

# Challenges in Understanding Clinical Notes: Why NLP Engines Fall Short and Where Background Knowledge Can Help

Sujan Perera  
Ohio Center of Excellence in  
Knowledge-enabled  
Computing (Kno.e.sis)  
Wright State University,  
Dayton OH, USA  
sujan@knoesis.org

Amit Sheth  
Ohio Center of Excellence in  
Knowledge-enabled  
Computing (Kno.e.sis)  
Wright State University,  
Dayton OH, USA  
amit@knoesis.org

Krishnaprasad Thirunarayan  
Ohio Center of Excellence in  
Knowledge-enabled  
Computing (Kno.e.sis)  
Wright State University,  
Dayton OH, USA  
tkprasad@knoesis.org

Suhas Nair  
ezDI, LLC  
suhas.nair@ezdi.us

Neil Shah  
ezDI, LLC  
neil@ezdi.us

## ABSTRACT

Understanding of Electronic Medical Records(EMRs) plays a crucial role in improving healthcare outcomes. However, the unstructured nature of EMRs poses several technical challenges for structured information extraction from clinical notes leading to automatic analysis. Natural Language Processing(NLP) techniques developed to process EMRs are effective for variety of tasks, they often fail to preserve the semantics of original information expressed in EMRs, particularly in complex scenarios. This paper illustrates the complexity of the problems involved and deals with conflicts created due to the shortcomings of NLP techniques and demonstrates where domain specific knowledge bases can come to rescue in resolving conflicts that can significantly improve the semantic annotation and structured information extraction. We discuss various insights gained from our study on real world dataset.

## Categories and Subject Descriptors

I.7.m [Computing Methodologies]: Document and Text Processing  
Miscellaneous

## Keywords

Natural Language Processing, Negation Detection, Knowledge Base

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
*DARE'13*, Nov. 1, 2013, San Francisco, CA, USA.  
Copyright 2013 ACM 978-1-4503-2425-0/13/11\$15.00  
<http://dx.doi.org/10.1145/2512410.2512427> ...\$15.00.

## 1. INTRODUCTION

The Patient Protection and Affordable Care Act has recently introduced a number of new regulations and guidelines with the goal of improving healthcare outcomes by 2015. A key initiative has been the promotion of management and exploitation of Electronic Medical Records(EMRs). However, processing EMRs accurately requires addressing and overcoming several technical and non-technical challenges. Specifically, the large volume of data that resides in hospital databases, 80% of which is unstructured, should be converted into machine understandable form preserving semantics, to enable automatic analysis.

Conversion of unstructured components of EMRs, referred to here as clinical notes, to a machine understandable form requires multiple tasks including: 1) identification of the domain entities (i.e., disorders, symptoms, procedures, findings and medications), 2) annotation of domain entities with standard vocabularies, 3) understanding different linguistic and semantic components of the sentence construction such as negation, conditioning and uncertainty, 4) detection of whether the document discusses present condition<sup>1</sup> or past condition of the patient, 5) detection of whether the document discusses patient's condition or his/her family condition.

Natural Language Processing(NLP) techniques have been developed to perform these tasks with reasonable accuracy. For example, cTAKES[5], MedLEE[3] and MetaMap[1] are among the major NLP engines tuned to perform the above mentioned tasks for clinical notes. Even though NLP engines that perform the above mentioned tasks have significantly improved their outputs<sup>2</sup>, they still exhibit significant shortcomings. For example, consider an EMR document that has the following two sentences<sup>3</sup>

<sup>1</sup>A condition includes disorders and symptoms

<sup>2</sup>NLP output is machine understandable structured output, e.g., an XML document

<sup>3</sup>All examples are from real world EMR documents

1. *He is not having any symptoms of chest pain or exertional syncope or dizziness.*
2. *I advised him that if he experiences chest pain, shortness of breath with exertion or dizziness or syncopal episodes to let us know and we can do appropriate workup.*

The first sentence clearly states that the patient does not have *chest pain*, *exertional syncope* and *dizziness* and NLP engine understands this correctly. The second sentence recommends actions **if** he experiences these conditions. But unfortunately, current NLP engines do not understand that the second sentence has a conditional statement and incorrectly outputs that the patient has *chest pain*, *shortness of breath*, *dizziness* and *syncope*. This leads to a conflicting scenario since the NLP output states both the presence and absence of the same condition for the same patient at the same time. This *conflict* must be resolved before a high-fidelity and meaningful analysis and inference can be made for clinical applications such as Clinical Document Improvement(CDI) and Computer Assisted Coding(CAC).

We analyzed many conflicting instances such as the one identified above with the clinical domain and health informatics experts and recognized that contextual information(such as the medication taken by the patient and other conditions that the patient has) can be used to resolve these conflicts. For example, the absence of relevant medications for *chest pain* in the above example provides evidence that the patient does not have *chest pain* while the presence of such medications suggests the opposite. Motivated by these observations we try to imitate the thought process of domain experts to resolve the conflicts in the NLP output. This approach requires a reasonable amount of domain knowledge in order to attempt to partially imitate the reasoning of a domain expert in resolving or avoiding the conflicts. In order to factor that in, we built a comprehensive knowledge base by mining EMR documents and by leveraging the open knowledge available in different formats. Our evaluation demonstrates that the knowledge bases can play a crucial role in understanding EMRs by complementing NLP algorithms and overcoming some of their limitations.

The rest of the paper is organized as follows, Section 2 discusses the problem in detail with real world examples. Section 3 describes the knowledge base in detail. The implementation is discussed in Section 4 and the evaluation is presented in Section 5. Section 6 concludes with a summary of our results and intended future work.

## 2. THE PROBLEM

EMRs use complex natural language constructs that are easy to write and interpret by humans with common sense reasoning and working knowledge of the domain, but hard to process and make sense of automatically. That is, people have less problem understanding what is stated in the EMRs and can resolve apparent conflicts easily, but machines have a hard time understanding the stated facts especially in the presence of ambiguity. The following list contains linguistic and semantic components of the natural language statements that should be detected in order to understand the EMRs.

- Negated statements - The statements that convey the absence of domain concepts.

e.g., Patient has **no** angina, **no** orthopnea, PND, or lower extremity edema.

- Conditional statements - The statements that convey possible future status.  
e.g., I advised him that **if he experiences** any chest pain, shortness of breath with exertion or dizziness or syncopal episodes to let us know and we can do appropriate workup.
- Uncertain statements - The statements that convey some doubt about the patient status.  
e.g., She is **not sure** if she is just depressed or not.
- Statements about patient's medical history.  
e.g., When the patient **was last seen in the office**, he **was complaining** of fairly significant chest discomfort.
- Statements about patient's family.  
e.g., Also, several **family members** with high blood pressure.

Current NLP engines cannot properly understand the above exemplified components of the sentences and associate them with relevant domain entities(i.e., **no** with *angina*, *orthopnea*, *PND* and *lower extremity edema* in the first sentence). These shortcomings cause conflicts to be present in the output of NLP engines that are widely used for processing clinical notes.

Here we will demonstrate a number of scenarios where current negation detection algorithms fail to understand the semantics of the sentences. As we will show with the examples, although some of these shortcomings are due to pure NLP issues, there are complex situations where it needs a much more sophisticated solution that complements current NLP techniques.

Example 1:

The original sentence: *"the patient denies any chest discomfort, shortness of breath, orthopnea, paroxysmal nocturnal dyspnea, palpitations, or lower extremity edema."*

According to the above sentence, the patient has none of the conditions mentioned, but NLP output states that the patient has *palpitations* and *lower extremity edema*. This is mainly because the negation detection algorithm is designed in a way that the effect of negation is propagated to predefined number of entities to the right and to the left side of negation indicating term('denies').

Example 2 and Example 3 illustrate two instances where the NLP algorithm incorrectly associates the negation indication term to the domain entity. The first example misinterprets the semantics, while the second example ultimately turns out to be correct.

Example 2:

The original sentence: *"I do not have an explanation for this dyspnea."*

The sentence conveys that doctor was unable to find an explanation for *dyspnea*, hence implicitly says the patient has *dyspnea*. But the NLP algorithm incorrectly associates 'not' with *dyspnea*. This example shows that the algorithm needs to have a deeper understanding of the words in order to derive implicit semantics.

Example 3:

The original sentence: *"there was no evidence of ischemia."*

The term “no” is associated with “evidence”. The current NLP algorithm incorrectly associates “no” with *ischemia*, but it turns out to interpret the semantics of this particular example properly because the doctor interprets the absence of evidence as absence of such condition (according to closed world reasoning).

The first example showed that the NLP algorithm was not able to associate the negation indicator with the correct entities, while second example shows that even if the algorithm associates the negation indicator with the proper term, it does not interpret the semantics of the sentence properly. Hence, it requires a more sophisticated solution to accompany NLP to understand the semantics of such sentences.

Although the above examples illustrate only the complex scenarios for negation detection algorithms, similar instances exist for other aspects of the sentences as well (e.g., conditional and uncertainty). The failure to understand the semantics of the sentences leads to incorrect interpretations of patient status. It is hard to detect such misinterpretations if there is only one mention of a particular condition in the EMR. But when there are multiple mentions of the same condition within the document and NLP misunderstands one such instance, it leads to conflicting scenarios and NLP output reflects this conflict as illustrated earlier. We have found 620 such conflicts within 3172 EMR documents parsed by cTAKES engine. The corresponding error for applications that consume the output of an NLP engine (e.g., CDI or CAC) would be considered unacceptable in most cases. Our objective is to resolve these conflicts created due to different reasons by using contextual information available in the EMR document. Specifically, we will demonstrate that these conflicts can be resolved with the help of domain specific knowledge bases.

### 3. THE KNOWLEDGE BASE

We built a comprehensive knowledge base for the healthcare domain by mining EMR documents and leveraging the open knowledge available in different formats. The knowledge base is expressed in the Resource Description Framework Schema (RDFS) language. The primary knowledge source for the knowledge base is the Unified Medical Language System (UMLS) [2]. The UMLS consists of key domain concepts that we are interested in and their hierarchical relationships. We used the hierarchical relationships present in the UMLS for disorders, symptoms and findings. However, the medication hierarchy is not explicitly present in UMLS. Instead, UMLS contains *tradenamename\_of* relationships which links different brands of the same medication to its generic medication. Hence, we used *tradenamename\_of* relationship in order to build the hierarchical relationships among the medications. Specifically, all the different brands of the same medication are added as sub-classes of the generic medication.

Our approach requires knowledge about the relationships between disorders and symptoms (*'symptom A' is\_symptom\_of 'disorder B'*) and the relationship between disorders/symptoms and medications (*'disorder B' is\_treated\_with 'medication C'*) to resolve conflicts in NLP output<sup>4</sup>. While UMLS provides the required coverage in terms of domain concepts and their hierarchical relationships, it only contains a relationship between disorders/symptoms and medications (i.e.,

<sup>4</sup>The experiments in this paper use only the relationships between disorder/symptom and medications

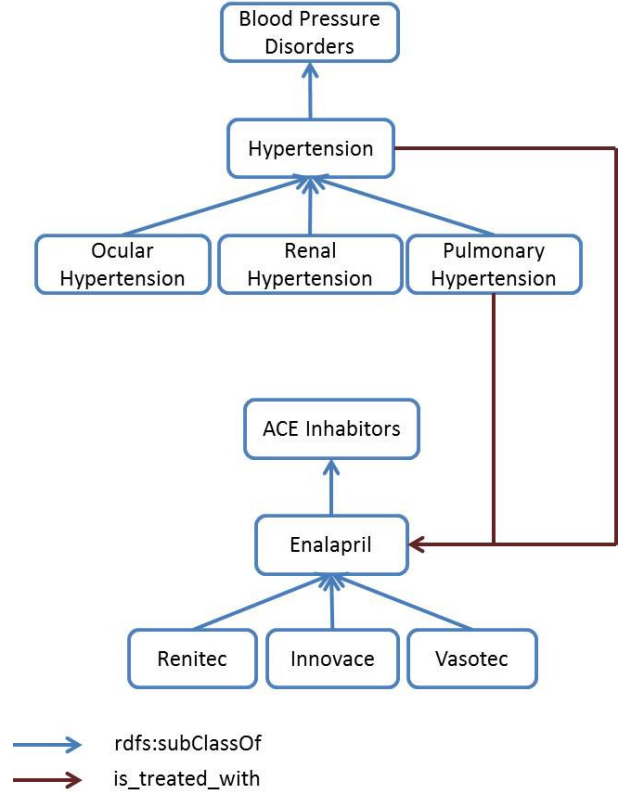


Figure 1: A Snippet from Knowledge Base

Concept Type	Number of Concepts
Condition (disorder or symptom)	125778
Medication	298993
Findings	172230
Procedures	262360

Table 1: Number of Concepts from each type of Domain Entities

Relationship Type	Number of Relationships
<i>is_symptom_of</i>	8299
<i>is_treated_with</i>	41182

Table 2: Number of Relationships among Concepts

*is\_treated\_with*). It lacks, for example, the knowledge about the associations of disorders with the symptoms. Hence, we developed a technique to mine the relationship between disorders and symptoms from EMRs[4]. Effectively, we added the *is\_treated\_with* relationship to our knowledge base from UMLS and the *is\_symptom\_of* relationship by mining EMR documents. Our knowledge base was further enriched by using two Web resources<sup>5</sup>. We decided to add the knowledge in these two Web resources after conducting a careful assessment about the quality of stated facts. Table 1 and Table 2 contains the statistics about a recent version of our knowledge base, which is subject to grow further. Figure 1 shows a snippet from the knowledge base.

## 4. IMPLEMENTATION

We choose cTAKES[5], an open source NLP engine under the Apache License, as our NLP engine to parse unstructured EMRs. The input to cTAKES is an unstructured EMR document and it outputs an XML document where nodes in the XML document represent different semantic types. Figure 2 shows an XML element *condition*<sup>6</sup> which describes a disorder found in the EMR. As shown in the example, cTAKES annotates each entity found in EMR with UMLS and SNOMED vocabularies via ‘code’ and ‘cui’ attributes and assigns values to the following attributes associated with each entity:

- *polarity* - This attribute can take values 0 or -1. -1 indicates that the entity is negated and 0 indicates it is not negated.
- *uncertainty* - This attribute can take four values<sup>7</sup>. Value 0 indicates patient’s current status, value 1 indicates patient’s history, value 2 indicates the patient’s family history and value 3 indicates the uncertain status.
- *conditional* - This attribute can take values either *true* or *false*. If the sentence discusses the entity in a conditional sense(e.g., ‘if pain persist, take tylenol’), it takes value true, else it takes value false.

The annotations of Figure 2 are interpreted for the disorder “atrial fibrillation” as indicating a non-conditional (conditional=“false”), the presence of disorder (polarity=“0”) right now(uncertainty=“0”).

To identify the conflicts, we take the XML document as the input to our method and search for *condition* nodes with the same ‘cui’ value, and ‘0’ as the uncertainty value but with different polarity values. These instances are identified as conflicts, because they state that the patient has particular condition and does not have it at the same time. We then collect the contextual information about that condition to assist with the disambiguation. Currently we collect only related medications. Although EMR documents

<sup>5</sup>The two Web resources are,  
1. <http://www.healthline.com>  
2. <http://www.druglib.com>

<sup>6</sup>A conditional element can contain either disorder or symptom.

<sup>7</sup>Although this attribute is supposed to indicate the uncertainty of the concept, only one value indicates about uncertainty. The other 3 values indicate whether the sentence discuss patient’s current condition or his/her medical history or his/her family history.

```
<condition value="Atrial Fibrillation"
  code="49436004:SNOMED"
  uncertainty="0" polarity="0"
  conditional="false" cui="C0004238"
  tui="T046"/>
```

Figure 2: XML element describes *Atrial Fibrillation* in cTAKES output

contain a lot of unstructured text, it has a semi-structured section, which contains information about the medications taken by the patient in a bulleted list. Since this section is a bulleted list and does not contain sentences, NLP engines can identify medications and populate their attributes (polarity, uncertainty, and conditional values) with great accuracy. This motivates us to use medications as reliable contextual information. We use our knowledge base to associate and confirm the applicability of medication to treat a particular condition, to resolve the conflict. We leverage the hierarchical relationships that exist in the knowledge base to identify the related medications for the particular disorder/symptom. For example, a particular patient might be taking *zocor* which is a trade name of *simvastatin* (*zocor* is a child of *simvastatin* in the knowledge base) and *simvastatin* is a type of *statin*. *Statins* are used to treat *high blood cholesterol levels*. Further, this relationship (*high blood cholesterol level is\_treated\_with statin*) is present in our knowledge base. Since we can infer that *zocor* is a *statin* by using the hierarchical relationships, we can associate *zocor* with *high blood cholesterol level*. If we find such associations between medications in EMR and conflicted disorder/symptom, we can resolve the conflict by assigning ‘0’ to the polarity value, otherwise we assign ‘-1’ to the polarity value.

## 5. A USE CASE

This section elaborates our approach with two real EMR documents. The first document illustrates how we resolve the conflict when a patient actually has the disorder, and the second document illustrates how we resolve the conflict when the patient does not have the disorder. The first scenario is created by shortcomings associated with negation detection in the NLP engine, while the second scenario is due to the failure of the NLP engine to understand that the sentence discusses the possibility of having a disorder/symptom and is uncertain.

The selected EMR for the first illustration has a conflict on *coronary artery disease*. The patient has *coronary artery disease* and it is mentioned clearly in his diagnosis list. However, the EMR also has the following sentence in an unstructured portion that the NLP engine is unable to interpret correctly, and outputs ‘-1’ for the *polarity* of *coronary artery disease*.

“Send for carotid duplex to rule out carotid artery stenosis given his risk factors and underlying coronary artery disease.”

According to the semantics of the sentence, the patient has *coronary artery disease* and the term ‘rule out’ is applied to

*carotid artery stenosis*, but NLP engine associates ‘rule out’ with *coronary artery disease* resulting in an incorrect value for the polarity attribute. Hence the generated XML document contains two *condition* elements one with *polarity* ‘0’ and the other with ‘-1’ and both have ‘0’ as their *uncertainty* value. This triggers our method to collect the context for *coronary artery disease* within the EMR.

The following set of medications were found in the EMR and identified as treating the *coronary artery disease* with the help of the knowledge base.

- Ramipril
- Nitroglycerin
- Enalapril
- Zocor
- Warfarin
- Aspirin

This set of medications helps us to confirm that the patient is more likely to have *coronary artery disease*.

The second EMR document has a conflict on *lower extremity edema*. It contains the following two sentences. “*Extremities : Warm and dry. No clubbing or cyanosis. No lower extremity edema.*” “*I have advised the patient on the side effect of potential lower extremity edema.*”

Clearly the first sentence states that the patient does not have *lower extremity edema* and the second sentence discusses the possibility of *lower extremity edema* as a side effect. But the NLP engine does not understand the semantics of the second sentence and interprets it as patient currently has *lower extremity edema*. Further examination of the EMR document with the help of the knowledge base did not indicate any medication that may treat *lower extremity edema*. This observation suggests that the patient is unlikely to have *lower extremity edema* as stated in the EMR.

## 6. EVALUATION

We provide a preliminary evaluation of our method by using real world data sources to see if it holds the expected promise. Information presented here is within the constraints and guidelines of the IRB protocol involving the use of the limited data set. We selected 25 documents from our corpus for the evaluation which have conflicts in their generated structured versions(XML documents). The XML documents were generated by parsing EMR documents through cTAKES NLP engine. There were 32 conflicting instances within these 25 documents. Table 3 summarizes the results of conflict resolution using contextual information.

The accuracy of the experiment is defined as,

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is true positives, TN is true negatives, FP is false positives and FN is false negatives.

The accuracy for this experiment was 71.87%. We will discuss three important observations we can make from our results.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	18	6
	Negative	3	5

**Table 3: Results of Conflict Resolution. Label ‘positive’ means patient has particular condition and ‘negative’ means patient does not have particular condition.**

### (1) False negatives are for common symptoms:

The 4 out of 6 instances categorized incorrectly as negative(false negative) are instances where our approach tries to resolve conflicts involving common symptoms. Within those 4 instances, *headache* and *obesity* appear once each and *shortness of breath* appears twice. Doctors may not always prescribe medications to cure these kinds of symptoms as they are very generic in nature and can co-occur with many diseases. As such, doctors prefer to rather cure the condition associated with these symptoms using very specific medications addressing the ultimate cause. Hence our method was not able to categorize these instances as belonging to the positive class even though the patient has these symptoms.

### (2) False positives are based on common medications:

We noticed that *aspirin* is the only medication that appeared in collected contextual information for all 3 instances incorrectly categorized as positive(false positives). So our method categorizes these negative instances as positive based on the evidence that the patient takes *aspirin*. *Aspirin* is very a generic medication and it can provide minimal conclusive information about the patient’s conditions. This observation illustrates the need for ranking of the domain relationships based on specificity, i.e., the relationship between *metoprolol* and *hypertension* is more specific than the relationship between *aspirin* and *hypertension*, because if the patient takes *metoprolol*, it provides strong evidence that the patient is a *hypertensive* patient than a patient who takes *aspirin* even though both medications are prescribed to *hypertensive* patients. This type of ranking will help to ignore evidences which provide minimal conclusive information about the patient’s condition and help to improve the results by eliminating false positives.

### (3) Conflicts over major disorders can be resolved with good accuracy:

Our approach has been able to resolve conflicts over major disorders with good accuracy. There were 17 conflicting instances within the total of 32 on major disorders and the following list contains the frequency of each disorder.

- 5 instances of *coronary artery disease*.
- 4 instances of *atrial fibrillation*.
- 2 instances of *peripheral vascular disease*.
- 1 instance each from *ischemia*, *cardiomyopathies*, *coronary heart disease*, *arthritis*, *ischemic cardiomyopathy*, and *aortic valve stenosis*.

We were able to resolve 15 out of 17 conflicts correctly. Since these are major conditions and patients will have to

face more serious problems if they do not treat these conditions well, they will be taking medications to cure or control the condition. This fact is reflected in the EMRs and helps us in resolving conflicts.

## 7. CONCLUSION AND FUTURE WORK

This paper demonstrates shortcomings of the state-of-the-art NLP algorithms used to understand unstructured EMRs and shows the need for complementary techniques for accurate interpretation of documents. As shown in the examples, there are instances which require much deeper understanding of the words and some common sense reasoning for better interpretation. Also, we proposed a method to resolve conflicting instances created due to shortcomings of NLP engines for clinical data processing with the help of comprehensive knowledge base. Our evaluation shows that this method can resolve the conflicting instances over major disorders with great accuracy and provide insights to improve the approach.

While we have demonstrated results of the proposed method in conflict resolution, we believe the knowledge-based approach can be used to detect and address many parsing issues in NLP algorithms. For example, although NLP algorithms detect that a patient does not have a particular disorder/symptom(not a conflicting instance), if the contextual information seems to strongly suggest otherwise, this can trigger a need for better analysis of the original sentences.

As the next step, we will work on ranking the relationships based on their specificity. We believe it can be done by analyzing the number of relationships each concept participates in and the nature of the participating concepts. A more comprehensive evaluation will also be shared in near future.

## 8. REFERENCES

- [1] Alan R Aronson. Metamap: Mapping text to the umls metathesaurus. *Bethesda, MD: NLM, NIH, DHHS*, 2006.
- [2] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270, 2004.
- [3] Carol Friedman, Philip O Alderson, John HM Austin, James J Cimino, and Stephen B Johnson. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174, 1994.
- [4] Sujan Perera, Cory Henson, Krishnaprasad Thirunarayan, Amit Sheth, and Suhas Nair. Data driven knowledge acquisition method for domain knowledge enrichment in the healthcare. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 1–8, October 2012.
- [5] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.