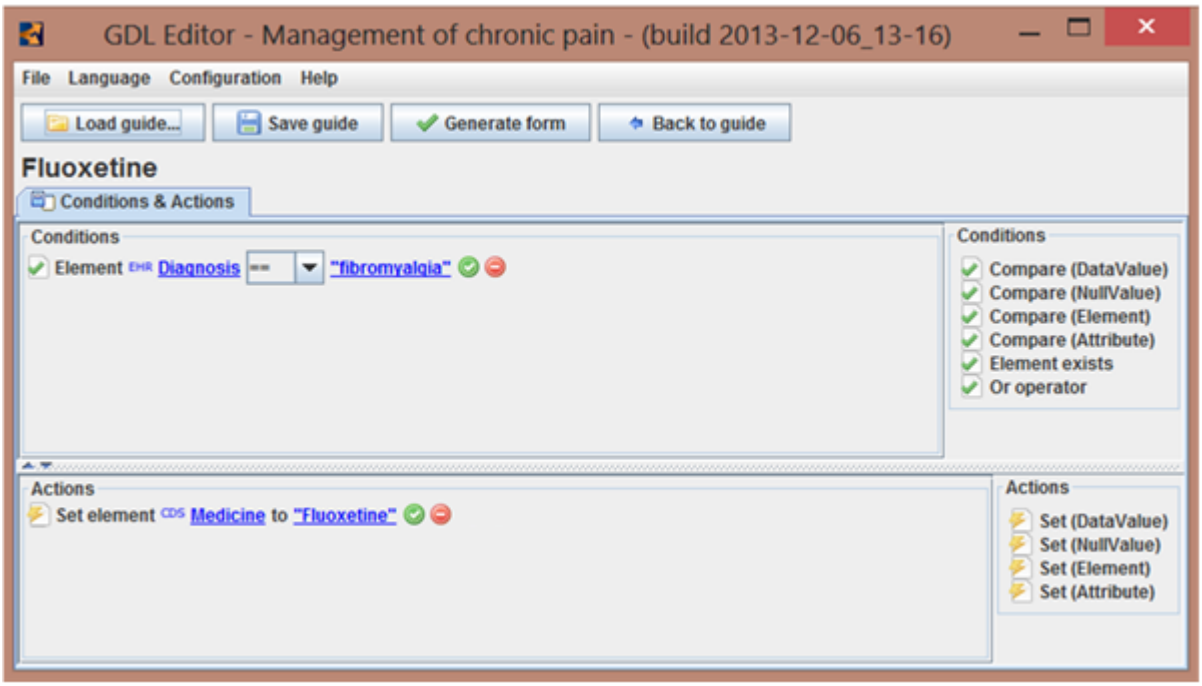


Figure 6.7: Drug recommendation rule in GDL editor

Interoperability could be measured using the four levels defined in the SemanticHEALTH report [131], where the lowest level of interoperability means no interoperability at all, and the highest level of interoperability means full semantic interoperability, shareable context, seamless co-operability. Clinical intolerability is manifested in our system by the adoption of: (1) openEHR Reference Model, a generic reference models for representing clinical data; (2) openEHR archetypes, an agreed clinical data structure definitions; (3) UMLS metathesaurus, a clinical terminology systems.

system Extensibility: The ability to allow system extensions without a major source code rewrite; Extensibility is particularly needed when new text analysis engines or new extraction rules to be added. Extensibility is manifested in our system by the adoption of: (1) Apache UIMA Java framework, a common and extensible means for representing unstructured information; (2) UIMA Ruta, a rule language to author clinical recommendation detection patterns.

system Integrability: The ease with which separately developed components, can be made to work together [87]. As the output of our formalization system is a CIG that

would need incremental refinement by human modeller, we implemented our system to integrate easily with CIG editors. Integrability is manifested in our system by the plug-in implementation mode that allows it to be hosted within a CIG editor application with minimal programing effort.

6.4 Summary

The proposed system can be effectively used to formalize the drug recommendation and procedure recommendation clinical contexts of CPGs into CDSS friendly format. More significantly, it provides human modellers a process to extend the system to formalize other clinical recommendations of CPGs. The system is focused on automating the CPG common formalization steps while allowing the human modeller to stay in control of all the knowledge extraction steps. By configuring the clinical recommendation detection pattern component, the human modeller could define how CPG text would be split into smaller chunks for analysis, and control the level of granularity of the clinical knowledge extracted. This control balances generality and specificity in order to maximize usefulness of the extracted knowledge. If the extracted knowledge is too specific/expressive it unnecessarily complicates the extraction rules. We believe that such configuration capabilities in our system would help reduce the human modeller's annoyance and dissatisfaction accompanied with either the lengthy manual CPG formalization steps, or the inflexibility of other automated CPG formalization approaches. The proposed system can be extended to find contradictions that exist between different CPGs by comparing the extracted rules in multiple CIGs and highlighting the rules that have the same conditions but with opposite or different actions.

Chapter 7

The CPG formalization system Scalability

7.1 Introduction

One of the important factors for human modellers to adopt a CPG formalization system such the one we proposed in Chapter 6 is the system's ability to scale. In this chapter we focus of two different type of scalabilities: 1) scaling out, which is the ability of the CPG formalization system to adapt new clinical recommendation type 2) scaling up, which is the effectiveness of CPG formalization system when applied on heterogeneous CPGs. The focus of this chapter is to on scaling up the system.

7.2 Methods

7.2.1 Scaling out

Scaling the system to adapt new type of clinical recommendations can be achieved by introducing new pattern as explained in Section 6.2.5, in which we used a pattern composed of two UMLS clusters, the *Chemical & Drugs* (CHEM) and the *Disorder* (DISO) for detecting drug recommendations. In [94] the authors depicted some relations between UMLS groups, which we present in Table 7.1.

Table 7.1: Some relations between Semantic Groups [94]

Semantic Group	**Relation**	**Semantic Group**
Chemicals & Drugs	treats	Disorders
Procedures	treats	Disorders
Devices	treats	Disorders
Genes & Molecular Sequences	carries out	Physiology
Genes & Molecular Sequences	property of	Chemicals & Drugs

The relations between UMLS groups could be used as a basis for new clinical recommendation patterns. For example human modellers could define a pattern for the procedure recommendation based on the co-location of the *Procedures* cluster (PROC) and the *Disorder* cluster (DISO). With every new clinical recommendation introduced, the classifier of *Clinical recommendation filtering* component which is responsible for removing the clinical recommendation instances wrongly extracted by the *Clinical recommendation pattern detection* component need to be retrained.

7.2.2 Scaling up

Scaling the system to formalize more heterogeneous CPGs with the same effectiveness can be achieved by refining one or both of the following two components:

- Refine the extraction patterns of the *Clinical recommendation pattern detection* component to be more constrained or more flexible.
- Train the classifier used in the *Clinical recommendation filtering* component with more heterogeneous instances of the extracted clinical recommendation.

We believe that investing human modeller effort in refining extraction patterns is more valuable than finding more instances to train the classifiers of the *Clinical recommendation filtering* component. Therefore we present a scalability approach that support human modellers to refine extraction pattern. Our scalability approach is based on Action Palettes [49] which we present in the next subsection.

7.2.2.1 Clinical Action Palettes

Action Palettes [49] is a set of action types that comprehensively categorize activities recommended by the majority of clinical guidelines. The set of these action types are extracted from a pool of randomly selected 100 recommendations (test and validation sets) each from the National Guideline Clearinghouse, then three recommendations from each guideline are randomly selected. The study [49] resulted in the creation of a library of 300 randomly selected clinical recommendations with 405 actions, and the following 12 action palettes were found sufficient to categorize all the 405 actions.

1. **Prescribe**: Order a treatment requiring medication or durable medical equipment.
2. **Perform therapeutic procedure**: Order activities that are therapeutic in nature.
3. **Educate/Counsel**: Inform the patient about means to improve/maintain health, or instruct on how to perform specific activities.
4. **Test**: Obtain or collect additional data through inquiry (ask patient), laboratory testing (chemistry panel, X-Rays, etc. . .) or other investigative procedures whose intent is not curative.
5. **Dispose**: Initiate an activity to direct the flow of patients, such as Admit, Discharge, Follow-up, Transfer, etc.
6. **Refer/Consult**: Direct a patient to another clinician for evaluation and/or treatment.
7. **Conclude**: Determine a diagnosis or clinical status
8. **Monitor**: Make serial observations according to specific criteria and schedule.
9. **Document**: Record one or more facts in the patient record. Document includes situations in which a document (such as a medical report) is to be forwarded to legal authorities or guardians of a minor child to inform or report a condition.
10. **Advocate**: Argue in support of a policy.

11. **Prepare**: Make ready for a particular guideline directed activity by training, equipping, or gaining new knowledge (e.g., through research).

12. **No recommendation**: A statement that no activity is advised, usually because of insufficient scientific evidence for or against the activity.

The authors of the study [49] showed that these action types can be used to construct a system for design of clinical decision support systems. In a more recent study [126], the authors extended Action Palettes by building a set of verbs that could be used to represent each action type in CPGs. Such mapping was done by classifying more than 700 recommendations from the Yale Guideline Recommendation Corpus (YGRC) to action types and extracted the verbs associated with each. The YGRC will be detailed in Section 7.3. From the list of extracted verbs, the authors identified a list of transitive verbs. Transitive verbs take a direct object to describe an action that is done to something or someone and to link the action taken with the object upon which that action is taken. A total of 279 verbs pertinent to the 14 action types was categorized and incorporated. The list below shows the mapping of verbs to their action types, and Figure 7.1 shows the grouping of these 14 action types.

1. **Inquire**: ask, assess, complete, conduct, gather, include, incorporate, inquire, obtain, review, screen, verify.

2. **Examine**: assess, auscultate, examine, include, inspect, palpate, percuss, perform, use

3. **Test**: assess, begin, carry out, check, conduct, continue, determine, do, evaluate, have, identify, indicate, measure, need, obtain, offer, perform, prefer, receive, recommend, repeat, require, reserve, restore, screen, take, test, trigger, undergo, use, utilize

4. **Monitor**: arrange, ascertain, assess, check, conduct, continue, determine, evaluate, examine, follow up, have, include, institute, maintain, manage, monitor, obtain, occur, offer, perform, provide, reassess, receive, recommend, repeat, require, review, screen, warrant

5. **Conclude**: assess, base, conclude, consider, contact, coordinate, determine, diagnose, distinguish, exclude, give (attention), recognize, recommend, respect, review, suspect, take (into account), use, weigh

6. **Advocate**: advocate, encourage, endorse, ensure, focus, recommend, work (to)

7. **Dispose**: admit, dispose, hospitalize, guide, observe, refer

8. **Document**: complete, document, identify, notate

9. **Educate/Counsel**: adhere, advise, benefit, clarify, counsel, deliver, discuss, educate, enable, encourage, explain, have, help, identify, include, incorporate, inform, instruct, involve, modify, negotiate, offer, promote, protect, provide, receive, recommend, reinforce, review, start, support, teach, tell, use

10. **Perform**: confine, ensure, follow, give, implement, include, incorporate, indicate, inspect, offer, operate, perform, place, receive, recommend, relate, resect, reserve, select, start, treat, undergo, use

11. **Prepare**: address, adhere, adjust, adopt, analyze, attempt, be (aware), become, begin, collect, continue, dedicate, define, develop, encourage, engage, ensure, establish, form, have, identify, include, incoporate, initiate, institute, know, lead, perform, plan, prepare, recommend, review, share, train, understand, undertake, use

12. **Prescribe**: add, adjust, administer, advance, apply, attempt, avoid, change, choose, continue, desensitize, dilute, discontinue, exercise, improve, increase, indicate, individualize, influence, initiate, institute, manage, offer, order, prefer, prescribe, provide, receive, recommend, reduce, repeat, replace, reserve, restart, review, start, suggest, supplement, taper, titrate, treat, use, utilize, warrant

13. **Prevent**: administer, avoid, cleanse, combine, continue, discard, encourage, give, immunize, minimize, practice, prevent, provide, receive, recommend, use

14. **Refer/Consult**: assess, conduct, consult, manage, obtain, offer, recommend, refer, seek, work (together)

We integrated all the verbs that belongs to the *Prescribe* action type in the *Clinical recommendation pattern detection* component, by adding a new condition to the clinical recommendation semantic relation (Section 6.2.5.4) to ensure that one of the 44 verbs of the *Prescribe* action type is co-located to “problem diagnosis” and “medication order”.

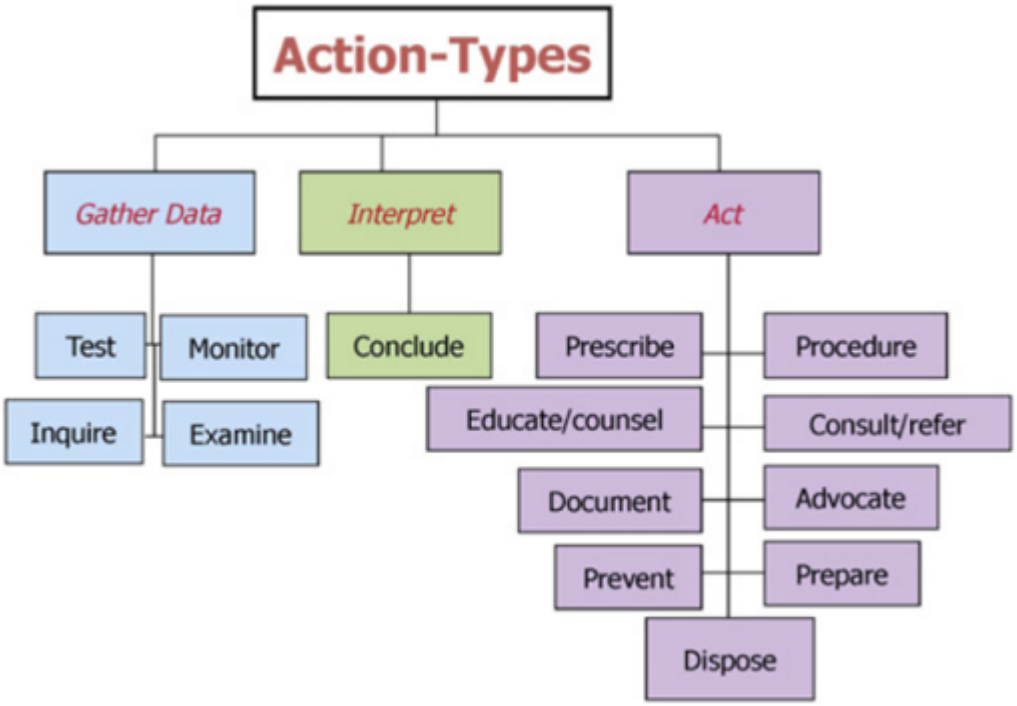


Figure 7.1: Action types adapted from [126]

7.3 Results & Discussion

We evaluated our approach with against the Yale Guideline Recommendation Corpus (YGRC) [68]. The YGRC is composed of 1275 recommendations which cover a broad range of diseases and mental disorders extracted from the NGC. The recommendation as defined by the authors in [68]: "a statement whose apparent intent is to provide guidance about the advisability of a clinical action".

Recommendations in YGRC were identified based on: (1) semantic indicator, (2) formatting, (3) headers and (4) presence of recommendation strength indicators. The following subsection describe the four criteria:

1. **Semantic indicators,** YGRC defined several semantic indicators to recognize recommendations. These indicators include:

   (a) Modal operators (e.g., terms such as "should," "must," "may") to express a level of obligation or permission or

   (b) Statements of suitability under specific circumstances (e.g., "is appropriate", "is indicated").

   - Example: An F 18-deoxyglucose positron emission tomography (FDG_PET) scan should be performed to investigate solitary pulmonary nodules in cases where a biopsy is not possible or has failed, depending on nodule size, position and CT characterization.

2. **Formatting,** YGRC defined several formatting indicators such as:

   (a) Enumeration of statements.

   (b) Boldface text.

   (c) Bulleted text.

3. **Headers,** YGRC used indicative headings and titles, such as 'Recommendations' and 'Recommended' to demarcate recommendation statements.

- Example: Recommendation: Treat duodenal ulcers with H2RAs or PPIs for 4 to 8 weeks.

4. **Presence of recommendation strength indicators.** Recommendation statements may be accompanied by an indicator of evidence quality or strength of recommendation. Following is an example from the YGRC in which strength of recommendation and quality of evidence is indicated:

    - Example: Rituximab is active in the treatment of Wm but associated with the risk of transient exacerbations of clinical effects of the disease and should only be used with caution, especially in patients with symptoms of hyper-viscosity and/or IgM levels >40 g/L. Level of evidence IIb, Grade of Recommendation B.

We achieved a precision of 70.1%, and a recall of 72.3%. In our evaluation we only selected the sentences that are drug recommendation from the YGRC and we excluded all the medical test recommendations or procedure recommendations.

7.4 Summary

Scalability is an important factor for human modellers to adopt a CPG formalization system, in this chapter we discussed how the CPG formalization system presented in Chapter 6 can scale up. We used a set of 44 transitive verbs that have been shown to be associated to the *Prescribe* action type and integrated them in the the *Clinical recommendation pattern detection* component of the CPG formalization system. We evaluated the system against the Yale Guideline Recommendation Corpus (YGRC) and achieved a precision of 70.1%, and a recall of 72.3%.

Chapter 8

Discussion and conclusion

We have presented an approach to formalize CPGs. In this chapter, the main contributions of this thesis are outlined and a number of directions for future work are presented.

8.1 Summary of Contributions

Recall in Chapter 1, the objectives of the work was to minimize the effort required by human modellers to bridge the gap between CPG and its formalized version by extending the automation of the formalization process. The overall purpose is decomposed into the intermediate goals of this work. The goals of this work are to:

1. Automatically disambiguate the narrative text of CPGs using medical knowledge bases and graph-based algorithms.

2. Develop a system upon the algorithms resulted from our first goal to transform CPGs into CIGs using a multi-step approach.

3. Allow human modellers to refine and add types of clinical recommendations without rebuilding the system.

We discuss our contributions according to these objectives. In this thesis, the overall problem of formalizing CPGs is approached with focus on 1) developing a set of autonomous components that can be developed and maintained independently and 2) give the human

modeller a control over the expressiveness of the extraction rules in the formalization system without rebuilding the system . The main contribution of this work are:

- A Graph-based Disambiguation approach using the UMLS semantic network (Section 4.4)[46], we proposed an algorithm based on the hypothesis that words closely located to each other in a text must have some degree of relatedness. We used the UMLS semantic network as our knowledge base to find the relatedness between words. For an ambiguous term, we take the neighbouring words before and after in a given window and check their respective semantic types using MetaMap. We select the semantic type of the one which has the smallest distance from the set of neighbouring word semantic types based on UMLS semantic network. We evaluated our method using the MSH-WSD [72] dataset containing 203 ambiguous words.We ran our algorithm on the MSH-WSD dataset with a window of size 3 and the resulting average accuracy was 60.3%.

- A Graph-based Disambiguation approach using the UMLS Metathesaurus (Section 4.5)[47], The approach presented in chapter 4 has the advantage of using a lightweight knowledge base which is the UMLS semantic network. Lightweight knowledge bases such as the UMLS semantic network give the human modeller the easiness of refining the disambiguation process by adding or removing UMLS semantic types and/or semantic relations. However, the main limitation of leveraging the UMLS semantic network as a knowledge base is the inability to disambiguate between two words that belong to the same semantic type. Because of this limitation we proposed another algorithm that uses the UMLS Metathesaurus[67] as its knowledge base, the proposed algorithm is inspired by the approach presented in [100]. We evaluated our method using the MSH-WSD [72] dataset containing 203 ambiguous words. We ran our algorithm on the MSH-WSD dataset with a window of size 2 and the resulting average accuracy was 59.2%.

- An analysis on the impact of using different UMLS subsets as a knowledge source on the unsupervised type of WSD algorithms (Section 4.5)[45], we analyzed how WSD accuracy is impacted by the different subsets of the UMLS Metathesaurus.

For the purpose of our analysis we implemented a graph-based unsupervised WSD algorithm that computes the importance of each node "sense" in the graph using the PageRank metric [27]. The way we split the UMLS Metathesaurus into smaller knowledge bases is by the different relations defined in the MRREL table, so each subset contains all the UMLS concepts but with only specific types of relations interconnecting them. We created four Metathesaurus subsets:

- PAR/CHD, a subset that contains only the parent and child relations;
- RB/RN, a subset that contains only the broader and the narrower relations;
- SIB, a subset that contains only the sibling relation;
- RO, a subset that contains only the other relation.

We executed our algorithm against the MSH WSD test data set, with a window size of 2, and we executed the algorithm using the 4 subsets of the MRREL table (PAR/CHD, RB/RN, RO, SIB). For each run we captured the accuracy for all terms/acronyms of the MSH-WSD data set. From our observation of the resulted accuracy of the 4 UMLS subsets, there is no real winner; each UMLS relation excels in disambiguating some terms/acronyms, and this indicates that using all relations of the UMLS MRREL table is not necessarily the best approach.

- A CPG formalization system (chapter 5-6), we implemented a multi-step CPG formalization approach, in which we designed the system to set boundaries around each of the aspects of the CPG formalization. Each aspect is implemented as a separate autonomous component in a CPG formalization pipeline. The disambiguation algorithm is integrated into the CPG formalization pipeline as an autonomous component. The system is based on the Unstructured Information Management Architecture (UIMA) [51] and data interchange between text analysis components which is performed using the UIMA Common Analysis Structure (CAS)[58]. The core of the system is the clinical recommendation pattern detection component which is a rule-based information extraction component. This component is the first level of our clinical recommendation extraction mechanism; its function is to extract text

fragments that contain the minimum necessary features of the clinical recommendation type in question. Extracting clinical recommendation based on the minimum necessary features follow the top-down approach [119] where only general rules that cover as many possible instances of clinical recommendation need to be defined, which means rules that have high coverage and poor precision. Because general extraction rules tend to be in small numbers and simple to define, the rule authoring task is a good fit for the medical experts who usually lack extensive knowledge in rule authoring. To further simplify the rule authoring task for the medical expert we used UIMA Ruta [80] as it has a defined rule-based language with the ability to build rules against the text as well as against the semantic annotations of the text. We also defined a process of four steps to structure the effort required. The human modeller might need to do multiple iterations of these four steps to either increase the accuracy of rules or to author extraction rules for multiple clinical recommendations.We implemented the proposed formalization system in JAVA and integrated it with the GDL editor [110]. we evaluated the accuracy of the drug recommendation knowledge extracted by our system. To build our gold standard for the drug recommendations to measure against, we used all sentences from the "Management of chronic pain. A national clinical guideline" CPG that contain medication and a disease, then we manually selected the sentences that are medication recommendation. Our evaluation was based on measuring the precision, sensitivity/recall and specificity of the extracted drug recommendation rules form the above CPG. The precision, sensitivity/recall and specificity are measured based on the correctness of our system in finding instances for the UIMA Ruta patterns defined by the medical expert. We achieved a precision of 72.5%, a sensitivity/recall of 76.8% and specificity of 97%.

8.2 Limitation and Future Directions

The following sections outline directions of future work related to the results presented in this thesis that were not in the scope of this work.

- Find the optimal window size for the Word sense disambiguation components:

  Investigating different approaches of using dynamic windows size for algorithms presented in Chapter Section 4.3 and 4.4 and measure how it impact the disambiguation accuracy in order to find the an optimal window size.

- Increase the precision of the CPG formalization system:

  The precision is impacted by the size and the quality of our training data set; in the presented example we used a training data set made of 126 sentences which is small to provide high precision. This issue could be lessened by feeding the outputted rules of the system back to the training data set, a step that requires a minor manual tagging of which rule are correctly extracted and which ones are wrongly extracted.

- Increase the sensitivity/recall of the CPG formalization system:

  The sensitivity/recall is impacted by how we split our CPG into smaller text chunks, e.g. in the presented example we split CPG into sentences, but some drug recommendations within the CPG have the drug and the medication located in two separate sentences, and therefore, these ones are missed by our extraction rules. This issue could be lessened by changing the size of our unit of analysis from one sentence to two consecutive sentences or to the whole paragraph, but such a modification would hurt the precision unless we add more rules to handle cross sentences extraction. Different cross sentences extraction approaches can be applied. One approach would be to perform cross sentence extraction when a sentence only contains one part of the clinical recommendation such as a sentence with only a disease, followed by a sentence that only contains the other part of the clinical recommendation such as a sentence with only a medication. This approach is very conservative and would not impact the precision of the in-sentence extraction rule; Incorporating other cross sentences extraction approaches that have more coverage would likely interfere with other in-sentence extraction rules. Therefore, with every cross sentence extraction approach we need to evaluate the cross sentence extraction precision gain to the in-sentence precision loss.

Bibliography

[1] Calcium supplementation in pregnant women. http://apps.who.int/iris/bitstream/10665/85120/1/9789241505376_eng.pdf, February 2015.

[2] First-trimester abortion in women with medical conditions. http://www.societyfp.org/_documents/resources/guidelines2012-2.pdf, February 2015.

[3] Management of chronic pain. A national clinical guideline. www.sign.ac.uk/pdf/SIGN136.pdf, February 2015.

[4] Management of lung cancer. www.sign.ac.uk/pdf/SIGN137.pdf, October 2015.

[5] National Guideline Clearinghouse. www.guideline.gov, February 2015.

[6] The openEHR Clinical Knowledge Manager (CKM). http://www.openehr.org/ckm/, July 2015.

[7] Eneko Agirre, Oier Lopez De Lacalle, Bernardo Magnini, et al. SemEval-2007 task 01: evaluating WSD on cross-language information retrieval. In *Advances in Multilingual and Multimodal Information Retrieval*, pages 908–917. Springer, 2008.

[8] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics, 2009.

[9] Eneko Agirre, Aitor Soroa, and Mark Stevenson. Graph-based Word Sense Disambiguation of biomedical documents. *Bioinformatics*, 26(22):2889–2896, 2010.

[10] Dimitra Alexopoulou, Bill Andreopoulos, Heiko Dietze, et al. Biomedical word sense disambiguation with ontologies and metadata: automation meets accuracy. *BMC bioinformatics*, 10(1):28, 2009.

[11] Luca Anselma, Paolo Terenziani, Stefania Montani, and Alessio Bottrighi. Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. *Artificial intelligence in medicine*, 38(2):171–195, 2006.

[12] Alan R Aronson. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association, 2001.

[13] Alan R Aronson and François-Michel Lang. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[14] Martin Atzmueller, Peter Kluegl, and Frank Puppe. Rule-Based Information Extraction for Structured Data Acquisition using TextMarker. In *LWA*, pages 1–7, 2008.

[15] Satanjeev Banerjee and Ted Pedersen. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Computational linguistics and intelligent text processing*, pages 136–145. Springer, 2002.

[16] Mathias Bank and Martin Schierle. A Survey of Text Mining Architectures and the UIMA Standard. In *LREC*, pages 3479–3486, 2012.

[17] Yehoshua Bar-Hillel. The present status of automatic translation of languages. *Advances in computers*, 1:91–163, 1960.

[18] T Beale, S Heard, D Kalra, and D Lloyd. The openEHR reference model: EHR information model. *The OpenEHR Foundation*, 1(2), 2008.

[19] Thomas Beale. Archetypes: Constraint-based domain models for future-proof information systems. In *OOPSLA 2002 workshop on behavioural semantics*, volume 105, 2002.

[20] Thomas Beale, Sam Heard, D Kalra, and D Lloyd. OpenEHR architecture overview. *The OpenEHR Foundation*, 2006.

[21] Thomas Bodenheimer. The American Health Care System-The Movement for Improved Quality in Health Care, 340 New Eng. *J. Med*, 488, 1999.

[22] Olivier Bodenreider and Alexa T McCray. Exploring semantic groups through visual approaches. *Journal of Biomedical Informatics*, 36:414–432, 2003.

[23] Branimir Boguraev and Mary Neff. A framework for traversing dense annotation lattices. *Language Resources and Evaluation*, 44(3):183–203, 2010.

[24] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[25] Aziz A Boxwala, Mor Peleg, Samson Tu, et al. GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. *Journal of biomedical informatics*, 37(3):147–161, 2004.

[26] Ulrik Brandes. A faster algorithm for betweenness centrality*. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[27] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[28] Rebecca F Bruce and Janyce M Wiebe. Decomposable modeling in natural language processing. *Computational Linguistics*, 25(2):195–207, 1999.

[29] Razvan Bunescu, Ruifang Ge, Rohit J Kate, et al. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial intelligence in medicine*, 33(2):139–155, 2005.

[30] M Butzlaff, HC Vollmar, B Floer, et al. Learning with computerized guidelines in general practice? A randomized controlled trial. *Family Practice*, 21(2):183–188, 2004.

[31] Mary Elaine Califf and Raymond J Mooney. Bottom-up relational learning of pattern matching rules for information extraction. *The Journal of Machine Learning Research*, 4:177–210, 2003.

[32] Angel X Chang and Christopher D Manning. TokensRegex: Defining cascaded regular expressions over tokens. Technical report, Technical Report CSTR 2014-02, Department of Computer Science, Stanford University, 2014.

[33] Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, et al. SystemT: an algebraic approach to declarative information extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 128–137. Association for Computational Linguistics, 2010.

[34] Laura Chiticariu, Yunyao Li, and Frederick R Reiss. Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems! In *EMNLP*, pages 827–832, 2013.

[35] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics, 2005.

[36] Paolo Ciccarese, Ezio Caffi, Silvana Quaglini, and Mario Stefanelli. Architectures and tools for innovative health information systems: the guide project. *International journal of medical informatics*, 74(7):553–562, 2005.

[37] Shlomi Codish and Richard N Shiffman. A model of ambiguity and vagueness in clinical practice guideline recommendations. In *AMIA Annual Symposium Proceedings*, volume 2005, page 146. American Medical Informatics Association, 2005.

[38] D Alan Cruse. *Lexical semantics*. Cambridge University Press, 1986.

[39] Chad Cumby and Dan Roth. Feature extraction languages for propositionalized

relational learning. In *Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data*, pages 24–31, 2003.

[40] Hamish Cunningham, Diana Maynard, and Kalina Bontcheva. *Text processing with gate*. Gateway Press CA, 2011.

[41] Hamish Cunningham, Diana Maynard, and Valentin Tablan. JAPE: a Java annotation patterns engine. 1999.

[42] David A Davis, Mary Ann Thomson, Andrew D Oxman, and R Brian Haynes. Evidence for the effectiveness of CME: a review of 50 randomized controlled trials. *Jama*, 268(9):1111–1117, 1992.

[43] Paul A De Clercq, Johannes A Blom, Hendrikus HM Korsten, and Arie Hasman. Approaches for creating computer-interpretable guidelines that facilitate decision support. *Artificial intelligence in medicine*, 31(1):1–27, 2004.

[44] Philip Edmonds and Scott Cotton. SENSEVAL-2: overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5. Association for Computational Linguistics, 2001.

[45] Wessam Gad El-Rab, Osmar R Zaïane, and Mohammad El-Hajj. Analyzing the Impact of UMLS Relations on Word-sense Disambiguation Accuracy. *Procedia Computer Science*, 21:295–301, 2013.

[46] Wessam Gad El-Rab, Osmar R Zaïane, and Mohammad El-Hajj. Biomedical text disambiguation using UMLS. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 943–947. ACM, 2013.

[47] Wessam Gad El-Rab, Osmar R Zaïane, and Mohammad El-Hajj. Unsupervised Graph-based Word Sense Disambiguation of Biomedical Documents. In *e-Health Networking, Applications Services (Healthcom), 2013 IEEE 15th International Conference on*, pages 649–652. IEEE, 2013.

[48] Gerard Escudero, Lluís Màrquez, and German Rigau. Naive Bayes and exemplar-based approaches to word sense disambiguation revisited. 2000.

[49] Abdelwaheb Essaihi, George Michel, and Richard N Shiffman. Comprehensive categorization of guideline recommendations: creating an action palette for implementers. In *AMIA Annual Symposium Proceedings*, volume 2003, page 220. American Medical Informatics Association, 2003.

[50] Ronen Feldman, Benjamin Rosenfeld, and Moshe Fresko. TEG - a hybrid approach to information extraction. *Knowledge and Information Systems*, 9(1):1–18, 2006.

[51] David Ferrucci and Adam Lally. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.

[52] Marilyn J Field, Kathleen N Lohr, et al. *Clinical Practice Guidelines:: Directions for a New Program*, volume 90. National Academies Press, 1990.

[53] John Fox, Nicky Johns, Colin Lyons, et al. PROforma: a general technology for clinical decision support systems. *Computer methods and programs in biomedicine*, 54(1):59–67, 1997.

[54] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[55] C Fry. Closing the gap between analytics and action. *INFORMS Analytics Mag*, 4(6):4–5, 2011.

[56] William A Gale, Kenneth W Church, and David Yarowsky. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5-6):415–439, 1992.

[57] William A Gale, Kenneth W Church, and David Yarowsky. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics, 1992.

[58] T Gotz and Oliver Suhre. Design and implementation of the UIMA Common Analysis System. *IBM Systems Journal*, 43(3):476–489, 2004.

[59] Rick Goud, Nicolette F de Keizer, Gerben ter Riet, et al. Effect of guideline based computerised decision support on decision making of multidisciplinary teams: cluster randomised trial in cardiac rehabilitation. *BMJ: British Medical Journal*, 338, 2009.

[60] Jeremy Grimshaw, RE Thomas, G MacLennan, et al. Effectiveness and efficiency of guideline dissemination and implementation strategies. 2004.

[61] Richard Grol and Jeremy Grimshaw. From best evidence to best practice: effective implementation of change in patients' care. *The lancet*, 362(9391):1225–1230, 2003.

[62] Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. *Cohesion in english*. Routledge, 2014.

[63] Nicolas Hernandez. Tackling interoperability issues within UIMA work flows. In *Language Resources and Evaluation (LREC'12)*, pages 3618–3625. European Language Resources Association (ELRA), 2012.

[64] David W Hosmer Jr and Stanley Lemeshow. *Applied logistic regression*. John Wiley & Sons, 2004.

[65] George Hripcsak, Paul D Clayton, T Allan Pryor, et al. The Arden Syntax for Medical Logic Modules. In *Proceedings/the... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*, pages 200–204. American Medical Informatics Association, 1990.

[66] Susanne M Humphrey, Willie J Rogers, Halil Kilicoglu, Dina Demner-Fushman, and Thomas C Rindflesch. Word sense disambiguation by selecting the best semantic type based on Journal Descriptor Indexing: Preliminary experiment. *Journal of the American Society for Information Science and Technology*, 57(1):96–113, 2006.

[67] Betsy L Humphreys, Donald AB Lindberg, Harold M Schoolman, and G Octo Barnett. The Unified Medical Language System An Informatics Research Collaboration. *Journal of the American Medical Informatics Association*, 5(1):1–11, 1998.

[68] Tamseela Hussain, George Michel, and Richard N Shiffman. The Yale Guideline Recommendation Corpus: a representative sample of the knowledge content of guidelines. *International journal of medical informatics*, 78(5):354–363, 2009.

[69] Nancy Ide and Jean Véronis. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, 24(1):2–40, 1998.

[70] TS Jayram, Rajasekar Krishnamurthy, Sriram Raghavan, Shivakumar Vaithyanathan, and Huaiyu Zhu. Avatar Information Extraction System. *IEEE Data Eng. Bull.*, 29(1):40–48, 2006.

[71] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. 1997.

[72] Antonio J Jimeno-Yepes, Bridget T McInnes, and Alan R Aronson. Exploiting MeSH indexing in MEDLINE to generate a data set for word sense disambiguation. *BMC bioinformatics*, 12(1):223, 2011.

[73] Dipak Kalra, Thomas Beale, and Sam Heard. The openEHR foundation. *Studies in health technology and informatics*, 115:153–173, 2005.

[74] Adam Kilgarri. Senseval: An exercise in evaluating word sense disambiguation programs. In *Proc. of the first international conference on language resources and evaluation*, pages 581–588, 1998.

[75] Adam Kilgarriff and Martha Palmer. Introduction to the special issue on SENSEVAL. *Computers and the Humanities*, 34(1-2):1–13, 2000.

[76] Adam Kilgarriff and Joseph Rosenzweig. Framework and results for English SENSEVAL. *Computers and the Humanities*, 34(1-2):15–48, 2000.

[77] Ellen Kilsdonk, Linda W Peute, Rinke J Riezebos, Leontien C Kremer, and Monique WM Jaspers. From an expert-driven paper guideline to a user-centred decision support system: A usability comparison study. *Artificial intelligence in medicine*, 59(1):5–13, 2013.

[78] Mi-Young Kim, Ying Xu, Osmar R Zaiane, and Randy Goebel. Recognition of Patient-Related Named Entities in Noisy Tele-Health Texts. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(4):59, 2015.

[79] Peter Kluegl, Martin Atzmueller, and Frank Puppe. Textmarker: A tool for rule-based information extraction. In *Proceedings of the Biennial GSCL Conference*, pages 233–240, 2009.

[80] Peter Kluegl, Martin Toepfer, Philip-Daniel Beck, Georg Fette, and Frank Puppe. UIMA Ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering*, pages 1–40, 2014.

[81] Robert Kosara, Silvia Miksch, Andreas Seyfang, and Peter Votruba. *Tools for acquiring clinical guidelines in Asbru*. 2002.

[82] A Latoszek-Berendsen, P de Clercq, J van den Herik, and A Hasman. Intention-based expressions in GASTINE. *Methods of information in medicine*, 48(4):391, 2009.

[83] Claudia Leacock and Martin Chodorow. Combining local context and WordNet similarity for word sense identification. 49(2):265–283, 1998.

[84] Claudia Leacock, Geoffrey Towell, and Ellen Voorhees. Corpus-based statistical sense resolution. In *Proceedings of the workshop on Human Language Technology*, pages 260–265. Association for Computational Linguistics, 1993.

[85] Claudia Leacock, Geoffrey Towell, and Ellen M Voorhees. Towards building contextual representations of word senses using statistical models. *Corpus Processing for Lexical Acquisition, MIT Press, Cambridge, Massachusetts*, pages 97–113, 1996.

[86] Yoong Keok Lee and Hwee Tou Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 41–48. Association for Computational Linguistics, 2002.

[87] Bass Len, Clements Paul, and Kazman Rick. Software architecture in practice. *Boston, Massachusetts Addison*, 2003.

[88] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM, 1986.

[89] Dekang Lin. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304, 1998.

[90] John Lyons. *Linguistic semantics: An introduction*. Cambridge University Press, 1995.

[91] Dmitry Malioutov and Kush Varshney. Exact rule learning via boolean compressed sensing. In *Proceedings of The 30th International Conference on Machine Learning*, pages 765–773, 2013.

[92] Bhaskara Marthi, Brian Milch, and Stuart Russell. First-order probabilistic models for information extraction. In *IJCAI 2003 Workshop on Learning Statistical Models from Relational Data*, 2003.

[93] Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham, and Yorick Wilks. Named entity recognition from diverse text types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, 2001.

[94] Alexa T McCray, Anita Burgun, and Olivier Bodenreider. Aggregating UMLS semantic types for reducing conceptual complexity. *Studies in health technology and informatics*, 84(0 1):216, 2001.

[95] Bridget T McInnes, Ted Pedersen, Ying Liu, Genevieve B Melton, and Serguei V Pakhomov. Knowledge-based method for determining the meaning of ambiguous

biomedical terms using information content measures of similarity. In *AMIA Annual Symposium Proceedings*, volume 2011, page 895. American Medical Informatics Association, 2011.

[96] Rada Mihalcea and Ehsanul Faruque. Senselearner: Minimally supervised word sense disambiguation for all words in open text. In *Proceedings of ACL/SIGLEX Senseval*, volume 3, pages 155–158, 2004.

[97] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244, 1990.

[98] Raymond J Mooney. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. 1996.

[99] Mark A Musen, Samson W Tu, Amar K Das, and Yuval Shahar. EON: A component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Informatics Association*, 3(6):367–388, 1996.

[100] Roberto Navigli and Mirella Lapata. Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. In *IJCAI*, pages 1683–1688, 2007.

[101] Hwee Tou Ng. Exemplar-based word sense disambiguation: Some recent improvements. 1997.

[102] Hwee Tou Ng and Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 40–47. Association for Computational Linguistics, 1996.

[103] Silvia Panzarasa, S Madde, Silvana Quaglini, Caterina Pistarini, and Mario Stefanelli. Evidence-based careflow management systems: the case of post-stroke rehabilitation. *Journal of biomedical informatics*, 35(2):123–139, 2002.

[104] Ted Pedersen and Rebecca Bruce. A new supervised learning algorithm for word sense disambiguation. In *AAAI/IAAI*, pages 604–609, 1997.

[105] Mor Peleg. Computer-interpretable clinical guidelines: a methodological review. *Journal of biomedical informatics*, 46(4):744–763, 2013.

[106] Mor Peleg, Aziz A Boxwala, Elmer Bernstam, et al. Sharable representation of clinical guidelines in GLIF: relationship to the Arden Syntax. *Journal of biomedical informatics*, 34(3):170–181, 2001.

[107] Kristi-Anne Polvani, Abha Agrawal, Bryat Karras, et al. Gem cutter, 2000.

[108] Silvana Quaglini, Mario Stefanelli, Giordano Lanzola, Vincenzo Caporusso, and Silvia Panzarasa. Flexible guideline-based patient careflow systems. *Artificial intelligence in medicine*, 22(1):65–80, 2001.

[109] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[110] Chen R. Guideline Definition Language (GDL). `www.openehr.org/news_events/releases/releases.php?id=79`, February 2015.

[111] Ganesh Ramakrishnan, Sachindra Joshi, Sreeram Balakrishnan, and Ashwin Srinivasan. Using ilp to construct features for information extraction from semi-structured text. In *Inductive Logic Programming*, pages 211–224. Springer, 2008.

[112] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. 1995.

[113] Thomas C Rindflesch, Halil Kilicoglu, Marcelo Fiszman, Graciela Rosemblat, and Dongwook Shin. Semantic MEDLINE: An advanced information management application for biomedicine. *Information Services and Use*, 31(1):15–21, 2011.

[114] Ronald L Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

[115] Chuck Rleger and Steve Small. Word expert parsing. In *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 2*, pages 723–728. Morgan Kaufmann Publishers Inc., 1979.

[116] Willie Rogers. Using the MetaMap UIMA Annotator, 2010.

[117] HR Rubin. Why don't physicians follow clinical practice guidelines. *A framework for*, 1999.

[118] Marek Ruzicka and Vojtech Svatek. Mark-up based analysis of narrative guidelines with the Stepper tool. *Studies in health technology and informatics*, 101:132–136, 2003.

[119] Sunita Sarawagi. Information extraction. *Foundations and trends in databases*, 1(3):261–377, 2008.

[120] Martijn J Schuemie, Jan A Kors, and Barend Mons. Word sense disambiguation in the biomedical domain: an overview. *Journal of Computational Biology*, 12(5):554–565, 2005.

[121] Andreas Seyfang, Begoña Martínez-Salvador, Radu Serban, et al. Maintaining formal models of living guidelines efficiently. In *Artificial Intelligence in Medicine*, pages 441–445. Springer, 2007.

[122] Yuval Shahar, Silvia Miksch, and Peter Johnson. An intention-based language for representing clinical guidelines. In *Proceedings of the AMIA Annual Fall Symposium*, page 592. American Medical Informatics Association, 1996.

[123] Warren Shen, AnHai Doan, Jeffrey F Naughton, and Raghu Ramakrishnan. Declarative information extraction using datalog with embedded extraction predicates. In *Proceedings of the 33rd international conference on Very large data bases*, pages 1033–1044. VLDB Endowment, 2007.

[124] Richard N Shiffman, Bryant T Karras, Abha Agrawal, et al. GEM: a proposal for a more comprehensive guideline document model using XML. *Journal of the American Medical Informatics Association*, 7(5):488–498, 2000.

[125] Richard N Shiffman, George Michel, Abdelwaheb Essaihi, and Elizabeth Thornquist. Bridging the guideline implementation gap: a systematic, document-centered

approach to guideline implementation. *Journal of the American Medical Informatics Association*, 11(5):418–426, 2004.

[126] Richard N Shiffman, George Michel, Richard M Rosenfeld, and Caryn Davidson. Building better guidelines with BRIDGE-Wiz: development and evaluation of a software assistant to promote clarity, transparency, and implementability. *Journal of the American Medical Informatics Association*, 19(1):94–101, 2012.

[127] Ravi Som Sinha and Rada Mihalcea. Unsupervised Graph-basedWord Sense Disambiguation Using Measures of Word Semantic Similarity. In *ICSC*, volume 7, pages 363–369, 2007.

[128] S Skonetzki, HJ Gausepohl, Minne van der Haak, et al. HELEN, a modular framework for representing and implementing clinical practice guidelines. *Methods of information in medicine*, 43(4):413–426, 2004.

[129] Margarita Sordo, Omolola Ogunyemi, Aziz A Boxwala, and Robert A Greenes. GELLO: an object-oriented query and expression language for clinical decision support: AMIA 2003 Open Source Expo. In *AMIA Annual Symposium Proceedings*, volume 2003, page 1012. American Medical Informatics Association, 2003.

[130] Michael M Stark and Richard F Riesenfeld. Wordnet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering*, page 21, 1998.

[131] V Stroetman, Dipak Kalra, Pierre Lewalle, et al. Semantic interoperability for better health and safer healthcare [34 pages]. 2009.

[132] Paolo Terenziani, Gianpaolo Molino, and Mauro Torchio. A modular approach for representing and executing clinical guidelines. *Artificial intelligence in medicine*, 23(3):249–276, 2001.

[133] William M Tierney. Improving clinical decisions and outcomes with information: a review. *International journal of medical informatics*, 62(1):1–9, 2001.

[134] Geoffrey Towell and Ellen M Voorhees. Disambiguating highly ambiguous words. *Computational Linguistics*, 24(1):125–145, 1998.

[135] George Tsatsaronis, Michalis Vazirgiannis, and Ion Androutsopoulos. Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri. In *IJCAI*, volume 7, pages 1725–1730, 2007.

[136] Samson W Tu, James R Campbell, Julie Glasgow, et al. The SAGE Guideline Model: achievements and overview. *Journal of the American Medical Informatics Association*, 14(5):589–598, 2007.

[137] Florentina Vasilescu, Philippe Langlais, and Guy Lapalme. Evaluating Variants of the Lesk Approach for Disambiguating Words. In *LREC*, 2004.

[138] Peter Votruba, Silvia Miksch, and Robert Kosara. *Linking clinical guidelines with formal representations*. Springer, 2003.

[139] Peter Votruba, Silvia Miksch, Andreas Seyfang, and Robert Kosara. Tracing the formalization steps of textual guidelines. *Studies in health technology and informatics*, pages 172–176, 2004.

[140] Kiri Wagstaff. Machine learning that matters. 2012.

[141] Warren Weaver. Translation. *Machine translation of languages*, 14:15–23, 1955.

[142] Marc Weeber, James G Mork, and Alan R Aronson. Developing a test collection for biomedical word sense disambiguation. In *Proceedings of the AMIA Symposium*, page 746. American Medical Informatics Association, 2001.

[143] Yorick Wilks. Preference semantics. Technical report, DTIC Document, 1973.

[144] Steven H Woolf. Evidence-based medicine and practice guidelines: an overview. *Cancer Control*, 7(4):362–367, 2000.

[145] Steven H Woolf, Richard Grol, Allen Hutchinson, Martin Eccles, and Jeremy Grimshaw. Clinical guidelines: potential benefits, limitations, and harms of clinical guidelines. *BMJ: British Medical Journal*, 318(7182):527, 1999.

[146] Fei Wu and Daniel S Weld. Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50. ACM, 2007.

[147] Jeremy C Wyatt. Practice guidelines and other support for clinical innovation. *Journal of the Royal Society of Medicine*, 93(6):299, 2000.

[148] Rong Xu and QuanQiu Wang. Large-scale extraction of accurate drug-disease treatment pairs from biomedical literature for drug repurposing. *BMC bioinformatics*, 14(1):181, 2013.

[149] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995.

[150] Daniel Yuen and Hanns Koehler-Kruener. Who's Who in Text Analytics. *Stamford, CT: Gartner, Inc*, 2012.

Appendix A

Glossary

A.1 Betweeness centrality

Betweenness is a metric of a node's centrality within a graph. Betweenness centrality for a node v in a graph G, is equal to the number of shortest paths from all vertices to all others that pass through v.

A.2 Page Rank

Page Rank is an algorithm to compute a numeric value that represents the importance of a page present on the web. When one page links to another page, it is casting a vote for the other page. Importance of the page that is casting the vote determines the importance of the vote. Importance of each vote is taken into account when a page's Page Rank is calculated.

A.3 Formalize

The Oxford Dictionary defines formalize as "Give a definite structure or shape to". In this thesis formalize is the execution of the process that generate a structured computer interpretable guideline from a narrative clinical practice guideline.

A.4 Formalization activities

Formalization activities are the tasks that collectively compose the clinical practice guideline formalization process.

A.5 openEHR Archetypes

Archetypes are the keystone of the openEHR architecture. They are the models used for the capture of clinical information into a machine readable specification. Each archetype is a computable definition, or specification, for a single, discrete clinical concept. The specification is expressed in Archetype Definition Language (ADL).

Appendix B

UMLS Semantic Network

The next two sections list the hierarchy of the types and the relations of the UMLS [67] semantic network.

B.1 Semantic Types

Physical Object
→ Organism
→ → Plant
→ → Fungus
→ → Virus
→ → Bacterium
→ → Archaeon
→ → Eukaryote
→ → Animal
→ → → Vertebrate
→ → → → Amphibian
→ → → → Bird
→ → → → Fish
→ → → → Reptile
→ → → → Mammal

→ → → → → Human
→ Anatomical Structure
→ → Embryonic Structure
→ → Anatomical Abnormality
→ → → Congenital Abnormality
→ → → Acquired Abnormality
→ → Fully Formed Anatomical Structure
→ → → Body Part, Organ, or Organ Component
→ → → Tissue
→ → → Cell
→ → → Cell Component
→ → → Gene or Genome
→ Manufactured Object
→ → Medical Device
→ → → Drug Delivery Device
→ → Research Device
→ → Clinical Drug
→ Substance
→ → Chemical
→ → → Chemical Viewed Functionally
→ → → → Pharmacologic Substance
→ → → → → Antibiotic
→ → → → Biomedical or Dental Material
→ → → → Biologically Active Substance
→ → → → → Neuroreactive Substance or Biogenic Amine
→ → → → → Hormone
→ → → → → Enzyme
→ → → → → Vitamin
→ → → → → Immunologic Factor
→ → → → → Receptor
→ → → → Indicator, Reagent, or Diagnostic Acid

→ → → → Hazardous or Poisonous Substance
→ → → Chemical Viewed Structurally
→ → → → Organic Chemical
→ → → → → Nucleic Acid, Nucleoside, or Nucleotide
→ → → → → Organophosphorus Compound
→ → → → → Amino Acid, Peptide, or Protein
→ → → → → Carbohydrate
→ → → → → Lipid
→ → → → → → Steroid
→ → → → → → Eicosanoid
→ → → → Inorganic Chemical
→ → → → Element, Ion, or Isotope
→ → Body Substance
→ → Food
Conceptual Entity
→ Idea or Concept
→ → Temporal Concept
→ → Qualitative Concept
→ → Quantitative Concept
→ → Functional Concept
→ → → Body System
→ → Spatial Concept
→ → → Body Space or Junction
→ → → Body Location or Region
→ → → Molecular Sequence
→ → → → Nucleotide Sequence
→ → → → Amino Acid Sequence
→ → → → Carbohydrate Sequence
→ → → Geographic Area
→ Finding
→ → Laboratory or Test Result

→ → Sign or Symptom
→ Organism Attribute
→ → Clinical Attribute
→ Intellectual Product
→ → Classification
→ → Regulation or Law
→ Language
→ Occupation or Discipline
→ → Biomedical Occupation or Discipline
→ Organization
→ → Health Care Related Organization
→ → Professional Society
→ → Self-help or Relief Organization
→ Group Attribute
→ Group
→ → Professional or Occupational Group
→ → Population Group
→ → Family Group
→ → Age Group
→ → Patient or Disabled Group

B.2 Semantic Relations

Is a
Associated with
→ Physically related to
→ → Part of
→ → Consists of
→ → Contains
→ → Connected to
→ → Interconnects

→ → Branch of
→ → Tributary of
→ → Ingredient of
→ Spatially related to
→ → Location of
→ → Adjacent to
→ → Surrounds
→ → Traverses
→ Functionally related to
→ → Affects
→ → → Manages
→ → → Treats
→ → → Disrupts
→ → → Complicates
→ → → Interacts with
→ → → Prevents
→ → Brings about
→ → → Produces
→ → → Causes
→ → Performs
→ → → Carries out
→ → → Exhibits
→ → → Practices
→ → Occurs in
→ → → Process of
→ → Uses
→ → Manifestation of
→ → Indicates
→ → Result of
→ Temporally related to
→ → Co-occurs with

→ → Precedes
→ Conceptually related to
→ → Evaluation of
→ → Method of
→ → Conceptual part of
→ → Issue in
→ → Degree of
→ → Analyzes
→ → → Assesses effect of
→ → Measurement of
→ → Measures
→ → Diagnoses
→ → Property of
→ → Derivative of
→ → Developmental form of

Appendix C

MSH-WSD Dataset

The next two sections list the terms and acronyms of the MSH-WSD dataset [72] formatted as follow:

[Term or acronym] → *[UMLS Concept unique identifier]* - *[UMLS Concept Description]*

C.1 Terms

Adrenal → C0001625 - Suprarenal gland
Adrenal → C0014563 - therapeutic epinephrine
Arteriovenous Anastomoses → C0225984 - Structure of anatomic arteriovenous anastomosis (body structure)
Arteriovenous Anastomoses → C0684204 - Surgical construction of arteriovenous shunt, NOS
Astragalus → C0039277 - Tibial tarsal bone
Astragalus → C0330845 - Plants, Astragalus
B-Cell Leukemia → C0023434 - Well-Differentiated Lymphocytic Lymphomas
B-Cell Leukemia → C2004493 - Lymphocytic Leukemias, B-Cell
Borrelia → C0006033 - Genus Borrelia (organism)
Borrelia → C0024198 - Steere's disease
Brucella abortus → C0006304 - Brucella melitensis bv. Abortus
Brucella abortus → C0302363 - infection; Brucella, abortus

Callus → C0006767 - Callus, Bony

Callus → C0376154 - Skin callus

Cardiac pacemaker → C0030163 - supplies cardiac pacemaker

Cardiac pacemaker → C0037189 - Structure of sinoatrial node (body structure)

Cell → C0007634 - THE CELL

Cell → C1136359 - Telephones, Cellular

Cement → C0011343 - Dental Cementum

Cement → C1706094 - Resins, Adhesive, Orthodontic Bracket

Cholera → C0008354 - Vibrio cholerae infection

Cholera → C0008359 - Vaccines, Cholera

Cilia → C0008778 - Cilium, NOS

Cilia → C0015422 - Structure of eyelashes (body structure)

Coffee → C0009237 - Coffee, NOS

Coffee → C0085952 - coffee <Coffea>

Cold → C0009264 - Temperatures, Cold

Cold → C0009443 - VIRAL UPPER RESPIRATORY INFECTION

Cold → C0024117 - respiratory tract; disorder, obstructive, chronic

Compliance → C0009563 - Volume Change to Pressure Change Ratio

Compliance → C1321605 - Treatment Compliance

Cortex → C0001614 - Disorder of adrenal cortex (disorder)

Cortex → C0007776 - Structure of pallium (body structure)

Cortical → C0001613 - Suprarenal cortex

Cortical → C0007776 - Structure of pallium (body structure)

Cortical → C0022655 - Structure of cortex of kidney (body structure)

Crack → C0040441 - tooth; fracture

Crack → C0085163 - Rocks - cocaine

Crown → C0010384 - Total Dental Crowns, Temporary

Crown → C0226993 - Tooth Crowns

Digestive → C0012238 - Digestive tract function, NOS

Digestive → C0012240 - Systema digestorium

drinking → C0001948 - use; alcohol

drinking → C0684271 - Drinkings

Eels → C0013671 - Order anguilliformes (organism)

Eels → C0677644 - Spectroscopy, Electron Energy-Loss

ERUPTION → C0015230 - Spots [D]

ERUPTION → C1533692 - Tooth Eruptions

Erythrocytes → C0014772 - Whole Blood Erythrocytic Cell Counts

Erythrocytes → C0014792 - Reticuloendothelial System, Erythrocytes

Exercises → C0015259 - Physical exercises (regime/therapy)

Exercises → C0452240 - Therapy, Exercise

Familial Adenomatous Polyposis → C0032580 - Polyposus, Familial Multiple

Familial Adenomatous Polyposis → C0162832 - POLYPOSIS, ADENOMATOUS INTESTINAL

Fish → C0016163 - SECTION C FISHES

Fish → C0162789 - Techniques, FISH

Follicle → C0018120 - Ovary follicle

Follicle → C0221971 - Hair Follicles

Follicles → C0018120 - Ovary follicle

Follicles → C0221971 - Hair Follicles

Gamma-Interferon → C0021740 - Type II Interferon, Recombinant

Gamma-Interferon → C0021745 - type II interferon

Ganglion → C0017067 - Neural Ganglion

Ganglion → C1258666 - Myxoid Cysts

Glycoside → C0007158 - Steroids, Cardiotonic

Glycoside → C0017977 - Glycosides [Chemical/Ingredient]

Haemophilus ducreyi → C0007947 - virulent; bubo

Haemophilus ducreyi → C0018481 - Hemophilus ducreyi

Hemlock → C0242872 - Hemlocks

Hemlock → C0949851 - Tsugas

Heregulin → C0626201 - SMDF

Heregulin → C0752253 - Sensory-and-motor-derived factor

Hybridization → C0020202 - Hybridizations, Genetic

Hybridization → C0028602 - Nucleic Acid Hybridizations
INDO → C0021246 - Indomethacin product (substance)
INDO → C0021247 - Netherlands East Indies
Iris → C0022077 - Iris, NOS
Iris → C1001362 - Plants, Iris
LABOR → C0022864 - Obstetric Labor
LABOR → C0043227 - Working, function (observable entity)
Lactation → C0006147 - Nursing
Lactation → C0022925 - Lactation, NOS
Language → C0023008 - Languages
Language → C0033348 - Programming Languages
Laryngeal → C0023078 - voicebox
Laryngeal → C0023081 - PROSTHESIS, LARYNGEAL (TAUB)
Lawsonia → C0752045 - Lawsonia McOrist et al. 1995
Lawsonia → C1068388 - Plants, Lawsonia
Leishmaniasis → C0023281 - Leishmaniosis
Leishmaniasis → C1548483 - Vaccines, Leishmaniasis
lens → C0023308 - Lens disorders
lens → C0023317 - Structure of lens of eye (body structure)
lens → C0023318 - Lenses
Lupus → C0024131 - vulgaris; lupus
Lupus → C0024138 - lupus; discoid
Lupus → C0024141 - Systemic lupus erythematosus, unspecified
lymphogranulomatosis → C0019829 - Sarcoma;Hodgkins
lymphogranulomatosis → C0036202 - syndrome; Schaumann
Malaria → C0024530 - Unspecified malaria (disorder)
Malaria → C0206255 - Vaccines, Malarial
Medullary → C0001629 - Suprarenal medulla
Medullary → C0025148 - Myelencephalon
Milk → C0026131 - Milk, NOS
Milk → C0026140 - Mother's milk (substance)

Moles → C0027960 - Skin Moles

Moles → C0324740 - Talpidae

Murine sarcoma virus → C0026399 - Virus, Moloney Sarcoma

Murine sarcoma virus → C0026630 - Sarcoma Viruses, Murine

NEUROFIBROMATOSIS → C0085113 - Watson disease

NEUROFIBROMATOSIS → C0162678 - Syndromes, Neurofibromatosis

Nurse → C0006147 - Nursing

Nurse → C0028661 - Sr - Nursing sister

Nursing → C0006147 - Nursing

Nursing → C0028677 - Nursings

Parotitis → C0026780 - parotitis; infectious

Parotitis → C0030583 - Parotitis, NOS

Pharmaceutical → C0013058 - Pharmaceuticals

Pharmaceutical → C0031336 - Pharmacy (field)

Phosphorus → C0031705 - Phosphorus, NOS

Phosphorus → C0080014 - Phosphorus, Dietary [Chemical/Ingredient]

Phosphorylase → C0017916 - Phosphorylases [Chemical/Ingredient]

Phosphorylase → C0917783 - Polyphosphorylase

Plague → C0032064 - Yersinia pestis; infection

Plague → C0032066 - Vaccine, Plague

Plaque → C0011389 - Tooth plaque

Plaque → C0333463 - Senile Plaques

Platelet → C0005821 - thrombocytes

Platelet → C0032181 - Whole Blood Platelet Counts

Pleuropneumonia → C0026934 - Pleuropneumonia

Pleuropneumonia → C0032241 - Pleuropneumonias

Pneumocystis → C0032305 - Pulmonary pneumocystosis (disorder)

Pneumocystis → C0597258 - Pneumocystis species (organism)

Polymyalgia Rheumatica → C0032533 - Syndrome, Forestier-Certonciny

Polymyalgia Rheumatica → C0039483 - POLYMYALGIA RHEUMATICA

posterior pituitary → C0032009 - Posterior Pituitary Glands

posterior pituitary → C0032017 - Posterior pituitary hormones (substance)

Potassium → C0032821 - Potassium, NOS

Potassium → C0162800 - Potassium, Dietary [Chemical/Ingredient]

Projection → C0016538 - PROJECTIONS PREDICTIONS

Projection → C0033363 - Thought projection

Radiation → C0851346 - Rays

Radiation → C1522449 - Therapy, Radiation

Respiration → C0035203 - Ventilation, NOS

Respiration → C0282636 - Respiration, Cellular

Retinal → C0035298 - Tunica interna of eyeball

Retinal → C0035331 - Vitamin A Aldehyde

Root → C0040452 - Tooth Roots

Root → C0242726 - Roots, Plant

SARS → C1175175 - Severe Acute Respiratory Syndrome [Disease/Finding]

SARS → C1175743 - Urbani SARS-Associated Coronavirus

SARS-associated coronavirus → C1175175 - Severe Acute Respiratory Syndrome [Disease/Finding]

SARS-associated coronavirus → C1175743 - Urbani SARS-Associated Coronavirus

Schistosoma mansoni → C0036319 - Schistosoma mansonus

Schistosoma mansoni → C0036330 - schistosomiasis; Schistosoma mansoni

Semen → C0036563 - Zygotes, Plant

Semen → C0036614 - Seminal Plasma

sex factor → C0015435 - Transfer Factors, Resistance

sex factor → C0036881 - Sex Factors

Sodium → C0037473 - Sodium, NOS

Sodium → C0037570 - Sodium, Dietary [Chemical/Ingredient]

Staph → C0038160 - Staphylococl infectn,unspcf

Staph → C0038170 - Staphylococcus, NOS

STEM → C0162731 - STEM

STEM → C0242767 - Stems, Plant

Sterilization → C0038280 - Sterilization for infection control

Sterilization → C0038288 - Sterilizations, Reproductive

Strep → C0038395 - Streptococcus infection

Strep → C0038402 - Streptococcus, NOS

Synapsis → C0039062 - synaptic junction

Synapsis → C0598501 - Synapsis, Chromosomal

THYMUS → C0040112 - Thymus gland

THYMUS → C0040113 - Thymus, NOS

THYMUS → C1015036 - Thymus Plants

Tolerance → C0013220 - Tolerances, Drug

Tolerance → C0020963 - Tolerance, Immune

tomography → C0040395 - Tomography (procedure)

tomography → C0040405 - X-Ray Tomography, Computed

Torula → C0010414 - TORULOSIS

Torula → C0010415 - Torulas

Ventricles → C0007799 - Ventricles, Cerebral

Ventricles → C0018827 - Ventricular

veterinary → C0042615 - veterinary medicine (field)

veterinary → C0206212 - Veterinary Technicians

Wasp → C0043041 - Wasps

Wasp → C0258432 - Wiskott-Aldrich Syndrome Protein [Chemical/Ingredient]

Yellow Fever → C0043395 - YF - Yellow fever

Yellow Fever → C0301508 - Yellow fever vaccine product

C.2 Acronyms

AA → C0001972 - Anonymous, Alcoholics

AA → C0002520 - aminoacid

ADA → C0001457 - EC 3.5.4.4

ADA → C0002456 - Dental Association, American

ADH → C0001942 - Oxidoreductase, Alcohol-NAD+

ADH → C0003779 - Vasopressin, Arginine

ADP → C0001459 - Pyrophosphate, Adenosine

ADP → C0004374 - Processing, Electronic Data

Ala → C0001898 - L-Isomer Alanine

Ala → C0002563 - Pentanoic acid, 5-amino-4-oxo-

Ala → C0051405 - Fatty Acid cis, cis, cis 18:3 n-3

ALS → C0002736 - spinal; sclerosis, lateral (amyotrophic)

ALS → C0003372 - Serums, Antilymphocyte

ANA → C0002463 - Nurses' Associations, American

ANA → C0003243 - Factors, Antinuclear

BAT → C0006298 - Tissue, Brown Adipose

BAT → C0008139 - Order Chiroptera (organism)

BLM → C0005740 - BLM

BLM → C0005859 - Syndrome, Bloom-Torre-Machacek

BPD → C0006012 - Personality Disorders, Borderline

BPD → C0006287 - ventilator lung; newborn

BR → C0006137 - Brazil (geographic location)

BR → C0006222 - Bromides [Chemical/Ingredient]

BSA → C0005902 - Surface Areas, Body

BSA → C0036774 - Serum Albumin, Bovine [Chemical/Ingredient]

BSE → C0085105 - Self-Examinations, Breast

BSE → C0085209 - Spongiform Encephalopathy, Bovine

Ca → C0006675 - IV, Coagulation Factor

Ca → C0006754 - California Aldasoro et al.

Ca → C0006823 - Canada (geographic location)

Ca → C0019564 - Horn, Ammon's

CAD → C0011905 - Diagnosis, Computer-Assisted

CAD → C1956346 - Disorder of coronary artery (disorder)

CAM → C0007578 - Molecules, Cell Adhesion

CAM → C0178551 - Membranes, Chorioallantoic

CCD → C0008928 - Scheuthauer-Marie-Sainton syndrome

CCD → C0751951 - Syndrome, Shy-Magee

CCl4 → C0007022 - Tetrachloromethane (substance)

CCl4 → C0209338 - Small Inducible Cytokine A4

CDA → C0002876 - Dyserythropoietic Anemias, Congenital

CDA → C0092801 - Cladribine product

CDR → C0011485 - Deoxyriboside, Cytosine

CDR → C0021024 - Regions, Hypervariable

CH → C0008115 - PRC

CH → C0039021 - SZ

CI → C0008107 - CL

CI → C0022326 - Republic of Cote diIvoire

CIS → C0007099 - Preinvasive Carcinoma

CIS → C0162854 - Commonwealth of Independent States

CLS → C0265252 - Syndrome, Coffin-Lowry

CLS → C0343084 - Systemic Capillary Leak Syndrome

CNS → C0028654 - Specialists, Clinical Nurse

CNS → C0927232 - Systems, Central Nervous

CP → C0007789 - paralysis; cerebral

CP → C0008925 - Uranostaphyloschisis (disorder)

CP → C0033477 - Propionicibacterium acnes

CPDD → C0008838 - Platinum, diamminedichloro-, (SP-4-2)-

CPDD → C0553730 - Pyrophosph cryst-unspec

CRF → C0010132 - CRH-Corticotrophin rel horm

CRF → C0022661 - Unspecified chronic renal failure

cRNA → C0056208 - RNA, Complementary [Chemical/Ingredient]

cRNA → C1321571 - Nurse anesthetists

CTX → C0010583 - Zytoxan

CTX → C0238052 - Xanthomatosis, Cerebrotendinous [Disease/Finding]

DAT → C0002395 - simple senile dementia

DAT → C0114838 - Transporters, Dopamine-Specific Neurotransmitter

DBA → C0025923 - Mouse, Inbred DBA

DBA → C1260899 - red cell; aplasia, congenital

dC → C0011485 - Deoxyriboside, Cytosine

dC → C0012764 - Washington, DC

DDD → C0011037 - TDE

DDD → C0026256 - Ortho,para-DDD

DDS → C0010980 - Sulphadione

DDS → C0085104 - Targetings, Drug

DDS → C0950121 - Wilms' tumour and nephrotic syndrome with pseudohermaphroditism

DE → C0011198 - Delaware (geographic location)

DE → C0017480 - GM

DI → C0011848 - diabetes; insipidus

DI → C0032246 - Ploidy, NOS

DON → C0012020 - Norleucine, 6-diazo-5-oxo-

DON → C0028652 - Vice President for Nursing

eCG → C0018064 - Gonadotropins, Equine [Chemical/Ingredient]

eCG → C1623258 - Electrocardiography NOS (regime/therapy)

EGG → C0013710 - Eggs (edible) (substance)

EGG → C0029974 - X-bearing ovum

EM → C0014921 - Estramustine [Chemical/Ingredient]

EM → C0026019 - Microscopy.electron

EMS → C0013961 - Services, Medical Emergency

EMS → C0015063 - Sulfonate, Ethylmethane

Epi → C0014563 - therapeutic epinephrine

Epi → C0014582 - Pidorubicin

ERP → C0008310 - X-ray gastrointestinal ERCP

ERP → C0015214 - Potentials, Evoked

FA → C0015625 - Short Limb Dwarfism-Saddle Nose-Spinal Alterations-Metaphyseal Striation Syndrome

FA → C0016410 - Vitamin M

FAS → C0015683 - Synthase, Fatty Acid

FAS → C0015923 - syndrome; fetal, alcohol (dysmorphic)

Fe → C0302583 - Iron, NOS

Fe → C0376520 - Iron, Dietary [Chemical/Ingredient]

FTC → C0041713 - USFTC

FTC → C0206682 - Well-Differentiated Follicular Carcinoma

GAG → C0017346 - Genes, gag

GAG → C0017973 - Mucopolysaccharides

Gas → C0016204 - Wind symptom (finding)

Gas → C0017110 - Gases [Chemical/Ingredient]

HCl → C0020259 - muriaticum acidum/hydrochlor

HCl → C0023443 - reticuloendotheliosis; leukemic

HGF → C0021760 - Plasmacytoma Growth Factor

HGF → C0062534 - Scatter Factor

HHV 8 → C0036220 - Skin cancer, Kaposi's sarcoma

HHV 8 → C0376526 - Virus-HHV8

Hip → C0019552 - Regio coxae

Hip → C0022122 - Os ischii

HIV → C0019682 - Virus-HIV

HIV → C0019693 - Unspecified human immunodeficiency virus [HIV] disease

HPS → C0079504 - oculocutaneous albinism

HPS → C0242994 - Infections, Hantavirus

HR → C0010343 - HRV

HR → C0018810 - Rates, Heart

IA → C0021487 - Intra-Arterial Injections

IA → C0022037 - Iowa (geographic location)

Ice → C0020746 - Water Ice

Ice → C0025611 - Tina

Ice → C0534519 - P45

Ion → C0022023 - Ions [Chemical/Ingredient]

Ion → C0022024 - Physical medicine iontophoresis -RETIRED-

IP → C0021069 - Precipitations, Immune

IP → C0021171 - Syndrome, Bloch-Sulzberger

ITP → C0021540 - Triphosphate, Inosine

ITP → C0043117 - Werlhof's syndrome

JP → C0022341 - JPN

JP → C0031106 - Prepubertal periodontitis (disorder)

MAF → C0079786 - MAF

MAF → C0919482 - Transcription Factors, Maf

MBP → C0014063 - Proteins, Myelin Basic

MBP → C0065661 - MBP

MCC → C0007129 - Tumor, Merkel Cell

MCC → C0162804 - MUTATED IN COLORECTAL CANCERS

MHC → C0024518 - MHC

MHC → C0027100 - Myosin Heavy Chains [Chemical/Ingredient]

MRS → C0024487 - Spectroscopy, MR

MRS → C0025235 - Syndrome, Melkerson Rosenthal

NBS → C0027819 - Neuroblastomas

NBS → C0398791 - Syndrome, Nijmegen Breakage

NM → C0025033 - NM

NM → C0027972 - NM

NPC → C0028587 - Pores, Nuclear

NPC → C0220756 - Vertical Ophthalmoplegias, Supraoptic

OCD → C0028768 - reaction; obsessive-compulsive

OCD → C0029421 - osteochondrosis; dissecans

OH → C0028905 - Ohio (geographic location)

OH → C0063146 - OH

Orf → C0013570 - Sore mouth (ovine)

Orf → C0079941 - Regions, Protein Coding

ORI → C0206601 - United States Office of the Assistant Secretary for Health Office of Research Integrity

ORI → C0242961 - Replication Origins

PAC → C0033036 - SVE

PAC → C0949780 - PACs (Chromosomes)

PAF → C0032172 - Thrombocyte Aggregating Activity

PAF → C0037019 - Syndromes, Dysautonomia-Orthostatic Hypotension

PCA → C0030131 - p-Chloroamphetamine [Chemical/Ingredient]

PCA → C0030625 - PCA

PCA → C0078944 - PCA - Pt controlled analgesia

PCA → C0149576 - Structure of posterior cerebral artery (body structure)

PCA → C0429865 - Principal Components Analysis

PCB → C0032447 - POLYCHLOROBIPHENYL CPDS

PCB → C0033223 - Procarbazine [Chemical/Ingredient]

PCD → C0022521 - Triads, Kartagener

PCD → C0162638 - type I programmed cell death

PCP → C0030855 - Phenol, pentachloro-

PCP → C0031381 - Piperidine, 1-(1-phenylcyclohexyl)-

PCP → C0032305 - Pulmonary pneumocystosis (disorder)

PEP → C0031642 - Phosphoenolpyruvate [Chemical/Ingredient]

PEP → C0135981 - Peplomycin [Chemical/Ingredient]

PHA → C0030779 - PHA

PHA → C0031858 - vulgaris Lectins, Phaseolus

pI → C0022171 - Points, Isoelectric

pI → C0812425 - S-Phase Fraction

POL → C0017360 - pol genes

POL → C0032356 - Poland (geographic location)

PR → C0034044 - RQ

PR → C0034833 - Receptors, Progestin

PVC → C0032624 - vinylchloride polymer

PVC → C0151636 - VPC's

RA → C0002893 - refractory; anemia

RA → C0003873 - Systemic rheumatoid arthritis

RA → C0034625 - Radium, NOS

RB → C0035335 - Retinoblastomas

RB → C0035930 - Rubidium, NOS

RBC → C0014772 - Whole Blood Erythrocytic Cell Counts

RBC → C0014792 - Reticuloendothelial System, Erythrocytes

rDNA → C0012931 - Recombinant DNA

rDNA → C0012933 - ribosomal DNA

RSV → C0035236 - Viruses, Respiratory Syncytial

RSV → C0086943 - virus, Rous sarcoma

SCD → C0002895 - Sickling disorder due to hemoglobin S (disorder)

SCD → C0085298 - sudden; cardiac death

SLS → C0037231 - spastic quadriplegia-congenital ichthyosiform erythroderma-oligophrenia syndrome

SLS → C0037506 - Sulfuric acid monododecyl ester sodium salt

SPR → C0164209 - TACR1

SPR → C0597731 - Surface Plasmon Resonances

SS → C0039101 - Synoviomas

SS → C0085077 - Syndrome, Sweet's

TAT → C0017375 - tat Genes

TAT → C0039341 - Trans-Activator of Transcription of HIV

TAT → C0039756 - Thematic Apperception Tests

Tax → C0039371 - Taxes

Tax → C0144576 - TAX

TEM → C0040975 - Triethylenemelamine [Chemical/Ingredient]

TEM → C0678118 - Transmission Electron Microscopy

TLC → C0008569 - TLC

TLC → C0040509 - Total lung capacity (TLC)

TMJ → C0039493 - TMJ - Temporomandibular joint

TMJ → C0039496 - TMJPDS-Temprmand jt pn dys syn

TMP → C0040079 - TMP

TMP → C0041041 - Trimethoprim [Chemical/Ingredient]

TNC → C0076088 - TNC

TNC → C0077400 - Troponin-C

TNT → C0041070 - Trinitrotoluene, device (physical object)

TNT → C0077404 - Troponin-T

TPA → C0032143 - TTPA

TPA → C0039654 - TPA (tetradecanoylphorbol acetate)

TPO → C0021965 - Tyrosine Iodinase

TPO → C0040052 - TSF

TRF → C0021759 - TRF (T cell replacing factor)

TRF → C0040162 - TRH-Thyrotrophin rel horm

TSF → C0021756 - T-stimulating factor

TSF → C0040052 - TSF

TYR → C0041484 - Tyrosinase

TYR → C0041485 - Tyrosine, L-isomer

US → C0041618 - USS - Ultrasound scan

US → C0041703 - USA - United States of America

WBS → C0004903 - Wiedemann-Beckwith-Combs syndrome

WBS → C0175702 - WS

WT1 → C0027708 - WT1

WT1 → C0148873 - WT33