ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

CONCEPT VALIDATION SYSTEM ON MEDICAL SENTENCES

By: Elias Muluneh

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF THE ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

October, 2009

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUSTE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

CONCEPT VALIDATION SYSTEM ON MEDICAL SENTENCES

By: Elias Muluneh

ADVISOR:

Fitsum Admasu (PhD)

APPROVED BY

EXAMINING BOARD:

1. Fitsum Admasu (PhD), Advisor ______________________
2. Dejene Ejigu (PhD) ______________________
3. ____________________________ ______________________

Acknowledgement

My gratitude goes to a lot of people. First, I would like to thank my advisor Dr. Fitsum, for the patience and dedication he had to help me with the ambitious plans I had. I also thank the Hayat Medical College for collaborating on different levels on this work. I especially thank Dr. Alemseged who has assisted me in so many ways whenever I needed advices that required medical knowledge.

Many thanks also to all of my friends who have supported and encouraged me to work on this thesis. I am certain; I wouldn‟t have made it without you. Thank you, Henock for all the cheering, and encouragement. It has always been a great experience working with you. Thank you Mulugeta, for the technical support, explaining to you what I was doing always made me question my taught. Thank you Fekadu, the motivation I got from you, kept me going for long. Thank you also for proof reading my little writings. Thank you so much Yonan, for being there from start. That necessary wakeup call was always yours.

Nebiyou, the ideas we shared were very valuable. Those sleepless nights wouldn‟t be better if it wasn‟t at your office. I could go for pages and pages if I wrote about all the people who have inputs to my work. I thank you all. Thanks to all of our instructors in the MSc. program. Personally, I have learnt a lot from you.

Last, I cannot possibly get past this page without mentioning the person who has paid dearly to get me to this point. Lee, you are truly the one. Thanks a lot. You have a very special place in this all.

Table of Contents

List of Figures

List of Tables

Abstract

Due to advances in word processing technologies, spell checking and grammar checks have become ubiquitous. However, checking conceptual errors from any given text has received little attention from similar researches. An example for conceptual error is contradiction.

The detection of conceptual errors on any plain text is a challenging task for computers. Two of the major challenges are the unstructured nature of text documents and the lack of representation of common sense and domain specific knowledge in machine understandable format.

Researches show that thousands of people are killed per annum due to medication errors. Medication errors are mainly caused by the omission of facts about the patient"s conditions or current drug intakes at the point of prescription.

To tackle this problem, many hospitals have implemented computer physician order entry (CPOE) systems. CPOE systems are capable of detecting drug – drug interactions at the time of prescription. However, most information about the patient is stored in the form of textual narration in the medical records. Information contained in the narrated record about the patient is therefore not used by CPOE systems to validate the subjected medications.

This thesis analyses sample medical notes to find patterns that could be explored towards using information encoded in plain medical notes for the detection of conceptual errors. The thesis proposes an ontology based architecture for a conceptual error detection system on medical notes. It also demonstrates results from the application of the prototype.

Chapter 1. Introduction

1.1. Background

Word processors are used in businesses, homes and educational institutions [2]. One of the advantages of modern word processing software is the ability to point out spelling and grammatical errors. Recent researches on spelling and grammar checkers have focused on detecting erroneously spelled words even though they exist in the dictionary [1]**.** A common example to what is caught by a modern spell checker is homophone (a word spelled correctly but at a wrong context).

Example: I will **sea** you tomorrow.

When most trivial spelling and grammatical errors are avoided by the usage of common word processing tools [3], verifying the conceptual correctness of the written material is totally left to the author. Checking and validation efforts so far have not gone beyond the word and sentence level [1].

Well spelled and grammatically correct statements may have conceptual errors. First, different statements in the same text could contradict with each other. Second, any statement could contradict with world known fact. Currently, little work has been done to automatically detect such conceptual errors.

Recent researches in the field of Information Extraction show that different approaches can be used to extract the concept out of plain text [4, 5]. But the focus of these researches was only with the information extraction and not in validation of the represented concepts.

Our hypothesis is that domain ontology can be used to validate concepts. Ontology is defined as the representation of different domain knowledge in machine process-able format [6].

Basically, Conceptual errors can be defined as contradictions that occur between concepts that are represented in a given textual document. Contradictions occur whenever information that is communicated in one or more sentences is incompatible. Incompatibilities are manifested in many ways [7]. Contradictions arise from relatively obvious features such as antonymy, negation, or numeric mismatches. They also arise from complex differences in the structure of assertions, discrepancies based on world-knowledge [8].

Based on the source and intensity of knowledge necessary to validate any given sentence, contradictions could be classified in to two categories.

1. Contradictions that occur between two concepts defined in the same document. Such types of contradictions only require the reader to have common sense or domain specific knowledge.

   Example: John is in London today.
   Today, John will spend his day on a balcony in Washington.

   Note: from the above example, we can judge that the two statements are contradictory because we know that John cannot be at both locations at the same date. In the medical texts, similar examples could be cited.

   Example: The patient is asthmatic.
   After examination, the patient took Aspirin for 3 days.

   These two sentences are contradictory in the eyes of the medical professionals because Aspirin should not be administered to a patient who is known to be Asthmatic.

2. Contradictions that occur between a given document and world known facts. The other classes of contradictions are contradictions that occur between information contained in sentences and world known facts.

   Example: George W. Bush was the 4th president of Canada in 2000.

   Note: The above example is only contradictory against world knowledge from which we know that George W. Bush was a president of the USA; not Canada.

Detection of contradictions in the first category requires only common sense or domain specific knowledge. On the other hand detection of the second classification of contradictions is dependent on comprehensive world knowledge.

Dick Crouch, Cleo Condoravdi, et al, [9] argues that the ubiquitous nature of ambiguity in natural language poses a great challenge to contradiction detection. Determining if two texts are talking about the same subject is also another challenge. Hence, [9] identified this challenge as reference resolution. The representation of domain and/or world knowledge is also important in determining contradiction. Paraphrases also make detection of contradiction difficult. An example to the contradiction caused by paraphrases could be “Y is solved by X” and “X doesn‟t overcome Y”.

Dick Crouch, Cleo Condoravdi, et al, [9] Point out that detection of contradiction should also include determination of intentionality. For example, given the following text: “x is faster than y” and a hypothesis “y is fast”; this pair shouldn‟t be considered contradictory because the hypothesis doesn‟t necessarily intend to argue y is faster than x.

The importance of contradiction detection is highlighted by the most recent acceptance it found in the research communities. In 2008, NIST (National Institute for Standards and Technology) organized the fourth Recognition of Textual Entailment challenge (RTE4). Among other requirements, RTE4 had a requirement to detect if the pair of statements had contradiction.

A few scenarios that could also benefit from a concept validation system could be:

•Fact checking: Fact checkers could use conceptual error detection systems to quickly check facts on texts they are supposed to check. Fact checkers are people hired by periodical publishers to check if article submissions are factually correct.

• Conceptual error detection on police detective reports such as testimonials, and other police reports could assist the police detectives in finding cues.

• Detection of contradiction in the preparation of Legal documents could reduce the chances of conflict later on.

• Detection of conceptual errors from policies and procedures could be of help for the implementing organization as it would save cost.

• Detection of contradictory standards is also an area that could benefit from a similar system. The more and more standards are put in an industry; it is not simple to cope up with all of those standards. Therefore, new standards could be subject to conceptual errors. Detection of such errors could be of great help to the implementers.

In the healthcare delivery process, a large amount of medical documentation is maintained to facilitate the communication and or storage of information about the patient. One classic example to this practice is the maintenance of clinical charts. Maintaining paper based clinical charts has a number of limitations. Recent technological advances in this area have brought electronic medical record keeping for each patient [10].

Electronic medical records partially address the problems posed by paper based documentation system. EMRs“ do not suffer from fragmentation of patient records, physical paper record missing or unreadable hand writing. However, most of the narrative information about the patient is still encoded in plain text form making the information unavailable for automated decision support system [11].

A number of research papers including [11, 33] suggest that a structured data entry is necessary to make the information available for easier reporting and decision support systems. However [11] reports that structured data entry has been proven to be difficult.

The focus of this thesis work is to detect three of the most common forms of contradiction that occurs between concepts that are represented within the same narrative medical note with respect to known medical rules. This work demonstrates that the application of simplistic as well as domain specific approaches can result in a valid conceptual error detection system. As will be seen in the literature review and related works section, currently there are no applications that propose usage of domain specific knowledge to contradiction detection from plain text. Therefore, this paper will explore usage of domain specific ontology for a better precision of a contradiction detection system.

1.2. Motivation

Detection of conceptual errors for general use makes sense in improving the quality of any writing. But, the availability of a comprehensive knowledgebase on general subjects and the complexity of the target language pose great challenges. However, the risk posed by conceptual errors in the medical domain calls for research in this area.

A major study, [12] reports that between 44,000 and 98,000 people die of medical errors in the United States alone, per year. Medical Errors could occur at any point in the healthcare delivery system. Researches [12] show that medication errors, surgical errors, diagnostic inaccuracies and system failures are the most common areas medical errors occur. However, we are more interested in preventable medication errors as it is an area that could benefit from a natural language processing application. This thesis work focuses on a tool that could help physicians avoid medication errors.

A very simple example in contradicting medical documentation could be a medical report that contains contradicting facts about the same patient; one part of a clinical note reporting that the patient has one condition while the other part of the document builds on a completely contradicting argument. Such conceptual errors could occur not only from a single man's error but also because of different procedural considerations. The same patient might be medically examined by different practitioners. Conceptual errors could also be introduced by usage of transcription service to convert audio medical reports to electronic medical records. Detection of such contradictions results in an improved quality of health care service.

1.3. Objective

The main objective of this research is to analyze conceptual errors manifested in medical notes and unlock information encoded in plain text medical documents to concept validation. Recognizing different patterns eminent in medical notes, the general objective of this research is to explore those patters to improve the general quality of medical documentations. Hence, the proof of concept prototype constructed to demonstrate this thesis should be able to detect different types of conceptual medical errors from a set of sentences extracted from sample medical notes.

This research should also focus on possible ways for the extraction of formal and machine process-able knowledge from the unstructured plain medical notes. This should allow for reasoning towards detection of conceptual errors.

1.4. Scope

This research will mainly focus on finding approaches to conceptually validate plain text medical notes. It will not deal with spelling or grammatical errors as those are dealt with other researches [1].

The scope of this research is limited to medical notes (documents) written in English. Even though conceptual errors exist in paragraphs and even in the bigger document, this research will focus on the basic reasoning task and is limited to validation of concepts at the sentence level.

1.5. Methodology

To achieve the main goals of this thesis work, a number of methodologies were applied. To gain a deeper understanding of the problem as well as to explore the possibilities, literature review was performed on two related areas. The major activities performed at this phase of the research are:

- Review of literature on avoidable medication errors, contraindications, their causes and classification.
- Review of literature on the logical definition of contradictions, the behavior and work performed on the detection of contradictions from natural language processing task point of view.

On the second phase of the research work, analysis of sample medical notes was performed with the help of domain experts. At this phase, 50 sample narrative medical discharge summaries were analyzed to study the pattern of appearance of important medical concepts in clinical notes.

A model and architecture to the validation of most significant conceptual errors in medication was proposed after the analysis of the problem and the structure of medical notes.

A proof of concept prototype was built following the architecture proposed on the previous phase. The outcome of the research is then evaluated on the proof of concept prototype. The evaluation was mainly performed to check the robustness of the system.

1.6. Organization of the thesis

This document is organized in eight chapters. The first chapter introduced the problem this thesis addresses. It also stated the motivation, scope and methodology used in the research.

The second chapter summarizes literatures reviewed for better understanding of the problem as well as available approaches towards solving similar problems. Chapter three is a report on recent and related work on contradiction detection.

Chapter four breaks down the classifications of conceptual errors based on the type and intensity of knowledge necessary to detect them. This chapter also discusses details of medication errors that are the very subjects of this thesis work.

Chapters Five and Six discuss the proposed architecture and implementation of the concept validation system. Chapter Seven discusses the evaluation performed on the concept validation system prototype.

Finally, chapter Eight concludes the paper by discussing the contribution of the work and future works related to this thesis work.

Chapter 2. Literature Review

This section starts with a brief summary of literatures on preventable medical errors. Related research areas and approaches to contradiction detection are also reviewed.

2.1. Medical Errors

The IOM (Institute of medicine) defines medical error as "the failure to complete a planned action as intended or the use of a wrong plan to achieve an aim." An adverse event is defined as "an injury caused by medical management rather than by the underlying disease or condition of the patient." Some adverse events are not preventable and they reflect the risk associated with treatment, such as a life-threatening allergic reaction to a drug when the patient had no known allergies to it. However, the patient who receives an antibiotic to which he or she is known to be allergic, goes into anaphylactic shock, and dies, represents a preventable adverse event [13].

Out of the estimated 44,000 to 98000 people that die because of medical errors, medication errors are the most common causes of medical deaths. These types of errors claim over 7000 patients a year in the US alone. Many others suffer from permanent disability because of similar types of errors. However, these types of medical errors are widely considered as the once that are the most preventable [14].

2.1.1. Preventable Medication Errors

These are preventable mistakes in prescribing and delivering medication to patients. Such mistakes include prescribing a drug to which the patient is known to be allergic to or two or more drugs whose interaction is known to produce side effects[15].

Research by AHRQ (Agency for Healthcare Research and Quality) supported investigators is helping to characterize these errors (called preventable adverse drug events, or ADEs) and suggest how to prevent them. These researches reported that in a study of inpatient care in two tertiary care hospitals, errors in ordering medicines accounted for 56 percent, of preventable adverse drug events. Another finding showed that dosage errors, in particular, were primarily due to the physician's lack of knowledge about the drug or about the patient for whom it was prescribed [15].

The American Hospital Association lists the following as some common types of medication errors: [16]

- incomplete patient information (not knowing about patients' allergies, other medicines they are taking, previous diagnoses, and lab results)
- unavailable drug information (such as lack of up-to-date warnings)
- miscommunication of drug orders, which can involve poor handwriting, confusion between drugs with similar names, misuse of zeroes and decimal points, confusion of metric and other dosing units, and inappropriate abbreviations
- lack of appropriate labeling as a drug is prepared and repackaged into smaller units
- Environmental factors, such as lighting, heat, noise, and interruptions that can distract health professionals from their medical tasks.

However, [17] points out that nearly half of all adverse drug events have some form of "preventability" and many do not represent errors of commission but, rather, errors of omission. This implies a failure on the part of someone (pharmacist, physician, patient, or the interactions between these groups) to detect certain factors that most likely led to the adverse event. These factors include:

1. Failure to detect a disease state, contraindication to the drug therapy;
2. Failure to detect a significant drug interaction;
3. Failure to detect a significant drug allergy;
4. Failure to prescribe the correct dose for a specific patient;
5. Failure to monitor drugs with narrow therapeutic indexes; and
6. Patient knowledge deficits.

Reviewing the electronic medical record may improve detection of errors and adverse events by monitoring in "real time" and by integrating multiple data sources (e.g., laboratory, pharmacy, and billing) [18]. This paper recognizes that the use of computers to search the electronic medical record can find errors and adverse events not detected by traditional chart review or provider self-reporting.

2.2. Contradictions

Little work has been done towards detection of contradiction. However, [8] observed that contradiction occurs when two sentences are extremely unlikely to be true simultaneously. For two sentences to be contradictory, they need to refer to the same event. However, determining if two sentences are co-referent is probabilistic rather than certain. This problem was also identified by [9] as reference resolution problem.

Two categories of contradictions have been identified by [8]. The first category includes contradictions that occur because of the usage of antonyms, negation and numeric features. This category of contradiction is relatively easy to detect. The second category of contradictions contains contradictions that need world knowledge or commonsense knowledge to detect.

Textual Entailment is formally defined as a relationship between a coherent text **T**, and the hypothesis **H**. **T** is said to entail **H** (**T** → **H**) if the meaning of **H** can be inferred from the meaning of **T** [19]. Textual entailment recognition is therefore the process of determining if a given natural language text is inferred from semantic of another one.

[9] Recognized detection of entailment and contradiction relation between texts as a minimal matrix for the evaluation of text understanding. After this observation, Recognition of Textual Entailment (RTE) came as a research area that focuses on the sole task of recognition of entailment between a given text and a Hypothesis. Recognition of Textual Entailment involves processing of the sentences at the lexical level as well as **syntactic** and **semantic** level.

To encourage researchers in this area, a number of RTE challenges were performed. To the date of this writing, four RTE challenges were performed and the 5^{th} RTE challenge has been announced to be performed in 2009. While the main theme of the challenge stayed the same, it has evolved by adding extensional task with each iteration.

At the announcements, researchers who chose to participate in the RTE challenge were provided with a training set of sentences in the form of Text (T) and Hypothesis (H). In the most recent challenge these sentence sets were manually tagged as Entailed, Contradictory or Unknown. After training their system using the test set, researchers are expected to test their system on a test set provided by the organizers.

Starting from RTE 3, which was performed in 2007, the researchers had the option to participate in a two way challenge or on a three way challenge. The two way challenge is when the researchers submit the classification in to only two classes; Entailment or unknown. The three way challenge is when the researchers choose to submit the classification in the form of Entailment, Contradiction or Unknown.

The submissions were finally evaluated by the agreements they have with manually tagged test set. However, [20] in 2008 made an observation that most of the systems find it difficult to detect contradictions.

In this thesis, the main focus is on conceptual error detection but we will consider the approaches used for Textual Entailment as both are types of textual Inferences.

A number of approaches have been tested to the recognition of textual entailment. Many of the approaches are observed using machine learning techniques by considering entailment and non-entailment problem as a classification problem. A few papers have proposed and tested Logical inference approaches. This section discussed these approaches briefly.

2.2.1. Logical Inference

[21] Presented a RTE system that works by using logical inference. First, the authors used a system called BLUE (Boeing Language Understanding Engine) to perform a full semantic interpretation of both sentences. Then, they used knowledge obtained from WordNet and DIRT paraphrase database to infer a relationship.

WordNet is one of the most used lexical resources in NLP. It organizes words in semantic networks: the nodes, synsets, represent senses, and contain a number of single or multi-word terms which have the same or very similar meaning; the edges represent different types of semantic relations, such as hyponym-hypernym, meronym-holonym, antonymy, cause-effect and pertains to. Among the members of a synset the synonymy relation holds. By far the most commonly used relation in WordNet is the hyponym-hypernym, which gives a taxonomic view of the resource [32].

DIRT (Discovery of Inference Rules from Text) is an unsupervised method of discovering inference rules from text [31]. DIRT is a generalization of the Distributional Hypothesis, which states that words that occurred in the same contexts tend to have similar meanings. Instead of applying the Distributional Hypothesis to words, It is applied to paths in dependency trees. Essentially, if two paths tend to link the same sets of words, the hypothesis is that their meanings are similar.

During parsing, BLUE generates a semi formal structure called Logical Form (LF). The LF is a simplified tree structure with logic type elements generated by rules parallel to the grammar rules. Below is an example to the logical form generated by BLUE [21].

```
;;; LF for "A soldier was killed in a gun battle."
(DECL
    ((VAR _X1 "a" "soldier")
     (VAR _X2 "a" "battle" (NN "gun" "battle")))
    (S (PAST) NIL "kill" _X1 (PP "in" _X2)))
```

The LF is then used to generate ground logical assertions of the form r(x ,y) by applying a set of syntactic rewrite rules. The final logic for the above example could be seen below.

```
;;; logic for "A soldier was killed in a gun battle."
object(kill01,soldier01),
in(kill01,battle01),
modifier(battle01,gun01).
```

After generating the logic from both the text T and the hypothesis H, the system uses a set of rules to determine if H entails T. The simplest example of the applied rules is that of "subsumes" relation. Consider the following example. The logic of "A person likes a person" subsumes the logic of "A man loves a woman". These kinds of relations between sentences are considered entailment.

In addition to this, the system also measures the logical equivalence of sentences regardless of differences in part of speech. Hence, "A person attacks with a bomb" and "There is a bomb attack by a person" are recognized as equivalent. Although the heuristics used by the system could go wrong, these kinds of measures could be used to measure simple equivalence and sub-sumption.

2.2.2. Probabilistic or classification based approach

Logical Inference could result in a great explanation for detected entailment or none thereof. But it suffers from performance loss from lack of robustness. For this reason, many submissions of the RTE challenges have demonstrated a different trend. Many submissions use machine learning techniques by considering the RTE problem as a classification problem. Most of these submissions differ with each other in feature selection [20]. [22] Present a number of features they extract so that the machine learning algorithm uses them to decide if the given set is a Textual Entailment or not.

Ontology Alignment for RTE

[23], with their submission to the RTE 4 challenge presented a system that works by aligning ontology"s acquired from the given text (T) and hypothesis (H). To achieve this, their system performed three separate processes.

Ontology Acquisition: On this phase, The system finds a formal representation of the given text (T) and the Hypothesis(H). This formal representation is based on a description logic type of ontology. To generate the formal representation of the two texts, the system first performs a syntactic analysis with a parser called Minipar. Minipar is used to generate the dependency relations. These dependency relations are transformed to a semi-Semantic structure using a set of transformation rules.

After the syntactic analysis, the system performs a semantic analysis to convert the semi-semantic structure to a semantic structure. On this process, the system uses RoDEO, a Named Entity Recognition system developed by the same authors in 2008. Apart from recognizing the named entities in the given text (T) and Hypotesis (H), RoDEO finds the class of the named entity. [24] RoDEO, upon initial evaluation, classified the 1019 named entity to 678 classes. This is far more specific classification than any other existing classifiers.

On the semantic analysis, the RTE system also tries to preserve the semantic relation of adjectives and nouns modifying other nouns. To achieve this, the system preserves the verbs that represent commonsense knowledge. For example, in "Jurassic Park is a novel written by Michael Crichton." The system knows that „Jurassic Park" and „Michael Crichton" are named entities. By using the RoDEO system, it also knows that Jurassic Park is a Book and that Michael Crichton is a writer. The system then tries to find the verb that relates the two classes; „writer" and „book". After the system selects the verb „write" as the one that relates the two terms, it divides the semi semantic tuple to represent the relations.

Subject (write, Michael Crichton |noun - writer) and

Object (write, Jurassic Park |noun - book)

As a last step to the ontology acquisition phase, the system performs an ontology analysis on the output of the semantic analysis activity. At this stage, the system generates two OWL ontology‟s containing the semantic representation of the text (T) and the Hypothesis (H)

Ontology Alignment

After the two ontology‟s are generated on the ontology acquisition phase, the RTE System performs a two-step operation to align them to create an aligned ontology Ontology-A. First, classes are aligned to create equivalent classes. On the second step, the properties are aligned to create equivalent properties.

Textual Entailment

Data collected on the ontology alignment phase is used to decide if there is a textual entailment relation or not. To achieve this, the system integrates WEKA, an open source machine learning package, to make the decision if there is an entailment relation or not. WEKA is trained with RTE3 test set. The authors selected three machine learning algorithms for their entry on the RTE4 completion. WEKA B40 decision tree classifier was however the one algorithm that resulted in the best performance of 68% on their 2 way RTE4 challenge submission.

Even though all the above approaches have been applied on the RTE task, [25] argues that there is a tradeoff between informatively and robustness.

Informatively is the ability of a system to take into account all available relevant information.

Robustness is the ability of a system to proceed on reasonable assumptions, where relevant information is missing.

Deep logical techniques like logical inference are informative but not robust. Shallow techniques like bag of words overlap method are robust but not very informative. [25] Characterizes most of the RTE systems as somewhere between deep and shallow techniques and therefore suffering from Informativeness and robustness tradeoff.

Chapter 3. Related Work

There are a number of works that are related to the work in this thesis. A few of the related applications range from contradiction detection systems to computer physician order entry systems that validate prescribed medications. This chapter reviews these related areas of research and the outcome of the works performed.

A contradiction detection application is built at Xerox concerning quality maintenance for document collections. The system includes a large textual database containing engineer-authored documents (tips) about the repair and maintenance of printers and photocopiers. Over time, duplicate and inconsistent material builds up, undermining the utility of the database to field engineers. Human validators who maintain the quality of the document collection would benefit from Entailment and Contradiction Detection (ECD) text analysis tools that locate; points of contradiction and entailment between different but related tips in the database [9].

The ECD system uses detailed, and hand written syntactic and semantic rules to analyze the given text. This work was concluded by making the observation that construction of such a system is feasible. However, the authors relied on logical definition of the phenomenon and did not report any empirical result for their system [8]. De Marneffe, Marie-Catherine et al. Constructed a contradiction detection system that uses predefined features to classify the contradictions and the non-contradictions. To find patterns of contradictions, the authors selected a few linguistic features [8].

The authors of this system first used a simple alignment strategy to identify if two sentences are co-referent or not. After filtering the co-referent sets of texts, their system extracted features on which they applied logistic regression to classify the contradictions. Some of the extracted features to characterize contradiction include; Polarity, Number date and time features, Antonymy features, structural features and factivity features, modality and relational features.

The polarity features capture the presence (or absence) of linguistic markers of negative polarity. The authors argue that polarity differences are good clues to contradiction. Aligned antonyms can also be seen as clues to contradiction. Polarity feature could be extracted by running an analysis on the dependency tree of both the text and the hypothesis. To extract antonymy features, the system uses WordNet and VerbOcean to make a list of contrasting words. Once the contrasting words are identified from the text and hypothesis, their polarity is also put into consideration to determine contradiction.

For the extraction of numeric features, the system first normalizes dates and numbers. At this stage it represents numbers as range. This enables the system to correctly recognize that „over 100" doesn"t contradict 200. Numeric features are considered contradictory when they are incompatible and when the surrounding words match.

This work demonstrated that detection of contradictions of type"s antonym, number and polarity is feasible using current systems. It makes the observation that detection of world knowledge and factive contradictions is difficult. However, concluding on their system, the authors report that the low accuracy achieved on new text renders their system in-practical.

Another notable work from the RTE challenge submissions is The BLUE system. BLUE, as described in the literature review section is a system that uses Logical inference to recognize Entailment and Contradiction. This system was put to test on the fourth Recognition of Textual Entailment challenge in 2008. The authors reported that the output of their system was 56.5% correct with answer to the 2 way challenge. The two way challenge was a challenge to identify a pair of sentences as Entailment or No Entailment (Unknown). On the three way challenge however, the authors reported that they achieved 48.1% accuracy at best in identifying if the two sentences are Entailed (Yes), Contradictory (No) or Unknown.

BLUE clearly identified the cases for which it could determine entailment or contradiction. For the pairs BLUE could determine entailment or contradiction for, it was correct 65% of the times. The most significant problem with this system, according to the authors was the lack or limitation of knowledge. The challenge required lexical and world knowledge that varies from core knowledge to domain specific task. For example, BLUE incorrectly predicted unknown for T: "**A bus collision** … resulted in … 30 **fatalities**." H:"30 were **killed** in a **road accident**.". Here, the system needs to realize that a bus collision is a road accident, an operation that needs more than a simple hypernym check.

In the clinical practice research, computer physician order entry (CPOE) programs are proposed to improve the error prone process of prescription. Using CPOE, Illegibly handwritten prescriptions, drug interactions and incorrect dosages are detected [14]. However, these programs mainly depend on structured data entry. With lack thereof clinical structured data, including [11], information encoded in narrative format is rendered useless.

The work detailed in this thesis, however, is different in that it uses a robust algorithm to use information encoded in plain text narrative medical notes towards the validation of conceptual errors.

Chapter 4. Approach

Conceptual errors could be classified into two broad categories depending on the type of knowledge needed to detect. We have classified the conceptual errors that require comprehensive world knowledge and the other category as requiring the reader to have common sense or domain specific knowledge to validate. However, conceptual errors that occur between concepts represented in the same document could further be classified depending on the specific type of knowledge representation required to validate. For example conceptual errors that occur because of wrongful usage of antonyms could be validated using only a dictionary [7, 8]. On the other hand numeric contradictions couldn‟t use the same dictionary to validate.

This section presents a logical classification of conceptual errors that occur between concepts represented in the same document. This classification is used to further narrow down on the problem this thesis tries to solve. Based on the type of knowledge representation requirement for validation, we have classified conceptual errors in to 5 different categories. This section also breaks down the types of conceptual errors that are mostly exhibited in medication errors.

4.1. Classification of conceptual errors

This section discusses the theoretical classifications applied in this thesis towards conceptual errors. In general, we classified conceptual errors that occur in a given document from the knowledgebase requirement point of view:

- Contradictions that occur because of usage of synonyms, antonyms and negation.
  - Abebe is a boy.
  - Abebe is a girl.
- Nativity contradictions.
  - Abebe was born in Addis Ababa.
  - Abebe was born in Washington.
- Temporal contradictions (contradictions that occur because of the sequence of events in time)
  - Abebe was dead on 12:00AM June 3rd.
  - Abebe was playing football on 12:00AM June 4th.
- Factual contradictions

    - Abebe is the Judge in the Teddy afro hit and run Case.
    - Abebe was called to the stand to give a testimony on the Teddy afro hit and run case.
- Numeric contradictions
    - Abebe is 2.00 meters tall.
    - Abebe is 1 meter and 83cms tall.

Detection of contradictions in each of the above classes of conceptual errors is difficult because of the unstructured nature of natural language. It is also difficult because of the kind of general knowledge it requires to detect the contradictions. It requires having domain specific knowledge to validate nativity contradictions when it requires completely different type of knowledge to validate temporal contradictions.

The following is a simplistic example of how detection of contradictory concepts could be complicated.

- Contradictions that occur by usage of synonyms, antonyms and negation
    - Abebe is a boy.
    - Abebe is not a girl.

  We can observe here that the Antonym is negated, hence there is no contradiction. Different negation patterns pose a great challenge in identifying these kinds of contradictions. Another example for this could be,
    - Abebe is a boy.
    - Both Abebe and Kebedech are not boys, the second is a girl, the first is not.

  In this case we can observe that the sentence is a valid sentence and does not contain any contradiction. However, it poses complexity to detect using simplistic methods.

- Nativity contradictions.
    - Born in Addis Ababa, Abebe is now living in the US.
    - Abebe was born in Ethiopia.

  In this example, Human reader could easily decide that the two sentences are not contradictory. However that is only possible assuming the human reader has the basic geographic knowledge that Addis Ababa is a city in Ethiopia. An algorithm that uses word overlap to detect contradiction could easily detect this as contradiction.

- Temporal contradictions (contradictions that occur because of the sequence of events in time)
    - Abebe was dead on 12:00AM June 3rd, 2009.
    - Abebe was playing football on 12:00AM June 4th, 2009.

  From our common sense knowledge we know that a dead person could not play football after his/her time of death. Representation of this common sense knowledge so that it could later be used to make these kinds of validations is difficult.

- Factual contradictions
    - Abebe is the Judge in the Tedy afro hit and run Case.
    - Abebe was called to the stand to give a testimony on the Tedy afro hit and run case.

  This kinds of contradictions are the most difficult to detect because of the enormous amount of factual possibilities. A quick demonstrator to this is the fact that Abebe could be reported as doing anything except being the judge and the witness at the same time.

- Numeric contradictions
    - Abebe is 2.00 meters tall. Abebe is 6.5 feet tall.

  In this Example also, we know that 2.00 Meters is equal to 6.5 feet. This knowledge has to be taken into consideration when evaluating if the two sentences are contradictory or not.

However, this work is specifically focused on conceptual errors that are prominent to the medical domain. As indicated in the literature review section, medication errors are areas majorly affecting the healthcare quality. Therefore the rest of this paper will outline approaches to deal with such problems from Natural language understanding point of view.

4.2. Analysis of medication errors

Through discussions with domain experts and knowledge acquired when performing the literature reviews, the following scenarios have been identified to be the areas the concept validation system could work best on.

- Wrong diagnosis

Wrong diagnosis could occur because of different factors. Two of the most prominent factors that could benefit from a concept validation system are:

- Contradictory diagnosis

  These kinds of errors are exhibited when a patient is diagnosed with two different diagnoses and when they both cannot be true at the same time.

- Missed adverse drug effects

  A medication might be administered to a patient even when it has been observed that the patient exhibited some adverse effects. Such errors could lead to more adverse effects unless the physician does act on them accordingly.

- Wrong prescription

Wrong prescription of drugs occurs when the wrong medication is prescribed to the wrong patient. Many reasons contribute to wrong prescription of drugs. A few of the most common reasons are:

- Unnoticed documented allergies

  Medical charts grow bigger and bigger each time a patient visits a healthcare facility. Often written and documented allergies to specific medicine pass unnoticed by the physicians. This could lead to the wrong medication being prescribed to the patient.

- Food drug interaction

  Even though there is no customary practice to document food intakes of patients, it is important to note that many medicines interact with food. Therefore, physicians pass written communication to alert caretakers. However this process is still prone to error because of un-considered medicines.

- Other Reasons
  - Drug name confusion

    Some drugs have similar names that often get confused either by the physician or the pharmacist. A classic example to such errors is the prescription of Lamisil for epilepsy while the right medicine is Lamictal.

  - Medicine interaction

Not noticing that the patient is already on some form of medication, a physician prescribes another medication. Drug interactions are often documented and communicated to practitioners. Attributed to many factors, physicians often prescribe interacting medications that leads to adverse effects. These kinds of errors are considered preventable.

- Over or under dosage
  Physicians make numeric errors that lead to adverse drug effects. This is often caused by omission of patients profile like the age, pregnancy status, current drug use etc.

4.3. Analysis of medical notes

For a better understanding of the text contained in medical notes, analysis of 50 sample discharge summaries was performed. The discharge summaries were randomly selected from an online transcription service provider [26]. With the help of a medical domain expert, all mentions of disease, medication and conditions were marked.

After the occurrence of these concepts was marked, the next step was to analyze and try to observe the pattern of appearance of the important concepts in the medical notes. The 50 discharge summaries were composed of 1596 sentences. In these sentences 901 disease and conditions were mentioned. On the other hand, 327 clinical drugs were also mentioned.

The mentions of diseases were then analyzed as follows.

Medication and Disease

98% of the mentions of diseases, known conditions and medication have a direct relationship with the current patient.

Example:

> *The patient is a 45-year-old male complaining of abdominal pain. The patient also has a long-standing history of* ***diabetes*** *which is treated with* ***Micronesia*** *daily.*

Here, it can be observed that the patient has a known condition of diabetes, and that the clinical drug Micronesia is being administered to him. This kinds of sentences being the most dominant in the analysis, it cannot be concluded that all appearances of diseases and clinical drugs have direct relationship with the current patient.

Negated Disease

Disease mentions, having a direct relationship with the patient, at times appear to negate the existence of the disease or condition on the patient.

Examples:

> *He denied **diarrhea, dysuria, hematuria**, urgency and frequency. He denied any history of rash.*
> *Abdomen: Normal bowel sounds. It is soft and nontender without **hepatosplenomegaly***
> *PAST MEDICAL HISTORY: None. No history of **hypertension**, **diabetes**, heart **disease**, **liver disease** or **cancer**.*
> *I think this is still his allergic rhinitis rather than a **sinus infection**.*

Allergies

Mentions of the clinical drugs sometimes are to indicate the existence patient‟s allergic condition towards the clinical drug.

Examples:

> *This is a 1-year-old male who comes in with a cough and congestion because of his allergic reaction to **aspirin**.*
> *ALLERGIES: **Sulfa**, **aspirin**, **Darvon**, **codeine**, **NSAID**, **amoxicillin**, and **quinine**.*

Negated Medication

Medications could also be mentioned with the intent of indicating that they were not administered to the patient.

Examples:

The patient was ordered ***Demerol*** *50 mg IM for pain, but he refused and did not want pain medication.*

Naprosyn *was ruled out because she is allergic to it.*

Common Characteristics on medication and diseases

From the analysis above, we can conclude that most sentences in the medical notes are written about a single patient, the patient"s medical conditions and medications. However there were exceptions where we noted that the sentences were written about family medical history. On extremely rare conditions, we have also noticed that some sentences were written about other patients.

Among the analyzed sentences, we have identified one case where the patient was accidentally injected with a used needle. On that specific medical note, there were sentences written to explain the medical condition of the patient the needle was first used on.

Occurrences of clinical drugs were mostly characterized as prescribing or discussing the administration of the clinical drug. Some of these sentences described allergic condition towards medications. Terminologies that denoted diseases and conditions were mostly observed as describing the presence or none thereof of the condition.

Chapter 5. Architecture

This section discusses the architecture of the proposed concept validation system. The proposed architecture attempts to explore the pattern observed on the analysis of the medical documents. The pre-processor, the knowledgebase, the ontology extractor and the reasoner are the main components in the architecture of the concept validation system.

The input to the concept validation system is a sequence of sentences. These sentences are initially passed to the pre-processer. The pre-processer is responsible for filtering out sentences deemed un-important in the concept validation process.

The knowledge base is a store for background knowledge as well as restrictions that are applied in the concept validation. The background knowledge in the knowledge base component of the concept validation system is a standard representation of medical concepts. This standard representation along with the standard classifications that apply on the represented concepts will be used in different stages of the concept validation process.

After the pre-processing, the sentences are passed to the concept mapping component. The concept mapping component is responsible to identifying medical concepts from the sentences and mapping them to the standard representation in the knowledge base.

The output of the concept mapping component is passed to the ontology extractor. The ontology extractor is a component in the concept validation system that constructs ontology of the concepts represented in the medical note. The very reason for extracting the ontology from the text is to reason on a structured representation of the information in the unstructured plain text.

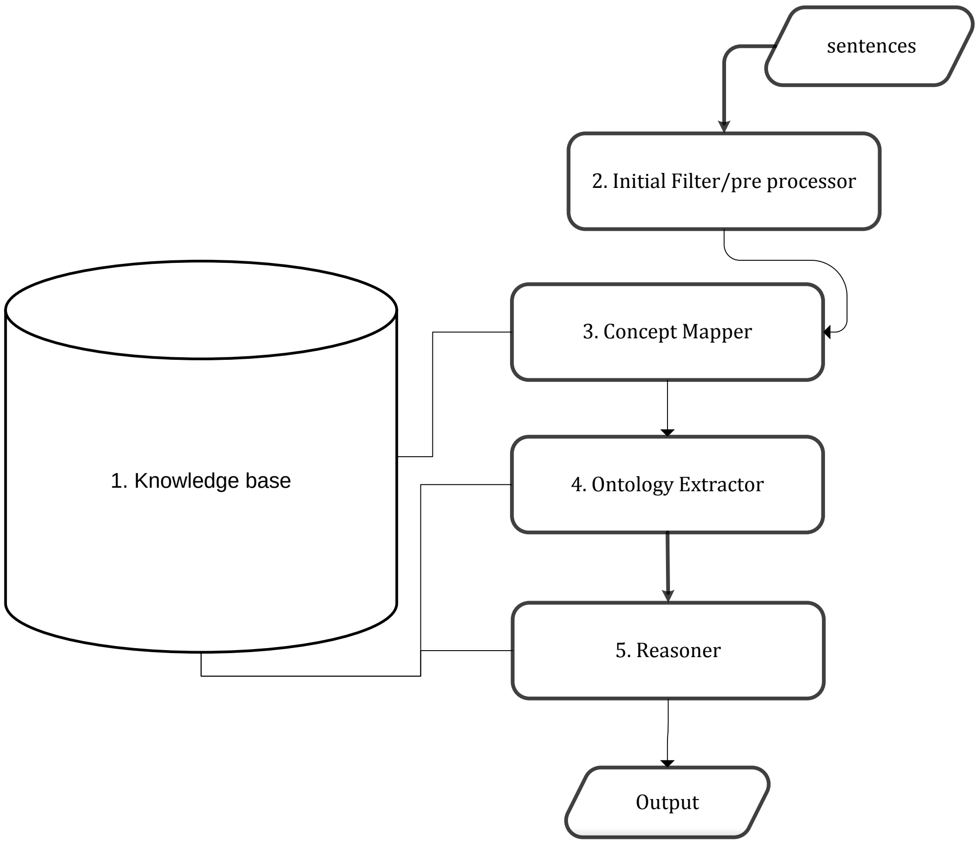

Figure 5-1 Architecture of the concept validation system

As can be seen in figure 5.1, the output of the ontology extractor is then passed to the reasoner. The reasoner component in the system is responsible to using domain specific restriction knowledge to validate the ontology representation of the information.

Detail of these components is discussed in the upcoming section.

5.1. The Knowledge base

This component of the system is a data store to knowledge deemed important for the validation of concepts in the validation system. Hence, it stores both background knowledge as well as knowledge necessary to identify errors.

A concept could be represented using different terminologies. A term could also be spelled differently among cultures. A demonstrator to this could be the British and US English spelling differences. There is also morphological difference in appearance of the term. For a successful understanding of the medical notes, we need to bridge this gap. We need to be able to identify exact concepts from the given text regardless of these differences.

It is also important to be able to represent the concepts in a standard format so that restriction rules could be represented in a universally accessible format. To this end, a Thesaurus of terminologies in the domain of interest will suffice. This thesaurus will therefore serve as background knowledge. The knowledge base also contains the rule database that is used to represent only the restriction that shall not be violated. The restriction rules could be defined as conditions or events that should not occur at the same time.

An example to restriction rules could be medicine x could not be administered to a patient along with medicine y. Hence, a detailed list of contra-indications should be represented in the restriction rule store so that it will be used later. This restriction rule should be transcribed using unique concept IDs from the thesaurus. This is done because we do not want to miss precision in light of lexical and morphological differences in appearance of the terminology.

The rule database has to be implemented in such a way that incorporation and or editing of existing rules is simple to the domain experts. It should be noted rules represented in the rule store are basically knowledge from the domain experts.

The last functionality of the knowledge base is to give frame of reference to the validation task. Hence an upper level ontology is proposed to be feasible to act as a frame of reference. A more comprehensive and broad ontology might enable more reasoning capacity but the resource requirements and complexity of such an ontology makes it infeasible to implement.

5.2. Initial Filter / Pre processor

The initial filter component of the concept validation system shall be able to filter only informative and important sentences for the rest of the components. Therefore, the main purpose of the initial filter component is to try to determine insignificant and noisy sentences. These sentences are discarded at this stage so that their noises will not affect the outcome of the concept validation system.

Questions and sentences referring to family medical history in medical documentation are the most significant noisy sentences we have identified so far. Hence, using their patterns, this component shall discard these noisy sentences.

5.3. Concept mapping component

This component of the concept validation system is responsible to the recognition of important medical concepts from the sentences and to mapping them to a standard conceptual representation.

It is clear that concepts could be represented using different terminologies. Terms could also appear in different lexical and morphological forms. Hence, this component should put these differences as well as spelling differences in different cultures when mapping the concepts to standard representation.

To an effective mapping of terminologies in the text, all lexical and semantic information from the sentences should be put to use. To this end, the mapping component should be able to disambiguate which concept is represented using the part of speech the sentence appears in.

An example to the disambiguation task could be the recognition of “Cold” as a disorder from the following sentence.

The patient has cold flu.

His left foot is cold.

From the first sentence we can conclude that cold is a disorder while in the second sentence cold is just a finding. Hence, the mapping component should be able to use the part of speech the term cold appears in before mapping it to a standard format from the knowledge base.

The mapping component should also be able to find the categories of the concepts identified in the sentences. The high level categories the concept validation system is interested in identifying are clinical drugs, diseases, syndromes and allergic conditions.

5.4. Ontology Extractor

The main aim of this component is to convert the unstructured text input into a structured format that is suitable for the reasoning that will be performed on a later stage. The ontology extractor uses the upper level ontology from the knowledge base to the extraction of knowledge from the mapped text.

The main principle of this component is creating instances of the upper ontology classes and creating relationships between them. By default, we have learnt that the medical documentation is written about a specific patient. Therefore the first step of this component is to create an instance of Patient class. This instance is given the label "current patient" and will be referred to in different sections of the extracted concept.

Using the semantic categories as criterions, the ontology extractor selects only important concepts from the mapped text. Only clinical drugs, diseases, syndromes and known conditions are filtered from the mapped concepts. The ontology extractor shall create instances of clinical drug or disease with the label of standard notation from the thesaurus.

After creating instance of diseases and clinical drugs, the concept extractor has to determine if the created instance and the current patient have relationship. As observed in the analysis of sample medical notes, the relationship between the current patient and the disease could either be the patient being affected by the disease or to rule out the existence of the condition on the patient.

The ontology extractor should also determine if there exists any relationship between the patient and the clinical drug. By default, the component could assume that the clinical drug is administered or is to be administered to the patient. But in special cases, the extractor should acknowledge patterns of text that indicate the non administration of the clinical drug. It also should recognize if there are indicators that indicate an allergic condition the patient has towards the clinical drug.

After the recognition of the relationship between the patient and the identified concepts, the component should create the identified relationships in the extracted ontology.

This component of the system should be designed in such a way that it could be extended in the future to extract additional types of variables so that they could be used in the reasoning phase.

5.5. Reasoner

The reasoner is the decision making component of the concept validation system. This component takes the extracted knowledge and the restriction knowledge from the knowledge base to produce the output.

Chapter 6. Implementation of the prototype

A proof of concept prototype was developed in accordance with the proposed architecture of the concept validation system. This prototype was implemented to demonstrate and measure if a concept validation system is applicable in a selected few conceptual errors on medical notes studied above.

The implemented prototype is mainly limited to the extraction of concepts and validation of the following types of conceptual errors.

- Drug – drug interaction
- Drug – known allergic condition and
- Drug – known medical condition interaction in medical notes

The selection of these types of conceptual errors was done considering the severity of their effect. Another factor under consideration was the resource requirements they have. This chapter elaborates how the tools, components and approaches applied to the implementation of the proof of concept prototype.

6.1. Tools

Different tools and platforms were used towards the development of the prototype. Java programming language was used to implement the system. The system was tested on an Ubuntu Linux distribution.

The upper level ontology development was performed using protégé editor. The protégé editor is used to create the classes and the relationships between them. This ontology is exported as an OWL file for use in later stages of the development.

In addition, Jena, a Java based framework was used for the development of the reasoning component of the system. Furthermore, the rule-store was implemented on a MySQL database.

The other two big components used in this system are the UMLS Metathesaurus and MetaMap. These components are better explained in next sub sections.

6.2. Implementation

In accordance with the architecture of the proposed system, Figure 6.1 shows what components were used to perform the functionalities described in the architecture.

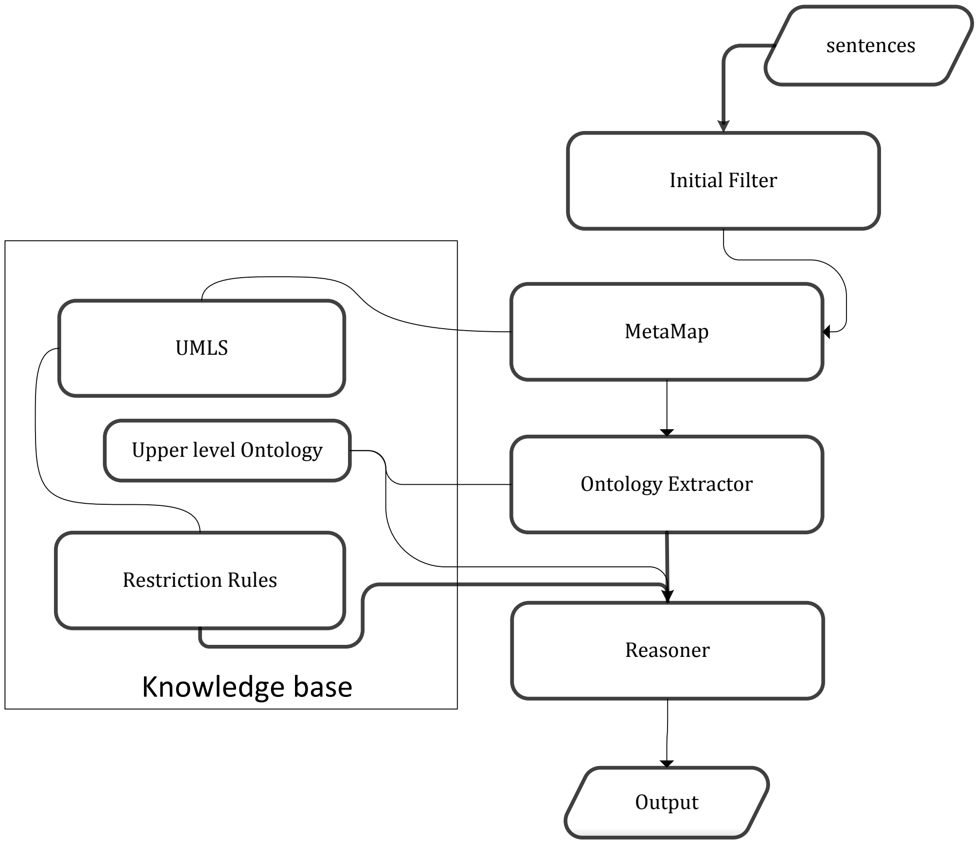

Figure 6-1 Implementation of the architecture

The Knowledge Base

The knowledge base component of the system is composed of three separate and interoperable components. The first one is the UMLS. The UMLS is a Unified Medical Language System that incorporates three types of information. These are the Meta thesaurus, the semantic network and the Lexicon.

- **The Metathesaurus**

The Metathesaurus is a very large, multi-purpose, and multi-lingual vocabulary database that contains information about biomedical and health related concepts, their various names, and the relationships among them. The Metathesaurus supplies information that computer programs can use to create standard data, interpret user inquiries, interact with users to refine their questions, and convert the users' terms into the vocabulary used in relevant information sources.

The Metathesaurus is organized by concept or meaning. In essence, its purpose is to link alternative names and views of the same concept together and to identify useful relationships between different concepts. All concepts in the Metathesaurus are assigned to at least one semantic type from the Semantic Network. This creates a consistent categorization [27].

- **The Semantic Network**

The Semantic Network consists of a set of broad subject categories, or Semantic Types, that provide a consistent categorization of all concepts represented in the UMLS Metathesaurus. Major groupings of semantic types include organisms, anatomical structures, biologic function, chemicals, events, physical objects, and concepts or ideas. The links between the semantic types provide the structure for the network and represent important relationships in the biomedical domain.

- **The Lexicon**

The UMLS also contains the lexicon entry for each word, or term. This entry records syntactic, morphological, and orthographic information. Lexical entries may be single or multi-word terms. Lexical information includes syntactic category, inflectional variation (e.g. singular and plural for nouns, the conjugation of verbs, the positive, comparative, and superlative for adjectives and adverbs), and allowable complementation patterns [28]**.**

UMLS is used as background knowledge in the concept validation system. Hence the information used from the UMLS could not exceed serving as a standard to the extraction and representation of restriction rules.

The second component of the knowledge base in the concept validation system is the **upper level ontology**. The main purpose of this ontology is to assist in the concept extraction and reasoning tasks in the concept validation system. This ontology is designed to only contain abstract knowledge important to the validation task.

The upper ontology contains abstract classes that are to be found in the medical notes and hence should only contain relationship types that could be pertinent to the validation task. Figure 6-2 on next page shows an extract of the upper level ontology developed for the prototype.

The upper ontology development was performed in a two step operation. On the first step, all the concepts deemed important to the concept validation were identified. Hence, six high level concepts were identified as important.

Person, Clinical Drug, Disease, Finding, Physician and Patient were the 6 high level concepts. After identifying these concepts, the next process was to identify relationships between these high level concepts.

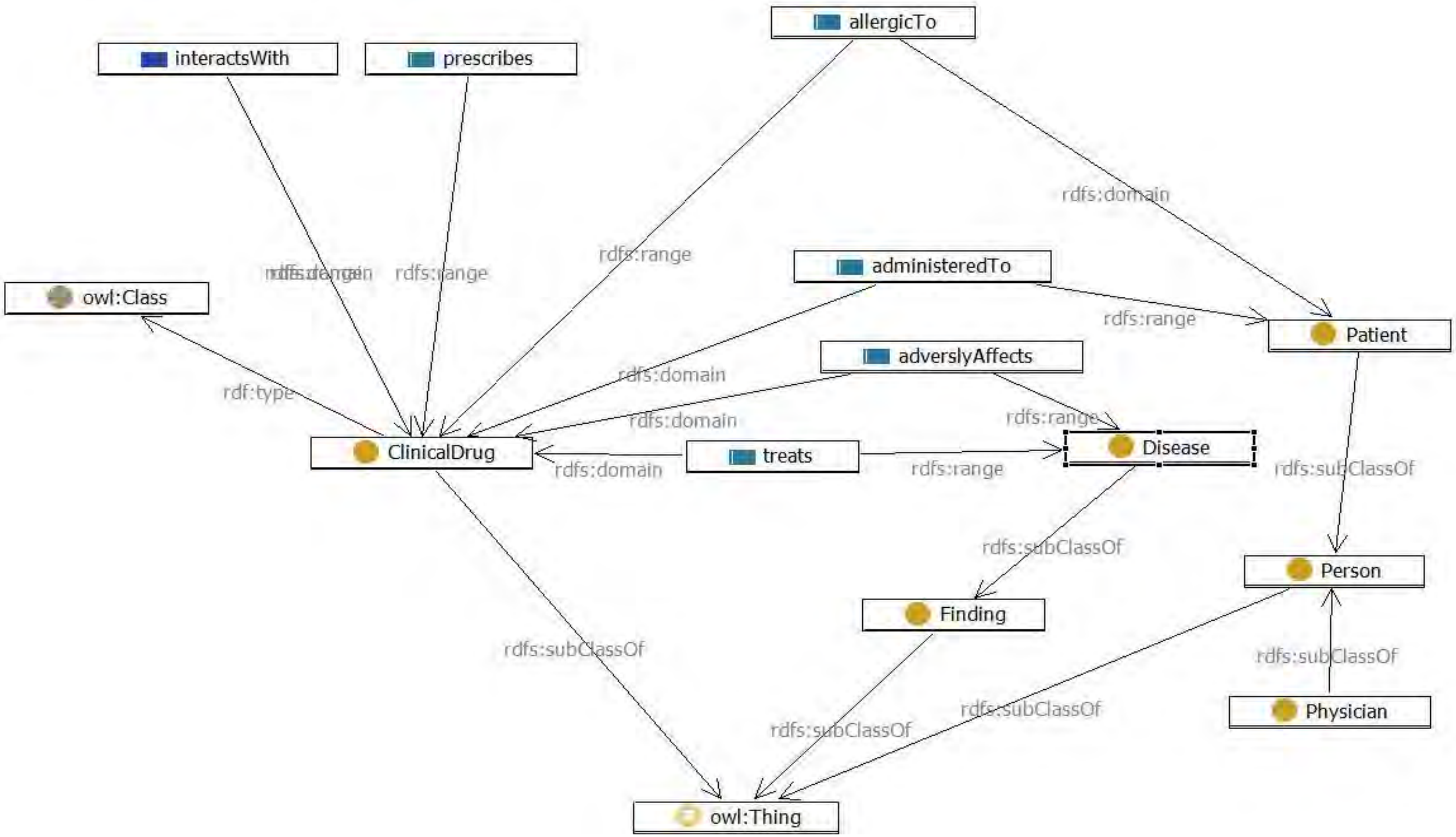

Figure 6-2 an upper level ontology implementation

The third and the last component of the Knowledge base is the restriction rule base. This component is responsible to representing and storing restriction rules. Restriction rules could be defined as two conditions that should not happen at the same time.

An example to a restriction rule could be, an asthmatic patient cannot take aspirin. This should be represented in such a way that it could later be used by the validation system.

To make the restriction rule base work in spite of differing morphological and lexical appearances of concepts, It is proposed that this rule be represented using UMLS preferred Terminology ID. Hence UMLS is used as a standard to represent the rules.

Another design criterion that was put to use here was that the rule store should have the capacity to represent different types of restriction rules. Drug – Drug interaction and Drug – Known medical condition Interaction are the basic types of restriction rules that will be stored. The rule store also needed to be efficient in light of growing number of rules.

Subject	object	Rule type
C0220892	C0332798	CD-DI
C0699141	C0220892	CD-CD
C0004057	C0004096	CD-DI

Table 6-1 Sample Entries from the restriction rule database

The table above contains the basic structure of the restriction rule in the implemented concept validation prototype. Both the subject and the object fields contain concepts with their UMLS concept ID. Hence, C0220892 is the UMLS concept ID for Penicillin. C0332798 is the unique concept ID for open wound.

The rule type in the restriction rules database contains a representation as to how the rule is to be interpreted. CD-DI is for example a rule type that represents adverse interaction between clinical drug and disease. On this prototype, we have only implemented Clinical Drug – Clinical Drug interaction and Clinical Drug – Disease/known condition interaction.

To facilitate the construction of the rule database, we constructed a graphical user interface that could ease the conversion of concepts in the text from to their representation in UMLS concept ID.

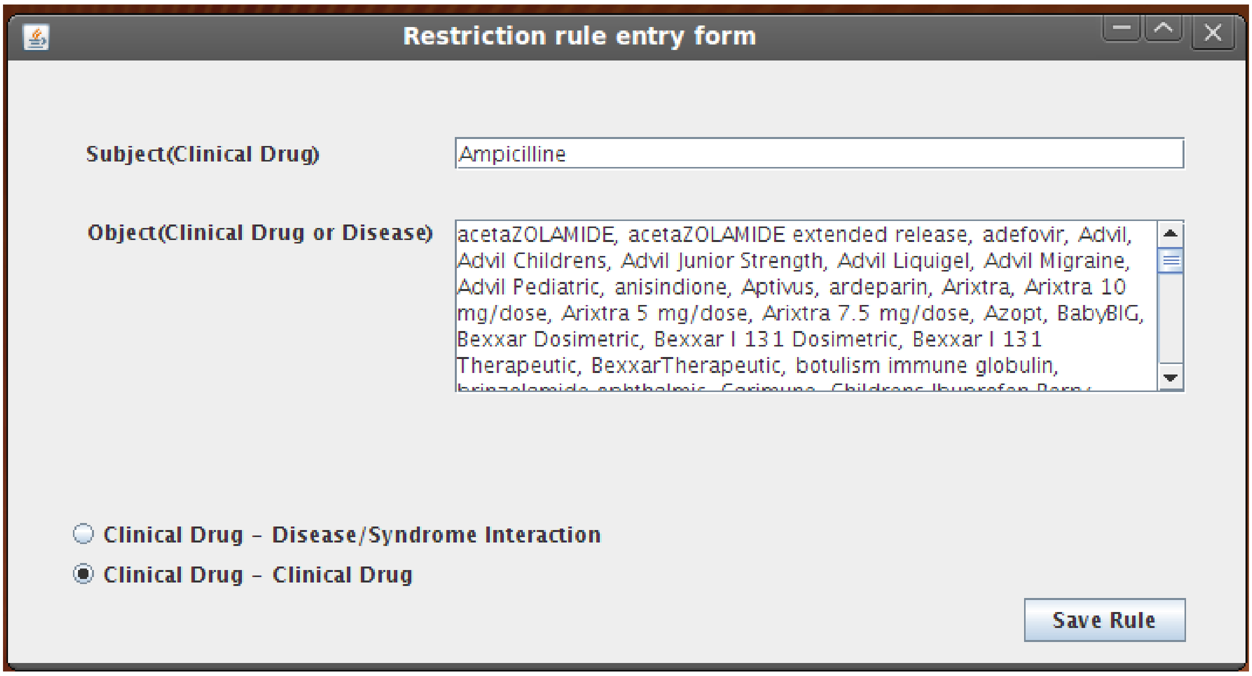

Figure 6-3 Restriction rule entry form

Figure 6.3 shows the restriction rule entry form. This form is used to enter any restriction rule that applies on the subject. Hence each of the entries in the comma separated list in the object field will have restriction type selected.

The rule database is one of the extendable components of the system. An extended implementation to the rule store could be the implementation of an additional type of rule in the rule store. Extending the rule store will also require writing the interpretation of added type of rule in the rule factory.

To develop the sample restriction rule set for the proof of concept prototype, restriction information was collected mainly from drugs.com. Drugs.com is a commercial website that stores and lists drug - drug, and drug - known condition relationships. For the sample, we collected full information about 26 drugs. We were able to identify 2236 restrictions for the 26 drugs.

Out of these 2236 restriction rules, 2139 are clinical drug – clinical drug interactions. The rest of the 97 rules are clinical drug – disease interaction restriction rules.

Initial Filter

The input sentences pass through an initial filtering step that discards out non informative sentences. Even though they are rare, these kinds of sentences could create a noise on the final output of the system. Good example for non informative sentences is question.

This component is also responsible to identify and discarding sentences that are not written about the current patient. From the analysis of the sample documents we have learnt that some sentences in the medical note describe or discuss the medical condition of the family of the patient. In extreme cases, we have noticed that some sentences are written about other patients. Hence this component of the concept validation system is responsible to detect and discard such sentences.

In real world settings, we recognize that reference resolution shall involve further steps. These steps include construction of an indexed database of all subjects. Such a database will be used to recognize co-referent sentences from within large documents. But, as stated in the scope of this document, this work only concentrates on the other components of concept validation system. Under such a setting, the distance and location of sentences can be taken into consideration to facilitate the decision making. Sentences in the same paragraph are more likely to discuss about the same idea than sentences in separate or far apart paragraphs.

Assuming that non co-referent sentences cannot contradict, the system avoids processing them further. In the medical notes case, this work excludes statements that are reported about patient‟s family history. Mainly, these family histories are marked by a parental medical history marker.

The implementation of the prototype automatically assumes that sentences containing words that represent any member of a family as referring to that family member than the patient. It is recognized that this approach is very simplistic and might miss.

Example*:*

> *FAMILY HISTORY: Remarkable for coronary artery disease, stroke, and congestive heart failure*

MetaMap

The unstructured representation of knowledge in the sentences is difficult for the reasoning task we want to perform. Basically, we want to identify 3 types of variables from the plain text; diseases, clinical drugs and their relationship with the current patient. To address this problem we need to find a tool that extracts only important concepts from the text into a standard conceptual representation. The most commonly used tool in the biomedical domain to this end is MetaMap.

After the initial filter is performed, the sentences that are deemed important are passed to MetaMap. MetaMap is a tool created at the national library of medicine to attempt to map words and phrases from any given sentence to concepts represented in UMLS.

MetaMap breaks down the sentences into phrases. For each phrase in the sentence, it returns the mapped concept from UMLS ranked by the mapping strength.

An evaluation of MetaMap by [29] shows that it is not a fool proof system. However it is still the most used tool available to identify biomedical concepts from plain text.

The output of MetaMap is an XML document representing each concept that has found a mapping in the UMLS. The XML document also contains the semantic group of the mapped concept. Hence from the semantic group of the mapped concept, one can learn if that concept is a medication, disease, syndrome or finding.

Given the sentence, “the patient also has a long-standing history of diabetes which is treated with Micronesia daily.” The following is an abstract from the output of the MetaMap.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<MMO>
 <Args>

...

<Mappings Count="1">
      <Mapping>
       <MapNegScore>-1000</MapNegScore>
       <Candidates Count="1">
        <Candidate>
            <UMLSCUI>C0011860</UMLSCUI>
            <UMLSConcept>diabetes</UMLSConcept>
            <UMLSPreferred>Diabetes Mellitus, Non-Insulin-
Dependent</UMLSPreferred>
            <MatchedWords Count="1">
                  <MatchedWord>diabetes</MatchedWord>
            </MatchedWords>
            <STs Count="1">
                  <ST>dsyn</ST>
            </STs>...
         <IsHead>yes</IsHead>
         ...
        </Candidate>

...

</MMO>
```

From the abstracted XML documents, we can learn that the term diabetes is recognized as a terminology in the UMLS. From <UMLSCUI>C0011860</UMLSCUI>, we can learn that the concept that the term is representing is mapped to UMLS concept ID C0011860. We can also make the observation that this Diabetes is semantically classified as <ST>dsyn</ST>. DSYN is a semantic classification for disease and syndrome in the UMLS semantic network.

Ontology Extraction

The XML output of MetaMap is consumed by the ontology extractor. At this stage, ontology of the statements is acquired for further analysis.

For the prototype, we have selected two types of important Information for the validation of concepts. Some of these extractions not necessarily build the information from scratch. Some enhance previously extracted and constructed knowledge about the patient. Naturally, this data could be taken from different parts of different documents.

1. Medical Information: Patient"s Diseases Information, Known Allergies,
2. Drug consumption information: including drug name, drug Intake start date, duration and dosage.

Taking the assumption that the subjected medical note is written all about a patient, the ontology extraction tool initialize a "current patient" object that is of type patient. After creating an ontology instance of "current patient" in the upper level ontology, the ontology extractor component goes on to parse the xml output of MetaMap.

Using XPath expressions, the ontology extraction tool selects all concepts that fall under the disease/syndrome semantic type.

"//Phrase/Mappings/Mapping/Candidates/Candidate[STs/ST='dsyn'/UMLSCUI/text()"

Execution of the above XPATH expression returns all UMLS concept IDs that are of semantic type dsyn. Dsyn is the short form of disease and syndrome in the semantic type in the UMLS semantic categorization.

After all diseases and syndrome mentions in the sentence are identified, the ontology extractor creates an ontology instance of the disease class and labels it with the concept ID of the disease. Note that at this point there is current patient object and an instance of disease in the ontology. According to the analysis on the sample discharge summaries, we have noted that mentions of disease could be to indicate the existence of the disease on the patient, or non-thereof. Before creating the relationship between the disease instance and the current patient, the ontology extractor uses NegEx.

NegEx comes with the regular expression patterns of pre-negations and post-negations. Pre-Negations are negations that come before the negated word

Example:

not cold, ***neither*** asthma *nor* bronchitis.

Post negations are negations that come after the word they negate.

Example:

Influenza was ***ruled out***.

[30] Evaluated the NegEx to other three negation detection algorithms reporting that it has a better or accuracy than the compared approaches.

Hence by using NegEx, the ontology extraction tool tries to determine if there exists a relationship between the disease and the current patient. In cases where there is a relationship, by default it creates “current patient” “Is affected by” “disease” relationship.

Using the same technique used to extract diseases, the ontology extractor also extracts clinical drug information. However, there is one more bit of information that is also on the watch when extracting clinical drugs. The ontology extractor tries to determine if the mention of the clinical drug is to indicate allergic condition of the current patient.

If there is an allergic condition marker in the sentence, the instance of the clinical drug will have an “allergic to” relationship with the patient. Otherwise, the clinical drug will be checked for negation with NegEx. After checking if it is not negated, a “clinical drug” “administered to” “current patient” relationship is created.

Reasoner

The last component in the concept validation architecture is the reasoner. The reasoner is mainly responsible for checking if the restriction rules in the restriction rule base are not satisfied. Hence, this component contains two sub-components.

The rule factory: this is a component of the system that translates the rule from the representation in the rule database to an ontology reasoning format. First this component selects the entries that involve the disease and clinical drugs that have instances in the extracted ontology. This functionality makes it easier for the system to execute and check only the important rules to be checked making the system efficient in light of a big rule database.

The rule factory is one of the extendable features of the concept validation system. Extending the rule database to include new types of rules would require that the interpretation of the new restriction rule type being implemented in this rule factory.

After selecting only the important rules from the database, the rule factory will translate the rules into SPARQL query. It creates a "check list" or rules that might have been violated.

To this end, the rule factory selects out only rules that involve instances of concepts in the extracted ontology.

Example:

```
select ?p
where
{
        ont:C0000043   ont:administeredTo ?p.
        ?p ont:affectedBy ont:C0553434
 }
```

The above example could be read as, select all instances of patient where the patient is affected by concept ID C0553434 and the patient is administered with drug identified by concept ID C0000043.

The rule factory also appends generic restriction rules that apply on all concepts of a certain class. An example to generic restriction rule could be "if a patient is allergic to x, x should not be administered to the patient". This will be represented in SPARQL query and shall be appended to the check list that contains other restriction rules in SPARQL.

Example:

```
select ?p
where
{
        ?p ont:allergicTo ?d.
        ?d ont:administeredTo ?p
}
```

The second sub component of the reasoner is the **Jena reasoner**. Jena is used to execute the SPARQL query emitted by the rule factory. Running the SPARQL query on the extracted ontology will either give result set or it might return null. A returned result set on a set of rule means that that restriction rule has been violated. Hence the reasoner will indicate the existence of conceptual error.

User Interface

A simple graphical user interface was constructed to simplify the usage of the concept validation system.

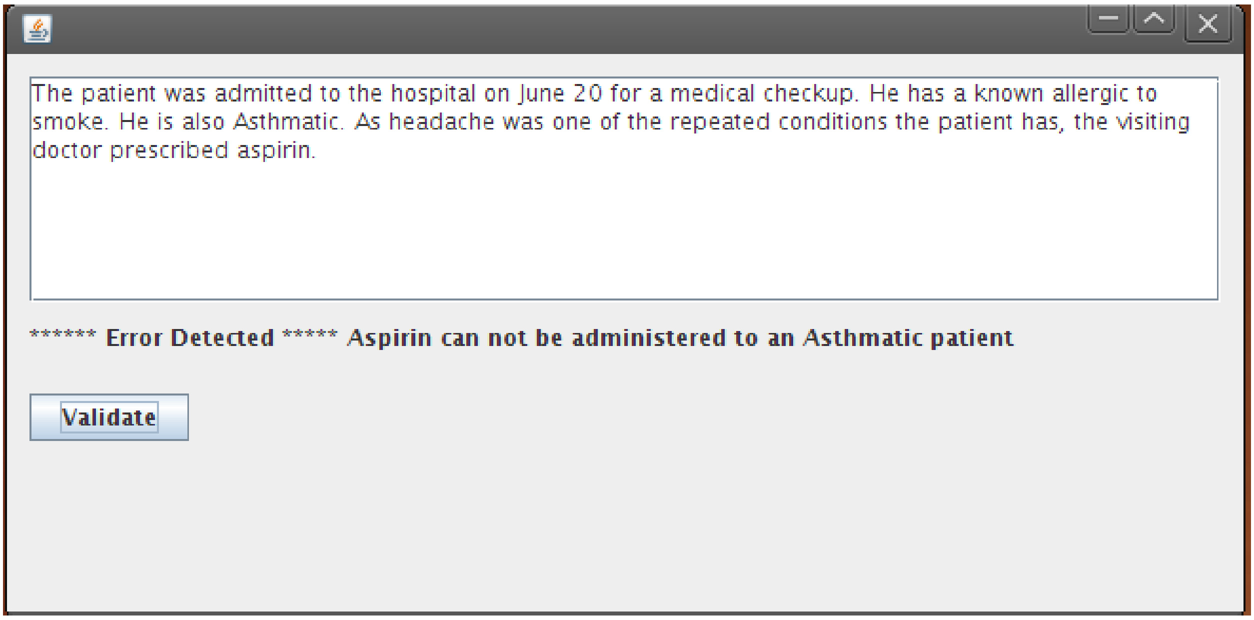

Figure 6-4 Graphical user interface for the concept validation system

Figure 6-4 is a screen shot of the graphical user interface for the concept validation system. The input to the concept validation system is a free text paragraph of medical narration.

Chapter 7. Evaluation

To objectively judge the performance of the developed prototype, an evaluation was performed on the system. To this end, the evaluation was more focused on checking the robustness of the system on using information provided in plain text format. This chapter presents the evaluation procedures followed and the experimental results obtained along with their explanations.

The evaluation of the concept validation system was performed in a 3 steps procedure. The first step in the evaluation procedure was the development of the test set. The test set is a set of medical notes that has to be submitted to the concept validation system. To test the robustness of the system, this test set is prepared with half of the documents having conceptual errors while the other half has no conceptual errors. After the test sets were developed an annotation was made on each test medical notes to mark the existence of conceptual errors or not. The third step of the evaluation process was to experiment with the test sets.

7.1. Development of the test set

For the evaluation of the concept validation prototype, a test set of 75 sample documents were prepared manually. The development of the test set was performed by applying three strategies. The first 50 documents in the test set were prepared by the domain experts without any bias. Half of these 50 documents were intentionally made conceptually erroneous when the other half is made free of conceptual errors.

On the development of the 25 conceptually erroneous test cases, it was made sure that the restriction rules that should apply on the conceptual errors were represented in the rule database. Hence, the evaluation on this test set only indicates the percentage of agreement the system has on represented rules. This means it doesn‟t indicate or measure the exhaustiveness of the rules database. In Annix B, Documents 1 and 2 are examples of this first type test set. The next 25 of the test set is prepared without conceptual errors. This is done to show the false positive identifications the concept validation system identified. A sample of the second type of test set could be found on Annix B, Document 10.

And the last 25 documents were prepared to trick the system. The tricks were mainly injected to mislead the system because of negation or non-co-reference. An example of these test samples is presented on Annix B, Document 56.

7.2. Annotation of the test set

After the preparation of the test documents, an annotation was made on the existence and explanation of the conceptual errors in each of the documents. This annotation was later used to measure the rate of agreement the system has with the expert‟s decision.

7.3. Experimental results

The experiment was conducted by feeding the plain text medical notes from the test set to the concept validation system. The output of the concept validation system was recorded for comparison with the manual annotation.

At this stage, output from the different components of the system was observed for analysis purpose. In cases where the concept validation system missed the right output, attention is paid to the output of the components to find the explanation for why the validation system missed.

Accordingly, table 7-1 below shows the experimental result obtained on the concept validation system.

Document Type	Number of samples	Number of Agreement	Wrong reports
Erroneous	25	20	5
No Error	25	23	2
Trick Documents	25	12	13
Total	75	55	20

Table 7-1 Experimental Results in number of documents

Out of the 75 test case documents, the concept validation system was in agreement with the expert‟s classification on the 55 documents. On 20 out of the 75 documents however the concept validation system either reported error where there is none or reported there is no error where there was one or more errors.

Document Type	Percentage of Agreement	Percentage of Loss
Erroneous	80	20
No Error	92	8
Trick Documents	48	52
Total	73.33333333	26.66666667

Table 7-2 Experimental results in percentage

Table 7-2 above shows in percentage the results of the experiment. For a graphic comparison of where the system performed well, we have presented the percentage of agreement in each of the three types of test sets in figure 7-1 below.

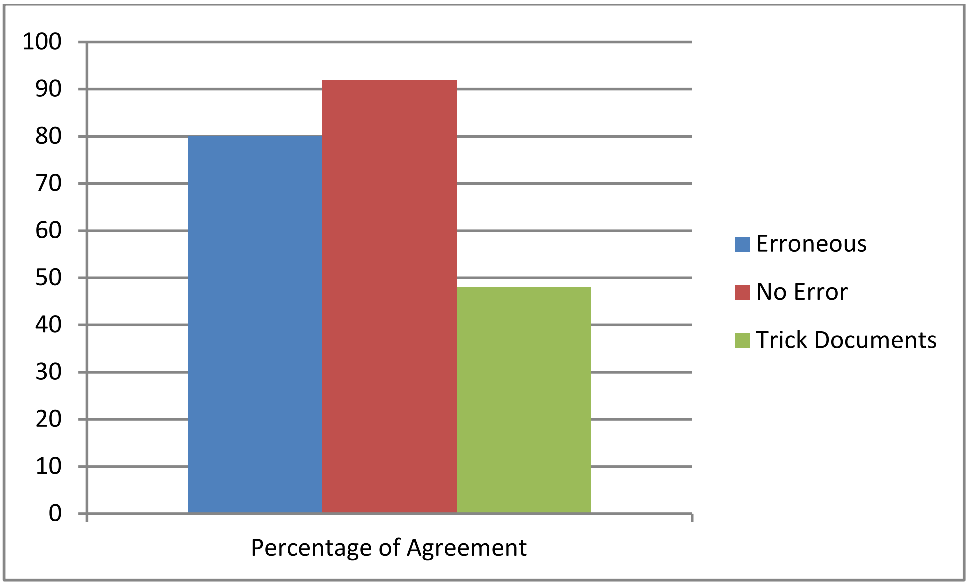

Figure 7-1 Graph of percentage of agreement of the system with expert's classification

As could be clearly seen from Figure 7-1 above, the concept validation prototype was able to be in agreement with the expert‟s decision on 80% of conceptually erroneous documents. Furthermore, the system was in agreement on 92% of documents that were classified as with no conceptual error.

The reason for the relatively higher (80% of agreement as compared to 50% – 60%) rate of agreement observed on Entailment and Contradiction detection system was because medical notes contain text with relatively predefined pattern. For example, as the analysis section showed, disease and medications are mostly discussed about the current patient. Had the case been different, we acknowledge that the rate of agreement would have been affected badly.

The system poorly performed with an agreement rate of 48% on trick sentences that were specifically prepared to trick the system into reporting wrongly. The better performance on the non trick test sets demonstrates that it is possible to explore the pattern in medical notes for the sole task of concept validation.

7.4. Discussions

Generally, the success rate achieved by the concept validation system prototype was encouraging. The loss of precision however, can be attributed to many factors. One the major factors that affected the accuracy of the system was the loss of precision in the MetaMap component. MetaMap was observed mapping the concepts to the wrong corresponding concept.

For example, MetaMap mapped ER in the sentence "He was given aspirin in the ER" To "Estrogen Receptors". ER was supposed to be mapped to Emergency Room. Another good example to the erroneous mapping of MetaMap is the case where MetaMap mapped Micronesia to a geographic location while in the context it was used; Micronesia was a type of medication.

The other class of loss of precision happened with un-represented negation patterns in the NexEx. NegEx was observed to be efficient with straight forward negation expressions. However, negations appear in very complex formats.

> Example: Patient was prescribed with chlorpropamide, aspirin and Penicillin: the medicines with the exception of the third have been administered to the patient.

Loss of precision is also observed because of weakness of the initial filter we implemented. The initial filter only discards one statement with family member marker. However, more sentences could be referring to the same family member after that sentence. Another observation was that the existence of family member markers could not guarantee that the sentence is all about the family member.

> Example: Her mother had never had this condition but she was thought to have allergies to Lovenox.

We have also observed loss of precision on child care cases where there was mention of family member in the sentences only to discuss about the patient. This could be improved by improving the initial filter component to use more robust co-reference resolution algorithm.

The other major variable that affected the precision of the concept validation system was the inability to identify the time frame a given text is referring about. For example, a patient might have taken a given medicine in the past. That medication history is recorded. On another section of the same document, if the patient is about to take a medication that is deemed to be reactive to the medicine the patient used to take, our system doesn"t have any way to detect that the previous medication has been stopped.

Adding this capability would improve the precision of the concept validation system prototype.

Chapter 8. Conclusion

Previous researches have shown that medication errors result in thousands of death every year in the United States alone. Most of these errors are deemed preventable as they occur because of omission of some kind of information. Some of the efforts towards avoiding these errors from the computer usage perspective are usage of computer physician order entry (CPOE). CPOE systems however depend on the entry of structured data.

The very fact that CPOEs depend on structured data renders knowledge represented in plain text medical charts un-usable towards the validation of contraindications and contradictions.

This paper has explored usage of available natural language processing tools and approaches combined with domain specific knowledge towards unlocking information contained in the medical notes. An approach to extract knowledge from the plain text medical notes was developed. Together with the applied background knowledge, a restriction rule data store was also developed. Later, a reasoning component was developed to use the background as well as domain specific restriction rules towards detecting conceptual error.

The developed prototype was evaluated to check the agreement of the system with the good judgment of the domain experts. The outcome of the experiments has shown that it is possible to utilize the patterns of text in medical notes towards making the information available for validation.

The experimental results have shown that loss of precision occurred mainly because of deficiencies in three of the components of the system. The initial filter component of the system had difficulty recognizing parental medical sentences that have no family member mention. This affected the precision of the system on cases that had a sequence of sentences that discussed family medical history. Our experiment has also shown that MetaMap was not able map some concepts from the medical notes to the right corresponding concepts in UMLS. On the other hand, NegEx, The component that recognizes negation patterns failed to recognize complex negation patterns.

Our system depended on the higher probability that sentences in a medical note discuss about a condition or medication about the current patient. Detection of conceptual errors in other notes that have a variety of patterns calls for further research in the area.

8.1. Contribution of the work:

Some of the major contributions of this paper are:

- Compiling a theoretical view towards conceptual errors, contradictions by classifying them by the type of knowledge necessary to validate.
- Demonstrating the possibility of applying domain specific knowledge towards validation of factual error on plain text medical notes.
- Demonstrating a noble architecture that uses existing research artifacts and approaches towards detection of conceptual errors in plain text.
- Demonstrating how information in plain text documents could be unlocked to validation applications. There are many applications that do validation on structured data. Information encoded in plain text documents was useless to such kinds of validation tools.

8.2. Future work

The results of this research have demonstrated that a concept validation system could unlock information encoded in plain text documents towards detection of conceptual errors. However, this work could mainly benefit from integration of different areas of researches. This section lists a brief list of areas of improvements for this research work.

- The concept validation system would be improved by Incorporation of time frame detection approach and algorithm.
- Expanding and refining the concept extraction component to include other variables such as pregnancy status, age, demographics and etc to the validation process.

3. Adding additional rule sets to accommodate the additional variable types.

References

1. Golding, Andrew, and Dan Roth, A Winnow-Based Approach to Context-Sensitive Spelling Correction, Machine Learning 34.1 (1999): 107-130.
2. http://en.wikipedia.org/wiki/Word_processor, last accessed on May 3, 2009.
3. Haigh, T., Remembering the Office of the Future: The Origins of Word Processing and Office Automation. Annals of the History of Computing, IEEE 28.4 (2006): 6-31.
4. Bennett, N. A., He, Q., Chang, C., & Schatz, B. R., Concept extraction in the interspace prototype, Urbana-Champaign, IL: CANIS - Community Systems Laboratory, University of Illinois at Urbana-Champaign, Champaign, IL.
5. Burget, R., Layout Based Information Extraction from HTML Documents." Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on 2 (2007): 624-628.
6. http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, last accessed on June 7, 2009
7. Sanda M. Harabagiu, Andrew Hickl, and V. Finley Lacatusu. Negation, Contrast and Contradiction in Text Processing, 2006, in proceedings of AAAI-06.
8. De Marneffe, Marie-Catherine, Anna N. Rafferty & Christopher D. Manning, Finding contradictions in text. In Proceedings of the Association for Computational Linguistics 2008, 1039–1047. Columbus, OH: Association for Computational Linguistics.
9. Dick Crouch, Cleo Condoravdi, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow, Entailment, intensionality and text understanding. In Graeme Hirst and Sergei Nirenburg, editors, HLT-NAACL 2003 Workshop: Text Meaning, Edmonton, Alberta, Canada, May 31 2003. Association for Computational Linguistics.
10. Adams WG, Mann AM, Bauchner H, Use of an electronic medical record improves the quality of urban pediatric primary care, Pediatrics 2003, 111:626-32.
11. Bleeker, Sacha, Gerarda Derksen-Lubsen, Astrid Van Ginneken, Johan Van Der Lei, and Henriëtte Moll, Structured data entry for narrative data in a broad specialty: patient history and physical examination in pediatrics, BMC Medical Informatics and Decision Making 6 (2006): 107-130.
12. Committee on Quality of Health Care in America and Institute Of Medicine, To Err Is Human: Building a Safer Health System. National Academies Press, 1 edition, April 2000.

13. http://www.ahrq.gov/qual/errback.htm, last accessed on May 10, 2009.
14. Jacquelyn Crane and Frederick G. Crane, Preventing medication Errors in hospitals through a systems approach and technological Innovation: A prescription for 2010,
15. Medical Errors: The Scope of the Problem. Fact sheet, Publication No. AHRQ 00-P037. Agency for Healthcare Research and Quality, Rockville, MD. http://www.ahrq.gov/qual/errors.htm last accessed on June 2, 2009
16. http://www.fda.gov/fdac/features/2000/500_err.html, last accessed on June 2, 2009.
17. http://cme.medscape.com/viewarticle/556487, last accessed on June 2, 2009.
18. Thomas, Eric, and Laura Petersen, Measuring errors and adverse events in health care, Journal of General Internal Medicine 18.1 (2003): 61-67.
19. http://ai-nlp.info.uniroma2.it/te/, last accessed on June 2, 2009
20. Voorhees, Ellen, Contradictions and justifications: Extensions to the textual entailment task, 2008, In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics.
21. Johan Bos , Katja Markert, Recognizing textual entailment with logical inference, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, p.628-635, October 06-08, 2005,
22. M. Tatu and D.Moldovan, A semantic approach to recognizing textual entailment. In Proceeding of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), pages 371–378, Vancouver, Canada.
23. R. Siblini and L. Kosseim, Using Ontology Alignment for the TAC RTE Challenge, Notebook Papers and Results, Text Analysis Conference (TAC-2008),2008,301-309
24. Sibini, Reda and Leila Kosseim, Rodeo: Reasoning over dependencies extracted online. In European Language Resources Association (ELRA) (ed.), 2008, Proceedings of the The 4th Web as Corpus: Can we do better than Google?, a workshop of the Sixth International Language Resources and Evaluation (LREC‟08). Marrakech, Morocco.
25. Bergmair, R, Monte Carlo Semantics: MCPIET at RTE4, In Text Analysis Conference (TAC 2008) Workshop - RTE-4 Track. National Institute of Standards and Technology (NIST). November 17-19, 2008. Gaithersburg, Maryland, USA.
26. http://www.mtsamples.com, last accessed on July 8, 2009.
27. http://www.nlm.nih.gov/pubs/factsheets/umls.html, last accessed on July 10, 2009.
28. http://www.nlm.nih.gov/pubs/factsheets/, last accessed on July 10, 2009.

29. Wanda Pratt and Meliha Yetisgen-Yildiz, A Study of Biomedical Concept Identification: MetaMap vs. People, AMIA Annu Symp Proc. 2003; 2003: 529–533.
30. S. Goryachev, M. Sordo, Q. T. Zeng, and L. Ngo. Implementation and evaluation of four different methods of negation detection. Technical report, DSG.
31. Dekang Lin and Patrick Pantel, Discovery of inference rules for question-answering. Natural Language Engineering, 7(4):343–360.
32. C. Zirn, V. Nastase, and M. Strube, Distinguishing between instances and classes in the wikipedia taxonomy. In Proceedings of the European Semantic Web Conference (ESWC), volume 5021 of Lecture Notes in Computer Science, pages 376{387, Berlin, Germany, 2008. Springer.
33. Van Ginneken AM, Structured data entry in ORCA: the strengths of two models combined. Proc AMIA Annu Fall Symp. 1996:797–801

Annex

Annex A: High Level Ontology:

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://localhost/patient.owl#"
    xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xml:base="http://localhost/patient.owl">
  <owl:Ontology rdf:about="">
    <owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Created with TopBraid Composer</owl:versionInfo>
  </owl:Ontology>
  <owl:Class rdf:ID="ClinicalDrug">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>
  <owl:Class rdf:ID="Finding">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>
  <owl:Class rdf:ID="Physician">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Person"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Person">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>
  <owl:Class rdf:ID="Disease">
    <rdfs:subClassOf rdf:resource="#Finding"/>
```

```
  </owl:Class>
  <owl:Class rdf:ID="Patient">
    <rdfs:subClassOf rdf:resource="#Person"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="adverslyAffects">
    <rdfs:domain rdf:resource="#ClinicalDrug"/>
    <rdfs:range rdf:resource="#Disease"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="administeredTo">
    <rdfs:range rdf:resource="#Patient"/>
    <rdfs:domain rdf:resource="#ClinicalDrug"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="allergicTo">
    <rdfs:domain rdf:resource="#Patient"/>
    <rdfs:range rdf:resource="#ClinicalDrug"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="prescribes">
    <rdfs:domain rdf:resource="#Physician"/>
    <rdfs:range rdf:resource="#ClinicalDrug"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="treats">
    <rdfs:domain rdf:resource="#ClinicalDrug"/>
    <rdfs:range rdf:resource="#Disease"/>
  </owl:ObjectProperty>
  <rdf:Property rdf:ID="interactsWith">
    <rdfs:domain rdf:resource="#ClinicalDrug"/>
    <rdfs:range rdf:resource="#ClinicalDrug"/>
  </rdf:Property>
</rdf:RDF>
```

Annex B: Sample Test Cases:

Document: 1

Text:

PLAN: I spent about 30 minutes with the patient discussing treatment options. I do believe that her moods would greatly benefit from hormone replacement therapy. As we discovered she has anorexia. We will try starting her back on Wellbutrin XL 150 mg daily. She may increase to 300 mg daily after three to seven days. Samples provided initially. If she is not obtaining adequate relief from medication alone, we will then suggest that we explore the use of hormone replacement therapy. I also recommended increasing her exercise. We will also obtain some screening lab work including CBC, UA, TSH, chemistry panel, and lipid profile. Follow up here in two weeks or sooner if any other problems. She is needing her annual breast exam as well.

Explanation: The patient has anorexia and she is prescribed some wellbutrin XL. It is known that wellbutin XL shouldn‟t be administered to an anorexia patient.

Document: 2

Text:

PLAN: I prescribed Neurontin 100 mg dispensed 30 with five refills one to two p.o. q.h.s. p.r.n. peripheral neuropathy. I offered Anodyne physical therapy, but she was not interested yet at this point. I suspect that her peripheral neuropathy may be due to her essential thrombocythemia and kidney disease. We did send her to lab for a CBC due to her anemia and essential thrombocythemia and she needs sed rate due to her peripheral neuropathy, ferritin due to her anemia, and Hemoccult cards x 3 due to anemia. She needs a DT immunization. Recheck with me in about three months. I refilled her Ambien 5 mg at h.s. for one year. She may get a flu shot next month.

Explanation: the patient has kidney disease and was prescribed Neurontin

Document: 10

Text:

COURSE IN THE ED: Patient arrived and was placed on monitors. An IV had been placed in the field and labs were drawn. X-rays of the C spine show no fracture and I've removed the C-collar. The lacerations were explored and no foreign body found. They were irrigated and closed with simple interrupted sutures. Labs showed normal CBC, Chem-7, and U/A except there was moderate protein in the urine. The blood alcohol returned at 0.146. A banana bag is ordered and his care will be turned over to Dr. G for further evaluation and care.

Explanation: This sample case doesn"t have any conceptual errors in it.

Document: 11

Text:

HISTORY OF PRESENT ILLNESS: The patient is a two-and-a-half-month-old male who has been sick for the past three to four days. His mother has described congested sounds with cough and decreased appetite. He has had no fever. He has had no rhinorrhea. Nobody else at home is currently ill. He has no cigarette smoke exposure. She brought him to the emergency room this morning after a bad coughing spell. He did not have any apnea during this episode. Patient was administered with chlorpropamide, aspirin and Penicillin.

Explanation: Patient was administered with chlorpropamide, aspirin and Penicillin. These medicines shouldn"t be administered to the same patient at the same time.

Document: 56

Text:

HISTORY OF PRESENT ILLNESS: The patient is a two-and-a-half-month-old male who has been sick for the past three to four days. His mother has described congested sounds with cough and decreased appetite. He has had no fever. He has had no rhinorrhea. Nobody else at home is currently ill. He has no cigarette smoke exposure. She brought him to the emergency room this morning after a bad coughing spell. He did not have any apnea during this episode. Patient was prescribed with chlorpropamide, aspirin and Penicillin: the medicines with the exception of the third have been administered to the patient.

Explanation: This paragraph was a trick paragraph prepared to test how the system will behave with complex negation patterns. The trick here is that the administration of Penicillin was negated. Had it not been negated, this paragraph would have a conceptual error. The system wrongfully detected a conceptual error.

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials for the thesis have been duly acknowledged.

Elias Muluneh

This thesis has been submitted for examination with my approval as an advisor.

Fitsum Admasu (Ph. D.)

Addis Ababa, Ethiopia

October 2009